Documentation 8.0

ZABBIX

29.10.2026

Contents

Lietotāja rokasgrāmata **5**

Laipni lūdzam Zabbix dokumentācijā **5**

Sāciet darbu ar Zabbix . . . 5
Zabbix Cloud . . . 5
Mobilā lietotne . . . 5
Izstrādātāju centrs . . . 5
Kopiena un citi resursi . . . 5
Instalēšana un pirmie soļi . . . 6
Lietotāja saskarnes instalēšana . . . 6
Ātrās darba sākšanas ceļveži . . . 11
Instalēšana: papildu rokasgrāmatas . . . 85
Atjaunināšana . . . 164
Zināmās problēmas . . . 180
Prasības . . . 188
Kas jauns un jaunināšanas piezīmes: 8.0 . . . 198
Veidnes . . . 203
Multivides veidi . . . 205
Vienumi . . . 206
Spraudņi . . . 206
Atklāšana un automātiskā reģistrācija . . . 206
Zema līmeņa atklāšana . . . 207
Datu bāzes . . . 207
Procesi . . . 207
Drošība . . . 208
Autentifikācija . . . 208
Logdaļas . . . 209
Lietotāja saskarne . . . 209
Dokumentācija . . . 212
Atjaunināšanas piezīmes . . . 212
Būtiskas izmaiņas . . . 212
Citi . . . 214
Jaunumi un jaunināšanas piezīmes: 8.0.x . . . 216
Zabbix ierīce . . . 216
Procesi . . . 220
serveris . . . 222
Starpniekserveris . . . 248
Aģents . . . 293
Aģents 2 . . . 323
Java gateway . . . 384
Sūtītājs . . . 389
Saņemt . . . 391
JS . . . 392
Tīmekļa pakalpojums . . . 393
Bridge adapter . . . 396
MCP server . . . 399
Vides mainīgie . . . 406
Konfigurācija . . . 408
Hosts un hostu grupas . . . 419
Vienumi . . . 437
Problēmu noteikšana ar trigeriem . . . 657

Vizualizācija . . . 708
Veidnes un veidņu grupas . . . 732
Veidnes pēc noklusējuma . . . 732
Paziņojumi par notikumiem . . . 746
Makrosi . . . 796
Lietotāji un lietotāju grupas . . . 811
Plānotie pārskati . . . 821
Datu eksports . . . 824
Konfigurācijas eksportēšana/importēšana . . . 833
Regulārās izteiksmes . . . 900
Notikumi un atrisināšana . . . 906
Problēmas apstiprināšana un slāpēšana . . . 908
Manuāla problēmu slēgšana . . . 911
Uzturēšanas periodi . . . 913
Tagošana . . . 918
Notikumu korelācija . . . 920
Atklāšana . . . 927
Tīkla atklāšana . . . 927
Aktīva aģenta automātiskā reģistrācija . . . 936
Zema līmeņa atklāšana . . . 940
Virtuālās mašīnas uzraudzība . . . 962
VMware monitoringa vienumu atslēgas . . . 964
Virtuālās mašīnas atklāšanas atslēgu lauki . . . 982
JSON piemēri VMware vienumiem . . . 986
VMware uzraudzības iestatīšanas piemērs . . . 990
Tīmekļa uzraudzība . . . 995
Tīmekļa monitoringa vienumi . . . 1004
Reālās dzīves scenārijs . . . 1005
Pakalpojumu uzraudzība . . . 1012
Pakalpojumu koks . . . 1013
SLA . . . 1017
Iestatīšanas piemērs . . . 1018
Pakalpojumu uzraudzības jaunināšana . . . 1023
Šifrēšana un drošība . . . 1023
Šifrēšana, izmantojot sertifikātus . . . 1032
Šifrēšana, izmantojot iepriekš koplietotas atslēgas . . . 1039
Drošs savienojums ar datubāzi . . . 1042
Drošs savienojums starp lietotāja saskarni un serveri . . . 1053
Piekļuves kontrole . . . 1057
Kriptogrāfija . . . 1061
Tīmekļa servera drošības nodrošināšana . . . 1061
Slepeno datu glabāšana . . . 1065
Google Chrome TLS sertifikāta uzticamība . . . 1073
Drošības labākā prakse . . . 1075
Lietotāja saskarne . . . 1075
Izvēlne . . . 1076
Lietotāja saskarnes sadaļas . . . 1084
Lietotāja iestatījumi . . . 1353
Globālā meklēšana . . . 1358
Lietotāja saskarne maintenance mode . . . 1360
Lapas parametri . . . 1361
Definīcijas . . . 1364
Papildu lietotāja saskarnes valodas . . . 1364
Izveidojot savu tēmu . . . 1365
Atkļūdošanas režīms . . . 1366
Zabbix izmantotās sīkdatnes . . . 1367
Laika joslas . . . 1368
Paroles atiestatīšana . . . 1369
Laika periods un hostu atlasītāji . . . 1369
Konfigurācijas parametri: Zabbix lietotāja saskarne . . . 1371
API . . . 1377
Metožu atsauce . . . 1382
1. pielikums. Atsauces komentāri . . . 2236

2. pielikums. Izmaiņas no 7.4 uz 8.0 . . . 2243
3. pielikums. Izmaiņas 8.0 versijā . . . 2245
Paplašinājumi . . . 2245
Ielādējamie moduļi . . . 2247
Lietotāja saskarnes moduļi . . . 2256
Pielikumi . . . 2257
Instalēšana un iestatīšana . . . 2257
Vienumi . . . 2303
Atbalstītās funkcijas . . . 2343
Makrokomandas . . . 2343
Versijas saderība . . . 2379
Laika perioda sintakse . . . 2380
Vienību simboli . . . 2381
Komandas izpilde . . . 2382
Atkāpšanās piemēri . . . 2383
Kas ir Zabbix . . . 2385

Zabbix Cloud **2389**

Zabbix Cloud **2389**
Iestatiet un pārvaldiet Zabbix Cloud . . . 2389
Uzziniet par Zabbix Cloud . . . 2389
Zabbix izvietošana mākonī . . . 2389
Mezglu konfigurācija . . . 2399
Pārskats . . . 2400
Cilne “Pārskats” . . . 2401
Piekļuves filtru cilne . . . 2404
Šifrēšanas cilne . . . 2405
Dublējumu cilne . . . 2405
Vēstures cilne . . . 2405
Uzturēšanas cilne . . . 2406
Jaunināšanas cilne . . . 2407
Lietotāju pievienošana . . . 2409
Ievads . . . 2409
Lietotāju pievienošana organizācijām . . . 2409
Lietotāju pievienošana mezgliem . . . 2410
Zabbix Cloud galvenās atšķirības . . . 2411
Ievads . . . 2411
Audita žurnāls . . . 2413
Zabbix Cloud starpniekservera iestatīšana . . . 2415

Mobilā lietotne **2424**

Mobilā lietotne **2424**
Zabbix iestatīšana lietotnei . . . 2424
Uzziniet par lietotni . . . 2424
Zabbix iestatīšana lietotnei . . . 2424
Lietotnes pārskats . . . 2425
Droša piekļuve no mobilajām ierīcēm . . . 2433

Izstrādātāju centrs **2434**

Izstrādātāju centrs **2434**
Veidojiet uz Zabbix bāzes . . . 2434
Spraudņa izveide . . . 2434
Spraudņu saskarnes . . . 2440
Python bibliotēka Zabbix համախպոգai . . . 2442
Instalēšana . . . 2442
Ātrās darba sākšanas ceļvedis . . . 2442
Zabbix API izmantošana . . . 2446
Datu vākšana no Zabbix aģenta . . . 2449
Sūtīt datus uz Zabbix serveri vai starpniekserveri . . . 2450
Atkļūdošanas žurnalēšana . . . 2454
Moduļi . . . 2454
Moduļa failu struktūra . . . 2455

Logrīki . . . 2462
Pamācības . . . 2471
Piemēri . . . 2497
Izmaiņas paplašinājumu izstrādē . . . 2497

Zabbix rokasgrāmatas lapas **2498**
zabbix_agent2 . . . 2498
zabbix_agentd . . . 2499
zabbix_get . . . 2501
zabbix_js . . . 2503
zabbix_proxy . . . 2504
zabbix_sender . . . 2506
zabbix_server . . . 2511
zabbix_web_service . . . 2513

Autortiesību paziņojums **2514**

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 Īsi, uz uzdevumiem orientēti pamācību materiāli, kas soli pa solim izskaidro pamatus — no pirmajiem konfigurācijas soļiem līdz pirmajam problēmas brīdinājumam.

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.

Mobilā lietotne

Iegūt lietotni Instalējiet Zabbix mobilo lietotni (drīzumā pieejama) savā ierīcē un sāciet pārraudzīt savu Zabbix vidi, atrodoties ceļā.

Iestatiet Zabbix lietotnei Instalējiet Zabbix bridge adapteri, lai mobilās ierīces varētu izveidot savienojumu ar Zabbix. Pēc tam pievienojiet savu ierīci un ieslēdziet pašpiegādes paziņojumus.

Lietotnes pārskats Uzziniet Zabbix mobilās lietotnes lietošanas pamatus.

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.

Instalēšana un pirmie soļi

Tālāk norādītā secība izklāsta Zabbix iestatīšanas procesu:

1. Pārbaudiet, vai nepieciešamā programmatūra ir instalēta un darbojas.
2. Instalējiet Zabbix.
3. Instalējiet Zabbix lietotāja saskarni.
4. Sāciet monitorēšanu.

Sīkāka informācija par katru soli ir sniegta tālāk.

Nepieciešamā programmatūra Zabbix labi darbojas ar tālāk norādīto programmatūru (to bieži dēvē 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, iespējams, jau ir instalēts. Šādā gadījumā, iespējams, būs jāiestata tikai datu bāzes serveris un tīmekļa serveris.

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

Instalējiet Zabbix **Vienkāršākais** veids, kā instalēt Zabbix, ir izmantot **distribūcijas pakotnes** (alternatīvi — konteinerus). Pakotnes instalē nepieciešamās programmatūras atkarības.

Precīzi, jūsu operētājsistēmai pielāgoti instalēšanas norādījumi ir pieejami **Zabbix tīmekļa vietnē**.

Atlasiet savu platformu un izpildiet sniegtos norādījumus. Sākumā ieteicams instalēt Zabbix serveri, Zabbix aģentu 2 un lietotāja saskarni (tīmekļa saskarni). Vēlāk varat pievienot Zabbix starpniekserverus un citus aģentus.

Instalēšanas procesa daļa ir datu bāzes izveide (konfigurācijas un vēstures datu glabāšanai) un sākotnējo datu importēšana tajā.

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

Jums būs jānorāda datu bāzes parole (izveidota iepriekšējā darbībā), lai lietotāja saskarne varētu sazināties ar datu bāzi.

Sāciet uzraudzību Piesakieties Zabbix tīmekļa saskarnē.

Izmantojiet **ātrās sākšanas rokasgrāmatas**, lai sāktu populāru resursu (Linux, Apache, MySQL vai citu) uzraudzību. Šajās rokasgrāmatās ir izskaidrots, ar kurām veidnēm jāstrādā, lai sāktu populāru resursu uzraudzību.

Varat sākt ar ātrās sākšanas rokasgrāmatu Linux servera uzraudzībai.

Lietotāja saskarnes instalēšana

Šajā sadaļā sniegti norādījumi par Zabbix lietotāja saskarnes instalēšanu.
Veiciet šīs darbības pēc Zabbix aizmugursistēmas instalēšanas.

Zabbix lietotāja saskarne ir rakstīta PHP valodā, un tai nepieciešams atbalstīts tīmekļa serveris, piemēram, Apache vai NGINX, ar konfigurētu PHP.
Informāciju par nepieciešamo programmatūru un bibliotēkām skatiet lietotāja saskarnes prasībās.

Lai Zabbix tīmekļa lietotāja saskarnē izmantotu citu valodu, nevis angļu valodu, tīmekļa serverī ir jāinstalē attiecīgā lokalizācija, kā arī PHP gettext paplašinājums.
Ņemiet vērā, ka valodas maiņa var mainīt laika/datuma formātu.

Lai iestatītu SSL Zabbix lietotāja saskarnei, skatiet sadaļu SSL iestatīšana Zabbix lietotāja saskarnei.

Instalēšanas vednis

Lai instalētu lietotāja saskarni, pārlūkprogrammā atveriet Zabbix lietotāja saskarnes URL:

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

Sveiciena ekrāns

Lietotāja saskarnes instalēšanas vednis tiek atvērts sveiciena ekrānā.

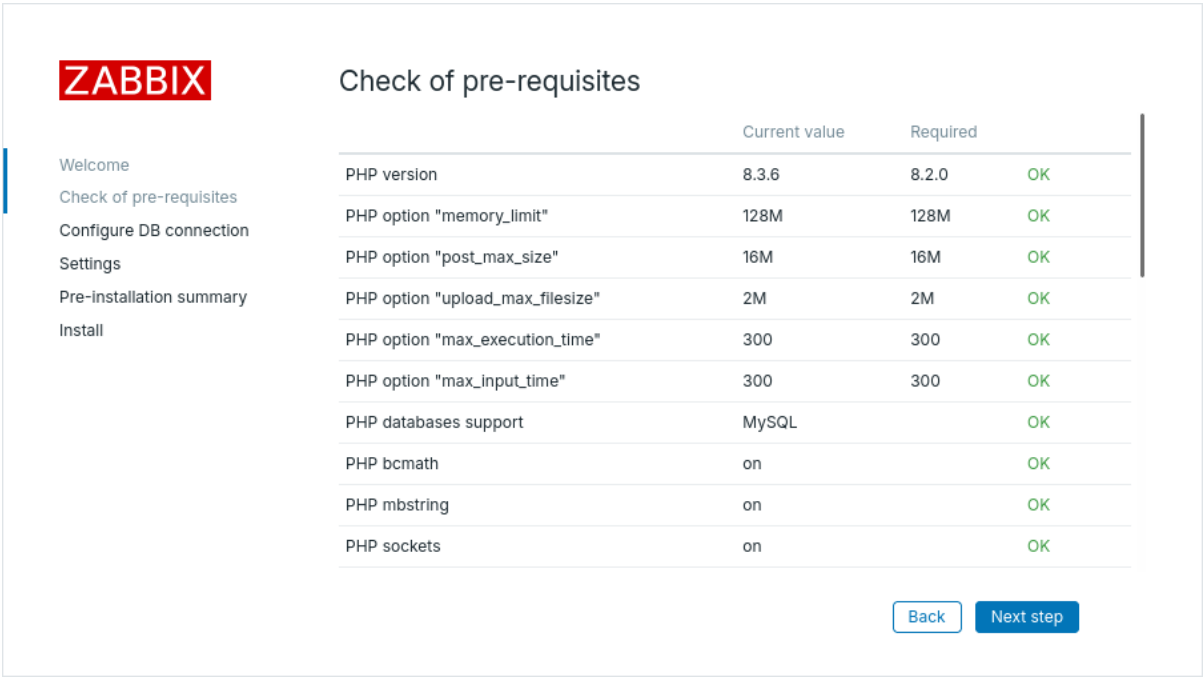


Sistēmas noklusējuma valoda ir angļu (en_US), un tiek izmantots ASV laika/datuma formāts. Lai mainītu sistēmas noklusējuma valodu, nolaižamajā izvēlnē *Noklusējuma valoda* atlasiet vajadzīgo valodu.

Lai turpinātu, noklikšķiniet uz pogas *Nākamais solis*.

Priekšnosacījumu pārbaude

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



Skatiet sadaļu Prasības, lai uzzinātu par nepieciešamajiem un neobligātajiem PHP paplašinājumiem, nepieciešamo PHP versiju un atbalstītajām datubāzēm. Ņemiet vērā, ka neizpildīti neobligātie priekšnosacījumi tiek parādīti ar statusu *Brīdinājums*, un iestatīšanas procesu var turpināt arī tad, ja tie nav izpildīti.

PHP konfigurācijas direktīvu nepieciešamās vērtības servera izmantotajā `php.ini` failā:

```
memory_limit = 128M ; Minimālā vērtība
post_max_size = 16M ; Minimālā vērtība
upload_max_filesize = 2M ; Minimālā vērtība
```

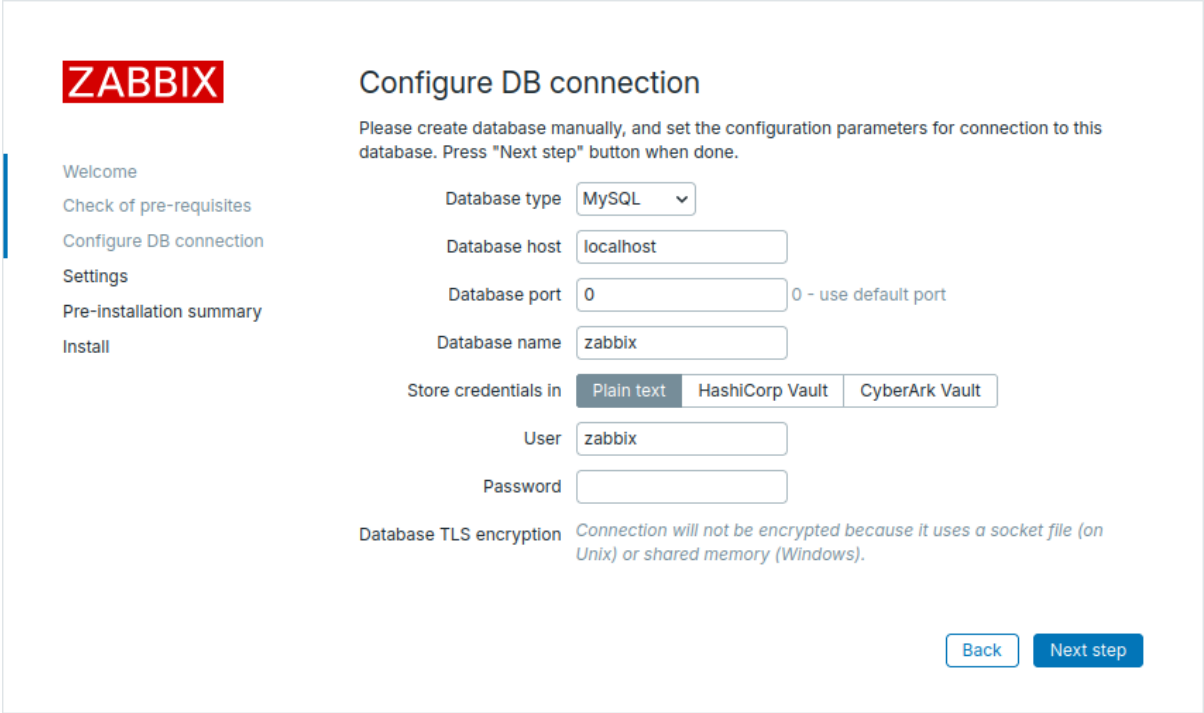
```
max_execution_time = 300 ; Minimālā vērtība; ir atļautas arī vērtības 0 un -1
max_input_time = 300 ; Minimālā vērtība; ir atļautas arī vērtības 0 un -1
session.auto_start = 0 ; Jābūt atspējotam
arg_separator.output = "&" ; Ir atļauts arī "&amp;"
```

Attention:
Ja jāmaina servera lietotājs vai grupa, pārbaudiet, vai jaunajam lietotājam vai grupai ir nepieciešamās atļaujas PHP sesiju direktorijai; pretējā gadījumā Zabbix iestatīšanu var nebūt iespējams turpināt.

Ja visi obligātie priekšnosacījumi ir izpildīti, pārejiet pie nākamās darbības.

Konfigurēt datubāzes savienojumu

Ievadiet informāciju savienojuma izveidei ar datubāzi.



Ja lauks *Datubāzes hosts* ir atstāts tukšs, tiek izmantota noklusējuma Unix domēna ligzda.

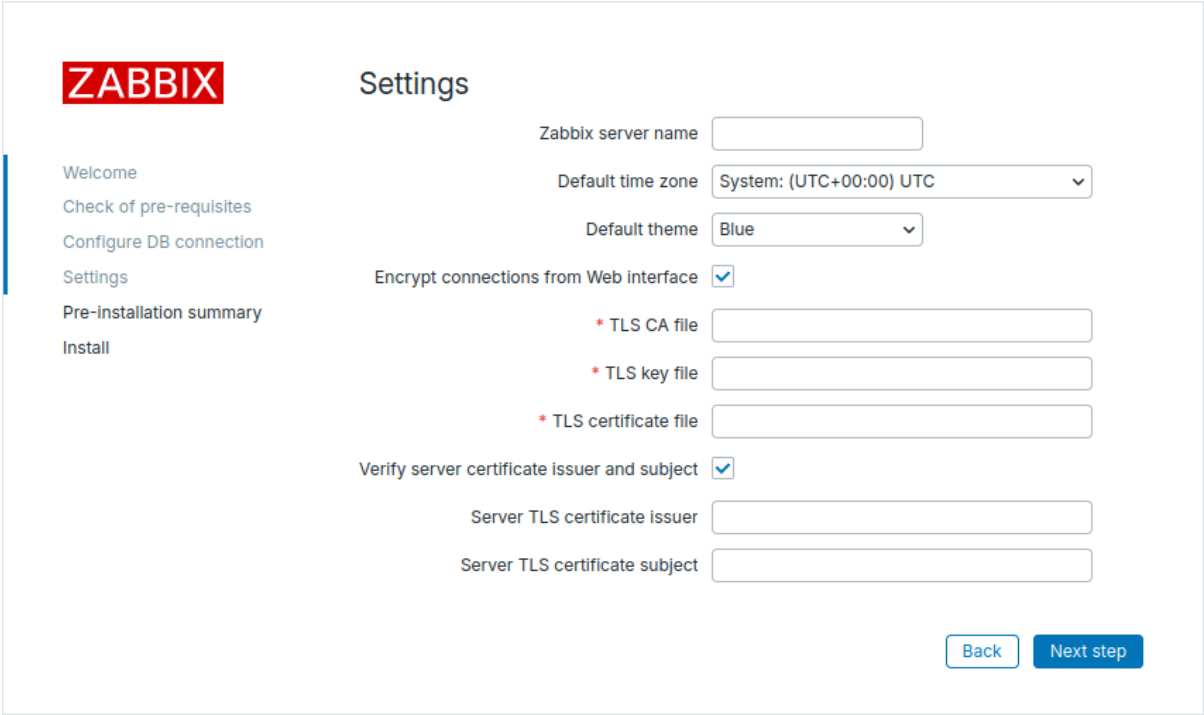
MySQL gadījumā, ievadot `localhost` laukā *Datubāzes hosts*, savienojums tiek izveidots, izmantojot noklusējuma Unix domēna ligzdu. Vednis nenodrošina iespēju konfigurēt datubāzes ligzdu. Lai izmantotu pielāgotu ligzdu, konfigurējiet to Zabbix servera iestatījumos (piemēram, izmantojot parametru DBSocket= Zabbix servera konfigurācijas failā (`zabbix_server.conf`)). Tādējādi lietotāja saskarnes iestatījumi būs saskaņoti ar servera savienojuma ar datubāzi iestatījumiem.

PostgreSQL gadījumā laukā *Datubāzes hosts* var ietvert vairākus `host:port` ierakstus, kas atdalīti ar komatiem. Virkne tiek sadalīta daļās, un katrs ports tiek validēts tāpat kā laukā *Datubāzes ports*. Pēc veiksmīgas validācijas virkne tiek parsēta hosts un portos, lai izveidotu savienojuma virkni. Ja hosts komatatdalītajā sarakstā ports nav norādīts, kā noklusējuma ports tiek izmantota *Datubāzes porta* vērtība. Ja ir norādīts ligzdas ceļš (piemēram, `/var/run/pgbouncer`), tā vietā tiek izmantota attiecīgā Unix domēna ligzda.

Lai konfigurētu drošu savienojumu ar datubāzi, atlasiet izvēles rūtiņu *Datubāzes TLS šifrēšana* un pārejiet uz sadaļu Lietotāja saskarne uz datubāzi, lai konfigurētu laukus. Izvēles rūtiņa *Datubāzes TLS šifrēšana* ir atspējota šādos gadījumos: - MySQL gadījumā *Datubāzes hosts* ir iestatīts uz *localhost*. - PostgreSQL gadījumā lauka *Datubāzes hosts* vērtība sākas ar slīpsvītru vai lauks ir tukšs.

Lai konfigurētu Zabbix noslēpumu izgūšanu no HashiCorp Vault, laukā *Akreditācijas datu glabāšana* atlasiet *HashiCorp Vault* un pārejiet uz sadaļu HashiCorp konfigurācija, lai aizpildītu izvērstos laukus. Lai konfigurētu Zabbix noslēpumu izgūšanu no CyberArk Vault, laukā *Akreditācijas datu glabāšana* atlasiet *CyberArk Vault* un pārejiet uz sadaļu CyberArk konfigurācija, lai aizpildītu izvērstos laukus.

Iestatījumi



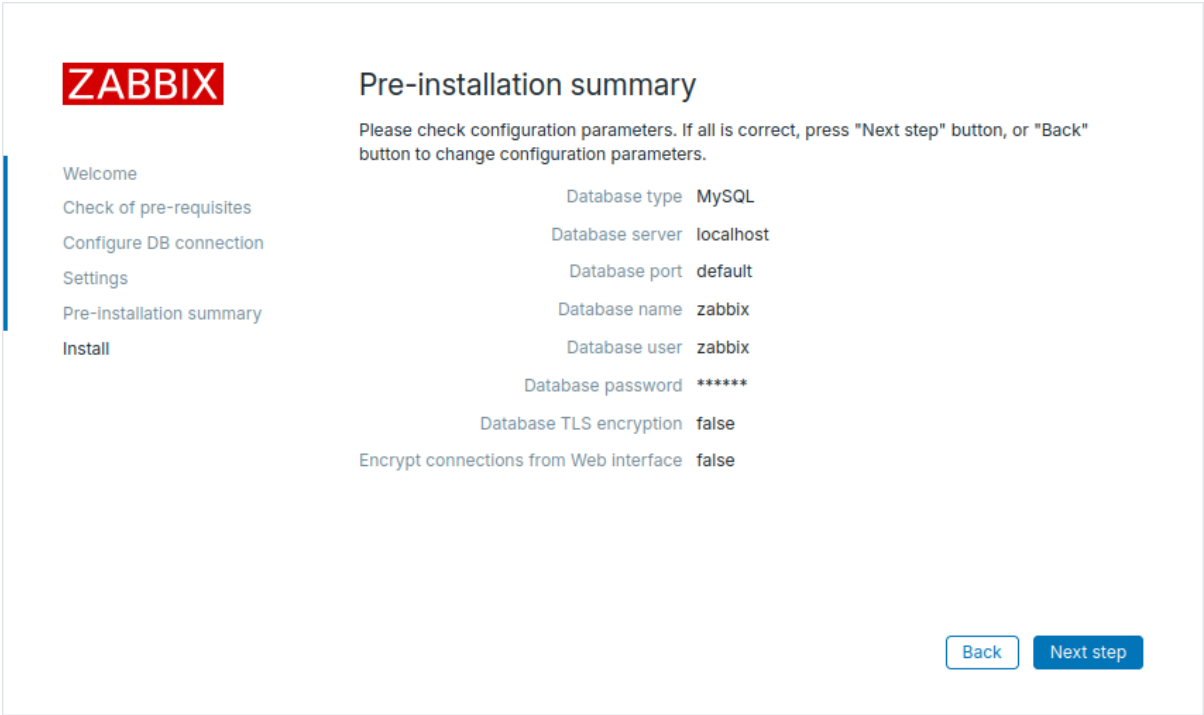
Zabbix servera nosaukuma ievadīšana nav obligāta; tomēr, ja tas tiek iesniegts, tas tiks parādīts izvēļņu joslā un lapu virsrakstos.

Iestatiet noklusējuma laika joslu un lietotāja saskarnes motīvu.

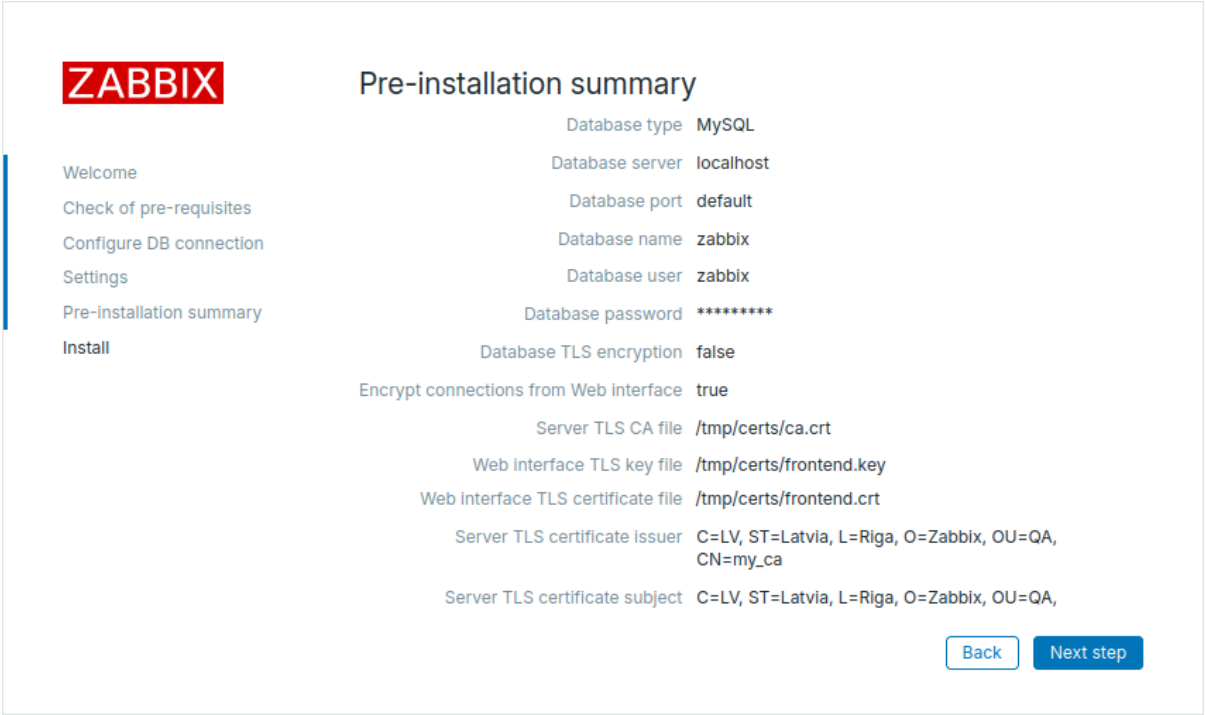
Lai konfigurētu TLS savienojumus starp Zabbix lietotāja saskarni un Zabbix serveri, atzīmējiet izvēles rūtiņu *Šifrēt savienojumus no tīmekļa saskarnes* un turpiniet ar TLS savienojuma konfigurēšanu, lai aizpildītu izvērstos laukus.

Pirmsinstalēšanas kopsavilkums

Pārskatiet iestatījumu kopsavilkumu.



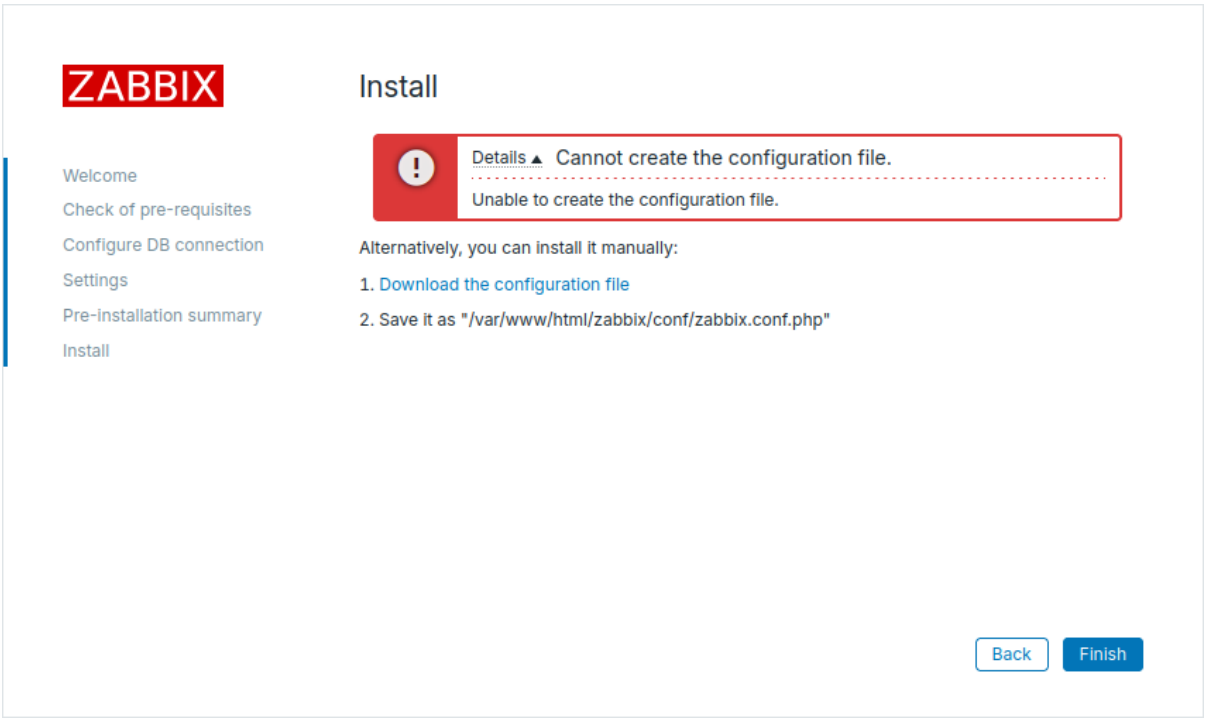
Ja ir konfigurēts TLS, vednis parādīs tā informāciju.



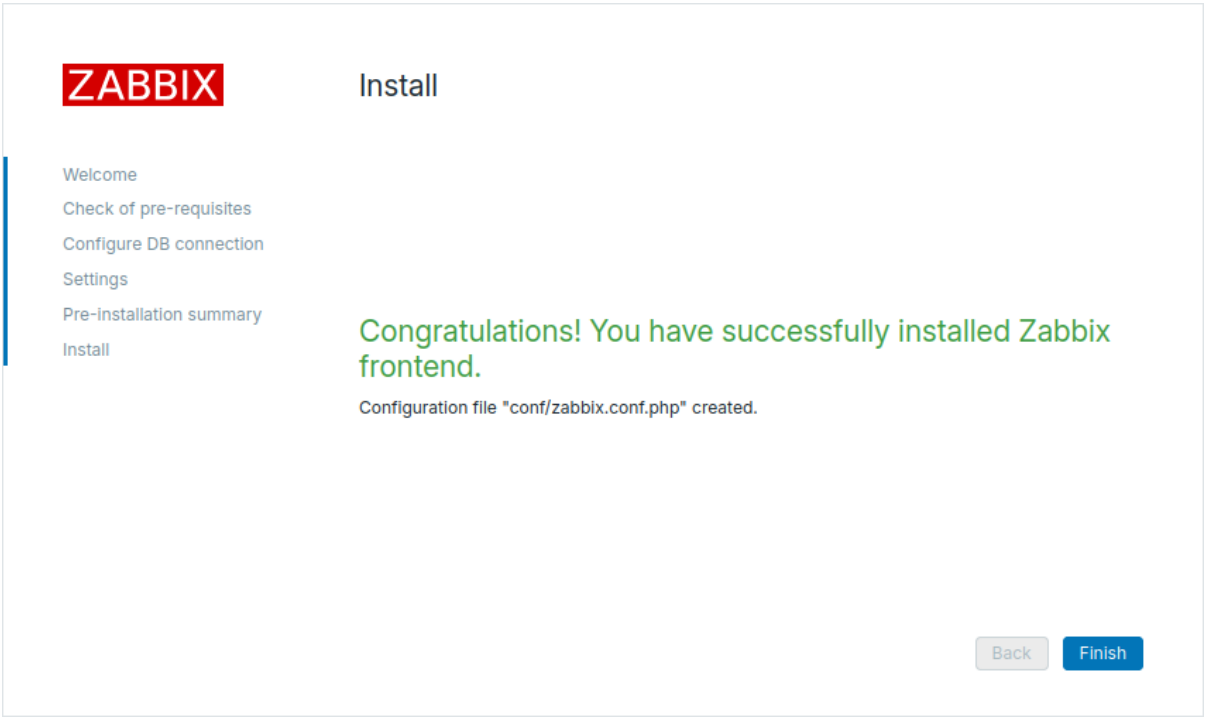
Instalēšana

Ja lietotājam, ar kuru darbojas tīmekļa serveris, ir rakstīšanas piekļuve `conf/` direktorijai, konfigurācijas fails tiks saglabāts automātiski, un jūs varat nekavējoties pāriet uz nākamo darbību.

Ja lietotājam, ar kuru darbojas tīmekļa serveris, nav rakstīšanas piekļuves `conf/` direktorijai, jums ir jālejupielādē konfigurācijas fails un jāievieto tas `conf/` direktorijā tīmekļa servera HTML dokumentu direktorijā, kurā nokopējāt Zabbix PHP failus.

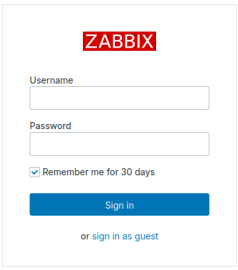


Pabeidziet instalēšanu.



Pieteikšanās

Piesakieties, izmantojot noklusējuma akreditācijas datus: - Lietotājvārds: *Admin* - Parole: *zabbix*



Lietotāja saskarnes konfigurācijas fails

Pēc instalēšanas varat pārkonfigurēt Zabbix lietotāja saskarni, atjauninot parametrus lietotāja saskarnes konfigurācijas failā `zabbix.conf.php`.

Konfigurācijas faila noklusējuma ceļš ir atkarīgs no operētājsistēmas un Zabbix instalēšanas metodes:

- Instalēšanai no pakotnēm: `/usr/share/zabbix/ui/conf/zabbix.conf.php`
- Instalēšanai no pirmkoda: `/var/www/html/zabbix/conf/zabbix.conf.php`
- Ierīcei: `/etc/zabbix/web/zabbix.conf.php`

Detalizētu informāciju par Zabbix lietotāja saskarnes konfigurēšanu skatiet konfigurācijas faila opcijās.

Ātrās darba sākšanas ceļveži

Kad esat instalējis Zabbix un tā lietotāja saskarni, ir pienācis laiks sākt uzraudzību.

Izmantojiet šos ātros ceļvežus, lai sāktu populāru resursu uzraudzību:

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

Ātrās darba sākšanas ceļveži ir vērsti uz oficiālo Zabbix veidņu izmantošanu uzraudzībai.
Veidne ir iepriekš konfigurēta uzraudzības elementu kopa.
Veidne ļauj mērogot uzraudzību — varat pārbaudīt veidni vienā uzraudzības mērķī (ierīcē, lietotnē u. c.), pielāgot to un pēc tam lietot vairākiem līdzīgiem uzraudzības mērķiem.

Lai gan veidnes ir piemērotas uzraudzībai uzreiz pēc to izmantošanas, iespējams, vēlēsities pielāgot to elementus.

Lai labāk izprastu veidni veidojošos pamatelementus, skatiet sadaļu Pamatkonfigurācija.

Pamata konfigurācija

Šīs rokasgrāmatas iepazīstina ar Zabbix konfigurācijas pamatelementiem:

- Pieteikšanās un lietotāja konfigurēšana
- Jauns hosts
- Jauns vienums
- Jauns trigeris
- Paziņojuma par problēmu saņemšana
- Jauna veidne

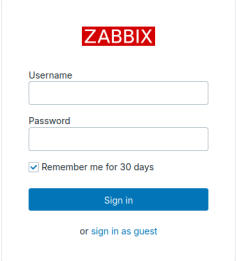
Pieteikšanās un lietotāja konfigurēšana

Pārskats

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

Pieteikšanās

Atveriet Zabbix lietotāja saskarnes URL pārlūkprogrammā: - Apache: http://<server_ip_or_name>/zabbix - NGINX: http://<server_ip_or_name>



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 tūlīt pēc pirmās pieteikšanās nomainīt Admin konta noklusējuma paroli.

Pastāvīga pieteikšanās

Lai saglabātu pieteikšanos līdz 30 dienām, pirms noklikšķināšanas uz *Pierakstīties* atlasiet *Atcerēties 30 dienas*.

Ja ir iespējota opcija Atcerēties 30 dienas:

- Jūsu sesija paliek aktīva 30 dienas.
- *Automātiskā atteikšanās* tiek ignorēta, un jūs paliekat pierakstījies līdz perioda beigām.
- Turpmākajos apmeklējumos 30 dienu laikā jūs tiksiet automātiski pierakstīts bez atkārtotas akreditācijas datu ievadīšanas.

Ja opcija Atcerēties 30 dienas ir atspējota:

- Jebkura iepriekš iespējotā automātiskā pieteikšanās tiek notīrīta.
- Sesija beigsies atbilstoši konfigurētajam *Automātiskās atteikšanās* intervālam.

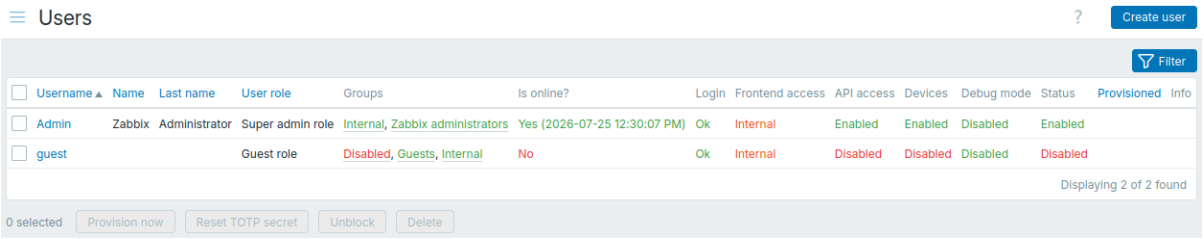
Aizsardzība pret brutāla spēka uzbrukumiem

Ja piecas reizes pēc kārtas neizdodas pieteikties, Zabbix lietotāja saskarne uz 30 sekundēm apturēs darbību, lai novērstu brutāla spēka un vārdnīcas uzbrukumus.

Pēc veiksmīgas pieteikšanās tiks parādīta neveiksmīga pieteikšanās mēģinājuma IP adrese.

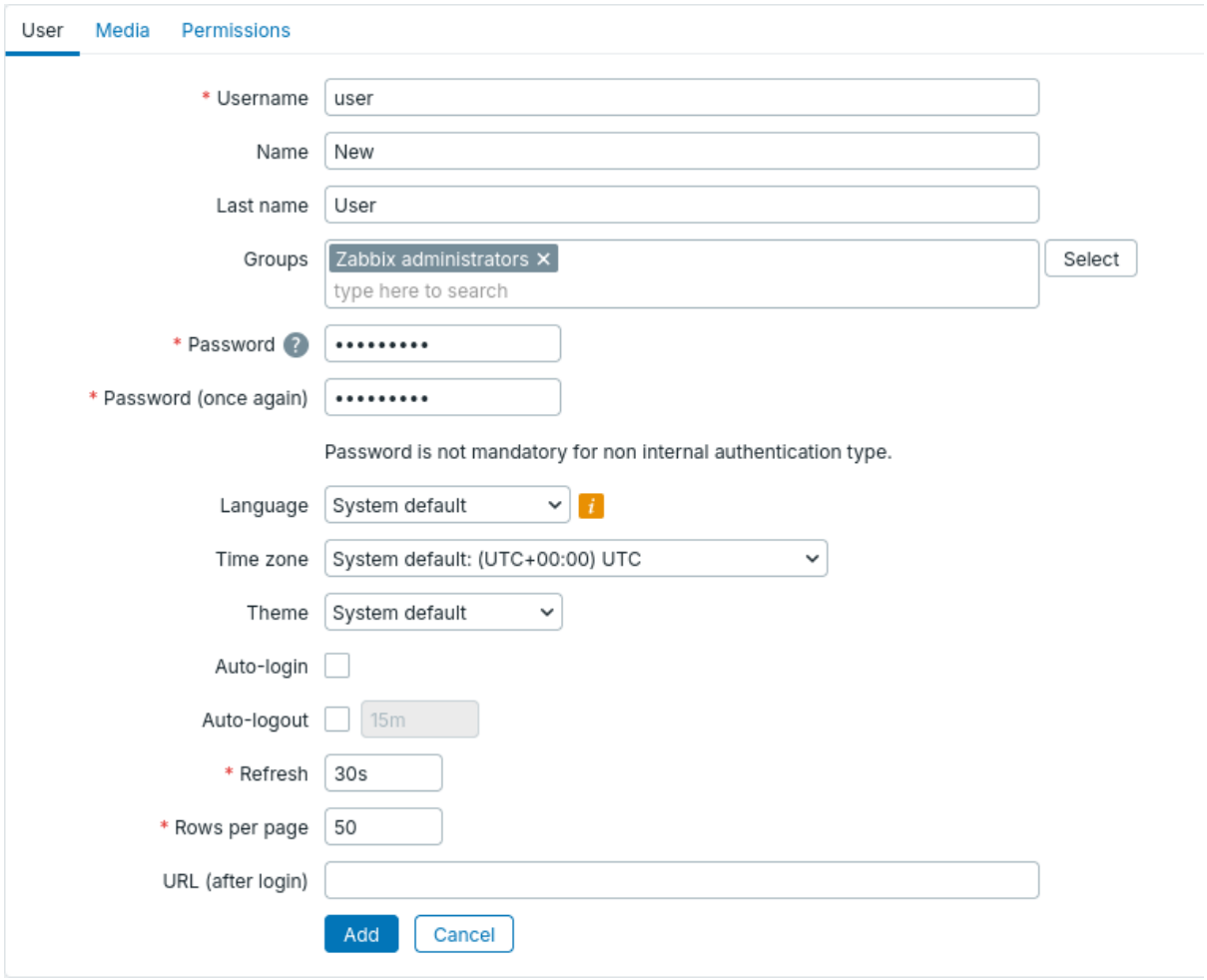
Lietotāja pievienošana

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



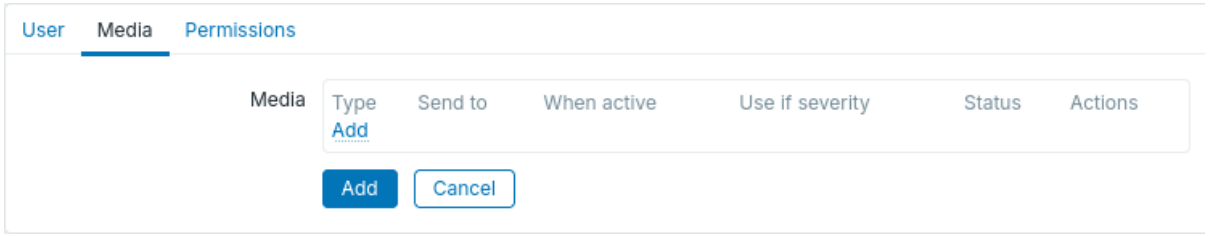
Lai pievienotu jaunu lietotāju, augšējā labajā stūrī atlasiet *Create user*.

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



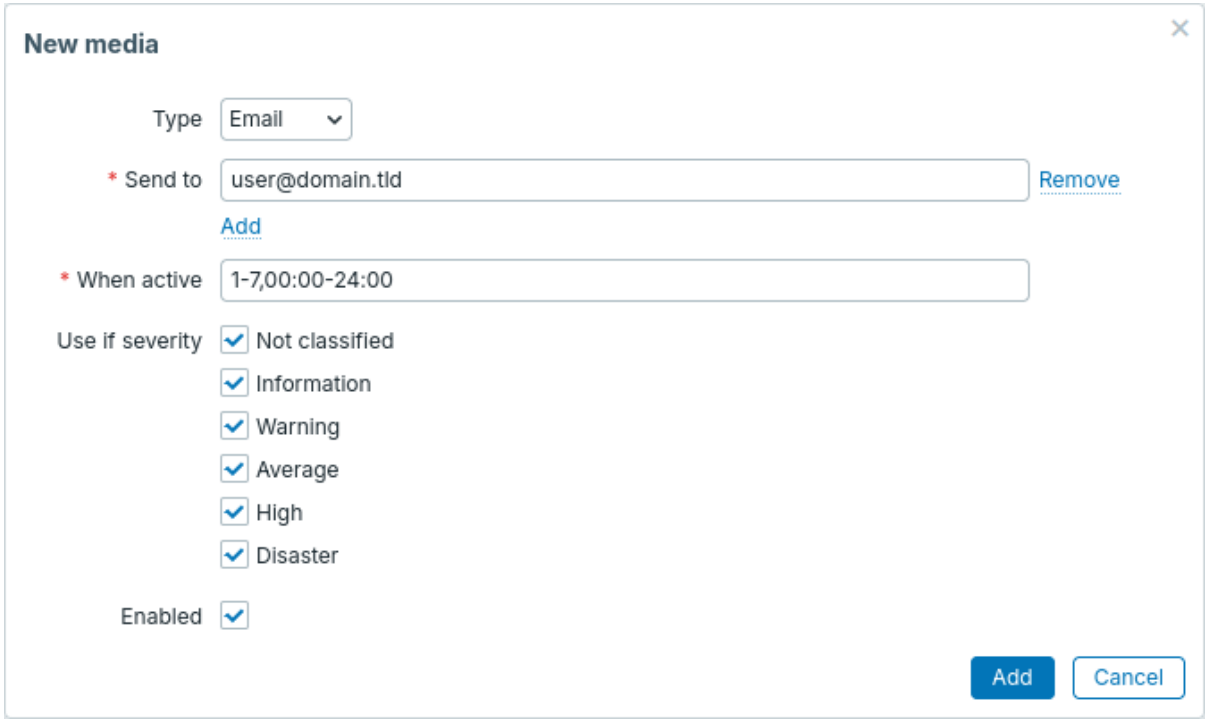
Visi obligātie ievades lauki ir atzīmēti ar sarkanu zvaigznīti. Sīkāku informāciju par šīs konfigurācijas formas ievades laukiem skatiet lapā User settings.

Pēc noklusējuma jauniem lietotājiem nav definēti nekādi mediji (paziņojumu piegādes metodes). Lai izveidotu vienu, dodieties uz cilni *Media* un noklikšķiniet *Add*.



Uznirstošajā logā ievadiet lietotāja e-pasta adresi.
 Visi mediju tipi pēc noklusējuma ir atspējoti, un tos var iespējot sadaļā *Alerts* > *Media types*.

Varat norādīt laika periodu, kurā medijs būs aktīvs (formāta aprakstu skatiet lapā Time period specification). Pēc noklusējuma medijs vienmēr ir aktīvs. Varat arī pielāgot trigeru smaguma pakāpes līmeņus, kuriem medijs būs aktīvs, taču pagaidām atstājiet tos visus iespējotus.



Noklikšķiniet *Add*, lai saglabātu mediju, un 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. Noklikšķiniet *Select* un sarakstā atlasiet vienu no lomām. Piemēram, atlasiet *Admin role*, lai piešķirtu piekļuvi visām Zabbix lietotāja saskarnes sadaļām, izņemot *Administration*. Vēlāk varat mainīt atļaujas vai izveidot vairāk lietotāju lomu. Pēc lomas atlasīšanas atļaujas parādīsies tajā pašā cilnē:

Users

User | Media | Permissions

* Role: Admin role ✕ | Select

User type: Admin

Permissions

User group permissions

Group	Type	Permissions
All groups	Hosts	None
All groups	Templates	None

Proxies: None

Proxy groups: None

Permissions can be assigned for user groups only.

Access to UI elements

Dashboards: Dashboards

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

APM: Traces | Metrics | Logs

Services: Services | SLA | SLA report

Inventory: Overview | Hosts

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

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

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

User profile settings

Redirect URL after login

Access to services

Read-write access to services: None

Read-only access to services: All

Access to modules

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

Access to API

Disabled

Devices

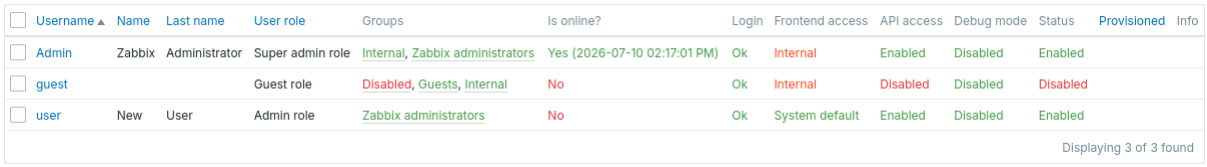
Enabled | Manage own devices

Access to actions

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

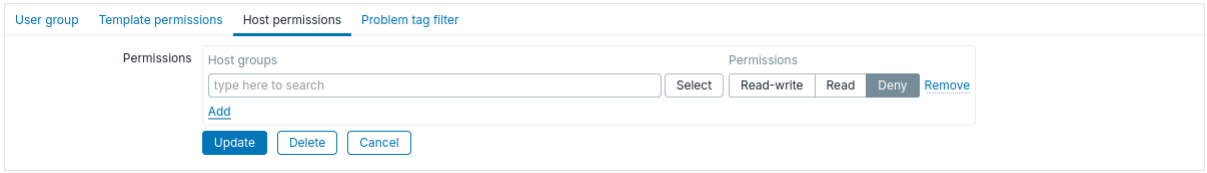
Add | Cancel

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

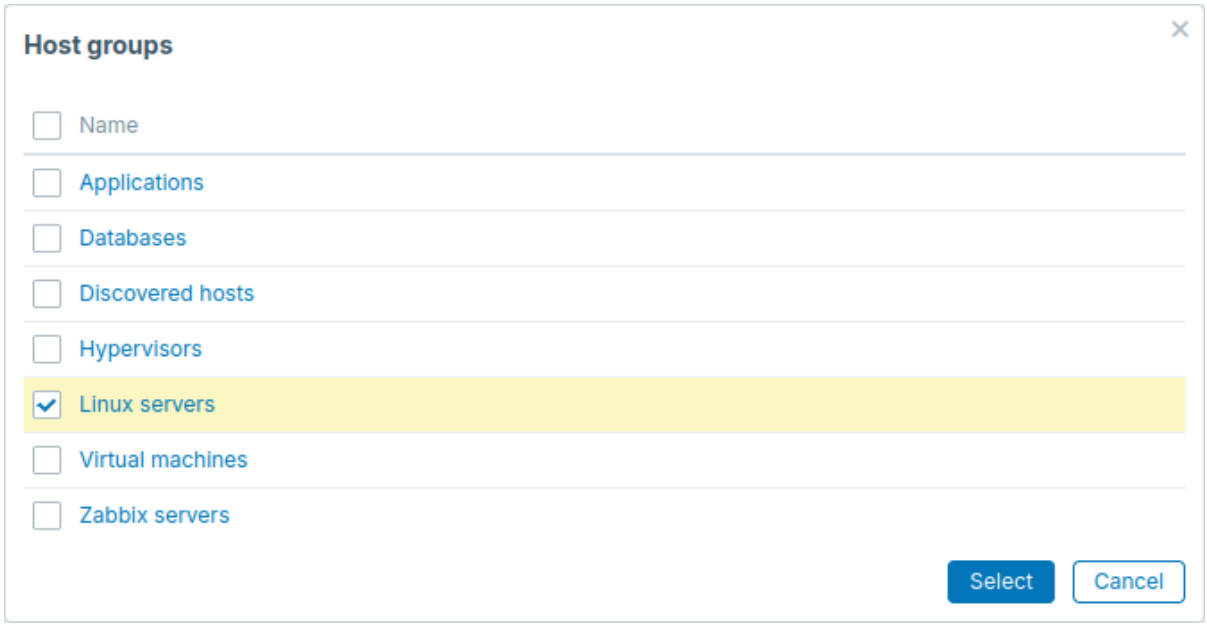


Atļauju piešķiršana

Pēc noklusējuma jaunam lietotājam nav atļauju piekļūt hosts un veidnēm. Lai piešķirtu lietotājam tiesības, noklikšķiniet uz lietotāja grupas slejā *Groups* (šajā gadījumā - *Zabbix administrators*). Lietotāju grupu rekvizītu formā atveriet cilni *Host permissions*, lai piešķirtu atļaujas hostu grupām. Noklikšķiniet uz Add, lai tiktu parādīts hostu grupas atlases lauks:



Pēc tam noklikšķiniet uz *Select* blakus laukam, lai skatītu hostu grupu sarakstu. Šim lietotājam ir jābūt tikai lasīšanas piekļuvei grupai *Linux servers*, tāpēc atzīmējiet atbilstošo izvēles rūtiņu sarakstā un noklikšķiniet uz *Select*, lai apstiprinātu 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 jāpārslēdzas uz cilni *Template permissions* un jānorāda veidņu grupas. Darbības ir identiskas atļauju piešķiršanai hostu grupām. Pārskats par veidnēm ir pieejams šīs ātrās darba sākšanas sadaļā New template.

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

Gatavs! Varat mēģināt pieteikties, izmantojot jaunā lietotāja akreditācijas datus.

Jauns hosts

Pārskats

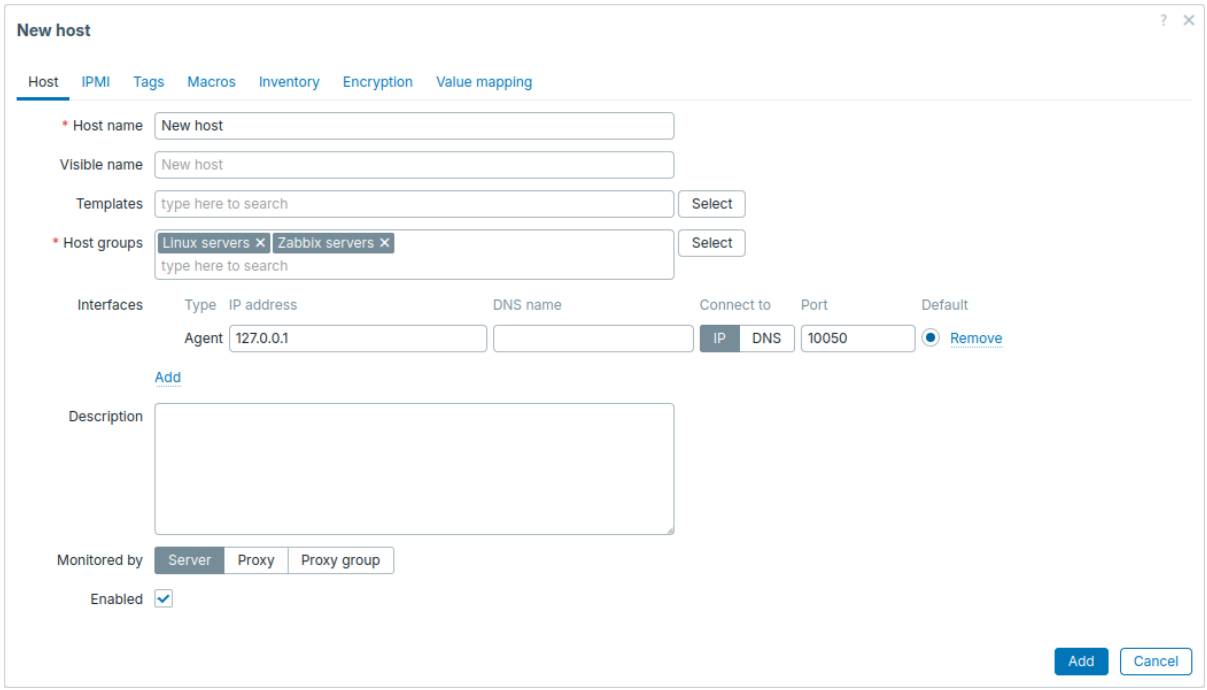
Šajā sadaļā jūs uzzināsiet, kā iestatīt jaunu hostu.

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

Hosts pievienošana

Informācija par Zabbix konfigurētajiem hostiem ir pieejama izvēlnes sadaļā *Datu apkopošana* > *Hosti*, kā arī izvēlnes sadaļā *Pārraudzība* > *Hosti*. Jau ir iepriekš definēts hosts ar nosaukumu "Zabbix serveris", taču mēs vēlamies apgūt vēl viena hosta pievienošanu.

Lai pievienotu jaunu hostu, noklikšķiniet uz *Izveidot hostu*. Tiek atvērta hosta konfigurācijas veidlapa.



Visi obligātie ievades lauki ir atzīmēti ar sarkanu zvaigznīti.

Šeit obligāti jānorāda vismaz:

Hosta nosaukums

- Ievadiet hosta nosaukumu. Ir atļauti burti un cipari, atstarpes, punkti, defises un pasvītras.

Hostu grupas

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

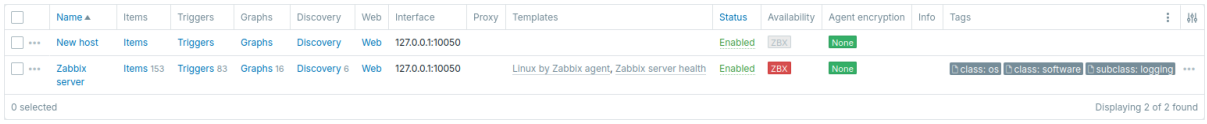
Note:
Visas piekļuves atļaujas tiek piešķirtas hostu grupām, nevis atsevišķiem hostiem. Tāpēc hostam ir jāpieder vismaz vienai grupai.

Saskarnes: IP adrese

- Lai gan tehniski tas nav obligāts lauks, hosta saskarne ir nepieciešama noteiktu metriku apkopošanai. Lai izmantotu pasīvās Zabbix aģenta pārbaudes, šajā laukā norādiet aģenta IP adresi vai DNS nosaukumu. Ņemiet vērā, ka Zabbix servera IP adrese vai DNS nosaukums jānorāda arī Zabbix aģenta konfigurācijas faila direktīvā 'Server'. Ja Zabbix aģents un Zabbix serveris ir instalēti vienā datorā, abās vietās jānorāda viena un tā pati IP adrese vai DNS nosaukums.

Pārējās opcijas var atstāt nemainītas, izmantojot to noklusējuma vērtības.

Kad esat pabeidzis, noklikšķiniet uz *Pievienot*. Jaunajam hostam vajadzētu būt redzamam hostu sarakstā.



Kolonnā *Pieejamība* ir ietverti hosta pieejamības indikatori katrai saskarnei. Kad ir definēta Zabbix aģenta saskarne, hosta pieejamību var noteikt, izmantojot aģenta pieejamības ikonu (apzīmēta ar *ZBX*):

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

Jauns vienums

Pārskats

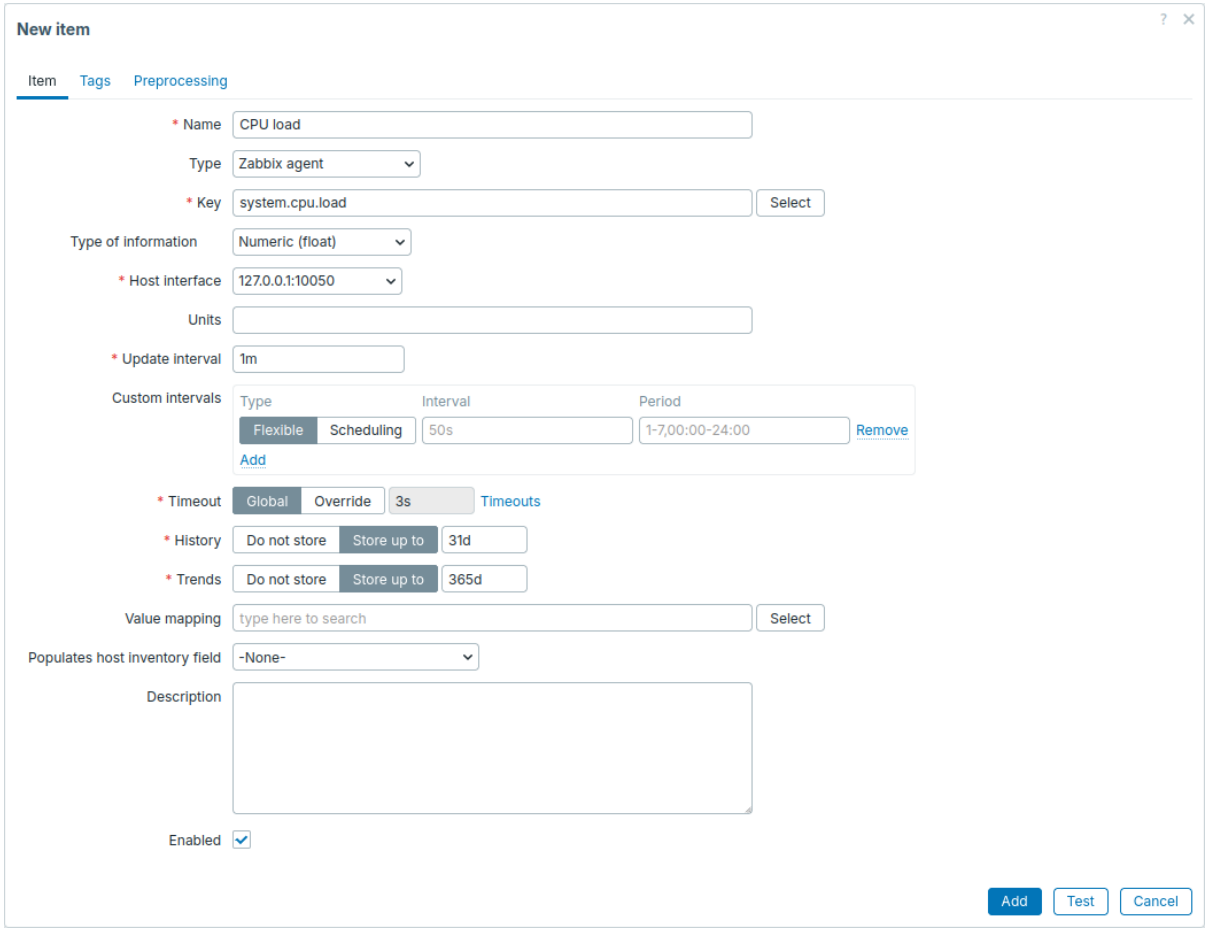
Šajā sadaļā jūs uzzināsiet, kā iestatīt vienumu.

Vienumi ir Zabbix datu apkopošanas pamats. 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, mēs ejam uz *Datu vākšana* > *Hosts* un atrodam izveidoto "New host".

Noklikšķiniet uz saites *Vienumi* rindā "New host", un pēc tam noklikšķiniet uz *Izveidot vienumu*. Tiek atvērta vienuma konfigurācijas forma.



Visi obligātie ievades lauki ir atzīmēti ar sarkanu zvaigznīti.

Mūsu parauga vienumam ievadāmā būtiskā informācija ir:

Nosaukums

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

Atslēga

- Manuāli ievadiet vērtību *system.cpu.load*. Tas ir vienuma tehniskais nosaukums, kas identificē informācijas veidu, kas tiks apkopots. Konkrētā atslēga ir tikai viena no iepriekš definētajām atslēgām, kas ir iekļautas Zabbix aģents.

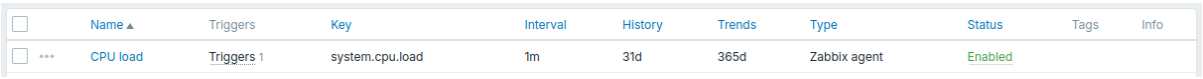
Informācijas tips

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

Note:
Iespējams, vēlēsities arī samazināt dienu skaitu, cik ilgi tiks glabāta vienuma vēsture, līdz 7 vai 14. Tā ir laba prakse, lai atvieglotu datubāzes slodzi, neglabājot lielu vēsturisko vērtību apjomu.

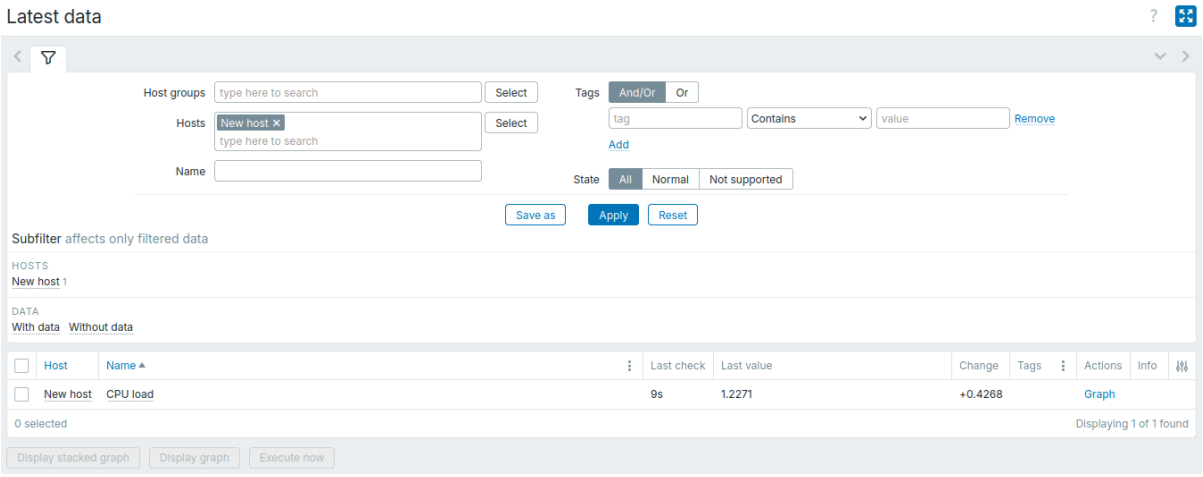
Citas opcijas var atstāt nemainītas, izmantojot to noklusējuma vērtības.

Kad esat pabeidzis, noklikšķiniet uz *Pievienot*. Jaunajam vienumam vajadzētu parādīties vienumu sarakstā.



Datu skatīšana

Kad vienums ir definēts, jums var rasties jautājums, vai tas patiešām apkopo datus. Lai to pārbaudītu, dodieties uz *Monitoring* > *Latest data*, filtra laukā atlasiet 'New host' un noklikšķiniet uz *Apply*.



Tomēr var paiet līdz pat 60 sekundēm, līdz tiek saņemti pirmie dati. Pēc noklusējuma tik bieži serveris nolasa konfigurācijas izmaiņas un sāk izpildīt jaunus vienumus.

Ja kolonnā 'Change' neredzat vērtību, iespējams, līdz šim ir saņemta tikai viena vērtība. Pagaidiet 30 sekundes, līdz tiek saņemta nākamā vērtība.

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

- vienuma 'Key' un 'Type of information' lauki ir aizpildīti tieši tā, kā redzams ekrānuzņēmumā;
- gan aģents, gan serveris darbojas;
- hosts statuss ir 'Monitored' un tā pieejamības ikona ir zaļa;
- hosts, kas atlasīts hosta filtrā, ir pareizs;
- vienums ir iespējots.

Grafiki

Kad vienums kādu laiku darbojas, iespējams, ir pienācis laiks apskatīt kaut ko vizuālu. Vienkāršie grafiki ir pieejami jebkuram uzraudzītam skaitliskam vienumam bez papildu konfigurēšanas. Šie grafiki tiek ģenerēti izpildes laikā.

Lai skatītu grafiku, dodieties uz *Monitoring* > *Latest data* un noklikšķiniet uz saites *Graph* blakus vienumam.



Jauns trigeris

Pārskats

Šajā sadaļā jūs uzzināsiet, kā iestatīt trigeri.

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

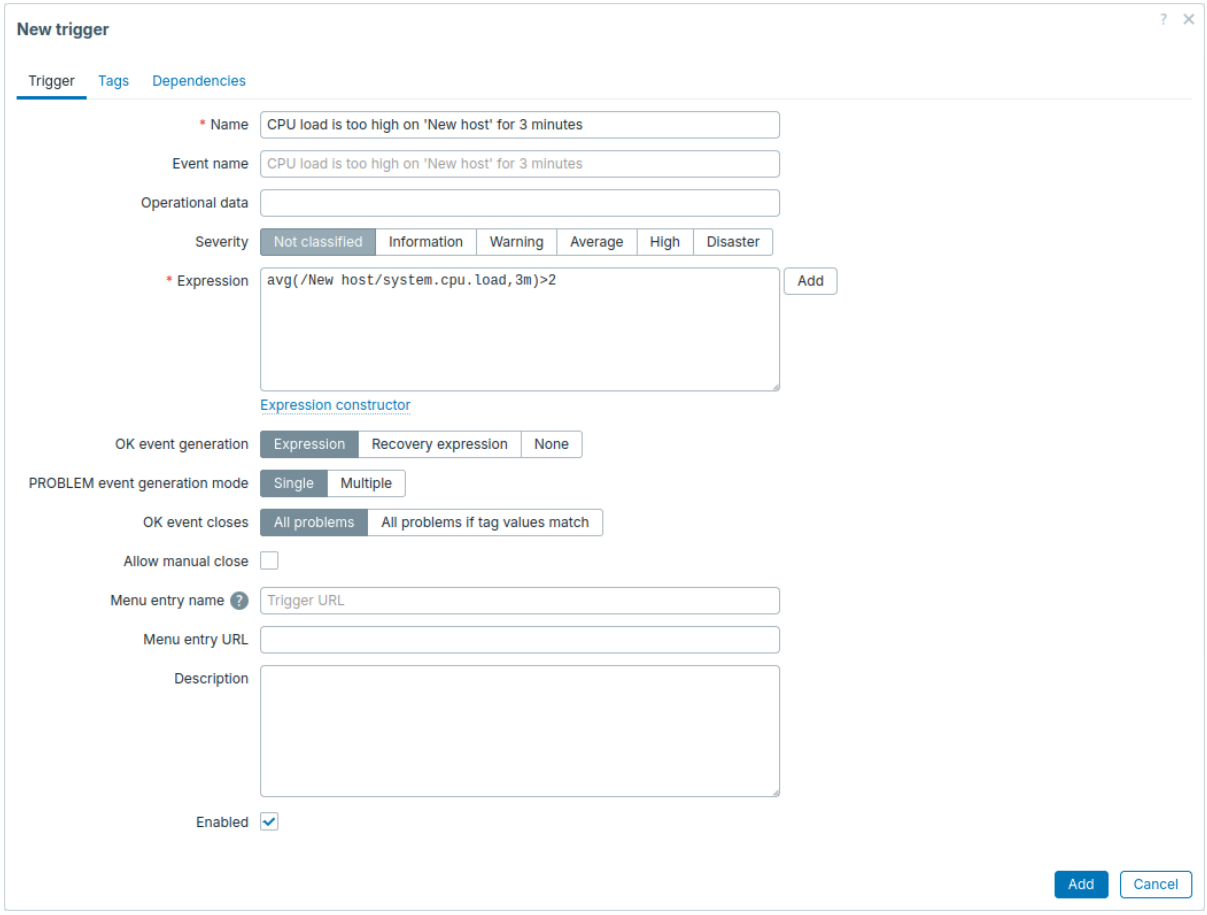
Ja ienākošie dati pārsniedz šo līmeni, trigeris tiks "aktivizēts" vai pāries stāvoklī 'Problem' - norādot, ka ir noticis kaut kas, kam var būt nepieciešama uzmanība. Ja līmenis atkal ir pieņemams, trigeris atgriežas stāvoklī 'Ok'.

Trigeru pievienošana

Lai konfigurētu trigeri mūsu vienumam:

1. Dodieties uz *Datu vākšana* > *Hosts*.
2. Noklikšķiniet uz saites *Trigeri* rindā *New host*.
3. Noklikšķiniet uz *Izveidot trigeri*.

Tiek atvērta trigera konfigurācijas forma.



Mūsu trigerim šeit jānorāda šāda būtiskā informācija:

Nosaukums

- Kā vērtību ievadiet *CPU load is too high on 'New host' for 3 minutes*. Tas būs trigera nosaukums, kas tiks rādīts 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. Šeit vienuma atslēga (`system.cpu.load`) tiek izmantota, lai atsauktos uz vienumu. Šī konkrētā izteiksme būtībā nozīmē, ka problēmas slieksnis tiek pārsniegts, ja CPU slodzes vidējā vērtība 3 minūšu laikā ir lielāka par 2. Vairāk par trigera izteiksmju sintaksi varat uzzināt šeit.

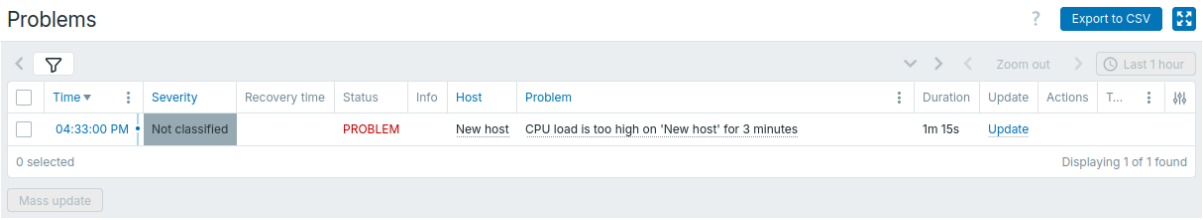
Citām opcijām var atstāt nemainītas to noklusējuma vērtības.

Kad esat pabeidzis, noklikšķiniet uz *Pievienot*. Jaunajam trigerim vajadzētu parādīties trigera sarakstā.

Trigeris stāvokļa attēlošana

Kad trigeris ir definēts, jūs varat vēlēties redzēt tā stāvokli.

Ja CPU noslodze ir pārsniegusi sliekšņa līmeni, ko definējāt trigerī, problēma tiks parādīta *Monitoring* > *Problems*.



Mirgošana statusa kolonnā norāda uz nesenu trigeris statusa maiņu, kas notikusi pēdējo 30 minūšu laikā.

Problēmas paziņojuma saņemšana

Pārskats

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

Tā kā vienumi apkopo datus un trigeri ir izstrādāti tā, lai ”nostrādātu” problēmsituāciju gadījumā, būtu noderīgi arī ieviest kādu brīdināšanas mehānismu, kas paziņo par svarīgiem notikumiem pat tad, ja Zabbix lietotāja saskarne netiek aktīvi pārbaudīta.

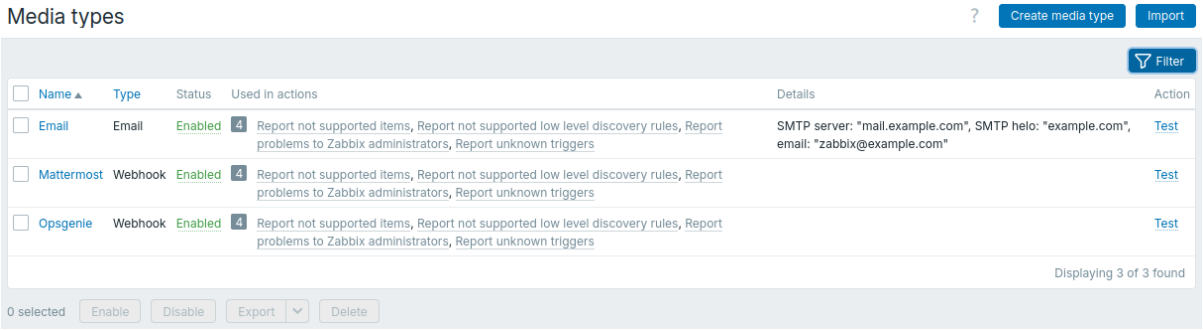
To nodrošina paziņojumi.

Tā kā e-pasts ir vispopulārākais problēmu paziņojumu piegādes veids, mēs uzzināsim, kā iestatīt e-pasta paziņojumu.

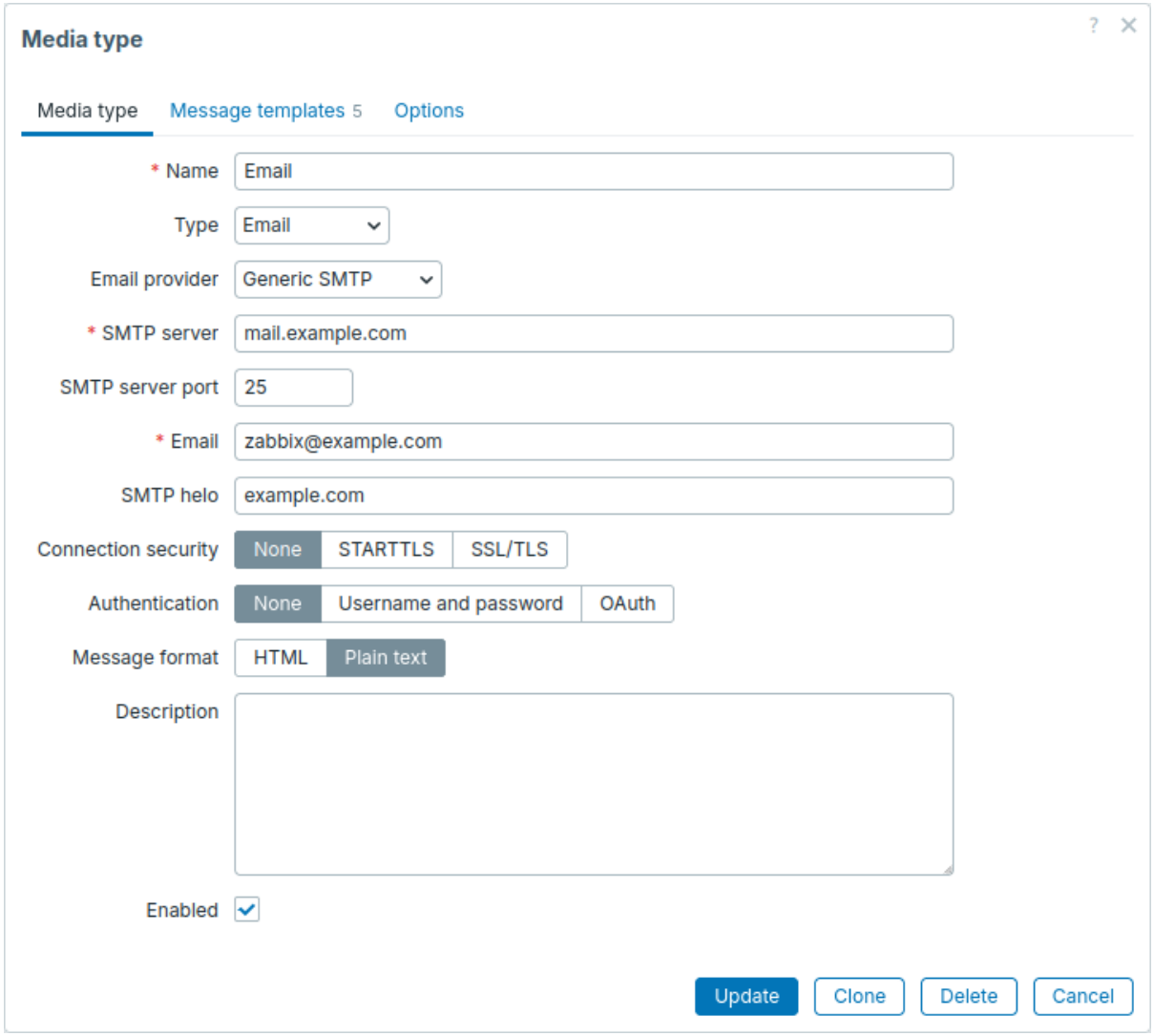
E-pasta iestatījumi

Sākotnēji Zabbix ir vairāki iepriekš definēti paziņojumu piegādes veidi. Email ir viens no tiem.

Lai konfigurētu e-pasta iestatījumus, dodieties uz *Alerts* > *Media types* un sarakstā ar iepriekš definētajiem mediju veidiem noklikšķiniet uz *Email*.



Tiek atvērta e-pasta iestatījumu konfigurācijas forma.



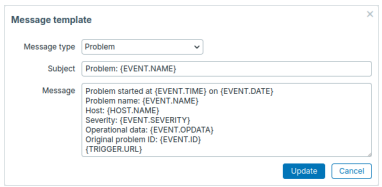
Visi obligātie ievades lauki ir atzīmēti ar sarkanu zvaigznīti.

Cilnē *Media type* iestatiet *SMTP server*, *SMTP helo* un *Email* vērtības atbilstoši savai videi.

Note:
Vērtība laukā *Email* tiks izmantota kā 'From' adrese paziņojumiem, kas nosūtīti no Zabbix.

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

Noklikšķiniet uz *Add*, lai izveidotu ziņojuma veidni, un kā ziņojuma tipu atlasiet *Problem*.

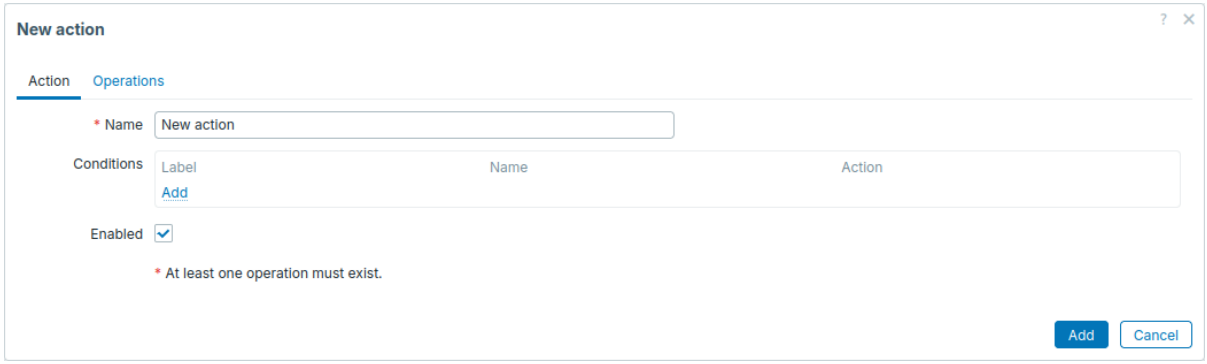


Kad esat gatavs, noklikšķiniet uz *Add* un saglabājiet formu.

Tagad esat konfigurējis *Email* kā darba mediju veidu. Mediju veids arī ir jāsaista ar lietotājiem, definējot konkrētas piegādes adreses (kā mēs 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 *Alerts* > *Actions* > *Trigger actions* un noklikšķiniet *Create action*.

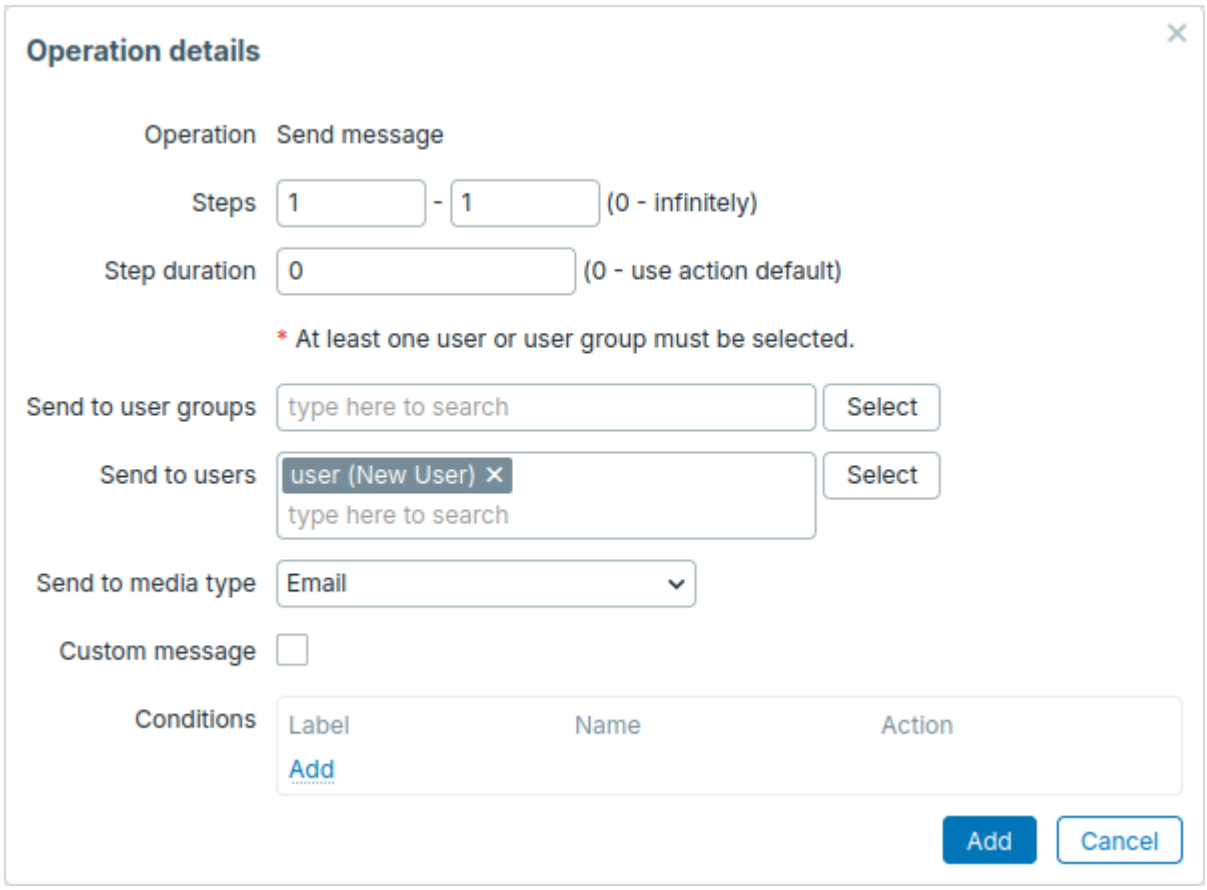


Visi obligātie ievades lauki ir atzīmēti ar sarkanu zvaigznīti.

Šajā formā ievadiet darbības nosaukumu.

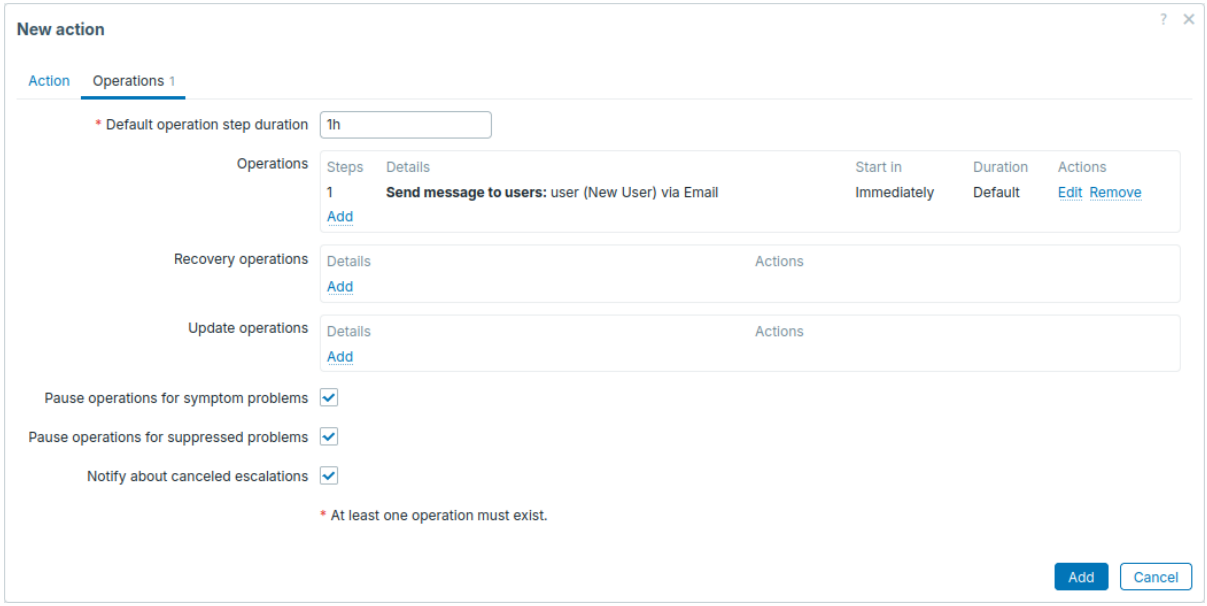
Visvienkāršākajā gadījumā, ja nepievienojam papildu konkrētākus nosacījumus, darbība tiks izpildīta pie jebkuras trigeris izmaiņas no 'Ok' uz 'Problem'.

Tomēr mums vēl jādefinē, ko darbībai vajadzētu darīt - tas tiek iestatīts cilnē *Operations*. Noklikšķiniet *Add* blokā *Operations*, lai atvērtu jaunu darbības formu.



Visi obligātie ievades lauki ir atzīmēti ar sarkanu zvaigznīti.

Šeit blokā *Send to Users* noklikšķiniet *Select* un atlasiet lietotāju ('user'), ko esam definējuši. Kā *Send to media type* vērtību atlasiet "Email". Kad tas ir izdarīts, noklikšķiniet *Add*, un darbība ir jāpievieno:



Tas ir viss vienkāršas darbības konfigurācijai, tāpēc darbības formā noklikšķiniet *Add*.

Paziņojuma saņemšana

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

Atveriet konsoli savā host un izpildiet:

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

Jūs varat palaist vienu vai vairākus no šiem procesiem.

Tagad dodieties uz *Monitoring* > *Latest data* un skatiet, kā ir palielinājušās 'CPU Load' vērtības. Atcerieties, lai mūsu trigeris "nostrādātu", 'CPU Load' vērtībai 3 minūtes pēc kārtas ir jābūt lielākai par '2'. Kad tas notiks:

- sadaļā *Monitoring* > *Problems* jūs redzēsiet trigeri ar mirgojošu `Problem` stāvokli;
- jūs saņemsiet problēmas paziņojumu savā e-pastā.

Attention:
Ja paziņojumi nedarbojas:

- vēlreiz pārbaudiet, vai gan e-pasta iestatījumi, gan darbība ir pareizi konfigurēti
- pārliecinieties, ka lietotājam, ko izveidojāt, ir vismaz lasīšanas tiesības uz host, kas ģenerēja notikumu, kā norādīts solī *Lietotāja pievienošana*. Lietotājam, kas ir daļa no lietotāju grupas 'Zabbix administrators', ir jābūt vismaz lasīšanas piekļuvei host grupai 'Linux servers', kurai pieder mūsu host.
- Papildus tam varat apskatīt darbību žurnālu, dodoties uz *Reports* > *Action log*.

Jauna veidne

Pārskats

Šajā sadaļā uzzināsiet, kā iestatīt veidni.

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

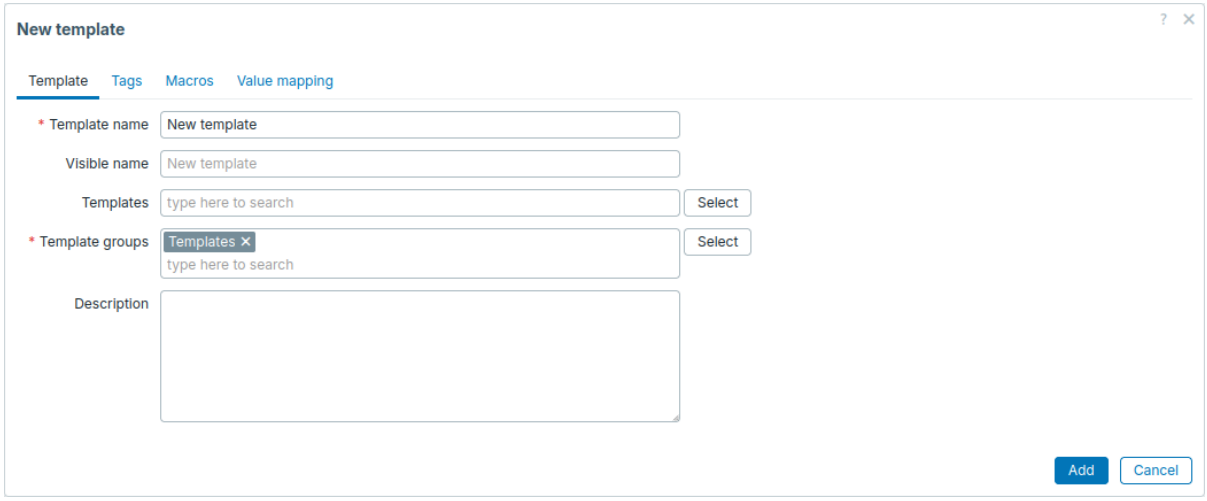
Lai gan visas šīs darbības pašas par sevi nodrošina lielu elastību, var šķist, ka nepieciešamības gadījumā, piemēram, tūkstoš hostu, tās ir pārāk daudz. Noderētu automatizācija.

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

Kad veidne tiek piesaistīta hostam, hosts pārmanto visas veidnes entītijas. Tādējādi iepriekš sagatavotu pārbaužu kopu var lietot ļoti ātri.

Veidnes pievienošana

Lai sāktu darbu ar veidnēm, vispirms ir jāizveido viena veidne. Lai to izdarītu, sadaļā *Data collection* > *Templates* noklikšķiniet uz *Create template*. Tiks atvērta veidnes konfigurācijas forma.



Visi obligātie ievades lauki ir atzīmēti ar sarkanu zvaigznīti.

Šeit jāievada šādi obligātie parametri:

Template name

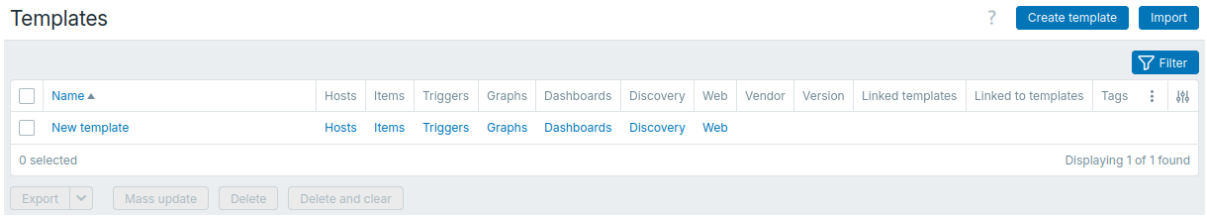
- Ievadiet veidnes nosaukumu. Ir atļauti burti, cipari, atstarpes un pasvītras.

Template groups

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

Note:
Piekļuves atļaujas veidņu grupām tiek piešķirtas lietotāju grupas konfigurācijā cilnē **Template permissions** tādā pašā veidā kā hosts atļaujas. Visas piekļuves atļaujas tiek piešķirtas grupām, nevis atsevišķām veidnēm, tāpēc veidnes iekļaušana vismaz vienā grupā ir obligāta.

Kad esat pabeidzis, noklikšķiniet uz *Add*. Jūsu jaunajai veidnei vajadzētu būt redzamai veidņu sarakstā. Varat arī izmantot filtru, lai atrastu savu veidni.

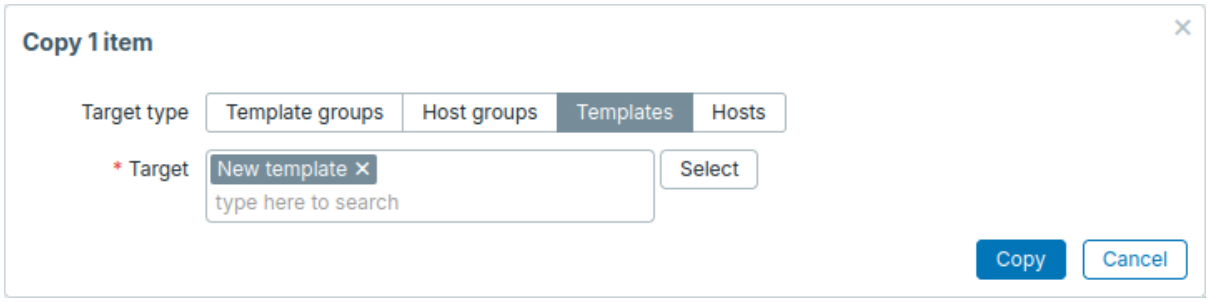


Kā redzat, veidne tur ir, taču tajā vēl nekā nav - ne vienumu, ne trigeru, ne citu entītiju.

Vienuma pievienošana veidnei

Lai pievienotu vienumu veidnei:

1. Dodieties uz *Datu apkopošana* > *Hosti*.
2. Sarakstā atrodiet “New host” un noklikšķiniet uz hipersaites *Vienumi*.
3. Sarakstā atzīmējiet vienuma “CPU Load” izvēles rūtiņu.
4. Zem saraksta noklikšķiniet uz *Kopēt*.
5. Atlasiet cilni *Veidnes*.
6. Atlasiet veidni, uz kuru kopēt vienumu.



Visi obligātie ievades lauki ir atzīmēti ar sarkanu zvaigznīti. 7. Noklikšķiniet uz *Kopēt*.

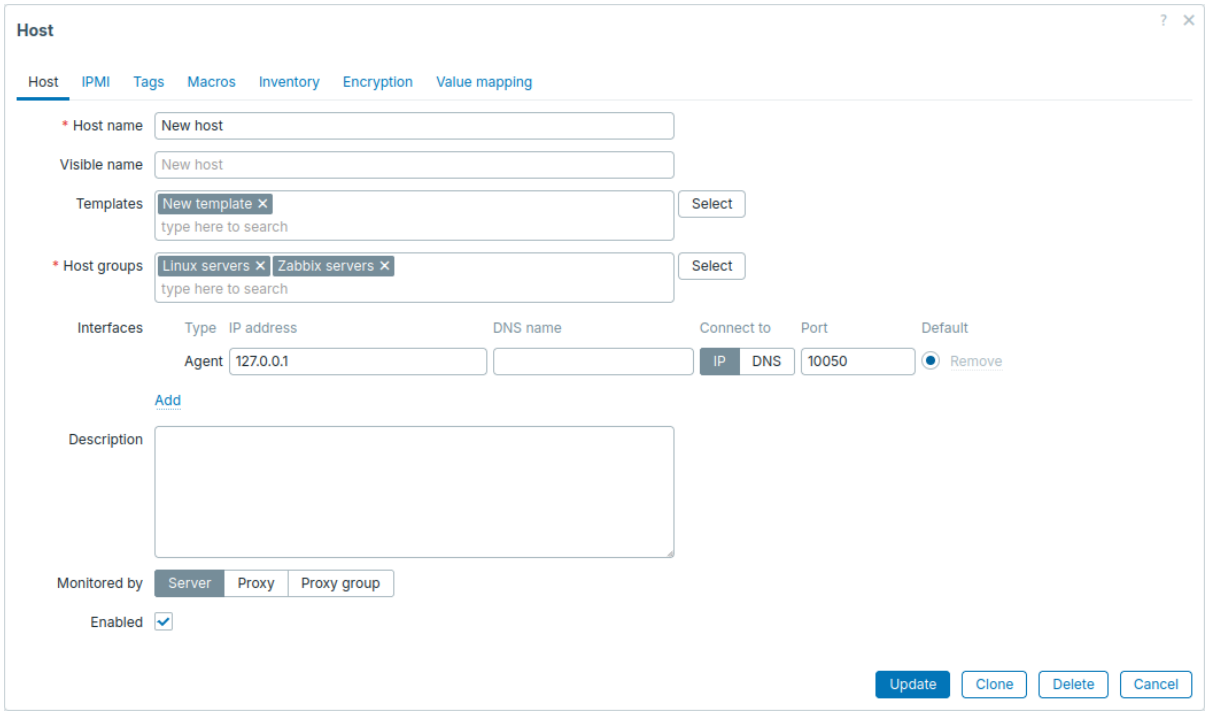
Ja tagad dodaties uz *Datu apkopošana* > *Veidnes*, veidnei “New template” vajadzētu būt vienam jaunam vienumam.

Pagaidām apstāsimies pie viena vienuma, taču līdzīgā veidā veidnei varat pievienot jebkurus citus vienumus, trigerus vai citas entītijas, līdz tā veido pietiekami pilnīgu entītiju kopu konkrētam mērķim (operētājsistēmas uzraudzībai, vienas lietojumprogrammas uzraudzībai).

Veidnes saistīšana ar hostu

Kad veidne ir gatava, tā tikai jāpievieno hostam. Lai to izdarītu:

1. Dodieties uz *Datu apkopošana* > *Hosti*.
2. Noklikšķiniet uz *Jauns hosts*, lai atvērtu tā konfigurācijas veidlapu, un atrodiet lauku *Veidnes*.
3. Laukā *Veidnes* sāciet rakstīt *Jauna veidne*. Jaunizveidotās veidnes nosaukumam vajadzētu parādīties nolaižamajā sarakstā.
4. Ritiniet uz leju, lai to atlasītu. Pārliecinieties, ka tā parādās laukā *Veidnes*.



5. Noklikšķiniet uz *Atjaunināt* veidlapā, lai saglabātu izmaiņas.

Tagad veidne ir pievienota hostam kopā ar visām tajā ietvertajām entītijām.

Šādā veidā to var lietot arī jebkuram citam hostam. Jebkuras izmaiņas vienumos, trigeros un citās entītijās veidnes līmenī tiks izplatītas uz hostiem, ar kuriem veidne ir saistīta.

Iepriekš definētu veidņu saistīšana ar hosts

Kā, iespējams, esat pamanījis, Zabbix tiek piegādāts ar iepriekš definētu veidņu kopu dažādām OS, ierīcēm un lietotnēm. Lai ļoti ātri sāktu uzraudzību, varat saistīt atbilstošo veidni ar hostu, taču ņemiet vērā, ka šīs veidnes ir jāpielāgo jūsu videi. Dažas pārbaudes var nebūt nepieciešamas, un aptaujas intervāli var būt pārāk bieži.

Papildu informācija par veidnēm ir pieejama.

Apache web server uzraudzība

Lai sāktu Apache web servera uzraudzību, veiciet šādu procesu:

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

Lai izmantotu šo rokasgrāmatu, jums ir jābūt instalētam Zabbix serverim un lietotāja saskarnei installed. Jums ir jāpiesakās Zabbix kā noklusējuma *Admin* lietotājam.

Piekļuve Apache statistikai

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

Apache serverī 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 sistēmās	RHEL bāzētās sistēmās
`apache2ctl -M | grep status`	`httpd -M | grep status`

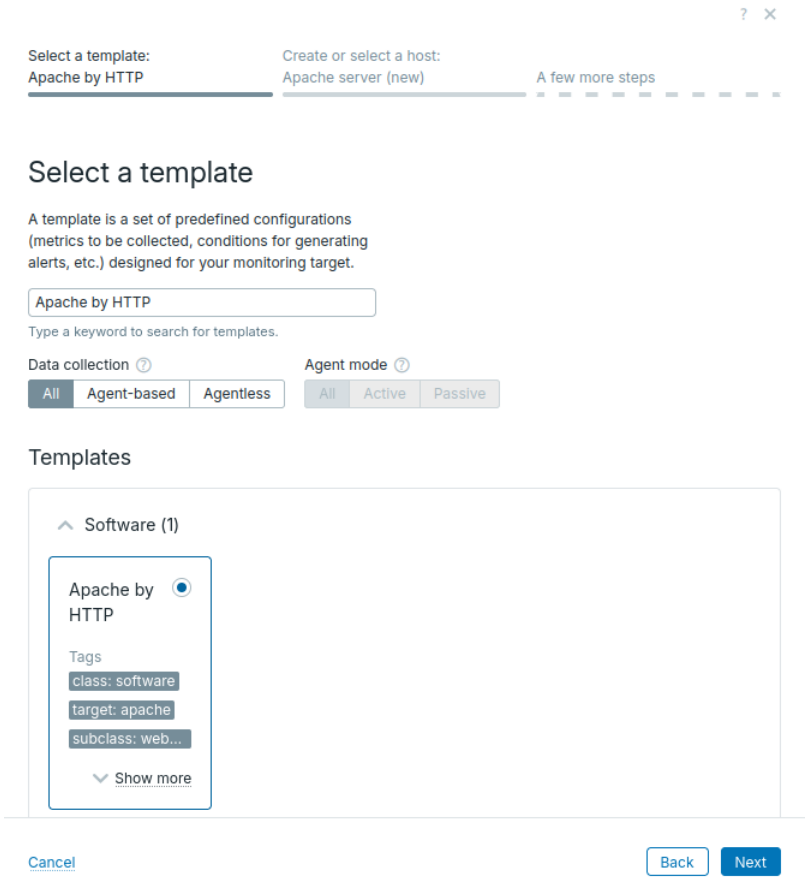
Ja izvades rezultātā redzat "`status_module (shared)`", tad statusa modulis jau ir iespējots. Ja nē, tad jums ir jāiespējo statusa modulis.

Apache monitoringa veidnes lietošana

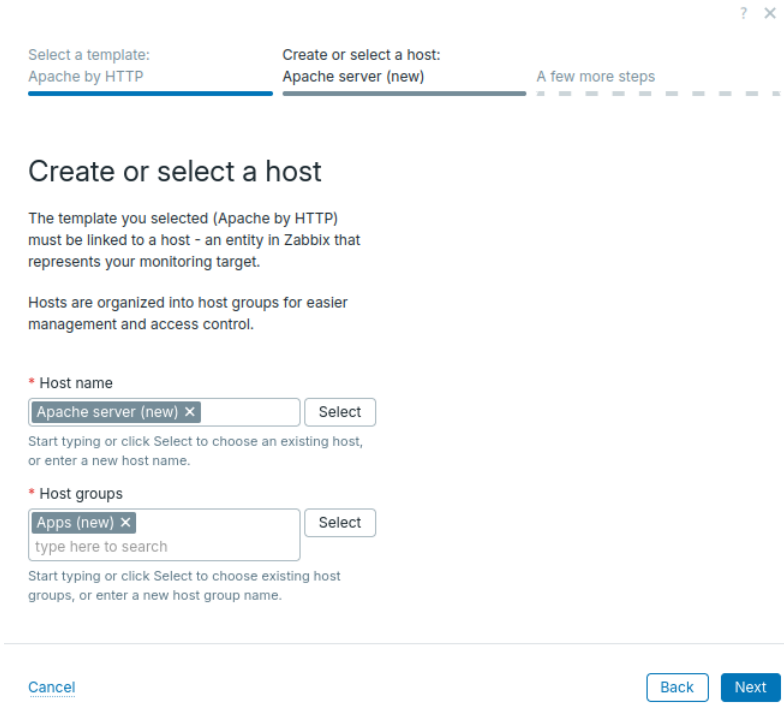
Zabbix šī veidne ir gatava lietošanai. Tajā ir iepriekš konfigurēti elementi, kas ir gatavi datu vākšanai, vizualizācijai un analīzei.

Lai lietotu veidni:

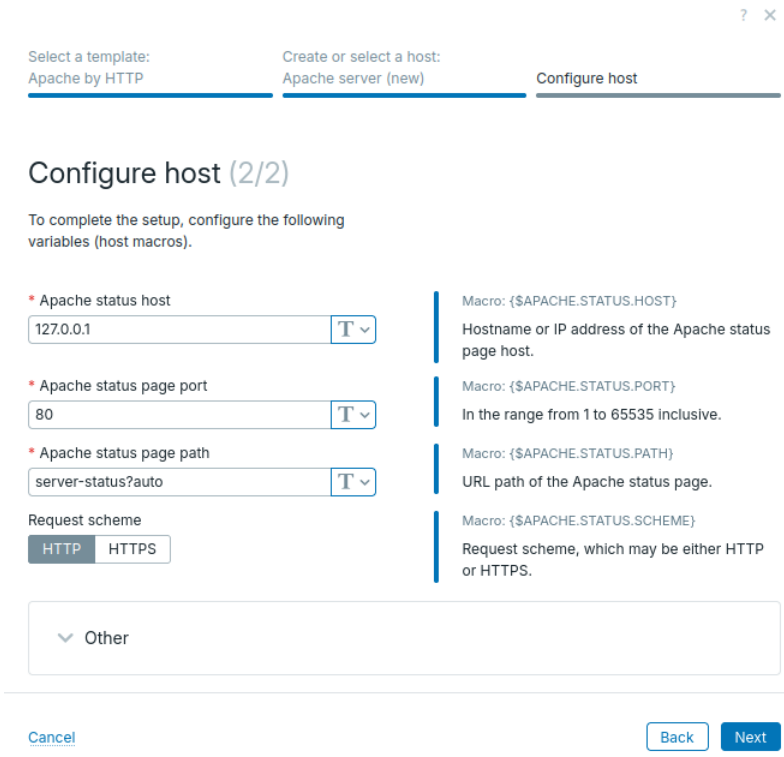
1. Startējiet Host Wizard sadaļā *Datu vākšana* > *Hosts*.
2. Atlasiet veidni *Apache by HTTP*.



3. Izveidojiet Apache hostu un pievienojiet tam grupu. Šis virtuālais hosts saturēs ar Apache saistītus datus.



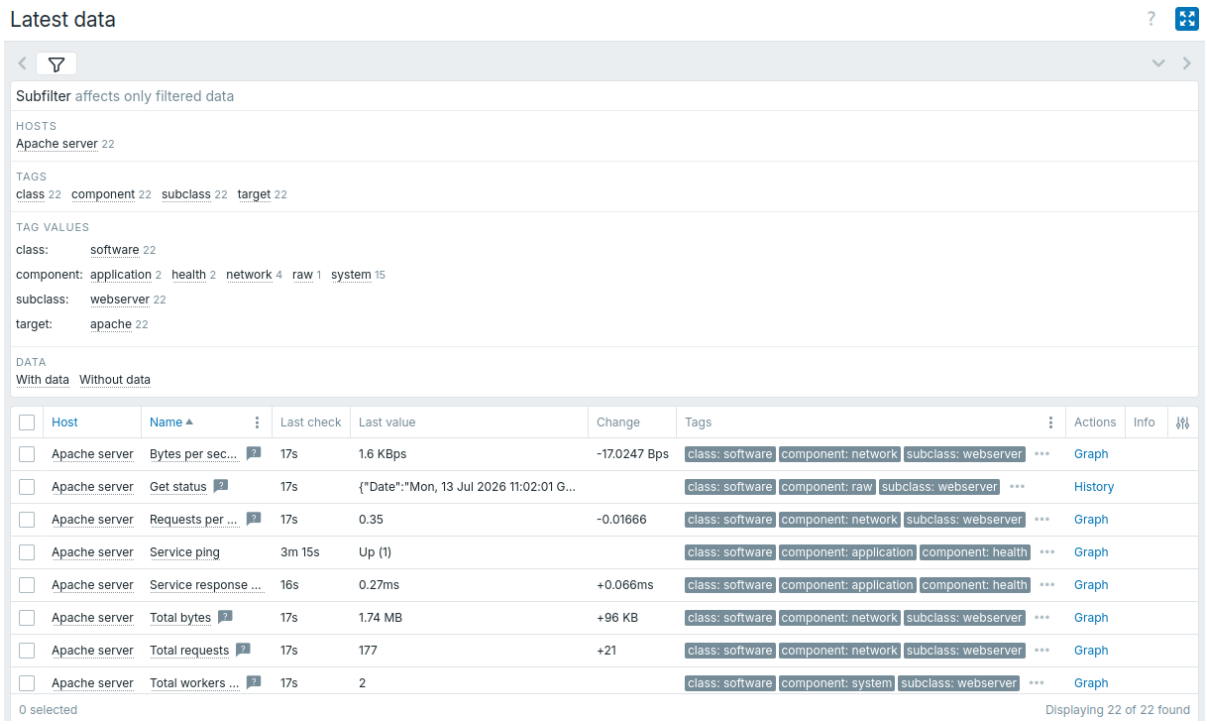
4. Konfigurējiet hostu. Svarīgi: ievadiet Apache tīmekļa servera adresi (127.0.0.1 lokālajā datorā, IP adresi attālajā datorā).



Pārējos iestatījumus varat atstāt noklusējuma vērtībās.

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

Skatīt jaunākos datus



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 server monitoring

Lai sāktu Linux servera uzraudzību, izpildiet šo procesu:

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

Lai sāktu uzraudzīt attālinātu Linux serveri, jums jāzina tā IP adrese un Zabbix servera IP adrese.

Linux servera uzraudzībai tiks izmantots Zabbix aģents. Norādījumi aģenta instalēšanai tiks sniegti procesa laikā.

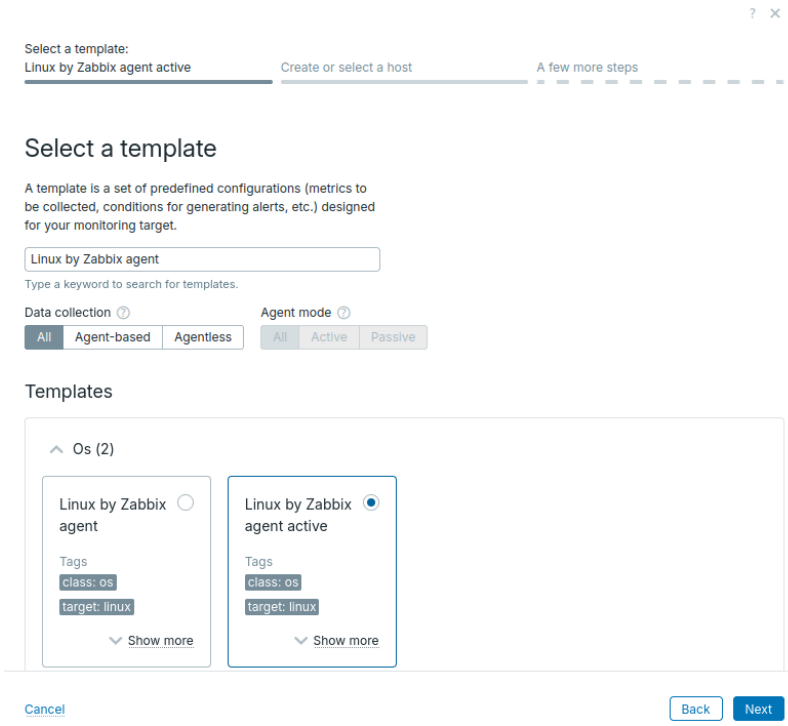
Lai izmantotu šo rokasgrāmatu, jums ir jābūt instalētam Zabbix serverim un lietotāja saskarnei installed. Jums ir jābūt pieteikušamies Zabbix kā noklusējuma *Admin* lietotājam.

Lietot Linux uzraudzības veidni

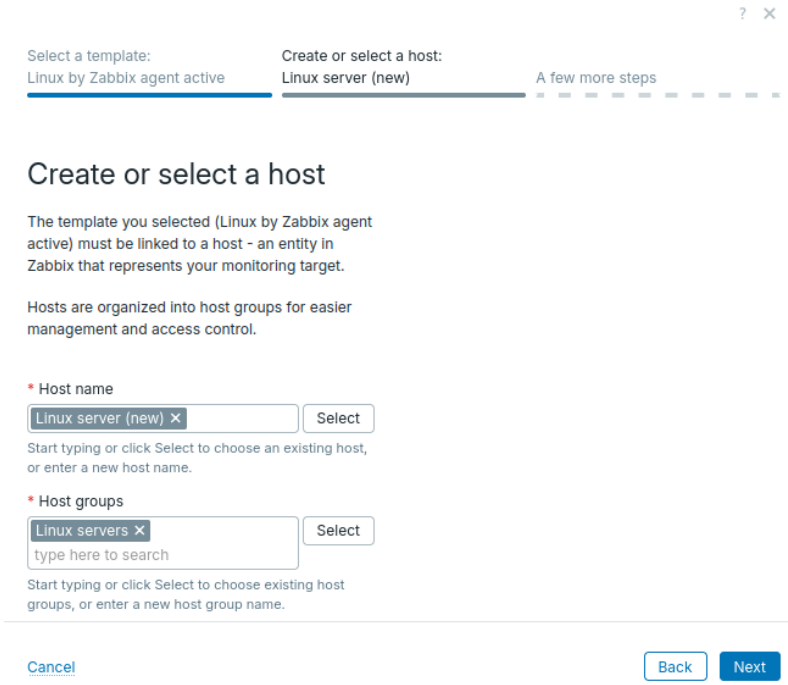
Zabbix šī veidne ir gatava lietošanai. Tā satur iepriekš konfigurētus elementus, kas ir gatavi datu vākšanai, vizualizācijai un analīzei.

Lai lietotu veidni:

1. Startējiet Host Wizard sadaļā *Data collection* > *Hosts*.
2. Atlasiet veidni *Linux by Zabbix agent active*.



3. Izveidojiet hostu Linux serverim un pievienojiet tam grupu. Šis virtuālais hosts saturēs visus datus, kas saistīti ar Linux servera uzraudzību.



4. Konfigurējiet un instalējiet aģentu.

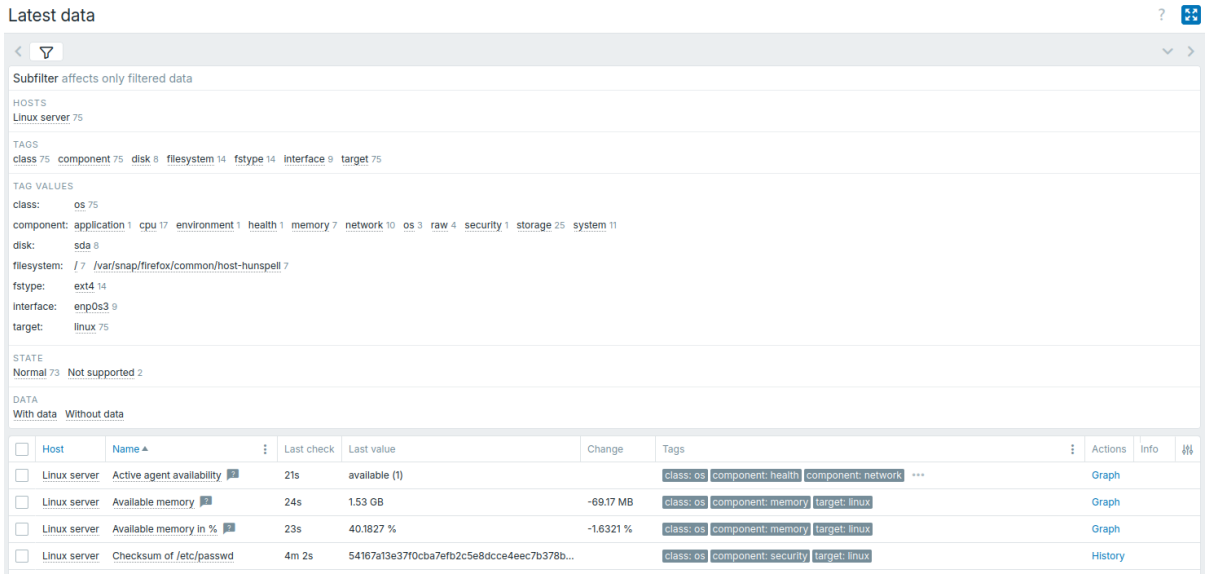


Lai instalētu aģentu, Linux serverī izpildiet norādīto komandu. Tā instalēs aģentu un konfigurēs aģenta savienojumu ar Zabbix serveri, izmantojot šifrēšanu.
 Ja aģents jau ir instalēts, šo soli varat izlaist (noklikšķiniet *Next*). Tomēr jums būs jānorāda iepriekš koplietotās atslēgas identitāte un iepriekš koplietotā atslēga, lai izveidotu savienojumu ar aģentu.

Pārējos iestatījumus varat atstāt noklusējuma vērtībās.

5. Pabeidziet hosta konfigurēšanu: noklikšķiniet *Create* un *Finish*.

Skatiet jaunākos datus



Apsveicam, Linux servera uzraudzība ir sākta!

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

MySQL servera uzraudzība

Lai sāktu MySQL servera uzraudzību, izpildiet šo procesu:

1. Izveidojiet ierobežotu MySQL lietotāju.
2. Pielietojiet MySQL uzraudzības veidni.
3. Apskatiet jaunākos datus.

Lai sāktu MySQL servera uzraudzību, 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 rokasgrāmatu, jums ir jābūt instalētam Zabbix serverim un lietotāja saskarnei installed. Jums jābūt pieteikušamies Zabbix kā noklusējuma *Admin* lietotājam.

Ierobežota MySQL lietotāja izveide

Ieteicams izveidot MySQL lietotāju, kura atļaujas ir ierobežotas tikai ar tām, kas nepieciešamas MySQL uzraudzībai. Aizstājiet <password> ar spēcīgu, unikālu paroli.

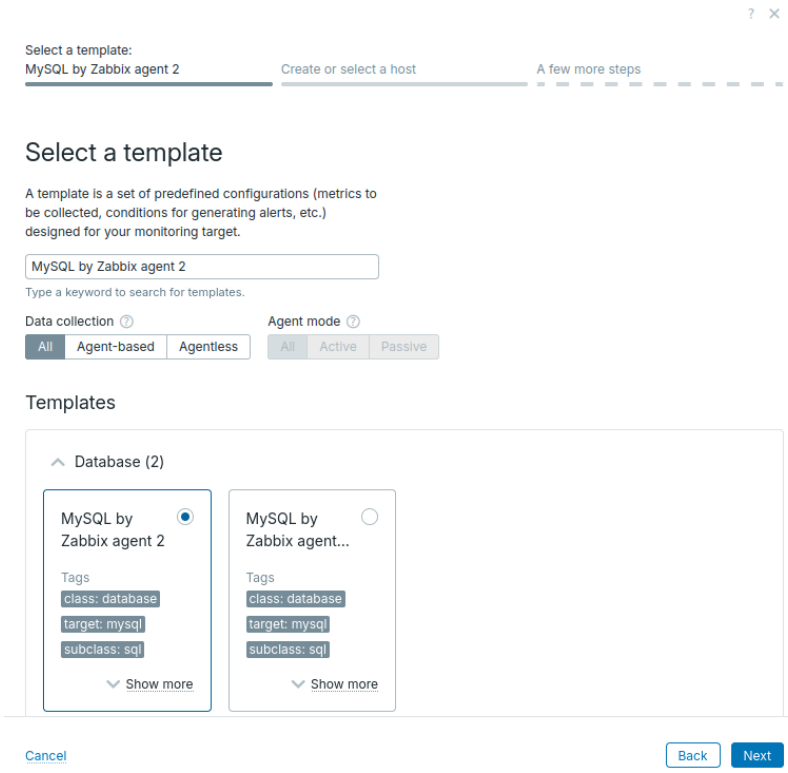
```
mysql> CREATE USER 'zbx_monitor'@'%' IDENTIFIED BY '<password>';
mysql> GRANT REPLICATION CLIENT,PROCESS,SHOW DATABASES,SHOW VIEW ON *.* TO 'zbx_monitor'@'%';
```

Lietot MySQL uzraudzības veidni

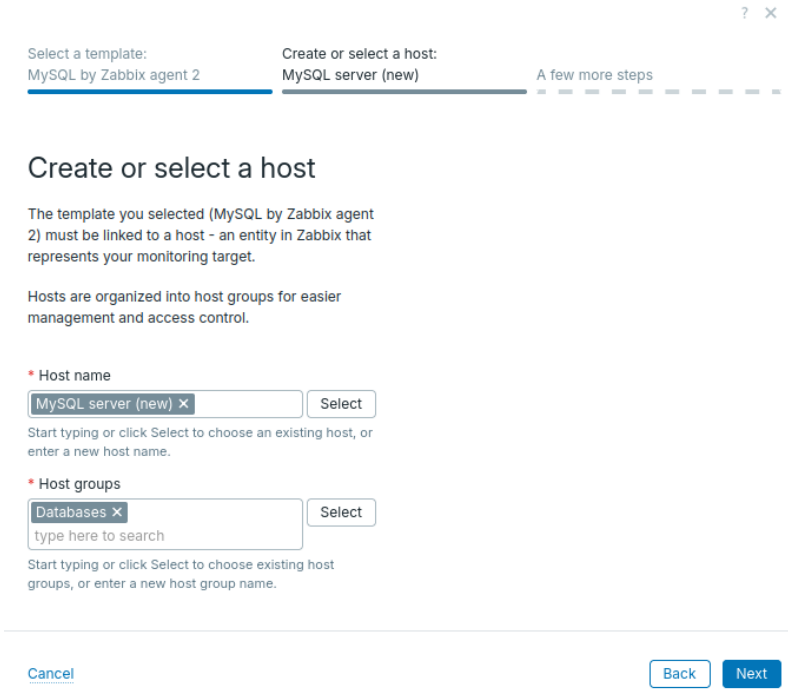
Zabbix šī veidne ir gatava lietošanai. Tā satur iepriekš konfigurētus elementus, kas ir gatavi datu vākšanai, vizualizācijai un analīzei.

Lai lietotu veidni:

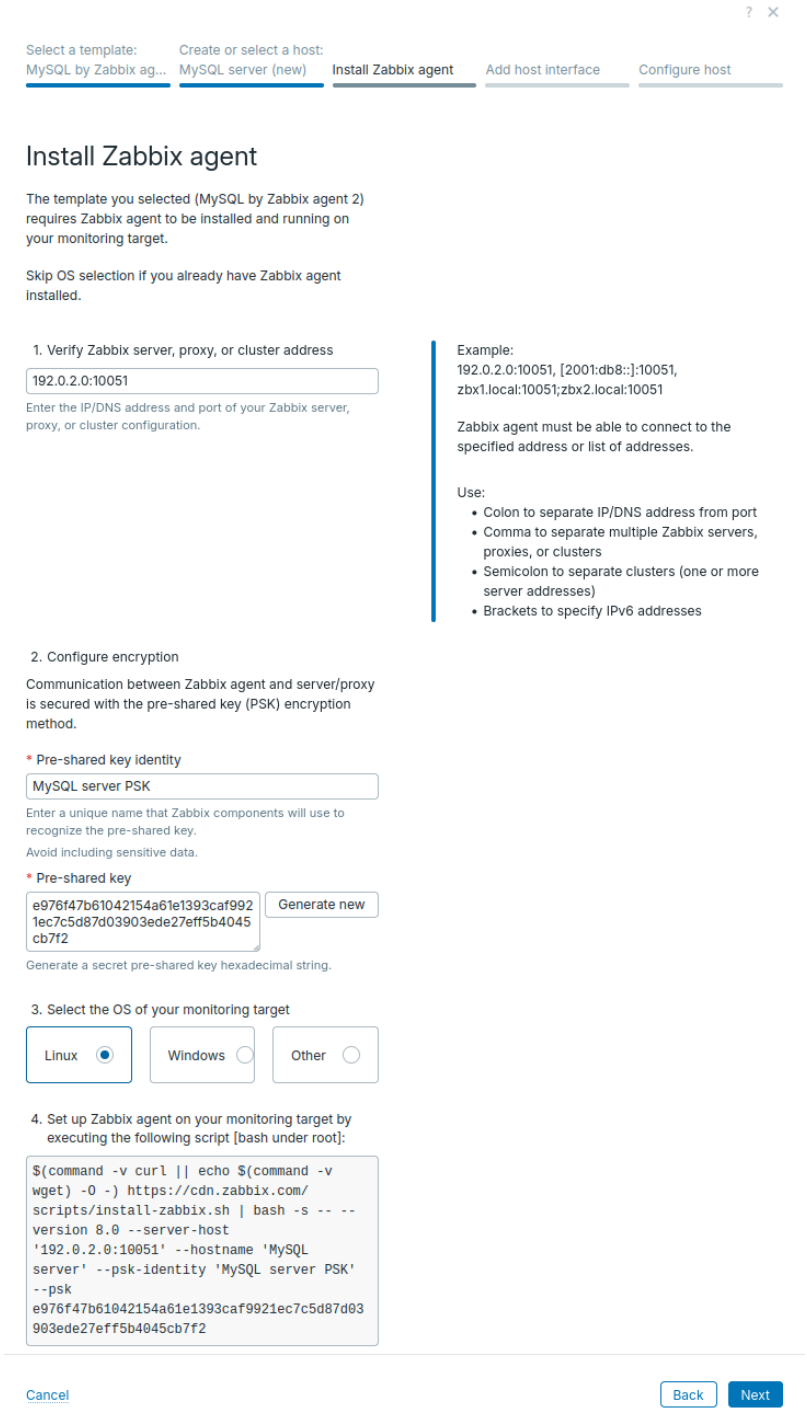
1. Startējiet Host Wizard sadaļā *Data collection* > *Hosts*.
2. Atlasiet veidni *MySQL by Zabbix agent 2*.



3. Izveidojiet MySQL servera hostu un pievienojiet tam grupu. Šis virtuālais hosts saturēs visus datus, kas saistīti ar MySQL servera uzraudzību.

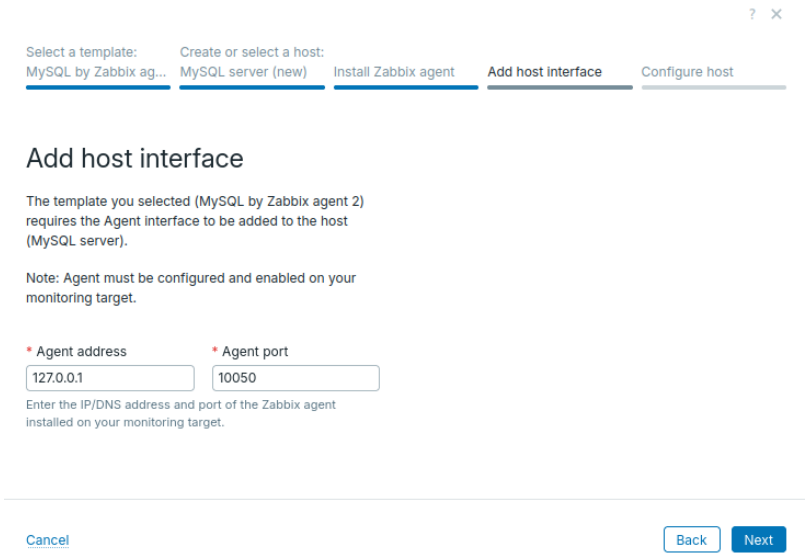


4. Konfigurējiet un instalējiet aģentu 2 (Zabbix aģents 2 nodrošina MySQL uzraudzību uzreiz pēc instalēšanas).

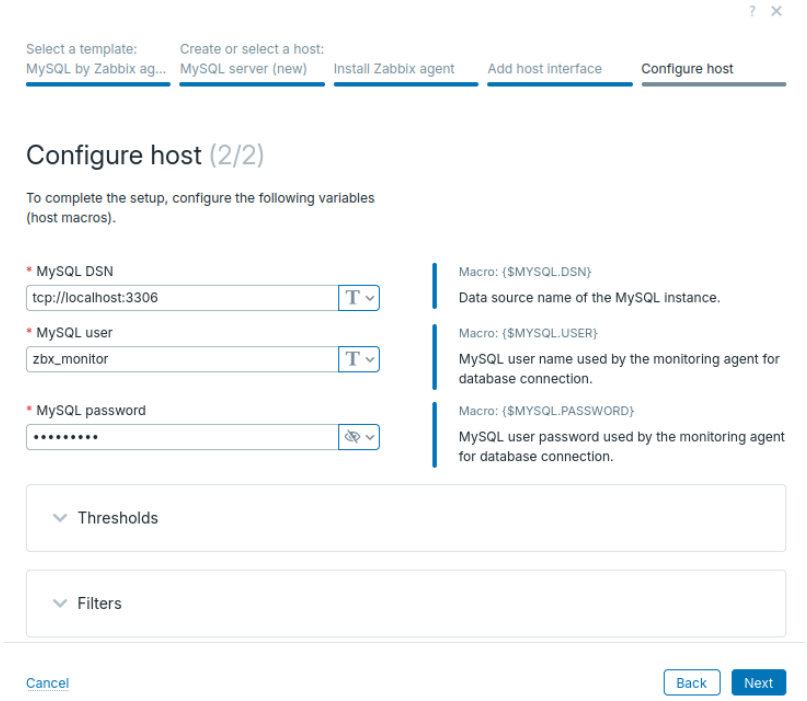


Lai instalētu aģentu, izpildiet norādīto komandu datorā, kurā darbojas MySQL serveris. Tā instalēs aģentu un konfigurēs aģenta savienojumu ar Zabbix serveri, izmantojot šifrēšanu.
 Ja aģents 2 jau ir instalēts, šo soli varat izlaist (noklikšķiniet *Next*). Tomēr jums tiks lūgts norādīt iepriekš koplietotās atslēgas identitāti un iepriekš koplietoto atslēgu, lai izveidotu savienojumu ar aģentu.

5. Ievadiet aģenta (Linux servera) IP adresi un portu.



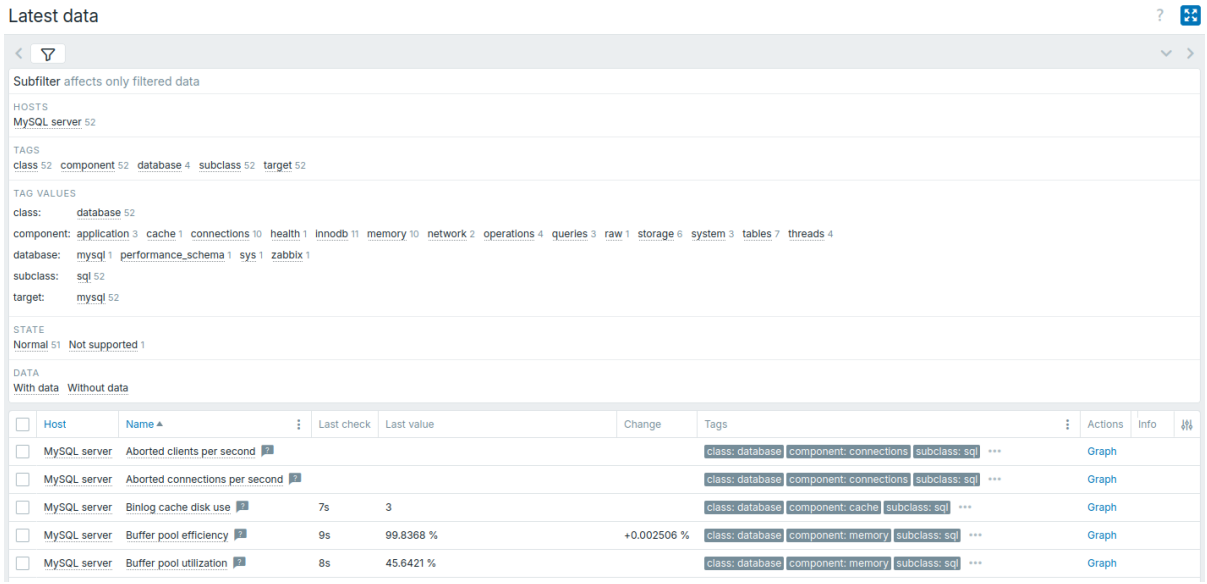
6. Pabeidziet hosta konfigurēšanu. Svarīgi: jums tiks lūgts ievadīt MySQL servera DSN, MySQL lietotāju un paroli. Norādiet lietotāju `zbx_monitor` ar jūsu izveidoto paroli.



Pārējos iestatījumus varat atstāt noklusējuma vērtībās.

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

Skatiet 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.

Vairāk ceļvežu

Šī dokumentācijas sadaļa satur īsas receptes Zabbix konfigurēšanai dažiem bieži nepieciešamiem uzraudzības mērķiem.

Tā ir veidota, domājot par jaunu Zabbix lietotāju, un to var izmantot kā navigācijas rīku citās dokumentācijas sadaļās, kurās ir informācija, kas nepieciešama uzdevuma atrisināšanai.

Ir pieejamas šādas ātrās uzziņas rokasgrāmatas:

- Linux uzraudzība ar Zabbix aģents
- Windows uzraudzība ar Zabbix aģents
- Apache uzraudzība, izmantojot HTTP
- MySQL uzraudzība ar Zabbix aģents 2
- VMware uzraudzība ar Zabbix
- Tīkla trafika uzraudzība ar Zabbix
- Tīkla trafika uzraudzība ar Zabbix, izmantojot aktīvās pārbaudes
- Tīmekļa vietņu uzraudzība ar Browser vienumiem
- Tīmekļa vietņu sertifikātu uzraudzība ar Zabbix aģents 2 (pasīvi)
- Tīkla komutatora vai maršrutētāja uzraudzība ar Zabbix
- Windows notikumu žurnāla uzraudzība, izmantojot aktīvās pārbaudes

Pārraugiet Linux ar Zabbix aģentu

Ievads

Šajā lapā ir aprakstītas darbības, kas nepieciešamas, lai sāktu pamata Linux iekārtu uzraudzību ar Zabbix. Šajā pamācībā aprakstītās darbības var piemērot jebkurai uz Linux balstītai operētājsistēmai.

Kam šī rokasgrāmata ir paredzēta

Šī rokasgrāmata ir paredzēta jauniem Zabbix lietotājiem un satur minimālo darbību kopu, kas nepieciešama, lai iespējotu pamata jūsu Linux iekārtas uzraudzību. Ja meklējat plašas pielāgošanas iespējas vai nepieciešama sarežģītāka konfigurācija, skatiet Zabbix rokasgrāmatas sadaļu Configuration.

Priekšnosacījumi

Pirms turpināt šo rokasgrāmatu, jums ir jālejupielādē un jāinstalē Zabbix serveris un Zabbix lietotāja saskarne saskaņā ar jūsu operētājsistēmas norādījumiem.

Zabbix aģenta instalēšana

Zabbix aģents ir process, kas atbild par datu apkopošanu.

Pārbaudiet savu Zabbix serveris versiju:

```
zabbix_server -V
```

Instalējiet tādas pašas versijas Zabbix aģents (ieteicams) Linux datorā, kuru vēlaties uzraudzīt. Atkarībā no jūsu uzraudzības vajadzībām tas var būt tas pats dators, kurā ir instalēts Zabbix serveris, vai pilnīgi cits dators.

Izvēlieties piemērotāko instalēšanas metodi:

- Palaist kā Docker konteineru - skatiet pieejamo attēlu sarakstu Zabbix Docker repozitorijā.
- Instalēt no Zabbix pakotnēm (pieejamas Alma Linux, CentOS, Debian, Oracle Linux, Raspberry Pi OS, RHEL, Rocky Linux, SUSE Linux Enterprise Server, Ubuntu).
- Kompilēt no pirmkoda.

Konfigurēt Zabbix uzraudzībai

Zabbix aģents var vākt metriku aktīvajā vai pasīvajā režīmā (vienlaikus).

> **Note:**
> Pasīvā pārbaude ir vienkāršs datu pieprasījums. Zabbix serveris vai starpniekserveris pieprasa noteiktus datus (piemēram, CPU noslodzi), un Zabbix aģents nosūta rezultātu atpakaļ serverim. Aktīvās pārbaudes prasa sarežģītāku apstrādi. Aģentam vispirms no servera(-iem) ir jāiegūst vienumu saraksts neatkarīgai apstrādei, un pēc tam tas masveidā nosūta datus atpakaļ. Papildu 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 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 aktīvajām vai pasīvajām Zabbix aģenta pārbaudēm.

Pasīvie pārbaudes veidi

Zabbix aģents

1. Atveriet aģenta konfigurācijas failu datorā, kurā ir instalēts aģents.

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

2. Parametram `Server` pievienojiet sava Zabbix servera IP adresi vai DNS nosaukumu (un neobligātu portu). Piemēram:

```
Server=192.0.2.1:10051
```

Zabbix aģents izmantos šo adresi, lai pieņemtu ienākošos savienojumus tikai no norādītajiem Zabbix serveriem vai starpniekserveriem datu izgūšanai.

3. Restartējiet Zabbix aģentu.

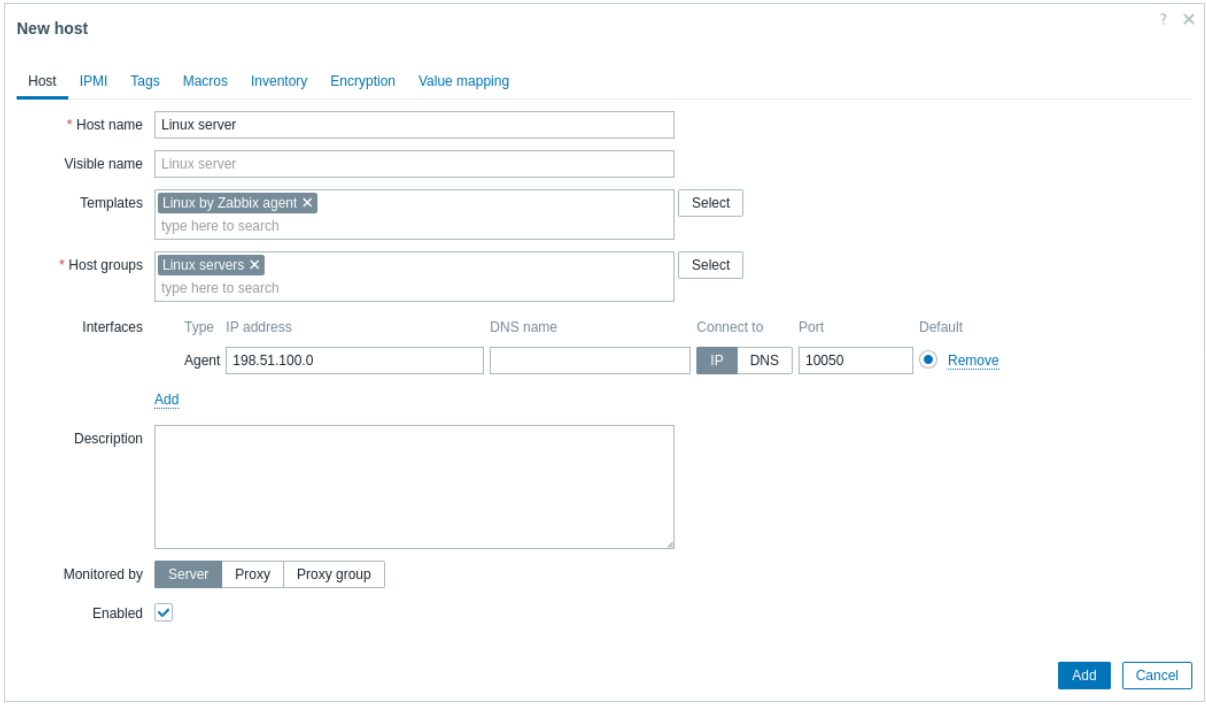
```
systemctl restart zabbix-agent
```

Zabbix lietotāja saskarne

1. Piesakieties Zabbix lietotāja saskarnē.

2. Izveidojiet hostu Zabbix tīmekļa lietotāja saskarnē.

- Laukā *Host name* ievadiet hosta nosaukumu (piem., ”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 (piem., ”Linux servers”).
- Parametrā *Interfaces* pievienojiet *Aģents* interfeisu un norādiet Linux iekārtas, kurā ir instalēts aģents, IP adresi vai DNS nosaukumu.



3. Noklikšķiniet uz *Add*, lai pievienotu hostu. Šis hosts attēlos uzraudzīto Linux iekārtu.

Aktīvās pārbaudes

Zabbix aģents

1. Atveriet aģenta konfigurācijas failu datorā, kurā ir instalēts aģents:

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

2. Parametram ServerActive pievienojiet sava Zabbix servera IP adresi vai DNS nosaukumu (un neobligāto portu). Piemēram:

```
ServerActive=192.0.2.1:10051
```

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, kuram jāsakrīt ar hosts nosaukumu, kas tiks definēts Zabbix lietotāja saskarnē. Mūsu piemērā tas ir:

```
Hostname=Linux server
```

Hostname vērtībai ir jāsakrīt, jo aktīvajām pārbaudēm Zabbix aģents to izmanto, lai no servera izgūtu pareizo hosts konfigurāciju. Konkrēti, aģents izveido savienojumu ar serveri un identificē sevi, izmantojot Hostname vērtību. Pēc tam serveris nodrošina šī hosts uzraudzības konfigurāciju. Ja šīs vērtības atšķiras, aģents nesaņems atbilstošo konfigurāciju, kā rezultātā trūks metriku vai radīsies uzraudzības problēmas.

4. Restartējiet Zabbix aģentu.

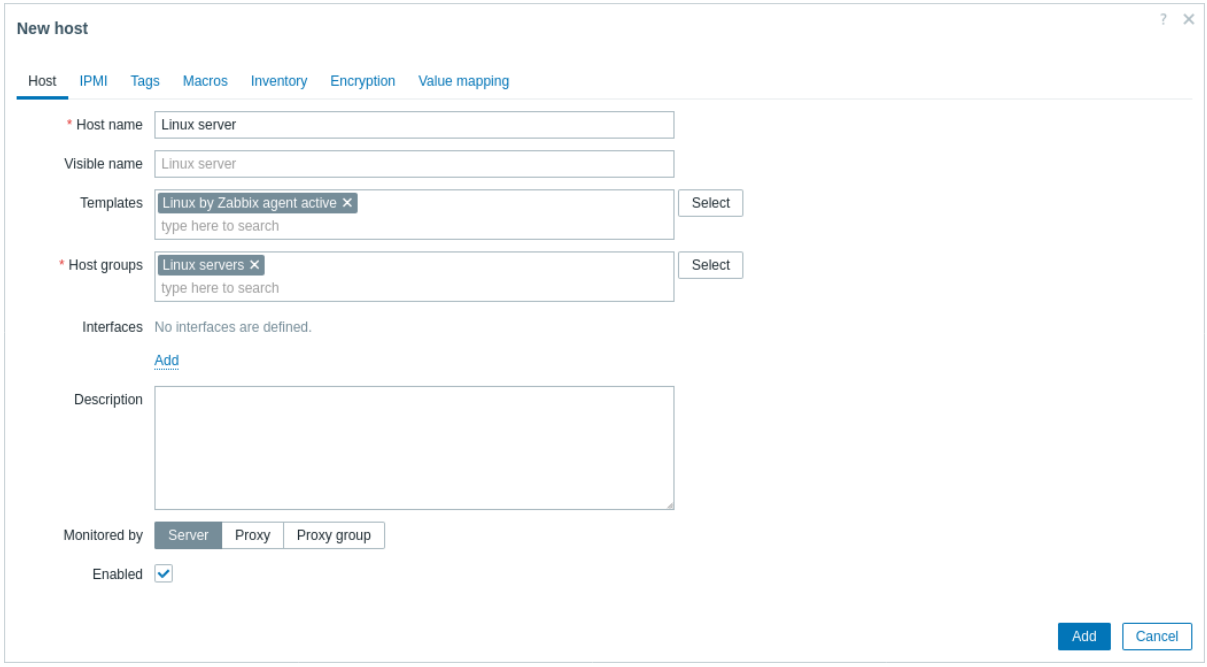
```
systemctl restart zabbix-agent
```

Zabbix lietotāja saskarne

1. Piesakieties Zabbix lietotāja saskarnē.

2. Izveidojiet hostu Zabbix tīmekļa saskarnē.

- Laukā *Host name* ievadiet hosta nosaukumu (piem., ”Linux serveris”), kas atbilst iepriekš aģenta konfigurācijas failā definētajai *Hostname* parametra vērtībai.
- Laukā *Templates* ievadiet vai atlasiet veidni ”Linux by Zabbix agent active”, kas tiks saistīta ar hostu.
- Laukā *Host groups* ievadiet vai atlasiet hostu grupu (piem., ”Linux servers”).
- Atstājiet *Interfaces* nedefinētu; interfeiss nav nepieciešams aktīvajām pārbaudēm, jo aģents uzsāk savienojumu ar serveri, nevis gaida savienojumu no servera.



3. Noklikšķiniet uz *Add*, lai pievienotu hostu. Šis hosts attēlos uzraudzīto Linux iekārtu.

Skatīt savāktos metriskos datus

Apsveicam! Šajā brīdī Zabbix jau uzrauga jūsu Linux iekārtu.

Lai skatītu savāktos metriskos datus, atveriet *Monitoring* > *Hosts* izvēlnes sadaļu un noklikšķiniet uz *Latest data* blakus hostam.

Name ▲	Interface	Availability	Tags	Status	Latest data	Problems
Linux server	127.0.0.1:10050	ZBX	class: os target: linux	Enabled	Latest data 64	1

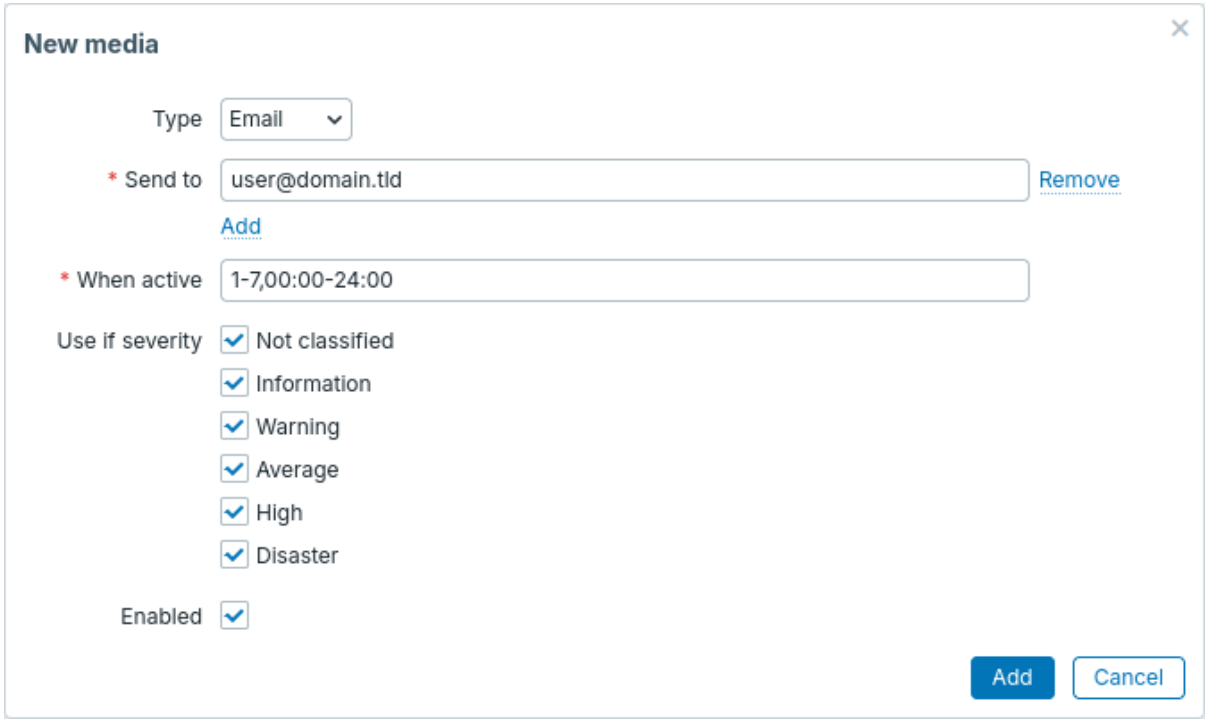
Šī darbība atvērs visu jaunāko metrisko datu sarakstu, kas savākti no Linux serveris hosta.

Host	Name ▲	Last check	Last value	Change	Tags
Linux server	/: Free inodes in %	54s	71.1694 %		component: storage filesystem: /
Linux server	/: Space utilization	53s	95.6273 %	+0.000327 %	component: storage filesystem: /
Linux server	/: Total space	52s	13.55 GB		component: storage filesystem: /
Linux server	/: Used space	51s	12.28 GB	+44 KB	component: storage filesystem: /
Linux server	Available memory	43s	2.36 GB	+24 KB	component: memory
Linux server	Available memory in %	42s	61.5978 %	+0.000398 %	component: memory

Iestatiet problēmu brīdinājumus

Zabbix var paziņot jums par problēmu jūsu infrastruktūrā, izmantojot dažādas metodes.
Šajā ceļvedī ir aprakstītas konfigurācijas darbības e-pasta brīdinājumu nosūtīšanai.

1. Dodieties uz *Lietotāja iestatījumi* > *Profils*, pārslēdzieties uz cilni *Multivide* un pievienojiet savu e-pastu.



2. Izpildiet ceļvedi par problēmas paziņojuma saņemšanu.

Nākamreiz, kad Zabbix noteiks problēmu, jums jāsaņem brīdinājums pa e-pastu.

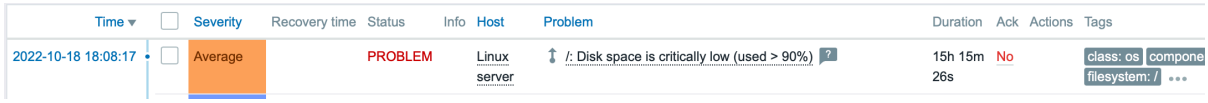
Pārbaudiet savu konfigurāciju

Linux vidē varat simulēt lielu CPU slodzi un tā rezultātā saņemt problēmas brīdinājumu, izpildot:

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

Iespējams, būs jāpalaiž vairāki md5sum procesi, lai CPU slodze pārsniegtu slieksni.

Kad Zabbix konstatēs problēmu, tā parādīsies sadaļā *Monitoring* > *Problems*.



Ja brīdinājumi ir konfigurēti, jūs saņemsiet arī problēmas paziņojumu.

Skatīt arī:

- Vienuma izveide - kā sākt papildu metriku uzraudzību (pielāgota uzraudzība bez veidnēm).
- Zabbix aģenta vienumi, Zabbix aģenta vienumi operētājsistēmai Windows - pilns metriku saraksts, ko varat uzraudzīt, izmantojot Zabbix aģentu operētājsistēmā Windows.
- Problēmu eskalācijas - kā izveidot daudzpakāpju brīdinājumu scenārijus (piem., vispirms nosūtīt ziņojumu sistēmas administratoram, pēc tam, ja problēma 45 minūšu laikā netiek atrisināta, nosūtīt ziņojumu datu centra vadītājam).
- Instalēšana no pakotnēm - kā instalēt Zabbix komponentus, izmantojot oficiālās RPM un DEB pakotnes dažādiem Linux distributīviem, nodrošinot piekļuvi jaunākajām funkcijām un kļūdu labojumiem.

Monitorējiet Windows ar Zabbix aģentu

Ievads

Šajā lapā ir aprakstītas darbības, kas nepieciešamas, lai sāktu Windows datoru pamata uzraudzību, izmantojot Zabbix.

Kam paredzēta šī rokasgrāmata

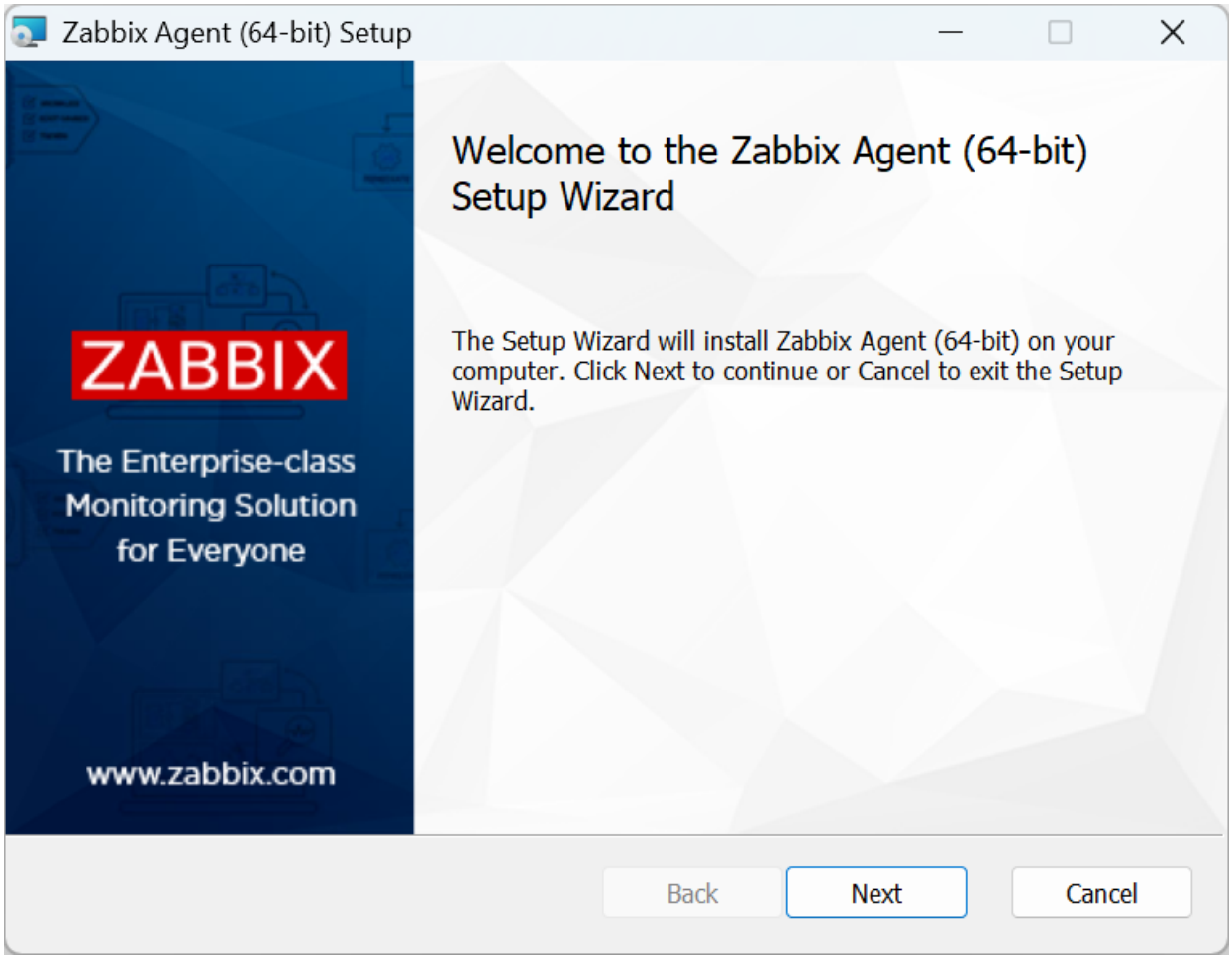
Šī rokasgrāmata ir paredzēta jaunajiem Zabbix lietotājiem un ietver minimālo darbību kopumu, kas nepieciešams, lai iespējotu Windows datora pamata uzraudzību. Ja meklējat plašas pielāgošanas iespējas vai nepieciešama sarežģītāka konfigurācija, skatiet Zabbix rokasgrāmatas sadaļu Configuration.

Priekšnosacījumi

Pirms turpināt šo instalēšanas rokasgrāmatu, jums ir jālejupielādē un jāinstalē Zabbix serveris un Zabbix lietotāja saskarne atbilstoši jūsu operētājsistēmas norādījumiem.

Instalēt Zabbix aģentu

Zabbix aģents ir process, kas atbild par datu apkopošanu. Tas jāinstalē Windows datorā, kuru vēlaties uzraudzīt. Izpildiet Zabbix aģenta instalēšanas norādījumus Windows.



Konfigurēt Zabbix uzraudzībai

Zabbix aģents var vākt metriku aktīvajā vai pasīvajā režīmā (vienlaikus).

Note:
Pasīvā pārbaude ir vienkāršs datu pieprasījums. Zabbix serveris vai starpniekserveris pieprasa noteiktus datus (piemēram, CPU noslodzi), un Zabbix aģents nosūta rezultātu atpakaļ serverim. Aktīvās pārbaudes prasa sarežģītāku apstrādi. Aģentam vispirms no servera(-iem) ir jāiegūst vienumu saraksts neatkarīgai apstrādei, un pēc tam dati jānosūta atpakaļ lielapjomā. 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 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 aktīvajām vai pasīvajām Zabbix aģenta pārbaudēm.

Pasīvie pārbaudes

Zabbix lietotāja saskarne

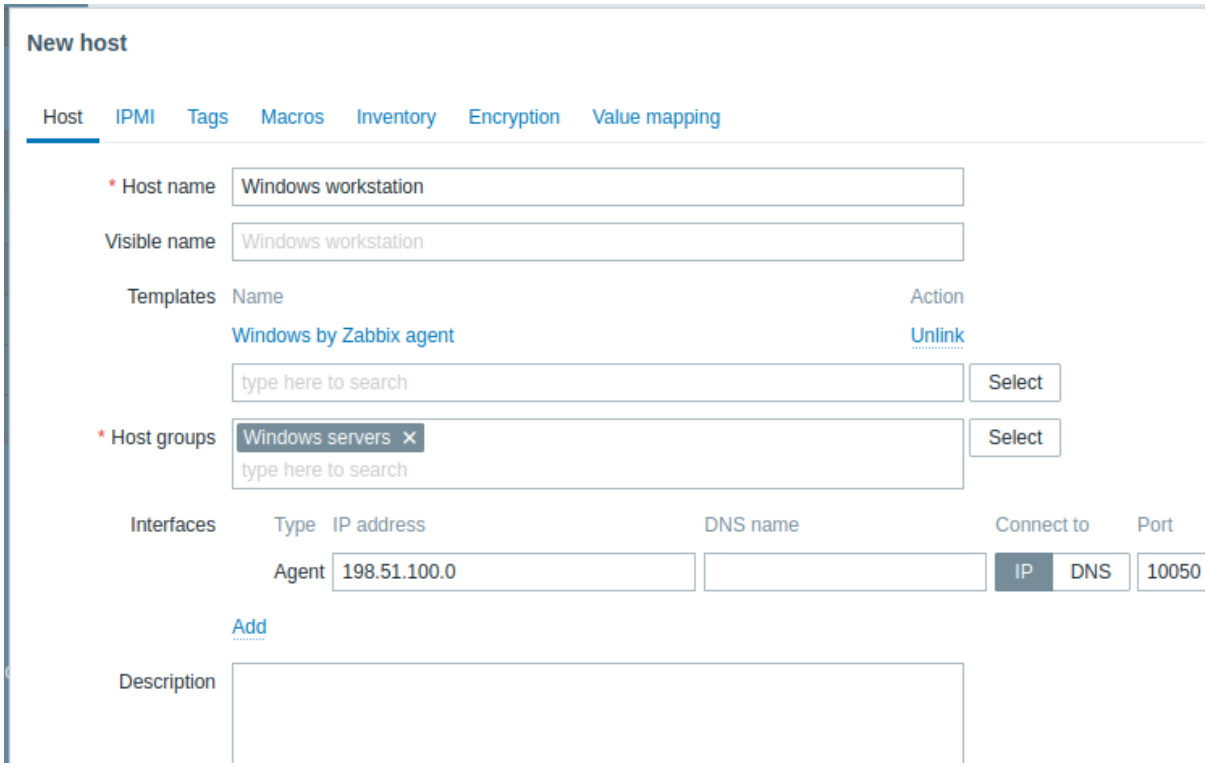
1. Piesakieties Zabbix lietotāja saskarnē.

2. Izveidojiet hostu Zabbix lietotāja saskarnē.

Šis hosts attēlos jūsu Windows datoru.

3. Parametrā *Interfaces* pievienojiet *Agent* interfeisu un norādiet Windows datora, kurā ir instalēts aģents, IP adresi vai DNS nosaukumu.

4. Parametrā *Templates* ievadiet 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 norādījāt pareizo informāciju, konfigurācijas fails jau ir atjaunināts. Pretējā gadījumā tā ir jānorāda manuāli. Atveriet aģenta konfigurācijas failu un parametram `Server` pievienojiet sava Zabbix servera IP adresi vai DNS nosaukumu.

Piemērs:

```
Server=192.0.2.22
```

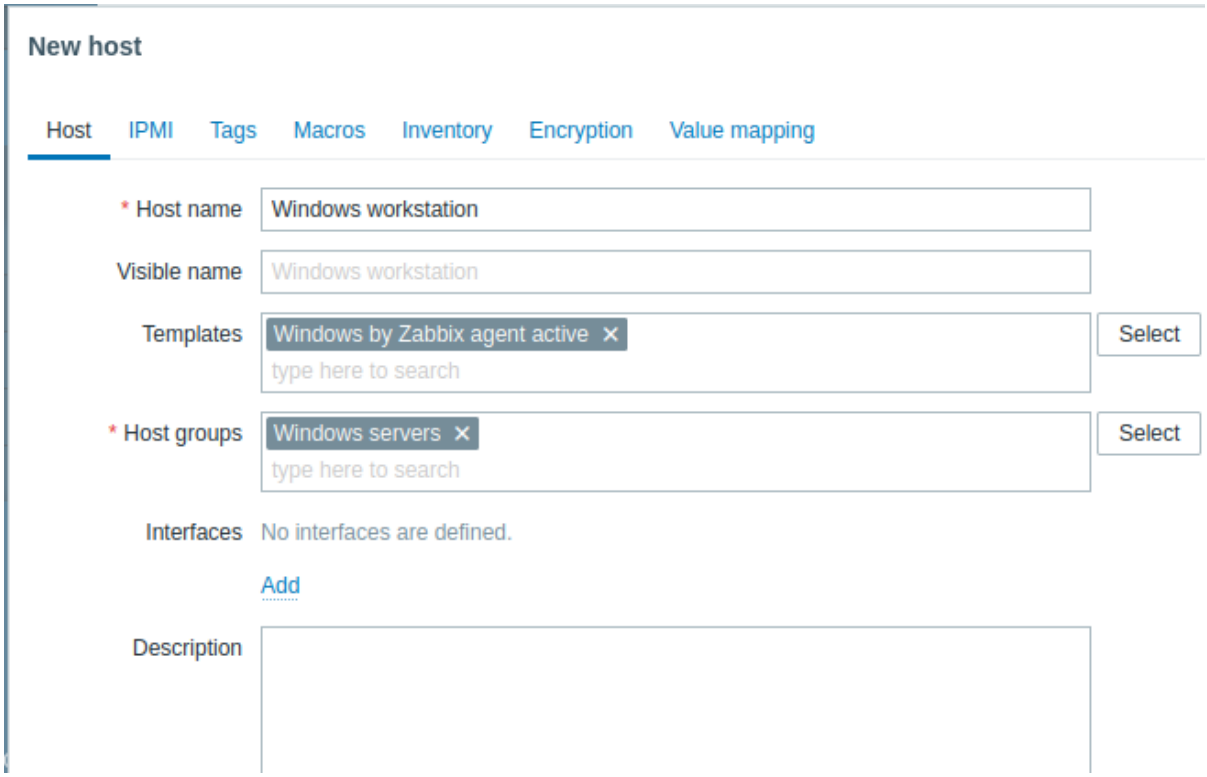
Aktīvās pārbaudes

Zabbix lietotāja saskarne

1. Piesakieties Zabbix lietotāja saskarnē.
2. Izveidojiet hostu Zabbix lietotāja saskarnē.

Šis hosts attēlos jūsu Windows datoru.

3. Parametrā *Templates* ievadiet vai atlasiet *Windows by Zabbix agent active*.



Zabbix aģents

Mapē `C:\Program files\Zabbix Agent` atveriet failu `zabbix_agentd.conf` un pievienojiet:

- Zabbix lietotāja saskarnē izveidotā hosta nosaukumu parametram *Hostname*.
- Jūsu Zabbix servera iekārtas IP adresi vai DNS nosaukumu parametram *ServerActive* (tas var būt jau aizpildīts, ja to norādījāt Zabbix aģenta iestatīšanas laikā).

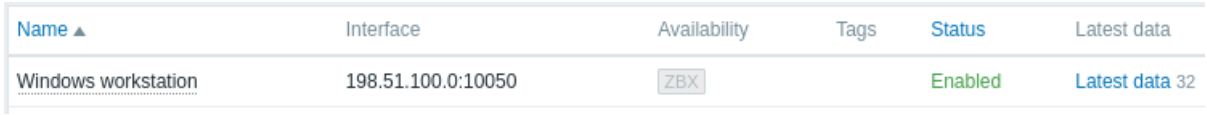
Piemērs:

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

Skatīt savāktos metriskos datus

Apsveicam! Šajā brīdī Zabbix jau uzrauga jūsu Windows datoru.

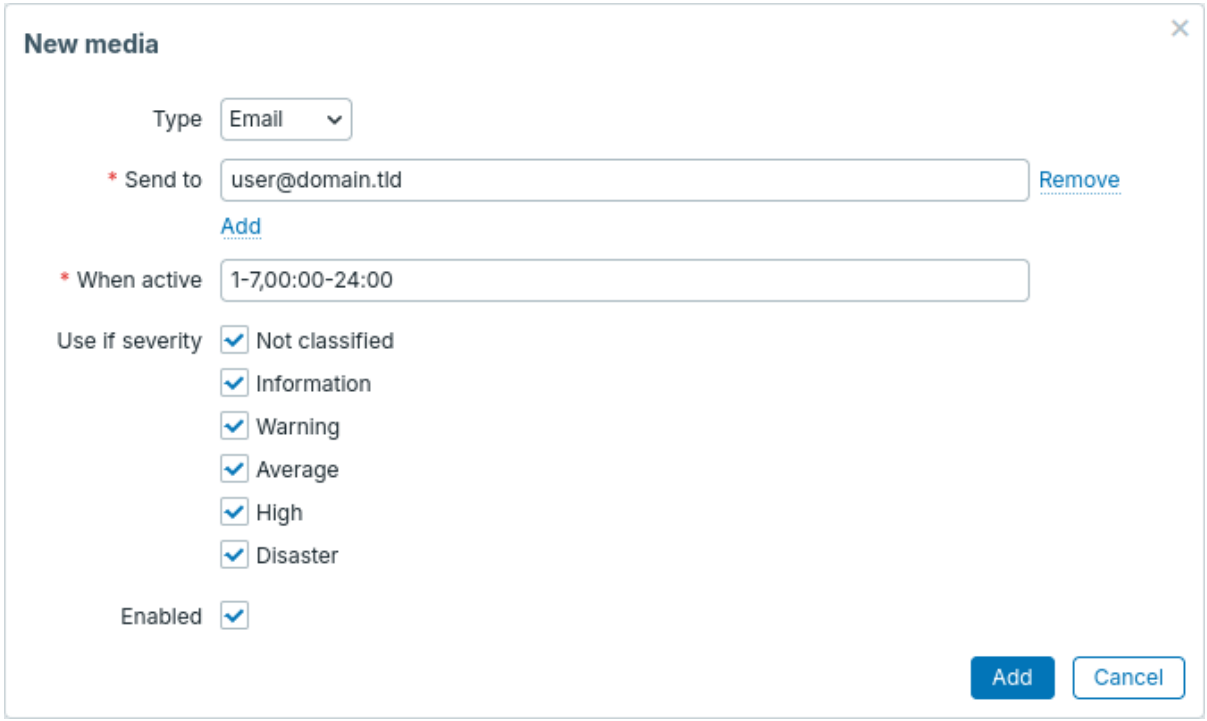
Lai skatītu savāktos metriskos datus, atveriet *Monitoring* > *Hosts* izvēlnes sadaļu un noklikšķiniet uz *Latest data* blakus hostam.



Iestatiet problēmu brīdinājumus

Zabbix var paziņot jums par problēmu jūsu infrastruktūrā, izmantojot dažādas metodes. Šajā ceļvedī ir sniegti konfigurācijas soļi e-pasta brīdinājumu nosūtīšanai.

1. Dodieties uz *Lietotāja iestatījumi* > *Profils*, pārslēdzieties uz cilni *Multivide* un pievienojiet savu e-pastu.



2. Izpildiet ceļvedi par problēmas paziņojuma saņemšanu.

Nākamajā reizē, kad Zabbix noteiks problēmu, jums vajadzētu saņemt brīdinājumu pa e-pastu.

Note:
Operētājsistēmā Windows varat izmantot CpuStres utilītu, lai simulētu lielu CPU slodzi un tā rezultātā saņemtu problēmas brīdinājumu.

Skatīt 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 vairāku soļu brīdinājumu scenārijus (piem., vispirms nosūtīt ziņojumu sistēmas administratoram, pēc tam, ja problēma netiek atrisināta 45 minūšu laikā, nosūtīt ziņojumu datu centra vadītājam).

Pārraugiet Apache, izmantojot HTTP

Ievads

Šī lapa parāda ātru un vienkāršu veidu, kā sākt uzraudzīt Apache tīmekļa serveri, neinstalējot nekādu papildu programmatūru.

Kam šī rokasgrāmata ir paredzēta

Šī rokasgrāmata ir paredzēta jauniem Zabbix lietotājiem un satur minimālo darbību kopu, kas nepieciešama, lai iespējotu Apache instalācijas pamata uzraudzību. Ja meklējat plašākas pielāgošanas iespējas vai nepieciešama sarežģītāka konfigurācija, skatiet Zabbix rokasgrāmatas sadaļu Konfigurācija.

Priekšnosacījumi

Pirms turpināt ar šo instalēšanas rokasgrāmatu, jums ir jālejupielādē un jāinstalē Zabbix serveris un Zabbix lietotāja saskarne saskaņā ar jūsu operētājsistēmas instrukcijām.

Sagatavojiet Apache

1. Pārbaudiet, kuru Apache versiju izmantojat:

RHEL bāzētā sistēmā palaidiet:

```
httpd -v
```

Debian/Ubuntu sistēmā palaidiet:

```
apache2 -v
```

2. Pārliecinieties, ka Status module ir iespējots jūsu Apache instancē.

RHEL bāzētā sistēmā palaidiet:

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

Debian/Ubuntu sistēmā palaidiet:

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

Ja sarakstā neredzat status_module, iespējojiet moduli, palaižot:

RHEL bāzētā sistēmā palaidiet:

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

Debian/Ubuntu sistēmā palaidiet:

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

3. Rediģējiet Apache konfigurācijas failu, lai atļautu piekļuvi statusa pārskatiem 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: `/etc/apache2/mods-enabled/status.conf`:

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

Pievienojiet failam šādas rindas (**aizstājiet 198.51.100.255** ar savu Zabbix servera IP adresi):

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

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

4. Restartējiet Apache

RHEL bāzētā sistēmā palaidiet:

```
sudo systemctl restart httpd
```

Debian/Ubuntu sistēmā palaidiet:

```
sudo systemctl restart apache2
```

5. Lai pārbaudītu, vai viss ir konfigurēts pareizi, palaidiet (**aizstājiet 198.51.100.255** ar savu Zabbix servera IP adresi):

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

Atbildei jāietver Apache tīmekļa servera statistika.

Konfigurējiet Zabbix uzraudzībai

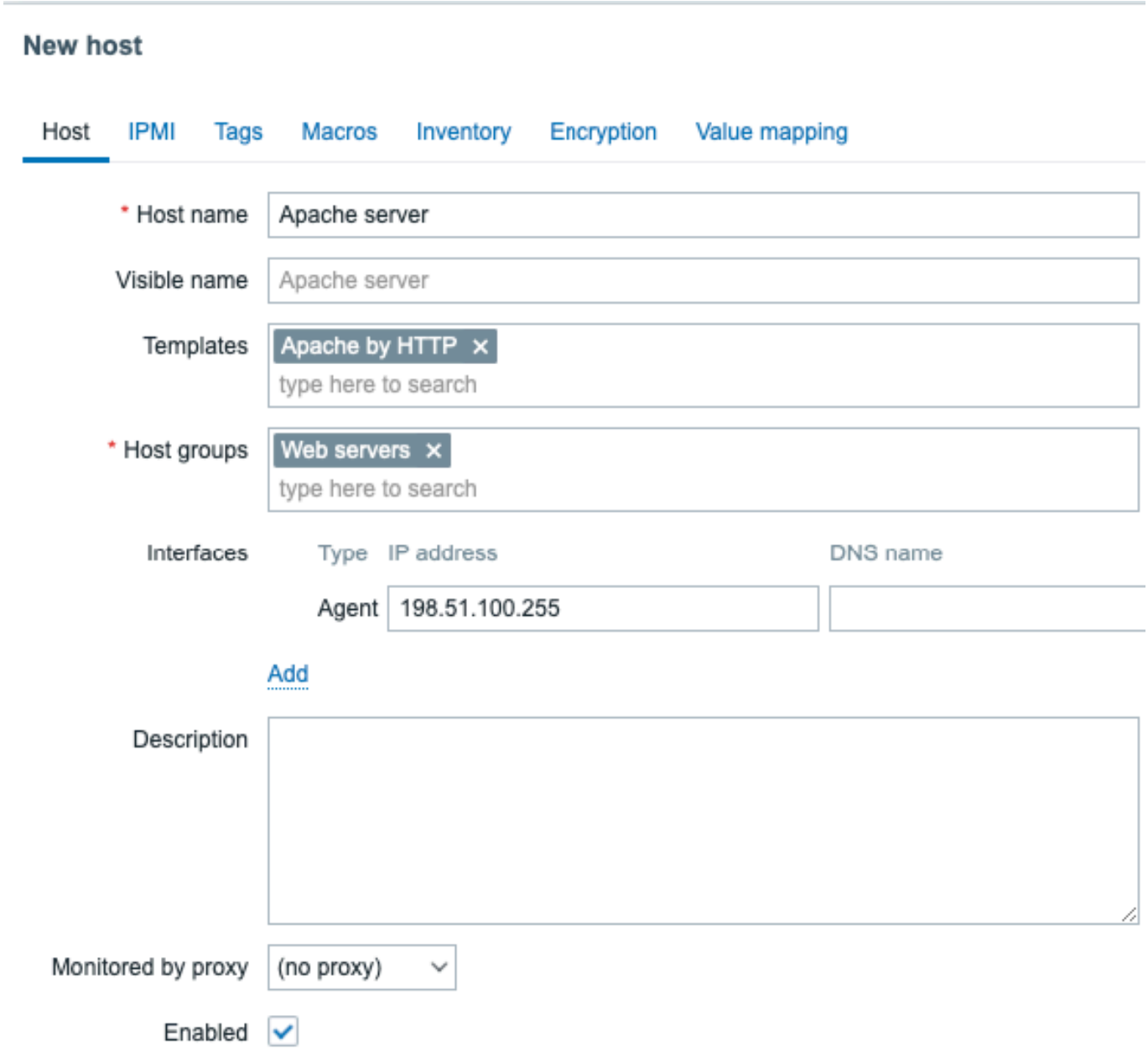
1. Piesakieties Zabbix lietotāja saskarnē.

2. Izveidojiet hostu Zabbix tīmekļa saskarnē.

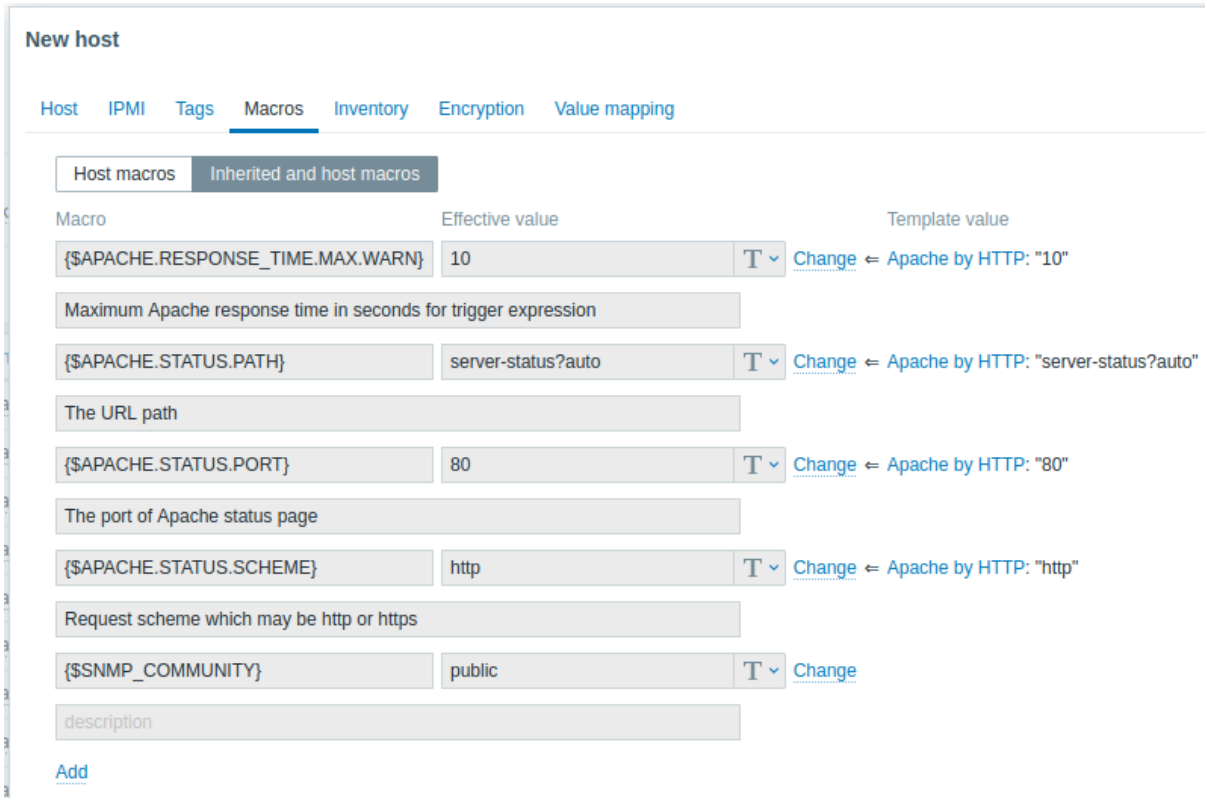
Šis hosts attēlos jūsu Apache serveri.

3. Parametrā *Interfaces* pievienojiet *Agent* saskarni un norādiet sava Apache instances IP adresi.
Zabbix aģents šajā iekārtā nav jāinstalē, saskarne tiks izmantota tikai {HOST.CONN} makro atrisināšanai.
Šis makro tiek izmantots veidnes vienumos, lai atrastu Apache instanci.

4. Parametrā *Templates* ievadiet vai atlasiet *Apache by HTTP*.



5. Pārslēdzieties uz cilni **Macros** un atlasiet režīmu *Inherited and host macros*. Pārbaudiet, vai makro {$APACHE.STATUS.PORT} un {$APACHE.STATUS.SCHEME} vērtības atbilst jūsu instalācijas iestatījumiem. Pēc noklusējuma ports ir 80 un shēma ir http. Mainiet makro vērtības, ja izmantojat citu portu un/vai shēmu.



Skatīt savāktos metriskos datus

Apsveicam! Šajā brīdī Zabbix jau uzrauga jūsu Apache tīmekļa serveri.

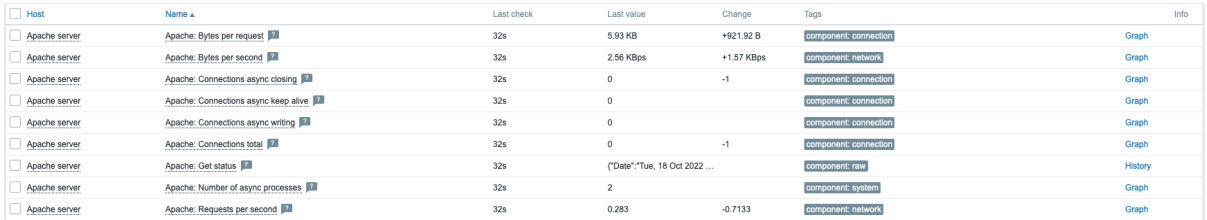
Lai skatītu savāktos metriskos datus, atveriet *Monitoring* > *Hosts* izvēlnes sadaļu un noklikšķiniet uz *Dashboards* blakus hostam.



Šī darbība jūs aizvedīs uz hosta informācijas paneli ar svarīgākajiem metriskajiem datiem, kas savākti no Apache /server-status lapas.



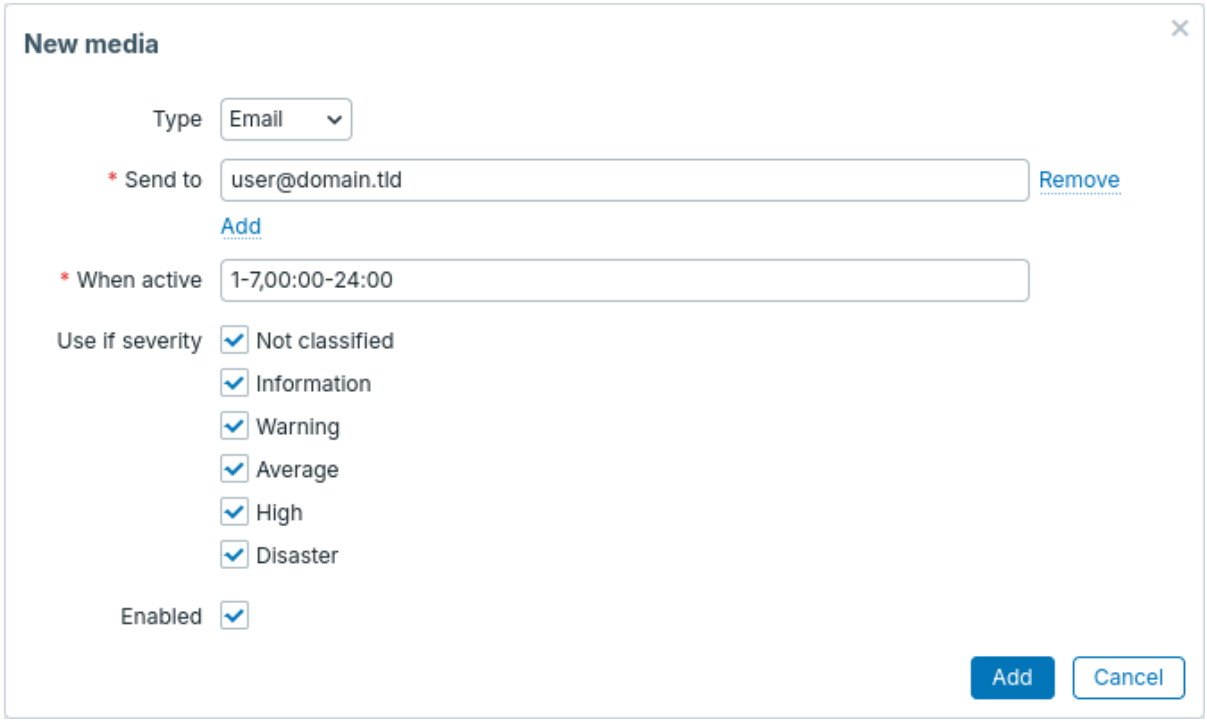
Alternatīvi, sadaļā *Monitoring* > *Hosts* varat noklikšķināt uz *Latest data*, lai skatītu visus jaunākos savāktos metriskos datus sarakstā.



Iestatiet problēmu brīdinājumus

Zabbix var paziņot jums par problēmu jūsu infrastruktūrā, izmantojot dažādas metodes.
Šajā rokasgrāmatā ir aprakstītas konfigurācijas darbības e-pasta brīdinājumu nosūtīšanai.

1. Dodieties uz *Lietotāja iestatījumi* > *Profils*, pārslēdzieties uz cilni *Media* un pievienojiet savu e-pastu.



2. Izpildiet rokasgrāmatu par problēmu paziņojumu saņemšanu.

Nākamreiz, kad Zabbix noteiks problēmu, jums vajadzētu saņemt brīdinājumu pa e-pastu.

Pārbaudiet savu konfigurāciju

Lai simulētu reālu problēmu un saņemtu testa problēmas brīdinājumu:

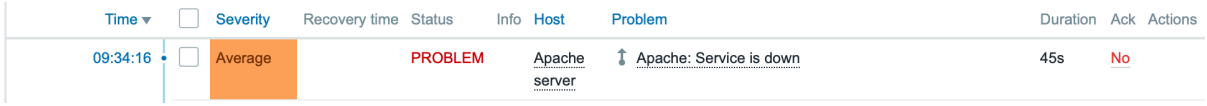
1. Atveriet *Apache server* host konfigurāciju Zabbix.

2. Pārslēdzieties uz cilni Macros un atlasiet *Inherited and host macros*.

3. Noklikšķiniet uz *Change* blakus makro {$APACHE.STATUS.PORT} un iestatiet citu portu.

4. Noklikšķiniet uz *Update*, lai saglabātu host konfigurāciju.

5. Pēc dažām minūtēm Zabbix noteiks problēmu *Apache service is down*, jo tagad tas nevar izveidot savienojumu ar instanci.
Tā parādīsies sadaļā *Monitoring* > *Problems*.



Ja brīdinājumi ir konfigurēti, jūs saņemsiet arī paziņojumu par problēmu.

6. Mainiet makro vērtību atpakaļ, lai novērstu problēmu un turpinātu Apache uzraudzību.

Skatīt arī:

- Tīmekļa servera nostiprināšana - ieteicamie iestatījumi lielākai tīmekļa servera drošībai.
- Vienuma izveide - kā sākt papildu metriku uzraudzību.
- HTTP vienumi - kā uzraudzīt pielāgotas metrikas, izmantojot HTTP aģentu.
- Problēmu eskalācijas - kā izveidot vairāku soļu brīdinājumu scenārijus (piem., vispirms nosūtīt ziņojumu sistēmas administratoram, pēc tam, ja problēma netiek atrisināta 45 minūšu laikā, nosūtīt ziņojumu datu centra vadītājam).

Pārraugiet MySQL ar Zabbix aģents 2

Ievads

Šajā lapā ir aprakstītas darbības, kas nepieciešamas, lai sāktu MySQL servera pamata uzraudzību.

Lai uzraudzītu MySQL serveri, ir vairāki pieejas veidi: Zabbix aģents, Zabbix aģents 2 vai Open Database Connectivity (ODBC) standarts. Šīs rokasgrāmatas galvenā uzmanība ir pievērsta MySQL servera uzraudzībai ar Zabbix aģentu 2, kas ir **ieteicamā** pieeja, jo tā nodrošina vienkāršu konfigurāciju dažādās vidēs. Tomēr šajā lapā ir sniegti arī norādījumi par citām pieejām, tāpēc varat izvēlēties to, kas vislabāk atbilst jūsu prasībām.

Kam šī rokasgrāmata ir paredzēta

Šī rokasgrāmata ir paredzēta jauniem Zabbix lietotājiem un satur minimālo darbību kopu, kas nepieciešama, lai iespējotu MySQL servera pamata uzraudzību. Ja meklējat plašākas pielāgošanas iespējas vai nepieciešama sarežģītāka konfigurācija, skatiet Zabbix rokasgrāmatas sadaļu Configuration.

Priekšnosacījumi

Pirms turpināt šo rokasgrāmatu, jums ir jālejupielādē un jāinstalē Zabbix serveris, Zabbix lietotāja saskarne un Zabbix aģents 2 saskaņā ar jūsu operētājsistēmas norādījumiem.

Atkarībā no jūsu vides daži šīs rokasgrāmatas soļi var nedaudz atšķirties. Šī rokasgrāmata ir balstīta uz vidi, kurā darbojas Ubuntu.

Izveidot 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, instalējot Zabbix), tomēr šim lietotājam ir vairāk privilēģiju, nekā nepieciešams vienkāršai uzraudzībai (privilēģijas DROP datubāzu, DELETE ierakstu no tabulām u. c. darbībām). Tāpēc MySQL lietotājs, kas paredzēts *tikai* MySQL servera uzraudzībai, ir jāizveido.

1. Pieslēdzieties MySQL klientam, izveidojiet lietotāju "zbx_monitor" (aizstājiet <*password*> lietotājam "zbx_monitor" ar sev vēlamu paroli), un GRANT piešķiriet lietotājam nepieciešamās 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;
```

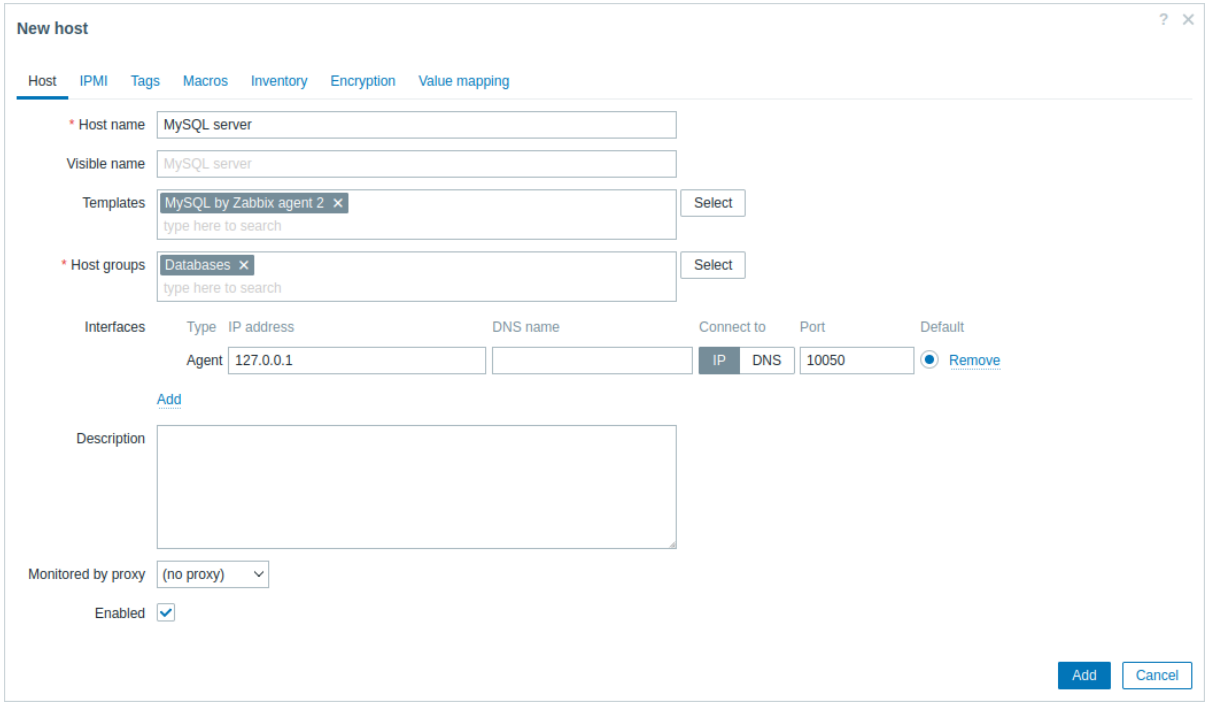
Kad lietotājs ir izveidots, varat pāriet uz nākamo soli.

Zabbix lietotāja saskarnes konfigurēšana

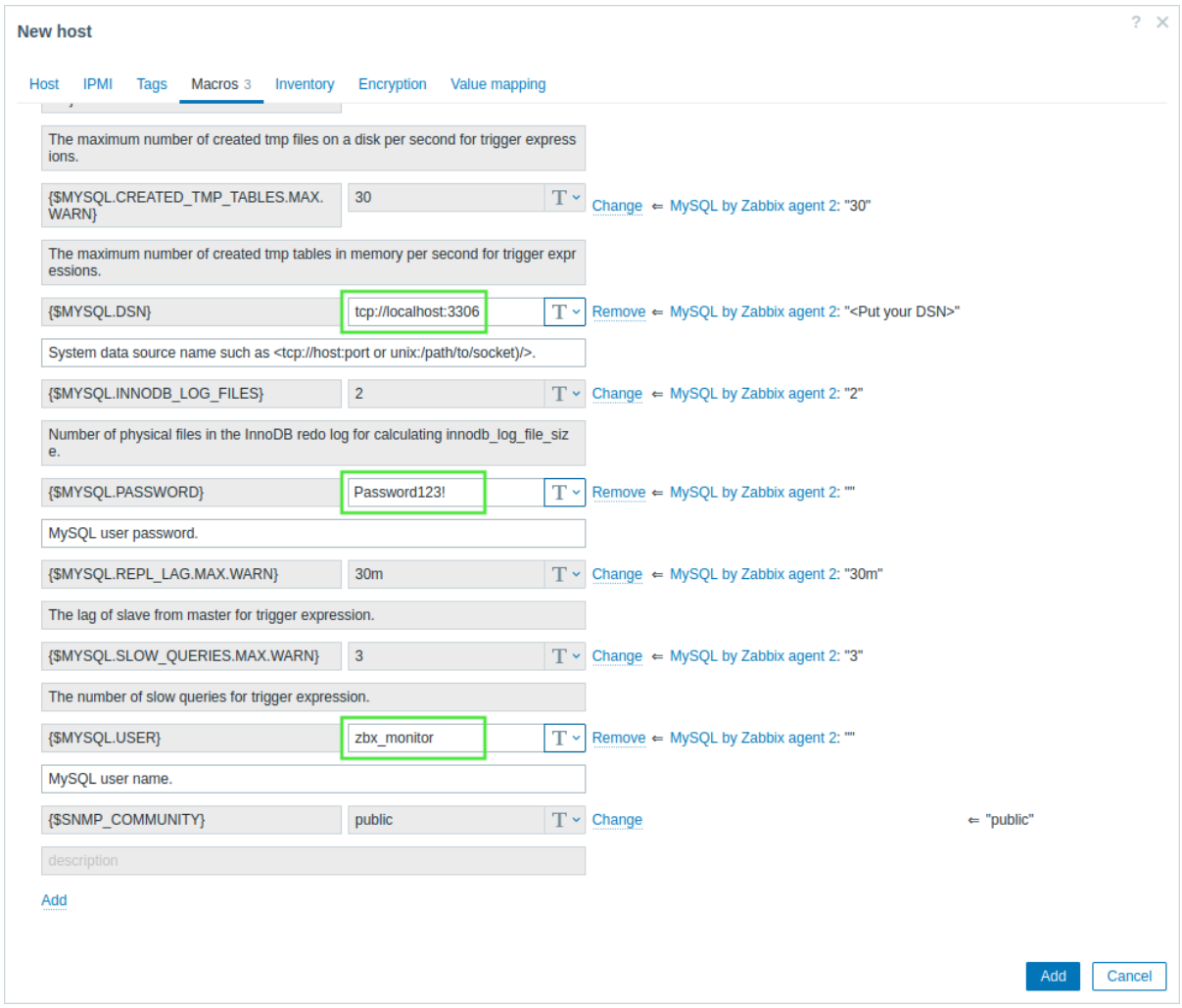
1. Piesakieties Zabbix lietotāja saskarnē.

2. Izveidojiet hostu Zabbix tīmekļa saskarnē:

- Laukā *Host name* ievadiet hosta nosaukumu (piem., "MySQL server").
- Laukā *Templates* ierakstiet vai atlasiet veidni "MySQL by Zabbix agent 2", kas tiks saistīta ar hostu.
- Laukā *Host groups* ierakstiet vai atlasiet hostu grupu (piem., "Databases").
- Laukā *Interfaces* pievienojiet saskarni ar tipu "Agent" un norādiet sava MySQL servera IP adresi. Šajā rokasgrāmatā 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 tālāk norādītos makro un noklikšķiniet uz *Change* blakus makro vērtībai, lai to atjauninātu:
  - {$MYSQL.DSN} - iestatiet MySQL servera datu avotu (MySQL Zabbix aģents 2 spraudņa konfigurācijas failā norādīto nosauktās sesijas savienojuma virkni). Šajā rokasgrāmatā tiek izmantots noklusējuma datu avots "tcp://localhost:3306", lai uzraudzītu MySQL serveri, kas ir instalēts tajā pašā datorā, kur Zabbix serveris un Zabbix aģents 2.
  - {$MYSQL.PASSWORD} - iestatiet iepriekš izveidotā MySQL lietotāja "zbx_monitor" paroli.
  - {$MYSQL.USER} - iestatiet iepriekš izveidotā MySQL lietotāja "zbx_monitor" lietotājvārdu.

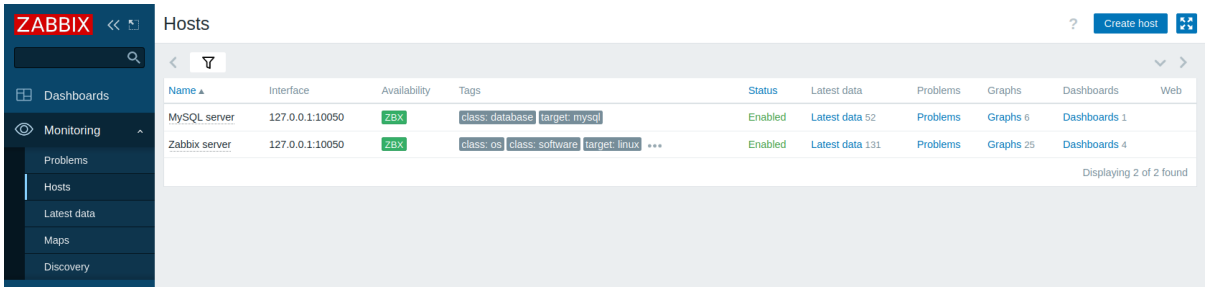


3. Noklikšķiniet uz *Add*, lai pievienotu hostu. Šis hosts attēlos jūsu MySQL serveri.

Skatīt savāktos metriskos datus

Apsveicam! Šajā brīdī Zabbix jau uzrauga jūsu MySQL serveri.

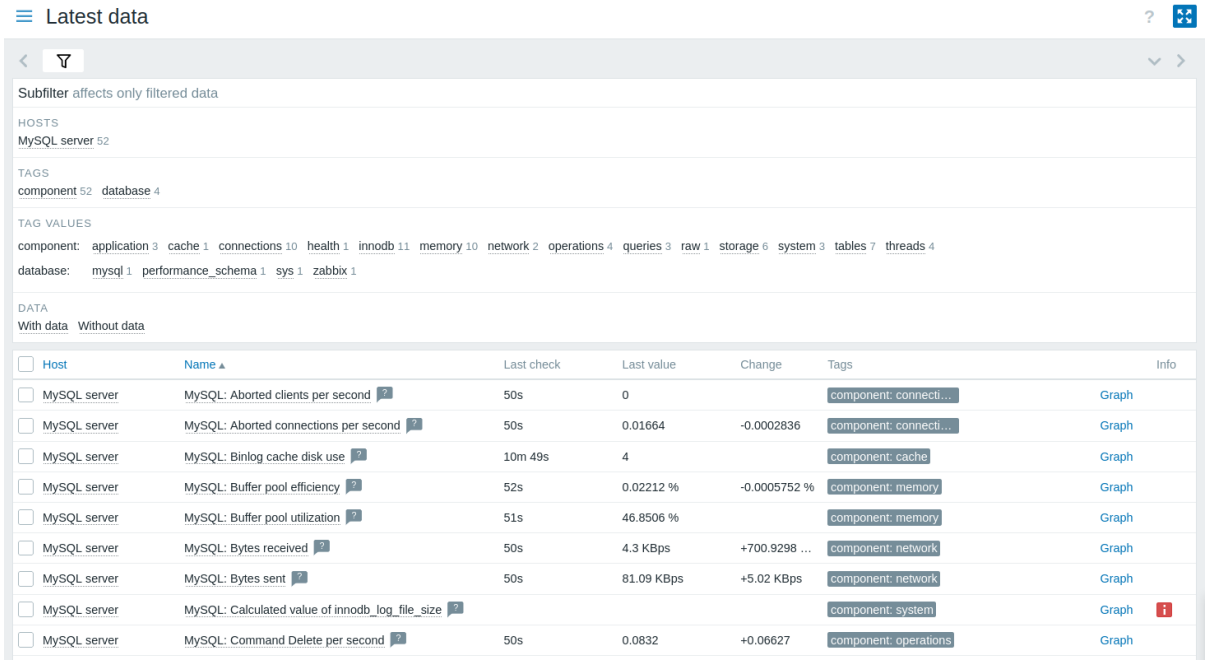
Lai skatītu savāktos metriskos datus, dodieties uz izvēlnes sadaļu *Monitoring > Hosts* un noklikšķiniet uz *Dashboards* blakus hostam.



Šī darbība jūs aizvedīs uz hosta informācijas paneli (konfigurētu veidnes līmenī) ar svarīgākajiem metriskajiem datiem, kas savākti no MySQL servera.



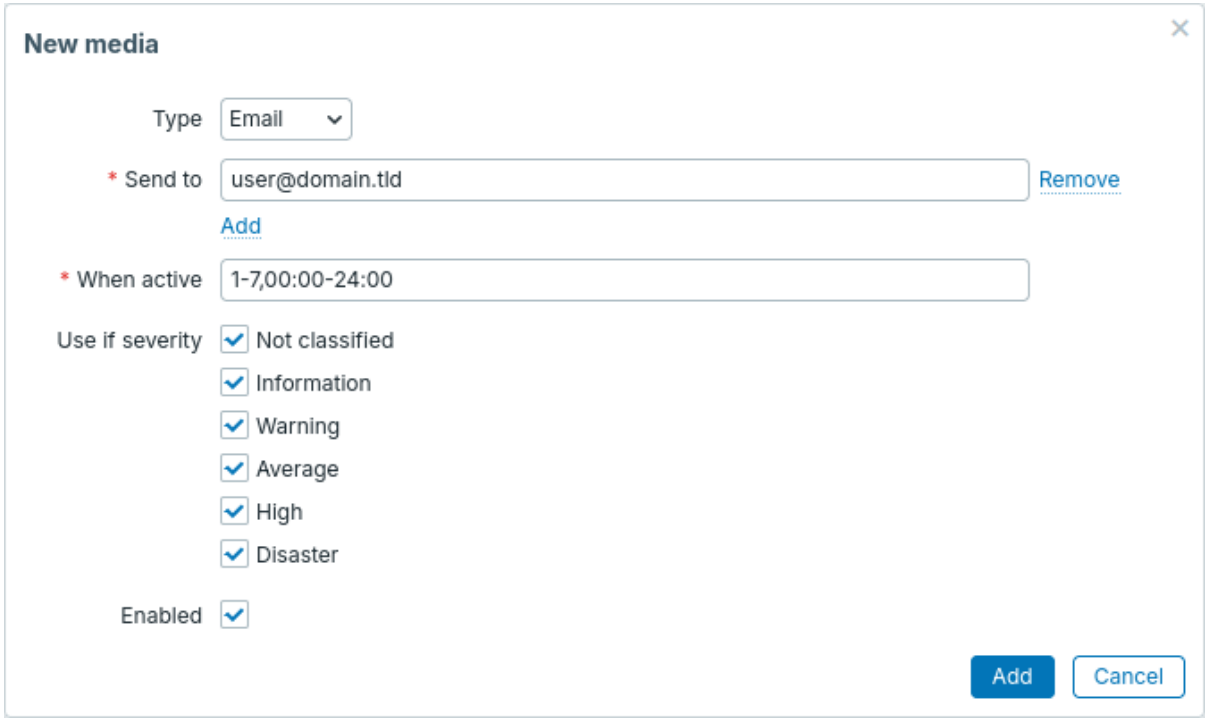
Alternatīvi, izvēlnes sadaļā *Monitoring > Hosts* varat noklikšķināt uz *Latest data*, lai skatītu visus jaunākos savāktos metriskos datus sarakstā. Ņemiet vērā, ka vienumam *MySQL: Calculated value of innodb_log_file_size* nav paredzēts datu, jo vērtība tiks aprēķināta, pamatojoties uz pēdējās stundas datiem.



Iestatiet problēmu brīdinājumus

Zabbix var paziņot jums par problēmu jūsu infrastruktūrā, izmantojot dažādas metodes. Šajā rokasgrāmatā ir sniegti pamata konfigurācijas soļi e-pasta brīdinājumu nosūtīšanai.

1. Dodieties uz *Lietotāja iestatījumi > Profils*, pārslēdzieties uz cilni *Media* un pievienojiet savu e-pastu.



2. Izpildiet rokasgrāmatu par problēmas paziņojuma saņemšanu.

Nākamreiz, kad Zabbix noteiks problēmu, jums vajadzētu saņemt brīdinājumu pa e-pastu.

Pārbaudiet savu konfigurāciju

Lai pārbaudītu savu konfigurāciju, mēs varam simulēt reālu problēmu, atjauninot host konfigurāciju Zabbix lietotāja saskarnē.

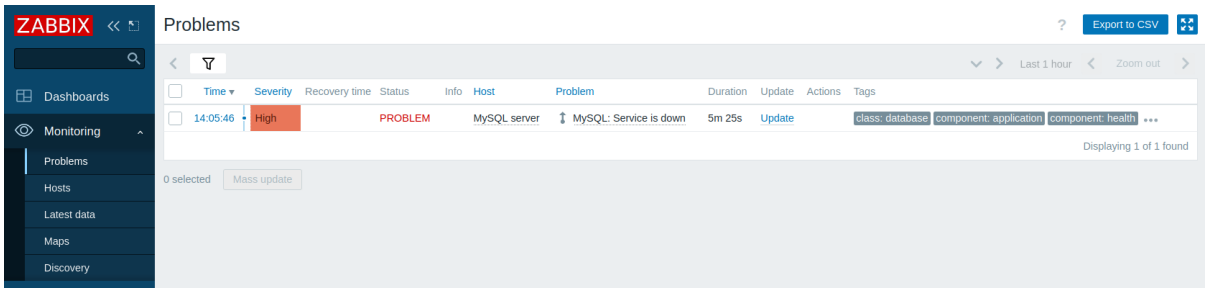
1. Atveriet sava MySQL servera host konfigurāciju Zabbix.

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 host 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 paziņojumu par problēmu.

6. Mainiet makro vērtību atpakaļ uz iepriekšējo vērtību, lai novērstu problēmu un turpinātu MySQL servera uzraudzību.

Citi MySQL uzraudzības veidi

MySQL servera uzraudzībai Zabbix aģenta 2 vietā varat izmantot arī Zabbix aģentu vai Open Database Connectivity (ODBC) standartu. Lai gan ieteicams izmantot Zabbix aģentu 2, var būt konfigurācijas, kas neatbalsta Zabbix aģentu 2 vai kurām nepieciešama pielāgota pieeja.

Galvenā atšķirība starp Zabbix aģentu un ODBC ir datu vākšanas metode - Zabbix aģents tiek instalēts tieši MySQL serverī un vāc datus, izmantojot iebūvēto funkcionalitāti, savukārt ODBC izmanto ODBC draiveri, lai izveidotu savienojumu ar MySQL serveri un izgūtu datus, izmantojot SQL vaicājumus.

Lai gan daudzas konfigurācijas darbības ir līdzīgas MySQL servera uzraudzībai ar Zabbix aģentu 2, pastāv arī dažas būtiskas atšķirības - lai varētu uzraudzīt MySQL serveri, jums jākonfigurē Zabbix aģents vai ODBC. Tālāk sniegtie norādījumi palīdzēs jums iepazīties ar šīm **atšķirībām**.

Pārraugiet MySQL ar Zabbix aģentu

Lai pārraudzītu MySQL serveri ar Zabbix aģentu, jums ir jā lejupielādē un jāinstalē Zabbix serveris, Zabbix lietotāja saskarne un Zabbix aģents atbilstoši jūsu OS norādījumiem.

Kad nepieciešamie Zabbix komponenti ir veiksmīgi instalēti, jums ir jāizveido MySQL lietotājs, kā aprakstīts sadaļā *Izveidot MySQL lietotāju*.

Pēc MySQL lietotāja izveides jums ir jākonfigurē Zabbix aģents, lai tas varētu izveidot savienojumu ar MySQL serveri un to pārraudzīt. Tas ietver vairāku lietotāja parametru konfigurēšanu pielāgotu aģenta pārbaužu izpildei, kā arī Zabbix aģentam nepieciešamo akreditācijas datu nodrošināšanu, lai izveidotu savienojumu ar MySQL serveri kā iepriekš izveidotais lietotājs ”zbx_monitor”.

Konfigurēt Zabbix aģentu

1. Pārejiet uz Zabbix aģenta papildu konfigurāciju direktoriju.

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

> **Attention:**
> Zabbix aģenta papildu konfigurāciju direktorijam jāatrodas tajā pašā direktorijā, kur atrodas jūsu Zabbix aģenta konfigurācijas fails; noklusējuma ceļš ir atkarīgs no operētājsistēmas un Zabbix instalēšanas metodes.

Tā vietā, lai Zabbix aģenta konfigurācijas failā definētu visus nepieciešamos lietotāja parametrus MySQL servera uzraudzībai, 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 `template_db_mysql.conf` faila saturu (fails atrodas Zabbix repozitorijā) uz jūsu izveidoto `template_db_mysql.conf` failu 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 aģenta mājas direktoriju (ja tas jūsu sistēmā nepastāv, tas ir jāizveido; noklusējums: */var/lib/zabbix*).

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

6. Izveidojiet *.my.cnf* failu Zabbix aģenta mājas direktorijā.

```
vi .my.cnf
```

7. Nokopējiet šādu saturu *.my.cnf* failā (aizstājiet <*password*> ar lietotāja ”zbx_monitor” paroli).

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

Konfigurējiet Zabbix lietotāja saskarni un pārbaudiet savu konfigurāciju

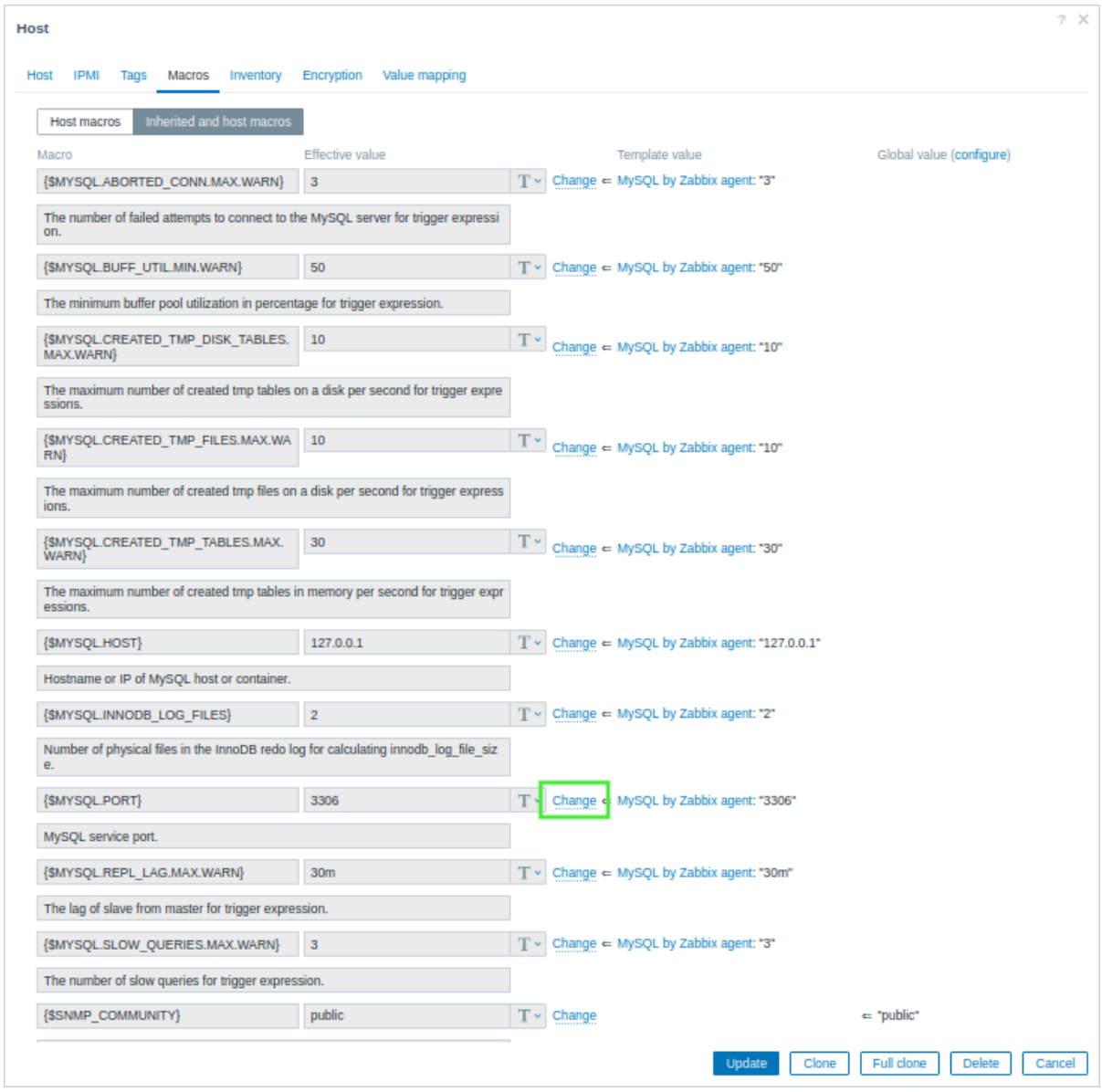
Lai konfigurētu Zabbix lietotāja saskarni, izpildiet norādījumus sadaļā *Konfigurējiet Zabbix lietotāja saskarni*, veicot šādas izmaiņas:

- Laukā *Templates* ievadiet vai atlasiet veidni ”MySQL by Zabbix agent”, kas tiks saistīta ar hostu.
- *Macros* konfigurēšana nav nepieciešama.

Kad Zabbix lietotāja saskarne ir konfigurēta, varat skatīt savāktos metriskos datus un iestatīt problēmu brīdinājumus.

Lai pārbaudītu savu konfigurāciju, izpildiet norādījumus sadaļā *Pārbaudiet savu konfigurāciju*, veicot šādas izmaiņas:

- MySQL servera hosta konfigurācijas sadaļā *Inherited and host macros* noklikšķiniet uz *Change* blakus {$MYSQL.PORT} makro vērtībai un iestatiet citu portu (piem., ”6033”).



Pārraugiet MySQL ar ODBC

Lai pārraudzītu 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 *Izveidot MySQL lietotāju*.

Pēc MySQL lietotāja izveides jums ir jāiestata 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 problēmu 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 vajadzētu būt 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*. Tālākajā *odbcinst.ini* faila piemērā draivera nosaukums ir "MariaDB Unicode".

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

5. Kopējiet tālāk norādīto saturu failā *odbc.ini* (aizstājiet *<password>* ar lietotāja "zbx_monitor" paroli). Šajā rokasgrāmatā kā MySQL servera adrese uzraudzībai tiek izmantota "127.0.0.1" (localhost) MySQL serverim, kas ir instalēts tajā pašā datorā, kur ODBC draiveris. Ņemiet vērā datu avota nosaukumu (DSN) "test", kas būs nepieciešams, kad konfigurēsiet Zabbix lietotāja saskarni.

```
[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 *Konfigurējiet Zabbix lietotāja saskarni*, veicot šādas izmaiņas:

- Laukā *Templates* ievadiet vai atlasiet veidni "MySQL by ODBC", kas tiks saistīta ar hostu.
- *Interfaces* konfigurēšana nav nepieciešama.
- {$MYSQL.DSN} makro vērtība sadaļā *Inherited and host macros* MySQL servera hosta konfigurācijā jāiestata uz DSN nosaukumu no *odbc.ini* faila.

Kad esat konfigurējis Zabbix lietotāja saskarni, varat skatīt savāktos metriskos datus, iestatīt problēmu brīdinājumus un pārbaudīt savu konfigurāciju.

Skatīt arī

- Vienuma izveide - kā sākt uzraudzīt papildu metriku.
- Problēmu eskalācijas - kā izveidot vairāku soļu brīdinājumu scenārijus (piem., vispirms nosūtīt ziņojumu sistēmas administratoram, pēc tam, ja problēma netiek atrisināta 45 minūšu laikā, nosūtīt ziņojumu datu centra vadītājam).
- ODBC uzraudzība - kā iestatīt ODBC citos Linux distributīvos un kā sākt uzraudzīt papildu ar datubāzi saistītu metriku ar ODBC.
- Veidne *MySQL by Zabbix aģents* - papildu informācija par veidni *MySQL by Zabbix aģents*.
- Veidne *MySQL by Zabbix aģents 2* - papildu informācija par veidni *MySQL by Zabbix aģents 2*.
- Veidne *MySQL by ODBC* - papildu informācija par veidni *MySQL by ODBC*.

Monitorējiet VMware 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 satur minimālo darbību kopu, kas nepieciešama, lai iespējotu VMware pamata uzraudzību. Ja meklējat plašas pielāgošanas iespējas vai nepieciešama sarežģītāka konfigurācija, skatiet Zabbix rokasgrāmatas sadaļu Virtuālo mašīnu uzraudzība vai sadaļu Konfigurācija.

Priekšnosacījumi

Pirms turpināt šo rokasgrāmatu, jums ir jālejupielādē un jāinstalē Zabbix serveris un lietotāja saskarne saskaņā ar jūsu operētājsistēmas norādījumiem.

Atkarībā no jūsu vides daži soļi šajā rokasgrāmatā var nedaudz atšķirties. Šī rokasgrāmata ir balstīta uz vidi, kurā darbojas Ubuntu.

Tiek pieņemts, ka VMware jau ir konfigurēts. Šī rokasgrāmata neaptver VMware konfigurēšanu.

Zabbix servera konfigurēšana

Lai pārraudzītu VMware, ir jāiespējo Zabbix *vmware collector* procesi. Plašāku informāciju par VMware pārraudzības darbību skatiet sadaļā Virtuālo mašīnu pārraudzība.

1. Atveriet Zabbix servera konfigurācijas failu.

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

2. Zabbix servera konfigurācijas failā atrodiet un iestatiet parametru `StartVMwareCollectors` uz 2 vai lielāku vērtību (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
```

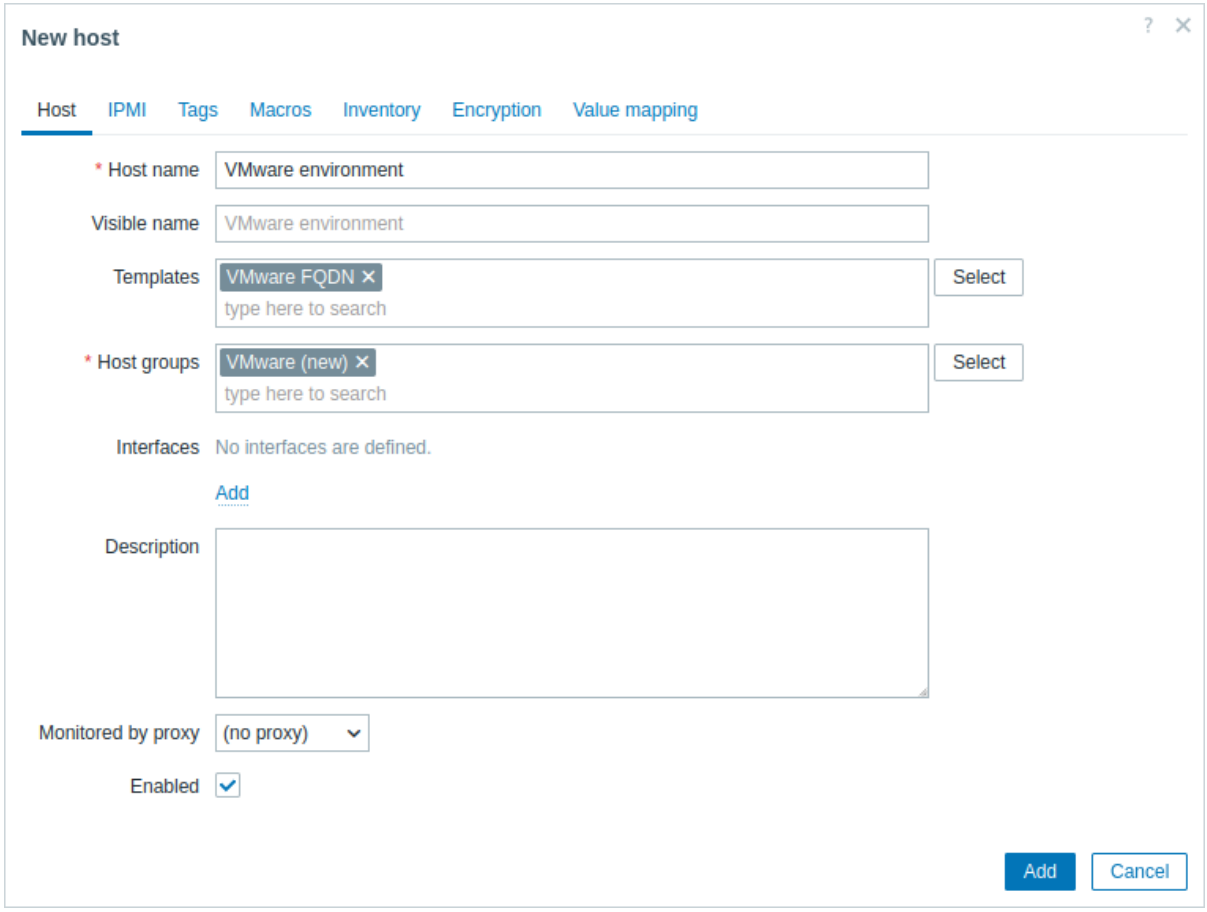
Kad *vmware collector* procesi ir palaisti, pārejiet pie nākamās darbības.

Konfigurēt Zabbix lietotāja saskarni

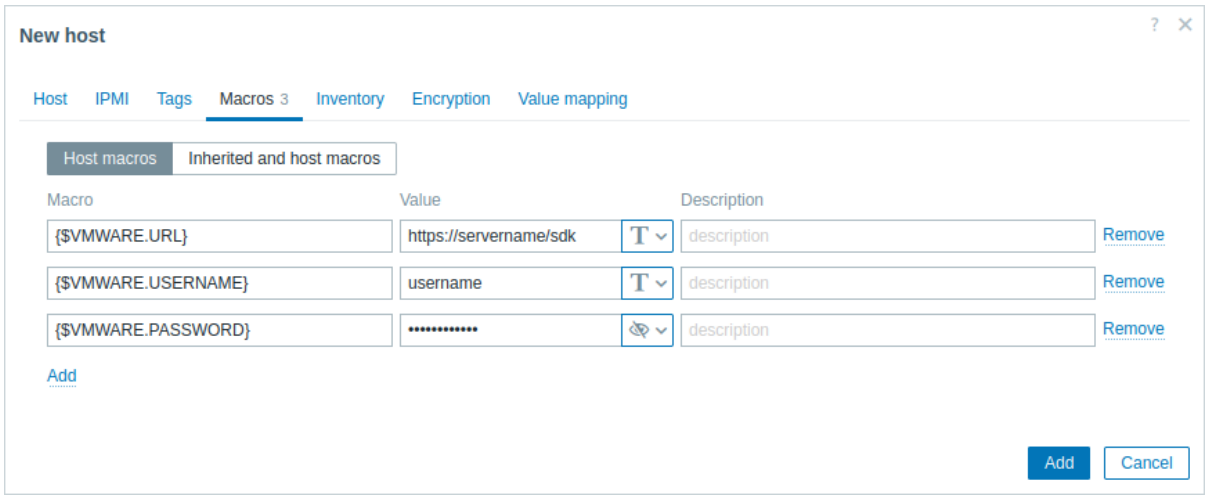
1. Piesakieties Zabbix lietotāja saskarnē.

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”). Plašāku informāciju par šīm veidnēm skatiet sadaļā Virtual machine monitoring.
- Laukā *Host groups* ierakstiet vai atlasiet hostu grupu (piemēram, jaunu hostu grupu ”VMware”).



- Cilnē *Macros* iestatiet šādus hosta makro:
  - {$VMWARE.URL} - VMware pakalpojuma (vCenter vai ESXi hipervizora) SDK URL (https://servername/sdk)
  - {$VMWARE.USERNAME} - VMware pakalpojuma lietotājvārds
  - {$VMWARE.PASSWORD} - VMware pakalpojuma {$VMWARE.USERNAME} lietotāja parole



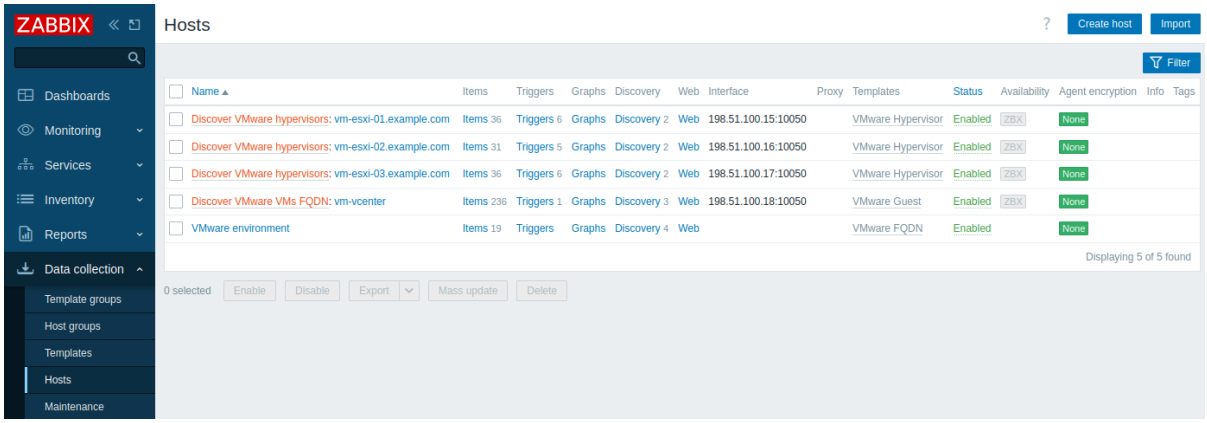
3. Noklikšķiniet uz pogas *Add*, lai izveidotu hostu. Šis hosts attēlos jūsu VMware vidi.

Skatīt savāktos metriskos datus

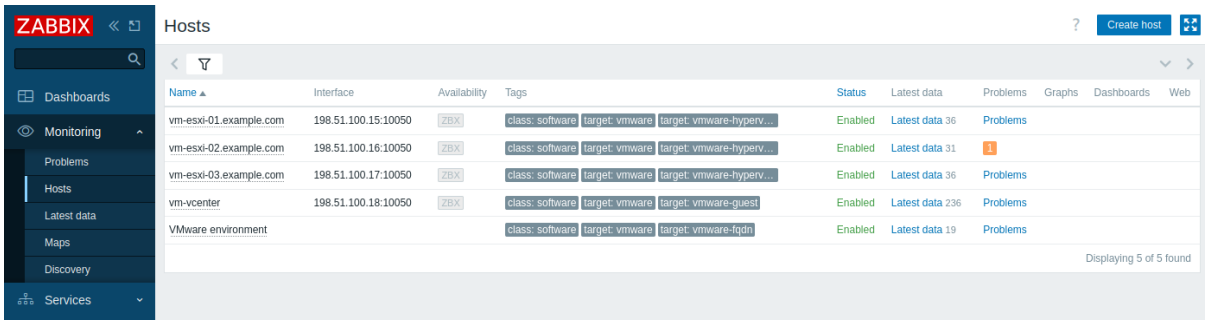
Apsveicam! Šajā brīdī Zabbix jau uzrauga jūsu VMware vidi.

Atkarībā no jūsu VMware vides konfigurācijas Zabbix var atklāt un pēc tam izveidot hosts atklātajiem objektiem. Ņemiet vērā, ka hosts atklāšanu un izveidi vajadzības gadījumā var arī izpildīt manuāli.

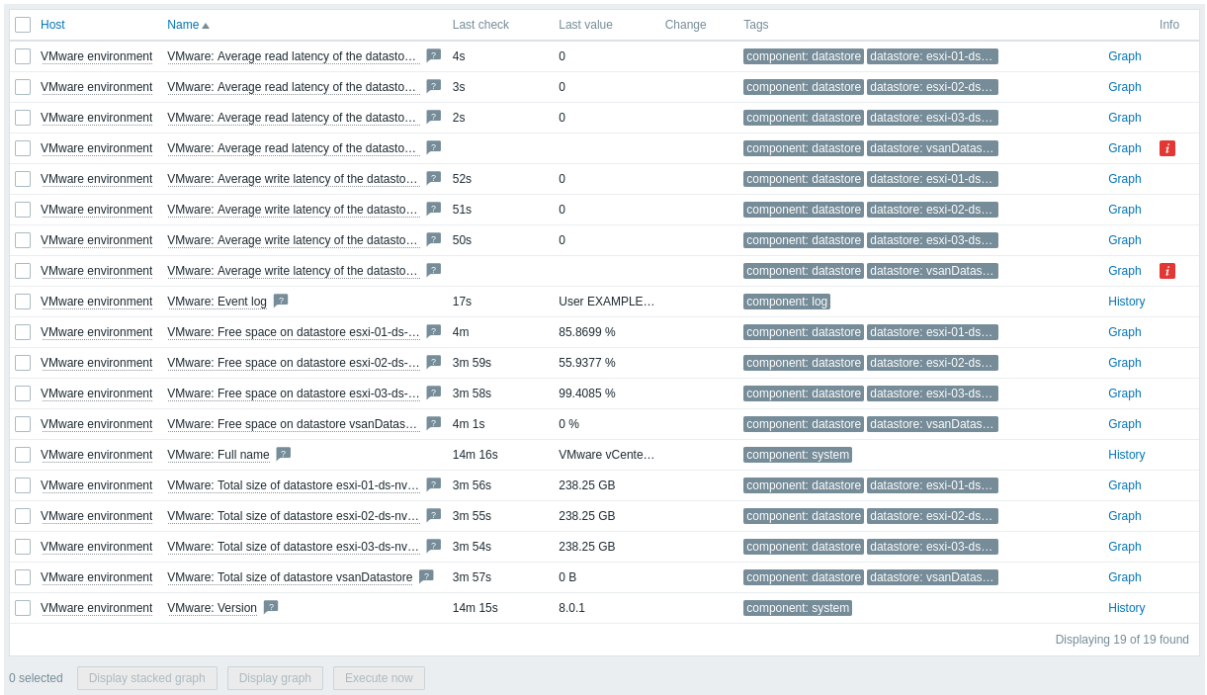
Lai skatītu izveidotos hosts, dodieties uz izvēlnes sadaļu *Datu vākšana > Hosts*.



Lai skatītu savāktos metriskos datus, dodieties uz izvēlnes sadaļu *Uzraudzība > Hosts* un noklikšķiniet uz *Jaunākie dati* blakus izveidotajam "VMware environment" hostam vai vienam no hostiem, kas tika izveidoti atklātajiem objektiem.



Šī darbība atvērs visu jaunāko metrisko datu sarakstu, kas savākti no atlasītā hosta.

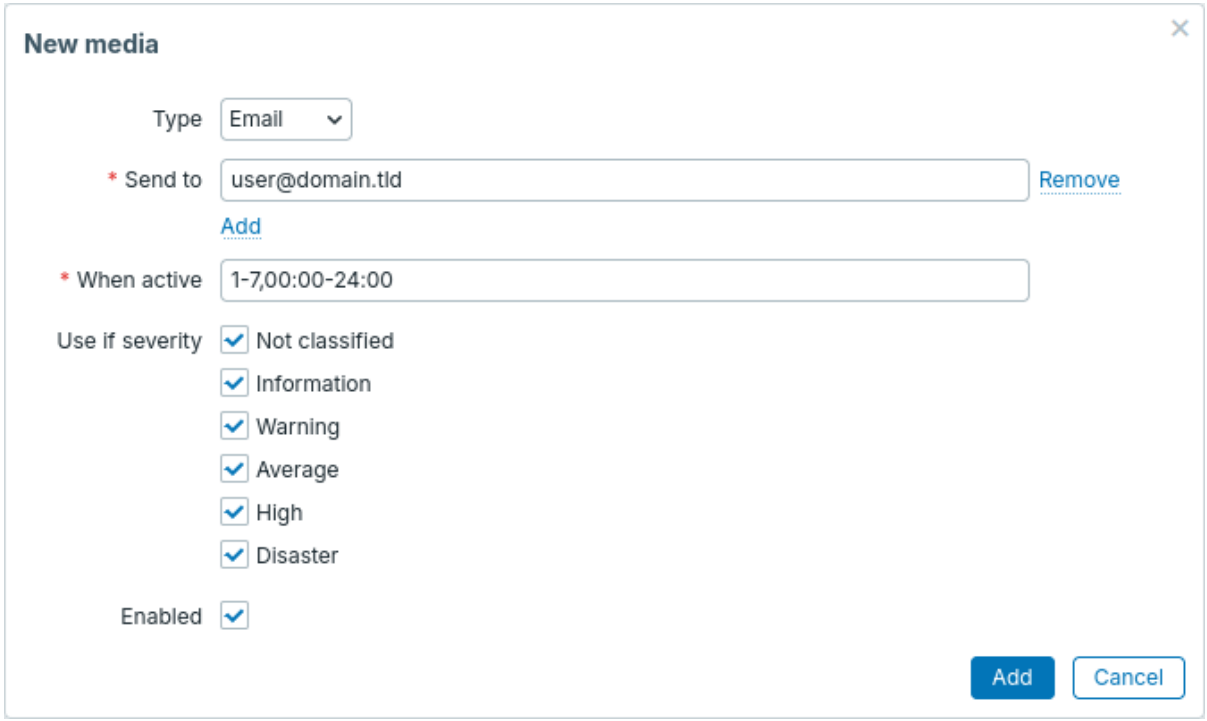


Ņemiet vērā, ka dažiem vienumiem nav datu un tiem ir stāvoklis *Not supported*. Tas ir tāpēc, ka Zabbix nevar atrast derīgus veiktspējas skaitītājus konkrētajā datu krātuvē, jo tie nav iespējoti uzraugāmajā VMware vidē.

Iestatīt problēmu brīdinājumus

Zabbix var paziņot jums par problēmu jūsu infrastruktūrā, izmantojot dažādas metodes. Šajā rokasgrāmatā ir sniegti pamata konfigurācijas soļi e-pasta brīdinājumu nosūtīšanai.

1. Dodieties uz *Lietotāja iestatījumi > Profils*, pārslēdzieties uz cilni *Media* un pievienojiet savu e-pasta adresi.



2. Izpildiet rokasgrāmatu par problēmas paziņojuma saņemšanu.

Nākamreiz, kad Zabbix noteiks problēmu, jums vajadzētu saņemt brīdinājumu pa e-pastu.

Skatīt arī

- Vienuma izveide - kā sākt uzraudzīt papildu metriku.
- Problēmu eskalācijas - kā izveidot vairāku soļu brīdinājumu scenārijus (piem., 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 utt.).
- VMware uzraudzības vienuma atslēgas - pilns VMware metriku saraksts, ko var uzraudzīt, izmantojot Zabbix.
- Veidne *VMware* - papildu informācija par *VMware* veidni.
- Veidne *VMware FQDN* - papildu informācija par *VMware FQDN* veidni.

Pārraugiet tīkla trafiku ar Zabbix

Ievads

Šī lapa soli pa solim izskaidro darbības, kas nepieciešamas, lai sāktu pamata tīkla trafika uzraudzību ar Zabbix.

Kam šī rokasgrāmata ir paredzēta

Šī rokasgrāmata ir paredzēta jauniem Zabbix lietotājiem un satur minimālo darbību kopu, kas nepieciešama, lai iespējotu pamata jūsu tīkla datplūsmas uzraudzību. Ja meklējat plašas pielāgošanas iespējas vai nepieciešama sarežģītāka konfigurācija, skatiet Zabbix rokasgrāmatas sadaļu Konfigurācija.

Priekšnosacījumi

Pirms turpināt šo rokasgrāmatu, jums ir jāveic lejupielāde un instalēšana Zabbix serveris, lietotāja saskarne un aģents saskaņā ar jūsu operētājsistēmas norādījumiem.
Ņemiet vērā, ka Zabbix aģents jāinstalē tajā iekārtā, kurai nepieciešama tīkla trafika uzraudzība.
Tā var būt gan tā pati hosts, kurā ir instalēts Zabbix serveris, gan arī cita hosts.

Šajā rokasgrāmatā tiks sniegti norādījumi par tīkla trafika uzraudzības konfigurēšanu *eth0* saskarnei atsevišķā iekārtā ar nosaukumu *Remote host*.

Konfigurējiet Zabbix uzraudzībai

Zabbix aģents var vākt metriku aktīvajā vai pasīvajā režīmā (vienlaikus). Plašāku informāciju skatiet sadaļā Pasīvie un aktīvie aģenta pārbaudes. Šajā rokasgrāmatā tiks aprakstīta uzraudzība, izmantojot pasīvās pārbaudes.

Zabbix aģenta konfigurēšana

1. Atveriet aģenta konfigurācijas failu datorā, kurā ir instalēts aģents:

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

2. Parametram *Server* pievienojiet sava Zabbix servera IP adresi vai DNS nosaukumu. Piemēram:

```
Server=192.0.2.22
```

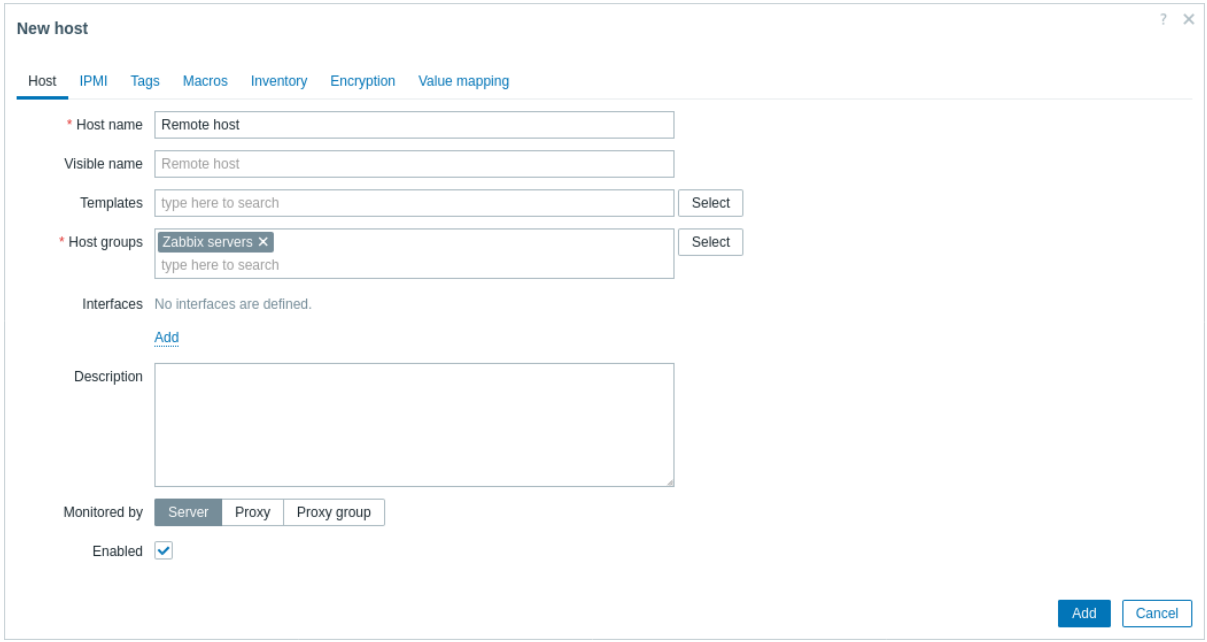
3. Restartējiet Zabbix aģentu:

```
systemctl restart zabbix-agent
```

Zabbix lietotāja saskarne

1. Piesakieties Zabbix lietotāja saskarnē.

2. Izveidojiet hostu Zabbix lietotāja saskarnē, norādot tās iekārtas IP adresi vai DNS nosaukumu, kurā ir instalēts aģents.

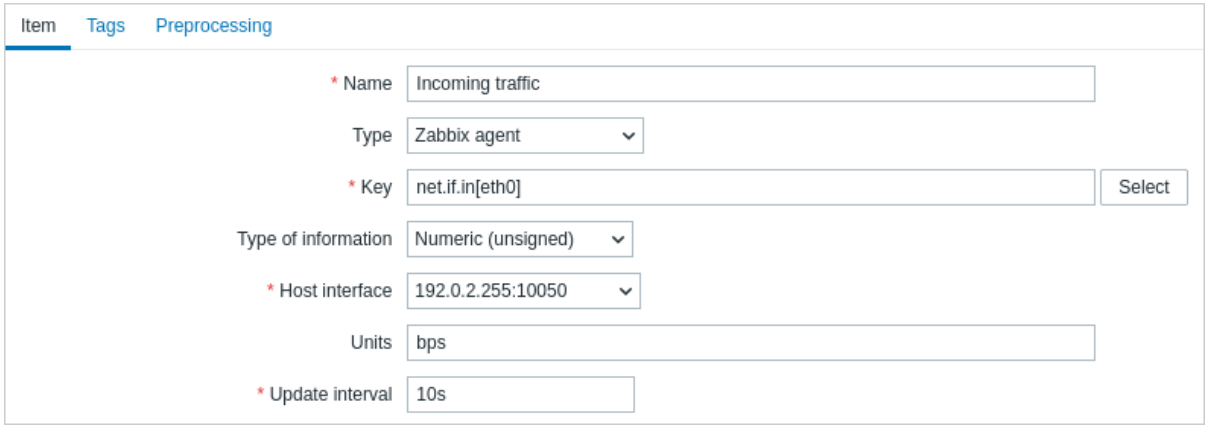


Izveidot vienumus

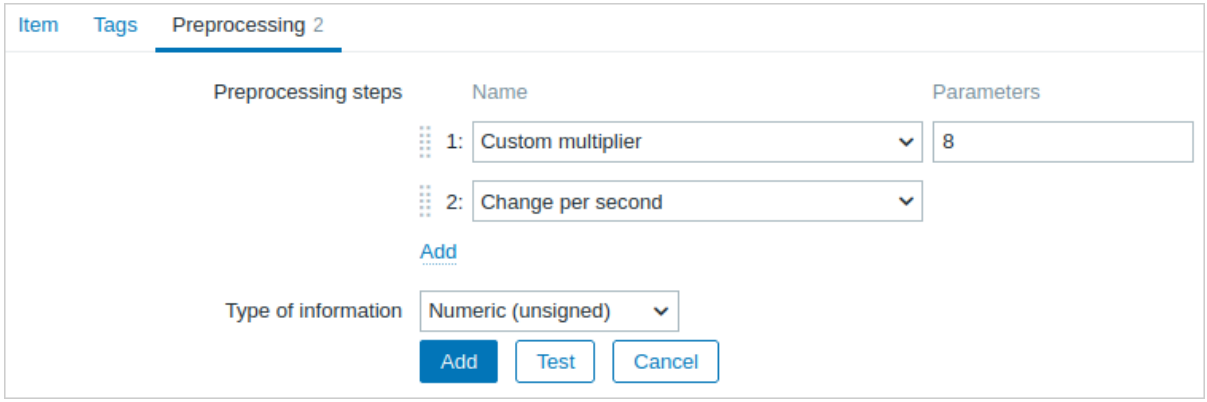
Izpildiet norādījumus par vienuma izveidi, lai pievienotu vienumus trafika uzraudzībai, proti:

- Ienākošais trafiks
- Izejošais trafiks
- Kopējais trafiks

Vienkāršs ienākošā trafika uzraudzības vienuma iestatījums izskatītos šādi:



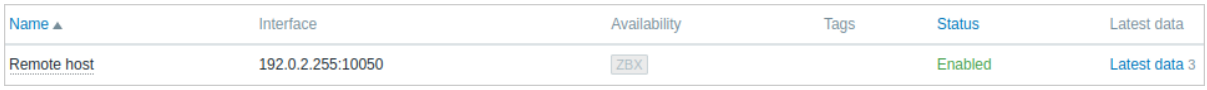
Lai savāktie dati būtu piemēroti praktiskai lietošanai, veidojot vienumus, varat iestatīt dažus priekšapstrādes soļus. Dotajā gadījumā tie var būt reizināšana ar 8 (lai pārvērstu baitus bitos) un attēlošana kā izmaiņas sekundē.



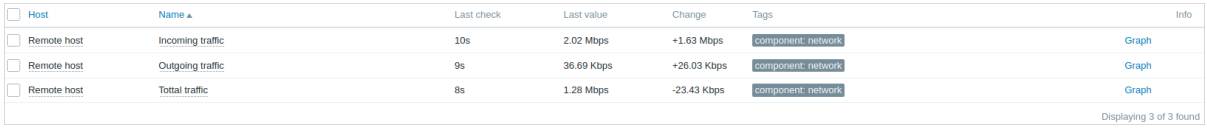
Skatīt apkopotos datus

Apsveicam! Šajā brīdī Zabbix jau uzrauga jūsu tīkla datplūsmu.

Lai skatītu apkopotos rādītājus, atveriet izvēlnes sadaļu *Uzraudzība > Hosts* un hosts rindā noklikšķiniet uz *Jaunākie dati*.

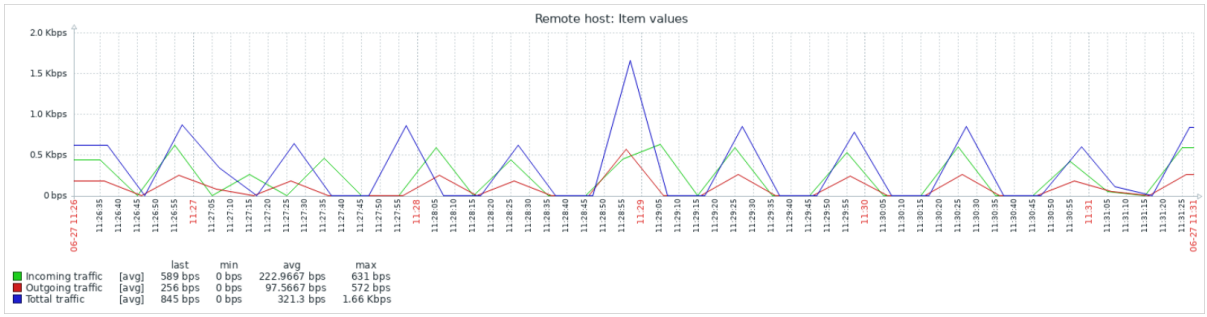


Tiks parādīti datplūsmas dati.



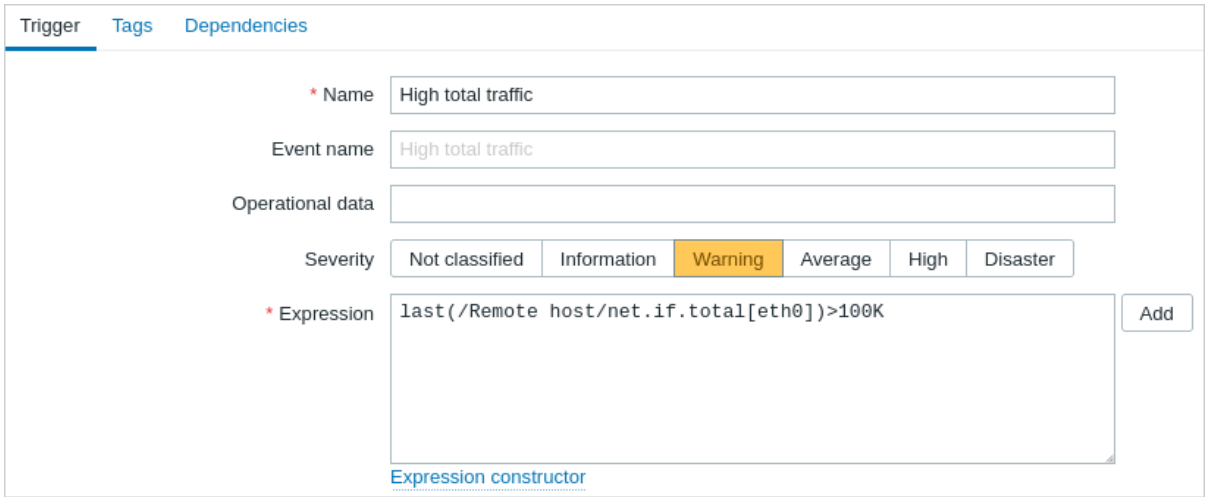
Skatīt grafikus

Savāktos datus var attēlot kā grafikus. Lai tos skatītu, sadaļā *Latest data* noklikšķiniet uz *Graph* vienuma rindā vai atlasiet nepieciešamos vienumus un noklikšķiniet uz *Display graph* zemāk.

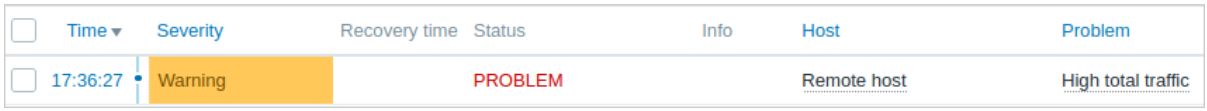


Konfigurēt trigerus

Varat iestatīt trigerus, lai noteiktu neparastu tīkla datplūsmu. Skatiet norādījumus par trigeru konfigurēšanu un pievienojiet trigeri, kas signalizē, ka kopējā datplūsma ir pārāk liela, piemēram:



Tagad palieliniet datplūsmu tā, lai tā pārsniegtu slieksni, ko esat iestatījis trigeru izteiksmē, un dodieties uz *Monitoring* > *Problems*, lai pārbaudītu, vai problēma tur ir uzskaitīta.



Iestatiet problēmu brīdinājumus

Ir vairāki veidi, kā saņemt paziņojumus par problēmu. E-pasts ir vispopulārākais no tiem, tāpēc sekojiet norādījumiem par problēmas paziņojuma iestatīšanu, izmantojot e-pastu. Varat arī izvēlēties citus mediju tipus, ko izmantot paziņojumu piegādei.

Skatīt arī:

- Problēmu eskalācijas - kā izveidot vairāku soļu brīdinājumu scenārijus (piem., 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 novēršanu, apspiest vai slēgt problēmu.
- Linux uzraudzība ar Zabbix aģentu - kā sākt pamata uzraudzību no svarīgākajiem vienumiem, piesaistot iepriekš konfigurētu veidni.

Tīkla trafika uzraudzība, izmantojot aktīvās pārbaudes

Ievads

Šī lapa soli pa solim izskaidro darbības, kas nepieciešamas, lai sāktu pamata tīkla trafika uzraudzību ar Zabbix, izmantojot aktīvās pārbaudes.

Kam šī rokasgrāmata ir paredzēta

Šī rokasgrāmata ir paredzēta jauniem Zabbix lietotājiem un satur minimālo darbību kopu, kas nepieciešama, lai iespējotu pamata tīkla trafika uzraudzību, izmantojot aktīvās pārbaudes. Ja meklējat plašas pielāgošanas iespējas vai nepieciešama sarežģītāka konfigurācija, skatiet Zabbix rokasgrāmatas sadaļu Configuration.

Priekšnosacījumi

Pirms turpināt šo rokasgrāmatu, jums ir jālejupielādē un jāinstalē Zabbix serveris, lietotāja saskarne un aģents saskaņā ar jūsu operētājsistēmas norādījumiem. Ņemiet vērā, ka aģents jāinstalē tajā iekārtā, kurai nepieciešama datplūsmas uzraudzība. Tā var būt gan tā pati iekārta, kurā ir instalēts Zabbix serveris, gan cita iekārta.

Šajā rokasgrāmatā tiks sniegti norādījumi par *eth0* interfeisa tīkla datplūsmas uzraudzības konfigurēšanu atsevišķā iekārtā ar nosaukumu *Remote host*.

Konfigurējiet Zabbix uzraudzībai

Zabbix aģents var vākt metriku aktīvajā vai pasīvajā režīmā (vienlaikus). Plašāku informāciju skatiet sadaļā Pasīvie un aktīvie aģenta pārbaužu veidi. Šajā rokasgrāmatā 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. Parametram ServerActive pievienojiet sava Zabbix servera IP adresi vai DNS nosaukumu (un neobligāto portu). Piemēram:

```
ServerActive=192.0.2.1:10051
```

Zabbix aģents izmantos šo adresi, lai izveidotu savienojumu ar Zabbix servera trappera portu (noklusējums: 10051) un pieprasītu aktīvo pārbaužu konfigurācijas datus.

3. Definējiet parametru Hostname, kuram jāsakrīt ar hosts nosaukumu, kas tiks definēts Zabbix lietotāja saskarnē. Mūsu piemērā tas ir:

```
Hostname=Remote host
```

Hostname vērtībai ir jāsakrīt, jo aktīvajām pārbaudēm Zabbix aģents to izmanto, lai no servera izgūtu pareizo hosts konfigurāciju. Konkrēti, aģents izveido savienojumu ar serveri un identificē sevi, izmantojot Hostname vērtību. Pēc tam serveris nodrošina šī hosts 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:

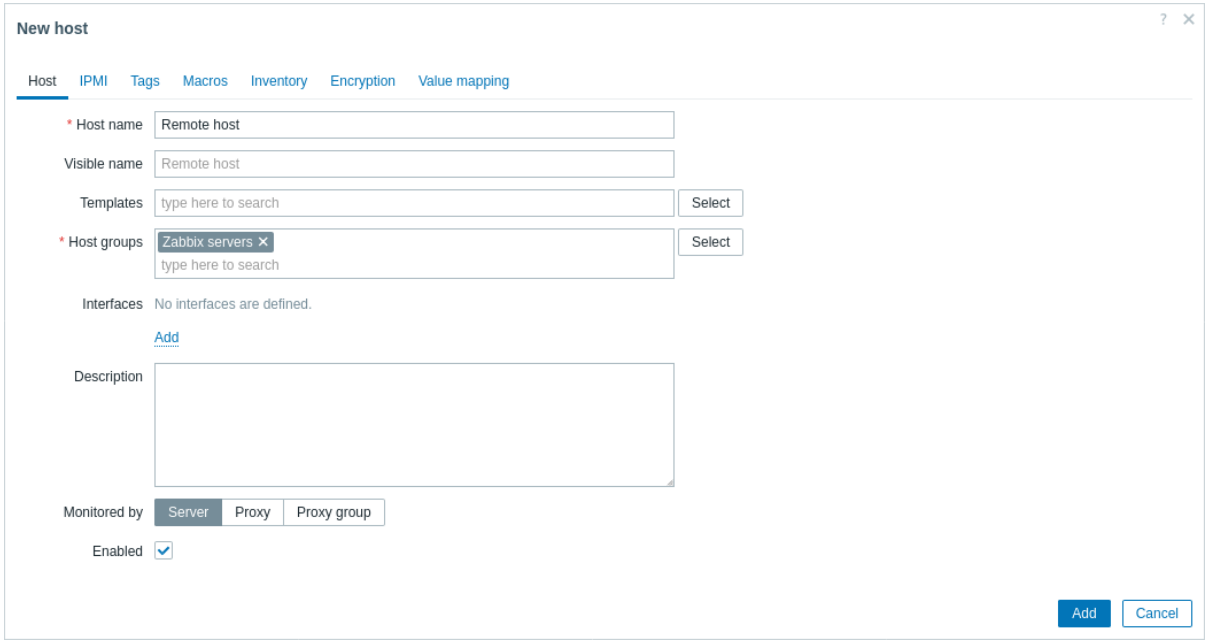
```
systemctl restart zabbix-agent2
```

Zabbix lietotāja saskarne

1. Piesakieties Zabbix lietotāja saskarnē.

2. Izveidojiet hostu Zabbix lietotāja saskarnē.

- Laukā *Host name* ievadiet hosta nosaukumu (piem., "Remote host"), kas atbilst iepriekš aģenta konfigurācijas failā definētajai Hostname parametra vērtībai.
- Laukā *Host groups* ievadiet vai atlasiet hostu grupu (piem., "Zabbix servers").
- Atstājiet *Interfaces* nedefinētu; interfeiss nav nepieciešams aktīvajām pārbaudēm, jo aģents uzsāk savienojumu ar serveri, nevis gaida savienojumu no servera.



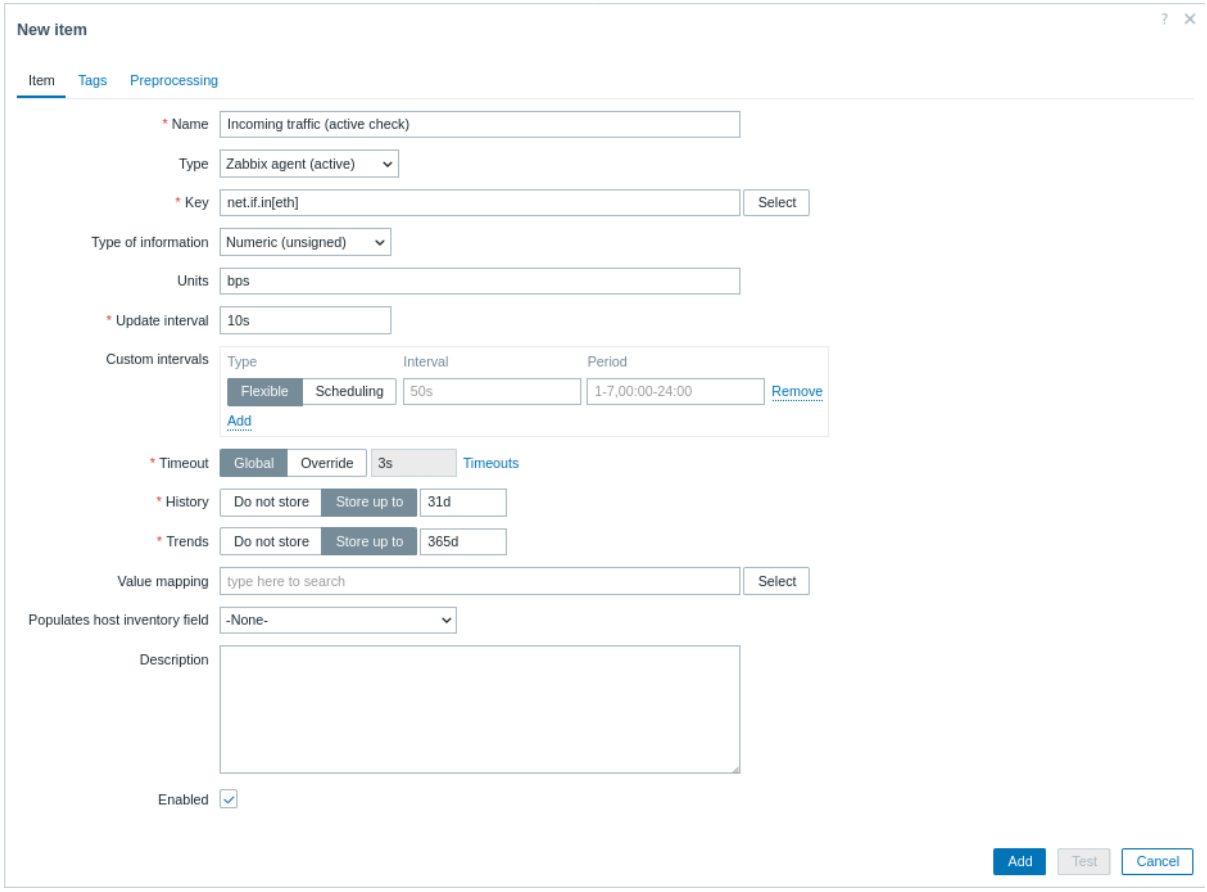
3. Noklikšķiniet uz *Add*, lai pievienotu hostu. Šis hosts attēlos uzraudzīto Linux iekārtu.

Vienumu izveide

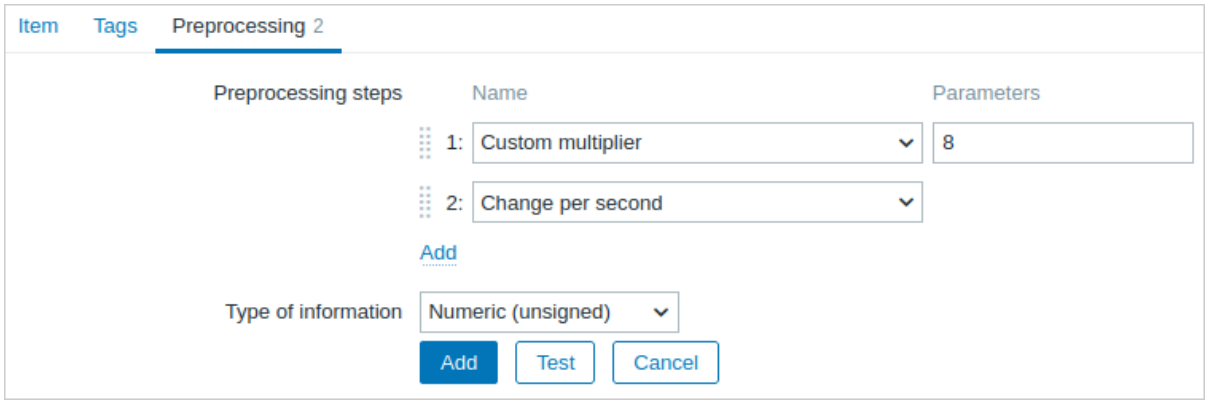
Izpildiet norādījumus par vienuma izveidi, lai pievienotu vienumus datplūsmas uzraudzībai, proti:

- Ienākošā datplūsma
- Izejošā datplūsma
- Kopējā datplūsma

Vienkārša ienākošās datplūsmas uzraudzības vienuma konfigurācija ar aktīvo pārbaudi izskatītos šādi:



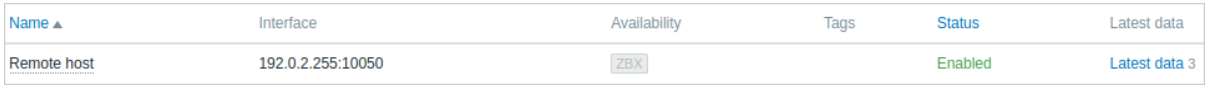
Lai savāktie dati būtu piemēroti praktiskai lietošanai, veidojot vienumus, varat iestatīt dažus priekšapstrādes soļus. Dotajā gadījumā tie var būt reizināšana ar 8 (lai pārvērstu baitus bitos) un attēlošana kā izmaiņas sekundē.



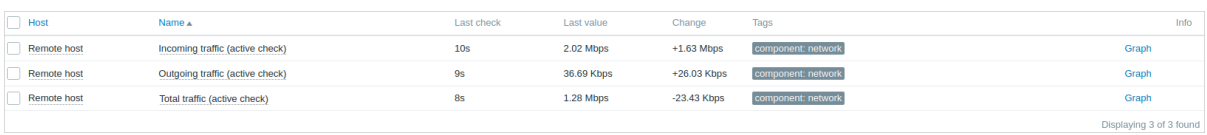
Skatīt savāktos datus

Apsveicam! Šajā brīdī Zabbix jau uzrauga jūsu tīkla datplūsmu.

Lai skatītu savāktos metriskos datus, atveriet izvēlnes sadaļu *Monitoring > Hosts* un hosts rindā noklikšķiniet uz *Latest data*.



Jūs redzēsiet datplūsmas datus.



Skatīt arī:

- Skatīt grafikus - kā attēlot savāktos datus kā grafikus.
- Konfigurēt trigerus - kā iestatīt trigerus, lai noteiktu neparastu tīkla datplūsmu.
- Iestatīt problēmu brīdinājumus - kā iestatīt paziņojumus par problēmsituācijām.

- Problēmu eskalācijas - kā izveidot vairāku soļu brīdinājumu scenārijus (piem., 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 novēršanu, apspiest vai slēgt problēmu.
- Linux uzraudzība ar Zabbix aģentu - kā sākt pamata uzraudzību no svarīgākajiem vienumiem, piesaistot iepriekš konfigurētu veidni.
- Instalēšana no pakotnēm - kā instalēt Zabbix komponentes, izmantojot oficiālās RPM un DEB pakotnes dažādiem Linux distributīviem, nodrošinot piekļuvi jaunākajām funkcijām un kļūdu labojumiem.

Pārraugiet tīmekļa vietnes ar pārlūka vienumiem

Ievads

Šajā lapā ir aprakstītas darbības, kas nepieciešamas, lai sāktu pamata vietņu uzraudzību ar pārlūka vienumiem.

Kam ir paredzēts šis ceļvedis

Šis ceļvedis ir paredzēts jauniem Zabbix lietotājiem un satur minimālo darbību kopu, kas nepieciešama, lai iespējotu pamata vietņu uzraudzību ar Browser vienumiem. Ja meklējat plašas pielāgošanas iespējas vai nepieciešama sarežģītāka konfigurācija, skatiet Browser vienumi lapu vai Zabbix rokasgrāmatas sadaļu Konfigurācija.

Priekšnosacījumi

Pirms turpināt šo rokasgrāmatu, jums ir jālejupielādē un jāinstalē Zabbix serveris un Zabbix lietotāja saskarne saskaņā ar jūsu operētājsistēmas norādījumiem.

Atkarībā no jūsu vides daži šīs rokasgrāmatas soļi var nedaudz atšķirties. Šī rokasgrāmata ir balstīta 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 pārlūku un mijiedarbojas ar to, izpildot testa komandas, piemēram, pogu klikšķināšanu vai teksta ievadīšanu. Kā piemērs šajā rokasgrāmatā tiks izmantots Selenium Server ar Chrome Docker konteinerā.

Tiek pieņemts, ka Docker jau ir konfigurēts. Šī rokasgrāmata neaptver Docker konfigurēšanu. Instalēšanas norādījumus skatiet 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** - piesaista 4444 portu jūsu saimniekdatorā 4444 portam konteinerā (šis ir ports, ko Selenium Server izmanto komandu pieņemšanai);
- **-p 7900:7900** - piesaista 7900 portu jūsu saimniekdatorā 7900 portam konteinerā (šis ir ports, ko izmanto Virtual Network Computing (VNC) serveris, ļaujot attālināti skatīt pārlūka GUI; nepieciešams VNC klients);
- **--shm-size="2g"** - piešķir konteineram 2 GB koplietotās atmiņas (tas ir svarīgi, lai Chrome darbotos pareizi, jo tam var būt nepieciešams ievērojams koplietotās atmiņas apjoms, lai izvairītos no avārijām);
- **-d** - palaiž konteineru atdalītā 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 Docker konteiners `browser` 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 tīmekļa lapas 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.

Zabbix servera konfigurēšana

Pārlūkprogrammas vienumus izpilda un apstrādā *pārlūkprogrammas aptaujātāja* Zabbix procesi, kas jāiespējo, pielāgojot StartBrowserPollers servera konfigurācijas parametru. Turklāt parametram WebDriverURL jānorāda iepriekš konfigurētais tīmekļa testēšanas galapunkts.

Pēc noklusējuma parametram StartBrowserPollers ir iestatīta vērtība 1, tāpēc 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 interface HTTP(S) URL. For example, http://localhost:4444 used with Selenium WebDriver star
#####
##### Mandatory: no
##### Default:
##### WebDriverURL=

WebDriverURL=192.0.2.1:4444
```

3. Restartējiet Zabbix serveri.

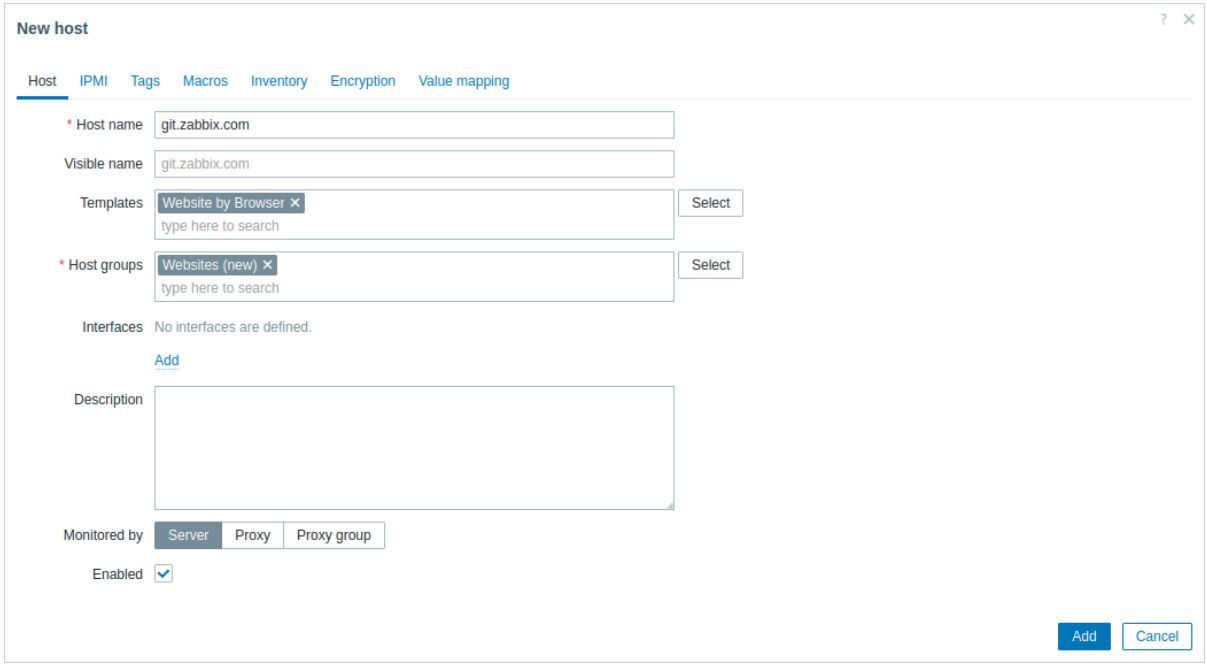
```
systemctl restart zabbix-server
```

Konfigurēt Zabbix lietotāja saskarni

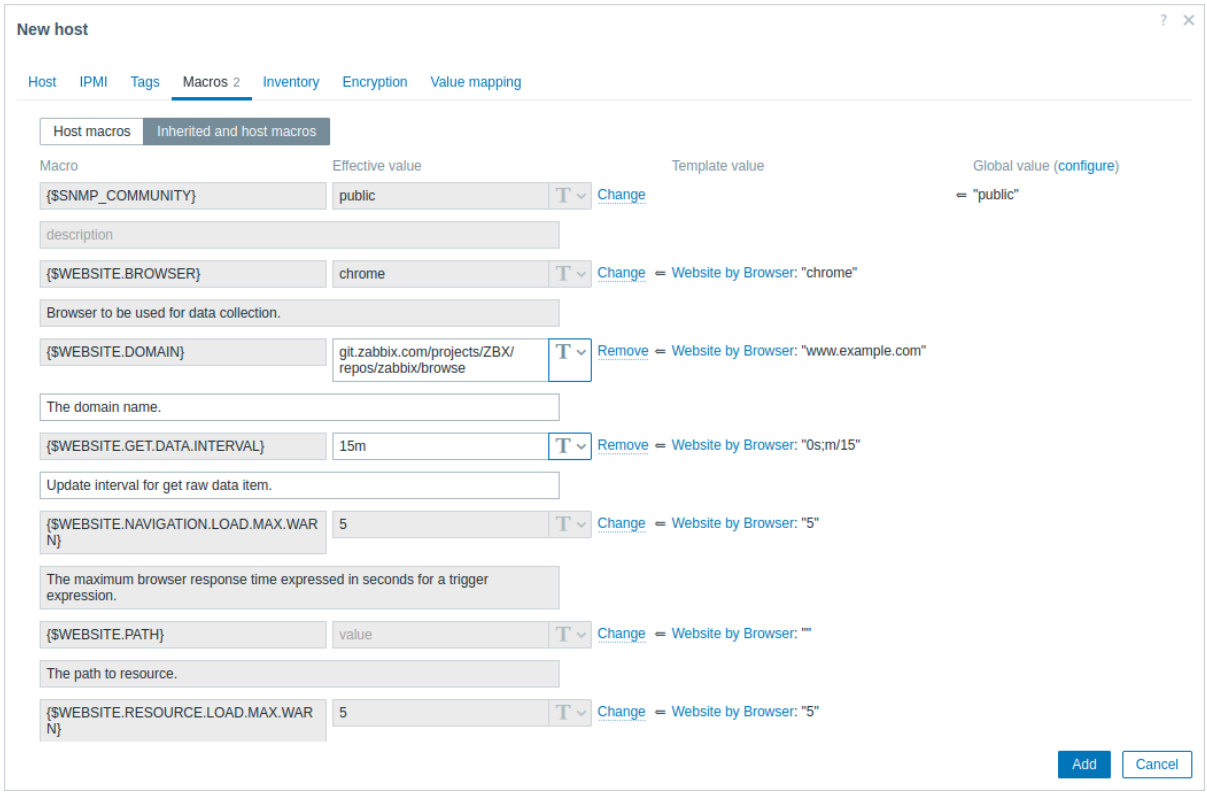
1. Piesakieties Zabbix lietotāja saskarnē.

2. Izveidojiet hostu Zabbix tīmekļa saskarnē:

- Laukā *Host name* ievadiet hosta nosaukumu (piemēram, "git.zabbix.com").
- Laukā *Templates* ievadiet vai atlasiet veidni "Website by Browser". Plašāku informāciju par šo veidni skatiet Website by Browser.
- Laukā *Host groups* ievadiet vai atlasiet hostu grupu (piemēram, jaunu hostu grupu "Websites").



- Cilnē *Macros* pārslēdzieties uz *Inherited and host macros*, atrodiet tālāk norādītos makro un noklikšķiniet uz *Change* blakus makro 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)

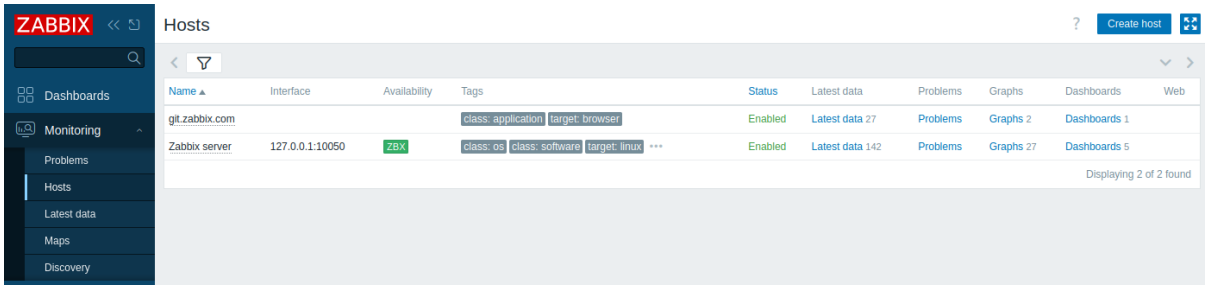


3. Noklikšķiniet uz pogas *Add*, lai izveidotu hostu. Šis hosts attēlos tīmekļa vietni, kuru vēlaties uzraudzīt.

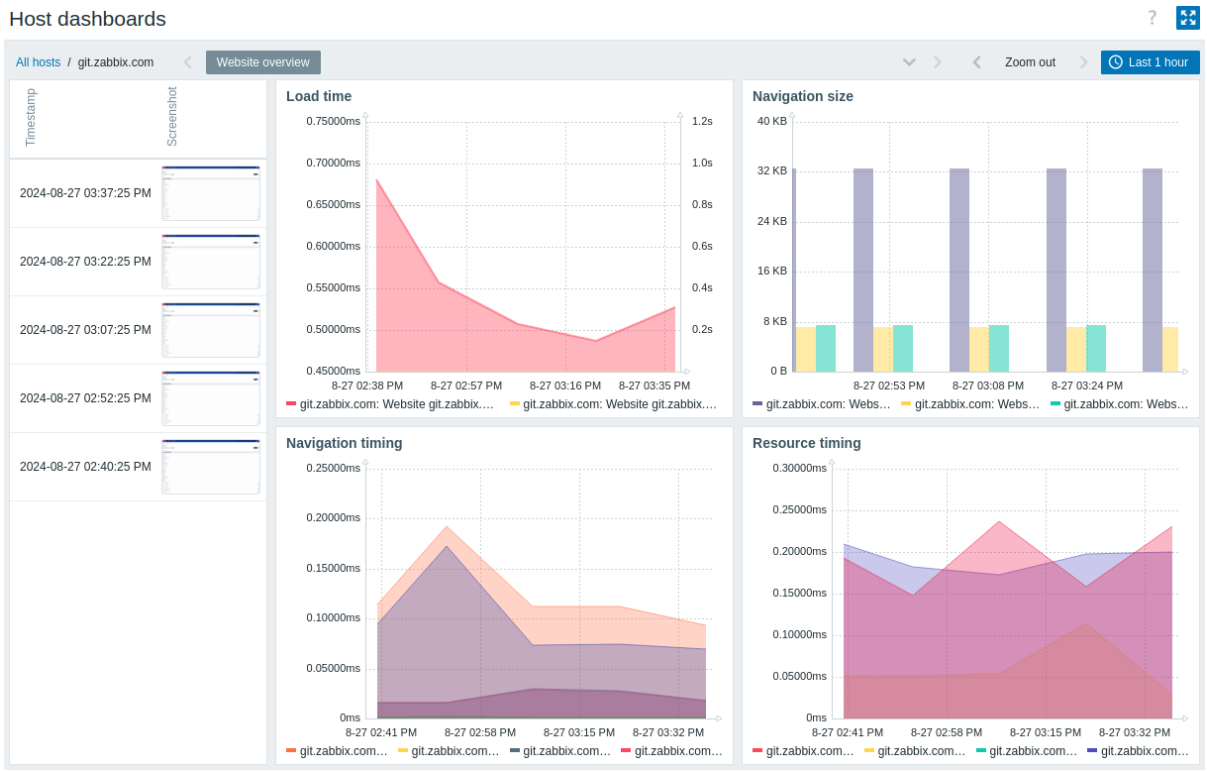
Skatīt savāktos metriskos datus

Apsveicam! Šajā brīdī Zabbix jau uzrauga jūsu norādīto vietni.

Lai skatītu savāktos metriskos datus, dodieties uz izvēlnes sadaļu *Monitoring > Hosts* un noklikšķiniet uz *Dashboards* blakus hostam.



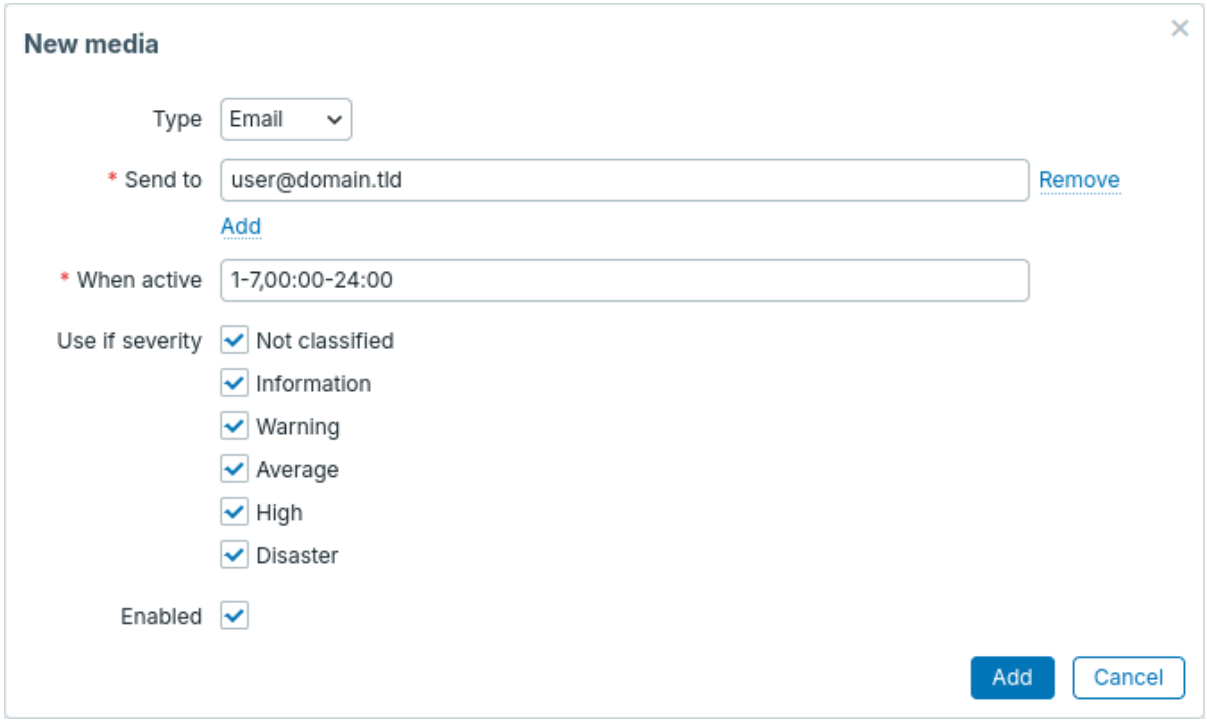
Šī darbība jūs aizvedīs uz hosta informācijas paneli (konfigurētu veidnes līmenī) ar svarīgākajiem metriskajiem datiem, kas savākti no vietnes.



Iestatiet problēmu brīdinājumus

Zabbix var paziņot jums par problēmu jūsu infrastruktūrā, izmantojot dažādas metodes. Šajā rokasgrāmatā ir sniegti pamata konfigurācijas soļi e-pasta brīdinājumu nosūtīšanai.

1. Dodieties uz *Lietotāja iestatījumi > Profils*, pārslēdzieties uz cilni *Media* un pievienojiet savu e-pasta adresi.



2. Izpildiet rokasgrāmatu par problēmas paziņojuma saņemšanu.

Nākamreiz, kad Zabbix noteiks problēmu, jums vajadzētu saņemt brīdinājumu pa e-pastu.

Pārbaudiet savu konfigurāciju

Lai pārbaudītu savu konfigurāciju, mēs varam simulēt reālu problēmu, atjauninot host konfigurāciju Zabbix lietotāja saskarnē.

1. Atveriet sava vietnes host konfigurāciju Zabbix.

2. Pārslēdzieties uz cilni *Macros* un atlasiet *Inherited and host macros*.

3. Noklikšķiniet uz *Change* blakus, piemēram, iepriekš konfigurētajai 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 host konfigurāciju.

5. Pēc dažiem mirkļiem Zabbix noteiks problēmu "Failed to get JSON of the requested website", jo tas nevarēs izveidot savienojumu ar norādīto 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 vietnes uzraudzību.

Skatīt arī

- Vienuma izveide - kā sākt uzraudzīt papildu metriku.
- Problēmu eskalācijas - kā izveidot vairāku soļu brīdinājumu scenārijus (piem., vispirms nosūtīt ziņojumu sistēmas administratoram, pēc tam, ja problēma netiek atrisināta 45 minūšu laikā, nosūtīt ziņojumu datu centra pārvaldniekam).
- Pārlūkprogrammas vienumi - kā konfigurēt pārlūkprogrammas vienumus.
- Veidne Website by Browser - papildu informācija par veidni *Website by Browser*.

Pārraugiet vietņu sertifikātus ar Zabbix aģentu 2 (pasīvs)

Ievads

Šajā rokasgrāmatā ir sniegts visaptverošs pārskats par to, kā iestatīt un uzraudzīt SSL/TLS sertifikātus, izmantojot `web.certificate.get` atslēgu Zabbix aģents 2. Tā ir izstrādāta, lai vienkāršotu sertifikātu uzraudzību vienai vai vairākām tīmekļa vietnēm, nodrošinot, ka administratori var ātri identificēt iespējamās problēmas, piemēram, beigušos vai nederīgus sertifikātus.

Kam šī rokasgrāmata ir paredzēta

Šī rokasgrāmata ir paredzēta jauniem Zabbix lietotājiem un satur minimālo darbību kopumu, kas nepieciešams, lai iespējotu pamata vietņu sertifikātu uzraudzību. Ja meklējat plašas pielāgošanas iespējas vai nepieciešama sarežģītāka konfigurācija, skatiet Zabbix rokasgrāmatas sadaļu Configuration.

Priekšnosacījumi

Pirms turpināt šo rokasgrāmatu, jums ir jālejupielādē un jāinstalē Zabbix serveris, Zabbix lietotāja saskarne un Zabbix aģents 2 saskaņā ar jūsu OS norādījumiem. Šajā pamācībā tiek pieņemts, ka gan Zabbix serveris, gan aģents ir instalēti vienā un tajā pašā datorā; tāpēc konfigurācijā tiek izmantots `127.0.0.1`.

Zabbix aģenta 2 konfigurēšana

1. Atveriet Zabbix aģenta konfigurācijas failu:

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

2. Iestatiet parametram Server vērtību 127.0.0.1, jo aģents un serveris darbojas vienā datorā:

```
Server=127.0.0.1
```

3. Saglabājiet failu un restartējiet Zabbix aģenta 2 pakalpojumu:

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

4. Pēc Zabbix aģenta 2 iestatīšanas un konfigurēšanas pārbaudiet tā pieejamību, izmantojot:

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

Zabbix aģentā 2 pēc noklusējuma ir iekļauts WebCertificate spraudnis, tāpēc atsevišķa instalēšana vai konfigurēšana nav nepieciešama.

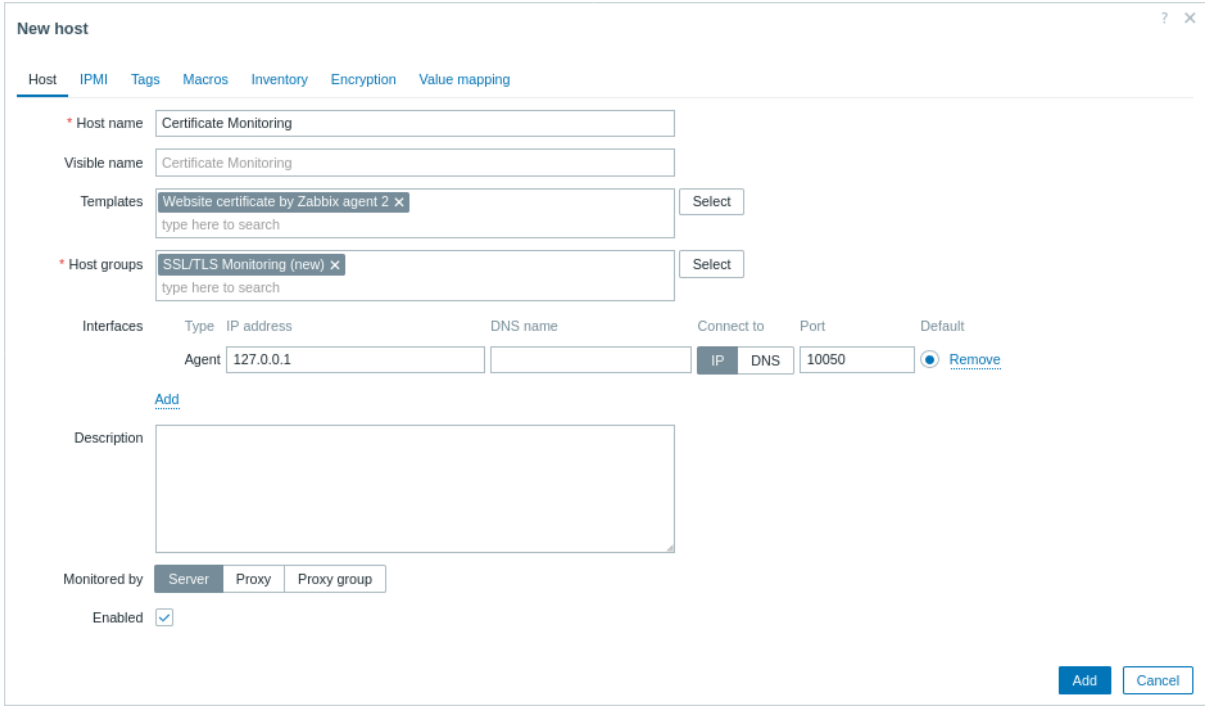
Zabbix lietotāja saskarnes konfigurēšana

1. Piesakieties Zabbix lietotāja saskarnē.

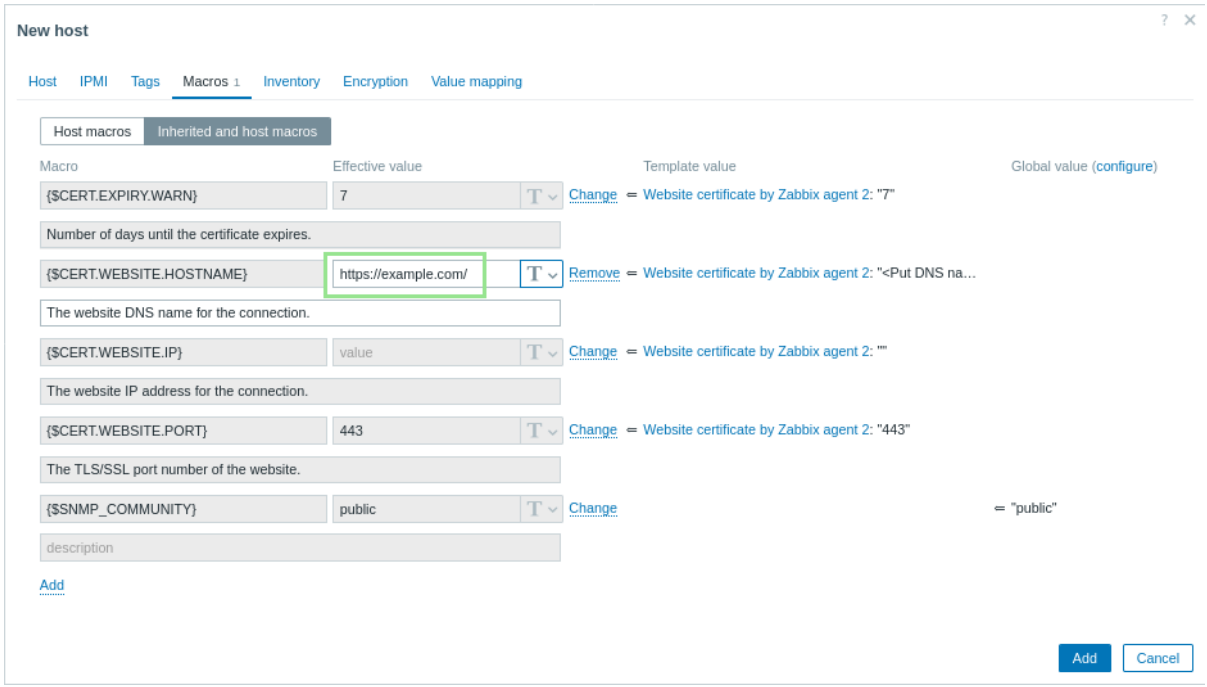
2. Dodieties uz *Monitoring* > *Hosts*.

3. Noklikšķiniet uz esoša hosta, kurā vēlaties uzraudzīt vietnes sertifikātus, vai, ja nepieciešams, izveidojiet hostu:

- Laukā *Host name* ievadiet hosta nosaukumu (piem., ”Certificate Monitoring”).
- Laukā *Templates* ievadiet vai atlasiet veidni ”Website certificate by Zabbix agent 2”, kas tiks saistīta ar hostu.
- Laukā *Host groups* ievadiet vai atlasiet hostu grupu (piem., ”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 tālāk norādītos makro un noklikšķiniet uz *Change* blakus makro 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, `{$CERT.WEBSITE.HOSTNAME}` makro var norādīt ar komatiem atdalītu to hostu nosaukumu sarakstu. Pēc izvēles `{$CERT.WEBSITE.PORT}` makro var 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

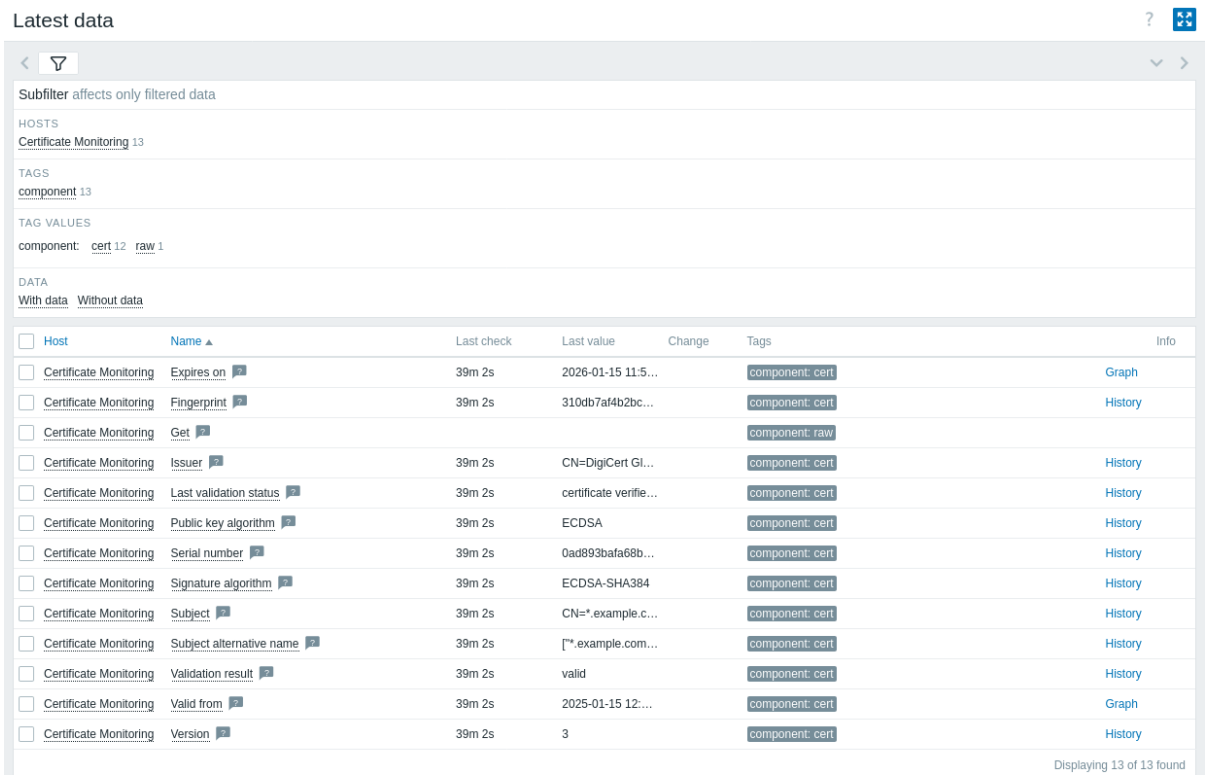
Katram `{$CERT.WEBSITE.HOSTNAME}` makro norādītajam vietnes nosaukumam Zabbix izveidos atbilstošu vienumu un trigeru kopu. Tas ļauj atsevišķi uzraudzīt un saņemt brīdinājumus par katras vietnes SSL sertifikātu.

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 metriskos datus

Apsveicam! Šajā brīdī Zabbix jau uzrauga vēlamo tīmekļa sertifikātu.

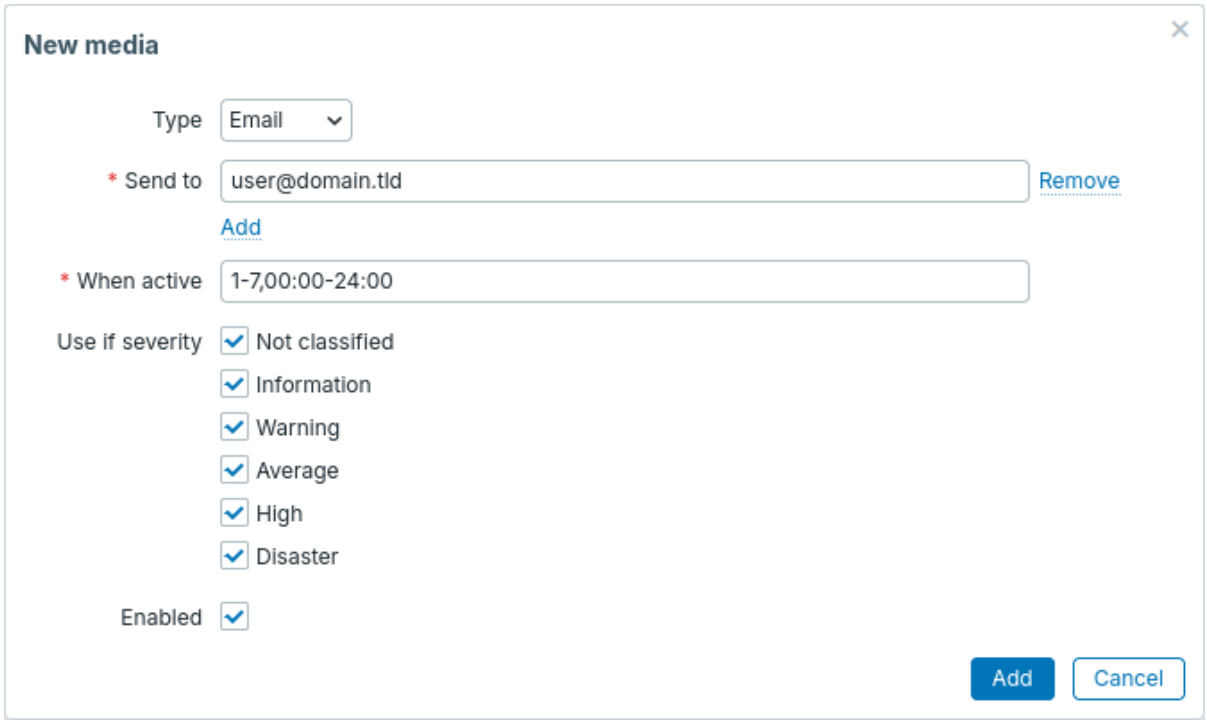
Lai skatītu savāktos metriskos datus, dodieties uz izvēlnes sadaļu *Monitoring > Hosts* un noklikšķiniet uz *Latest data* blakus hostam, lai skatītu visus jaunākos savāktos metriskos datus sarakstā, piemēram, derīguma termiņu, izdevēju un subjektu.



Iestatiet problēmu brīdinājumus

Zabbix var paziņot jums par infrastruktūras problēmām. Šajā rokasgrāmatā ir sniegti pamata konfigurācijas soļi e-pasta brīdinājumu nosūtīšanai.

1. Dodieties uz *Lietotāja iestatījumi > Profils*, pārslēdzieties uz cilni *Media* un pievienojiet savu e-pastu.



2. Izpildiet rokasgrāmatu par problēmas paziņojuma saņemšanu.

Nākamreiz, kad Zabbix noteiks problēmu, jums vajadzētu saņemt brīdinājumu pa e-pastu.

Pārbaudiet savu konfigurāciju

Lai pārbaudītu savu konfigurāciju, mēs varam simulēt reālu problēmu, atjauninot host konfigurāciju Zabbix lietotāja saskarnē.

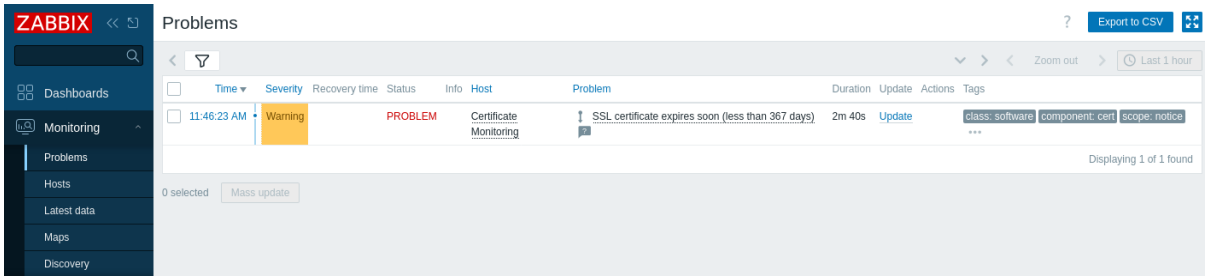
1. Atveriet sava ”Certificate Monitoring” host konfigurāciju Zabbix.

2. Pārslēdzieties uz *Macros* cilni un atlasiet *Inherited and host macros*.

3. Noklikšķiniet uz *Change* blakus iepriekš konfigurētajai `{$CERT.EXPIRY.WARN}` makro 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 host konfigurāciju.

5. Pēc dažiem mirkļiem Zabbix noteiks problēmu "SSL certificate expires soon" ar dienu skaitu līdz derīguma termiņa beigām. Problēma būs redzama *Monitoring > Problems*.



Ja ir konfigurēti brīdinājumi, jūs saņemsiet arī problēmas paziņojumu.

6. Mainiet makro vērtību atpakaļ uz iepriekšējo vērtību, lai atrisinātu problēmu un turpinātu sertifikāta vērtību uzraudzību.

Skatīt 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*.
- Monitor websites with Browser items - kā sākt pamata vietņu uzraudzību ar Browser vienumiem.

Pārraugiet tīkla komutatoru vai maršrutētāju ar Zabbix

Ievads

Šī rokasgrāmata soli pa solim palīdzēs jums sākt pamata tīkla slēdža vai maršrutētāja uzraudzību, izmantojot Zabbix. Kā piemērs tiek izmantots Cisco maršrutētājs, taču procedūra attiecas uz jebkuru SNMP iespējotu tīkla ierīci.

Kam šī rokasgrāmata ir paredzēta

Šī rokasgrāmata ir paredzēta jauniem Zabbix lietotājiem un tīkla administratoriem, kuri vēlas ātri iespējot pamata tīkla ierīču uzraudzību. Ja jums nepieciešama padziļināta pielāgošana vai papildu konfigurācijas opcijas, lūdzu, skatiet SNMP aģents lapu vai Zabbix rokasgrāmatas sadaļu Konfigurācija.

Priekšnosacījumi

Pirms turpināt šo rokasgrāmatu, pārliecinieties, ka jums ir:

- Instalēts Zabbix serveris un Zabbix lietotāja saskarne: instalējiet saskaņā ar savas operētājsistēmas instrukcijām (skatiet Instalēšana no pakotnēm un Tīmekļa saskarnes instalēšana).
- Instalēts Zabbix aģents, ja tiek uzraudzīti lokālā tīkla rādītāji.
- SNMP iespējota ierīce: tīkla slēdzis vai maršrutētājs (piemēram, Cisco maršrutētājs) ar iespējotu SNMP.
- Instalēti MIB faili: MIB failu instalēšana ļauj Zabbix pārvērst skaitliskos OID cilvēklasāmos 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:

1. Instalējiet MIB lejupielādes pakotni:

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

Ja nepieciešams pievienot ražotājam specifiskus MIB (piem., no Cisco, Juniper), ievietojiet tos atbilstošajā MIB direktorijā:

- Linux sistēmās bieži izmantotās atrašanās vietas ir `/usr/share/snmp/mibs/` vai `/usr/local/share/snmp/mibs/`.
- Zabbix instalācijās MIB failus var glabāt `/var/lib/zabbix/mibs/`.

Pārliecinieties, ka `MIBDIRS` vides mainīgais vai `snmp.conf` fails ietver pareizo ceļu.

Lai pārbaudītu, vai jūsu sistēma atpazīst jaunos MIB, izmantojiet:

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

Detalizētus norādījumus skatiet savas SNMP bibliotēkas dokumentācijā:

- Cisco MIB
- Juniper MIB

2. Rediģējiet `/etc/snmp/snmp.conf` un komentējiet rindu, kas sākas ar `mibs :`, lai sistēma varētu ielādēt visus pieejamos MIB.

3. Pārbaudiet, palaižot `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 šīs rokasgrāmatas soļi var nedaudz atšķirties. Šī rokasgrāmata ir balstīta 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 ir jākonfigurē tīkla ierīce, 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 maršrutētājam konfigurācija parasti ietver tālāk norādītās darbības.

SNMPv2 piemērs

1. Iespējojiet SNMP un iestatiet community string.

Pierakstieties sava Cisco maršrutētāja konsolē un ievadiet konfigurācijas režīmu:

```
configure terminal
```

Pēc tam iespējojiet SNMP, norādot tikai lasīšanas community string. Piemēram:

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

Aizstājiet `<your_community_string>` ar savu drošo community string. Piezīme: RO (Read-Only) opcija ļauj SNMP izgūt datus no ierīces, bet neļauj veikt 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 pēc pārstartēšanas:

```
write memory
```

SNMPv3 piemērs

SNMPv3 nodrošina uzlabotu drošību ar autentifikāciju un šifrēšanu. Tā konfigurācija ir drošāka nekā SNMPv2, un tā jāpārbauda, salīdzinot ar jūsu ierīcei specifisko dokumentāciju.

1. Izveidojiet SNMP grupu.

Konfigurējiet SNMPv3 grupu ar iespējotu 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
```

Papildu informācijai vai ar modeli saistītām instrukcijām varat skatīt ārējos Cisco SNMP konfigurācijas pamācību. Tomēr šajā ceļvedī ir sniegtas pamata darbības SNMP monitoringa iespējošanai.

Konfigurēt Zabbix lietotāja saskarni

Izveidot hostu Zabbix lietotāja saskarnē

1. Piesakieties Zabbix lietotāja saskarnē.

2. Pievienojiet jaunu hostu.

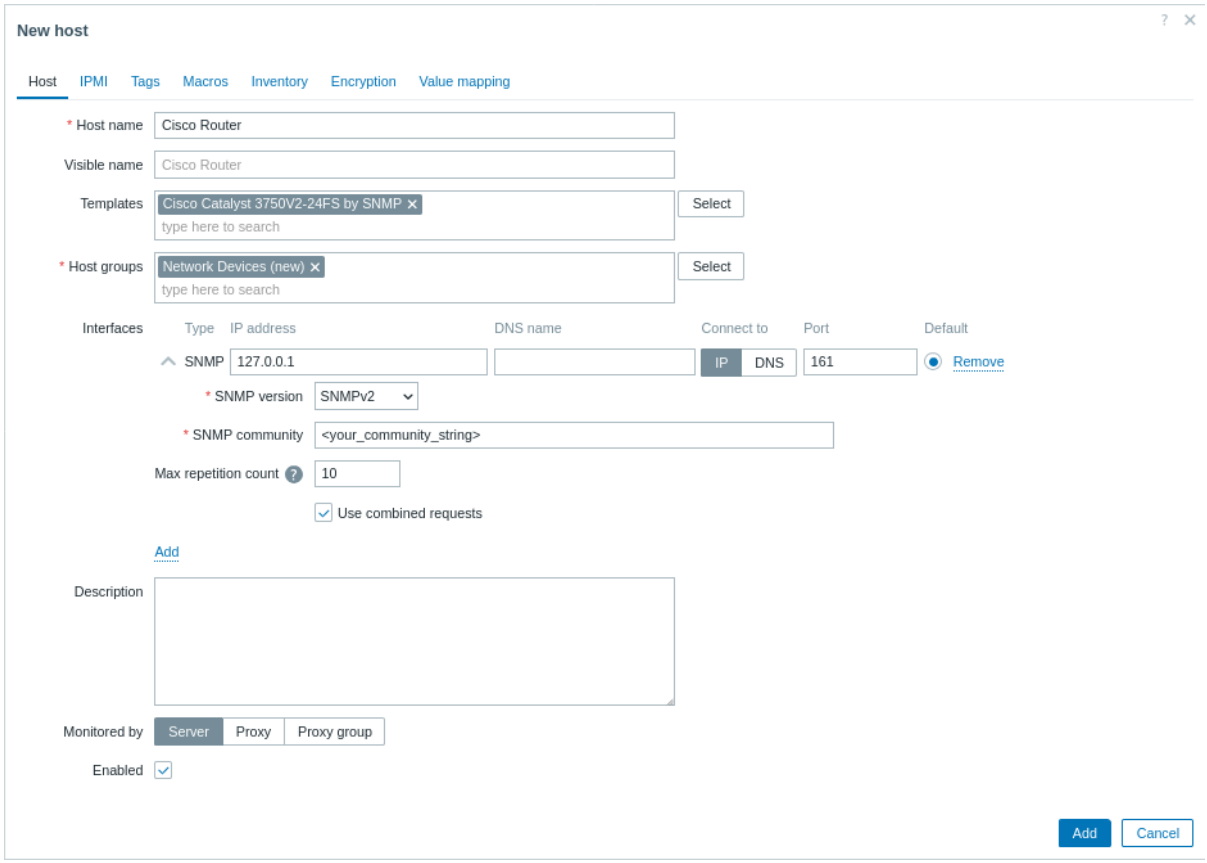
Dodieties uz *Data collection* > *Hosts* un noklikšķiniet uz *Create host*.

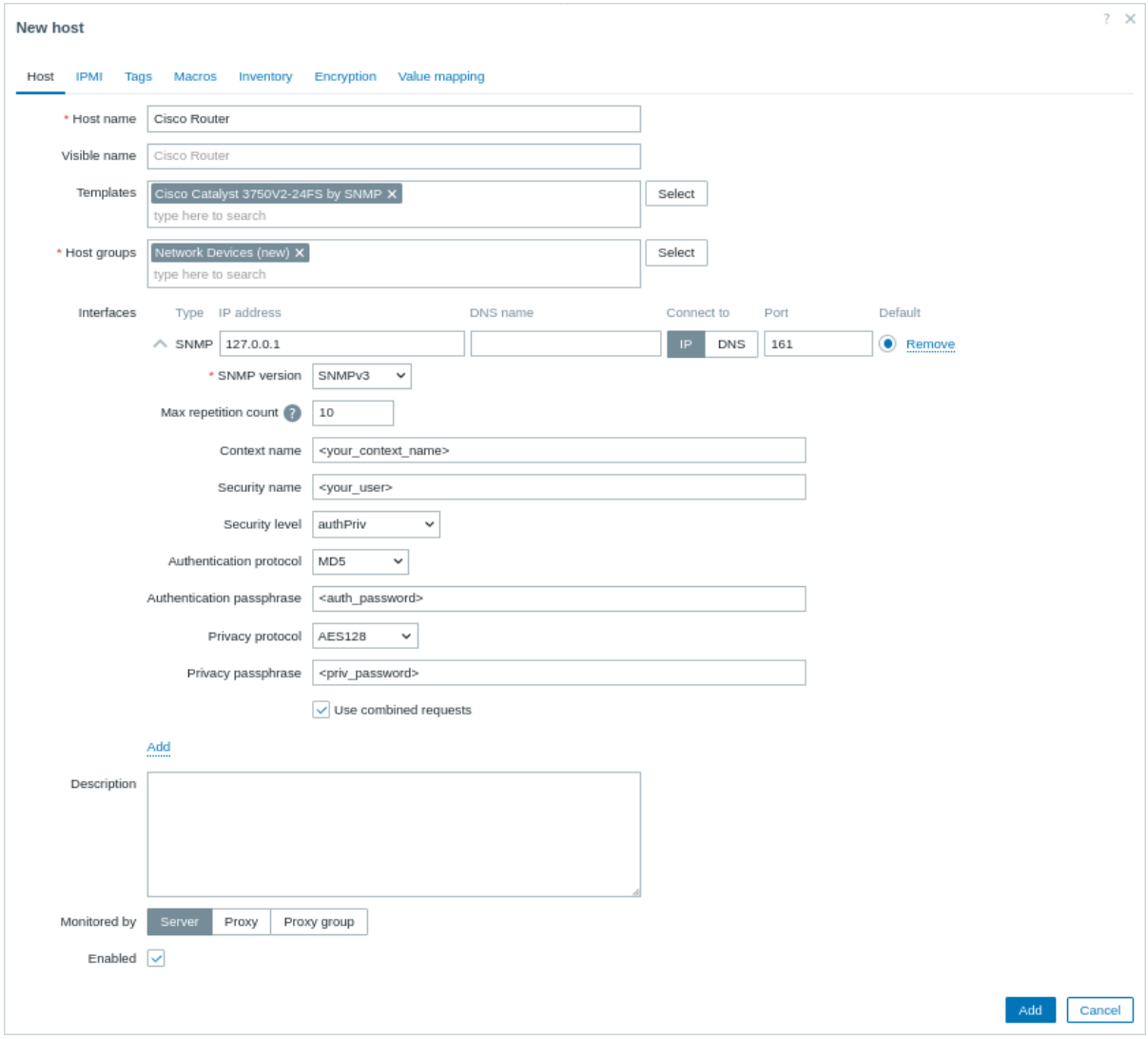
- *Host name*: ievadiet savai ierīcei nosaukumu (piem., ”Cisco Router”).
- Host groups: atlasiet esošu grupu vai izveidojiet jaunu grupu, piemēram, ”Network Devices”.
- Interfaces:
  - Noklikšķiniet uz *Add* sadaļā Interfaces.
  - Kā saskarnes tipu izvēlieties *SNMP*.
  - Ievadiet sava Cisco maršrutētāja IP adresi vai DNS nosaukumu.
  - Iestatiet noklusējuma SNMP portu (parasti 161).
  - Izmantojiet nolaižamo izvēlni, lai atlasītu atbilstošo SNMP versiju (piem., SNMPv2).
  - SNMPv1/v2 gadījumā laukā *SNMP community* ievadiet community string. SNMPv3 gadījumā tiks pieprasīti papildu akreditācijas dati (*Context name*, *Security name* un *Security level* u.c.).

3. Saistīt veidnes

Laukā *Templates* atlasiet SNMP veidni, kas vislabāk atbilst jūsu ierīcei. Zabbix piedāvā plašu iepriekš sagatavotu SNMP veidņu klāstu daudzām ierīču grupām. Piemēram, ja uzraugāt Cisco ierīci, izvēlieties veidni, kas atbilst jūsu ierīces OS vai modelim (piemēram, Cisco IOS SNMP vai Cisco Catalyst 3750<device model> SNMP).

4. Noklikšķiniet uz *Add*, lai saglabātu hostu.



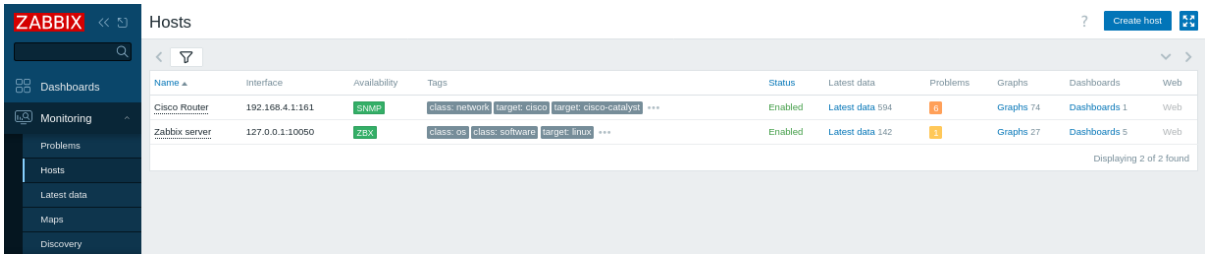


Skatīt savāktos metriskos datus

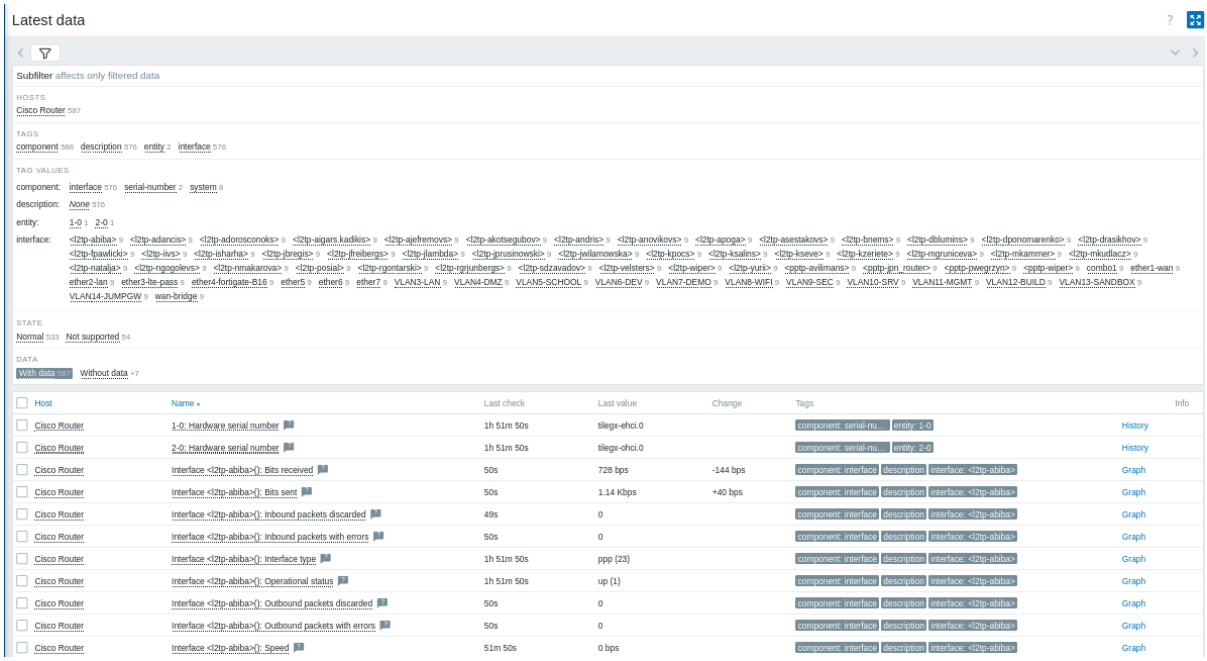
Apsveicam! Zabbix tagad ir iestatīts, lai uzraudzītu jūsu tīkla ierīci.

Jaunākie dati:

- Lietotāja saskarnē Zabbix dodieties uz Monitoring > Latest data.



- Atlasiet savu ”Cisco Router” hostu (vai atklātos hostus), lai skatītu metriskos datus, piemēram, aparatūras un tīkla darbspējas laiku, ICMP zudumu, ping un atbildes laiku u. c.



- Grafiki un ekrāni:

Lai vizualizētu veiktspējas datus, noklikšķiniet uz *Graphs* blakus SNMP vienumiem, lai skatītu detalizētus metriskos datus.

Kā nākamo soli varat:

- Pievienot pielāgotus SNMP vienumus, lai uzraudzītu papildu metriskos datus.
- Iestatīt problēmu brīdinājumus, lai saņemtu paziņojumus par iespējamām problēmām.

SNMP vienumu izveide

Kad hosts ir iestatīts, varat izveidot vienumus, lai uzraudzītu konkrētus metriskos rādītājus. Piezīme: šis solis nav obligāts, ja izmantojat veidni, jo veidnēs jau ir iekļauti noklusējuma vienumu komplekti.

1. Nosakiet SNMP OID:

Izmantojiet komandu `snmpwalk`, lai uzskaitītu pieejamos OID savā ierīcē. Piemēram:

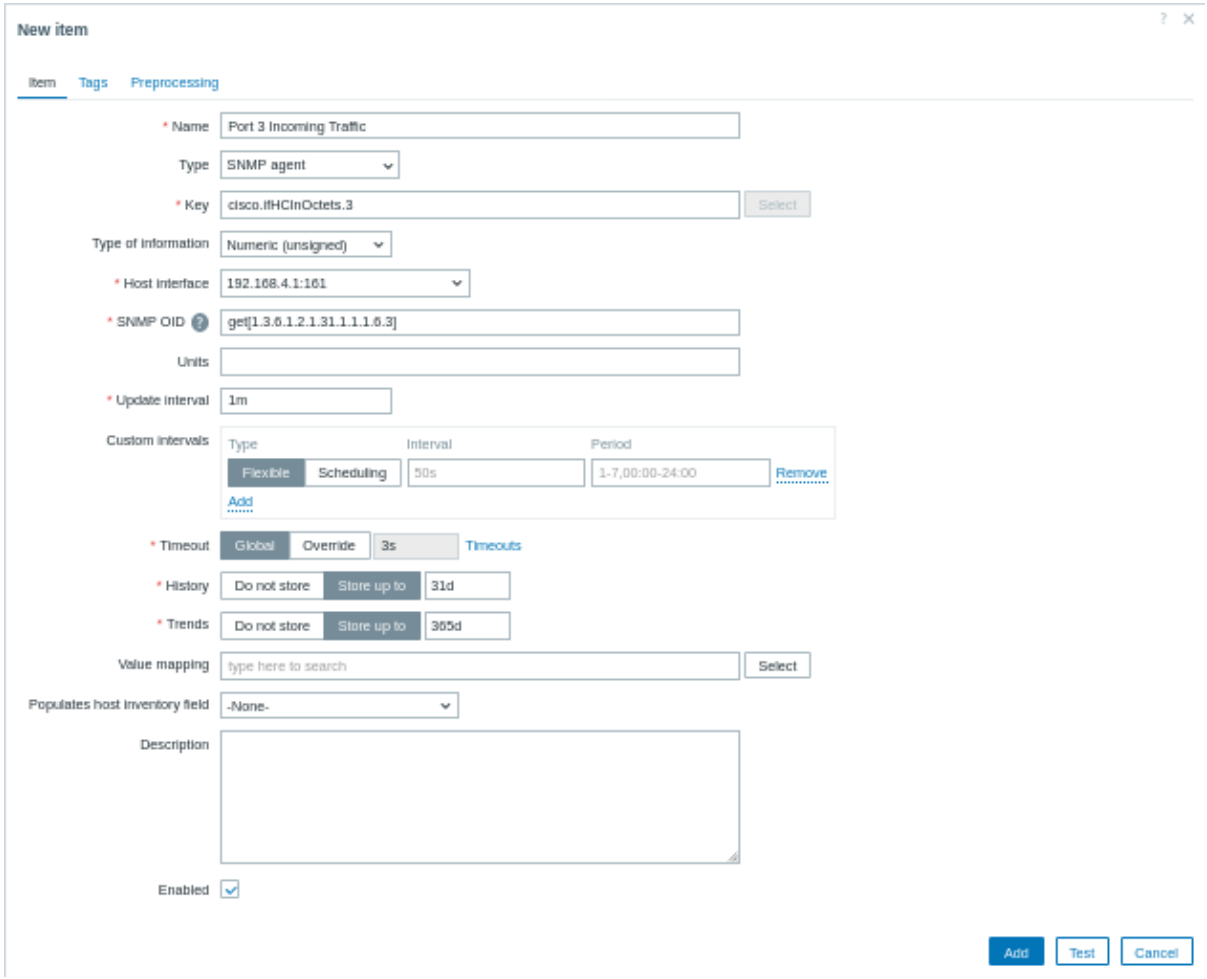
```
snmpwalk -v 2c -c <your_community_string> <device_IP> .
```

Atrodiet OID metriskajam rādītājam, kuru vēlaties uzraudzīt (piemēram, IF-MIB::ifHCInOctets.3 ienākošajai datplūsmai 3. portā). Lai iegūtu skaitlisko OID, varat izmantot:

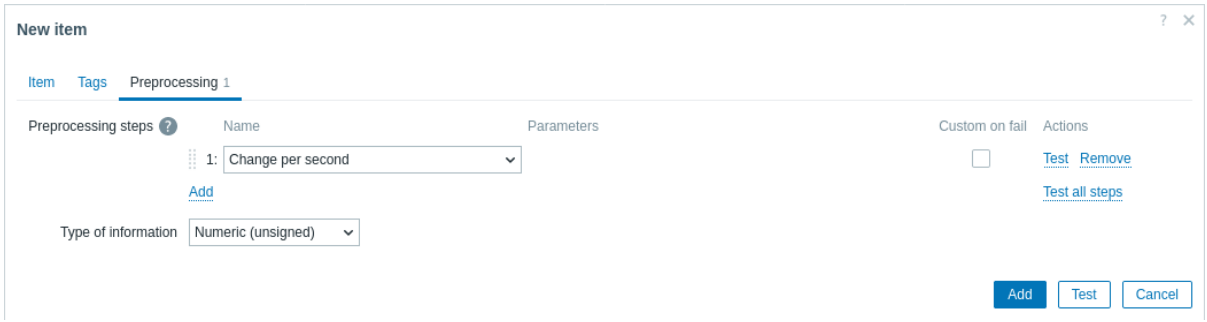
```
snmpget -v 2c -c <your_community_string> -On <device_IP> IF-MIB::ifHCInOctets.3
```

2. Izveidojiet SNMP vienumu:

- Dodieties uz *Datu vākšana* > *Hosts* un noklikšķiniet uz *Vienumi* cilnes savam SNMP hostam, pēc tam noklikšķiniet uz *Izveidot vienumu*.
- *Nosaukums*: ievadiet aprakstošu nosaukumu (piem., "3. porta ienākošā datplūsma").
- *Tips*: atlasiet *SNMP aģents*.
- *Atslēga*: norādiet jēgpilnu atslēgu (piem., `cisco.ifHCInOctets.3`).
- *Hosta interfeiss*: pārliecinieties, ka ir atlasīts SNMP interfeiss.
- *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.



- *Priekšapstrāde* (ja nepieciešams): ja vienums atgriež kumulatīvu skaitītāju (piemēram, interfeisa datplūsmu), dodieties uz *Priekšapstrāde* cilni un pievienojiet priekšapstrādes soli, piemēram, "Izmaiņas sekundē", 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, var būt nepieciešams konvertēt starp skaitliskajiem OID un tiem atbilstošajiem MIB nosaukumiem. Šis tulkojums palīdz vieglāk identificēt un novērst metriku problēmas.

- MIB nosaukuma tulkošana skaitliskā OID: izmantojiet komandu `snmptranslate` ar opciju `-On`. Piemēram, lai tulkotu MIB nosaukumu `IF-MIB::ifHCInOctets.3` tā skaitliskajā OID, izpildiet:

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

Šī komanda var izvadīt:

```
.1.3.6.1.2.1.31.1.1.1.6.3
```

- Skaitliskā OID tulkošana MIB nosaukumā: izmantojiet komandu `snmptranslate` ar opciju `-IR` (vai `-m ALL`), lai veiktu tulkojumu pretējā virzienā. Piemēram, lai tulkotu skaitlisko OID `.1.3.6.1.2.1.31.1.1.1.6.3` atpakaļ tā MIB nosaukumā, izpildiet:

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

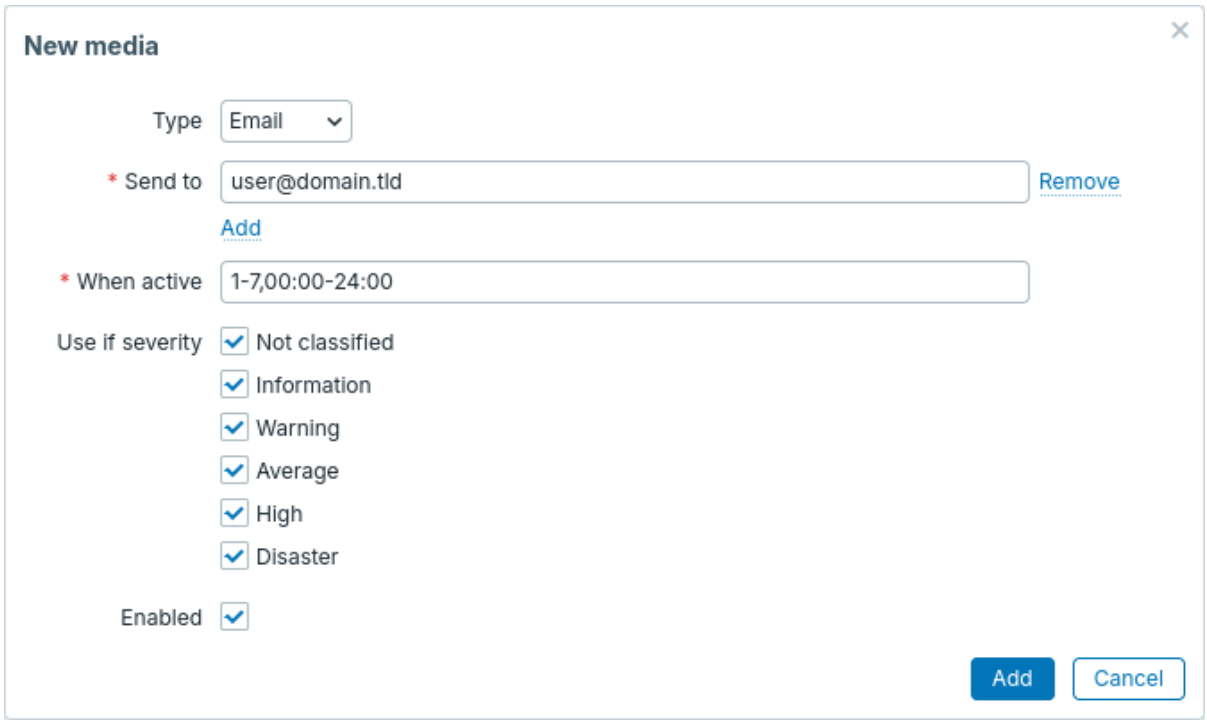
Šī komanda var izvadīt:

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

Iestatiet problēmu brīdinājumus

Šajā rokasgrāmatā ir sniegti pamata konfigurācijas soļi e-pasta brīdinājumu nosūtīšanai.

1. Dodieties uz *Lietotāja iestatījumi > Profils*, pārslēdzieties uz *Media* cilni un pievienojiet savu e-pasta adresi.



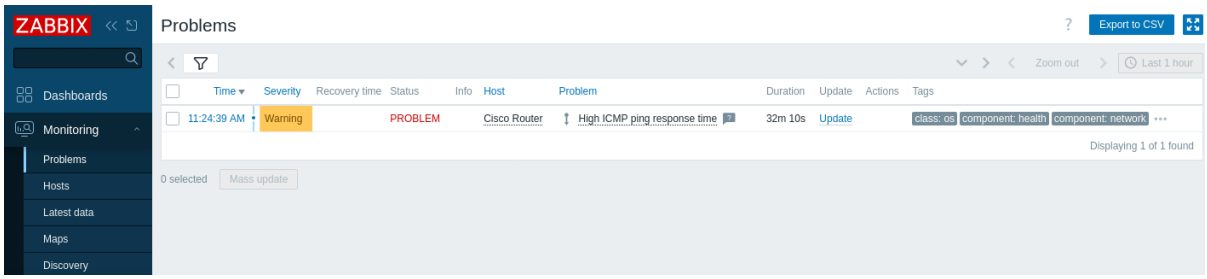
2. Izpildiet rokasgrāmatu par problēmas paziņojuma saņemšanu.

Nākamreiz, kad Zabbix noteiks 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 makro `{$ICMP_RESPONSE_TIME_WARN}` (vai līdzīgu atbildes laika sliekšņa makro).
4. Iestatiet ļoti mazu vērtību (piem., 0.001), lai aktivizētu brīdinājumu, kad ping atbildes laiks pārsniedz šo vērtību.
5. Noklikšķiniet *Update*, lai lietotu izmaiņas.
6. Pagaidiet dažus mirkļus, līdz Zabbix konstatē simulēto problēmu.
7. Dodieties uz *Monitoring > Problems*, lai pārbaudītu, vai parādās brīdinājums (piem., "High ICMP ping response time").



Ja brīdinājumi ir konfigurēti, jums arī jāsaņem problēmas brīdinājums.

8. Atjaunojiet makro vērtību tās sākotnējā iestatījumā un noklikšķiniet *Update*, lai saglabātu izmaiņas.
9. Apstipriniet, ka problēma ir atrisināta un pazūd no sadaļas *Problems*.

SNMP monitoringa problēmu novēršana

Ja pamanāt, ka SNMP ikona Zabbix lietotāja saskarne parādās SARKANA vai dati netiek apkopoti, izmēģiniet šādas darbības:

1. Pārbaudiet SNMP savienojumu.

SNMPv2 gadījumā palaidiet šādu komandu no sava Zabbix serveris:

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

Šī komanda pārbauda, vai ierīce atbild uz SNMP vaicājumiem.

SNMPv3 gadījumā norādiet 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 atbild drošā režīmā.

2. Pārliecinieties, ka MIB faili ir instalēti un iespējoti, kā aprakstīts priekšnosacījumos. Lai to nodrošinātu, šādai komandai, vaicājot tīkla ierīci, nedrīkst būt kļūdas:

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

Tam vajadzētu atgriezt tulkotus OID bez kļūdām.

3. Apstipriniet, ka Zabbix konfigurētā SNMP versija un akreditācijas dati atbilst tiem, kas iestatīti jūsu ierīcē. Piemēram, pārskatiet SNMP iestatījumus Zabbix hosts konfigurācijā un salīdziniet tos ar savas ierīces konfigurāciju. Cisco ierīcē varat pārbaudīt SNMP iestatījumus, 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 neviens ugunsmūris vai tīkla problēmas nebloķē SNMP trafiku (parasti 161. portā) starp Zabbix serveris un ierīci. Savienojumu 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, ja izmantojat iptables:

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

6. Pārskatiet Zabbix serveris 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.

Skatīt 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 noteikuma konfigurēšana - papildu informācija par to, kā konfigurēt tīkla atklāšanas noteikumu, ko Zabbix izmanto hosts un pakalpojumu atklāšanai.

Windows notikumu žurnāla monitorēšana, izmantojot aktīvās pārbaudes

Ievads

Šajā rokasgrāmatā ir paskaidrots, kā ar Zabbix, izmantojot aktīvās pārbaudes, uzraudzīt Windows notikumu žurnālus. Izmantojot Zabbix Windows specifiskās vienuma atslēgas, varat reāllaikā apkopot un analizēt kritiskus notikumus (piemēram, neveiksmīgus pieteikšanās mēģinājumus, sistēmas kļūdas u. c.).

Kam šī rokasgrāmata ir paredzēta

Šī rokasgrāmata ir paredzēta jauniem Zabbix lietotājiem un tīkla administratoriem, kuri vēlas uzraudzīt Windows notikumu žurnālus. Papildu konfigurācijas opcijām skatiet dokumentāciju par Windows specifiskajām vienuma atslēgām.

Priekšnosacījumi

Pirms turpināt darbu ar šo rokasgrāmatu, jums saskaņā ar savas operētājsistēmas norādījumiem ir jālejupielādē un jāinstalē Zabbix serveris un Zabbix lietotāja saskarne. Tāpat jums ir jālejupielādē un jāinstalē Zabbix aģents Windows datorā, kuru vēlaties uzraudzīt.

Zabbix aģenta konfigurēšana Windows notikumu žurnāla uzraudzībai

1. Atveriet Zabbix aģenta konfigurācijas failu savā Windows hostā un pārliecinieties, ka parametrs `ServerActive` ir iestatīts uz jūsu Zabbix servera IP adresi un parametrs `Hostname` atbilst hosta nosaukumam, kas tiks definēts Zabbix lietotāja saskarnē. Tas ļauj aģentam pieprasīt aktīvās pārbaudes savam hostam no norādītā Zabbix servera. Piemēram:

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

2. Restartējiet Zabbix aģenta servisu, lai lietotu izmaiņas:

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

3. Pārbaudiet, vai Windows hosts darbojas:

- Pārliecinieties, ka Zabbix aģenta serviss darbojas Windows hostā.
- Pārbaudiet, vai Windows hosts var izveidot savienojumu ar Zabbix serveri, izmantojot portu 10051. Lai pārbaudītu savienojamību no Windows hosta, atveriet PowerShell un izpildiet šo komandu:

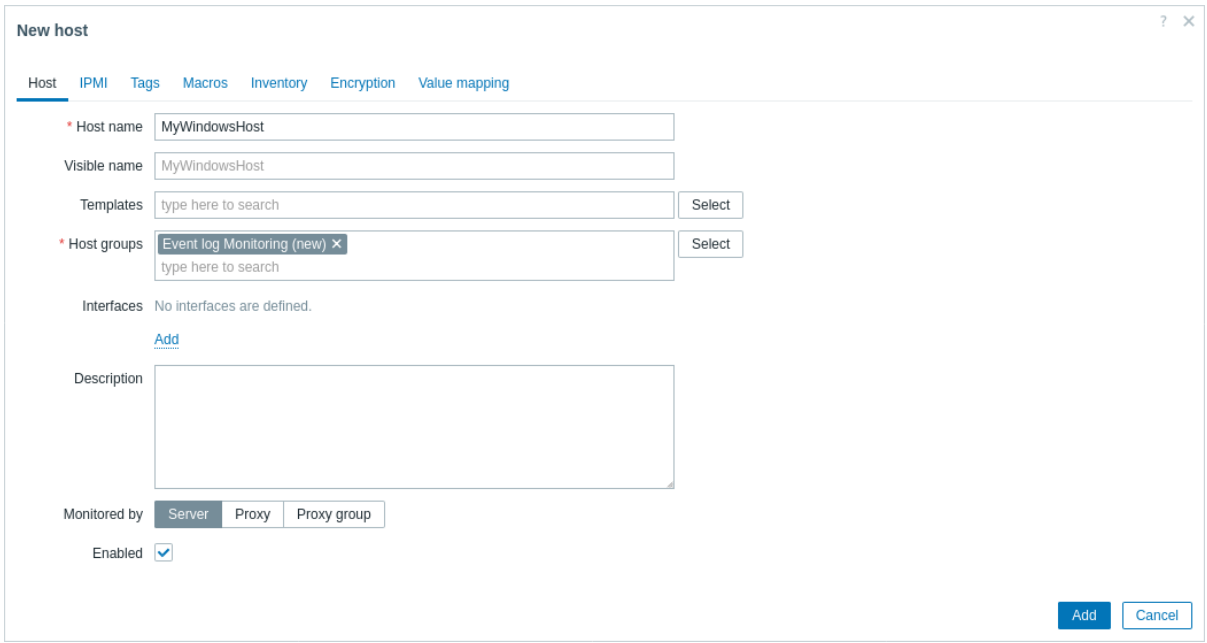
```
Test-NetConnection -ComputerName <Zabbix-server-IP> -Port 10051
```

Konfigurēt Zabbix lietotāja saskarni

1. Dodieties uz *Data collection* > *Hosts* un izveidojiet hostu:

- Laukā *Host name* ievadiet hosta nosaukumu (piem., ”MyWindowsHost”).
- Laukā *Host groups* ievadiet vai atlasiet hostu grupu (piem., ”Event log Monitoring”).
- Noklikšķiniet uz *Add*, lai saglabātu konfigurēto hostu.

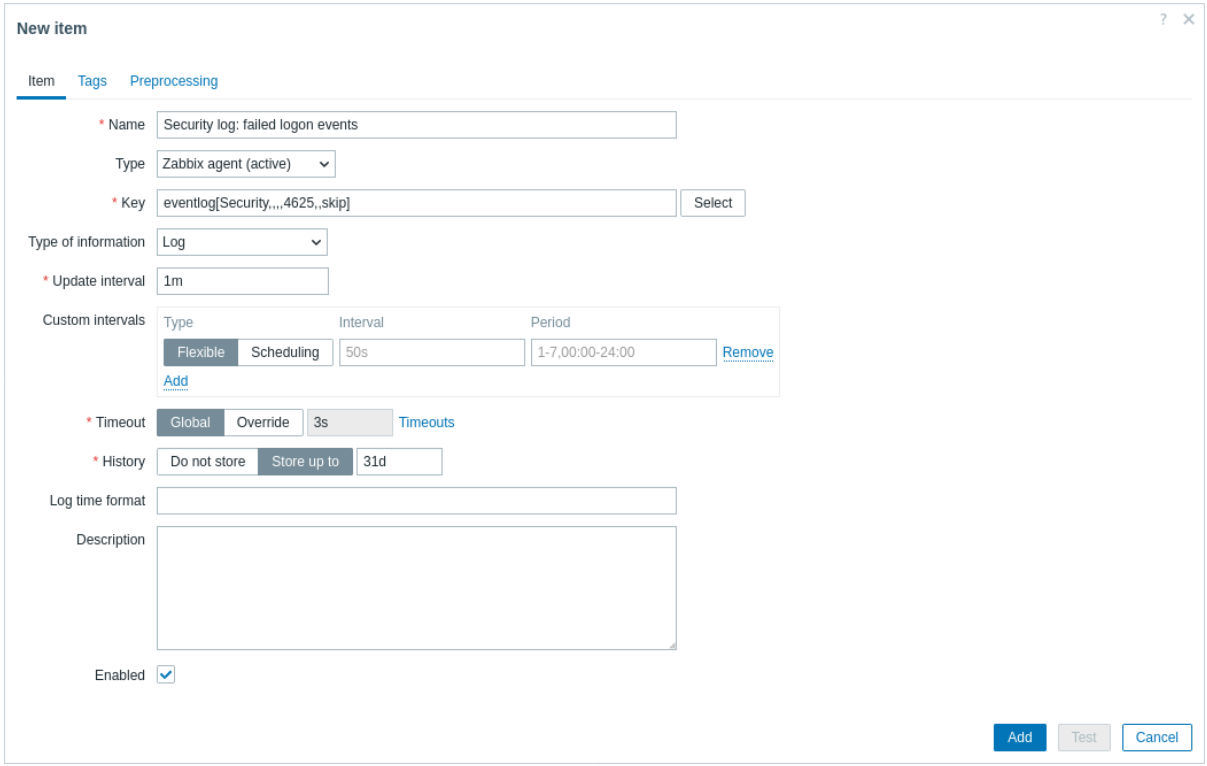
Note:
Laukā *Templates* varat pievienot veidni ”Windows by Zabbix agent active”, lai palīdzētu novērst problēmas, novērojot, vai citi aktīvie vienumi tajā pašā hostā tiek atjaunināti.



2. Izveidojiet jaunu vienumu ar šādiem parametriem:

- Laukā *Name* ievadiet aprakstošu vienuma nosaukumu (piem., ”Security log: failed logon events”).
- Nolaižamajā sarakstā *Type* atlasiet ”Zabbix agent (active)” (nepieciešams Event log monitoring).
- Laukā *Key* izmantojiet eventlog vienuma atslēgu. Piemēram, lai uzraudzītu neveiksmīgus pieteikšanās mēģinājumus (Event ID: 4625) Security žurnālā un ignorētu ierakstus, kas ir vecāki par vienuma pēdējo pārbaudi (izmantojot `skip` parametru), ievadiet šādu vienuma atslēgu: `eventlog[Security,,,,4625,,skip]`

- Nolaižamajā sarakstā *Type of information* atlasiet ”Log”.



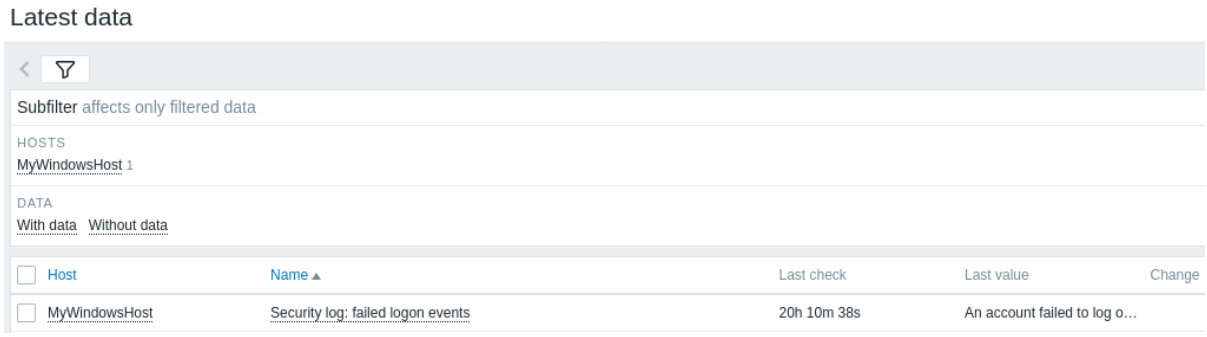
3. Noklikšķiniet uz *Add*, lai saglabātu vienumu.

Pārbaudiet un skatiet savāktos metriskos datus

Apsveicam! Zabbix tagad ir konfigurēts Windows notikumu žurnālu savākšanai. Lai pārbaudītu, vai notikumu žurnāli tiek savākti, varat testēt vienumu ”Security log: failed logon events”, izrakstoties no sava Windows konta un mēģinot pieteikties, izmantojot nepareizus akreditācijas datus.

Pēc tam skatiet savāktos žurnālus Zabbix lietotāja saskarnē:

1. Pārejiet uz *Monitoring* > *Latest data* Zabbix lietotāja saskarnē.



2. Filtrējiet pēc sava ”MyWindowsHost” hosta laukā *Name*.

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 pieejamas, pārejiet uz rokasgrāmatas sadaļu Troubleshooting.

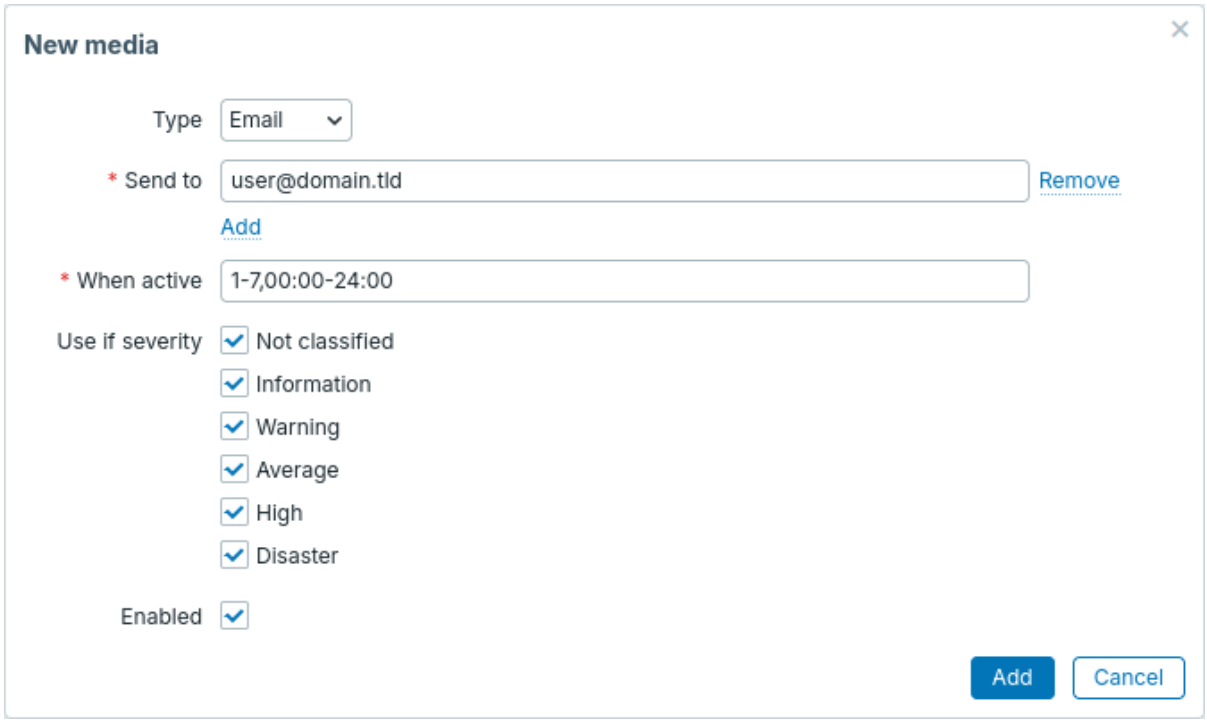
Problēmu brīdinājumu iestatīšana

Šajā rokasgrāmatā ir sniegti pamata konfigurācijas soļi e-pasta brīdinājumu nosūtīšanai.

1. Dodieties uz Datu vākšana > Hosts, lai definētu trigeri, kas aktivizējas, kad jūsu notikumu žurnāla vienums reģistrē jums interesējošo modeli. Piemēram, lai pamanītu neveiksmīgus pieteikšanās mēģinājumus drošības žurnālā, izmantojiet funkciju find():

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

2. Dodieties uz *Lietotāja iestatījumi > Profils*, atveriet cilni *Media* un pievienojiet savu e-pastu.



3. Izpildiet rokasgrāmatu par problēmas paziņojuma saņemšanu.

Nākamreiz, kad Zabbix noteiks problēmu, jums vajadzētu saņemt brīdinājumu pa e-pastu.

Problēmu novēršana

Ja rodas problēmas ar Windows notikumu žurnālu apkopošanu vai skatīšanu, izmantojiet tālāk sniegtos padomus, lai identificētu un novērstu biežāk sastopamās problēmas:

1. Zabbix serverī (Linux) uzskaitiet savus iptables noteikumus ar šādu komandu:

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

un pārliecinieties, ka ir ACCEPT noteikums TCP portam 10051.

2. Pārliecinieties, ka jūsu `eventlog[...]` atslēga izmanto precīzu žurnāla nosaukumu (reģistrjutīgi), notikuma ID, režīmu (piem., skip) un citus parametrus tieši tā, kā parādīts Windows-specific item keys.

Skatiet arī:

- Vienuma izveide - uzziniet, kā pievienot papildu metriku.
- Zabbix aģents uz Microsoft Windows - detalizētas instalēšanas instrukcijas.
- Windows monitorēšana ar Zabbix aģentu - visaptverošs ceļvedis par pamata monitorēšanas iestatīšanu Windows datoriem, izmantojot Zabbix aģentu.
- Windows specifiskās vienumu atslēgas - detalizēta informācija par Windows specifiskajām vienumu atslēgām, ko atbalsta Zabbix aģenti, tostarp tām, kas paredzētas notikumu žurnālu monitorēšanai.
- Žurnālfailu monitorēšana - instrukcijas par Zabbix konfigurēšanu centralizētai žurnālfailu monitorēšanai un analīzei, kas attiecas arī uz Windows notikumu žurnāliem.

Instalēšana: papildu rokasgrāmatas

Ja meklējat vienkāršāko veidu, kā instalēt Zabbix, skatiet Instalēšana un pirmie soļi.

Tālāk sniegtajās rokasgrāmatās aprakstīti **papildu** vai konkrētām sistēmām paredzēti Zabbix vai tā komponentu instalēšanas veidi:

- Instalēšana no konteineriem
- Instalēšana no pirmkoda
- Piezīmes par instalēšanu no pakotnēm
- Zabbix aģenta instalēšana no Windows MSI
- Zabbix aģenta 2 izveide operētājsistēmā Windows
- Zabbix aģenta izveide operētājsistēmā Windows
- macOS aģenta instalēšana no PKG
- Zabbix aģents IBM AIX sistēmā
- Zabbix aģenta izveide operētājsistēmā macOS
- Zabbix appliance
- AWS izvietošanas rokasgrāmata Zabbix serverim
- AWS izvietošanas rokasgrāmata Zabbix starpniekserverim
- Azure izvietošanas rokasgrāmata Zabbix serverim
- Azure izvietošanas rokasgrāmata Zabbix starpniekserverim
- Google Cloud izvietošanas rokasgrāmata

Instalēšana no konteineriem

Pārskats

Šajā lapā ir aprakstīts, kā izvietot Zabbix, izmantojot Docker vai Docker Compose.

Docker Compose ir ātrākais veids, kā izvietot Zabbix. Tas nolasa konfigurācijas failu un automātiski palaiž visus konteinerus, kas veido pilnu Zabbix iestatījumu, pareizajā secībā.

Docker (manuāla izvietošana) sasniedz to pašu rezultātu, izvietojot katru komponentu soli pa solim.

Priekšnosacījumi

Pirms darba sākšanas pārliecinieties, ka jūsu sistēmā ir instalēts Docker (1.12.0 vai jaunāka versija). Ja tas nav instalēts, skatiet Docker instalēšanas rokasgrāmatu. Izvietošanai, izmantojot Docker Compose, ir nepieciešams arī Docker Compose (2.24.0 vai jaunāka versija).

Dažiem Zabbix komponentiem ir nepieciešams atvērt konkrētus portus hosts, kurā darbojas Docker (piemēram, 10051/TCP Zabbix serverim, 162/UDP SNMP trapu saņemšanai). Pilnu Zabbix komponentu izmantoto portu sarakstu skatiet sadaļā Prasības. Zabbix serverim un aģentam noklusējuma portu var mainīt, attiecīgajā konteinerā iestatot vides mainīgo `ZBX_LISTENPORT`.

Pieejamie Docker attēli

Zabbix nodrošina Docker attēlu katram Zabbix komponentam, un tie visi ir publicēti vietnē Docker Hub. Katrs attēls tiek izmantots, lai izveidotu konteineru, kurā darbojas attiecīgais komponents.

Zabbix komponents	Docker attēls
Aģents	zabbix/zabbix-agent
Aģents 2	zabbix/zabbix-agent2
Serveris (MySQL)	zabbix/zabbix-server-mysql
Serveris (PostgreSQL)	zabbix/zabbix-server-pgsql
Tīmekļa saskarne (Apache + MySQL)	zabbix/zabbix-web-apache-mysql
Tīmekļa saskarne (Apache + PostgreSQL)	zabbix/zabbix-web-apache-pgsql
Tīmekļa saskarne (Nginx + MySQL)	zabbix/zabbix-web-nginx-mysql
Tīmekļa saskarne (Nginx + PostgreSQL)	zabbix/zabbix-web-nginx-pgsql
Starpniekserveris (SQLite3)	zabbix/zabbix-proxy-sqlite3
Starpniekserveris (MySQL)	zabbix/zabbix-proxy-mysql
Java gateway	zabbix/zabbix-java-gateway
Tīmekļa pakalpojums	zabbix/zabbix-web-service
SNMP slazdi	zabbix/zabbix-snmptraps
MCP serveris	zabbix/zabbix-mcp-server

Note:
Lai izmantotu SNMP slazdus, SNMP slazdu konteineram ir jābūt kopīgam sējumam ar Zabbix servera vai starpniekservera konteineru (skatiet kā izmantot šo attēlu vietnē Docker Hub un tālāk sniegto piemēru).

Visi Zabbix komponentu attēli vietnē Docker Hub ir balstīti uz atbalstīto operētājsistēmu jaunākajām galvenajām versijām. Šie attēli tiek automātiski atkārtoti izveidoti, kad tiek atjaunināti pamatā esošie OS attēli.

Attēlu birkas

Katrs attēls atbalsta birkas, lai izvēlētos bāzes operētājsistēmu un Zabbix versiju:

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

Atbalstītās `<os>` vērtības:

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

Atbalstītās `<version>` vērtības:

- `latest` - Jaunākā stabilā Zabbix versija Alpine Linux
- `<os>-latest` - Jaunākā stabilā Zabbix versija izvēlētajā OS
- `<os>-trunk` - Jaunākais izstrādes (nightly) būvējums izvēlētajā OS
- `<os>-X.X-latest` - Jaunākais konkrētas Zabbix galvenās versijas Zabbix mazākais laidiens izvēlētajā OS
- `<os>-X.X.*` - Konkrēts Zabbix mazākais laidiens izvēlētajā OS

Piemēri:

```
#### Jaunākais stabilais Zabbix starpniekserveris (SQLite3) Alpine Linux:
docker pull zabbix/zabbix-proxy-sqlite3:latest

#### Jaunākais stabilais Zabbix starpniekserveris (SQLite3) Ubuntu:
docker pull zabbix/zabbix-proxy-sqlite3:ubuntu-latest

#### Jaunākais Zabbix serveris (MySQL) izstrādes (nightly) būvējums Ubuntu:
docker pull zabbix/zabbix-server-mysql:ubuntu-trunk

#### Jaunākais Zabbix servera (MySQL) 8.0 mazākais laidiens Alpine Linux:
docker pull zabbix/zabbix-server-mysql:alpine-8.0-latest

#### Zabbix servera (MySQL) versija 8.0.1 Alpine Linux:
docker pull zabbix/zabbix-server-mysql:alpine-8.0.1
```

Docker Compose

Docker Compose ir ātrākais veids, kā izvietot Zabbix. Tas nolasa konfigurācijas failu (Compose failu) un pārvalda visu iestatīšanas procesu — Docker attēlu lejupielādi, iekšējā tīkla izveidi starp konteineriem, krātuves iestatīšanu, datubāzes inicializēšanu un visu komponentu palaišanu pareizā secībā.

Oficiālo Zabbix Dockerfiles repozitorijs nodrošina gatavus Docker Compose failus un uz `.env` balstītu konfigurācijas sistēmu, kas atbalsta dažādas operētājsistēmas, datubāzu aizmugursistēmas un Zabbix komponentu konfigurāciju.

1. Klonējiet repozitoriju, pārejiet uz to un pārslēdzieties uz versiju 8.0:

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

2. (Neobligāti) Pielāgojiet izvietošanu, izmantojot vides mainīgos. Šo darbību var izlaist, lai izmantotu noklusējuma iestatījumus.

3. Izvietojiet noklusējuma iestatījumus — Zabbix serveri, lietotāja saskarni (kas darbojas ar NGINX) un datubāzes aizmugursistēmu (MySQL vai PostgreSQL), katru savā konteinerā ar Alpine Linux.

```
#### Izmantojot MySQL kā datubāzi:
docker compose -f ./compose.yaml up -d

#### Izmantojot PostgreSQL kā datubāzi:
docker compose -f ./compose_pgsql.yaml up -d
```

Kad konteineri ir palaisti un darbojas (parasti 1–3 minūšu laikā), Zabbix serveris tiek palaists un lietotāja saskarne kļūst pieejama pieteikšanās veikšanai.

> **Note:**
> Izmantojiet `docker compose ps`, lai pārbaudītu konteineru statusu. Visiem konteineriem (izņemot `zabbix-docker-server-db-init-1` konteineru) jābūt statusam `Up`. Ja kādam konteineram tiek rādīts statuss `Exited`, pārbaudiet tā žurnālus, lai atrastu kļūdas, izmantojot `docker logs -f <container-name>`.

Vides mainīgie

Docker Compose darbību un Zabbix komponentu konfigurāciju var pielāgot, izmantojot vides mainīgos.

Compose līmeņa mainīgie (definēti `.env` failā) nosaka, kuri Docker attēli, porti un tīkla IP diapazoni tiek izmantoti.
Šos mainīgos var norādīt rindā pirms komandas `docker compose` vai rediģēt `.env` failu.

Piemēram, tālāk norādītā komanda izvieto pilnīgu Zabbix iestatījumu ar vairākiem konteineriem, izmantojot uz Ubuntu balstītu attēlu (noklusējuma Alpine Linux vietā) un publicējot lietotāja saskarni, izmantojot NGINX ar pielāgotiem HTTP (8282) un HTTPS (8443) portiem:

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

Komponentu līmeņa mainīgie (definēti failos `env_vars/.env_<component>`) nosaka Zabbix komponentu konfigurāciju.
Pirms komandas `docker compose` palaišanas rediģējiet attiecīgo `.env_<component>` failu.

Piemēram, varat pielāgot Zabbix servera pasīvo aptaujātāju skaitu un kešatmiņas iestatījumus, kā arī konfigurēt lietotāja saskarnes laika joslu, rediģējot tālāk norādītos mainīgos:

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

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

Note:
Komponentu līmeņa vides mainīgie atbilst Zabbix komponentu konfigurācijas parametriem, taču izmanto citu nosaukumu stilu (piemēram, ZBX_STARTPOLLERS atbilst StartPollers).
Daži mainīgie ir specifiski Docker, un dažus konfigurācijas parametrus nevar mainīt (piemēram, PIDFile un LogType).
Izmantojot vides mainīgos, skatiet sadaļu *Environment variables* katra komponenta Docker attēla aprakstā Docker Hub.

Sējumi

Docker Compose glabā pastāvīgos datus zbx_env/ direktorijā, kas tiek izveidota līdzās Compose failam. Šī direktorija saglabā komponentu datus starp konteineru restartēšanām un atjauninājumiem.

zbx_env/ saturs ir iepriekš noteikts katra komponenta attēlam. Piemēram:

- zbx_env/etc/ - izmanto Zabbix tīmekļa saskarnes konteiners
- zbx_env/usr/ un zbx_env/var/ - izmanto Zabbix servera konteiners

Papildinformāciju par katru sējumu skatiet sadaļā *Allowed volumes* katra komponenta Docker attēla aprakstā Docker Hub.

Makefile

Zabbix Dockerfiles repozitorijā ir pieejams arī Makefile, kas kalpo kā saīsinājums biežāk izmantotajiem Docker Compose uzdevumiem. Pilnās docker compose komandas vietā varat izmantot īsākas make komandas (izpildiet make help, lai skatītu pieejamās opcijas):

```
#### Noklusējuma izvietošana (Zabbix serveris, tīmekļa saskarne, MySQL — viss Alpine Linux vidē)
make up

#### Pielāgota izvietošana (Zabbix serveris, tīmekļa saskarne ar pielāgotiem NGINX portiem, PostgreSQL —
make up \
  OS=ubuntu \
  DB=pgsql \
  ZABBIX_WEB_NGINX_HTTP_PORT=8282 \
  ZABBIX_WEB_NGINX_HTTPS_PORT=8443
```

Attention:
Apturot vai noņemot konteinerus, vienmēr norādiet to pašu datubāzes tipu, kas tika izmantots izvietošanā (piemēram, make down DB=pgsql).

Pēc noklusējuma make up palaiž tikai minimālo pakalpojumu kopu (Zabbix serveri, tīmekļa saskarni un datubāzi). Tas ļauj saglabāt iestatījumu vieglu un novērš tādu komponentu palaišanu, kas var nebūt nepieciešami. Lai iekļautu papildu komponentus, varat izmantot tālāk norādītos Compose profilus:

```
#### Noklusējuma izvietošana + Zabbix aģents, Java gateway, tīmekļa pakalpojums un SNMP slazdi:
make up COMPOSE_PROFILES=full

#### COMPOSE_PROFILES=full + Zabbix starpniekserveris (MySQL un SQLite3):
make up COMPOSE_PROFILES=all
```

Docker (manuāla izvietošana)

Izmantojiet manuālu izvietošanu, ja vēlaties izvietot Zabbix soli pa solim, palaist atsevišķus komponentus, integrēt tos ar esošu vidi vai izmantot alternatīvu konteineru izpildlaiku, piemēram, Podman.

Piemēram, lai izvietotu Zabbix starpniekserveri, palaidiet šādu komandu:

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

Šī komanda:

- Lejupielādē zabbix/zabbix-proxy-sqlite3:alpine-8.0-latest Docker attēlu.
- Izveido un palaiž zabbix-proxy-sqlite3 konteineru, kas balstīts uz Docker attēlu (ar --init un -d karodziņiem).

- Konfigurē Zabbix starpniekservera `Server` parametru, izmantojot vides mainīgo `ZBX_SERVER_HOST`. Šis parametrs norāda Zabbix servera IP adresi, no kura starpniekserveris iegūs konfigurācijas datus un uz kuru tas nosūtīs savāktos datus. Ņemiet vērā, ka citi mainīgie, kas nepieciešami aktīvam starpniekservera darbības režīmam, izmanto noklusējuma vērtības un tos var neiekļaut.

Note:
Vides mainīgie atbilst Zabbix komponentu konfigurācijas parametriem, izmantojot atšķirīgu nosaukumu veidošanas stilu (piem., `ZBX_SERVER_HOST` atbilst `Server`). Daži mainīgie ir specifiski Docker, un dažus konfigurācijas parametrus nevar mainīt (piem., `PIDFile` un `LogType`). Izmantojot vides mainīgos, lūdzu, skatiet sadaļu *Environment variables* katra komponenta Docker attēla aprakstā Docker Hub.

- Pievieno Docker pārvaldītu krātuves vietu (piem., `/var/lib/docker/volumes/zabbix-proxy-data/`) konteinera `/var/lib/zabbix` direktorijam, lai Zabbix starpniekservera dati tiktu saglabāti pastāvīgi, pat ja konteiners tiek noņemts.

Note:
Lai iegūtu vairāk informācijas par katru sējumu, skatiet sadaļu *Allowed volumes* katra komponenta Docker attēla aprakstā Docker Hub.

Pēc Zabbix starpniekservera konteinera izvietošanas varat turpināt ar starpniekservera pievienošanu savā Zabbix lietotāja saskarnē un hosts konfigurēšanu, lai tos uzraudzītu šis starpniekserveris.

Tālāk redzamie piemēri aptver trīs papildu izvietošanas scenārijus:

- Zabbix serveris (MySQL) ar Java gateway
- Zabbix serveris (PostgreSQL) ar SNMP trapiem
- Zabbix serveris (MySQL) ar Java gateway uz RHEL 8-10

Lai iegūtu vairāk piemēru, lūdzu, skatiet katra komponenta Docker attēla aprakstu Docker Hub.

Zabbix servera (MySQL) izvietošana ar Java gateway

Šajā piemērā parādīts, kā izvietot Zabbix serveri ar MySQL aizmugursistēmu, Java gateway JMX uzraudzībai un NGINX bāzētu tīmekļa saskarni.

1. Izveidojiet īpašu Docker tīklu, lai visi Zabbix komponentu konteineri varētu sasniegt cits citu pēc konteineru nosaukumiem:

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

2. Palaidiet tukšu MySQL servera konteineru (aizstājiet `zabbix_pwd` un `root_pwd` ar drošām parolēm un izmantojiet šīs vērtības turpmākajās darbībās):

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

3. Sekojiet MySQL žurnāliem un gaidiet, līdz MySQL ir gatavs pieņemt savienojumus (izmantojiet `Ctrl+C`, lai izietu no žurnāliem):

```
docker logs -f mysql-server

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

4. Palaidiet Zabbix Java gateway konteineru:

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

5. MySQL servera konteinerā iespējojiet `log_bin_trust_function_creators`. Tas īslaicīgi atvieglo drošības ierobežojumus saglabātajām funkcijām; turpmākajās darbībās šis iestatījums tiks atspējots. Tas ļauj Zabbix datubāzes lietotājam izveidot saglabātās funkcijas bez `SUPER` privilēģijas, kas MySQL ir nepieciešama, ja ir iespējota binārā reģistrēšana (pēc noklusējuma kopš MySQL 8.0). Izpildiet šo komandu:

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

6. Palaidiet Zabbix servera konteineru un savienojiet to ar Java Gateway un MySQL servera konteineriem:

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

7. Sekojiet Zabbix servera žurnāliem un gaidiet, līdz Zabbix serveris pabeidz datubāzes shēmas inicializāciju (izmantojiet `Ctrl+C`, lai izietu no žurnāliem):

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

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

8. MySQL servera konteinerā atspējojiet `log_bin_trust_function_creators`. Tādējādi tiek atkārtoti ieviests drošības ierobežojums, kas neļauj lietotājiem bez SUPER privilēģijas izveidot saglabātās funkcijas. Izpildiet šo komandu:

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

9. Palaidiet Zabbix tīmekļa saskarnes konteineru un savienojiet to ar Zabbix servera un MySQL servera konteineriem:

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

Kad konteineri ir palaisti un darbojas, Zabbix serveris tiek startēts un tīmekļa saskarne kļūst pieejama pieteikšanās veikšanai.

> **Note:**
> Izmantojiet `docker ps`, lai pārbaudītu konteineru statusu. Visiem konteineriem jābūt statusam `Up`. Ja kādam konteineram tiek rādīts statuss `Exited`, pārbaudiet tā žurnālus, izmantojot `docker logs -f <container-name>`.

Zabbix servera (PostgreSQL) izvietošana ar SNMP trapēm

Šajā piemērā parādīts, kā izvietot Zabbix serveri ar PostgreSQL aizmugursistēmu, SNMP trapēm un NGINX bāzētu tīmekļa saskarni.

1. Izveidojiet īpašu Docker tīklu, lai visi Zabbix komponentu konteineri varētu sasniegt cits citu pēc konteineru nosaukumiem:

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

2. Palaidiet tukšu PostgreSQL servera konteineru (aizstājiet `zabbix_pwd` ar drošu paroli un izmantojiet šo vērtību turpmākajās darbībās):

```
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 SNMP trašu konteineru:

```
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 konteineru un savienojiet to ar PostgreSQL servera un SNMP trašu konteineriem:

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

5. Sekojiet Zabbix servera žurnāliem un gaidiet, līdz Zabbix serveris pabeidz datubāzes shēmas inicializāciju (izmantojiet `Ctrl+C`, lai izietu no žurnāliem):

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

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

6. Palaidiet Zabbix tīmekļa saskarnes konteineru un savienojiet to ar Zabbix servera un PostgreSQL servera konteineriem:

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

Kad konteineri ir palaisti un darbojas, Zabbix serveris tiek startēts un tīmekļa saskarne kļūst pieejama pieteikšanās veikšanai.

> **Note:**
> Izmantojiet `docker ps`, lai pārbaudītu konteineru statusu. Visiem konteineriem jābūt statusam `Up`. Ja kādam konteineram tiek rādīts statuss `Exited`, pārbaudiet tā žurnālus, izmantojot `docker logs -f <container-name>`.

Zabbix servera (MySQL) izvietošana ar Java gateway operētājsistēmā RHEL 8–10

Šajā piemērā parādīts, kā izvietot Zabbix serveri ar MySQL aizmugursistēmu, Java gateway JMX uzraudzībai un NGINX bāzētu tīmekļa saskarni, kas darbojas Red Hat Enterprise Linux 8, 9 vai 10.

Red Hat Enterprise Linux vidē ieteicamais konteineru izpildlaiks ir Podman, nevis Docker. Podman darbojas līdzīgi Docker, taču tam nav nepieciešams fonā darbojošs pakalpojums ar root tiesībām, tāpēc tas ir piemērotāks Red Hat vidēm.

1. Izveidojiet jaunu pod ar nosaukumu `zabbix` un atveriet portus Zabbix tīmekļa saskarnei un Zabbix servera trapperim:

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

2. (Neobligāti) Palaidiet Zabbix aģenta konteineru poda `zabbix` 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-80:latest
```

3. RHEL hosts izveidojiet direktoriju `./mysql/`:

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

4. Palaidiet tukšu MySQL servera konteineru (aizstājiet `zabbix_pwd` un `root_pwd` ar drošām parolēm un izmantojiet šīs vērtības turpmākajās darbībās):

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

5. Sekojiet MySQL žurnāliem un gaidiet, līdz MySQL ir gatavs pieņemt savienojumus (izmantojiet `Ctrl+C`, lai izietu no žurnāliem):

```
podman logs -f mysql-server

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

6. 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-80
```

7. MySQL servera konteinerā iespējojiet `log_bin_trust_function_creators`. Tas īslaicīgi atvieglo drošības ierobežojumus saglabātajām funkcijām; turpmākajās darbībās šis iestatījums tiks atspējots. Tas ļauj Zabbix datubāzes lietotājam izveidot saglabātās funkcijas bez `SUPER` privilēģijas, kas MySQL ir nepieciešama, ja ir iespējota bināra reģistrēšana (pēc noklusējuma kopš MySQL 8.0). Izpildiet šo komandu:

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

8. Palaidiet Zabbix servera konteineru un savienojiet to ar Java Gateway un MySQL servera konteineriem:

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

9. Sekojiet Zabbix servera žurnāliem un gaidiet, līdz Zabbix serveris pabeidz datubāzes shēmas inicializāciju (izmantojiet `Ctrl+C`, lai izietu no žurnāliem):

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

#### ... [info]: ** Creating 'zabbix' schema in MySQL
```

```
#### ... [info]: ** Database schema successfully created!
```

10. MySQL servera konteinerā atspējojiet `log_bin_trust_function_creators`. Tādējādi tiek atkārtoti ieviests drošības ierobežojums, kas neļauj lietotājiem bez SUPER privilēģijas izveidot saglabātās funkcijas. Izpildiet šo komandu:

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

11. Palaidiet Zabbix tīmekļa saskarnes konteineru un savienojiet to ar Zabbix servera un MySQL servera konteineriem:

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

Note:
Pods `zabbix` resursdatordatoram atver 80/TCP portu (HTTP), izmantojot 8080/TCP portu no `zabbix-web-mysql` konteinera.

Kad konteineri ir palaisti un darbojas, Zabbix serveris tiek startēts un tīmekļa saskarne kļūst pieejama pieteikšanās veikšanai.

Note:
Izmantojiet `docker ps`, lai pārbaudītu konteineru statusu. Visiem konteineriem jābūt statusam `Up`. Ja kādam konteineram tiek rādīts statuss `Exited`, pārbaudiet tā žurnālus, izmantojot `docker logs -f <container-name>`.

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 avotus

Zabbix servera, Zabbix starpniekservera vai Zabbix aģenta izveidei 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 jāizpilda:
> `./bootstrap.sh`

Konfigurējot Zabbix servera vai starpniekservera avotus, jānorāda izmantojamais datubāzes tips. Vienlaikus ar servera vai starpniekservera procesu var kompilēt tikai vienu datubāzes tipu.

Lai skatītu visas atbalstītās konfigurācijas opcijas, izvilktajā Zabbix avota direktorijā izpildiet:

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

Lai konfigurētu Zabbix servera un aģenta avotus, varat izpildīt, piemēram:

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

Lai konfigurētu Zabbix servera avotus (ar PostgreSQL u.c.), varat izpildīt:

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

Lai konfigurētu Zabbix starpniekservera avotus (ar SQLite u.c.), varat izpildīt:

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

Lai konfigurētu Zabbix aģenta avotus, varat izpildīt:

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

vai Zabbix aģentam 2:

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

Note:
Lai izveidotu Zabbix aģentu 2, jābūt instalētai atbalstītai Go versijai.

Piezīmes par kompilēšanas opcijām:

- `--enable-agent` - kompilē Zabbix aģentu, kā arī Zabbix get un Zabbix sender komandrindas utilītprogrammas.
- `--with-libcurl` - nepieciešams virtuālo mašīnu uzraudzībai, SMTP autentifikācijai un `web.page.*` Zabbix aģenta vienumiem, kā arī telemetrijas vaicājuma 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 (netiek atbalstīts Solaris). Izmantojiet šo opciju, ja plānojat izplatīt kompilētos bināros failus sistēmās, kurās nav nepieciešamo bibliotēku. Veidojot Zabbix serveri, šī opcija nav ieteicama. Lai serveri izveidotu statiski, ir nepieciešama katras ārējās bibliotēkas statiskā versija. Konfigurēšanas skripts to automātiski nepārbauda.
- `--with-stacksize=<value>` - iestata katra pavediena steka lielumu kilobaitos (piemēram, `--with-stacksize=512`). Šo vērtību var palielināt, ja Zabbix avārijas dēļ pārtrauc darbu vai sastingst steka pārpildes dēļ (piemēram, priekšapstrādes laikā sistēmās ar maziem noklusējuma pavedienu steka ierobežojumiem).
- `--with-apm` (izmantojiet tikai kopā ar `--enable-proxy`) - nepieciešams lietojumprogrammu veiktspējas uzraudzībai (APM) Zabbix starpniekserverī. Skatiet arī: Prasības (g++, gRPC, protobuf, Abseil).

Attention:
Ja `./configure` neizdodas trūkstošu bibliotēku vai citu problēmu dēļ, detalizētu informāciju par kļūdu skatiet failā `config.log`.
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 faktisko cēloni:
```
/usr/bin/ld: cannot find -lssl
/usr/bin/ld: cannot find -lcrypto
```

Skatiet arī:

- Zabbix kompilēšana ar šifrēšanas atbalstu
- Zināmās kompilēšanas problēmas

5 Izveidojiet un instalējiet visu

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

`make install`

Šo soli jāizpilda lietotājam ar pietiekamām tiesībām (parasti `root` vai izmantojot `sudo`).

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

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

6 Konfigurācijas failu pārskatīšana un rediģēšana

- rediģējiet Zabbix aģenta konfigurācijas failu (`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 noraidīti.

- rediģējiet Zabbix servera konfigurācijas failu (`zabbix_server.conf`).

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

Pārējie parametri ar noklusējuma vērtībām būs piemēroti, ja jums ir neliela instalācija (līdz desmit uzraudzītiem hostiem). Tomēr, ja vēlaties maksimāli uzlabot Zabbix servera (vai starpniekservera) veiktspēju, noklusējuma parametri ir jāmaina.

- ja esat instalējis Zabbix starpniekserveri, rediģējiet starpniekservera konfigurācijas failu (`zabbix_proxy.conf`).

Jānorāda servera IP adrese un starpniekservera resursdatora nosaukums (serverim tam ir jābūt zināmam), 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 palaišanai ir nepieciešams tīmekļa serveris ar PHP atbalstu. Instalēšana tiek veikta, vienkārši kopējot PHP failus no ui direktorijas uz tīmekļa servera HTML dokumentu direktoriju.

Biežāk sastopamās HTML dokumentu direktoriju atrašanās vietas Apache tīmekļa serveriem ir šādas:

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

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

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

Ja plānojat izmantot 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 aģents 2 ielādējamo spraudņu instalēšana

Zabbix aģents 2 ielādējamo spraudņu instalēšana ir nepieciešama tikai tad, ja vēlaties uzraudzīt mērķus, kurus neaptver iebūvētie spraudņi (piem., 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 norādītas specifiskas prasības un instalēšanas instrukcijas.

Lai instalētu no avota kodiem, vispirms lejupielādējiet un izvelciet ielādējamā spraudņa avota arhīvu.

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

```
make
```

> **Note:**
> Lai izveidotu Zabbix aģents 2 ielādējamos spraudņus, ir jābūt instalētai atbalstītai Go versijai.

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

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

Spraudņa konfigurācijas faila ceļš ir jānorāda Zabbix aģents 2 konfigurācijas faila parametrā `Include`:

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

Lai iegūtu plašāku informāciju par spraudņu konfigurēšanu, pārejiet uz iestatīšanu.

Zabbix nodrošinātie ielādējamie spraudņi izmanto vienkāršus makefile failus ar šādiem būves mērķiem:

- `make` - izveido spraudni
- `make clean` - dzēš visus failus, kas izveidoti, būvējot spraudni
- `make check` - palaiž pašpārbaudes (nepieciešams reāls uzraudzības mērķis, piemēram, PostgreSQL datubāze)
- `make style` - pārbauda Go koda stilu ar `golangci-lint`
- `make format` - formatē Go kodu ar `go fmt`
- `make dist` - izveido avota 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 datubāze.

Lai instalētu no avotiem, vispirms lejupielādējiet un izsaiņojiet avotu 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 iepakotu Java gateway JAR failā, palaidiet `make`. Ņemiet vērā, ka šim solim jūsu ceļā būs nepieciešami `javac` un `jar` izpildāmie faili.

```
make
```

Tagad jums ir zabbix-java-gateway-$VERSION.jar fails `src/zabbix_java/bin`. Ja jums ir ērti palaist Java gateway no `src/zabbix_java` izplatīšanas direktorijā, tad varat turpināt ar instrukcijām 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 web service instalēšana

Zabbix web service instalēšana ir nepieciešama tikai tad, ja vēlaties izmantot ieplānotos pārskatus.

Lai instalētu no avotiem, vispirms lejupielādējiet un izsaiņojiet avotu arhīvu.

Lai kompilētu Zabbix web service, palaidiet `./configure` skriptu ar `--enable-webservice` opciju.

> **Note:**
> Lai izveidotu Zabbix web service, ir jābūt instalētai atbalstītai Go versijai.

Palaidiet zabbix_web_service datorā, kur ir instalēts web service:

```
zabbix_web_service
```

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

Zabbix MCP servera instalēšana

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

Lai kompilētu Zabbix MCP serveri, MCP servera pirmkoda direktorijā izpildiet šo komandu:

```
make build
```

> **Note:**
> Lai izveidotu Zabbix MCP serveri, jābūt instalētai atbalstītai Go versijai.

Lai `zabbix_mcp_server -V` izvadē un Zabbix API pieprasījumu `User-Agent` norādītu konkrētu versiju, iestatiet mainīgo `VERSION`, piemēram, `make build VERSION=8.0.0`.

Lai instalētu bināro failu direktorijā `/usr/local/sbin` un konfigurācijas faila piemēru direktorijā `/etc/zabbix`, izpildiet:

```
sudo make install
```

Rediģējiet konfigurācijas failu `/etc/zabbix/zabbix_mcp_server.conf` un iestatiet vismaz šādus parametrus:

- `FrontendURL` - Zabbix API URL, kas beidzas ar `api_jsonrpc.php`;
- `AllowedIP` - to MCP klientu adreses, kuriem ir atļauts izveidot savienojumu.

Lai šifrētu savienojumus no MCP klientiem, iestatiet `TLSAccept=cert` un norādiet `TLSCertFile` un `TLSKeyFile`, vai palaidiet MCP serveri aiz uzticama reversā starpniekservera, kas apstrādā TLS.

Ja `zabbix` lietotājs nepastāv (piemēram, ja datorā ir instalēts tikai MCP serveris), izveidojiet to, kā aprakstīts sadaļā Lietotāja konta izveide.

Validējiet konfigurācijas failu un palaidiet zabbix_mcp_server:

```
zabbix_mcp_server -c /etc/zabbix/zabbix_mcp_server.conf -T
zabbix_mcp_server -c /etc/zabbix/zabbix_mcp_server.conf
```

Lai iegūtu papildinformāciju par servera konfigurēšanu un MCP klientu savienošanu, skatiet sadaļu MCP serveris.

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).

Steka izmērs dažās sistēmās ir pārāk mazs

Ja Zabbix avarē vai sasalst steka pārpildes dēļ, varat palielināt katra pavediena steka izmēru, konfigurējot avotus, izmantojot `--with-stacksize` opciju avotu konfigurēšanas laikā. Šī problēma var rasties sistēmās ar zemu noklusējuma pavedienu steka limitu, īpaši priekšapstrādes laikā, kad tiek izveidoti vairāki pavedieni.

Tālāk redzamais piemērs iestata steka izmēru uz 512 KB katram pavedienam:

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

Sistēmas pavedienu steka limitus izpildes laikā var pārbaudīt, Linux bāzētās sistēmās izmantojot komandu `ulimit -s`.

Piezīmes par instalēšanu no pakotnēm

Pārskats

Oficiālās Zabbix 8.0 **pirmsizlaides** pakotnes ir pieejamas Zabbix tīmekļa vietnē, kur varat atlasīt savu operētājsistēmu un Zabbix komponentu, lai ģenerētu instalēšanas norādījumus un komandas.

Papildus tālāk sniegtajām pakotņu instalēšanas piezīmēm skatiet arī lapu Instalēšana un iestatīšana.

Pakotnes ir pieejamas šādiem Linux distributīviem:

- 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:**
> Daži OS distributīvi (jo īpaši Debian bāzētie distributīvi) nodrošina savas Zabbix pakotnes. Šīs pakotnes Zabbix neatbalsta, un tās var būt novecojušas vai arī tajās var nebūt jaunāko funkciju un kļūdu labojumu. Ieteicams izmantot tikai oficiālās pakotnes no Zabbix oficiālā repozitorija. Ja iepriekš instalējāt Zabbix no savas operētājsistēmas repozitorija, skatiet darbības, kas jāveic, lai jauninātu Zabbix pakotnes no OS repozitorijiem.

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

Ja nepieciešams, Zabbix oficiālajā repozitorijā ir pieejamas atsevišķas pakotnes Zabbix aģentam/aģentam 2, Zabbix get un Zabbix sender.

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

- Windows aģenta instalēšana no MSI
- macOS aģenta instalēšana no PKG
- Zabbix aģents IBM AIX sistēmā
- Mantotie binārie faili (vecākām vai mazāk izplatītām sistēmām, piemēram, HP-UX, NetBSD, Tru64 un vecākām SLES versijām)

Piezīmes par pakotņu instalēšanu

Tālāk norādītās piezīmes attiecas uz visām sistēmām:

- Ja izmantojat PostgreSQL, `DBHost=localhost` (vai IP adreses) norādīšana Zabbix servera/starpniekservera konfigurācijā liek PostgreSQL izmantot tīkla ligzdu, nevis lokālo UNIX ligzdu; saistītos iestatīšanas norādījumus skatiet sadaļā SELinux konfigurēšana.
- Ja izmantojat TimescaleDB, skatiet papildu TimescaleDB iestatīšanas norādījumus.
- Ja instalējat Zabbix Java gateway (JMX lietojumprogrammu uzraudzībai), skatiet papildu iestatīšanas norādījumus RHEL bāzētām sistēmām un Debian bāzētām sistēmām.
- Ja instalējat Zabbix MCP serveri, instalējiet `zabbix-mcp-server` pakotni un pirms pakalpojuma palaišanas atjauniniet konfigurācijas failu (vismaz `FrontendURL` un `AllowedIP`).
- Lai palaistu Zabbix aģentu kā root lietotāju, skatiet Aģenta palaišana kā root lietotājam.

Tālāk norādītās piezīmes attiecas uz RHEL un tā atvasinājumiem:

- Ja esat iespējojis EPEL repozitoriju EL9, kas arī nodrošina Zabbix pakotnes, tas ir jāizslēdz no pakotņu atrisināšanas pirms oficiālo Zabbix pakotņu instalēšanas; skatiet Nejauša EPEL Zabbix pakotņu instalēšana.
- Informāciju par Zabbix pakotņu instalēšanu Red Hat UBI vidēs skatiet sadaļā Zabbix pakotnes RHEL vajadzībām 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 komunikāciju. Sistēmās, kur ir iespējots Security-Enhanced Linux (SELinux), var būt nepieciešams pievienot SELinux noteikumus, lai atļautu Zabbix izveidot/izmantot UNIX domēna ligzdas direktorijā SocketDir. Ligzdu faili tiek izmantoti Zabbix serveris (alerter, preprocessing, IPMI) un Zabbix starpniekserveris (IPMI), un tie ir pieejami, 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 datubāze tiek piekļūta pa tīklu (tostarp `localhost` PostgreSQL gadījumā), atļaujiet arī Zabbix lietotāja saskarne izveidot savienojumu ar datubāzi:

```
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 restartējiet Apache:

```
systemctl restart httpd
```

Pēc izvēles varat instalēt iepriekš definētu `zabbix-selinux-policy` pakotni no Zabbix Official Repository. Šī pakotne ir pieejama visām atbalstītajām OS versijām, lai vienkāršotu Zabbix izvietošanu un novērstu situācijas, kad lietotāji izslēdz SELinux konfigurācijas sarežģītības dēļ.
 AlmaLinux, CentOS Stream, RHEL, Rocky Linux un Oracle Linux 10. versijai SELinux politika tiek nodrošināta ar pakotni `zabbix-selinux-policy-main`; pakotnei `zabbix-selinux-policy` ir atkarība no `zabbix-selinux-policy-main` vai `selinux-policy-targeted-extra`. Pakotne `zabbix-selinux-policy-main` konfliktē ar `selinux-policy-targeted-extra`; tādēļ abas pakotnes nevar instalēt vienlaikus.

Attention:
Lai nodrošinātu maksimālu drošību, ieteicams iestatīt pielāgotus SELinux iestatījumus.

Pakotne `zabbix-selinux-policy` (`zabbix-selinux-policy-main` AlmaLinux, CentOS Stream, RHEL, Rocky Linux un Oracle Linux 10. versijai) satur pamata SELinux politiku, kas ļauj Zabbix izveidot un izmantot ligzdas, kā arī iespējo HTTPd savienojumu ar PostgreSQL (ko 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.

Lūdzu, ņemiet vērā, ka Debuginfo pakotnes ir pieejamas šādām versijām: Red Hat Enterprise Linux 7 un vecākām (tostarp atvasinājumiem), kā arī SUSE Linux Enterprise Server 15 un vecākām (tostarp atvasinājumiem).

Lai iespējotu zabbix-debuginfo repozitoriju:

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

```
[zabbix-debuginfo]
name=Zabbix Official Repository debuginfo - $basearch
baseurl=http://repo.zabbix.com/zabbix/8.0/stable/rhel/7/$basearch/debuginfo/
enabled=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-ZABBIX-A14FE591
gpgcheck=1
```

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

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

Kad tas ir iespējots, instalējiet pakotnes:

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

```
dnf install zabbix-debuginfo
```

- SUSE sistēmā instalējiet komponentam specifiskas debuginfo pakotnes:

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

Nestabilu laidienu repozitoriju iespējošana

Tālāk sniegtās instrukcijas ir paredzētas nestabilu Zabbix laidiena repozitoriju iespējošanai (pēc noklusējuma atspējoti), kas tiek izmantoti Zabbix mazāko versiju laidiena kandidātiem.

Vispirms instalējiet vai atjauniniet uz jaunāko zabbix-release pakotni. Lai savā sistēmā iespējotu rc pakotnes, veiciet tālāk norādītās darbības:

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.

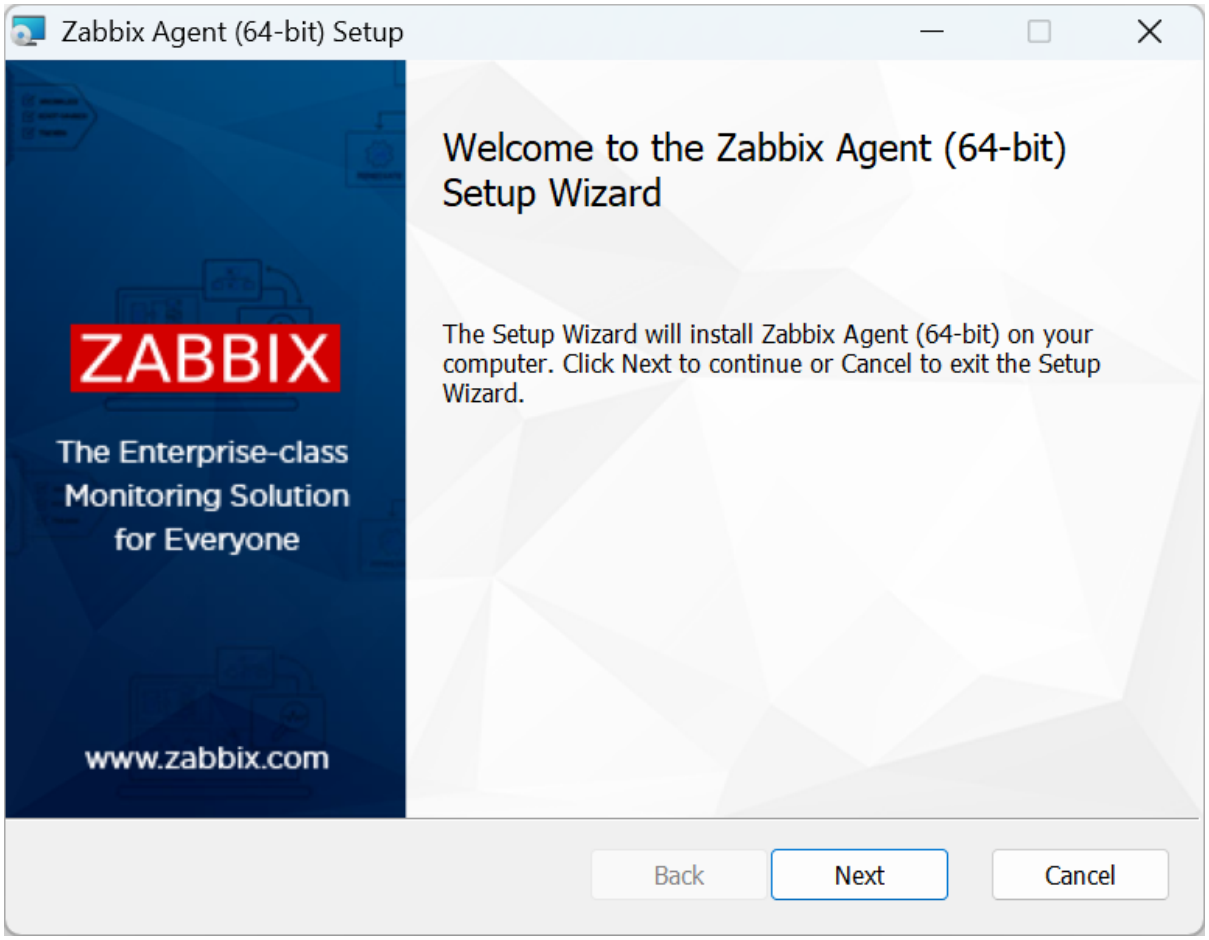
Lai gan instalēšana, izmantojot MSI pakotnes, tiek pilnībā atbalstīta, pareizai kļūdu apstrādei ieteicams instalēt vismaz Microsoft .NET Framework 2.

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

Instalēšana, izmantojot iestatīšanas vedni

Tālāk aprakstītās instalēšanas darbības attiecas gan uz Zabbix aģentu, gan uz Zabbix aģentu 2.

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

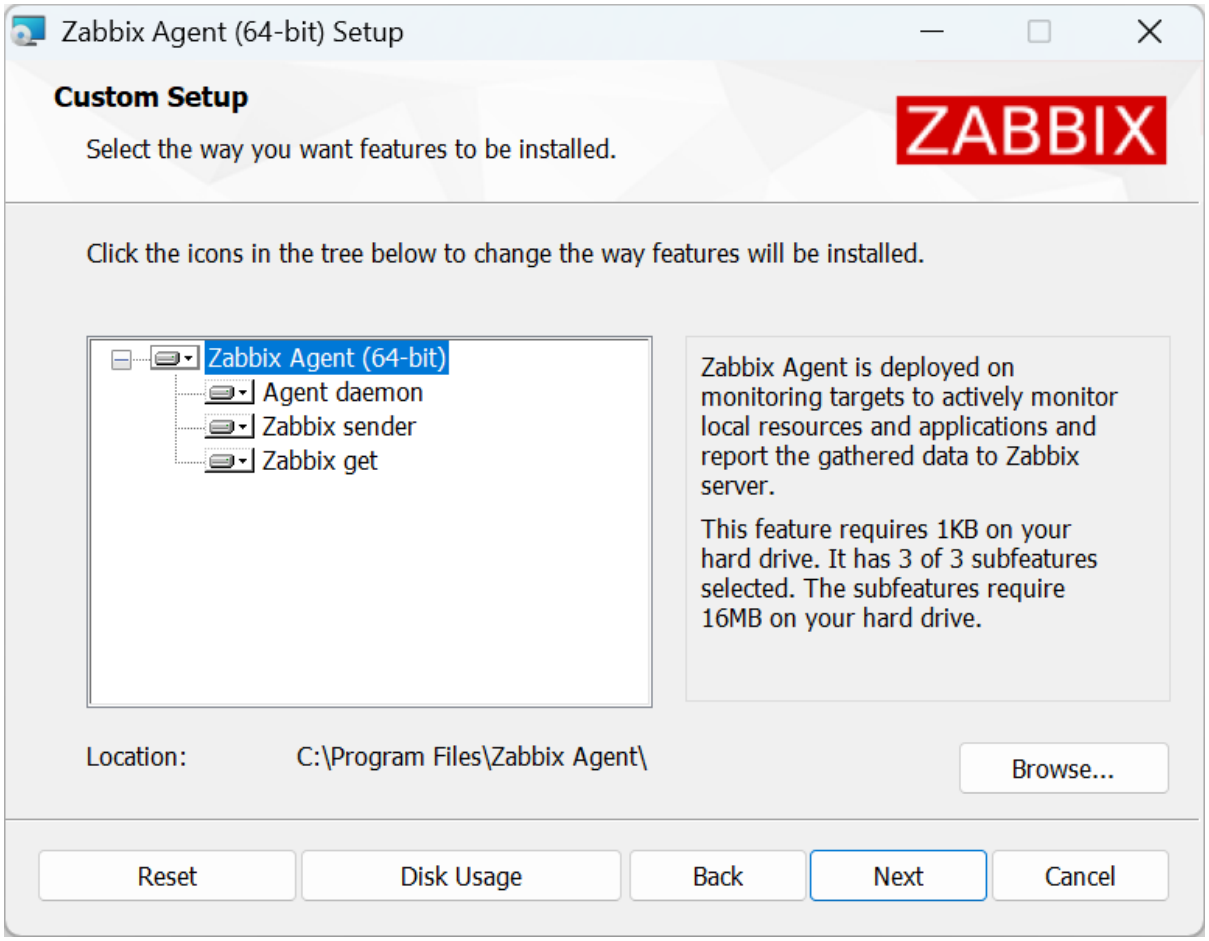


2. Pieņemiet galalietotāja licences līgumu:



3. Atlasiet instalējamos Zabbix komponentus (aģenta dēmonu, Zabbix sender, Zabbix get):

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

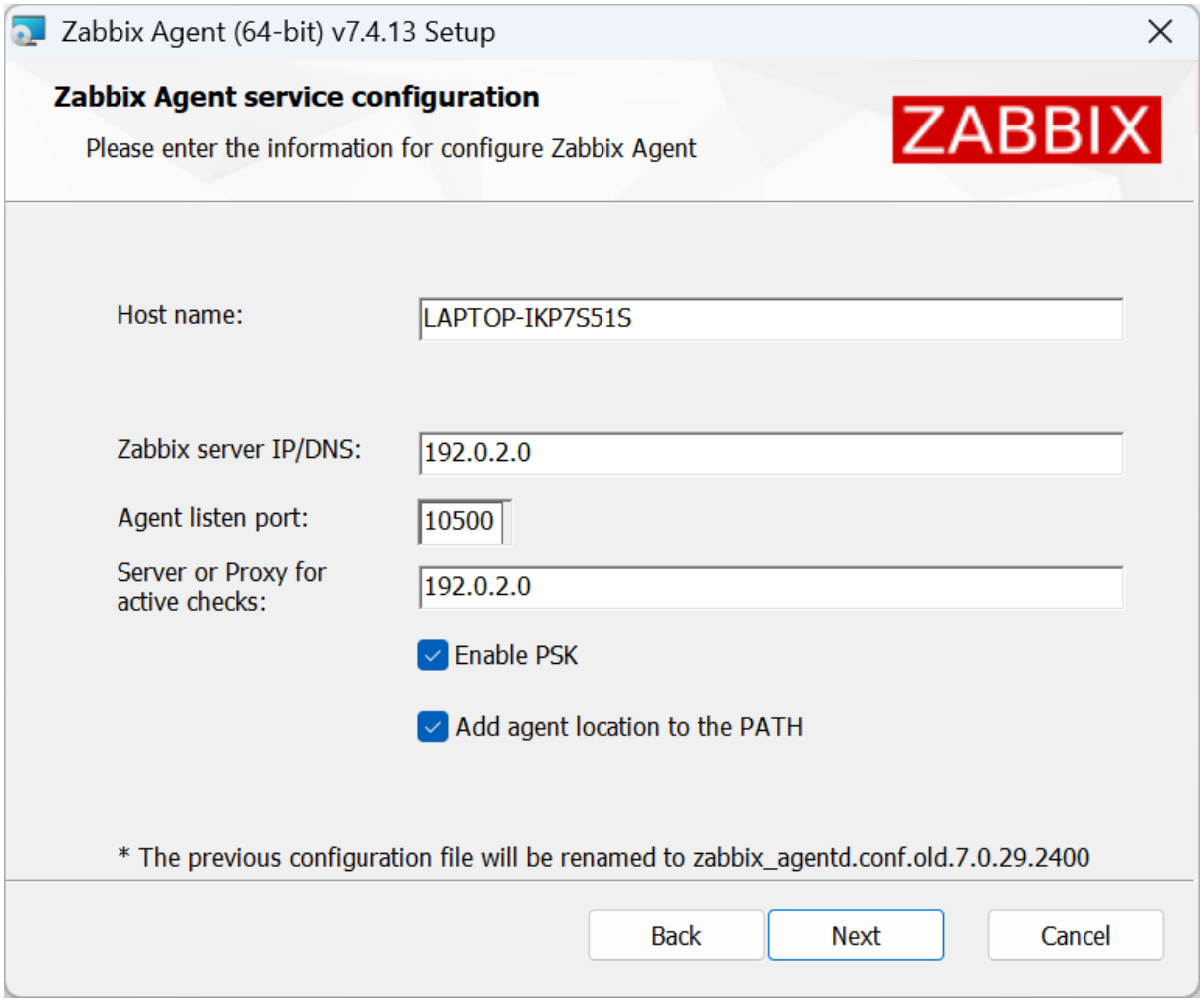


Note:
Ja tiek izmantota pielāgota atrašanās vieta ārpus `Program Files` direktorijas un mērķa mape jau pastāv, instalētājs nemainīs tās atļaujas. Jaunas instalēšanas laikā jums tiks piedāvāts manuāli iestatīt atbilstošās atļaujas vai turpināt, tās nemainot.

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

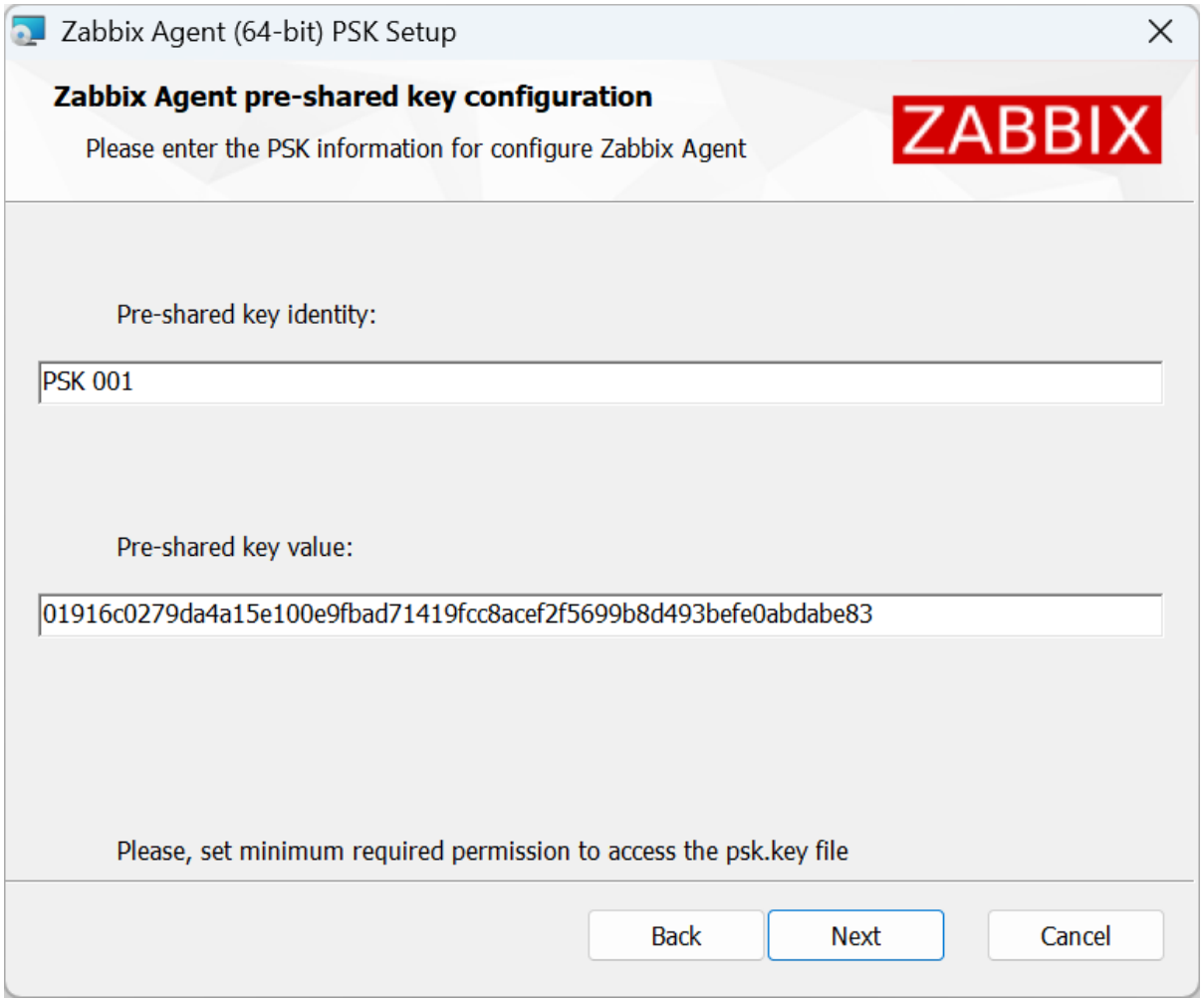
Parameter	Description
Host name	Tās mašīnas hosts, kurā tiek instalēts Zabbix aģents. Iestata parametru Hostname.
Zabbix server IP/DNS	Ar komatiem atdalītu IP adrešu saraksts, pēc izvēles CIDR apzīmējumā, vai Zabbix serveru vai Zabbix starpniekserveru DNS nosaukumi. Šis parametrs ir **obligāts**. Iestata parametru Server.
Agent listen port	Aģents šajā portā gaidīs savienojumus 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 pēc izvēles ports, kas atdalīts ar kolu. Iestata parametru ServerActive.
Enable PSK	Atzīmējiet izvēles rūtiņu, lai iespējotu TLS atbalstu izmantojot iepriekš koplietotas atslēgas. Iestata parametru TLSConnect un TLSAccept vērtību `psk`.
Add agent location to the PATH	Atzīmējiet izvēles rūtiņu, lai pievienotu Zabbix aģenta atrašanās vietu sistēmas PATH mainīgajam.

Note:
Ja tiek atrasts esošs Zabbix aģents, tiks parādītas tā konfigurācijas faila parametru vērtības. 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 tiks iestatīti arī Zabbix aģenta 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`, kurā ir atslēga, un iestata parametra TLSPSKFile vērtību uz atslēgas atrašanās vietu (noklusējums: `C:\Program Files\Zabbix Agent\psk.key`). **Ieteicams** ierobežot piekļuvi iepriekš koplietotās atslēgas failam, pielāgojot faila drošības iestatījumus tā, lai to varētu lasīt tikai Zabbix aģents (vai lietotājs, kas palaiž aģentu).

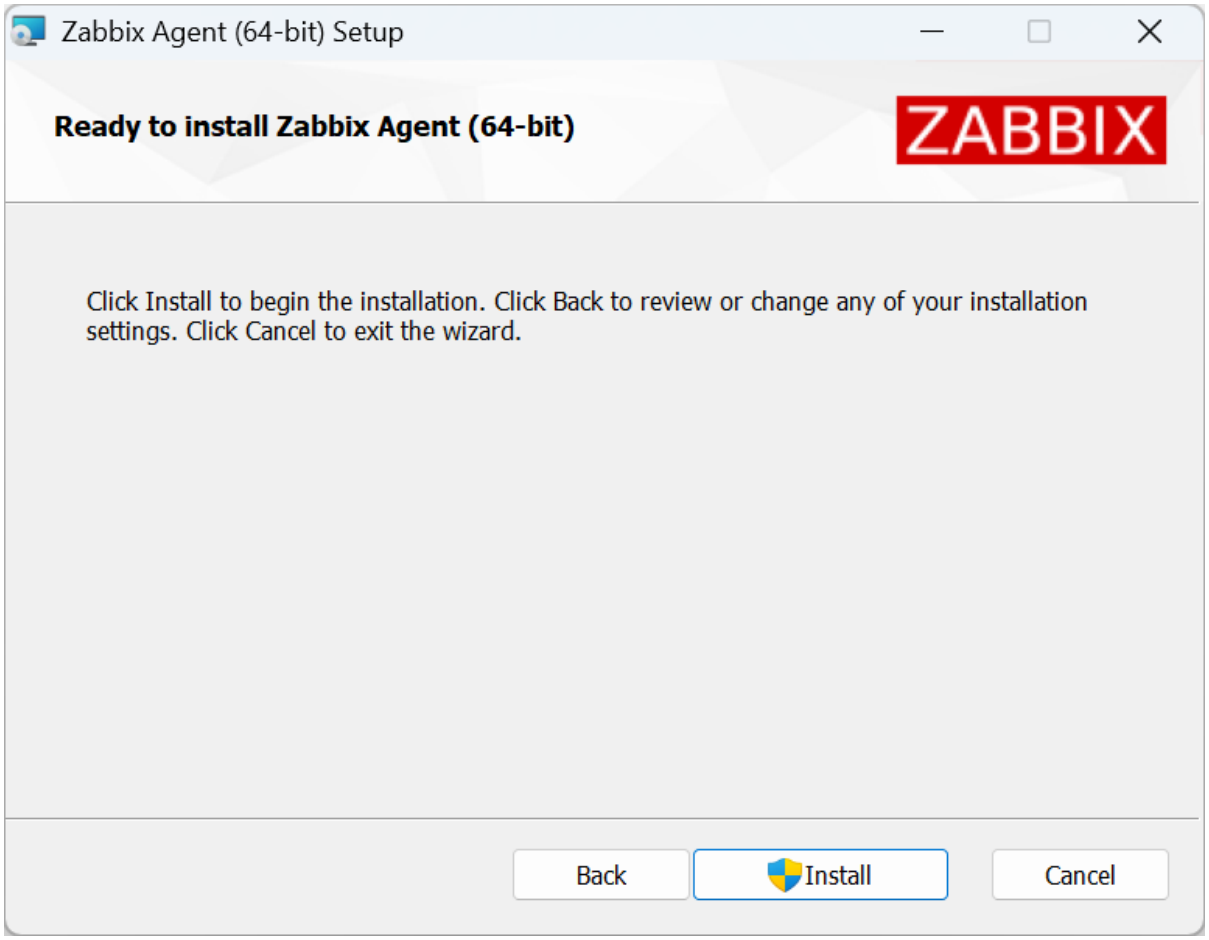


6. Noklikšķiniet uz *Install*, lai sāktu instalēšanu.

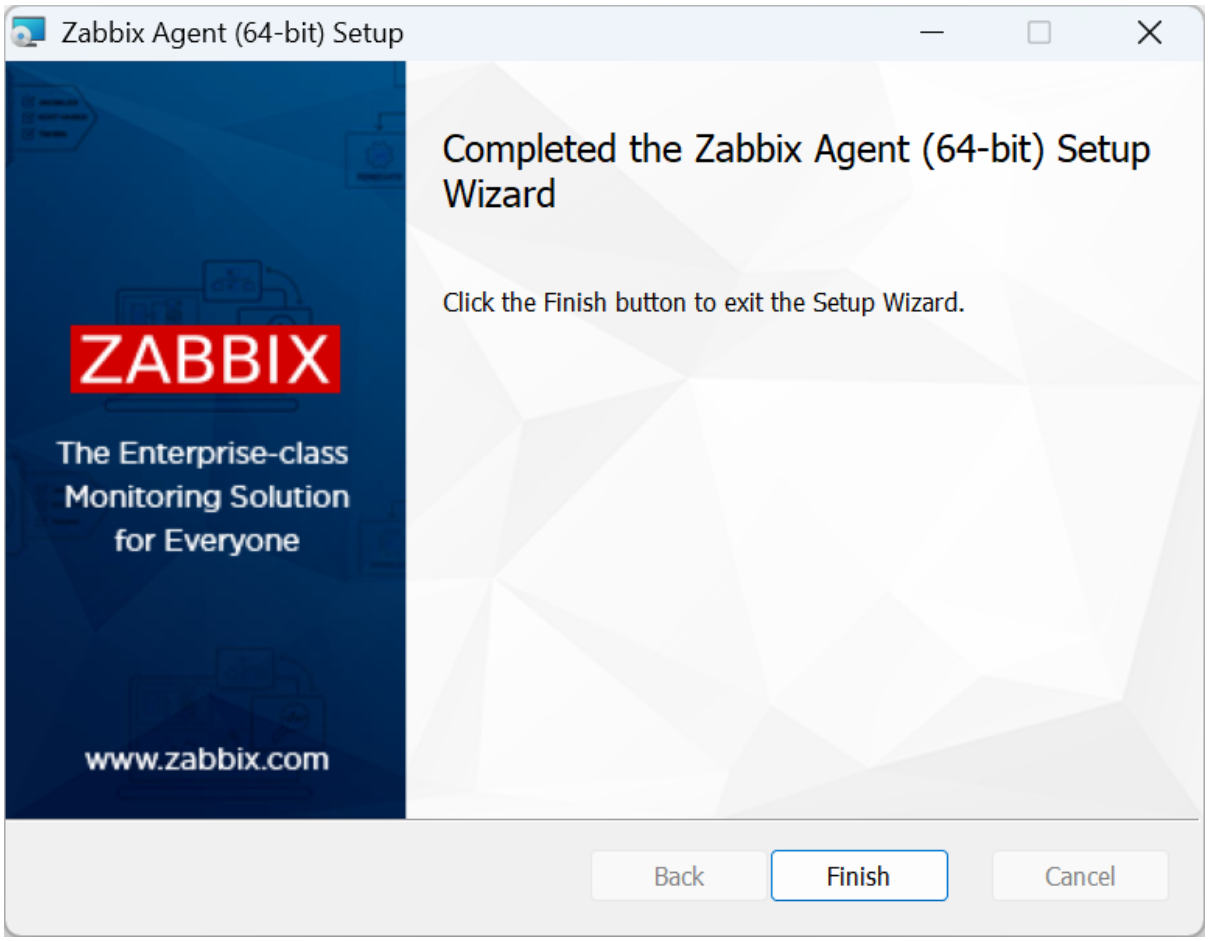
Visi atlasītie Zabbix komponenti un Zabbix aģenta konfigurācijas fails tiks instalēti jūsu norādītajā atrašanās vietā (noklusējums: `C:\Program Files\Zabbix Agent`). Tas pats attiecas uz Zabbix aģentu 2, izņemot to, ka tā iebūvēto spraudņu papildu konfigurācijas faili tiks instalēti apakšmapē `zabbix_agent2.d\plugins.d`.

Turklāt `zabbix_agentd.exe` (vai `zabbix_agent2.exe`) tiks iestatīts kā Windows pakalpojums ar atliktu automātisko palaišanu (vai automātisko palaišanu Windows versijās pirms Windows Vista/Server 2008).

Ja instalēšanas laikā darbojas cita Zabbix aģenta versija, jums tiks piedāvāts izvēlēties, vai aizvērt lietojumprogrammu un mēģināt to palaist no jauna, vai atstāt to atvērtu; otrajā gadījumā būs nepieciešama restartēšana.



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



Instalēšana no komandrindas

Zabbix aģentu var instalēt 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 instalēšanu bez lietotāja iejaukšanās un izmantot pielāgotas konfigurācijas, izmantojot parametrus.

Atbalstītie parametri

Zabbix agent MSI instalēšanas pakotnes atbalsta šādus parametrus gan Zabbix aģentam, gan Zabbix agent 2.

Note:
Zabbix agent/agent2 parametri instalēšanas laikā tiek iestatīti konfigurācijas failā. Noklikšķiniet uz parametra nosaukuma, lai skatītu tā detalizētu aprakstu un konfigurācijas piemērus lapā Zabbix agent (Windows). Par Zabbix agent 2 skatiet lapu Zabbix agent 2 (Windows).

Parameter	Description
ADDDEFAULT	Ar komatiem atdalītu komponentu saraksts, kas jāinstalē ar to noklusējuma konfigurāciju. Papildinformāciju skatiet ADDDEFAULT property. Iespējamās vērtības: `AgentProgram`, `GetProgram`, `SenderProgram`, `ALL` Piemērs: `ADDDEFAULT=AgentProgram,GetProgram`
ADDLOCAL	Ar komatiem atdalītu komponentu saraksts, kas jāinstalē lokāli. Papildinformāciju skatiet ADDLOCAL property. Iespējamās vērtības: `AgentProgram`, `GetProgram`, `SenderProgram`, `ALL` Piemērs: `ADDLOCAL=AgentProgram,SenderProgram`
ALLOWDENYKEY	Ar semikoliem atdalītu AllowKey vai DenyKey parametru saraksts, lai ierobežotu Zabbix aģenta pārbaudes. Ja nepieciešams, izmantojiet atpakaļslīpsvītru, lai atstarpotu norobežotāju (`\;`). 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 agent 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 MSI instalētājs nepalaiž Zabbix aģenta pakalpojumu.
ENABLEPATH	Izmantojiet `ENABLEPATH=1`, lai pievienotu Zabbix aģenta atrašanās vietu sistēmas PATH mainīgajam.
ENABLEPERSISTENTBUFFER	Tikai Zabbix agent 2. Iespējo lokālās pastāvīgās krātuves izmantošanu aktīvajiem vienumiem.
HOSTINTERFACE	Neobligāts parametrs, kas definē hosta saskarni.
HOSTMETADATA	Neobligāts parametrs, kas definē hosta metadatus.
HOSTMETADATAITEM	Neobligāts parametrs, kas definē vienumu, ko izmanto hosta metadatu iegūšanai.
HOSTNAME	Neobligāts parametrs, kas definē hosta nosaukumu.
HOSTNAMEITEM	Neobligāts parametrs, kas definē vienumu, ko izmanto hosta nosaukuma iegūšanai.
INCLUDE	Ar semikoliem atdalītu atsevišķu failu vai visu failu direktorijā saraksts, ko iekļaut Zabbix aģenta konfigurācijas failā.
IGNOREALREADYEXISTSFOLDER	Jaunām instalācijām izmantojiet `IGNOREALREADYEXISTSFOLDER=1`, lai atļautu instalēšanu esošā mērķa mapē ārpus `Program Files` direktorijas. Ja šis parametrs nav norādīts, instalēšana neizdodas, ja šāda mērķa mape jau pastāv.
INSTALLFOLDER	Pilns ceļš uz mapi, kurā tiks instalēti Zabbix komponenti un Zabbix aģenta konfigurācijas fails. Zabbix agent 2 gadījumā papildu konfigurācijas faili iebūvētajiem spraudņiem tiks instalēti apakšmapē `zabbix_agent2.d\plugins.d`. Piemērs: `INSTALLFOLDER="C:\Program Files\Zabbix Agent"`
LISTENIP	Ar komatiem atdalītu IP adrešu saraksts, uz kurām aģentam jāuzklausa.
LISTENPORT	Aģents uzklausīs savienojumus no servera šajā portā.
LOGFILE	Zabbix aģenta žurnālfaila nosaukums.
LOGTYPE	Žurnāla izvades veids.
NONMSICONFNAME	Pilns ceļš uz Zabbix agent pielāgotu konfigurācijas failu. Instalēšanas laikā jebkuri derīgi aģenta konfigurācijas parametri, kas atrodas šajā failā (ierobežoti līdz tiem, 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 agent 2. Fails, kurā Zabbix agent 2 jāglabā SQLite datubāze.
PERSISTENTBUFFERPERIOD	Tikai Zabbix agent 2. Laika periods, kurā dati jāglabā, ja nav savienojuma ar serveri vai starpniekserveri.

Parameter	Description
SERVER	Ar komatiem atdalītu IP adrešu saraksts, pēc izvēles CIDR notācijā, vai Zabbix serveru vai Zabbix starpniekserveru DNS nosaukumi. Šis parametrs ir **obligāts**, izņemot gadījumu, kad STARTAGENTS ir iestatīts uz 0.
SERVERACTIVE	Zabbix servera/starpniekservera adrese vai klastera konfigurācija, no kuras saņemt aktīvās pārbaudes.
SKIP	Izmantojiet `SKIP=fw`, lai MSI instalētājs nepievienotu Windows ugunsmūra izņēmuma noteikumu Zabbix aģentam.
SOURCEIP	Avota IP adrese izejošajiem savienojumiem ar Zabbix serveri vai Zabbix starpniekserveri, vai 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.
STARTUPTYPE	Zabbix aģenta pakalpojuma startēšanas veids. Iespējamās vērtības: **automatic** - startēt pakalpojumu automātiski Windows startēšanas laikā; **delayed** - *(noklusējums)* aizkavēt pakalpojuma startēšanu pēc tam, kad automātiski startētie pakalpojumi ir pabeiguši startēšanu (pieejams Windows Vista/Server 2008 un jaunākās versijās); **manual** - startēt pakalpojumu manuāli (lietotājam vai lietojumprogrammai); **disabled** - atspējot pakalpojumu, lai to nevarētu startēt lietotājs vai lietojumprogramma. Piemērs: `STARTUPTYPE=disabled`
STATUSPORT	Tikai Zabbix agent 2. Ja iestatīts, aģents uzklausīs HTTP statusa pieprasījumus šajā portā (http://localhost:<port>/status).
TIMEOUT	Norāda, cik ilgi jāgaida (sekundēs), lai izveidotu savienojumu un apmainītos ar datiem ar Zabbix starpniekserveri vai serveri.
TLSACCEPT	Ienākošie savienojumi, ko 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, kurā ir augstākā līmeņa CA sertifikāti vienaudža sertifikāta pārbaudei.
TLSCERTFILE	Pilns ceļš uz failu, kurā ir aģenta sertifikāts vai sertifikātu ķēde.
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, kurā ir atsauktie sertifikāti.
TLSKEYFILE	Pilns ceļš uz failu, kurā ir Zabbix aģenta privātā atslēga.
TLSPSKFILE	Pilns ceļš uz failu, kurā ir Zabbix aģenta iepriekš koplietotā atslēga. 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 visus simbolus nodot argumentiem lietotāja definētiem parametriem.

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=192.0.2.1^
 INSTALLFOLDER="C:\Program Files\Zabbix Agent"^
 HOSTNAME=LAPTOP-IKP7S51S^
 TLSACCEPT=psk^
 TLSCONNECT=psk^
 TLSPSKIDENTITY="PSK 001"^
 TLSPSKFILE="C:\Program Files\Zabbix Agent\psk.key"^
 TLSPSKVALUE=1f87b595725ac58dd977beef14b97461a7c1045b9a1c963065002c5473194952^
 ENABLEPATH=1^
 ALLOWDENYKEY="AllowKey=system.run[type C:\Windows\System32\drivers\etc\hosts];DenyKey=system.run[*]"
```

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 (piem., `[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 (ierobežoti līdz tiem, kas uzskaitīti tabulā iepriekš), tiks ierakstīti jaunizveidotajā aģents konfigurācijas failā. Lai saglabātu esošo aģents konfigurāciju, definējiet saglabājamos parametrus.

```
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.

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

Zabbix aģenta 2 būvēšana operētājsistēmā Windows

Pārskats

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

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

Lai izveidotu Zabbix aģentu 2, nepieciešams:

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

Zabbix aģentu 2 var izveidot, izmantojot kādu no tālāk norādītajām metodēm:

- Izmantojot vcpkg - automatizēta pieeja, kas vienkāršo atkarību pārvaldību, izmantojot C++ pakotņu pārvaldnieku.
- Manuāla izveide - manuāla pieeja, kuras gadījumā pirms aģenta kompilēšanas jāinstalē visas atkarības.

Attention:
Pirms izveides procesa sākšanas ņemiet vērā:

- Lai izpildītu komandas, izmantojiet komandu uzvedni, kas palaista lietotājam ar pietiekamām atļaujām rakstīt aizsargātās mapēs. Tomēr, instalējot OpenSSL un PCRE2, izmantojiet MSYS2 terminālī.
- Ieteicams izveidot darba direktoriju `C:\Zabbix` visiem pirmkoda failiem un izveides mapēm. Tomēr kompilētie komponenti jāinstalē mapē `C:\Zabbix\x64` (vai `C:\Zabbix\x86` 32 bitu izveidēm).

Zabbix aģenta 2 izveide, izmantojot vcpkg

Šajā sadaļā ir sniegti norādījumi par Zabbix aģenta izveidi, izmantojot 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, kurā ir iekļauts vcpkg pakotņu pārvaldnieks.

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

3. Lejupielādējiet MinGW izplatījumu, 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`

Attention:
MinGW versijai jāatbilst minimālajām prasībām.

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

4. Inicializējiet vcpkg un instalējiet Zabbix aģenta 2 izveidei nepieciešamās atkarības (ņemiet vērā, ka tas var aizņemt zināmu 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 izvelciet to mapē `C:\Zabbix\zabbix-8.0.0`.

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

- 64 bitu būvējumiem:

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

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

:: Iestatiet Crypt32 bibliotēkas saistītāja karodziņus:
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 pašreizējās sesijas sistēmas mainīgajam `PATH`:
set PATH=C:\Zabbix\mingw32\bin;%PATH%
set PATH=C:\Zabbix\Go\bin;%PATH%

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

:: Iestatiet Crypt32 bibliotēkas saistītāja karodziņus:
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 mapē `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 mapē `C:\Zabbix\zabbix-8.0.0\src\go\conf`.

Lai palaistu aģentu, kopējiet `zabbix_agent2.exe` un tā konfigurācijas failus atsevišķā mapē (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.conf C:\Zabbix\agent2\
xcopy /E /I C:\Zabbix\zabbix-8.0.0\src\go\conf\zabbix_agent2.d C:\Zabbix\agent2\zabbix_agent2.d\

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

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

Zabbix aģenta 2 ielādējamo spraudņu kompilēšana

1. Lejupielādējiet Zabbix spraudņa avotu, kas atbilst jūsu Zabbix aģenta 2 versijai (piem., `zabbix-agent2-plugin-postgresql-8.0.0.ta`
un izvelciet to uz `C:\Zabbix`.

> **Attention:**
> Pirms spraudņa instalēšanas, lūdzu, pārbaudiet tā README failu. Tajā var būt norādītas konkrētas prasības un instalēšanas instrukcijas.

2. Pārejiet uz izvilkto spraudņa direktoriju un kompilējiet spraudni:

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

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

#### 32 bitu būvēm:
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 Zabbix aģents 2 to var ielādēt. Norādiet spraudņa binārā faila ceļu spraudņa konfigurācijas failā, piemēram, `postgresql.conf` failā PostgreSQL spraudnim:

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

Spraudņa konfigurācijas faila ceļš ir jānorāda Zabbix aģenta 2 konfigurācijas faila parametrā Include:

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

Lai iegūtu plašāku informāciju par spraudņu konfigurēšanu, pārejiet uz 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 instalators). Instalēšanas laikā noteikti norādiet `C:\Zabbix\msys64` kā instalācijas mapi.

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

3. Lejupielādējiet MinGW izplatīšanu, kas izmanto Microsoft Visual C runtime 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`

Attention:
MinGW versijai ir jāatbilst minimālajām prasībām.

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

OpenSSL instalēšana

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

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

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

2. Lejupielādējiet OpenSSL avota arhīvu un izvelciet to mapē `C:\Zabbix\openssl-3.5.0`.

3. Pārejiet uz izvilkto 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:
Noteikti noņemiet rakstīšanas piekļuvi direktorijai `C:\Zabbix\x64\OpenSSL-Win64-350-static` lietotājiem, kas nav administratori. Pretējā gadījumā aģents ielādēs SSL iestatījumus no ceļa, ko var modificēt lietotāji bez paaugstinātām privilēģijām, tādējādi radot potenciālu drošības ievainojamību.

- Opcija `no-shared` padara `libcrypto.lib` un `libssl.lib` OpenSSL statiskās bibliotēkas pašpietiekamas, tāpēc Zabbix binārie faili ietver OpenSSL un tiem nav nepieciešami ārēji DLL faili. Tas nozīmē, ka Zabbix bināros failus var kopēt uz citiem Windows datoriem bez OpenSSL bibliotēkām; tomēr, izlaižot jaunu OpenSSL kļūdu labojumu versiju, Zabbix aģents būs jākompilē atkārtoti.
- Ja opcija `no-shared` netiek izmantota, Zabbix izpildes laikā paļaujas uz OpenSSL DLL failiem. Tas nozīmē, ka OpenSSL atjauninājumiem var nebūt nepieciešama Zabbix aģenta atkārtota kompilēšana; tomēr, kopējot to uz citiem datoriem, ir jākopē arī OpenSSL DLL faili.

Papildinformāciju par citām OpenSSL konfigurācijas opcijām skatiet OpenSSL dokumentācijā.

4. Konfigurējiet un instalējiet OpenSSL, izpildot skriptu (ņemiet vērā, ka tas var aizņemt zināmu 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 izvelciet to mapē `C:\Zabbix\zabbix-8.0.0`.

Ja nepieciešams ģenerēt pirmkoda arhīvu no neapstrādāta pirmkoda repozitorija (piemēram, lai lietotu pielāgotus ielāpus vai veidotu no jaunākā pirmkoda), palaidiet tālāk norādītās komandas **Linux** datorā, kurā ir instalēts Go (nepieciešams Zabbix aģenta 2 konfigurēšanai):

```
git clone https://git.zabbix.com/scm/zbx/zabbix.git
cd zabbix
./bootstrap.sh
./configure
make dist
```

Tiks izveidots pirmkoda arhīvs, ko pēc tam var kopēt uz Windows datoru.

2. Atveriet komandu uzvedni ar administratora privilēģijām. Pēc tam pārejiet uz Zabbix būvēšanas direktoriju un kompilējiet Zabbix aģentu; noteikti pareizi norādiet direktorijas, kurās ir instalēti OpenSSL un PCRE2:

- 64 bitu būvējumiem:

```
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

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

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

- 32 bitu būvējumiem:

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

#### Ar TLS atbalstu:
mingw32-make ARCH=x86 PCRE2="C:\Zabbix\x86\PCRE2" OPENSSL="C:\Zabbix\x86\OpenSSL-Win32-350-static"

#### Bez TLS atbalsta:
mingw32-make ARCH=x86 PCRE2="C:\Zabbix\x86\PCRE2"
```

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

Lai palaistu aģentu, nokopējiet zabbix_agent2.exe bināro failu un tā konfigurācijas failus atsevišķā mapē (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.conf C:\Zabbix\agent2\
xcopy /E /I C:\Zabbix\zabbix-8.0.0\src\go\conf\zabbix_agent2.d C:\Zabbix\agent2\zabbix_agent2.d\

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

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

Building Zabbix agent on Windows

Pārskats

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

Šie norādījumi attiecas uz Windows versijām, kas atbalsta Visual Studio 2022.

Zabbix aģenta izveidei nepieciešams:

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

Zabbix aģentu var izveidot, izmantojot kādu no tālāk norādītajām metodēm:

- Izmantojot vcpkg - automatizēta pieeja, kas vienkāršo atkarību pārvaldību, izmantojot C++ pakotņu pārvaldnieku.
- Manuāla izveide - manuāla pieeja, kuras gadījumā pirms aģenta kompilēšanas jāinstalē visas atkarības.

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

> **Attention:**
> Pirms izveides procesa sākšanas, lūdzu, ņemiet vērā:

> - Komandu izpildei izmantojiet "x64 Native Tools Command Prompt" (iekļauta Build Tools for Visual Studio 2022), kas palaista lietotājam ar pietiekamām atļaujām rakstīt aizsargātās mapēs.
> - Ieteicams izveidot darba direktoriju C:\Zabbix, kur glabāt visus pirmkoda failus un izveides mapes. Tomēr kompilētie komponenti jāinstalē mapē C:\Program Files\Zabbix\x64.

Zabbix aģenta izveide, izmantojot vcpkg

Šajā sadaļā sniegti norādījumi par Zabbix aģenta izveidi, izmantojot 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, kurā iekļauti rīki, kas nepieciešami aģenta izveidei, izmantojot vcpkg:

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

2. Atveriet cmd.exe ar administratora privilēģijām, inicializējiet vcpkg un instalējiet Zabbix aģenta izveidei nepieciešamās atkarības (ņemiet vērā, ka tas var aizņemt zināmu 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 avota arhīvu un izvelciet to mapē C:\Zabbix\zabbix-8.0.0.

4. Pārejiet uz Zabbix izveides direktoriju (C:\Zabbix\zabbix-8.0.0\build\win32\project) un izveidojiet tālāk norādīto build.bat skriptu; noteikti pareizi norādiet direktorijas, kurās 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. Atveriet "x64 Native Tools Command Prompt" un kompilējiet Zabbix aģentu, izpildot skriptu:

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

Ja atvērāt citu komandu uzvedni, vispirms inicializējiet Visual Studio izveides vidi:

```
"%ProgramFiles(x86)%\Microsoft Visual Studio\2022\BuildTools\VC\Auxiliary\Build\vcvarsall.bat" x64
cd C:\Zabbix\zabbix-8.0.0\build\win32\project
build.bat
```

Ceļus, kuros ir atstarpes, ievietojiet pēdiņās.

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

Lai palaistu aģentu, nokopējiet `zabbix_agentd.exe` un tā konfigurācijas failu tam paredzētā mapē (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.conf C:\Zabbix\agent\

C:\Zabbix\agent\zabbix_agentd.exe -c C:\Zabbix\agent\zabbix_agentd.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, kurā ir iekļauti aģenta manuālai būvēšanai nepieciešamie rīki:

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

OpenSSL instalēšana

> **Note:**
> Lai kompilētu Zabbix aģentu 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 Strawberry Perl instalēšanas laikā tika kompilēts Netwide Assembler (NASM; nepieciešams OpenSSL kompilēšanai):

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

Ja NASM nav kompilēts, instalējiet to manuāli. Plašāku informāciju skatiet NASM dokumentācijā.

4. Lejupielādējiet OpenSSL avota arhīvu un izvelciet to mapē `C:\Zabbix\openssl-3.5.0`.

5. Pārejiet uz izvilkto mapi 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 aģentu operētājsistēmā Windows, izvēlaties pielāgotu OpenSSL mapi (piemēram, `C:\zabbix` vai `C:\openssl-64bit`), noteikti noņemiet rakstīšanas piekļuvi šai mapei lietotājiem, kas nav administratori. Pretējā gadījumā aģents ielādēs SSL iestatījumus no ceļa, ko var modificēt lietotāji bez paaugstinātām privilēģijām, tādējādi radot potenciālu drošības ievainojamību.

- Opcija `no-shared` padara libcrypto.lib un libssl.lib OpenSSL statiskās bibliotēkas pašpietiekamas, tādēļ Zabbix binārie faili ietver OpenSSL un tiem nav nepieciešami ārēji DLL faili. Tas nozīmē, ka Zabbix bināros failus var kopēt uz citiem Windows datoriem bez OpenSSL bibliotēkām; tomēr, izlaižot jaunu OpenSSL kļūdu labojumu versiju, Zabbix aģents būs jākompilē atkārtoti.
- Ja opcija `no-shared` netiek izmantota, Zabbix izpildlaikā paļaujas uz OpenSSL DLL failiem. Tas nozīmē, ka OpenSSL atjauninājumiem var nebūt nepieciešama Zabbix aģenta atkārtota kompilēšana; tomēr, kopējot to uz citiem datoriem, ir jākopē arī OpenSSL DLL faili.

Plašāku informāciju par citām OpenSSL konfigurācijas opcijām skatiet OpenSSL dokumentācijā.

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

Attention:
Palaidiet testus bez administratīvajām privilēģijām; pretējā gadījumā tas var radīt neparedzētus rezultātus vai drošības riskus. Ja kāds tests neizdodas, problēmu novēršanas informāciju skatiet OpenSSL dokumentācijā.

```
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 un galvenes 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 avota arhīvu un izvelciet to mapē C:\Zabbix\zabbix-8.0.0.

Ja nepieciešams ģenerēt avota arhīvu no neapstrādāta avota repozitorija (piemēram, lai lietotu pielāgotus ielāpus vai veidotu no jaunākā avota koda), Linux datorā izpildiet šādas komandas:

```
git clone https://git.zabbix.com/scm/zbx/zabbix.git
cd zabbix
./bootstrap.sh
./configure
make dist
```

Tiks izveidots avota arhīvs, ko pēc tam var kopēt uz Windows datoru.

2. Pārejiet uz Zabbix būvēšanas direktoriju, atveriet "x64 Native Tools Command Prompt" un kompilējiet Zabbix aģentu (vai citus komponentus); noteikti pareizi norādiet direktorijas, kurās ir instalēti OpenSSL un PCRE2:

```
"%ProgramFiles(x86)%\Microsoft Visual Studio\2022\BuildTools\VC\Auxiliary\Build\vcvarsall.bat" x64

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
```

Ceļi, kuros ir atstarpes, jāiekļauj pēdiņās.

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

Lai palaistu aģentu, nokopējiet zabbix_agentd.exe un tā konfigurācijas failu īpaši paredzētā mapē (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.conf C:\Zabbix\agent\

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

macOS aģenta instalēšana no PKG

Pārskats

Zabbix aģentu var instalēt macOS, izmantojot PKG instalēšanas pakotnes, kas pieejamas lejupielādei.

Zabbix aģenta pakotnes ir pieejamas ar šifrēšanu vai bez tās.

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 binārus 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 veidot aģents binārus 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:

```
/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
```

Pēc tam instalējiet Automake un pkg-config:

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

PCRE, OpenSSL un GnuTLS bibliotēku sagatavošana ir atkarīga no tā, kādā veidā 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 veidošanai 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
```

AWS izvietošanas rokasgrāmata Zabbix serverim

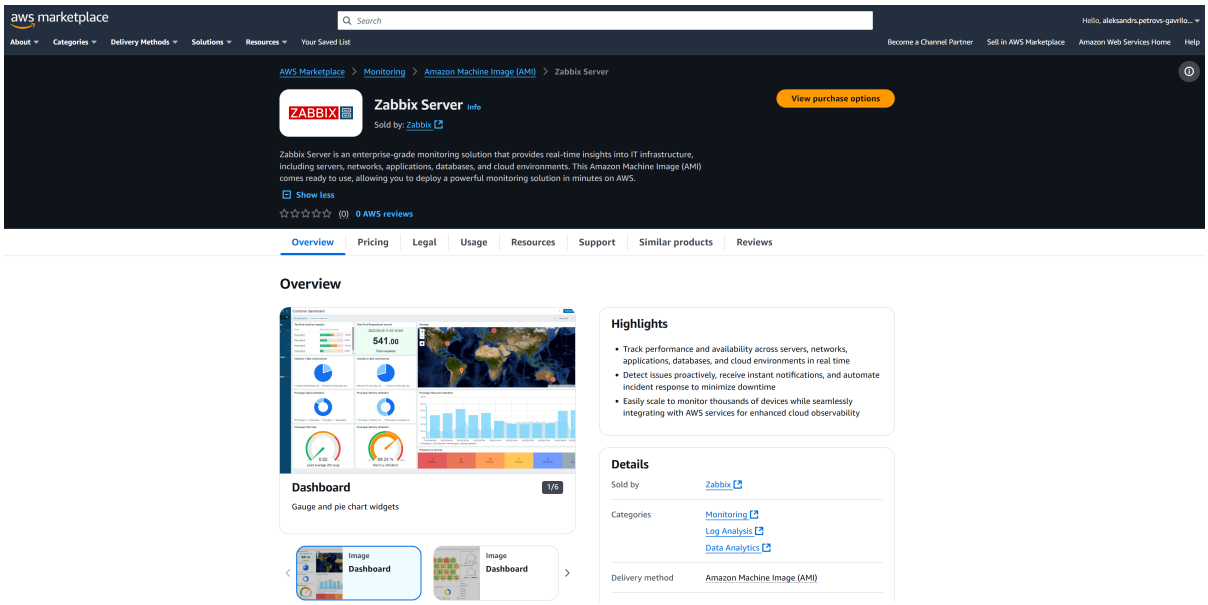
Pārskats

Šajā lapā aprakstīts, kā izvietot Zabbix serveri, izmantojot AWS.

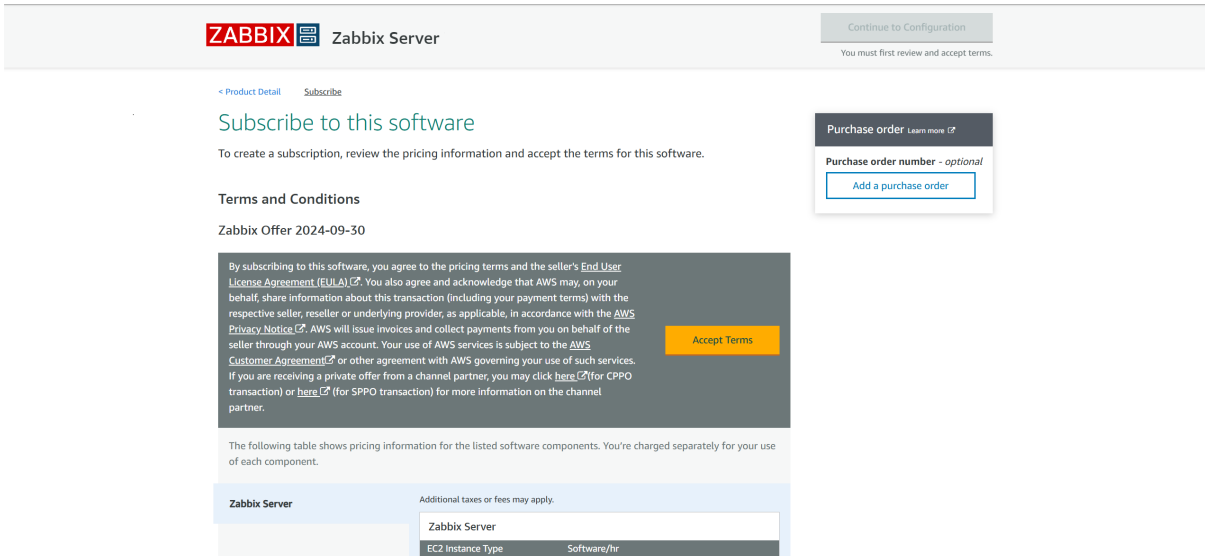
> **Attention:**
> Platformā izvietošanai ir pieejamas vairākas Zabbix versijas. Noteikti izlasiet tās pašas Zabbix versijas dokumentāciju, kuru esat izvietojis. Dokumentāciju dažādām Zabbix versijām var atlasīt versiju nolaižamajā izvēlnē.

Ierīces izvietošana

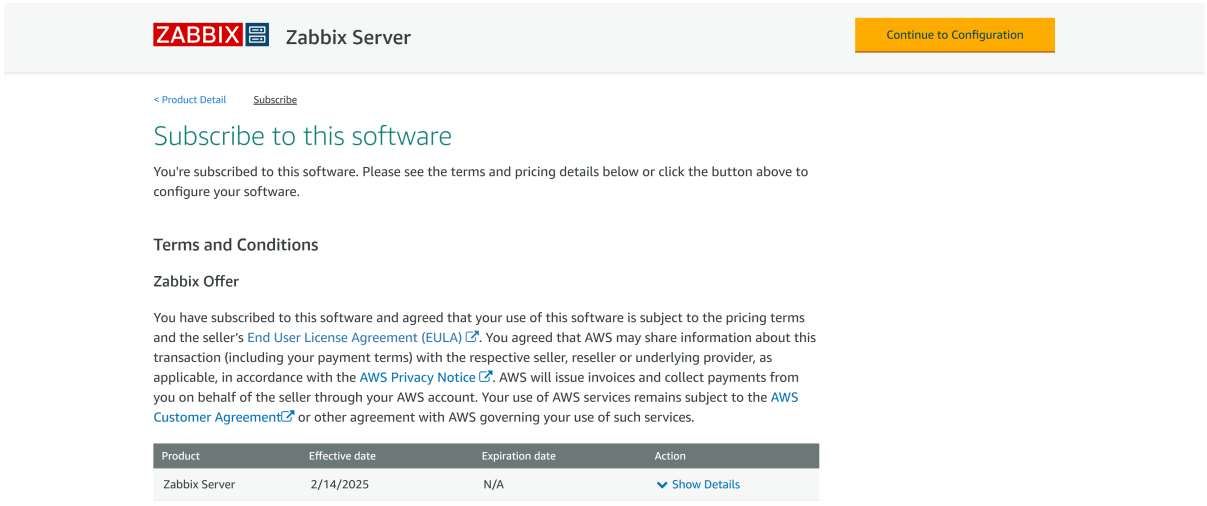
Lai izvietotu Zabbix ierīci, vispirms atveriet Zabbix AWS Marketplace lapu. Pēc tam noklikšķiniet uz *View purchase options*, lai izvietotu attēlu un pārietu uz abonēšanas lapu.



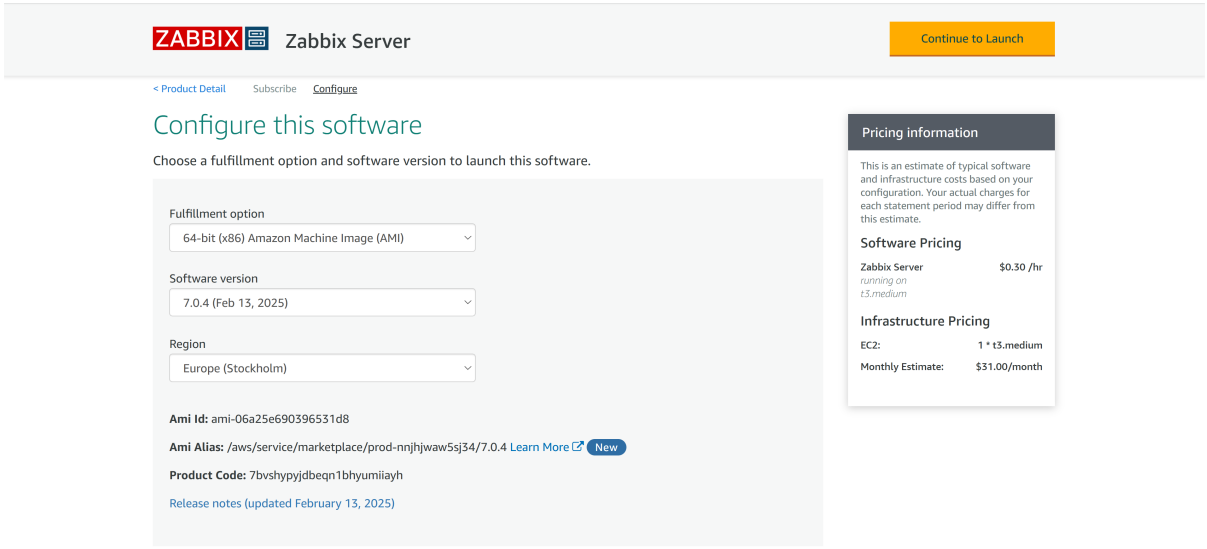
Noklikšķiniet uz pogas *Accept terms*, lai piekristu *Terms and Conditions*.



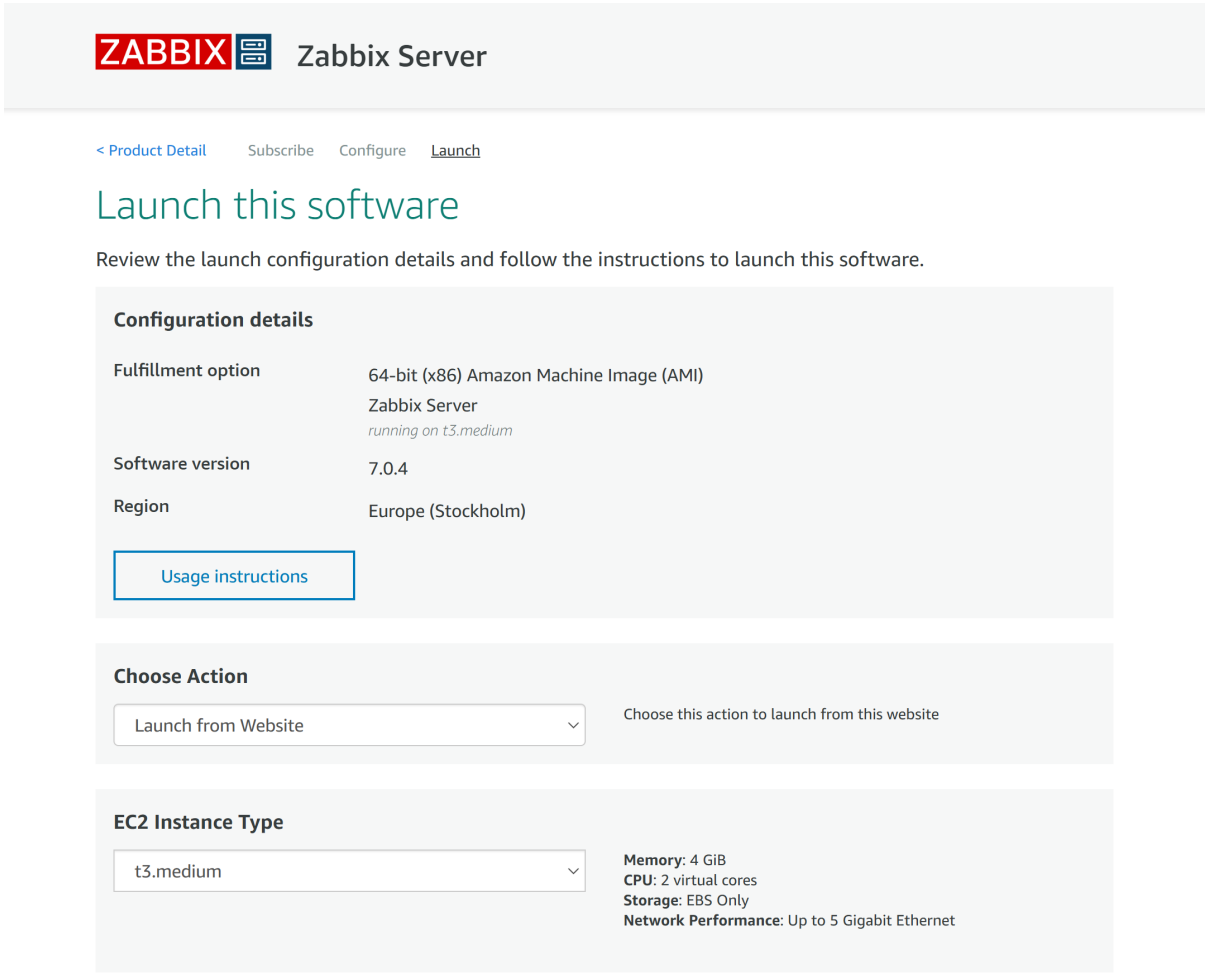
Noklikšķiniet uz pogas *Continue to Configuration*, lai pārietu uz Zabbix ierīces konfigurācijas sadaļu.



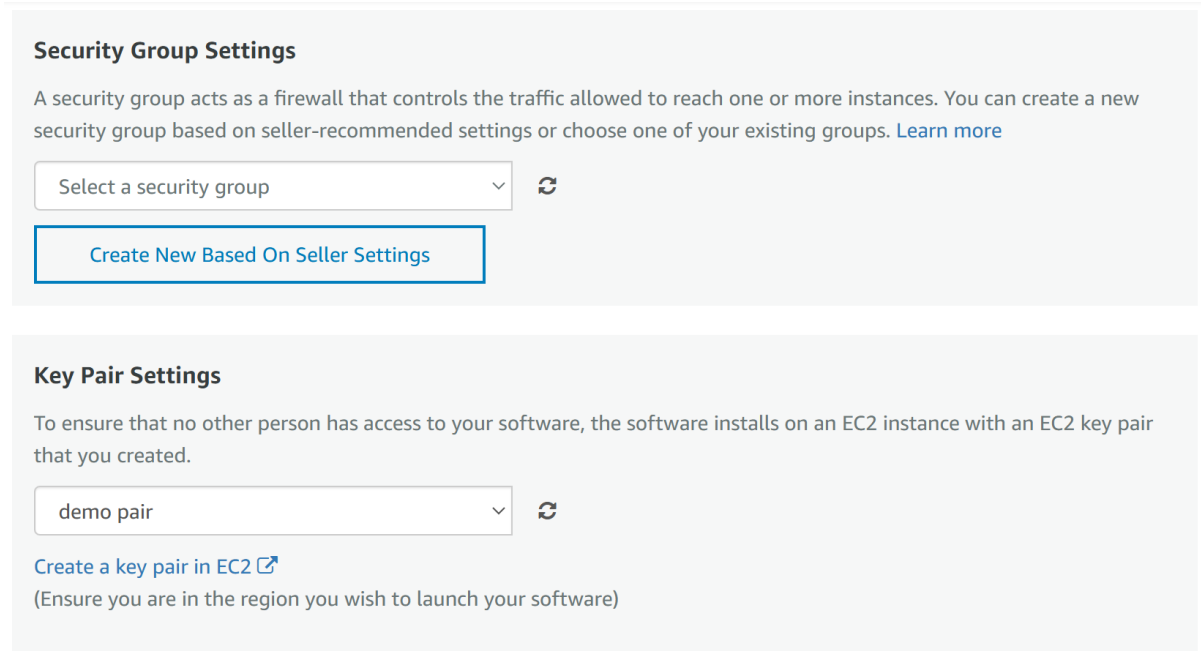
Atlasiet Fulfilment opciju, Zabbix versiju un vēlamo reģionu.



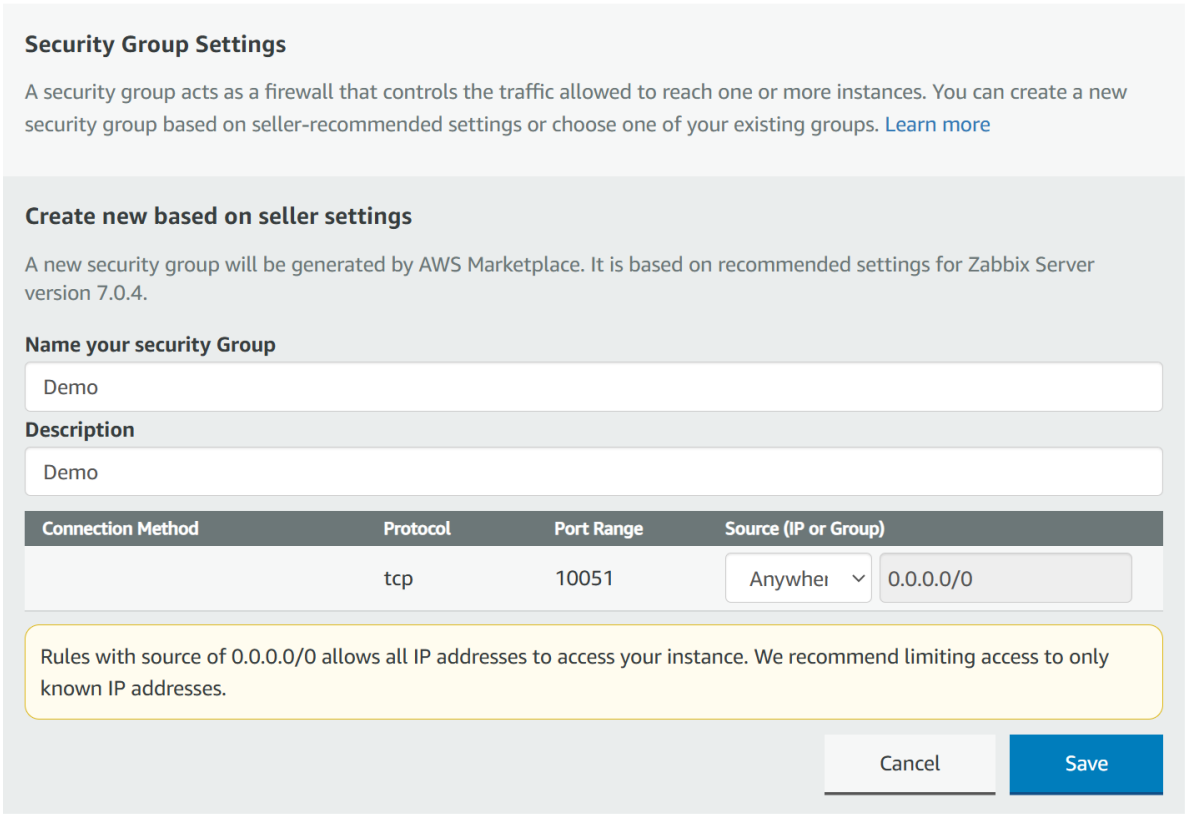
Noklikšķiniet uz pogas *Continue to Launch*. Palaišanas lapā atlasiet vēlamo palaišanas opciju, EC2 instances tipu, VPC un apakštīkla iestatījumus.



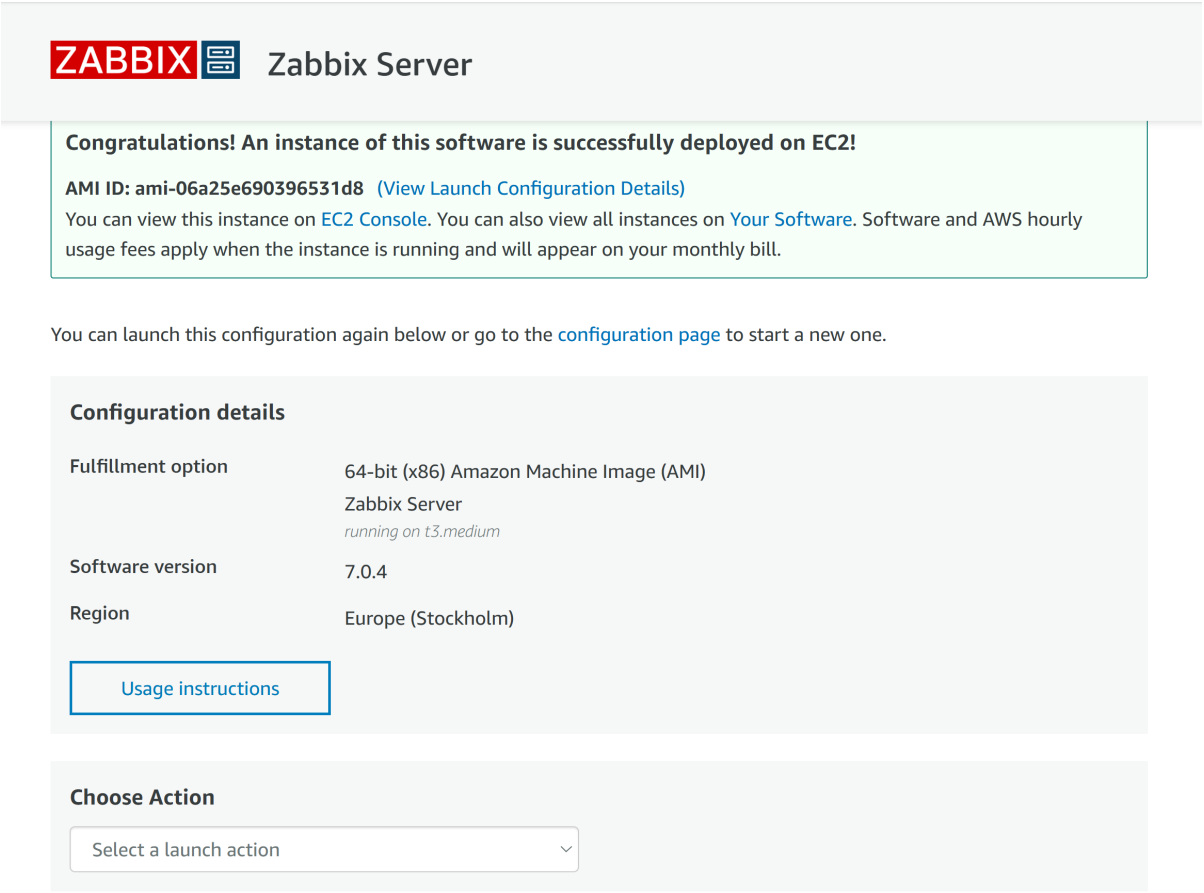
Palaišanas lapas apakšdaļā norādiet drošības grupas iestatījumus un atlasiet vai izveidojiet EC2 atslēgu pāri - noteikti saglabājiet savu privāto atslēgu drošā vietā!



Ja atlasāt *Create a new Security Group based on seller settings*, tiks atļauts TCP ports 10051 (Zabbix serveris). Piezīme: ja vēlaties piekļūt instancei, izmantojot SSH un HTTP/HTTPS (lietotāja saskarne), drošības grupas iestatījumos būs jāatļauj arī SSH un HTTP/HTTPS savienojumi.

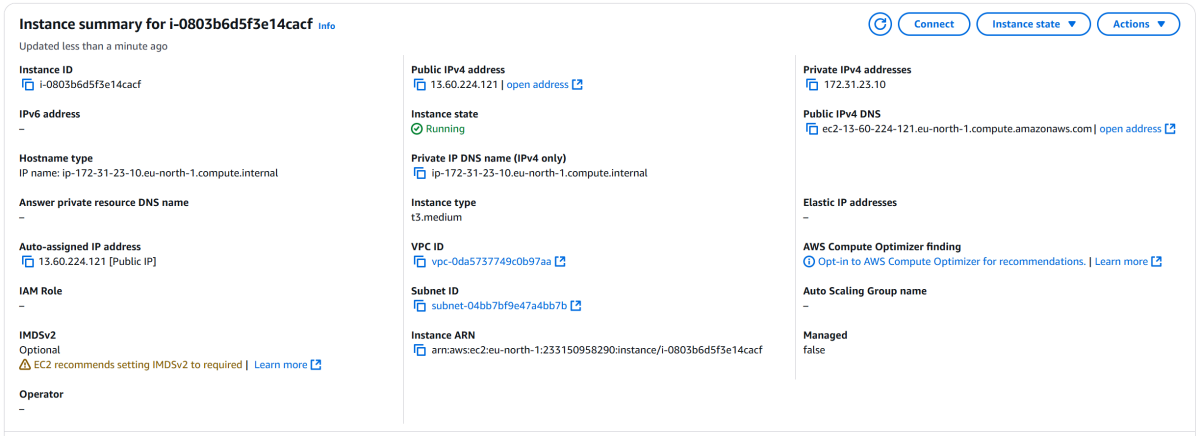


Apsveicam! Jūsu Zabbix ierīce ir veiksmīgi izvietota.

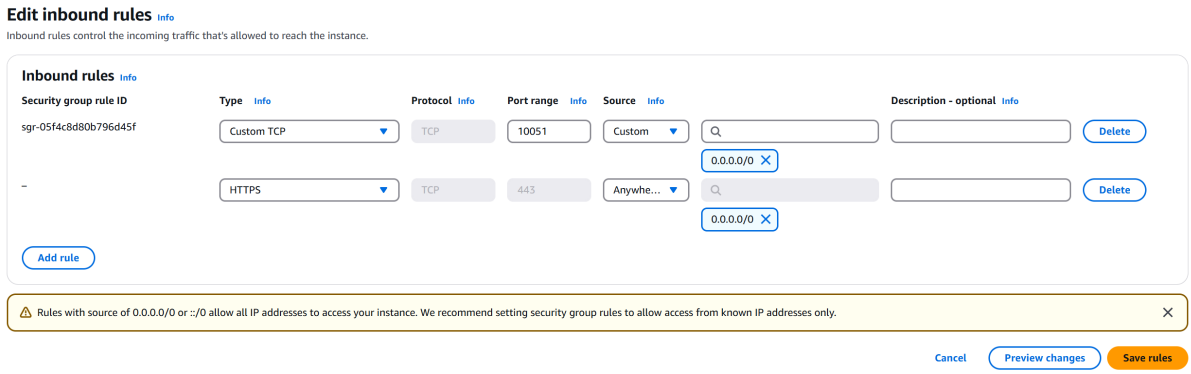


Piekļuve appliance

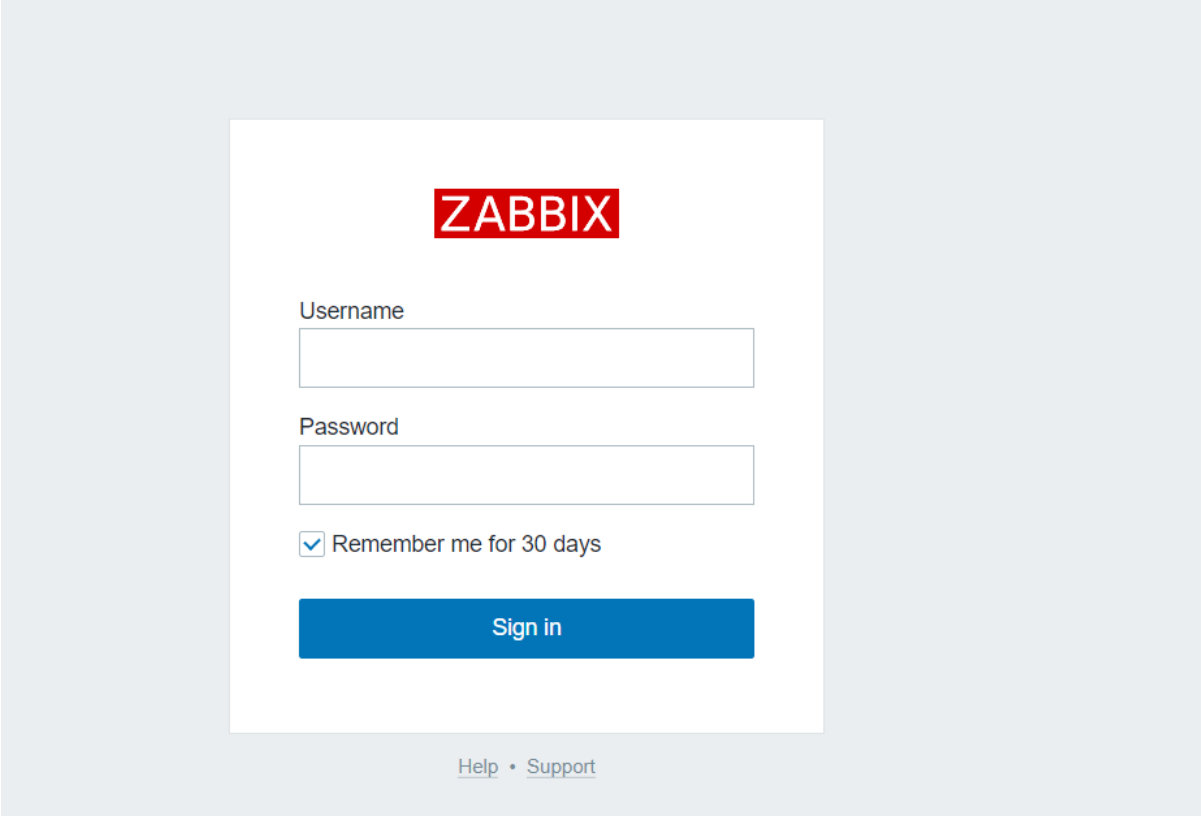
Atveriet sadaļu *Instances* un atlasiet tikko izvietoto Zabbix appliance instanci.



Lai piekļūtu Zabbix lietotāja saskarnei, izmantojot instances publisko IPv4 adresi vai publisko IPv4 DNS, drošības grupā atļaujiet ienākošo HTTPS datplūsmu 443. portā.



Tagad varat piekļūt Zabbix lietotāja saskarnei, atverot Zabbix appliance publisko IPv4 adresi vai publisko IPv4 DNS.



Lai iegūtu paroli, ir jāpievienojas datoram, izmantojot seriālo konsoli vai SSH. Noklusējuma SSH lietotājvārds ir rocky. Neaizmirstiet izmantot iepriekš atlasīto vai ģenerēto atslēgu pāri. Ņemiet vērā, ka akreditācijas datu ģenerēšana var aizņemt laiku. Akreditācijas datu ģenerēšana var ilgt dažas minūtes. Pirms mēģināt izveidot savienojumu, uzgaidiet dažas minūtes.

```
Last login: Tue Nov 16 15:16:07 2021 from ec2-13-48-4-200.eu-north-1.compute.amazonaws.com
********************************************************************************

Zabbix frontend credentials:

Username: Admin

Password: gJ8f7XpgJbsnc2BU48g

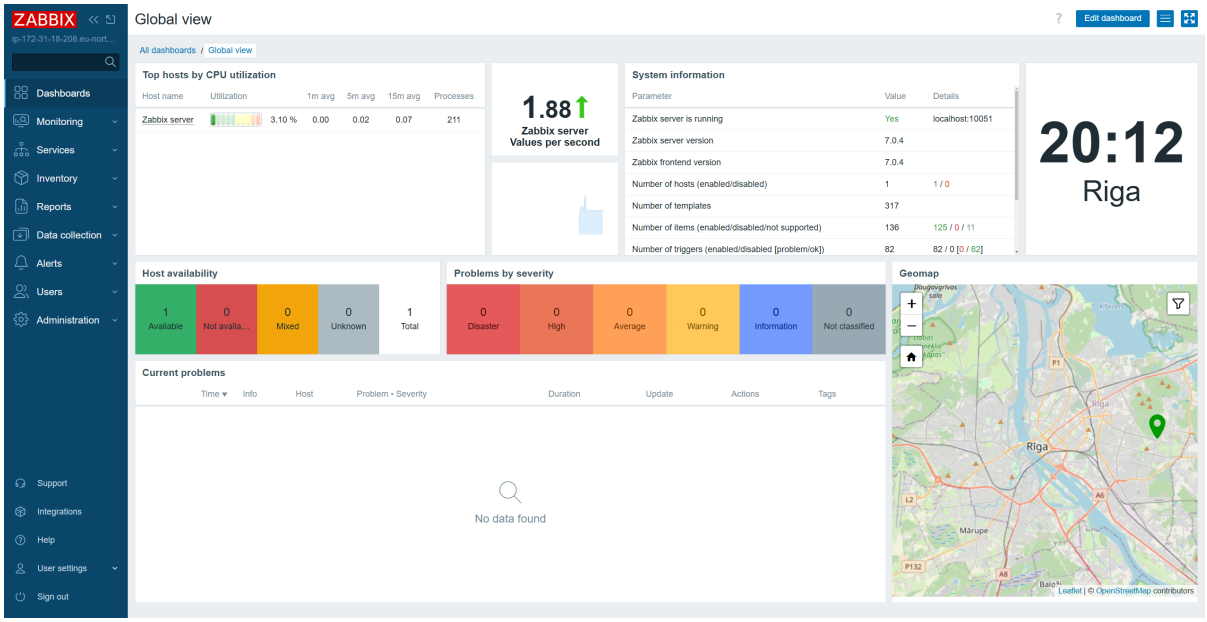
To learn about available professional services, including technical suppport and training, please visit https://www.zabbix.com/services

Official Zabbix documentation available at https://www.zabbix.com/documentation/current/

Note! Do not forget to change timezone PHP variable in /etc/php.d/99-zabbix.ini file.

********************************************************************************
```

Tagad varat izmantot šo informāciju, lai pieteiktos Zabbix lietotāja saskarnē.



Piekļuve datubāzei

Zabbix datubāzei var piekļūt, izmantojot lietotāju `root`, neievadot paroli.

```
[root@ip-172-31-29-138 ~]# mysql -uroot
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 124
Server version: 8.0.26 MySQL Community Server - GPL

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

Akreditācijas dati tiek glabāti konfigurācijas failā `/root/.my.cnf`.

Zabbix lietotāja saskarnes noklusējuma laika joslu var mainīt sadaļā *Administration* > *General* > *GUI*.

AWS izvietošanas rokasgrāmata Zabbix starpniekserverim

Pārskats

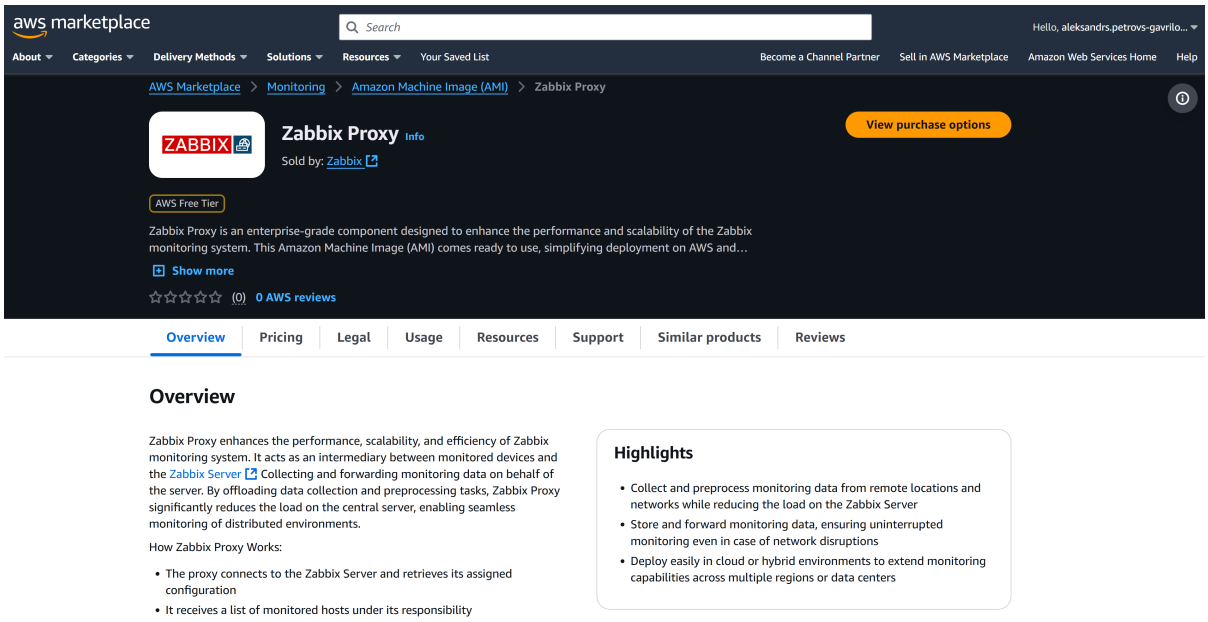
Šajā lapā aprakstīts, kā izvietot Zabbix starpniekserveri, izmantojot AWS.

Attention:
Platformā izvietošanai ir pieejamas vairākas Zabbix versijas. Noteikti izlasiet tās pašas Zabbix versijas dokumentāciju, kuru esat izvietojis. Dokumentāciju dažādām Zabbix versijām var atlasīt versiju nolaižamajā izvēlnē.

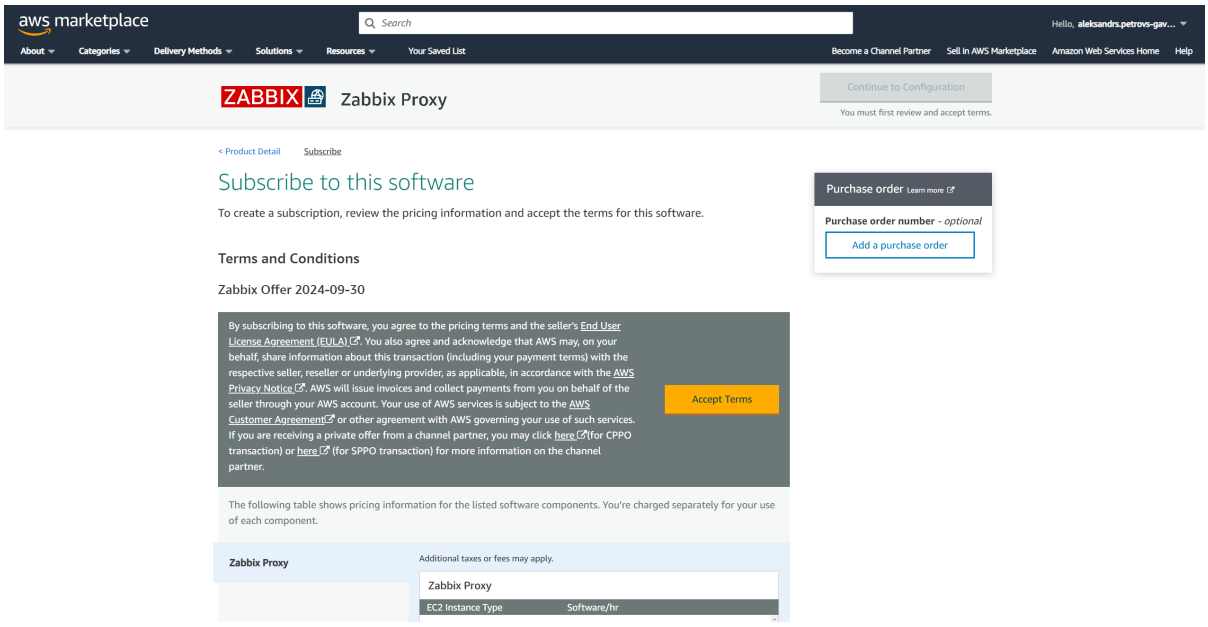
Zabbix starpniekservera izvietošana

Attention:
Lai izmantotu Zabbix starpniekservera komponenti, ir nepieciešams funkcionējošs Zabbix servera eksemplārs.

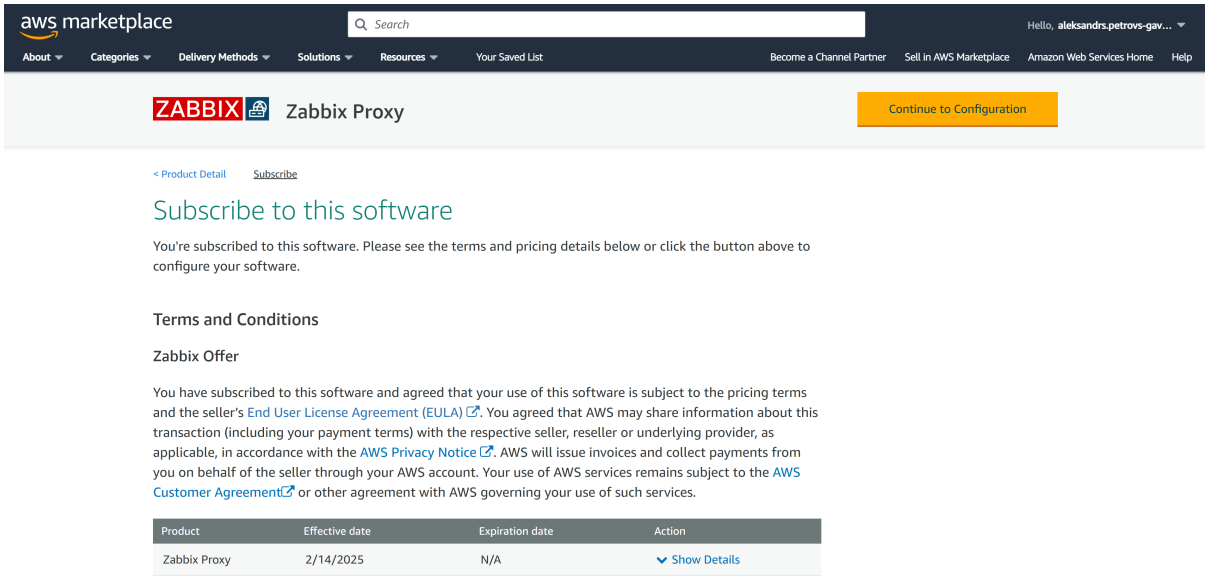
Lai izvietotu Zabbix starpniekserveri, vispirms dodieties uz Zabbix AWS marketplace lapu. Pēc tam noklikšķiniet uz *View purchase options*, lai izvietotu attēlu un pārietu uz abonēšanas lapu.



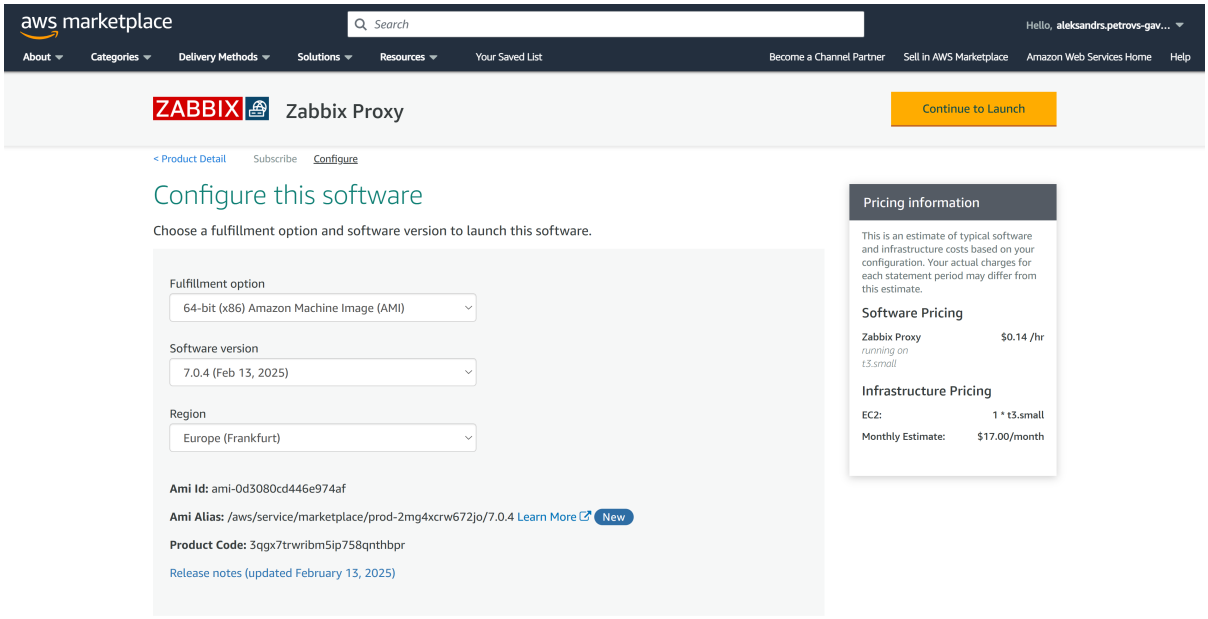
Noklikšķiniet uz pogas *Accept terms*, lai piekristu *Terms and Conditions*.



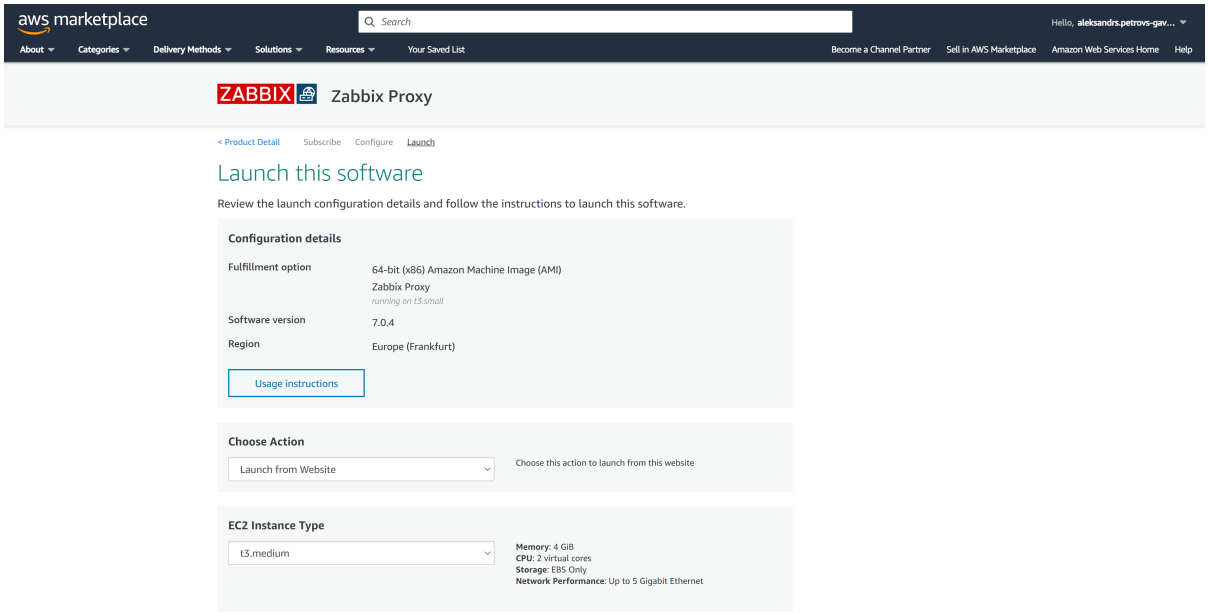
Noklikšķiniet uz pogas *Continue to Configuration*, lai pārietu uz Zabbix starpniekservera konfigurācijas sadaļu.



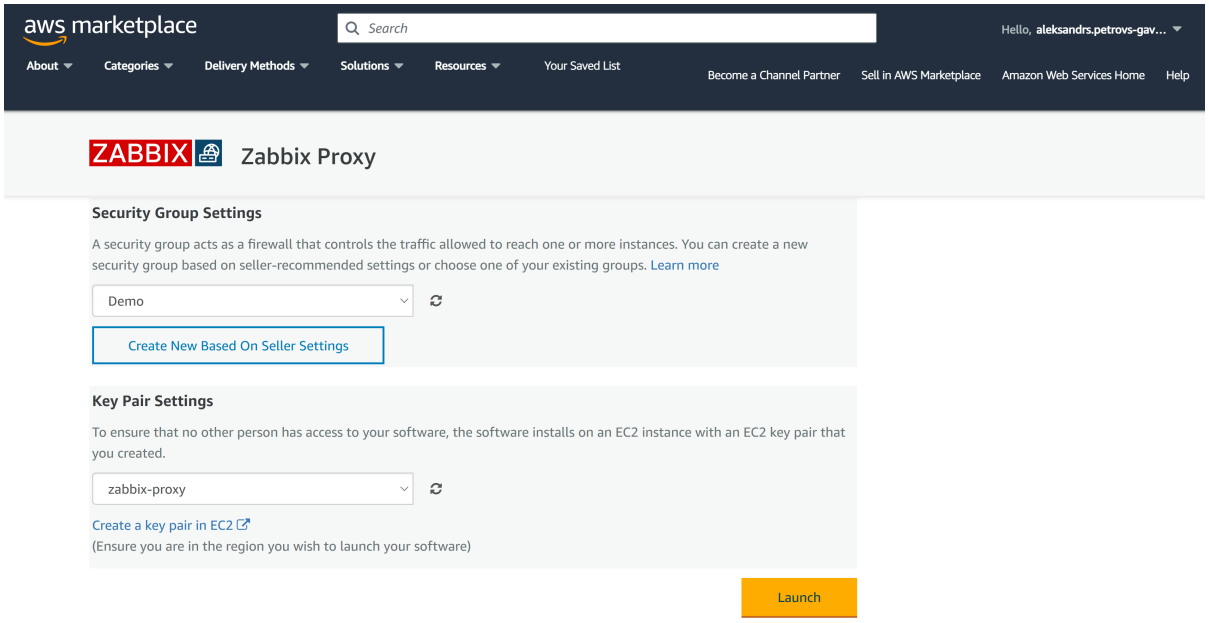
Atlasiet Fulfilment opciju, Zabbix versiju un vēlamo reģionu.



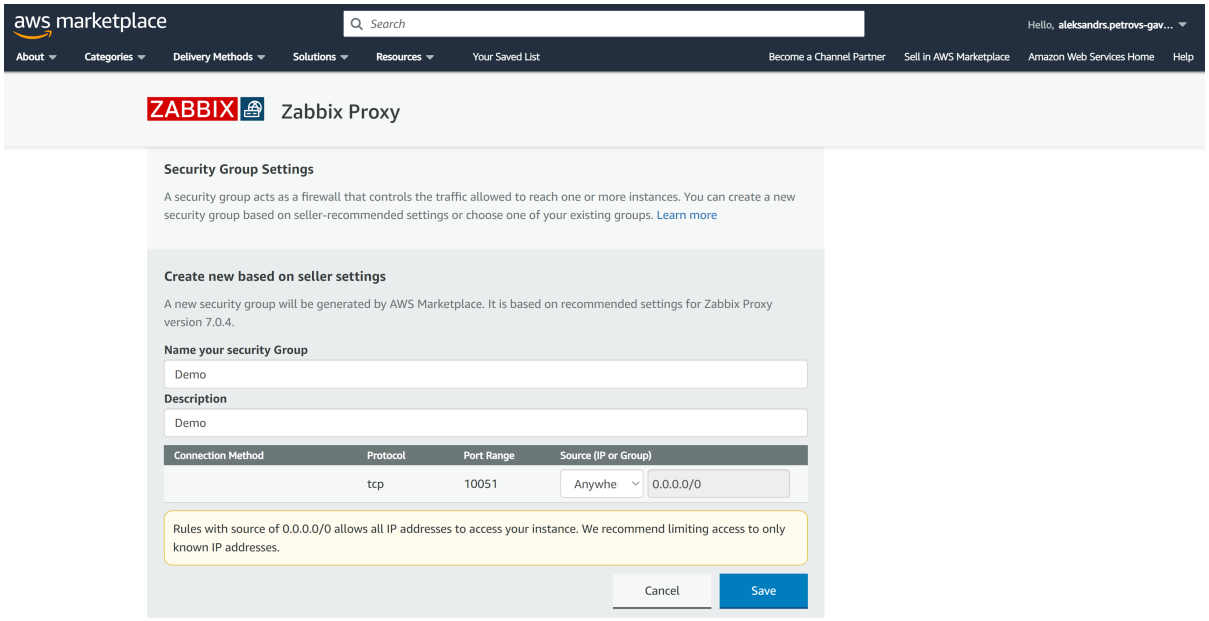
Noklikšķiniet uz pogas *Continue to Launch*. Lapā Launch atlasiet vēlamo Launch opciju, EC2 instances tipu, VPC un apakštīkla iestatījumus.



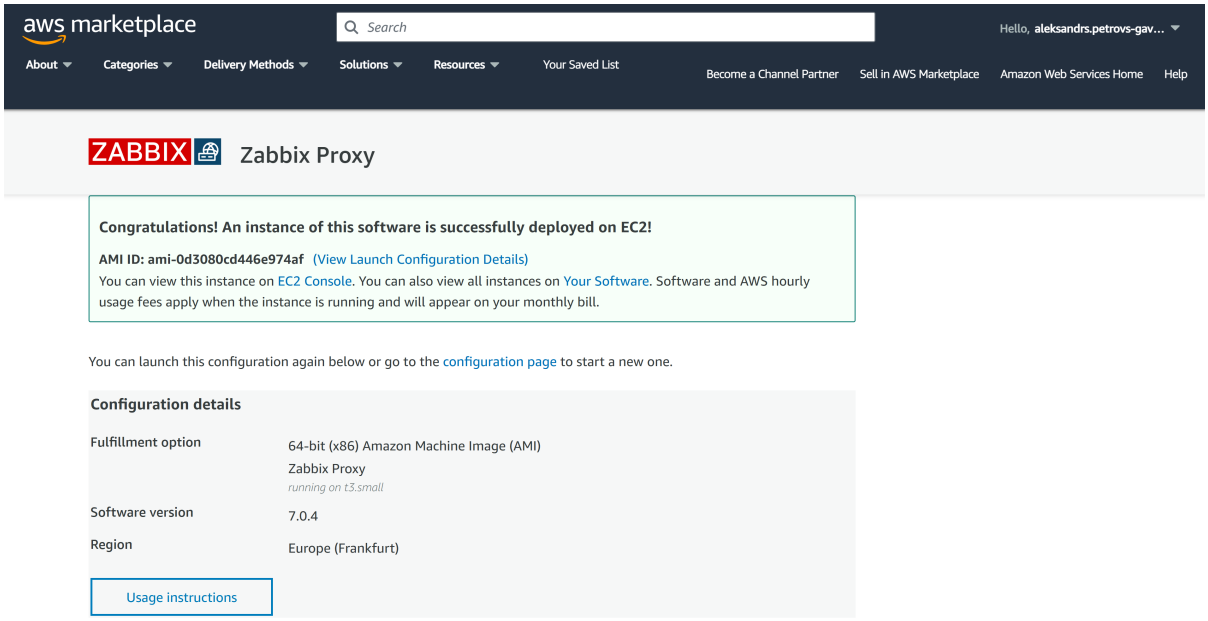
Launch lapas apakšdaļā norādiet drošības grupas iestatījumus un atlasiet vai izveidojiet EC2 atslēgu pāri. Noteikti saglabājiet savu privāto atslēgu drošā vietā.



Ja atlasāt *Create a new Security Group based on the Seller settings*, tiks atļauts TCP ports 10051 (Zabbix serveris). Piezīme: ja vēlaties piekļūt instancei, izmantojot SSH, drošības grupas iestatījumos būs jāatļauj arī SSH savienojumi.

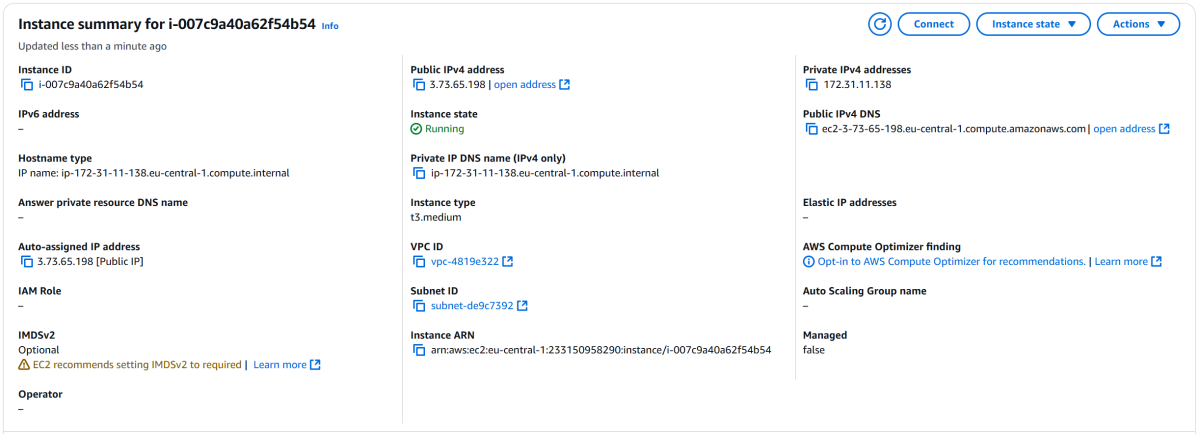


Visbeidzot noklikšķiniet uz pogas *Launch*. Apsveicam! Jūsu Zabbix starpniekserveris tagad tiek izvietots.



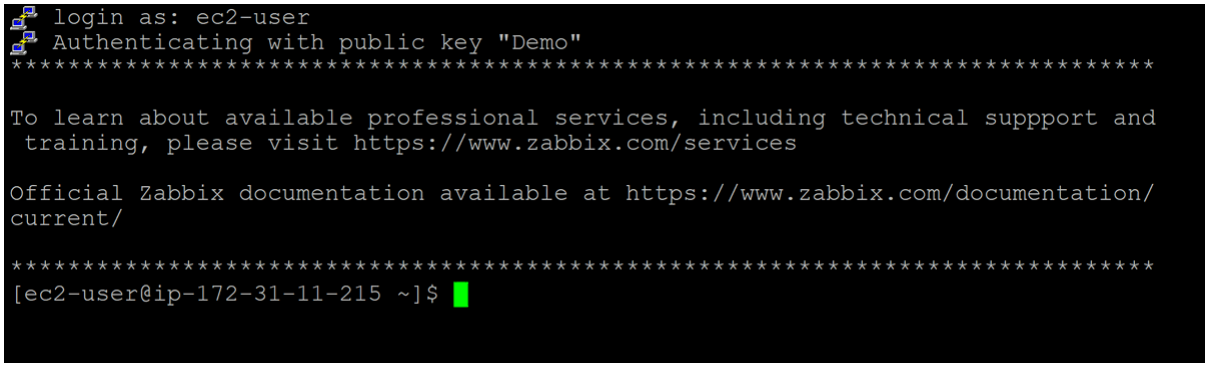
Zabbix starpniekservera konfigurēšana un piekļuve

EC2 konsolē atveriet sadaļu Instances un atlasiet tikko izvietoto Zabbix starpniekservera instanci.



Izmantojot SSH, izveidojiet savienojumu ar Zabbix starpniekservera publisko IPv4 adresi vai publisko IPv4 DNS. Noklusējuma SSH lietotājvārds ir `rocky`. Neaizmirstiet izmantot iepriekš ģenerēto atslēgu pāri.

Akreditācijas datu ģenerēšana var ilgt dažas minūtes. Pirms mēģināt izveidot savienojumu, uzgaidiet dažas minūtes.



Zabbix starpniekservera konfigurācijas fails atrodas `/etc/zabbix/zabbix_proxy.conf`.

Konfigurācijas failu var atvērt ar jebkuru teksta redaktoru, piemēram, vi:

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

Sākotnējā starpniekservera konfigurācija ietver parametru `ProxyMode`, `Server` un `Hostname` modificēšanu Zabbix starpniekservera konfigurācijas failā.

Starpniekservera režīmu var mainīt, rediģējot parametru `ProxyMode`. Aktīvajam režīmam iestatiet `ProxyMode=0`, bet pasīvajam režīmam — `ProxyMode=1`.

Parametram `Server` jāietver Zabbix servera IP adrese. Piemēram:

```
Server=192.168.50.141
```

Parametram `Hostname` jāietver Zabbix starpniekservera resursdatora nosaukums. Šeit definētajam resursdatora nosaukumam jāatbilst starpniekservera nosaukumam, kas norādīts starpniekservera konfigurācijā Zabbix lietotāja saskarnē. Piemēram:

```
Hostname=EC2-Frankfurt
```

Kad konfigurēšana ir pabeigta, restartējiet Zabbix starpniekserveri, izmantojot šo komandu:

```
sudo systemctl restart zabbix-proxy
```

Piezīme: izmantojiet sudo, lai rediģētu konfigurācijas failus un izpildītu komandrindas komandas.

```
### Option: ProxyMode
#       Proxy operating mode.
#       0 - proxy in the active mode
#       1 - proxy in the passive mode
#
# Mandatory: no
# Default:
# ProxyMode=0

### Option: Server
#       If ProxyMode is set to active mode:
#               IP address or DNS name of Zabbix server to get configuration data from and send data to.
#       If ProxyMode is set to passive mode:
#               List of comma delimited IP addresses, optionally in CIDR notation, or DNS names of Zabbix
server.
#               Incoming connections will be accepted only from the addresses listed here.
#               If IPv6 support is enabled then '127.0.0.1', '::127.0.0.1', '::ffff:127.0.0.1' are treated
 equally
#               and '::/0' will allow any IPv4 or IPv6 address.
#               '0.0.0.0/0' can be used to allow any IPv4 address.
#               Example: Server=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com
#
# Mandatory: yes
# Default:
# Server=

Server=127.0.0.1
```

```
### Option: Hostname
#       Unique, case sensitive Proxy name. Make sure the Proxy name is known to the server!
#       Value is acquired from HostnameItem if undefined.
#
# Mandatory: no
# Default:
# Hostname=

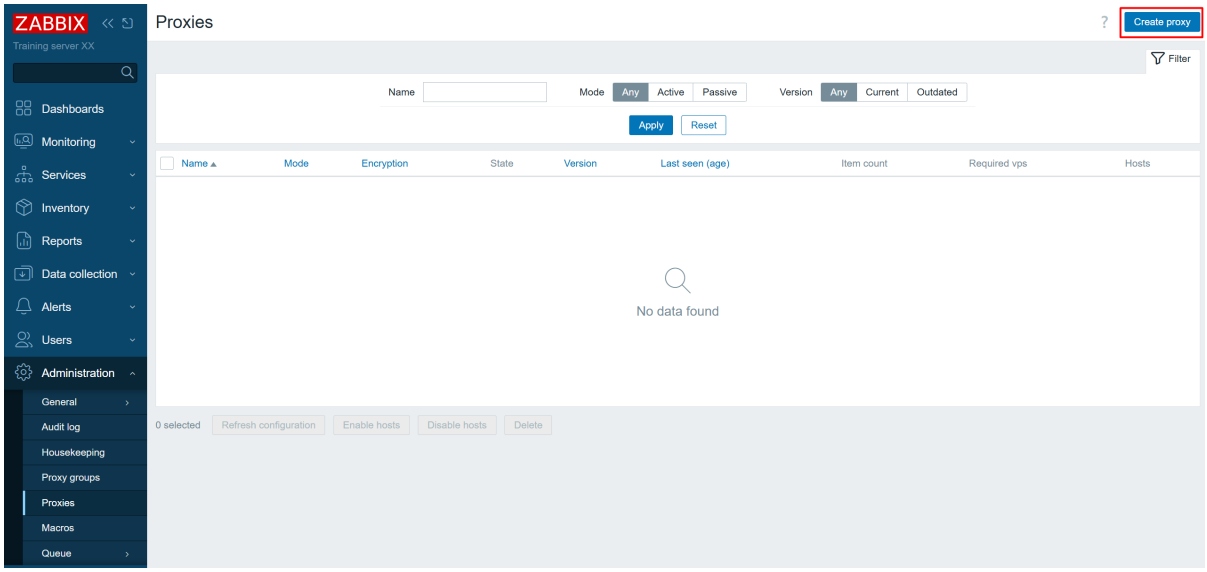
Hostname=EC2-Frankfurt
```

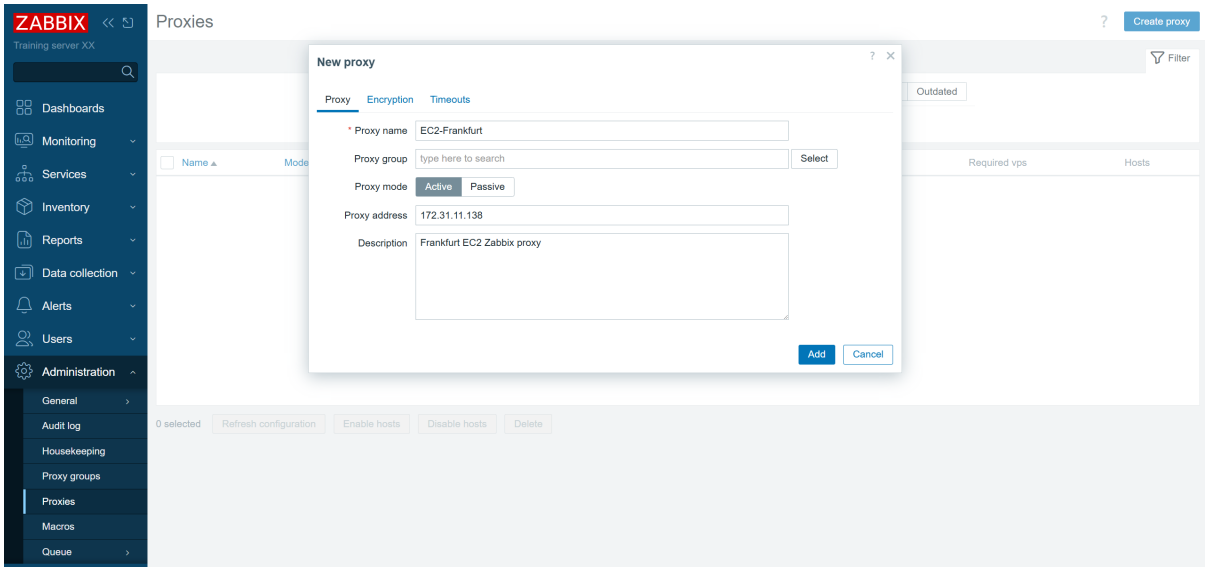
Atveriet Zabbix lietotāja saskarni un pārejiet uz *Administration* > *Proxies*. Noklikšķiniet uz pogas *Create proxy*.

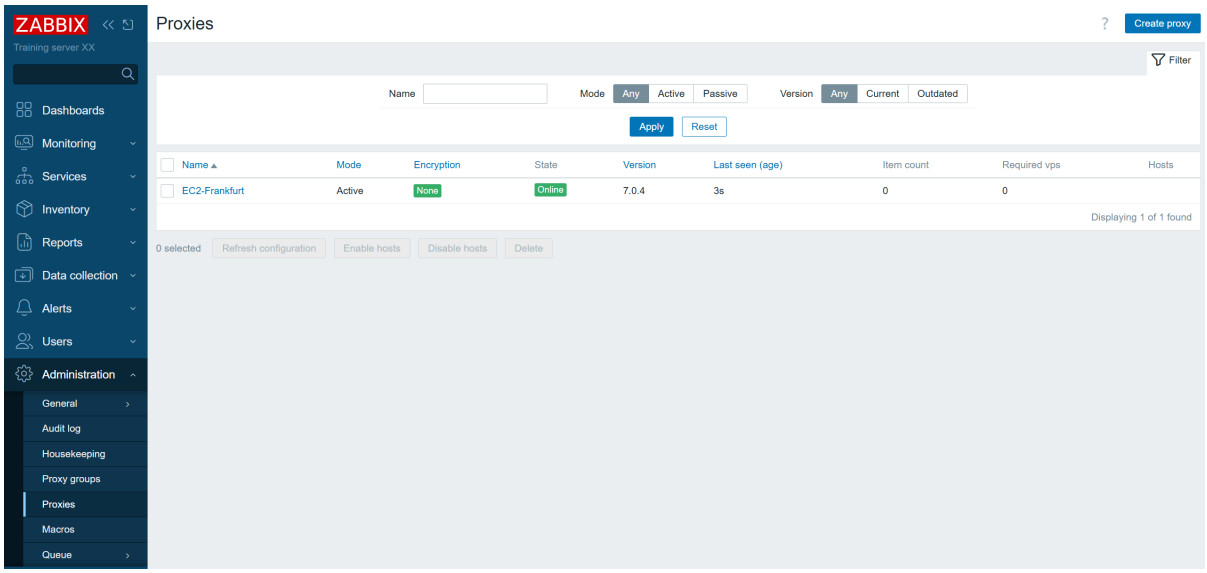
Ievadiet to pašu starpniekservera nosaukumu, kas konfigurācijas failā norādīts parametra `Hostname` vērtībā.

Atlasiet starpniekservera režīmu un norādiet starpniekservera IP adresi (aktīviem starpniekserveriem nav obligāti).

Noklikšķiniet uz *Create proxy*.







Starpniekservera datubāzes atrašanās vieta

SQLite starpniekservera datubāze tiek glabāta `/tmp/zabbix_proxy.sqlite3`, un tās atrašanās vieta ir norādīta ar parametru `DBName` Zabbix starpniekservera konfigurācijas failā.

Azure izvietošanas rokasgrāmata Zabbix serverim

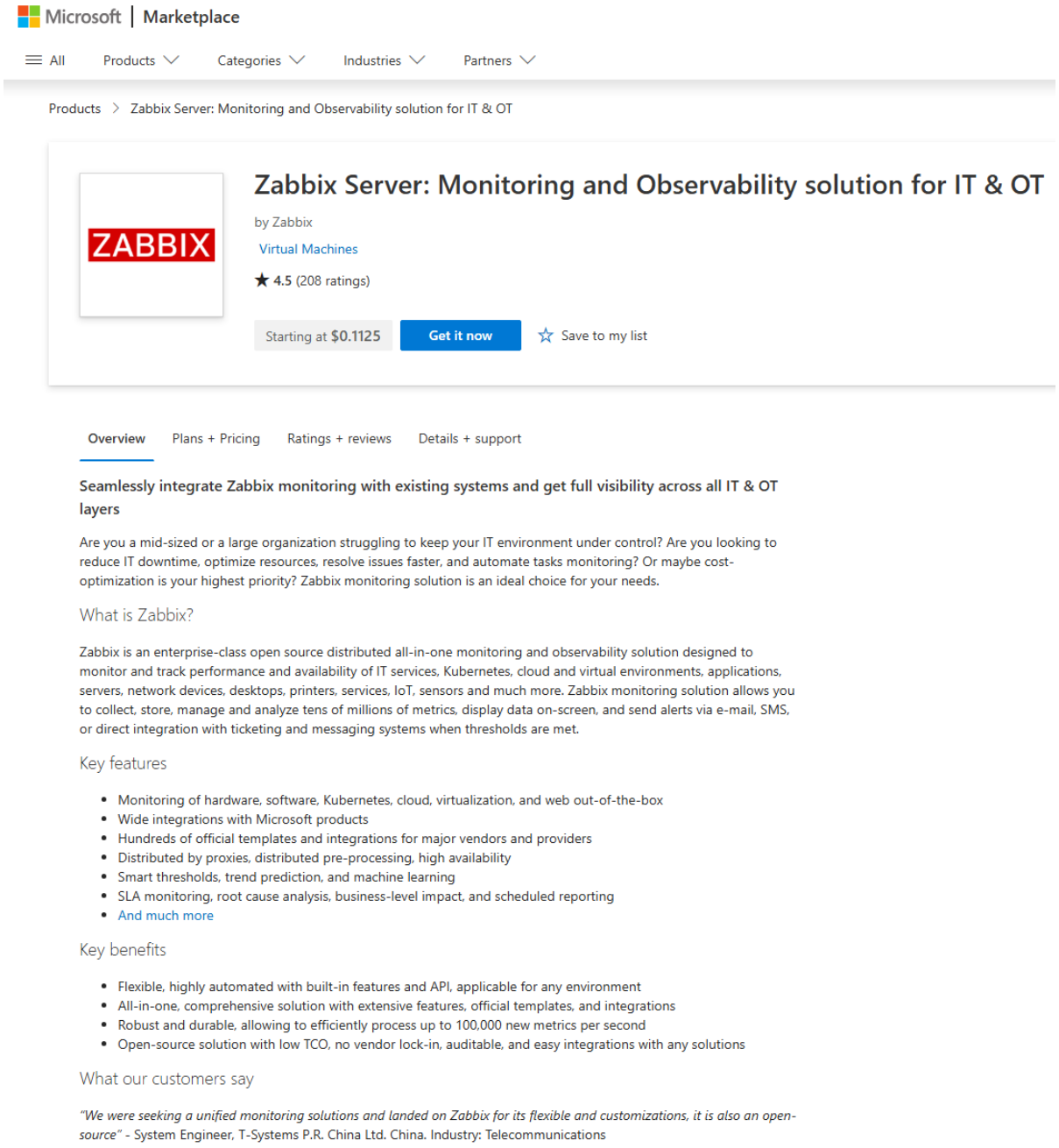
Pārskats

Šajā lapā aprakstīts, kā izvietot Zabbix serveri, izmantojot Azure.

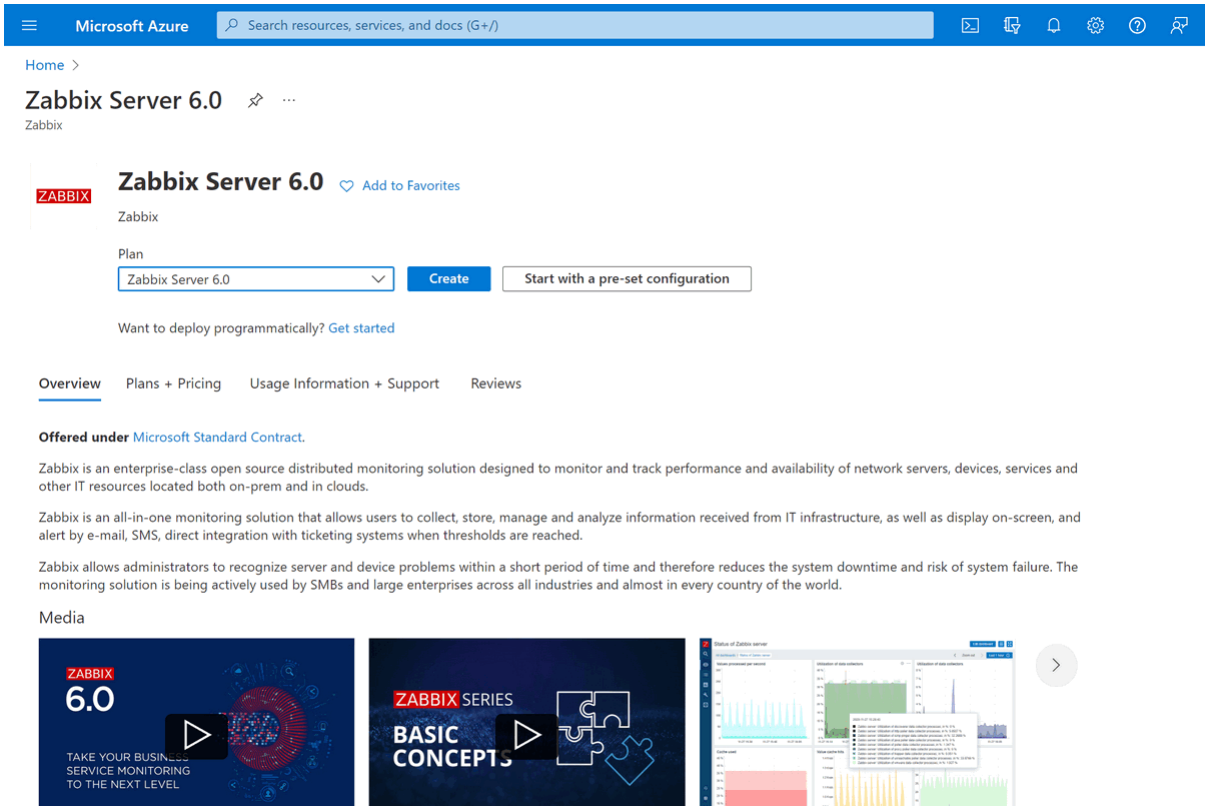
Attention:
Platformā izvietošanai ir pieejamas vairākas Zabbix versijas. Noteikti izlasiet tās pašas Zabbix versijas dokumentāciju, kuru esat izvietojis. Dokumentāciju dažādām Zabbix versijām var atlasīt versiju nolaižamajā izvēlnē.

Ierīces izvietošana

Lai izvietotu Zabbix ierīci, vispirms atveriet Zabbix Azure Marketplace lapu. Pārskatiet kopsavilkumu un pēc tam noklikšķiniet uz *Get it now*.



Pārskatiet izvietošanas kopsavilkumu, cenu plānus un citu informāciju. Pēc tam noklikšķiniet uz *Create* vai *Start with a pre-set configuration*, lai izvietotu Zabbix ierīci.

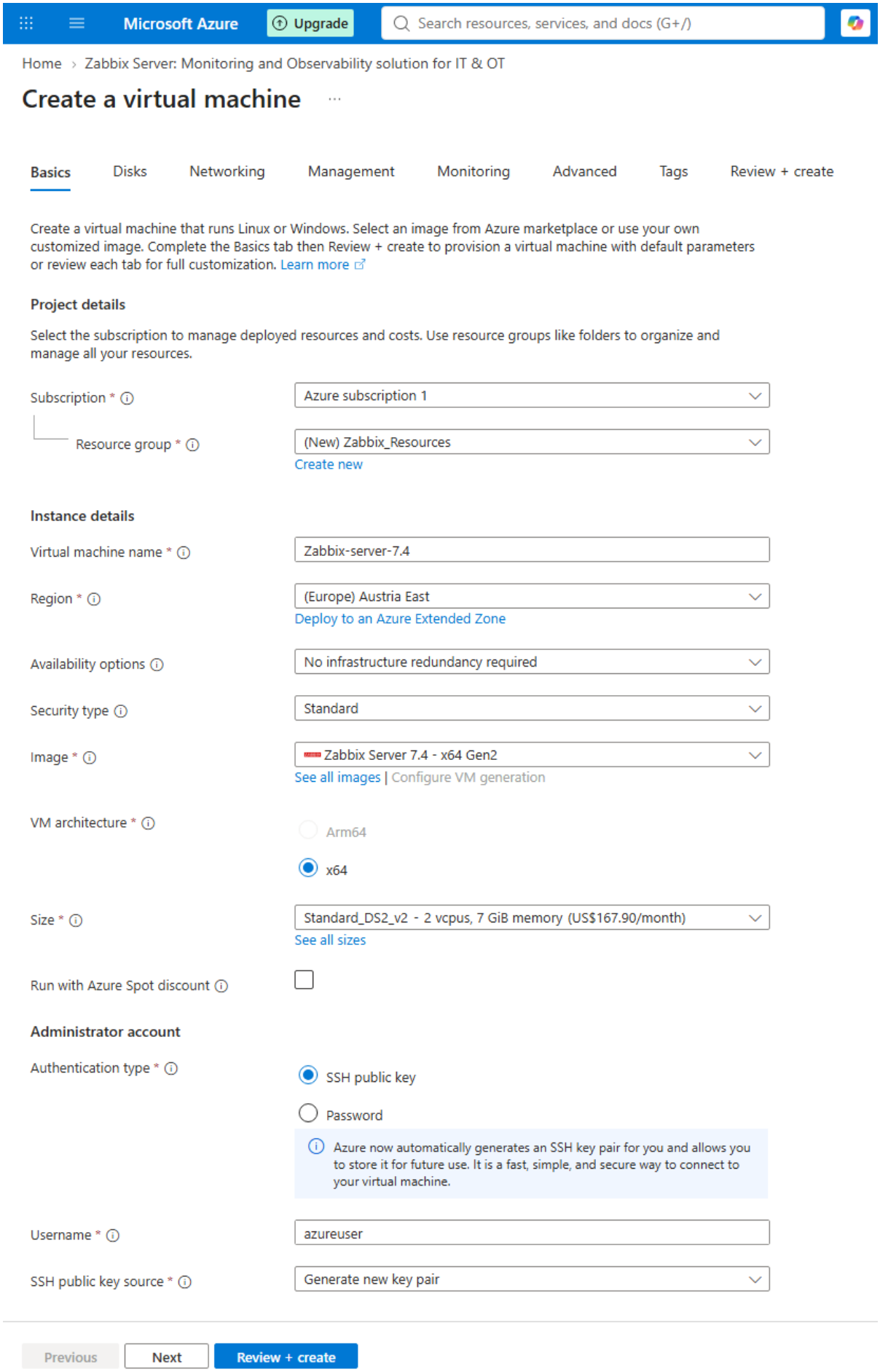


Konfigurējiet virtuālo mašīnu, atlasot virtuālās mašīnas izmēru, izveidojot administratora kontu un definējot diska, tīkla, pārvaldības un citus iestatījumus. Šie iestatījumi var atšķirties atkarībā no jūsu infrastruktūras prasībām.

Kā piemēru varat izmantot tālāk norādīto konfigurāciju:

Basics cilne:

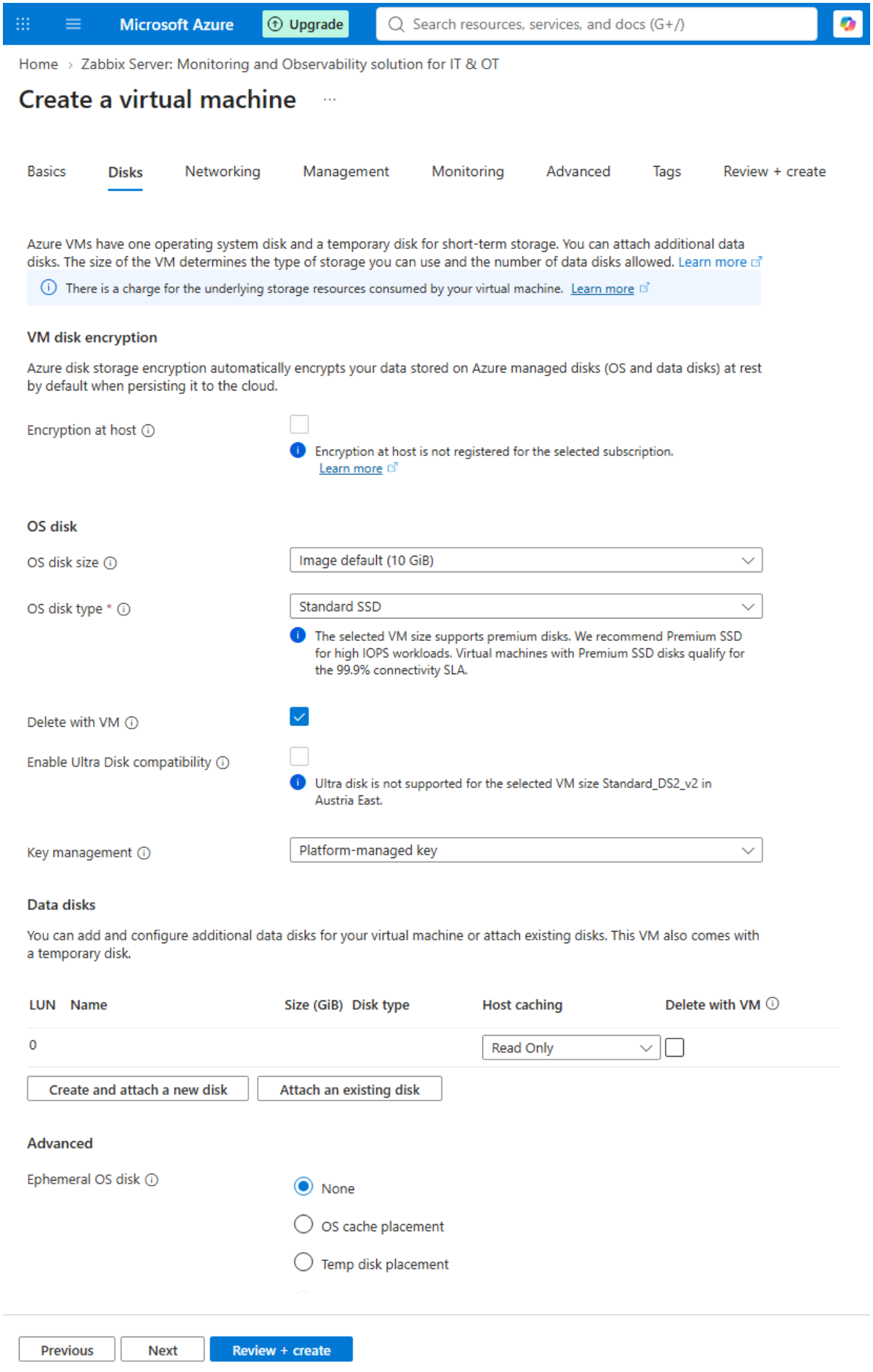
- *Subscription*: Azure subscription 1
- *Resource group*: Zabbix_Resources
- *Virtual machine name*: Zabbix-server--7.4
- *Availability options*: No infrastructure redundancy required
- *Security type*: Standard
- *Image*: Zabbix Server 7.4 - x64 Gen2
- *Size*: Standard_DS2_V2 - 2 vcpus, 7 GiB memory
- *Authentication type*: SSH public key
- *Username*: azureuser
- *SSH public key source*: Generate new key pair



Disks cilne:

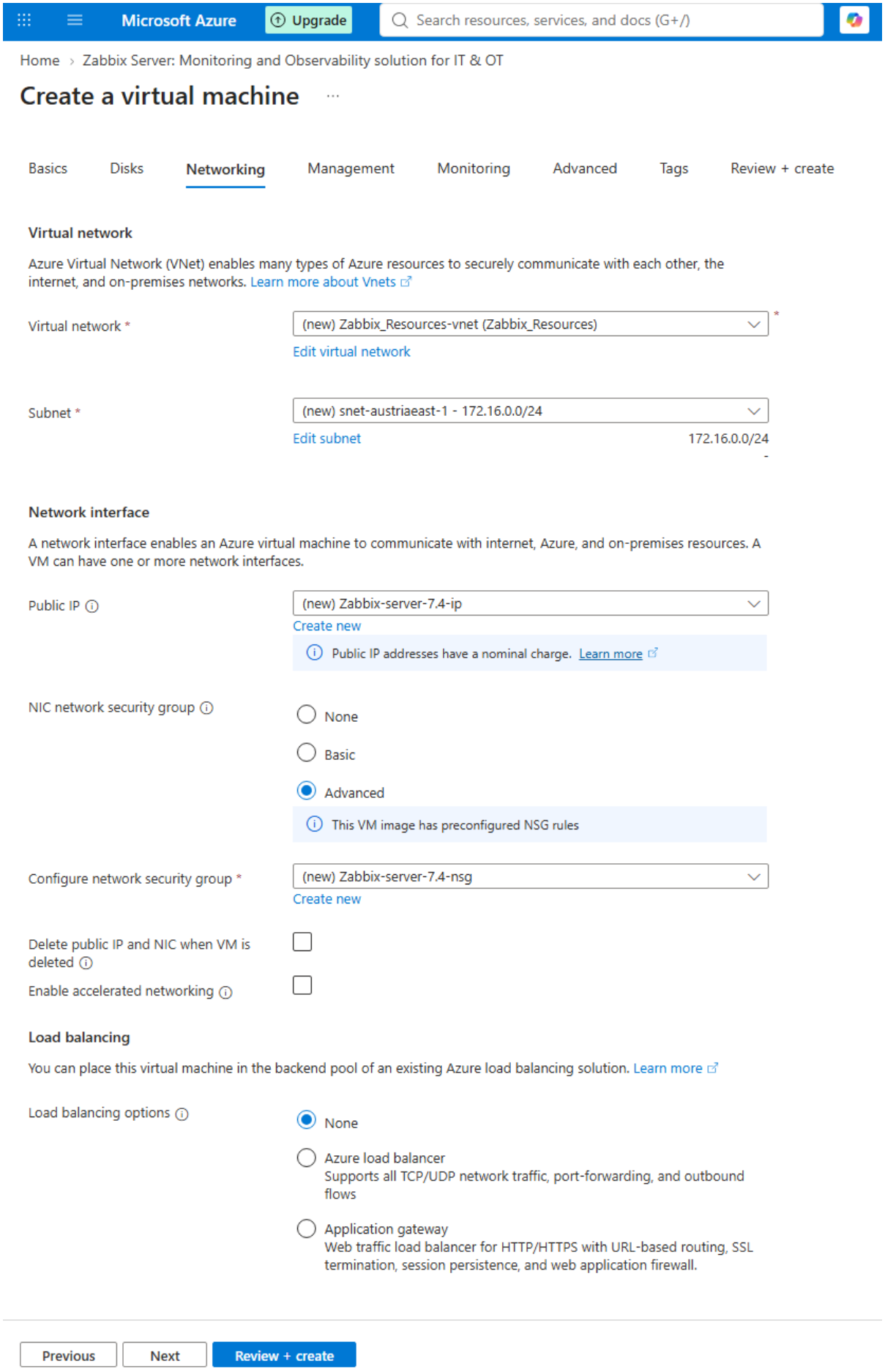
- *Encryption at host*: nav atzīmēts
- *OS disk type*: Standard SSD

- *Delete with VM*: atzīmēts
- *Enable Ultra Disk compatibility*: nav atzīmēts

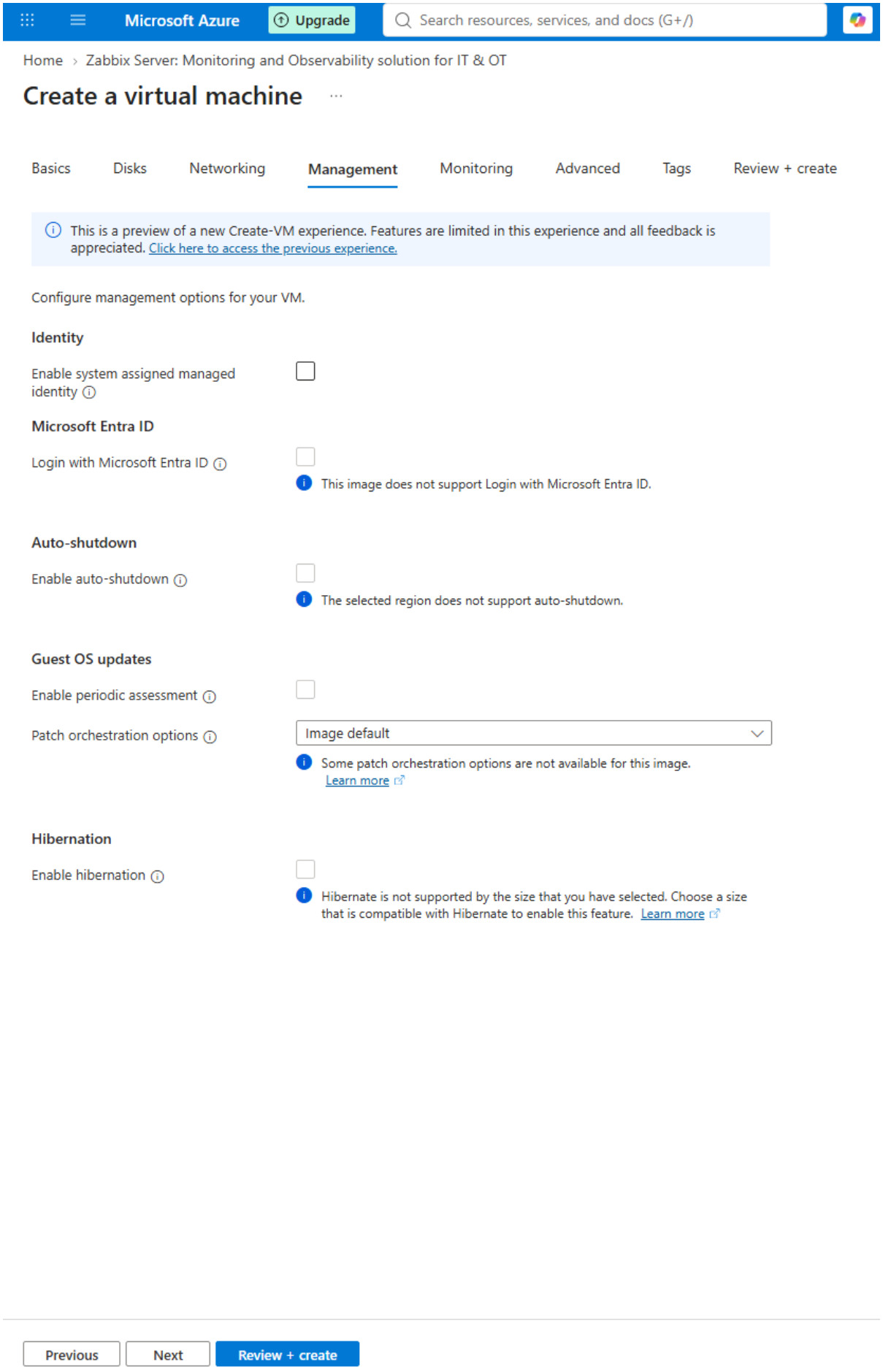


Networking cilne:

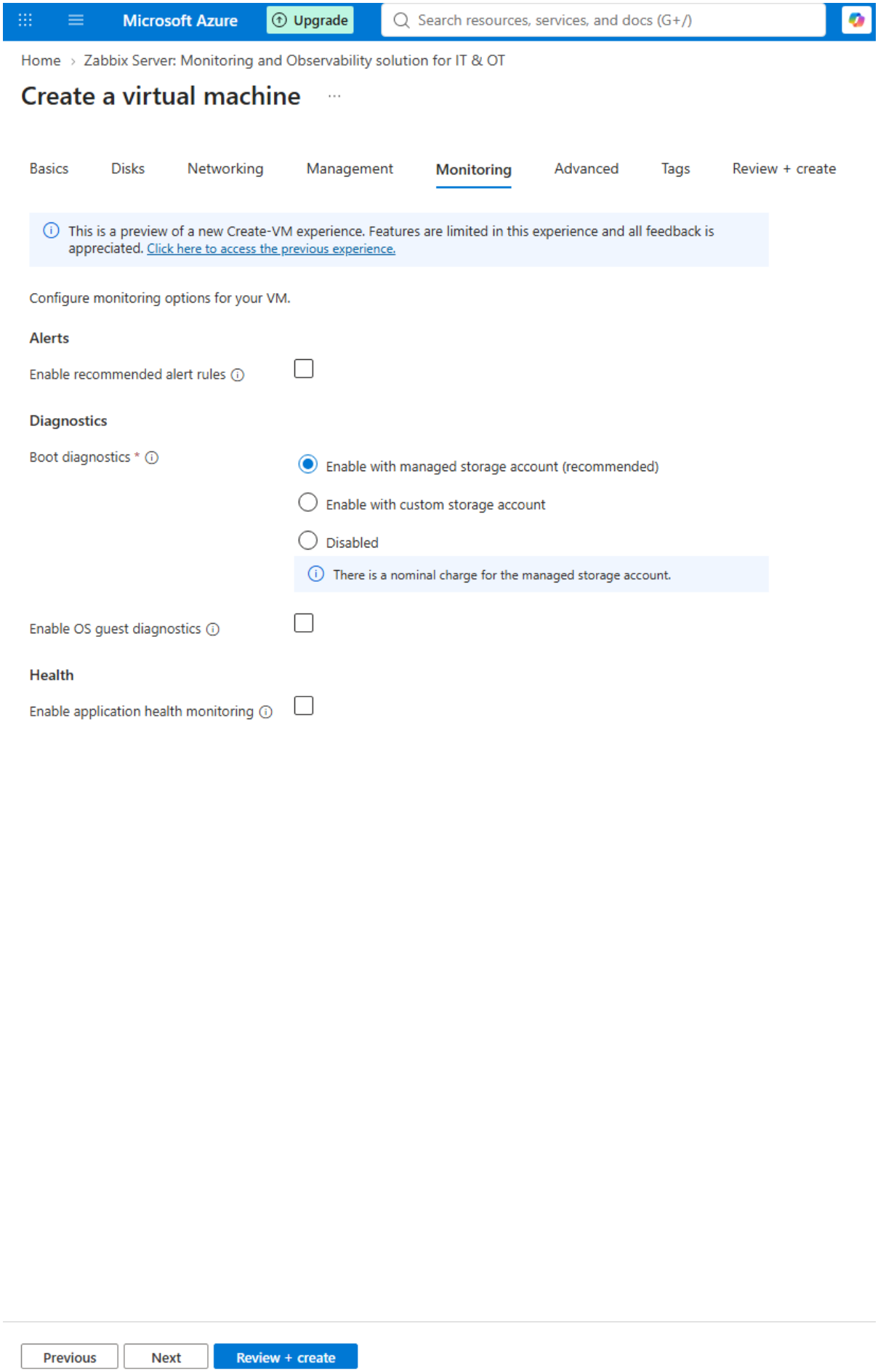
- *Virtual network*: Zabbix_Resources-vnet
- *Public IP*: Zabbix-server-7.4-ip
- *NIC network security group*: Advanced
- *Configure network security group*: Zabbix-server-7.4-nsg



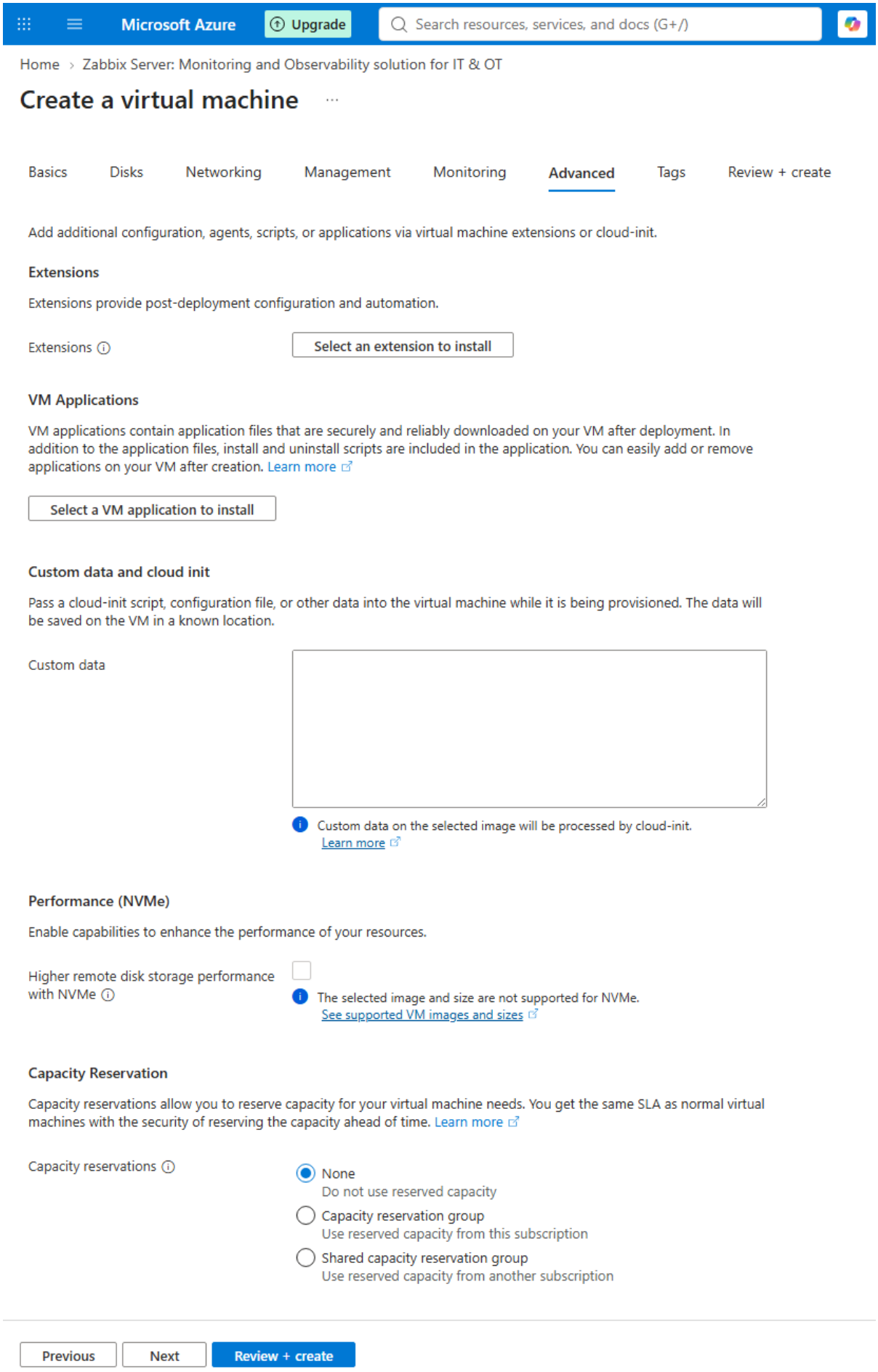
Management cilnē parametrus var atstāt pēc noklusējuma.



Monitoring cilnē parametrus var atstāt pēc noklusējuma.



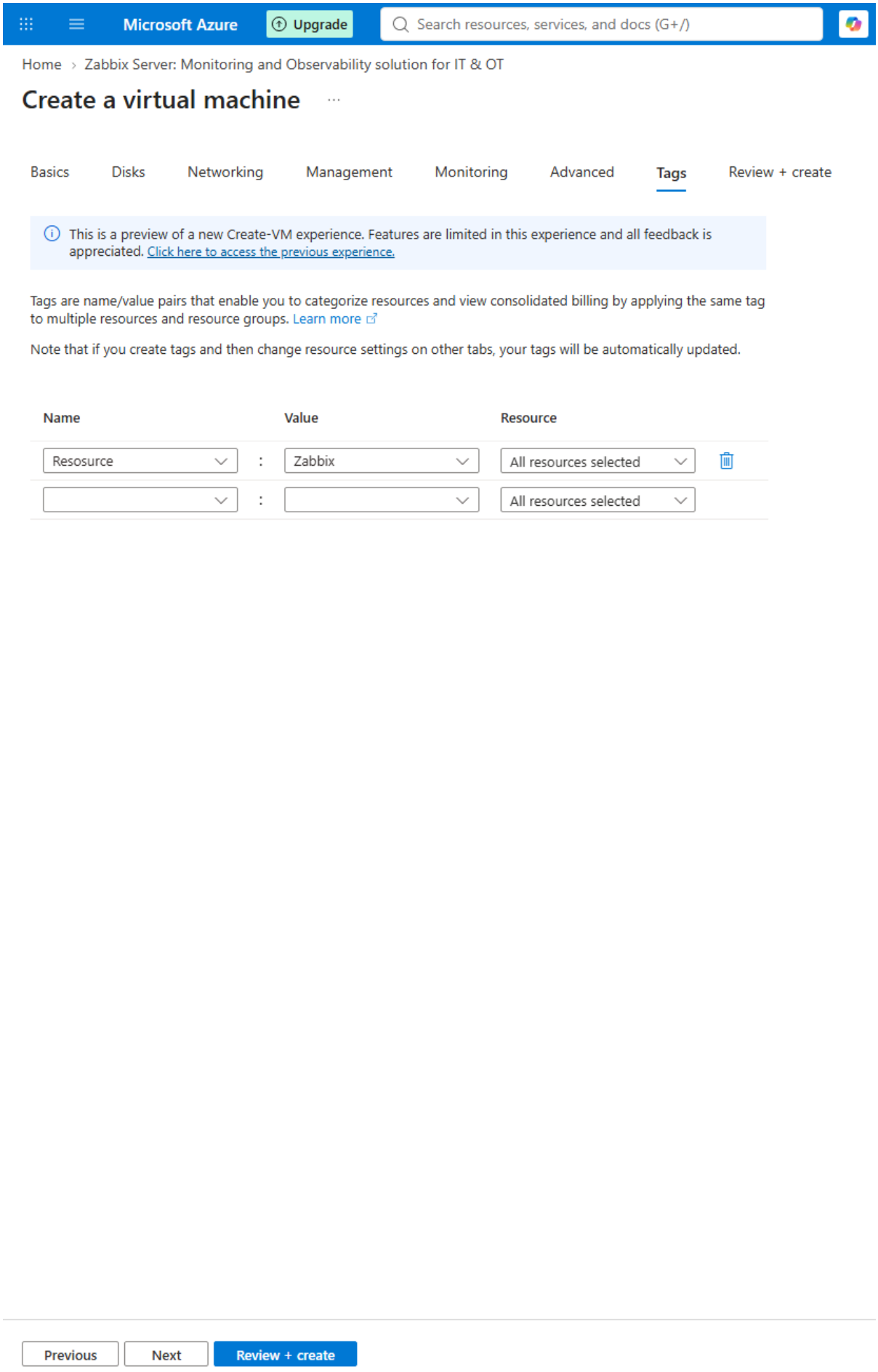
Advanced cilnē parametrus var atstāt pēc noklusējuma.



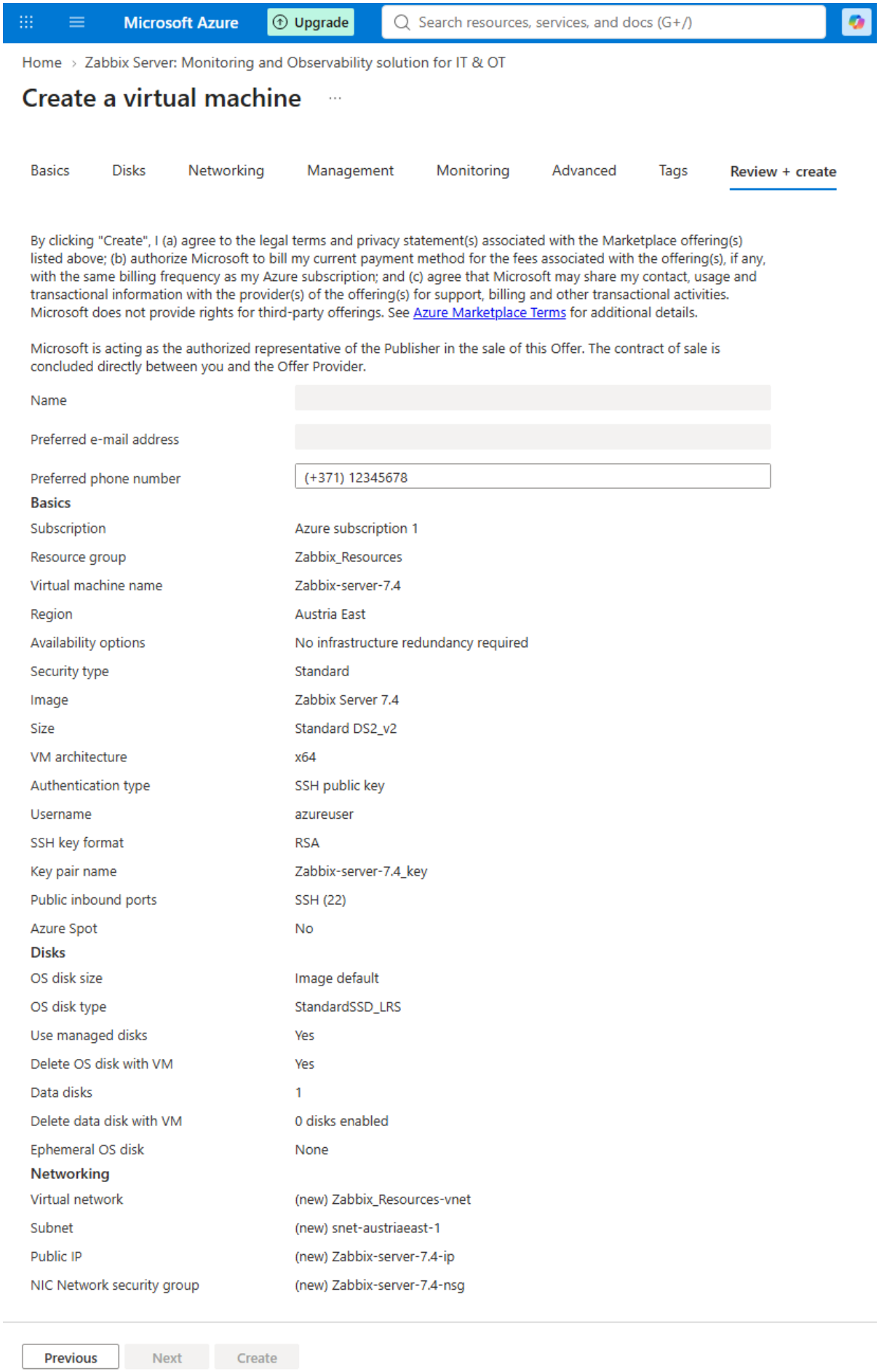
Tags cilne:

- *Name*: Resource
- *Value*: Zabbix

- *Resource*: All resource selected



Apstipriniet virtuālās mašīnas informāciju un izveidojiet virtuālo mašīnu.



Jūs tiksit novirzīts uz virtuālās mašīnas izvietošanas statusa lapu. Uzgaidiet, līdz izvietošana ir pabeigta.

Kad izvietošana ir pabeigta, atveriet resursu un pierakstiet tā publisko IP adresi. Tagad jums būs jāpieslēdzas Zabbix serverim, izmantojot SSH (vai citu pieejamu metodi), lai iegūtu Zabbix lietotāja saskarnes pieteikšanās akreditācijas datus.

```
********************************************************************************

Zabbix frontend credentials:

Username: Admin

Password: zfpvagcs0s+srUUK+EM

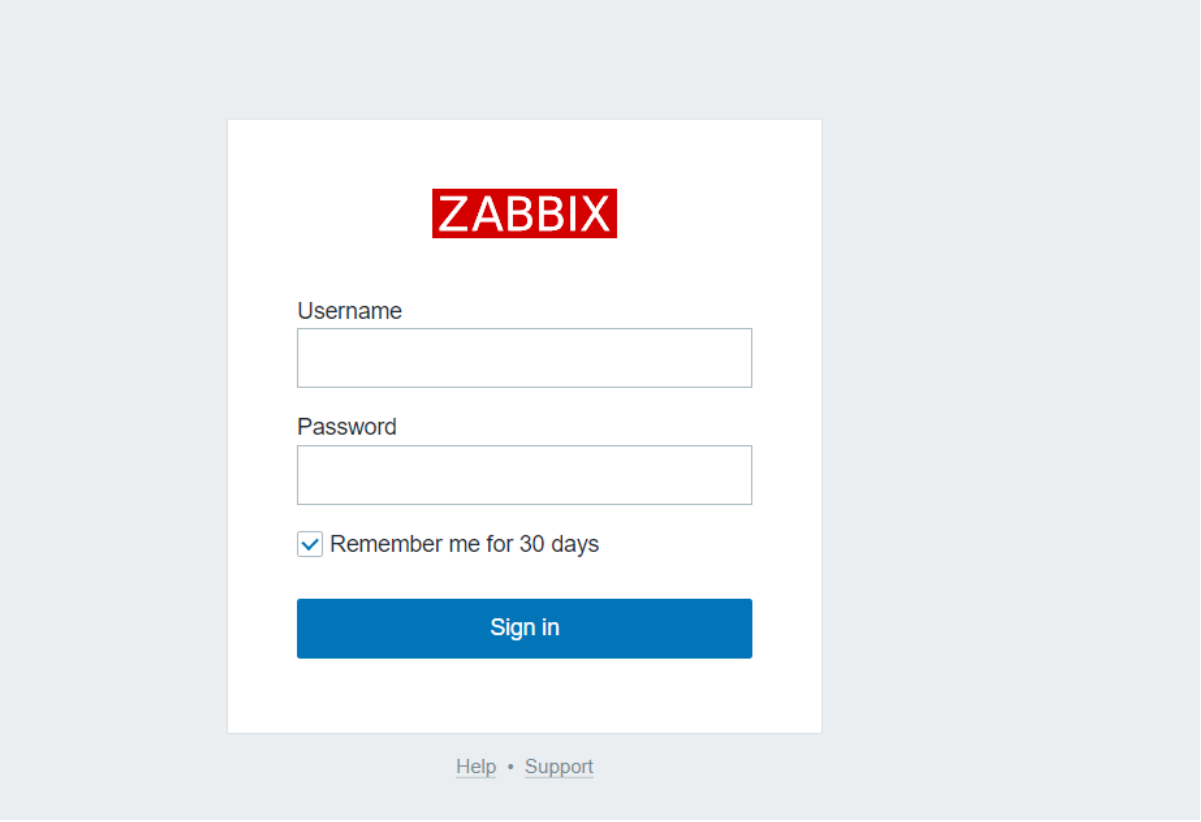

To learn about available professional services, including technical suppport and
 training, please visit https://www.zabbix.com/services

Official Zabbix documentation available at https://www.zabbix.com/documentation/
current/

Note! Do not forget to change timezone PHP variable in /etc/php-fpm.d/zabbix.con
f file.


********************************************************************************
[azureuser@Zabbix-server-6 ~]$
```

Piekļūstiet Zabbix lietotāja saskarnei, tīmekļa pārlūkprogrammā atverot Zabbix ierīces publisko IP adresi. Lai pieteiktos, izmantojiet iepriekšējā darbībā iegūto lietotājvārdu un paroli.



Piekļuve datubāzei

Zabbix datubāzei var piekļūt, izmantojot lietotāju `root`, neievadot paroli. Akreditācijas dati tiek glabāti konfigurācijas failā `/root/.my.cnf`.

Azure izvietošanas rokasgrāmata Zabbix starpniekserverim

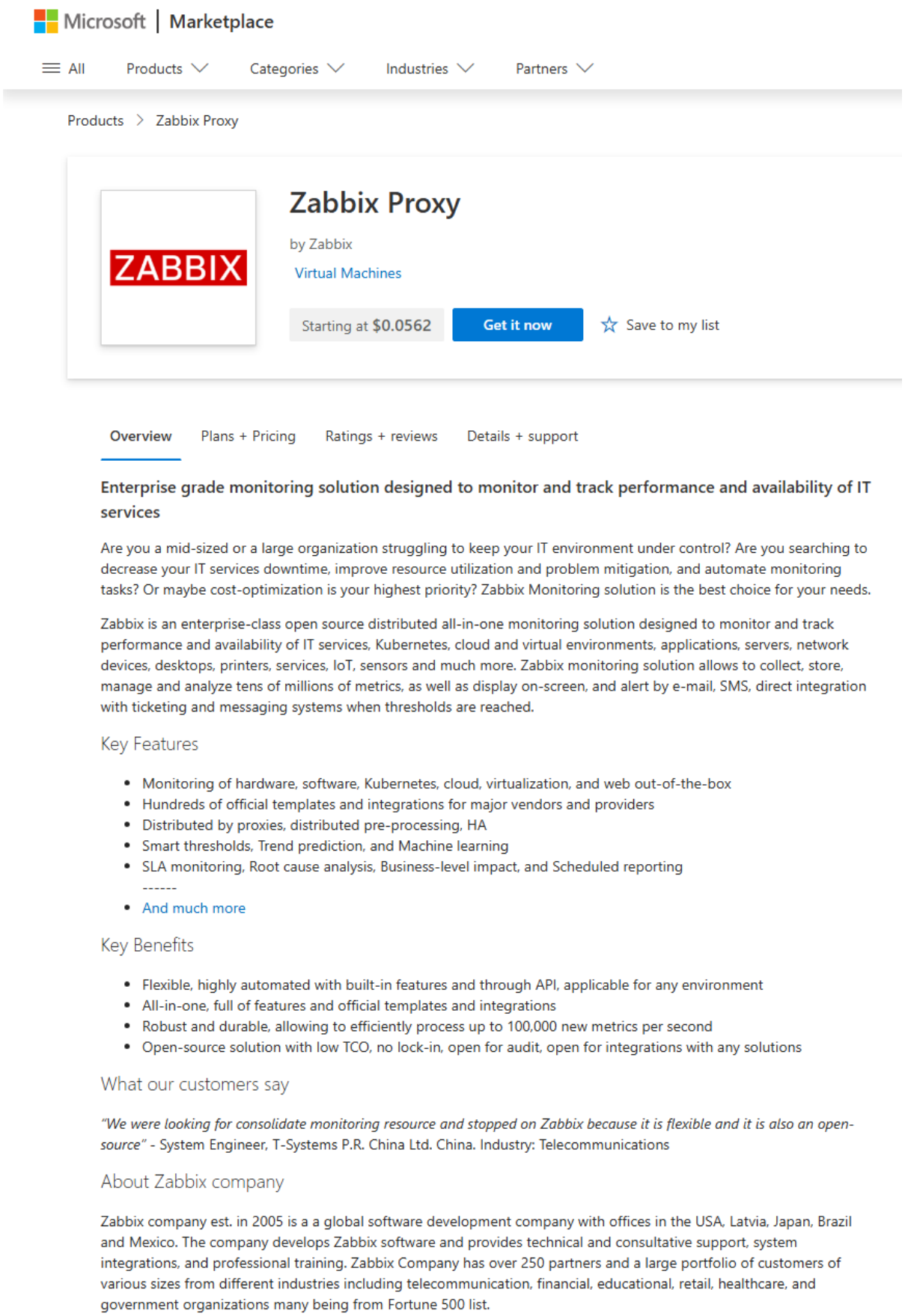
Pārskats

Šajā lapā aprakstīts, kā izvietot Zabbix starpniekserveri, izmantojot Azure.

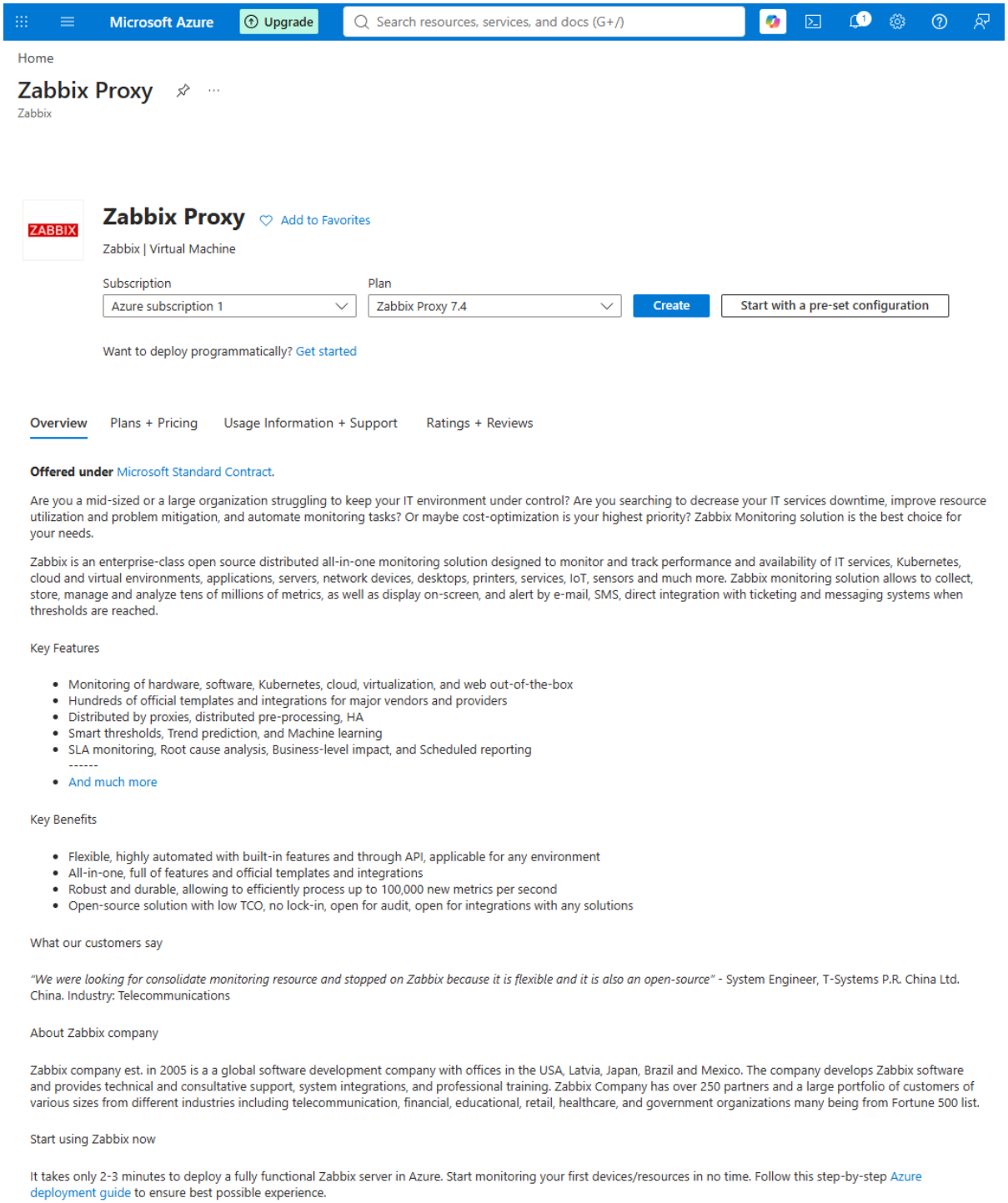
Attention:
Platformā izvietošanai ir pieejamas vairākas Zabbix versijas. Noteikti izlasiet tās pašas Zabbix versijas dokumentāciju, kuru esat izvietojis. Dokumentāciju dažādām Zabbix versijām var atlasīt versiju nolaižamajā izvēlnē.

Zabbix starpniekservera izvietošana

Lai izvietotu Zabbix starpniekserveri, vispirms atveriet Zabbix Azure marketplace lapu. Noklikšķiniet uz *Get it now*.



Šeit varat izlasīt papildu informāciju par pašreizējo izvietošanu, piemēram, vispārīgu pārskatu, cenu plānus un citu informāciju. Noklikšķiniet uz *Create* vai *Start with a pre-set configuration*.

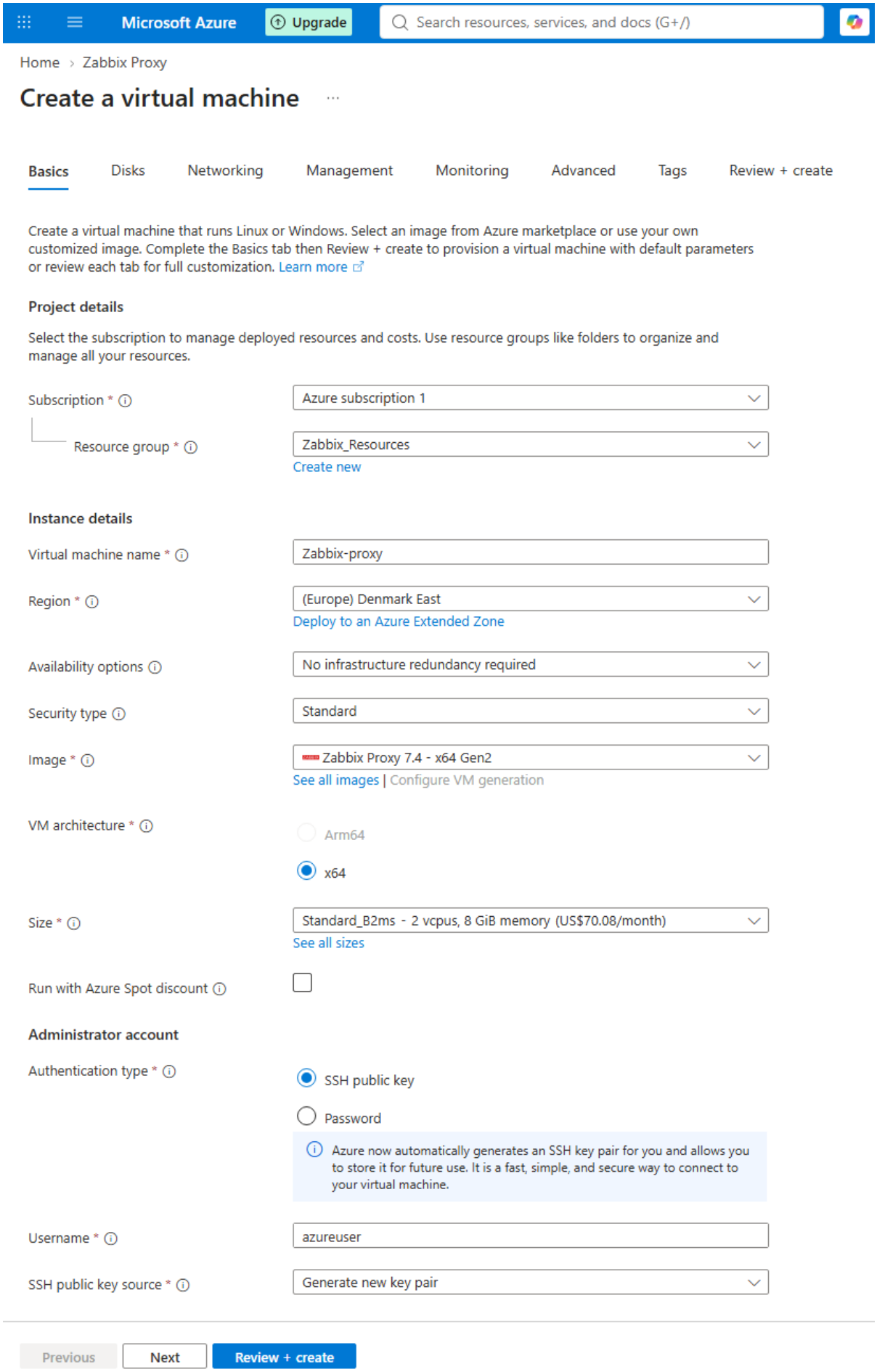


Turpiniet ar detalizētu virtuālās mašīnas konfigurēšanu: atlasiet virtuālās mašīnas lielumu, administratora kontu, disku, tīklu, pārvaldību un citus iestatījumus. Šie iestatījumi var atšķirties atkarībā no jūsu infrastruktūras prasībām.

Kā piemēru varat izmantot tālāk norādīto konfigurāciju:

Cilne *Basics*:

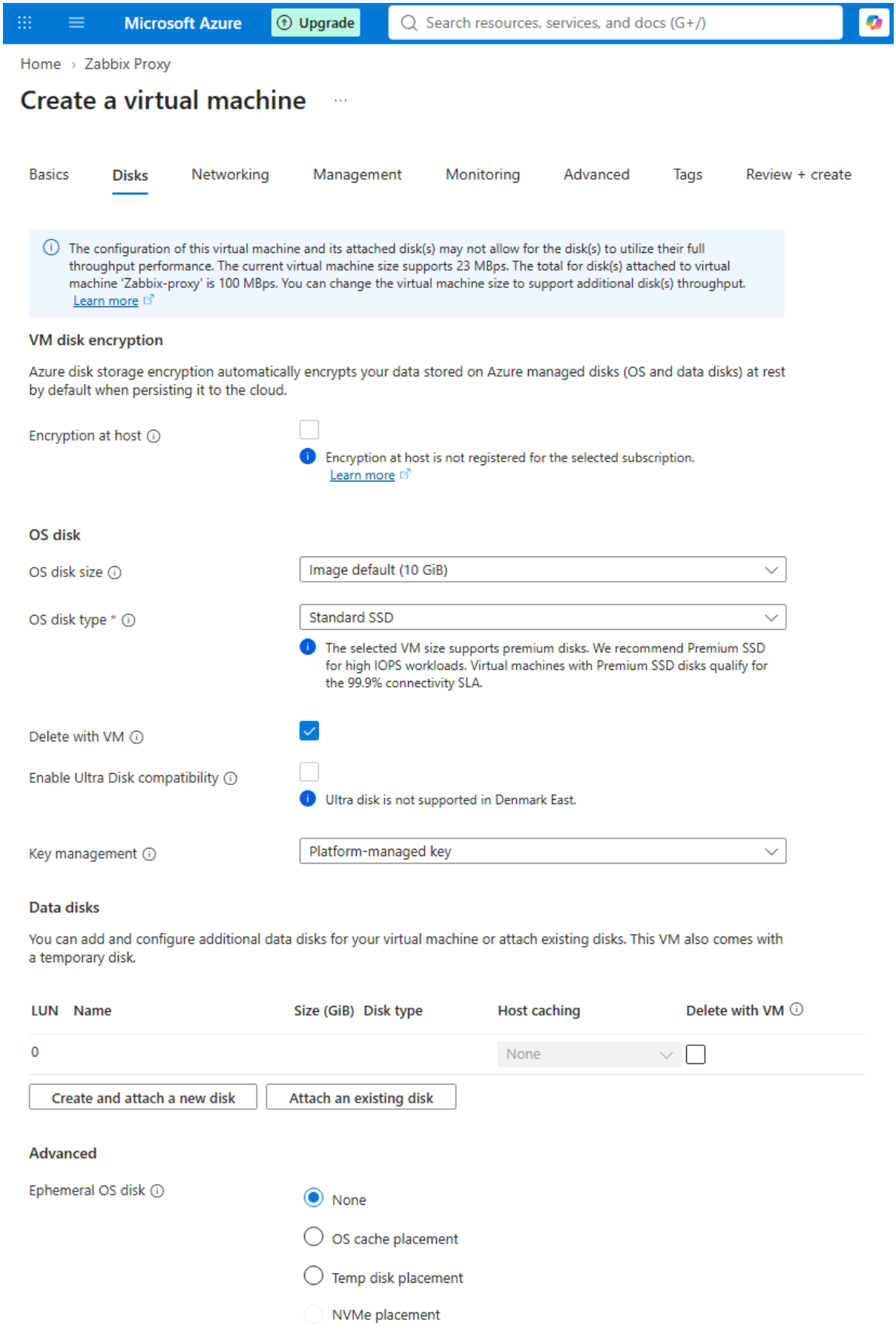
- *Subscription*: Azure subscription 1
- *Resource group*: Zabbix_Resources
- *Virtual machine name*: Zabbix-proxy
- *Availability options*: No infrastructure redundancy required
- *Security type*: Standard
- *Image*: Zabbix Proxy 7.4 - x64 Gen2
- *Size*: Standard_B2ms - 2 vcpus, 8 GiB memory
- *Authentication type*: SSH public key
- *Username*: azureuser
- *SSH public key source*: Generate new key pair



Cilne *Disks*:

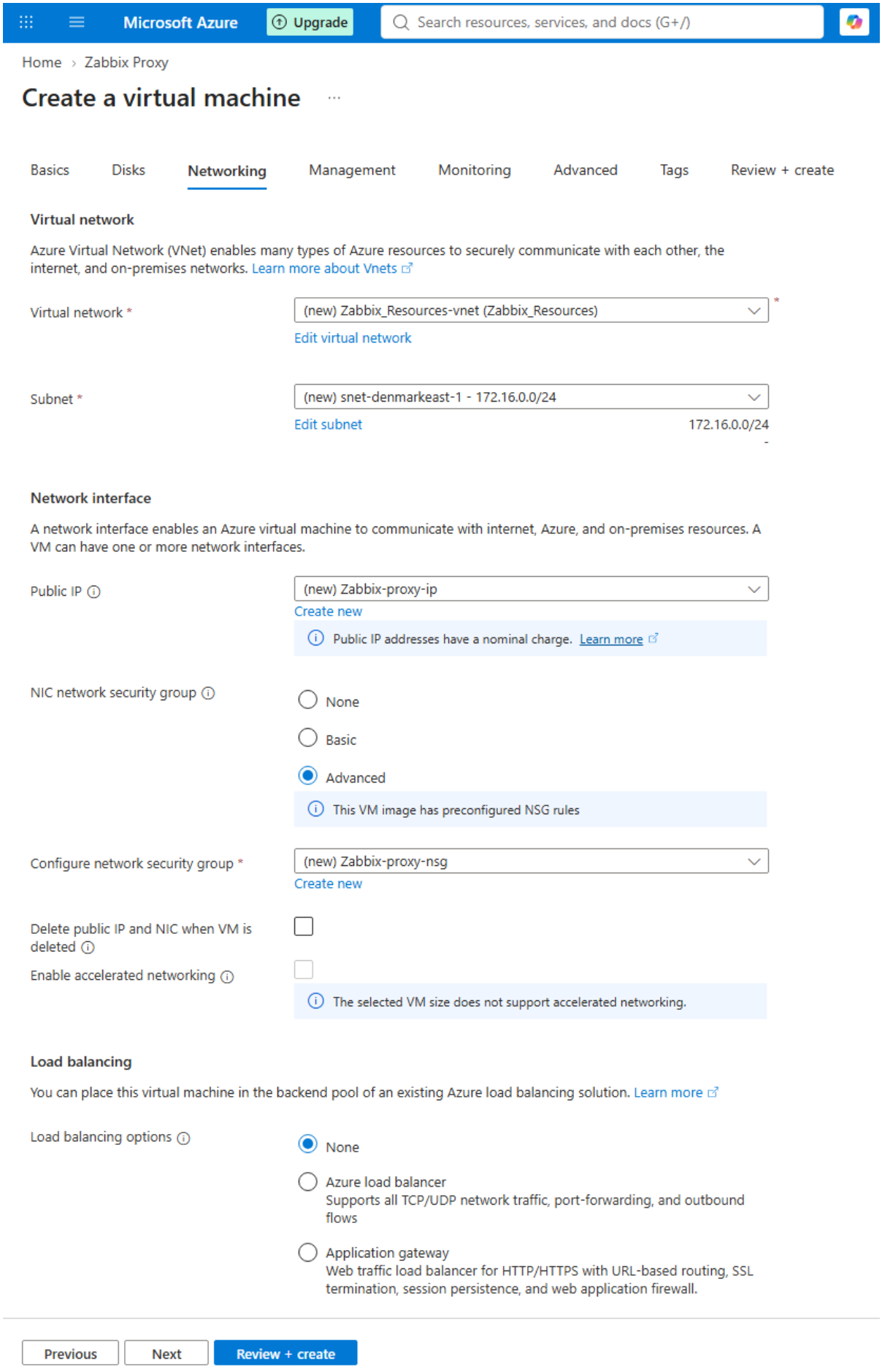
- *Encryption at host*: unchecked

- *OS disk type*: Standard SSD
- *Delete with VM*: checked
- *Enable Ultra Disk compatibility*: unchecked

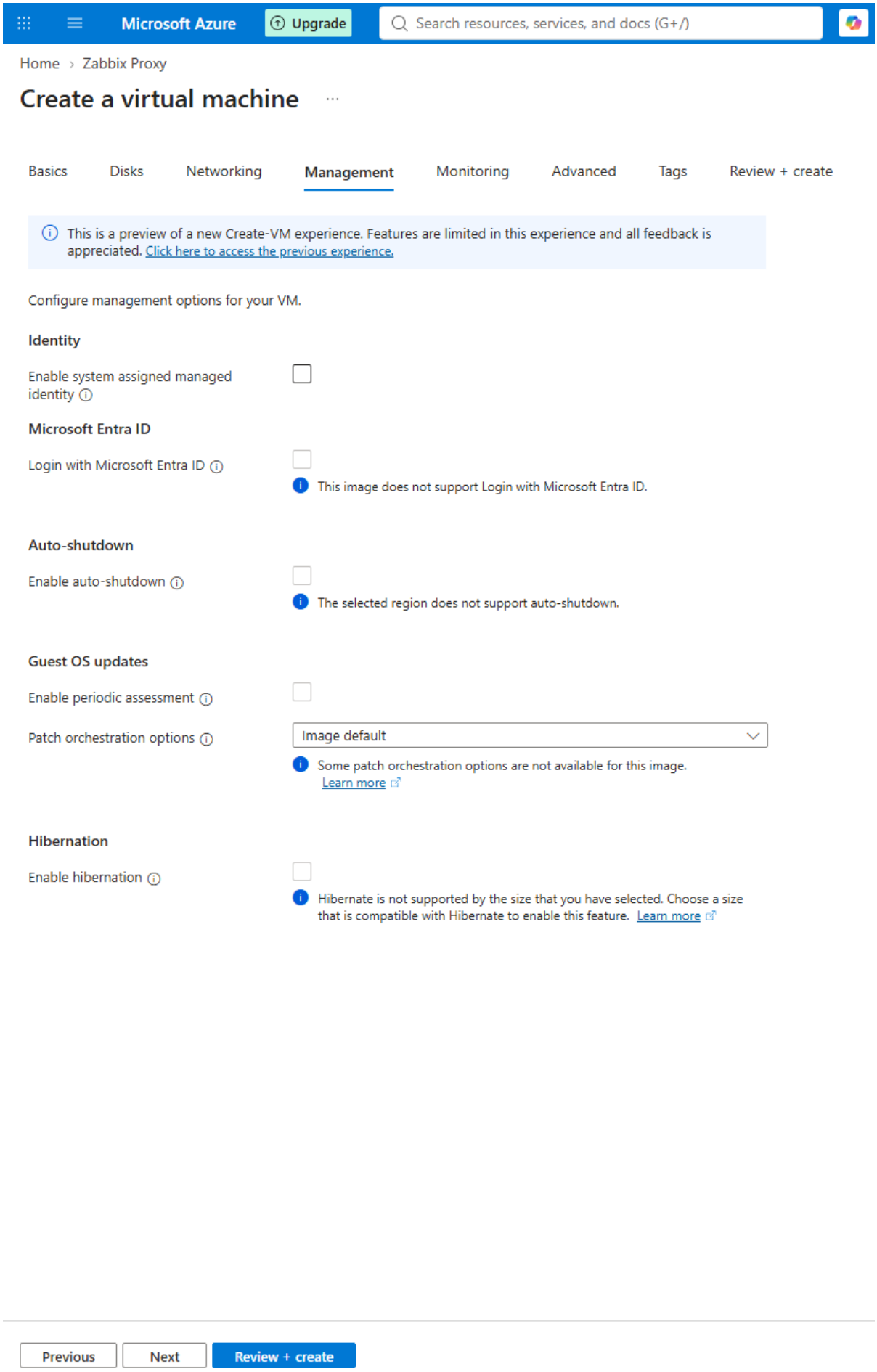


Cilne *Networking*:

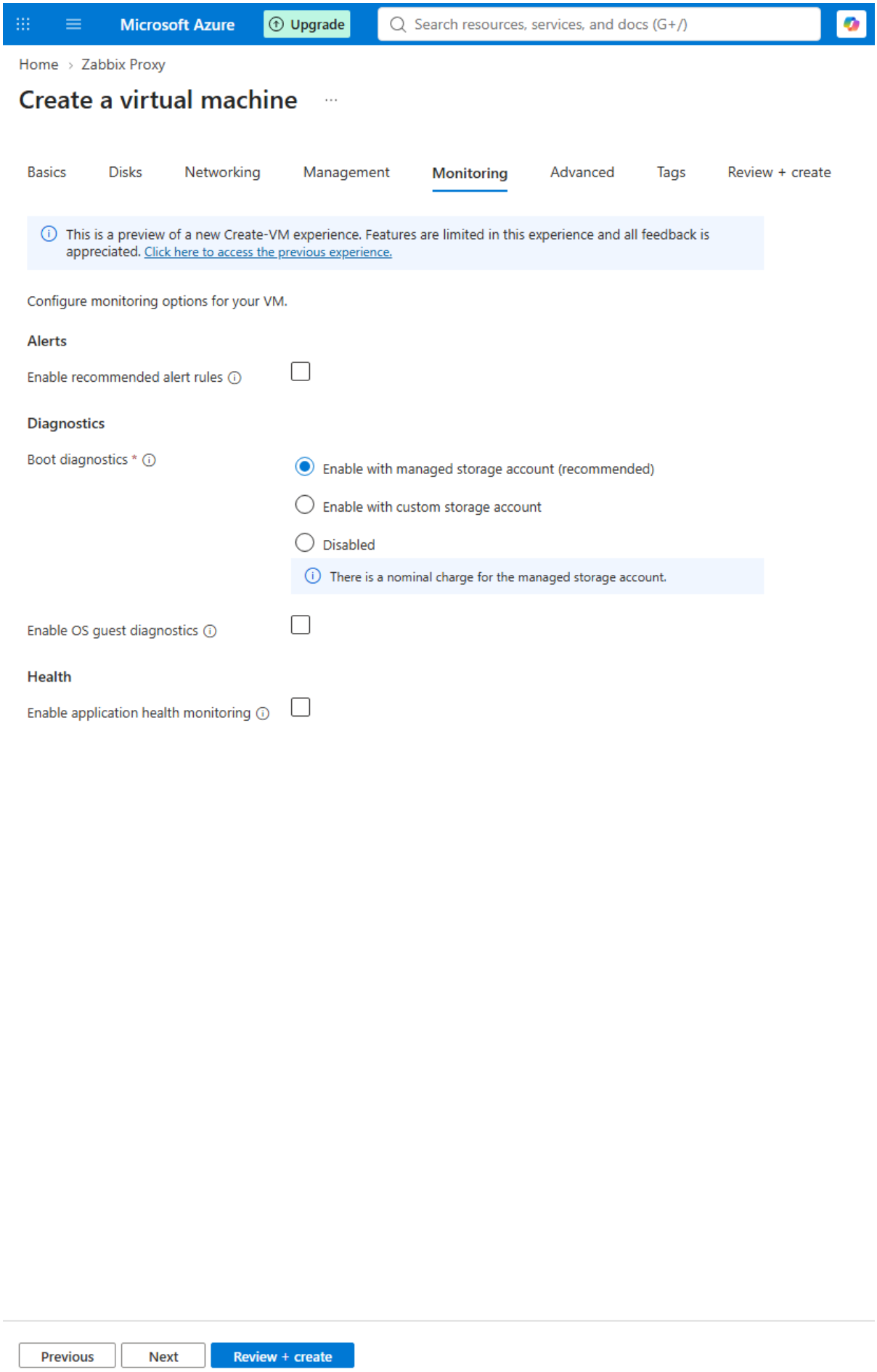
- *Virtual network*: Zabbix_Resources-vnet
- *Public IP*: Zabbix-proxy-ip
- *NIC network security group*: Advanced
- *Configure network security group*: Zabbix-proxy-nsg



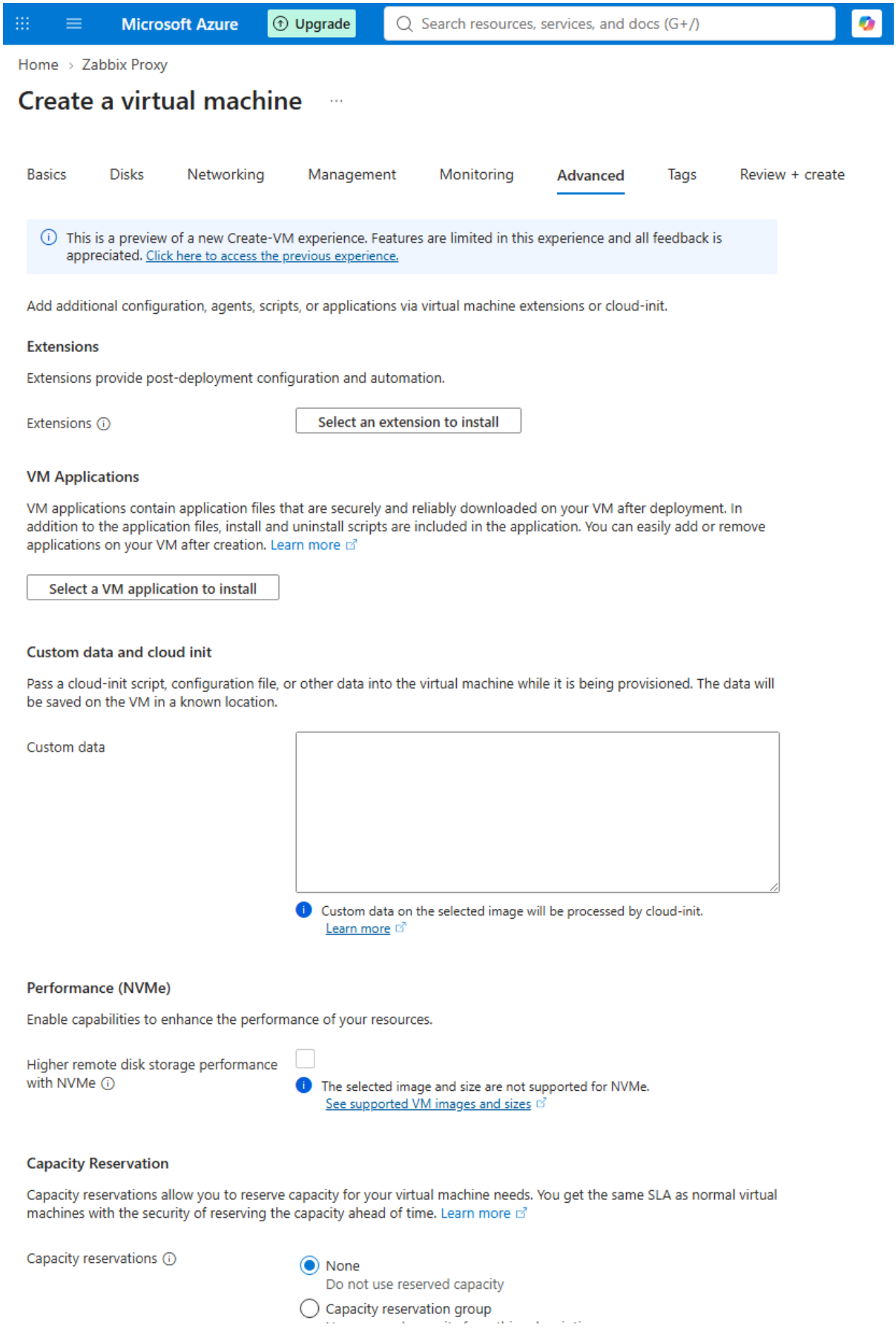
Cilnē *Management* parametrus var atstāt pēc noklusējuma.



Cilnē *Monitoring* parametrus var atstāt pēc noklusējuma.



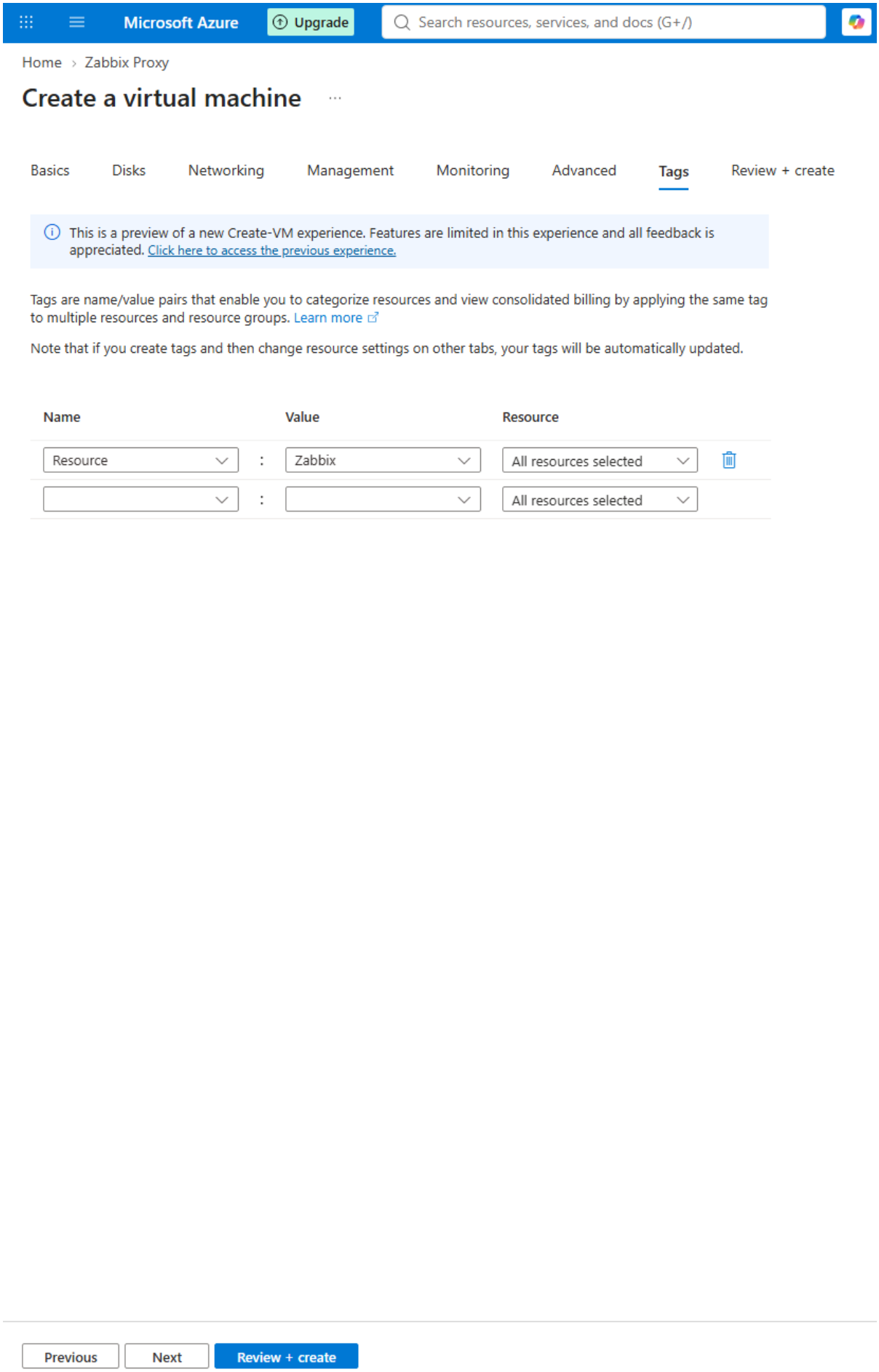
Cilnē *Advanced* parametrus var atstāt pēc noklusējuma.



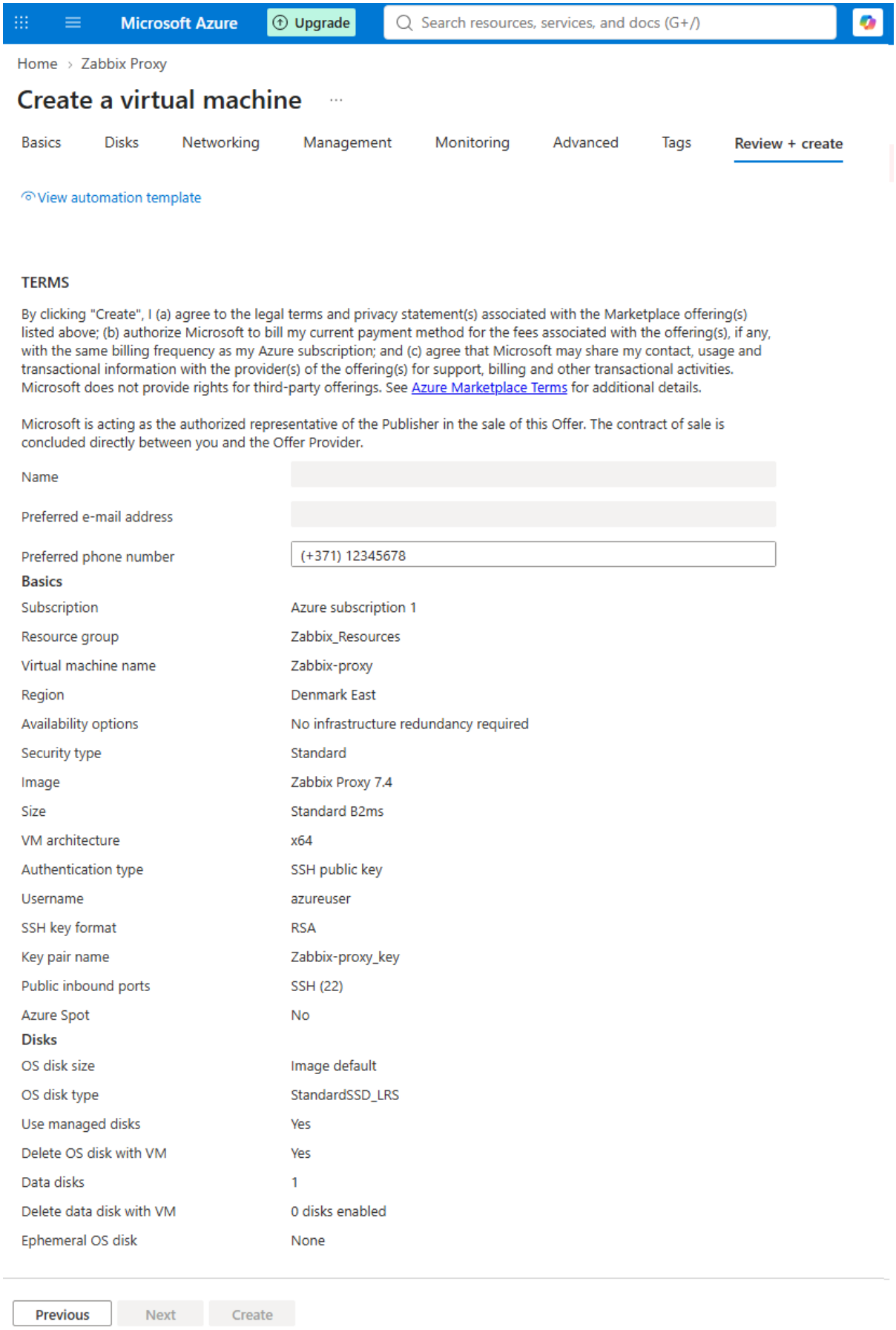
Cilne *Tags*:

- *Name*: Resource
- *Value*: Zabbix

- *Resource*: All resource selected



Apstipriniet virtuālās mašīnas informāciju un izveidojiet virtuālo mašīnu.



Jūs tiksit novirzīts uz virtuālās mašīnas izvietošanas statusa lapu. Uzgaidiet, līdz izvietošana ir pabeigta.

Kad izvietošana ir pabeigta, atveriet resursu un pierakstiet tā publisko IP adresi. Izveidojiet savienojumu ar Zabbix starpniekserveri, izmantojot SSH un administratora akreditācijas datus, kas izveidoti virtuālās mašīnas izvietošanas laikā, lai konfigurētu starpniekserveri.

```
********************************************************************************

To learn about available professional services, including technical suppport and
 training, please visit https://www.zabbix.com/services

Official Zabbix documentation available at https://www.zabbix.com/documentation/
current/

********************************************************************************
[azureuser@Zabbix-proxy ~]$
```

Zabbix starpniekservera konfigurācijas fails atrodas mapē `/etc/zabbix/zabbix_proxy.conf`.

Sākotnējā starpniekservera konfigurācija ietver parametru `ProxyMode`, `Server` un `Hostname` modificēšanu Zabbix starpniekservera konfigurācijas failā.

Starpniekservera režīmu var mainīt, rediģējot parametru `ProxyMode`. Aktīvajam režīmam iestatiet `ProxyMode=0`, bet pasīvajam režīmam — `ProxyMode=1`.

Parametram `Server` jāietver Zabbix servera IP adrese.

Parametram `Hostname` jāietver Zabbix starpniekservera resursdatora nosaukums. Šeit definētajam resursdatora nosaukumam jāatbilst starpniekservera nosaukumam, kas norādīts starpniekservera konfigurēšanas laikā Zabbix lietotāja saskarnē.

```
############ GENERAL PARAMETERS #################

### Option: ProxyMode
#       Proxy operating mode.
#       0 - proxy in the active mode
#       1 - proxy in the passive mode
#
# Mandatory: no
# Default:
# ProxyMode=0

### Option: Server
#       If ProxyMode is set to active mode:
#               IP address or DNS name (address:port) or cluster (address:port;address2:port) of Zabbix server to get
configuration data from and send data to.
#               If port is not specified, default port is used.
#               Cluster nodes need to be separated by semicolon.
#       If ProxyMode is set to passive mode:
#               List of comma delimited IP addresses, optionally in CIDR notation, or DNS names of Zabbix server.
#               Incoming connections will be accepted only from the addresses listed here.
#               If IPv6 support is enabled then '127.0.0.1', '::127.0.0.1', '::ffff:127.0.0.1' are treated equally
#               and '::/0' will allow any IPv4 or IPv6 address.
#               '0.0.0.0/0' can be used to allow any IPv4 address.
#               Example: Server=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com
#
# Mandatory: yes
# Default:
# Server=

Server=127.0.0.1
```

```
### Option: Hostname
#       Unique, case sensitive Proxy name. Make sure the Proxy name is known to the server!
#       Value is acquired from HostnameItem if undefined.
#
# Mandatory: no
# Default:
# Hostname=

Hostname=Zabbix proxy
```

Atveriet Zabbix lietotāja saskarni un pārejiet uz *Administration* > *Proxies*. Noklikšķiniet uz pogas *Create proxy*.

Ievadiet tādu pašu starpniekservera nosaukumu kā parametra *Hostname* vērtība starpniekservera konfigurācijas failā.

Atlasiet starpniekservera režīmu un norādiet starpniekservera IP adresi (aktīviem starpniekserveriem tā nav obligāta).

Noklikšķiniet uz pogas *Create proxy*, lai izveidotu starpniekserveri.

Starpniekservera datubāzes atrašanās vieta

SQLite starpniekservera datubāzi var atrast `/tmp/zabbix_proxy.sqlite3`, un to norāda parametrs `DBName` Zabbix starpniekservera konfigurācijas failā.

Google Cloud izvietošanas rokasgrāmata

Pārskats

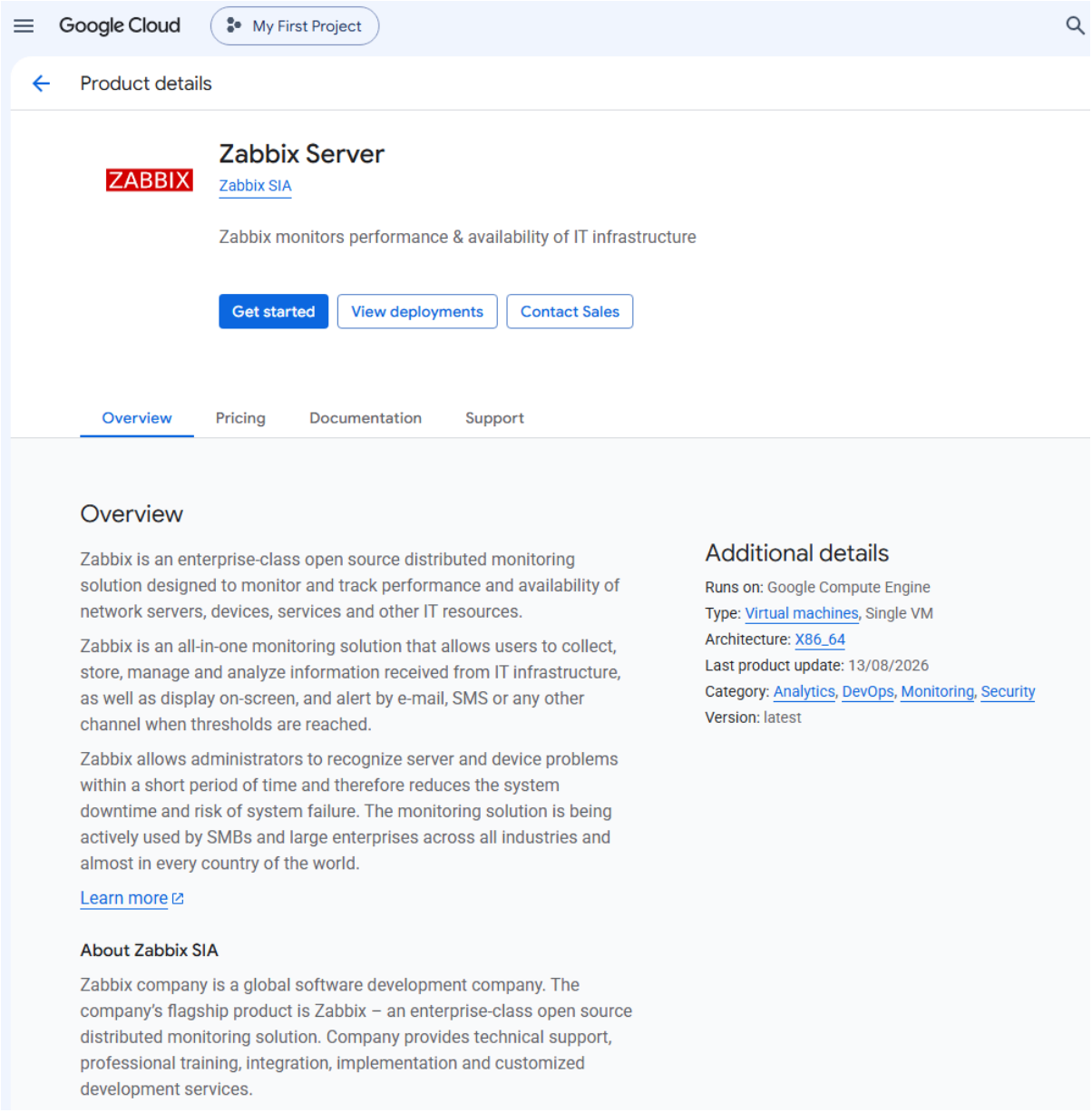
Šajā lapā aprakstīts, kā izvietot Zabbix, izmantojot Google Cloud Marketplace.

Attention:
Platformā izvietošanai ir pieejamas vairākas Zabbix versijas. Noteikti izlasiet tās pašas Zabbix versijas dokumentāciju, kuru esat izvietojis. Dokumentāciju dažādām Zabbix versijām var atlasīt versiju nolaižamajā izvēlnē.

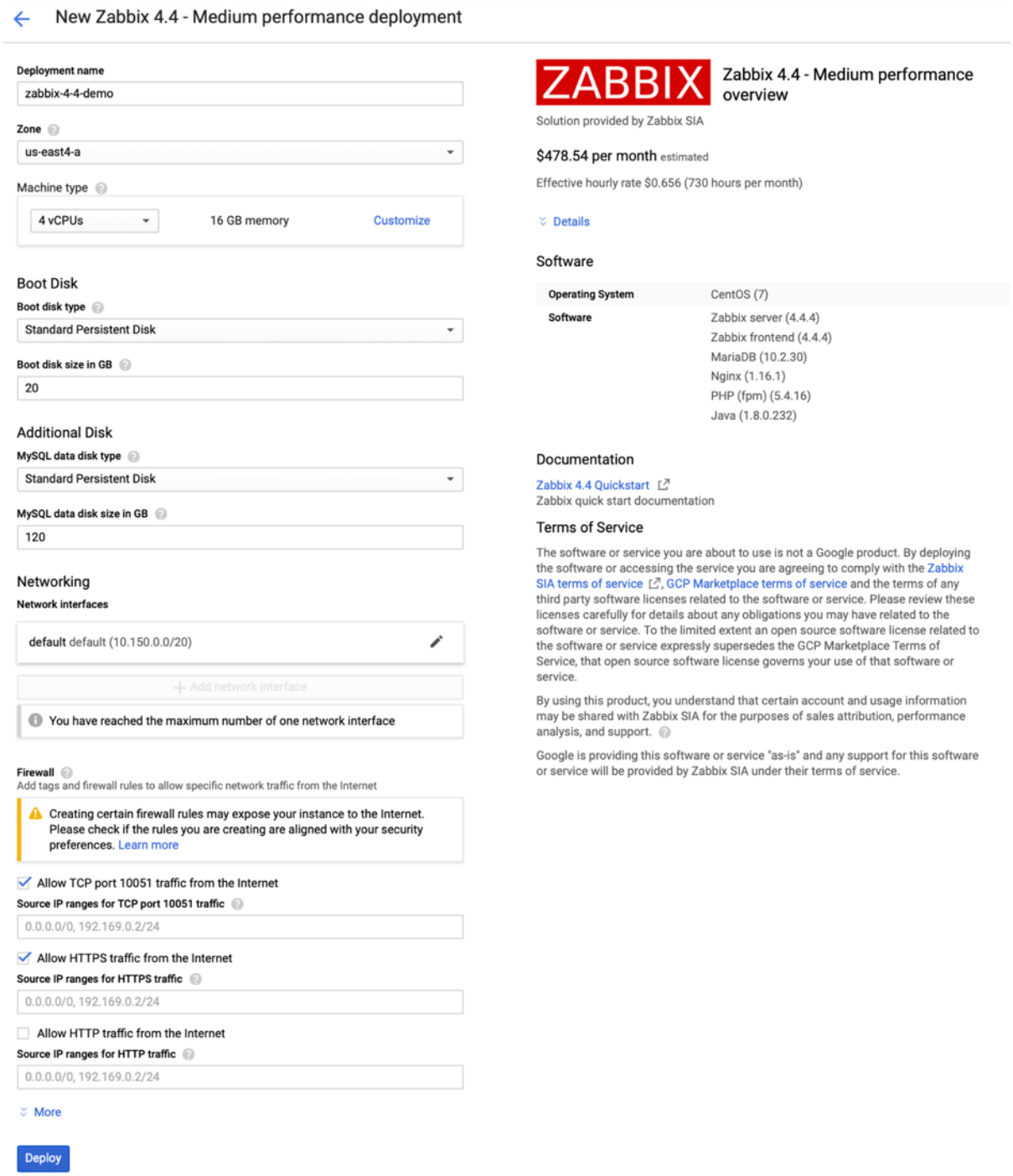
Ierīces izvietošana

Vispirms izvietojiet Zabbix ierīci. To ir ļoti vienkārši izdarīt, izmantojot Google Cloud Marketplace. Vienkārši atveriet produktu Zabbix Server marketplace.

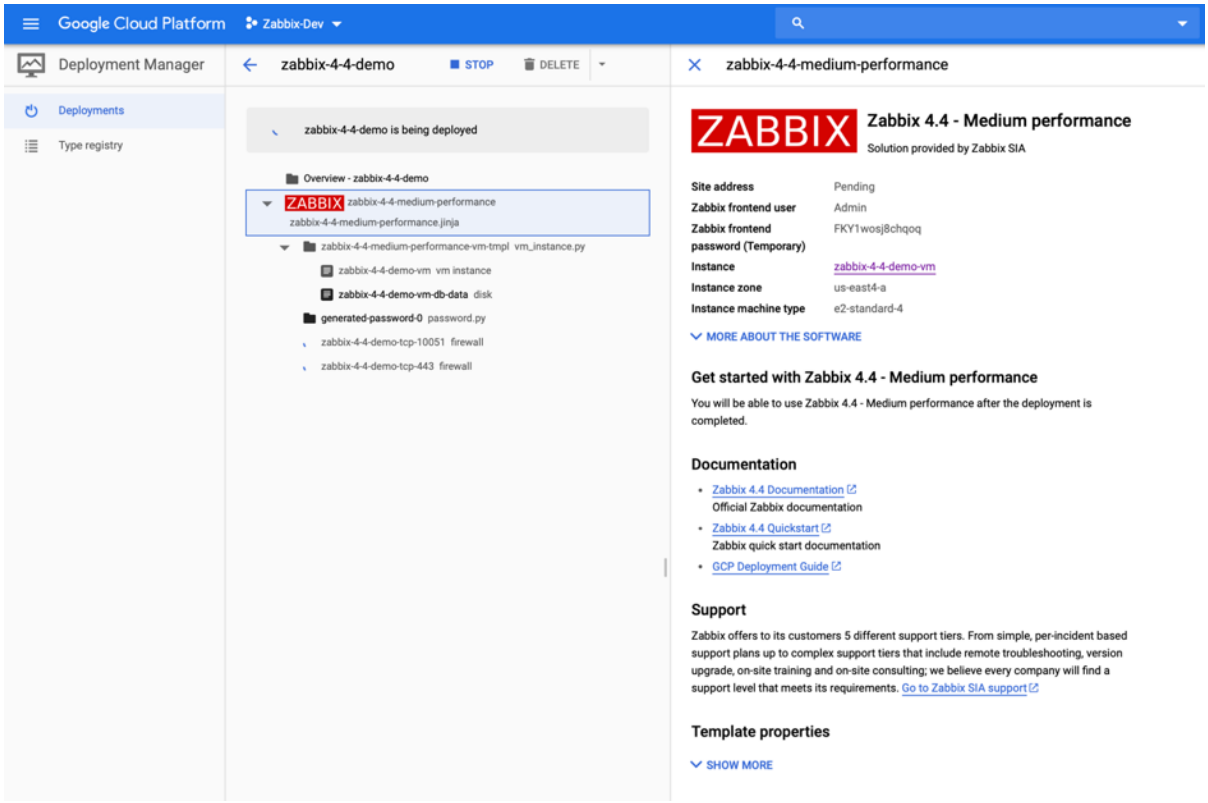
Noklikšķiniet uz *Get started*.



Varat izmantot noklusējuma iestatījumus vai tos pielāgot. Kad esat pabeidzis, noklikšķiniet uz *Deploy*.



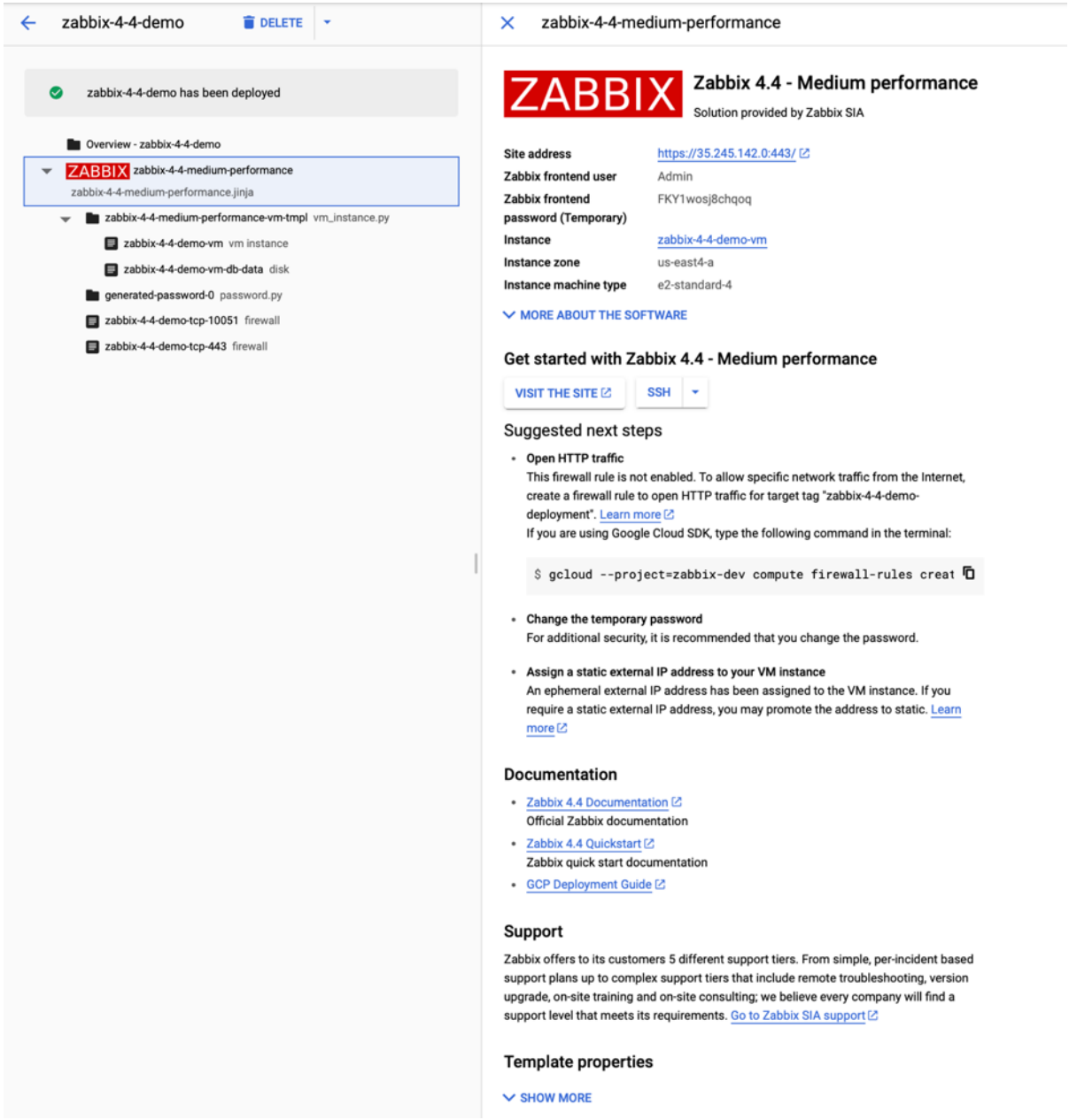
Izvietošana tagad notiek. Uzgaidiet, līdz tā tiek pabeigta.



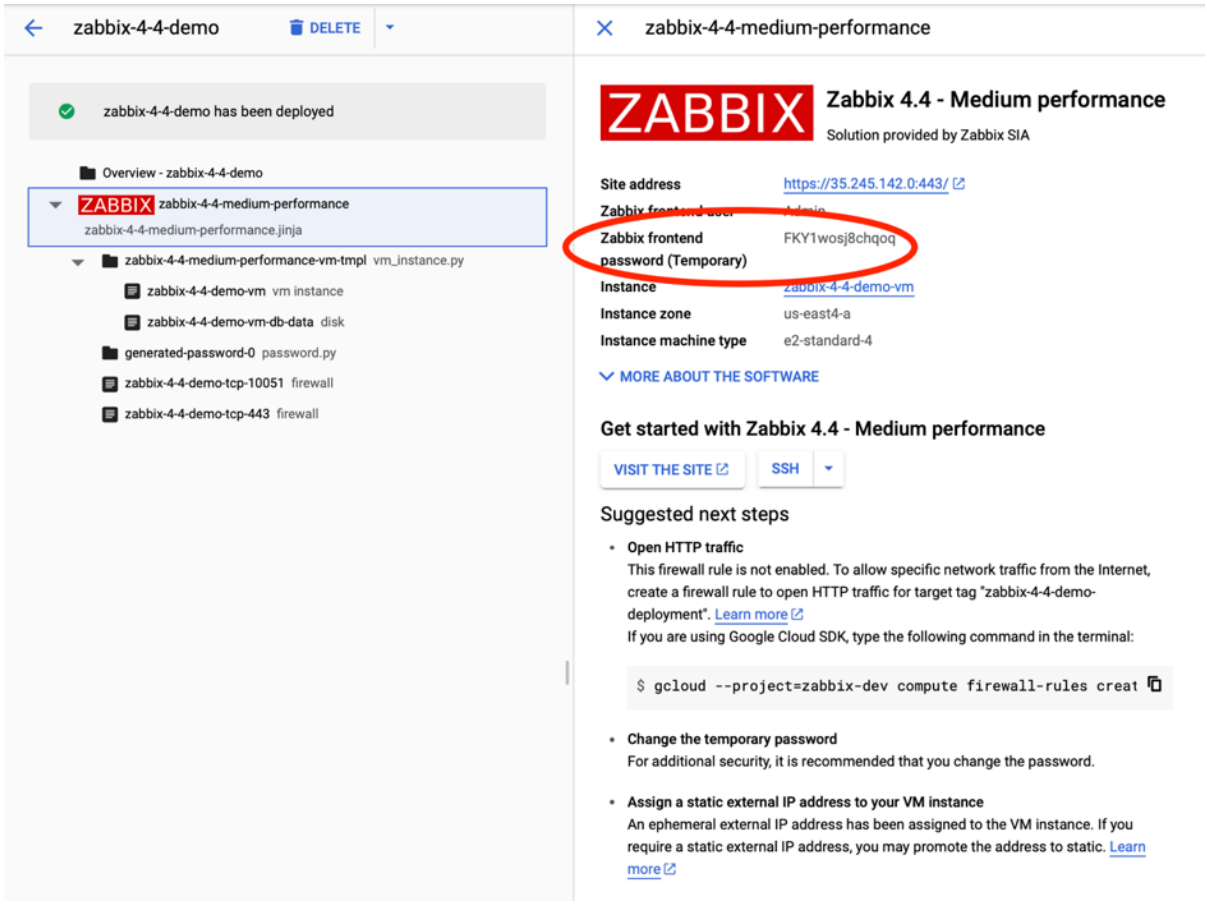
Tas arī viss! Izvietošana tagad ir sākta.

Piekļuve appliance

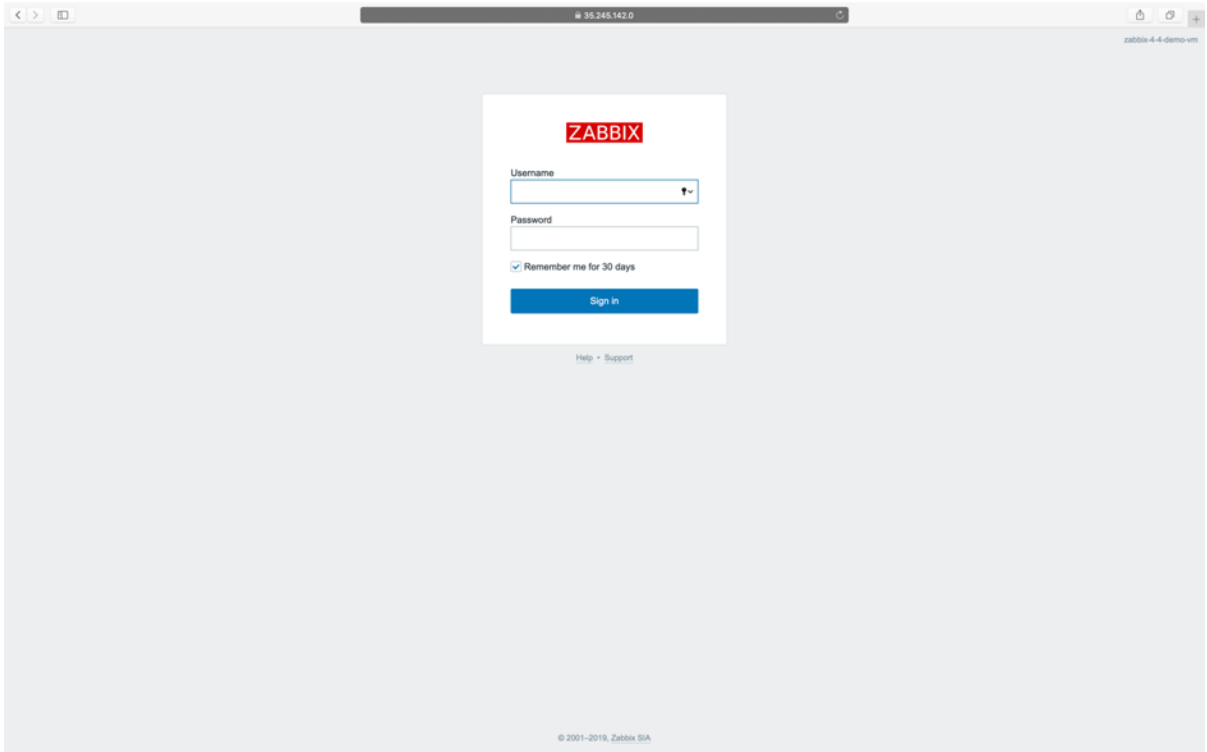
Kad izvietošana ir pabeigta, vajadzētu tikt parādītam ziņojumam, kas norāda, ka appliance tiek izvietots.

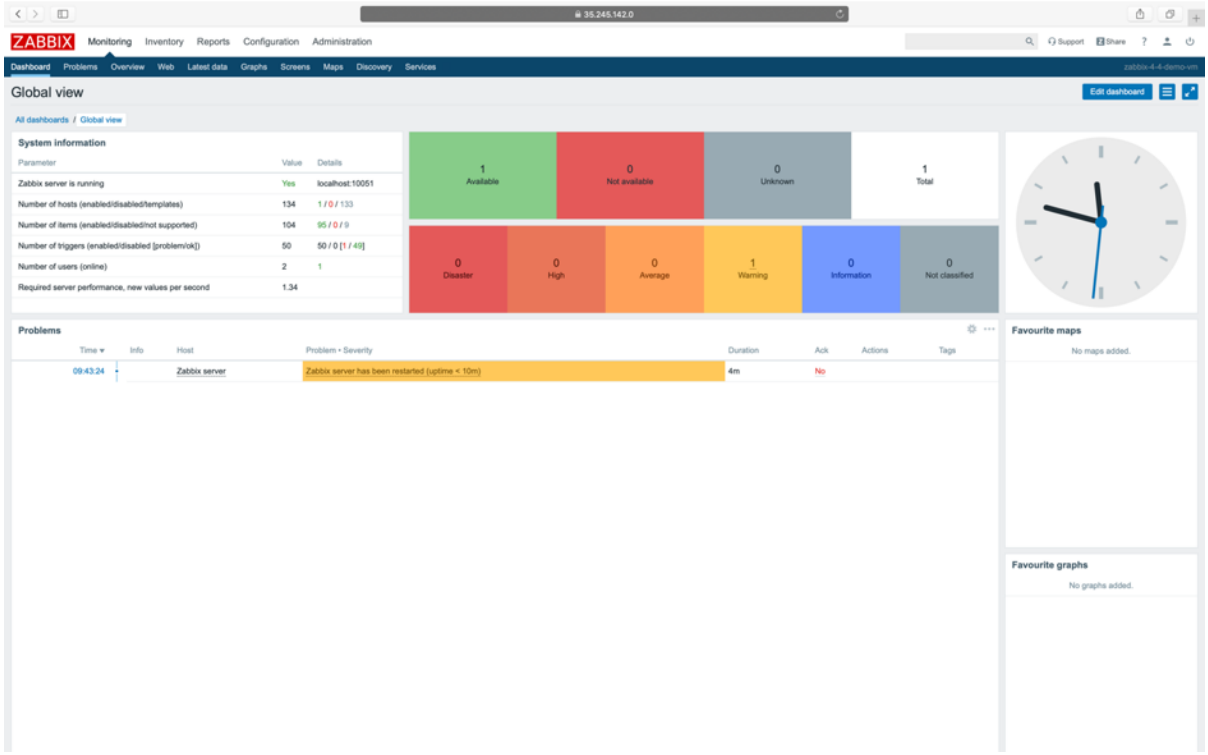


Atveriet tīmekļa pārlūkprogrammu un dodieties uz *Site Address*, lai skatītu Zabbix lietotāja saskarni. Pirms tam iegūstiet Zabbix *Admin* lietotāja paroli, lai pieteiktos savā Zabbix appliance.

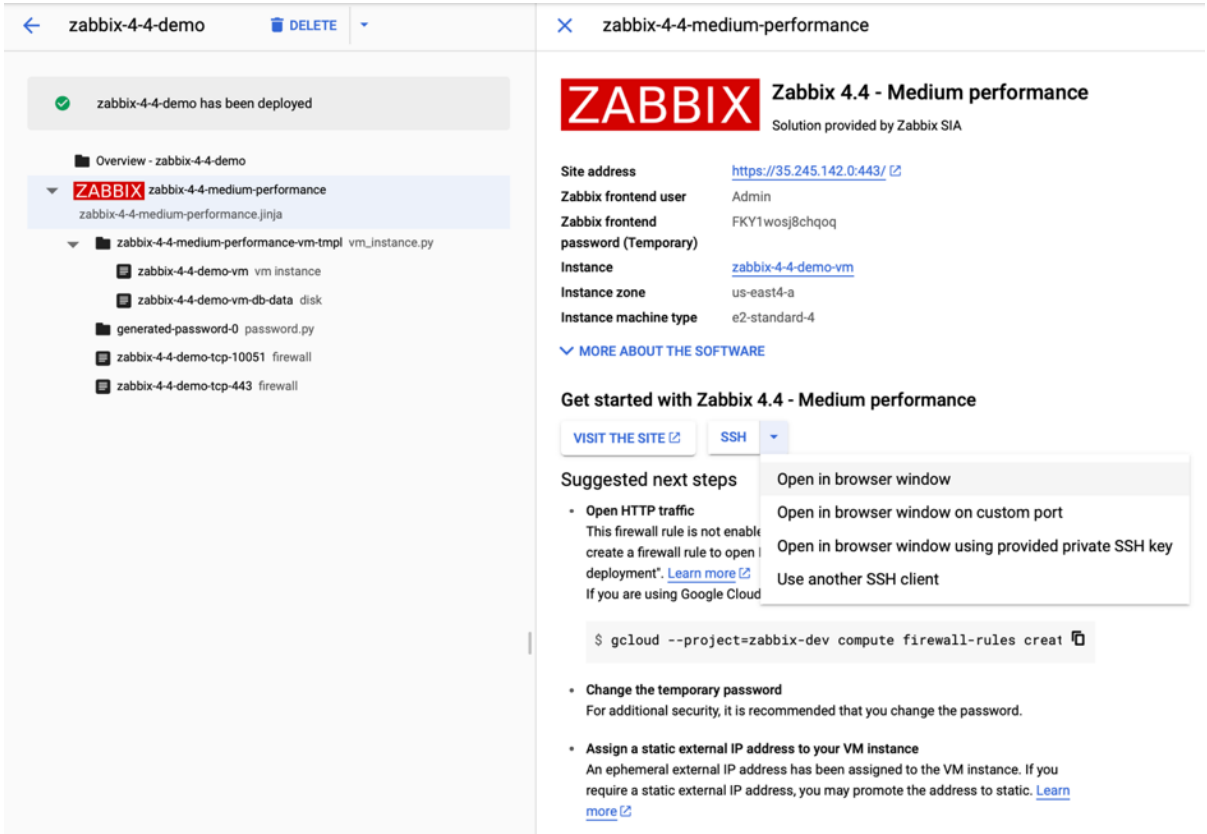


Lieliski! Tagad jums ir Zabbix appliance.





Varat arī pieteikties instancē, lai mijiedarbotos ar datubāzi. Lai to izdarītu, atgriezieties Google konsolē un, kā parādīts tālāk, sekojiet sarkanajai bultiņai, lai sāktu SSH sesiju, izmantojot opciju *Open in browser window*.



MySQL ir pieejams bez `root` paroles. Parole tiek glabāta konfigurācijas failā `/root/.my.cnf`. Pēc noklusējuma Zabbix appliance izmanto UTC laika joslu. Ja tā jāmaina, rediģējiet PHP konfigurācijas direktīvu `date.timezone` failā `/etc/php.d/99-zabbix.ini` un restartējiet php-fpm procesu: `systemctl restart php-fpm`.

Atjaunināšana

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

Noderīgi būs arī zināt, kuri komponenti jums ir jāatjaunina.

Pārskats

Detalizēti jaunināšanas norādījumi ir pieejami:

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

Saistītie norādījumi:

- Par serveriem augstas pieejamības (HA) klasterī skatiet HA klastera jaunināšana
- Par TimescaleDB datubāzi skatiet TimescaleDB shēmas jaunināšana

Jaunināmie komponenti

Zabbix aģentu jaunināšana ir ieteicama, bet nav obligāta.

Zabbix starpniekserveru jaunināšana ir ļoti ieteicama.
Zabbix serveris pilnībā atbalsta starpniekserverus, kuriem ir tāda 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 apkopošanu, attālo komandu izpildi, tūlītēju vienuma vērtību pārbaudi).
Konfigurācijas atjaunināšana arī ir atspējota, un novecojuši starpniekserveri darbosies tikai ar veco konfigurāciju.

Attention:
Starpniekserveri, kas ir vecāki par Zabbix servera iepriekšējā 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, parādot brīdinājumu. Papildinformāciju skatiet sadaļā Versiju saderība.

Lai jaunināšanas laikā samazinātu dīkstāvi un datu zudumu, ieteicams apturēt, jaunināt un palaist Zabbix serveri, pēc tam vienu pēc otra apturēt, jaunināt un palaist Zabbix starpniekserverus. Servera dīkstāves laikā darbojošies starpniekserveri turpinās vākt datus. Kad serveris būs palaists un darbosies, novecojušie starpniekserveri nosūtīs datus jaunākajam serverim (tomēr starpniekservera konfigurācija netiks atjaunināta) un paliks daļēji funkcionāli. Visi paziņojumi par problēmām Zabbix servera dīkstāves laikā tiks ģenerēti tikai pēc jaunināta servera palaišanas.

Ja Zabbix starpniekserveris tiek palaists pirmo reizi un SQLite datubāzes fails nav pieejams, 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 un izies.

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

Jaunināšanas piezīmes

Tieša jaunināšana uz Zabbix 8.0.x tiek atbalstīta no versijām **2.0**.x. Lai veiktu jaunināšanu 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 ar Zabbix, ja ārējā programmatūra nav saderīga ar jaunināto Zabbix versiju.

Pieejamas šādas jaunināšanas piezīmes:

Jaunināšana no	Lasīt pilnas jaunināšanas piezīmes	Svarīgākās izmaiņas starp versijām
7.4.x	Par: Zabbix 8.0	Zabbix aģenti, kas ir vecāki par 1.6. versiju, vairs netiek atbalstīti. Paaugstinātas minimālās nepieciešamās datubāzu versijas. Minimālā nepieciešamā PHP versija paaugstināta no 8.0.0 uz 8.2.0. Rakstzīme % pievienota `UnsafeUserParameters` sarakstam Zabbix aģentam un Zabbix aģentam 2.
7.2.x	Par: Zabbix 7.4 Zabbix 8.0	Pārtraukts PCRE bibliotēkas atbalsts.

Jaunināšana no	Lasīt pilnas jauninājumu piezīmes	Svarīgākās izmaiņas starp versijām
7.0.x	Par: Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	Pārtraukts Oracle DB atbalsts.
6.4.x	Par: Zabbix 7.0 Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	Minimālā 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 Windows aģenta konfigurācijas faila noklusējuma atrašanās vieta. Oracle DB novecojis. Atcelts vecais skaitliskais (peldošā komata) vērtības tips.
6.2.x	Par: Zabbix 6.4 Zabbix 7.0 Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	Minimālā nepieciešamā MySQL versija paaugstināta no 8.0.0 uz 8.0.30. Zabbix serverim/starpniekserverim nepieciešama bibliotēka 'libevent_pthreads'. Pēc pirmās palaišanas pēc jauninājumu Zabbix starpniekserveris ar SQLite3 automātiski dzēš veco datubāzes versiju (kopā ar visu vēsturi) un izveido jaunu.
6.0.x LTS	Par: Zabbix 6.2 Zabbix 6.4 Zabbix 7.0 Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	Minimālā nepieciešamā PHP versija paaugstināta no 7.2.5 uz 7.4.0. Pakalpojumu uzraudzība būtiski pārstrādāta. Jauninājumu laikā jāizveido deterministiskie trigeri. Ja MySQL/MariaDB ir iespējota binārā reģistrēšana, tam nepieciešamas superlietotāja privilēģijas vai mainīgā/konfigurācijas parametra *log_bin_trust_function_creators = 1* iestatīšana. Norādījumus par mainīgā iestatīšanu skatiet sadaļā Datubāzes izveides skripti.
5.4.x	Par: Zabbix 6.0 Zabbix 6.2 Zabbix 6.4 Zabbix 7.0 Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	Paaugstinātas minimālās nepieciešamās datubāzu versijas. Serveris/starpniekserveris netiks palaists, ja datubāze būs novecojusi. Datubāzes struktūras izmaiņu dēļ zaudēti audita žurnāla ieraksti.
5.2.x	Par: 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 minimālās nepieciešamās datubāzu versijas. Apkopotie vienumi noņemti kā atsevišķs tips.
5.0.x LTS	Par: 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	Minimālā nepieciešamā PHP versija paaugstināta no 7.2.0 uz 7.2.5. Paroļu jaukšanas algoritms mainīts no MD5 uz bcrypt.

Jaunināšana no	Lasīt pilnas jauninājanas piezīmes	Svarīgākās izmaiņas starp versijām
4.4.x	Par:	Pārtraukts IBM DB2 atbalsts.
	Zabbix 5.0	Minimālā nepieciešamā PHP versija paaugstināta no 5.4.0 uz 7.2.0.
	Zabbix 5.2	Paaugstinātas minimālās nepieciešamās datubāzu versijas.
	Zabbix 5.4	Mainīta Zabbix PHP failu direktorija.
	Zabbix 6.0	
	Zabbix 6.2	
	Zabbix 6.4	
	Zabbix 7.0	
	Zabbix 7.2	
	Zabbix 7.4	
	Zabbix 8.0	
4.2.x	Par:	Noņemti Jabber un Ez Texting multivides tipi.
	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	Par:	Vecāki starpniekserveri vairs nevar nosūtīt datus jaunīnātam serverim.
	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	Par:	Bibliotēkas ’libpthread’ un ’zlib’ tagad ir obligātas.
	Zabbix 4.0	Pārtraukts vienkārša teksta protokola atbalsts, un galvene ir obligāta.
	Zabbix 4.2	Zabbix aģenti, kas ir vecāki par 1.4. versiju, vairs netiek atbalstīti.
	Zabbix 4.4	Parametrs Server pasīvā starpniekservera konfigurācijā tagad ir
	Zabbix 5.0	obligāts.
	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	

Jaunināšana no	Lasīt pilnas jauninājumu piezīmes	Svarīgākās izmaiņas starp versijām
3.2.x	Par: 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 kā aizmugursistēmas datubāzes atbalsts Zabbix serverim/lietotāja saskarnei pārtraukts. POSIX paplašināto regulāro izteiksmju vietā tiek atbalstītas ar Perl saderīgas regulārās izteiksmes (PCRE). Zabbix serverim obligātas bibliotēkas 'libpcre' un 'libevent'. Pievienotas izejas koda pārbaudes lietotāja parametriem, attālajām komandām un system.run[] vienumiem bez karoga 'nowait', kā arī Zabbix servera izpildītajiem skriptiem. Lai atbalstītu jauno funkcionalitāti, Zabbix Java gateway ir jājaunina.
3.0.x LTS	Par: 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 jaunināšana var būt lēna atkarībā no vēstures tabulas lieluma.
2.4.x	Par: 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	Minimālā nepieciešamā PHP versija paaugstināta no 5.3.0 uz 5.4.0. Jānorāda aģenta parametrs LogFile.

Jaunināšana no	Lasīt pilnas jauninājumu piezīmes	Svarīgākās izmaiņas starp versijām
2.2.x LTS	Par: 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	Noņemta uz mezgliem balstīta izkliedētā uzraudzība.
2.0.x	Par: 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	Minimālā nepieciešamā PHP versija paaugstināta no 5.1.6 uz 5.3.0. Lai serveris darbotos pareizi, nepieciešama reģistrjutīga MySQL datubāze; lai Zabbix serveris pareizi darbotos ar MySQL datubāzi, nepieciešama rakstzīmju kopa utf8 un salīdzināšana utf8_bin. Skatiet datubāzes izveides skriptus. PHP paplašinājums ’mysqli’ nepieciešams ’mysql’ vietā.

Upgrade from packages

Pārskats

Šajā sadaļā ir sniegti soļi, kas nepieciešami veiksmīgai jaunināšanai, 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 jaunināšanu 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 jaunināšanu, 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 Dublējiet Zabbix konfigurācijas failus, PHP failus un Zabbix bināros failus

Dublējiet esošos Zabbix konfigurācijas failus, PHP failus un Zabbix bināros failus.

Konfigurācijas failiem izpildiet:

```
mkdir /opt/zabbix-backup/
cp /etc/zabbix/zabbix_server.conf /opt/zabbix-backup/
##### Apache gadījumā:
cp /etc/apache2/conf-enabled/zabbix.conf /opt/zabbix-backup/
##### Nginx gadījumā:
cp /etc/nginx/conf.d/zabbix.conf /opt/zabbix-backup/
cp /etc/zabbix/nginx.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
```

Iespējams, jums būs arī manuāli jāizdzēš jebkuras vecās Zabbix pakotnes no darba direktorijas (piem., `rm zabbix-release_latest+debia`
pirms jaunās lejupielādes, lai pakotņu pārvaldnieks jauninášanas procesa laikā neizmantotu novecojušu versiju atkārtoti.

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

Uz **Debian 13** 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+debian13_all.deb
```

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 saiti no Zabbix repozitorija. Tomēr ņemiet vērā, ka šo versiju pakotnes var neietvert visus Zabbix komponentus, un, lai jauninātu šos komponentus no pakotnēm, apsveriet OS jaunināšanu. Komponentu sarakstu skatiet sadaļā Zabbix pakotnes.

Uz **Ubuntu 26.04** 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+ubuntu26.04_all.deb
```

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 saiti no Zabbix repozitorija. Tomēr ņemiet vērā, ka šo versiju pakotnes var neietvert visus Zabbix komponentus, un, lai jauninātu šos komponentus no pakotnēm, apsveriet OS jaunināšanu. Komponentu sarakstu skatiet sadaļā Zabbix pakotnes.

Var tikt parādīts uzvednis 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 aģentu 2, komandā aizstājiet `zabbix-agent` ar `zabbix-agent2 zabbix-agent2-plugin-*`.

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

Var tikt parādīta uzvedne par Zabbix servera (vai starpniekservera) 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 situācijai atbilstošāko opciju. Piemēram, ievadiet `D`, lai salīdzinātu pašreizējo un jauno konfigurāciju, pēc tam izlemiet, vai vēlaties instalēt pakotnes uzturētāja versiju (`Y` vai `I`).

Pēc tam jauniniet Zabbix lietotāja saskarni un restartējiet tīmekļa serveri.

Apache gadījumā izpildiet:

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

Nginx gadījumā izpildiet:

```
apt install zabbix-nginx-conf
systemctl restart nginx
```

Nginx gadījumā var tikt parādīta uzvedne par `nginx.conf` konfigurācijas failu. Ievadiet situācijai atbilstošāko opciju:

```
Configuration file '/etc/zabbix/nginx.conf'
 ==> Modified (by you or by a script) since installation.
 ==> Package distributor has shipped an updated version.
   What would you like to do about it ?  Your options are:
    Y or I  : install the package maintainer's version
    N or O  : keep your currently-installed version
      D     : show the differences between the versions
      Z     : start a shell to examine the situation
 The default action is to keep your current version.
*** nginx.conf (Y/I/N/O/D/Z) [default=N] ?
```

6 Pārskatiet komponenta konfigurācijas parametrus

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

Par jauniem izvēles parametriem skatiet lapu Kas jauns.

Ja kā tīmekļa serveri izmantojat Nginx un ja jauninašanas laikā tika atkārtoti instalēts konfigurācijas fails `/etc/zabbix/nginx.conf`, pārskatiet šo failu un atjauniniet direktīvas `listen` un `server_name`, lai tās atbilstu jūsu Zabbix lietotāja saskarnei.

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 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).

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 jaunināšanas 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 jaunināšanu, 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 saiti 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.

Pēc tam notīriet `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 atjaunināšanu uz RHEL.

5 Zabbix komponentu jaunināšana

Lai jauninātu Zabbix komponentus, izpildiet:

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

- Ja izmantojat PostgreSQL, komandā aizstājiet `mysql` ar `pgsql`.
- Ja jaunināt starpniekserveri, komandā aizstājiet `server` ar `proxy`.
- Ja jaunināt aģentu 2, komandā aizstājiet `zabbix-agent` ar `zabbix-agent2 zabbix-agent2-plugin-*`.
- Ja ir instalētas arī citas Zabbix pakotnes, apsveriet to jaunināšanu, iekļaujot tās komandā.

Attention:
Zabbix aģenta 2 jaunināšana ar komandu `dnf install zabbix-agent2` var izraisīt kļūdu. Papildinformāciju skatiet sadaļā *Known issues*.

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

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

6 Pārskatiet komponenta konfigurācijas parametrus

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

Par jauniem izvēles 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 jaunināšanas jums var būt nepieciešams notīrīt tīmekļa pārlūka sīkfailus un tīmekļa pārlūka kešatmiņu, lai Zabbix tīmekļa 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).

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.

Jaunināšanas problēmas

Ar atsoļošanu saistīts Pārsniegts funkciju parametru rakstzīmju ierobežojums pēc jaunināšanas

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

Tā kā jaunināšanas 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 jaunināšanas 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:

```
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
```

3. `vfs.file.contents[/tmp/ONE]` vienums, lai nolasītu /tmp/ONE.
4. Divi aprēķinātie vienumi:

Calc_228 — tā parametrs satur 228 atpakaļvērstās slīpsvītras, kas ir ekrānētas, tāpēc tās **atbilst** 114 atpakaļvērstajām slīpsvītrām no faila /tmp/ONE:

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

Calc_232 — tā parametrs satur 232 atpakaļvērstās slīpsvītras, kas ir ekrānētas, tāpēc tās **neatbilst** 114 atpakaļvērstajām slīpsvītrām no faila /tmp/ONE, jo ir nepieciešamas vismaz 116 reālas atpakaļvērstās slīpsvītras:

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

5. Divi trigeri:

Trig_228 — līdzīgi kā Calc_228, tā parametrs satur 228 atpakaļvērstās slīpsvītras, lai **atbilstu** 114 reālām atpakaļvērstajām slīpsvītrām no faila /tmp/ONE:

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

Trig_232 — līdzīgi kā Calc_232, tā parametrs satur 232 atpakaļvērstās slīpsvītras, lai atbilstu 116 reālām atpakaļvērstajām slīpsvītrām, tāpēc tas **neatbilst** 114 atpakaļvērstajām slīpsvītrām no faila /tmp/ONE:

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

6. Calc_228 atgriež 1, un Trig_228 **nostrādā**.
7. Calc_232 atgriež 0, un Trig_232 **nenostrādā**.

Jaunināšana

1. Palaidiet jaunināšanu uz 8.0.0 un pārbaudiet žurnālus, vai tajos nav šāda brīdinājuma:

```
2485502:20250228:115442.236 DBpatch_6050165(): cannot save in DB function parameter: resulting size 477 is
functionid:33792 function:'find'
used on host: 'Zabbix server'
  in trigger: 'TRIG_228'.
Current parameter value:
'$,1m:now,"regexp","\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\a"'
Resulting escaped value would be:
'$,1m:now,"regexp","\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
DB upgrade and Zabbix server can continue to run, but to make sure this function works correctly
MANUAL INTERVENTION IS REQUIRED !
Need to manually reduce the size of this parameter with macro as a workaround.

2485502:20250228:115442.237 DBpatch_6050165(): cannot save in DB function parameter: resulting size 485 is
functionid:33795 function:'find'
used on host: 'Zabbix server'
```

```
  in trigger: 'TRIG_232'.
Current parameter value:
'$,1m:now,"regexp","\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\a"'
Resulting escaped value would be:
'$,1m:now,"regexp","\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
DB upgrade and Zabbix server can continue to run, but to make sure this function works correctly
MANUAL INTERVENTION IS REQUIRED !
Need to manually reduce the size of this parameter with macro as a workaround.
```

2. Datubāzes jaunināšana **tiek pabeigta**, un Zabbix serveris turpina darboties.
3. Calc_228 joprojām atgriež 1, savukārt Calc_232 atgriež 0.
4. Trigeri tika jaunināti šādi:

Trig_228:

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

Trig_232:

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

Abi trigeri — Trig_228 un Trig_232 — **tagad nostrādā**! Tas ir negaidīti, tāpēc ir nepieciešama manuāla iejaukšanās.

Manuāla iejaukšanās

Trigeru izteiksmes ir jāatjaunina tā, lai to atpakaļvērstās slīpsvītras būtu atsoļotas, taču to nevar izdarīt, jo rezultējošie parametri būtu pārāk gari.

Tāpēc jums jāpievieno makrosi:

{$228_BACKSLASHES_A}:

```
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
```

{$232_BACKSLASHES_A}:

```
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
```

un **atjauniniet** trigeru izteiksmes, lai tās izmantotu:

Trig_228:

```
find(/Zabbix server/vfs.file.contents[/tmp/ONE],1m:now,"regexp",{$228_BACKSLASHES_A})
```

Trig_232:

```
find(/Zabbix server/vfs.file.contents[/tmp/ONE],1m:now,"regexp",{$232_BACKSLASHES_A})
```

Tagad abi trigeri darbojas tāpat kā iepriekš — tikai Trig_228 **nostrādā**.

Ar aprēķinātajiem vienumiem problēmu nav, taču konsekvences labad arī tos var atjaunināt:

Calc_228:

```
find(/Zabbix server/vfs.file.contents[/tmp/ONE],1m:now,"regexp",{$228_BACKSLASHES_A})
```

Calc_232:

```
find(/Zabbix server/vfs.file.contents[/tmp/ONE],1m:now,"regexp",{$232_BACKSLASHES_A})
```

Upgrade from containers

Pārskats

Šajā sadaļā ir aprakstītas darbības, kas nepieciešamas veiksmīgai jaunināšanai uz Zabbix **8.0**.x konteineriem.

Ir pieejami atsevišķi norādījumu komplekti šādiem gadījumiem:

- Zabbix komponentu attēlu jaunināšana

- Docker Compose faila jaunināšana

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

Attention:
Pirms jaunināšanas sākšanas pārliecinieties, ka lietotājiem ir nepieciešamās atļaujas datubāzei, lai veiktu jaunināšanu.

 Jauninot no Zabbix 6.0 vai vecākas versijas, jaunināšanas laikā būs jāizveido deterministiski trigeri. Ja MySQL/MariaDB ir iespējota binārā žurnalēšana, tam ir nepieciešamas superlietotāja privilēģijas vai arī jāiestata mainīgais/konfigurācijas parametrs `log_bin_trust_function_creators = 1`. Norādījumus par mainīgā iestatīšanu skatiet sadaļā Datubāzes izveides skripti.

 Ņemiet vērā, ka, izpildot no konsoles, mainīgais tiks iestatīts tikai īslaicīgi un tiks noņemts, 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.

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

Zabbix komponenta attēla jaunināšana

Šīs instrukcijas atjaunina jūsu Docker attēlus, kas tiek izmantoti kā bāze jūsu Zabbix komponentu konteineriem.

Tālāk norādītajās darbībās kā piemērs tiek izmantots Zabbix starpniekserveris. Aizstājiet `zabbix-proxy-sqlite3` ar sava komponenta konteinera nosaukumu.

1. Pārbaudiet visus konteinerus, lai noteiktu to nosaukumus un pašreizējās attēlu versijas:

```
docker ps -a

#### CONTAINER ID   IMAGE                                          (...) NAMES
#### 5a0c2e3d02b4   zabbix/zabbix-proxy-sqlite3:alpine-7.4-latest   (...) zabbix-proxy-sqlite3
```

2. Lejupielādējiet jauno attēla versiju:

```
docker pull zabbix/zabbix-proxy-sqlite3:alpine-8.0-latest
```

3. Apturiet Zabbix komponenta konteineru:

```
docker stop zabbix-proxy-sqlite3
```

4. Noņemiet Zabbix komponenta konteineru:

Attention:
Pirms turpināt, pārliecinieties, ka esat pierakstījis visus konfigurācijas karogus un sējumu montāžas, kas tika izmantoti sākotnējā `docker run` komandā, jo tie būs nepieciešami konteinera atkārtotai izveidei.

```
docker rm zabbix-proxy-sqlite3
```

5. Palaidiet jaunu konteineru, izmantojot atjaunināto attēlu:

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

Note:
Plašāku informāciju par Zabbix konteineru palaišanu skatiet sadaļā Docker (manuāla izvietošana).

6. Pārbaudiet, vai jaunināšana bija veiksmīga, apskatot konteinera žurnālus:

```
docker logs -f zabbix-proxy-sqlite3
```

Docker Compose faila jaunināšana

Šīs instrukcijas jaunina Docker Compose failus, kas tiek izmantoti Zabbix komponentu izvietošanai. Izpildiet šīs instrukcijas, ja Zabbix instalējāt, izmantojot Docker Compose.

1. Pārbaudiet visus konteinerus, lai noteiktu to nosaukumus un pašreizējās attēlu versijas:

```
docker ps -a

#### CONTAINER ID   IMAGE                                                  (...) NAMES
#### d095f486deef   zabbix/zabbix-server-mysql:alpine-7.4-latest          (...) zabbix-docker-zabbix-server-1
#### 70627d833da0   zabbix/zabbix-web-nginx-mysql:alpine-7.4-latest      (...) zabbix-docker-zabbix-web-ngina
#### aaaa664cc185   zabbix/zabbix-server-mysql:alpine-7.4-latest          (...) zabbix-docker-server-db-init-1
#### 9ab3e77492ff   mysql:8.0-oracle                                       (...) zabbix-docker-mysql-server-1
```

2. Pārejiet uz Zabbix Docker repozitoriju, ko iepriekš klonējāt, iegūstiet jaunākos atjauninājumus un pārslēdzieties uz 8.0 zaru:

Attention:
Ja savā lokālajā repozitorijā esat modificējis `.env`, `compose.yaml` vai citus failus, `git pull` var tos pārrakstīt. Pirms šīs darbības veikšanas izveidojiet lokālo izmaiņu rezerves kopiju.

```
cd zabbix-docker
git pull
git checkout 8.0
```

3. Izvietojiet Zabbix komponentus, izmantojot atjaunināto Docker Compose failu:

```
#### Ar MySQL kā datubāzi:
docker compose -f ./compose.yaml up -d

#### Ar PostgreSQL kā datubāzi:
docker compose -f ./compose_pgsql.yaml up -d
```

Note:
Pēc tam, kad visi konteineri ir atkārtoti izveidoti un palaisti, var paiet 1–3 minūtes, līdz datu apkopošana tiek atsākta. Papildinformāciju par Zabbix konteineru palaišanu skatiet sadaļā Docker Compose.

4. Pārbaudiet, vai jaunināšana bija veiksmīga, apskatot konteineru žurnālus (aizstājiet `zabbix-docker-zabbix-server-1` ar sava komponenta konteinera nosaukumu):

```
docker logs -f zabbix-docker-zabbix-server-1
```

Upgrade from sources

Pārskats

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

Warning:
Pirms jauninašanas noteikti izlasiet attiecīgās **jauninašanas piezīmes**!

Varat arī pārskatīt 8.0 prasības.

Note:
Jauninašanas laikā var būt noderīgi izmantot divas paralēlas SSH sesijas: vienā izpildīt jauninašanas darbības, bet otrā pārraudzīt servera/starpniekservera žurnālus. Piemēram, otrajā SSH sesijā palaidiet `tail -f zabbix_server.log` vai `tail -f zabbix_proxy.log`, lai reāllaikā skatītu jaunākos žurnālfaila ierakstus un iespējamās kļūdas. Tas var būt īpaši svarīgi produkcijas vidēm.

Servera jauninašanas process

1. Apturiet serveri

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

2 Izveidojiet esošās Zabbix datubāzes dublējumkopiju

Šis ir ļoti svarīgs solis.
Pārliecinieties, ka jums ir datubāzes dublējumkopija.
Tā palīdzēs, ja jaunināšanas procedūra neizdosies (nepietiek vietas diskā, strāvas padeves pārtraukums vai cita neparedzēta problēma).

3 Dublējiet konfigurācijas failus, PHP failus un Zabbix bināros failus

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

4 Instalējiet jaunos 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 ir nepieciešamas kādas izmaiņas.

Par jaunajiem izvēles parametriem skatiet lapu Kas jauns.

6 Palaidiet jaunos Zabbix bināros failus

Palaidiet jaunos bināros failus.
Pārbaudiet žurnālfailus, lai pārliecinātos, ka binārie faili ir veiksmīgi palaisti.

Zabbix serveris automātiski jauninās datubāzi.
Startējoties Zabbix serveris ziņo par pašreizējām (obligāto un neobligāto) un nepieciešamajām datubāzes versijām.
Ja pašreizējā obligātā versija ir vecāka par nepieciešamo versiju, Zabbix serveris automātiski izpilda nepieciešamos datubāzes jauninašanas labojumus.
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 labojumiem neizdodas, Zabbix serveris netiks palaists.
Zabbix serveris netiks palaists arī tad, ja pašreizējā obligātā datubāzes versija ir jaunāka par nepieciešamo versiju.
Zabbix serveris tiks palaists tikai tad, ja pašreizējā obligātā datubāzes versija atbildīs 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 (izveidot tabulu, dzēst tabulu, izveidot indeksu, dzēst indeksu).
- Pārliecinieties, ka ir pietiekami daudz brīvas vietas diskā.

7 Instalējiet jaunu Zabbix lietotāja saskarni

Minimālā nepieciešamā PHP versija ir 8.2.0.
Ja nepieciešams, atjauniniet to un izpildiet instalēšanas norādījumus.

8 Notīriet tīmekļa pārlūkprogrammas sīkfailus un kešatmiņu

Pēc jauninašanas, lai Zabbix tīmekļa lietotāja saskarne darbotos pareizi, iespējams, būs jānotīra tīmekļa pārlūkprogrammas sīkfaili un kešatmiņa.

Starpniekservera jauninašanas process

1 Apturiet starpniekserveri

Apturiet Zabbix starpniekserveri.

2 Dublējiet konfigurācijas failus un Zabbix starpniekservera bināros failus

Izveidojiet Zabbix starpniekservera binārā faila un konfigurācijas faila rezerves kopiju.

3. Instalējiet jaunos starpniekservera bināros failus

Izmantojiet šīs instrukcijas, lai kompilētu Zabbix starpniekserveri no pirmkoda.

4 Pārskatiet starpniekservera konfigurācijas parametrus

Pārliecinieties, ka pārskatāt jaunināšanas piezīmes, lai pārbaudītu, vai konfigurācijas parametros ir nepieciešamas kādas izmaiņas.

Par jauniem izvēles parametriem skatiet lapu Kas jauns.

5 Jauna Zabbix starpniekservera palaišana

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

Zabbix starpniekserveris automātiski atjauninās datubāzi.
Datubāzes atjaunināšana notiek līdzīgi kā, palaižot Zabbix serveri.

Aģenta jaunināšanas process

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

Šajā sadaļā aprakstīto jaunināšanas procedūru var izmantot gan Zabbix aģenta, gan Zabbix aģenta 2 jaunināšanai.

1 Apturiet aģentu

Apturiet Zabbix aģentu.

2 Dublējiet konfigurācijas failus un Zabbix aģenta bināros failus

Izveidojiet Zabbix aģenta binārā faila un konfigurācijas faila rezerves kopiju.

3. Jaunu aģenta bināro failu instalēšana

Izmantojiet šīs instrukcijas, lai kompilētu Zabbix aģentu no pirmkoda.

Varat arī lejupielādēt iepriekš kompilētus Zabbix aģentus no Zabbix lejupielādes lapas.

4 Pārskatiet aģenta konfigurācijas parametrus

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

Par jauniem izvēles parametriem skatiet lapu Kas jauns.

5 Jaunā Zabbix aģenta palaišana

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

Jaunināšana starp mazajām versijām

Jauninot starp 8.0.x mazajām versijām (piemēram, no 8.0.1 uz 8.0.3), serverim/starpniekserverim/aģentam ir jāveic tādas pašas darbības kā jaunināšanas laikā starp galvenajām versijām. Vienīgā atšķirība ir tā, ka, jauninot starp mazajām versijām, datubāzē netiek veiktas nekādas izmaiņas.

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).

MySQL datubāzes rakstzīmju kopa un kolācija

Zabbix instalācijās, kurās tiek izmantota MySQL datubāze, kas izveidota ar utf8mb3 rakstzīmju kopu un utf8mb3_bin kolāciju, var rasties problēmas. Piemēram, zināma MySQL problēma datubāzes shēmas jaunināšanas vai citu DDL darbību laikā var izraisīt kļūdas `Row size too large (> 8126).<br>` Ir stingri ieteicams datubāzes, kurās tiek izmantota utf8mb3 rakstzīmju kopa un utf8mb3_bin kolācija, pārveidot uz utf8mb4 rakstzīmju kopu un utf8mb4_bin kolāciju, kā aprakstīts sadaļā Zabbix datubāzes rakstzīmju kopas un kolācijas labošana.

Atmiņas un failu deskriptoru noplūde ar libcurl 8.21.0

Zināma problēma libcurl 8.21.0 var izraisīt atmiņas un failu deskriptoru noplūdes Zabbix funkcijās, kas izmanto libcurl, apstrādājot ar TLS aizsargātus savienojumus.
 Šīs problēmas risinājums ir pazemināt libcurl versiju uz 8.20.0, līdz kļūst pieejams laidiens ar labojumu.

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 jārediģē 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.

Datubāzes TLS savienojums ar MariaDB

Datubāzes TLS savienojums netiek atbalstīts ar `verify_ca` opciju 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, ko var izraisīt īpaši izveidotas atbildes no neuzticamas ierīces. Neuzticama IPMI ierīce var nosūtīt īpaši izveidotus datus, kas izraisa OpenIPMI bibliotēkas avāriju, un tādējādi var tikt pārtraukts Zabbix servera process, 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 acīmredzama iemesla, proti, dažas lapas (vai to daļas) tiek parādītas vienā valodā, bet citas lapas (vai to daļas) - citā valodā. Parasti problēma var rasties, ja ir vairāki lietotāji, no kuriem daži izmanto vienu lokalizāciju, bet citi - citu.

Zināms risinājums ir atspējot vairākpavedienu darbību PHP un Apache.

Problēma ir saistīta ar lokalizācijas iestatīšanas darbību PHP: lokalizācijas informācija tiek uzturēta katram procesam, nevis katram pavedienam. Tāpēc vairākpavedienu vidē, kad viens Apache process izpilda vairākus projektus, ir iespējams, ka lokalizācija tiek mainīta citā pavedienā un tas ietekmē datu apstrādi Zabbix pavedienā.

Papildinformāciju skatiet saistītajos problēmu ziņojumos:

- ZBX-10911 (Lietotāja saskarnes lokalizāciju pārslēgšanās problēma)
- ZBX-16297 (Skaitļu apstrādes problēma 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

Neeksistējoši DNS ieraksti Zabbix aģenta konfigurācijas faila (`zabbix_agentd.conf`) `Server` parametrā var palielināt Zabbix aģenta atbildes laiku operētājsistēmā Windows. Tas notiek tāpēc, ka Windows DNS kešatmiņas dēmons negatīvās atbildes IPv4 adresēm nekešo. Tomēr IPv6 adresēm negatīvās atbildes tiek kešotas, tāpēc iespējams risinājums ir IPv4 atspējošana hostā.

YAML eksportēšana/importēšana

Pastāv dažas zināmas problēmas ar YAML eksportēšanu/importēšanu:

- Kļūdu ziņojumi nav tulkojami;
- Derīgs JSON ar .yaml faila paplašinājumu dažkārt nevar tikt importēts;
- Cilvēkam lasāmi datumi bez pēdiņām tiek automātiski pārveidoti par Unix laikspiedoliem.

Iestatīšanas vednis SUSE ar NGINX un php-fpm

Lietotāja saskarnes iestatīšanas vednis nevar saglabāt konfigurācijas failu SUSE ar NGINX + php-fpm. To izraisa iestatījums vienībā /usr/lib/systemd/system/php-fpm.service, kas neļauj Zabbix rakstīt uz /etc. (ieviests PHP 7.4).

Ir pieejami divi risinājumi:

- Iestatiet ProtectSystem opciju uz 'true' nevis 'full' php-fpm systemd vienībā.
- Manuāli saglabājiet 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 pārejas uz PCRE2

Zabbix 6.0 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 daudz priekšrocību, pāreja uz PCRE2 var izraisīt to, ka daži esošie PCRE regexp paraugi kļūst nederīgi vai darbojas citādi. Īpaši tas ietekmē paraugu *[\w-\.]*. Lai šo regexp atkal padarītu derīgu, nemainot semantiku, mainiet izteiksmi uz *[-\w\.]* . Tas notiek tāpēc, ka PCRE2 domuzīmi uztver kā norobežotāju, izveidojot diapazonu rakstzīmju klasē.

Geomap widget kļūda

Geomap widget kartes var netikt ielādētas pareizi, ja esat veicis jauninašanu no vecākas Zabbix versijas ar NGINX un jauninašanas laikā nepārslēdzāties uz jauno NGINX konfigurācijas failu.

Lai novērstu 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, 𐐀, 𐐨). 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.

Profilēšana, izmantojot tcmalloc, ja atmiņas patēriņš ir pārmērīgs

Ja jums ir aizdomas, ka jūsu Zabbix instalācija izmanto pārāk daudz atmiņas, varat izmantot tcmalloc atmiņas profilēšanas funkciju, lai izpētītu Zabbix servera/starpniekservera atmiņas patēriņu.

1. Instalējot Zabbix no avota kodiem, konfigurējiet papildu karogus:

```
export CFLAGS="-std=gnu99 -g -O0"
```

Karogs `-std=gnu99` ir nepieciešams Zabbix servera, Zabbix starpniekservera vai Zabbix aģenta būvēšanai. 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 un ziņot par atmiņas patēriņu:

```
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. Izveidojiet profila izdruku, nosūtot signālu 5 mērķa procesam. 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 visiem atmiņas piešķīrumiem.

Skatiet arī:

- ZBX-25050 un ZBX-25584 saistītajiem problēmu ziņojumiem.
- GCC Option Summary par kompilēšanas opcijām (`-std=gnu99`, `-g`, `-O0` u. c.).
- Gperftools Heap Profiler dokumentāciju par tcmalloc profilēšanas vides mainīgajiem.

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
```

Šķiet, ka šo kļūdu izraisa problēmas ar rollback operācijām, kas saistītas ar ārējās atslēgas ierobežojumiem.

Skatiet arī:

- ZBX-26060 saistītās problēmas ziņojumam.
- MySQL Bug #96758 "Rollbacks with Foreign Keys on single node" augšupējās problēmas aprakstam.

Pieteikšanās un systemd

Mēs iesakām izveidot lietotāju *zabbix* kā sistēmas lietotāju, proti, bez iespējas pieteikties. Daži lietotāji šo ieteikumu ignorē un izmanto to pašu kontu, lai pieteiktos (piemēram, izmantojot SSH) hostā, kurā darbojas Zabbix. Tas var izraisīt Zabbix dēmona avāriju pēc atteikšanās. Šādā gadījumā Zabbix servera žurnālā tiks parādīts aptuveni šāds ieraksts:

```
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 systemd noklusējuma iestatījuma `RemoveIPC=yes` dēļ, kas ir konfigurēts failā `/etc/systemd/logind.conf`. Kad atsakāties no sistēmas, iepriekš Zabbix izveidotie semafori tiek noņemti, izraisot avāriju.

Citāts no systemd dokumentācijas:

```
RemoveIPC=

Controls whether System V and POSIX IPC objects belonging to the user shall be removed when the
user fully logs out. Takes a boolean argument. If enabled, the user may not consume IPC resources
after the last of the user's sessions terminated. This covers System V semaphores, shared memory
and message queues, as well as POSIX shared memory and message queues. Note that IPC objects of the
root user and other system users are excluded from the effect of this setting. Defaults to "yes".
```

Šai problēmai ir 2 risinājumi:

1. (ieteicams) Pārtrauciet izmantot kontu *zabbix* jebkam citam, izņemot Zabbix procesus, un izveidojiet atsevišķu kontu citām darbībām.
2. (nav ieteicams) Iestatiet `RemoveIPC=no` failā `/etc/systemd/logind.conf` un restartējiet sistēmu. Ņemiet vērā, ka `RemoveIPC` ir visas sistēmas parametrs; tā maiņa ietekmēs visu sistēmu.

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 darba sākšanai. Katra Zabbix instalācija ir unikāla. Noteikti veiciet savas Zabbix sistēmas veiktspējas etalonmērījumus sagatavošanas vai izstrādes vidē, lai pirms Zabbix instalācijas izvietošanas produkcijas vidē pilnībā izprastu savas prasības.

Instalācijas izmērs	Pārraudzītie rādītāji[1]	CPU/vCPU kodoli	Atmiņa (GiB)	Datu bāze	Amazon EC2[2]
Mazs	1 000	2	8	MySQL Server, Percona Server, MariaDB Server, PostgreSQL	m6i.large/m6g.large
Vidējs	10 000	4	16	MySQL Server, Percona Server, MariaDB Server, PostgreSQL	m6i.xlarge/m6g.xlarge
Liels	100 000	16	64	MySQL Server, Percona Server, MariaDB Server, PostgreSQL	m6i.4xlarge/m6g.4xlarge
Ļoti liels	1 000 000	32	96	MySQL Server, Percona Server, MariaDB Server, PostgreSQL	m6i.8xlarge/m6g.8xlarge

[1] Izmēra noteikšanas nolūkos 1 pārraudzītais rādītājs aptuveni tiek attēlots ar formulu: 1 rādītājs = 1 vienums + 1 trigeris + 1 grafiks. Šis ir tikai aptuvens novērtēšanas noteikums, nevis formula pārraudzīto rādītāju aprēķināšanai.
 [2] Piemērs ar Amazon vispārēja lietojuma EC2 instancēm, izmantojot ARM64 vai x86_64 arhitektūru. Pirms instalēšanas produkcijas vidē, Zabbix instalācijas novērtēšanas un testēšanas laikā jāizvēlas atbilstošs instances tips, piemēram, optimizēts skaitļošanai, atmiņai vai krātuvei.

Note:
Faktiskā konfigurācija lielā mērā ir atkarīga no aktīvo vienumu skaita un atsvaidzināšanas biežuma (detalizētu informāciju skatiet šīs lapas sadaļā datu bāzes izmērs). Lielām instalācijām ir ļoti ieteicams datu bāzi darbināt atsevišķā serverī.

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	-	

Platforma	Serveris	Aģents	Aģents 2	Komentāri
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.

Trešās puses ārējā apkārtējā programmatūra

Ja programmatūra/bibliotēka ir norādīta kā obligāta, tā ir absolūti nepieciešama. Izvēles programmatūra ir nepieciešama dažu specifisku funkciju atbalstam.

Programmatūra	Obligātuma statuss	Atbalstītās versijas	Komentāri
MySQL/Percona	Viena no	8.4.0-9.7.X	Nepieciešama, ja MySQL (vai Percona) tiek izmantots kā Zabbix aizmugursistēmas datubāze. Nepieciešams InnoDB dzinis. Servera/starpniekservera izveidei iesakām izmantot C API (libmysqlclient) bibliotēku. Ražošanas vidēs stingri iesakām izmantot tikai LTS laidienus. Laidieni ar apzīmējumu ”rolling release”, ”early access” vai līdzīgu apzīmējumu nav ieteicami.
MariaDB		10.11.00-12.3.X	Nepieciešams InnoDB dzinis. Servera/starpniekservera izveidei iesakām izmantot MariaDB Connector/C bibliotēku. Skatiet arī: Iespējamās strupceļa situācijas ar MariaDB un Piekļuve lietotāja saskarnes elementiem ar MariaDB 10.5.1-10.5.9. Ražošanas vidēs stingri iesakām izmantot tikai LTS laidienus. Laidieni ar apzīmējumu ”rolling release”, ”early access” vai līdzīgu apzīmējumu nav ieteicami.
PostgreSQL		15.0-18.X	Nepieciešama, ja PostgreSQL tiek izmantots kā Zabbix aizmugursistēmas datubāze. Atkarībā no instalācijas lieluma var būt nepieciešams palielināt PostgreSQL konfigurācijas parametra *work_mem* vērtību (noklusējuma vērtība ir 4 MB), lai datubāze konkrētai darbībai izmantotu pietiekamu atmiņas apjomu un vaicājuma izpilde neaizņemtu pārāk daudz laika.
TimescaleDB for PostgreSQL		2.20.X-2.29.X	Nepieciešama, ja TimescaleDB tiek izmantota kā PostgreSQL datubāzes paplašinājums. Noteikti instalējiet TimescaleDB Community Edition, kas atbalsta saspiešanu. Sīkāku informāciju par PostgreSQL un TimescaleDB versiju saderību skatiet TimescaleDB dokumentācijā.

Programmatūra	Obligātuma statuss	Atbalstītās versijas	Komentāri
Elasticsearch		7.X	Nepieciešams, ja Elasticsearch tiek izmantots kā Zabbix aizmugursistēmas datubāze. Elasticsearch var izmantot tikai vēstures datu glabāšanai, un tas tiek atbalstīts tikai ar Zabbix serveriem. Elasticsearch atbalsts pašlaik ir eksperimentāls. Skatiet arī serverim/starpniekserverim nepieciešamo programmatūru server/proxy.
ClickHouse		26.3.0.0-26.9.X.X	Nepieciešams, ja ClickHouse tiek izmantots kā Zabbix aizmugursistēmas datubāze vēstures datu glabāšanai (tiek atbalstīts tikai ar Zabbix serveriem) vai kā telemetrijas datu avots. Lai ClickHouse izmantotu kā telemetrijas datu avotu, nepieciešama ClickHouse versija 26.8.17.4 vai jaunāka. Sistēmas laikam Zabbix serverī un ClickHouse serverī jābūt sinhronizētam; pretējā gadījumā telemetrijas vaicājumi var neiekļaut dažus datus vai atgriezt vienus un tos pašus datus divreiz.
SQLite	Izvēles	3.3.5-3.53.X	SQLite tiek atbalstīta tikai ar Zabbix starpniekserveriem. Nepieciešama, ja SQLite tiek izmantota kā Zabbix starpniekservera datubāze.
smartmontools		7.1 vai jaunāka	Nepieciešama Zabbix aģentam 2.
who			Nepieciešama lietotāju skaita spraudnim.
dpkg			Nepieciešama spraudnim system.sw.packages.
pkgtool			Nepieciešama spraudnim system.sw.packages.
rpm			Nepieciešama spraudnim system.sw.packages.
pacman			Nepieciešama spraudnim system.sw.packages.
q applets			`qlist` un `qsize`, kas ir daļa no q applets, ir nepieciešami spraudnim system.sw.packages operētājsistēmā Gentoo Linux.

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

Lietotāja saskarne

Zabbix lietotāja saskarnes minimālais atbalstītais ekrāna platums ir 1200 px.

Ja programmatūra/bibliotēka ir norādīta kā obligāta, tā ir noteikti nepieciešama. Neobligātā programmatūra ir nepieciešama dažu konkrētu funkciju atbalstam.

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 Trešās puses ārējo apkārtējo program-matūru	
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*)

Programmatūra	Obligātuma statuss	Atbalstītās versijas	Komentāri
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*). Dažām lietotāja saskarnes valodām var būt nepieciešama versija 2.3.0 vai jaunāka, lai novērstu iespējamu teksta pārklāšanos grafikos.
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, SMTP autentifikācijai un telemetrijas datu avota testēšanai.

Trešās puses 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 starppārlūkprogrammu izstrādes procesu.
jQuery UI		1.12.1	Lietotāja saskarnes mijiedarbību, efektu, logrīku un motīvu kopa, kas izveidota uz jQuery bāzes.
SAML PHP Toolkit		4.3.1	PHP rīkkopa, kas nodrošina SAML 2.0 autentifikācijas atbalstu, lai varētu pierakstīties Zabbix.
Duo Universal PHP library		1.1.2	PHP bibliotēka, kas nodrošina Duo Universal Prompt daudzfaktoru autentifikācijas atbalstu Zabbix.
Symfony Yaml Component		5.1.0	Nodrošina Zabbix konfigurācijas elementu eksportēšanu un importēšanu YAML formātā.

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

Attention:
Fontiem, kas nav noklusējuma DejaVu fonts, var būt nepieciešama PHP funkcija imagerotate. Ja tās trūkst, šie fonti, attēlojot grafiku, var tikt renderēti nepareizi. Šī funkcija ir pieejama tikai tad, ja PHP ir kompilēts ar komplektācijā iekļauto GD, kas nav Debian un citu izplatījumu gadījumā.

Trešās puses 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 un nepieciešamo bibliotēku atkarību pārvaldībai.
PHPUnit		8.5.29	PHP vienību testēšanas ietvars Zabbix lietotāja saskarnes testēšanai.
SASS		3.4.22	Priekšprocesora skriptu valoda, kas tiek interpretēta un kompilēta kaskādes 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 programmatūra/bibliotēka ir norādīta kā obligāta, tā ir absolūti nepieciešama. Neobligātā programmatūra/bibliotēkas ir nepieciešamas noteiktu funkciju atbalstam.

Prasība	Obligātuma statuss	Apraksts
libpcre2	Jā	PCRE2 bibliotēka ir nepieciešama Perl saderīgo regulāro izteiksmju (PCRE) atbalstam. Tiek atbalstīts 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 libc sastāvdaļa).
libresolv		Nepieciešama DNS izšķiršanai (var būt libc sastāvdaļa).
libiconv		Nepieciešama teksta kodējuma/formāta konvertēšanai (var būt libc sastāvdaļa). Obligāta Zabbix serverim operētājsistēmā Linux.
libz		Nepieciešama saspiešanas atbalstam.
libm		Matemātikas bibliotēka. Nepieciešama tikai Zabbix serverim.
libmysqlclient	Viena no	Nepieciešama, ja tiek izmantots MySQL.
libmariadb		Nepieciešama, ja tiek izmantots MariaDB.
libpq5		Nepieciešama, ja tiek izmantots PostgreSQL; *libpq5* versijai jāatbilst izmantotās PostgreSQL datubāzes versijai vai jābūt jaunākai par to.
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 funkcijām: - Tīmekļa uzraudzībai, VMware uzraudzībai un HTTP aģenta vienumiem (visiem: versija 7.19.1 vai jaunāka); - Zabbix aģenta web.page.* vienumiem (versija 7.19.1 vai jaunāka; skatiet arī aģenta prasības); - SMTP autentifikācijai (Basic: versija 7.20.0 vai jaunāka; OAuth: versija 7.33 vai jaunāka; skatiet arī lietotāja saskarnes prasības); - Elasticsearch (versija 7.28.0 vai jaunāka); - Telemetrijas vaicājuma vienumiem (versija 7.28.0 vai jaunāka). Ieteicams izmantot versiju 7.28.0 vai jaunāku visām funkcijām. Lai izmantotu uzlabotās cURL funkcijas `web.page.*` vienumiem, restartējiet Zabbix serveri/starpniekserveri. SMTP autentifikācijai izpildes laikā izmantojiet pakotni `libcurl-full`.
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 versiju.
libunixodbc		Nepieciešama datubāzu 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.

Prasība	Obligātuma statuss	Apraksts
c-ares		Nepieciešama asinhronai DNS izšķiršanai, ja Zabbix ir konfigurēts ar opciju `--with-ares`. Pretējā gadījumā tiks izmantots *libevent*. Minimālā versija: 1.16.0. DNS vaicājumu kešatmiņas un kļūmjpārejas funkcijai (vaicājumu kešatmiņa pēc noklusējuma ir iespējota) nepieciešama c-ares versija 1.26.0 vai jaunāka. Sistēmās, kur c-ares nodrošina operētājsistēmas pakotne, pārliecinieties, ka instalētā izpildlaika bibliotēka nodrošina DNS kešatmiņai un kļūmjpārejai nepieciešamās izšķīrēja funkcijas.
g++		C++ kompilators. Nepieciešams, lai kompilētu Zabbix starpniekserveri ar opciju `--with-apm` lietojumprogrammu veiktspējas uzraudzībai (APM) (/manual/concepts/proxy/distributed_monitoring#adding-proxies).
gRPC		Nepieciešams APM Zabbix starpniekserverī, ja tas ir kompilēts ar opciju `--with-apm`. Nepieciešamās bibliotēkas: *grpc++*, *grpc*, *gpr*. Zabbix starpniekservera kompilēšanai nepieciešama arī utilīta `grpc_cpp_plugin`. Nepieciešamā versija ir pieejama šo operētājsistēmu standarta pakotņu krātuvēs: - Ubuntu 23.04 vai jaunāka versija; - Debian 12 (Bookworm) vai jaunāka versija; - Alpine Linux 3.15 vai jaunāka versija; - RHEL, Rocky Linux, CentOS Stream 9 vai jaunāka versija (tikai no EPEL krātuves); - Fedora X vai jaunāka versija. Vecākās operētājsistēmu versijās *gRPC* jāizveido no pirmkoda.
protobuf		Nepieciešama APM Zabbix starpniekserverī, ja tas ir kompilēts ar opciju `--with-apm`. Zabbix starpniekservera kompilēšanai nepieciešams arī kompilators `protoc`.
Abseil		Nepieciešama APM Zabbix starpniekserverī, ja tas ir kompilēts ar opciju `--with-apm`. Nepieciešamās bibliotēkas: *absl_synchronization*, *absl_base*. Parasti tiek instalēta kopā ar *gRPC*.

Aģents

Requirement	Mandatory status	Description
libpcre2	Yes	PCRE2 bibliotēka ir nepieciešama Perl Compatible Regular Expression (PCRE) atbalstam. Tiek atbalstīta PCRE2 v10.x. Nepieciešama žurnālu monitorēšanai. Nepieciešama arī operētājsistēmā Windows.
libpthread		Nepieciešama mutex un lasīšanas-rakstīšanas slēdžu atbalstam (var būt daļa no libc). Nav nepieciešama operētājsistēmā Windows.
libresolv		Nepieciešama DNS atrisināšanai (var būt daļa no libc). Nav nepieciešama operētājsistēmā Windows.
libiconv		Nepieciešama teksta kodējuma/formāta pārveidošanai uz UTF-8 žurnālu vienumos, faila saturā, faila regex un regmatch vienumos (var būt daļa no libc). Nav nepieciešama operētājsistēmā Windows.
libgnutls or *libopenssl*	No	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. Nav atbalstīta operētājsistēmā Windows.
libcurl		Nepieciešama paplašinātam Zabbix aģents web.page.* vienumu atbalstam. Bez libcurl ir pieejama pamata funkcionalitāte (piem., web.page.get[http://example.com]). Ar libcurl aģents atbalsta papildu iespējas, piemēram, HTTP URL ar akreditācijas datiem (piem., http://user:password@example.com) un HTTPS URL. Nepieciešama versija 7.19.1 vai jaunāka (ieteicama 7.28.0 vai jaunāka). Lai izmantotu atjaunināts cURL funkcijas, restartējiet Zabbix aģents.
libmodbus		Nepieciešama tikai tad, ja tiek izmantota Modbus monitorēšana. Versija 3.0 vai jaunāka.

Aģents 2

Requirement	Mandatory status	Description
Go	Yes	Nepieciešams, lai no avota izveidotu Zabbix aģents 2 un tā spraudņus. Tiek atbalstīts Go 1.25 vai jaunāks. Instalēšanas norādījumus skatiet go.dev. Zabbix aģents 2 un tā spraudņu izmantotās Go bibliotēkas ir uzskaitītas Zabbix Git repozitorijā (repozitorijā ar `indirect` atzīmētās bibliotēkas ir citu nepieciešamo bibliotēku atkarības): - Zabbix aģents 2 - Spraudņu atbalsts - PostgreSQL - MongoDB - MSSQL - Ember+ - NVIDIA GPU - Piemēra spraudnis
libpcre2	Yes	PCRE2 bibliotēka ir nepieciešama Perl Compatible Regular Expression (PCRE) atbalstam. Tiek atbalstīta PCRE2 v10.x. Nepieciešama žurnālu uzraudzībai. Nepieciešama arī operētājsistēmā Windows.
libopenssl	No	Nepieciešama, izmantojot šifrēšanu. UNIX platformās nepieciešama OpenSSL 1.0.1 vai jaunāka versija. OpenSSL bibliotēkai jābūt ar iespējotu PSK atbalstu. LibreSSL netiek atbalstīta. Microsoft Windows sistēmās nepieciešama OpenSSL 1.1.1 vai jaunāka versija.

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, nepieciešamās atkarības jau ir iekļautas pirmkoda kokā.

Ja Zabbix ieguvāt no savas distribūcijas pakotnes, 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), tālāk ir norādīts to bibliotēku versiju saraksts, ar kurām Java gateway darbība ir pārbaudīta. Zabbix var darboties arī ar citām šo bibliotēku versijām.

Tālāk esošajā tabulā ir uzskaitīti JAR faili, kas pašlaik ir iekļauti Java gateway sākotnējā kodā:

Bibliotēka	Obligāta	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		1.5.25	
logback-core		1.5.25	
slf4j-api		2.0.16	

Java gateway var izveidot, izmantojot Oracle Java vai atvērtā pirmkoda OpenJDK (8. vai jaunāku versiju). Zabbix nodrošinātās pakotnes tiek kompilētas, izmantojot OpenJDK. Tālāk esošajā tabulā ir uzskaitītas OpenJDK pakotnes, ko distribūcijas izmanto Zabbix pakotņu izveidei:

Distribūcija	OpenJDK pakotne
AlmaLinux 10	`java-21-openjdk-devel` (amd64, arm64: 21.0.12.0.8-1.2)
AlmaLinux 9	`java-17-openjdk-devel` (amd64, arm64: 17.0.15.0.6-2)
AlmaLinux 8	`java-1.8.0-openjdk-devel` (amd64: 1.8.0.332.b09-2; arm64: 1.8.0.382.b05-2)
Amazon Linux 2023	`java-22-amazon-corretto-devel` (amd64, arm64: 22.0.2+9-1)
CentOS Stream 10	`java-21-openjdk-devel` (amd64, arm64: 21.0.11.0.10-2)

Distribūcija	OpenJDK pakotne
CentOS Stream 9	`java-17-openjdk-devel` (amd64, arm64: 17.0.15.0.6-2)
CentOS Stream 8	`java-1.8.0-openjdk-devel` (amd64, arm64: 1.8.0.362.b08-3)
CentOS 7	`java-1.8.0-openjdk-devel` (amd64: 1.8.0.282.b08-1)
Debian 13	`default-jdk-headless` (amd64, arm64: 2:1.21-76)
Debian 12	`default-jdk-headless` (amd64, arm64: 2:1.17-74)
Debian 11	`default-jdk-headless` (amd64: 2:1.11-72)
OpenSUSE Leap 16	`java-21-openjdk-devel` (amd64, arm64: 21.0.11.0-160000.2.1)
OpenSUSE Leap 15	`java-17-openjdk-devel` (amd64: 17.0.5.0-150400.3.9.3; arm64: 17.0.8.0-150400.3.27.1)
Oracle Linux 10	`java-21-openjdk-devel` (amd64, arm64: 21.0.11.0.10-2.0.1)
Oracle Linux 9	`java-17-openjdk-devel` (amd64, arm64: 17.0.15.0.6-2.0.1)
Oracle Linux 8	`java-1.8.0-openjdk-devel` (amd64: 1.8.0.372.b07-4.0.1); `java-11-openjdk-devel` (arm64: 11.0.20.0.8-3.0.1)
Oracle Linux 7	`java-1.8.0-openjdk-devel` (amd64: 1.8.0.282.b08-1)
Raspberry Pi OS 13	`default-jdk-headless` (arm64, armhf: 2:1.21-76)
Raspberry Pi OS 12	`default-jdk-headless` (arm64, armhf: 2:1.17-74)
Raspberry Pi OS 11	`default-jdk-headless` (arm64: 2:1.11-72; armhf: 2:1.11-72+b4)
RHEL 10	`java-21-openjdk-devel` (amd64, arm64: 21.0.12.0.8-1.2)
RHEL 9	`java-17-openjdk-devel` (amd64, arm64: 17.0.15.0.6-2)
RHEL 8	`java-1.8.0-openjdk-devel` (amd64: 1.8.0.372.b07-4; arm64: 1.8.0.382.b05-2)
RHEL 7	`java-1.8.0-openjdk-devel` (amd64: 1.8.0.282.b08-1)
Rocky Linux 10	`java-21-openjdk-devel` (amd64, arm64: 21.0.12.0.8-1.2)
Rocky Linux 9	`java-17-openjdk-devel` (amd64, arm64: 17.0.15.0.6-2)
Rocky Linux 8	`java-1.8.0-openjdk-devel` (amd64: 1.8.0.372.b07-4; arm64: 1.8.0.382.b05-2)
SLES 16	`java-21-openjdk-devel` (amd64, arm64: 21.0.11.0-160000.1.1)
SLES 15	`java-17-openjdk-devel` (amd64: 17.0.5.0-150400.3.9.3; arm64: 17.0.8.0-150400.3.27.1)
Ubuntu 26.04	`default-jdk-headless` (amd64, arm64: 2:1.25-77)
Ubuntu 24.04	`default-jdk-headless` (amd64, arm64: 2:1.21-75+exp1)
Ubuntu 22.04	`default-jdk-headless` (amd64, arm64: 2:1.11-72build2)
Ubuntu 20.04	`default-jdk-headless` (amd64, arm64: 2:1.11-72)

Tilta adapteris

Zabbix tilta adapterim nav īpašu prasību.

Noklusējuma porti

Šajā sarakstā ir norādīti porti, kurus Zabbix komponenti var izmantot noklusējuma konfigurācijā:

Zabbix komponents	Protokols un ports	Savienojumu sāk
Zabbix aģents	TCP 10050	Zabbix serveris vai Zabbix starpniekserveris (tikai pasīvās pārbaudes)
Zabbix aģents 2	TCP 10050	Zabbix serveris vai Zabbix starpniekserveris (tikai pasīvās pārbaudes)
Zabbix serveris	TCP 10051	Aktīvie Zabbix aģenti, aktīvais Zabbix starpniekserveris, Zabbix sender, Zabbix lietotāja saskarne
Zabbix starpniekserveris	TCP 10051	Aktīvie Zabbix aģenti, Zabbix sender, Zabbix serveris (tikai pasīvais starpniekserveris)
Zabbix Java gateway	TCP 10052	Zabbix serveris vai starpniekserveris
Zabbix tīmekļa serviss	TCP 10053	Zabbix serveris
Zabbix MCP serveris	TCP 8443	MCP klienti (MI aģenti)

Zabbix komponents	Protokols un ports	Savienojumu sāk
Zabbix lietotāja saskarne	TCP 80	Tīmekļa pārlūkprogrammas, Zabbix tīmekļa serviss (ieplānotie pārskati), API klienti
	TCP 443	
SNMP slazdi (snmptrapd)	UDP 162	SNMP ierīces
Zabbix bridge adapteris	TCP 10054	Zabbix serveris

Warning:
Atveriet tikai tos portus, kas nepieciešami jūsu izmantotajiem komponentiem, un ierobežojiet katru ugunsmūra noteikumu līdz konkrētajiem hostiem, kuriem nepieciešams izveidot savienojumu (piemēram, atļaujiet portu 10050 tikai no Zabbix servera vai starpniekservera, nevis no jebkuras adreses). Nepakļaujiet Zabbix portus tieši internetam.
 Pirms portu atvēršanas nostipriniet instalāciju: iespējojiet šifrēšanu starp komponentiem, ierobežojiet atļautos hostus komponentu konfigurācijas failos un pārskatiet drošības paraugpraksi.

Datubāzes izmērs

Zabbix konfigurācijas dati aizņem fiksētu diska vietas apjomu un būtiski nepieaug.

Zabbix datubāzes izmērs galvenokārt ir atkarīgs no šiem mainīgajiem, kas nosaka glabāto 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 atsvaidzināšanas intervālu 60 sekundes, vērtību skaits sekundē tiek aprēķināts kā 3000/60 = **50**.

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

- Housekeeper iestatījumi vēsturei

Zabbix glabā 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 glabāt 30 dienu vēsturi un saņemam 50 vērtības sekundē, kopējais vērtību skaits būs aptuveni (**30***24*3600)* **50** = 129.600.000 jeb apmēram 130M vērtību.

Atkarībā no izmantotā datubāzes dzinēja, saņemto vērtību tipa (peldošie skaitļi, veseli skaitļi, virknes, žurnālfaili utt.), diska vietas apjoms vienas vērtības glabāšanai var svārstīties no 40 baitiem līdz simtiem baitu. Parasti tas ir apmēram 90 baiti uz vienu vērtību skaitliskajiem vienumiem[2]. Mūsu gadījumā tas nozīmē, ka 130M vērtību prasīs 130M * 90 baiti = **10.9GB** diska vietas.

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

- Housekeeper iestatījums tendencēm

Zabbix katram vienumam tabulā **trends** glabā 1 stundas max/min/avg/count vērtību kopu. Šie dati tiek izmantoti tendenču un ilgtermiņa grafiku veidošanai. Vienas stundas periods nav pielāgojams.

Zabbix datubāzei atkarībā no datubāzes tipa nepieciešami aptuveni 90 baiti uz katru ierakstu. Pieņemsim, ka mēs vēlamies glabāt tendenču datus 5 gadus. 3000 vienumu vērtībām būs nepieciešams 3000*24*365* **90** = **2.2GB** gadā jeb **11GB** 5 gadiem.

- Housekeeper iestatījumi notikumiem

Katram Zabbix notikumam ir nepieciešami aptuveni 250 baiti diska vietas[1]. Ir grūti novērtēt, cik daudz 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 tiek atjaunoti, tāpēc var pieņemt vienu event_recovery ierakstu uz katru notikumu. Tas nozīmē papildu 80 baitus uz vienu notikumu.

Papildus notikumiem var būt tagi, un katram taga ierakstam nepieciešami aptuveni 100 baiti diska vietas[1]. Tagu skaits uz notikumu (#tags) ir atkarīgs no konfigurācijas. Tātad katram būs nepieciešami papildu #tags * 100 baiti diska vietas.

Tas nozīmē, ka, ja mēs vēlamies glabāt 3 gadu notikumus, tam būs nepieciešams 3*365*24*3600* (250+80+#tags*100) = **~30GB**+#tags*100B diska vietas[2].

Note:
[1] Vairāk, ja notikumu nosaukumos, tagos un vērtībās ir ne-ASCII rakstzīmes.
 [2] Izmēra aplēses ir balstītas uz MySQL un var atšķirties citās datubāzēs.

Tabulā ir formulas, ko var izmantot, lai aprēķinātu Zabbix sistēmai nepieciešamo diska vietu:

Parameter	Formula for required disk space (in bytes)
Zabbix configuration	Fiksēts izmērs. Parasti 10MB vai mazāk.
History	days*(items/refresh rate)*24*3600*bytes items : vienumu skaits days : dienu skaits, cik ilgi glabāt vēsturi refresh rate : vienumu vidējais atsvaidzināšanas 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 glabāt vēsturi bytes : baitu skaits, kas nepieciešams vienas tendences glabāšanai; atkarīgs no datubāzes dzinēja, parasti ~90 baiti.
Events	days*events*24*3600*bytes events : notikumu skaits sekundē. Sliktākajā gadījumā viens (1) notikums sekundē. days : dienu skaits, cik ilgi glabāt vēsturi bytes : baitu skaits, kas nepieciešams viena notikuma glabāšanai; atkarīgs no datubāzes dzinēja, parasti ~330 + vidējais tagu skaits uz notikumu * 100 baiti.

Tātad kopējo nepieciešamo diska vietu var aprēķināt šādi:

Konfigurācija + Vēsture + Tendences + Notikumi

Diska vieta netiks izmantota uzreiz pēc Zabbix instalēšanas. Datubāzes izmērs pieaugs un pēc tam noteiktā brīdī pārtrauks pieaugt, kas ir atkarīgs 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.

Kas jauns un jauninašanas piezīmes: 8.0

Zabbix 8.0.0 ir turpinājums Zabbix 7.4.0, pievienojot jaunas funkcijas un uzlabojumus.

Šajā lapā arī:

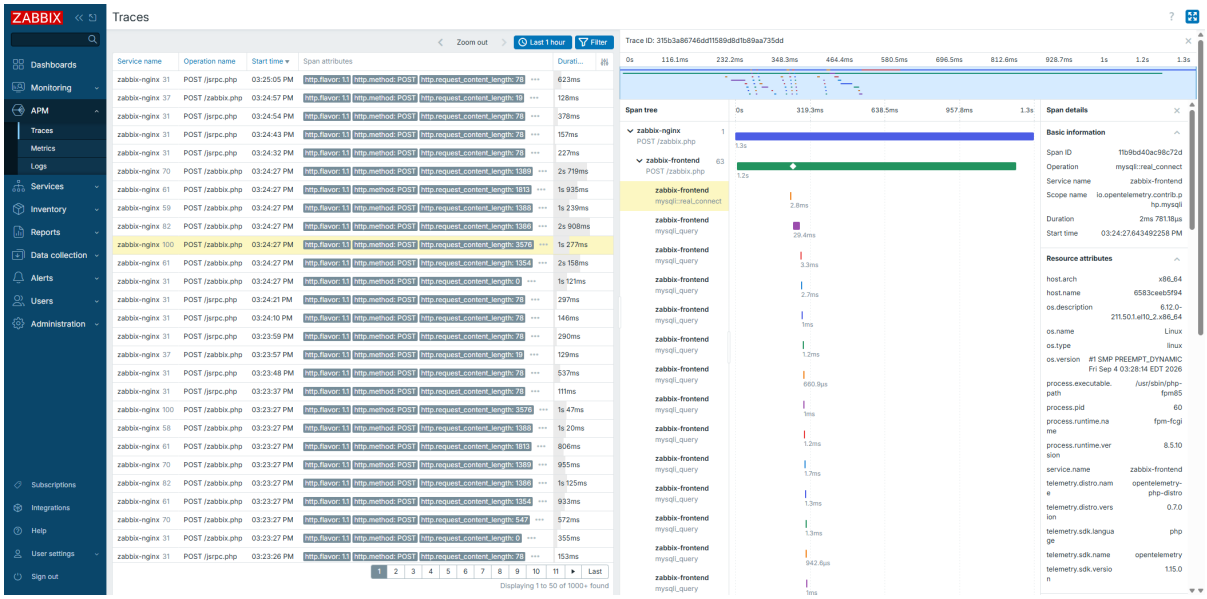
- Kas jauns veidnēs, tostarp jaunas veidnes un izmaiņas esošajās veidnēs.
- Jauninašanas piezīmes, tostarp nesaderīgas izmaiņas.

Skatiet arī: API izmaiņas.

Lietojumprogrammu veiktspējas uzraudzība Zabbix tagad atbalsta lietojumprogrammu veiktspējas uzraudzību (APM), kuras pamatā ir OpenTelemetry — atvērts standarts telemetrijas datu apkopošanai no lietojumprogrammām un pakalpojumiem. Šie dati palīdz noskaidrot, kā pieprasījumi plūst caur lietojumprogrammām un pakalpojumiem, izmērīt to veiktspēju un atrast lēnu vai neveiksmīgu darbību cēloni.

Izmantojot APM, varat:

- Apkopot telemetrijas datus ar Zabbix starpniekserveri, kas tagad var saņemt datus no lietojumprogrammām vai OpenTelemetry kolektoriem, izmantojot OpenTelemetry protokolu (OTLP/gRPC), un ierakstīt tos datu avotā; skatiet jauno cilni *APM* sadaļā starpniekservera konfigurācija.
- Norādīt telemetrijas datu avotu (ClickHouse) jaunajā sadaļā *Administrēšana* > *Datu avots* un skatīt datus jaunajā izvēlnē *APM*.
- Izmantot telemetrijas datus trigeros un grafikos, izmantojot jauno telemetrijas vaicājuma vienuma tipu.



Ir pievienoti arī šādi parametri un vienumi:

- Zabbix servera parametri: `TelemetryProvider` un `StartTelemetryQueryPollers`.
- Zabbix starpniekservera parametri: `TelemetryProvider`, `StartTelemetryQueryPollers`, `StartAPMCollectors`, `APMListenIP`, `APMListenPort`, `APMTLSCAFile`, `APMTLSCertFile` un `APMTLSKeyFile`.
- Zabbix lietotāja saskarnes parametrs: `$TELEMETRY_PROVIDERS[]`.
- Zabbix starpniekservera kompilēšanas opcija: `--with-apm`; oficiālās Zabbix starpniekservera pakotnes tiek veidotas ar šo opciju.
- Iekšējais vienums telemetrijas datu apkopošanas uzraudzībai: `zabbix[apm,<type>]`.

ClickHouse atbalsts Tagad varat izmantot ClickHouse kā Zabbix aizmugursistēmas datubāzi, lai glabātu vienumu vērtību vēsturi, kā arī kā telemetrijas datu avotu lietojumprogrammu veiktspējas uzraudzībai.

Informāciju par vēstures glabāšanas iestatīšanu un prasībām skatiet sadaļā ClickHouse iestatīšana. Ja vēlaties migrēt vēstures datus no esošas Zabbix datubāzes (MySQL vai PostgreSQL) uz ClickHouse, skatiet ClickHouse shēmas un vēstures migrācijas skriptus.

Šī funkcija maina arī veidu, kā Zabbix konfigurācijas failos iestatāt vēstures glabāšanas nodrošinātājus. Šīs izmaiņas attiecas gan uz ClickHouse, gan Elasticsearch iestatījumiem. Plašāku informāciju skatiet sadaļā Atjaunināšanas piezīmes.

Sarežģīta notikumu apstrāde Sarežģīta notikumu apstrāde (CEP) veic uz noteikumiem balstītu Zabbix notikumu apstrādi.

Piemēram, iespējams:

- grupēt saistītos notikumus un atzīmēt pirmo notikumu sērijā kā cēloni (bet nākamos — kā simptomus)
- norādīt notikumu derīguma periodu
- norādīt noteiktu notikumu maksimālo skaitu
- saņemt paziņojumus par līdzīgiem notikumiem pēc atrašanās vietas
- saņemt paziņojumus, ja kāds notikums noteiktā laika periodā notiek pārāk bieži (vai nenotiek vispār)

Sarežģīta notikumu apstrāde notikumu analīzei ievieš slīdošus **laika logus**. Piemēram, 10 minūšu logā visi notikumi, kas ietilpst šajā logā, veido vienotu kontekstu. Logam virzoties uz priekšu, vecāki notikumi to atstāj (tiek "izstumti"), savukārt tajā var nonākt jauni notikumi.

Laika loga koncepcija nodrošina loģiskus punktus darbību izpildei. Piemēram, notikumu var pārveidot vai "papildināt", kad tas nonāk logā — var paaugstināt vai pazemināt smaguma pakāpi, pievienot vai pielāgot tagus, aktivizēt apspiešanu u. c.

Līdzīgas darbības var izpildīt, kad notikums atstāj logu vai kad tiek atrasts noteikts modelis.

Notikumu plūsmas analīzi nosaka CEP noteikumi. CEP noteikums parasti sastāv no laika loga, nosacījumiem un darbībām. Darbības ir saistītas ar izpildes punktu.

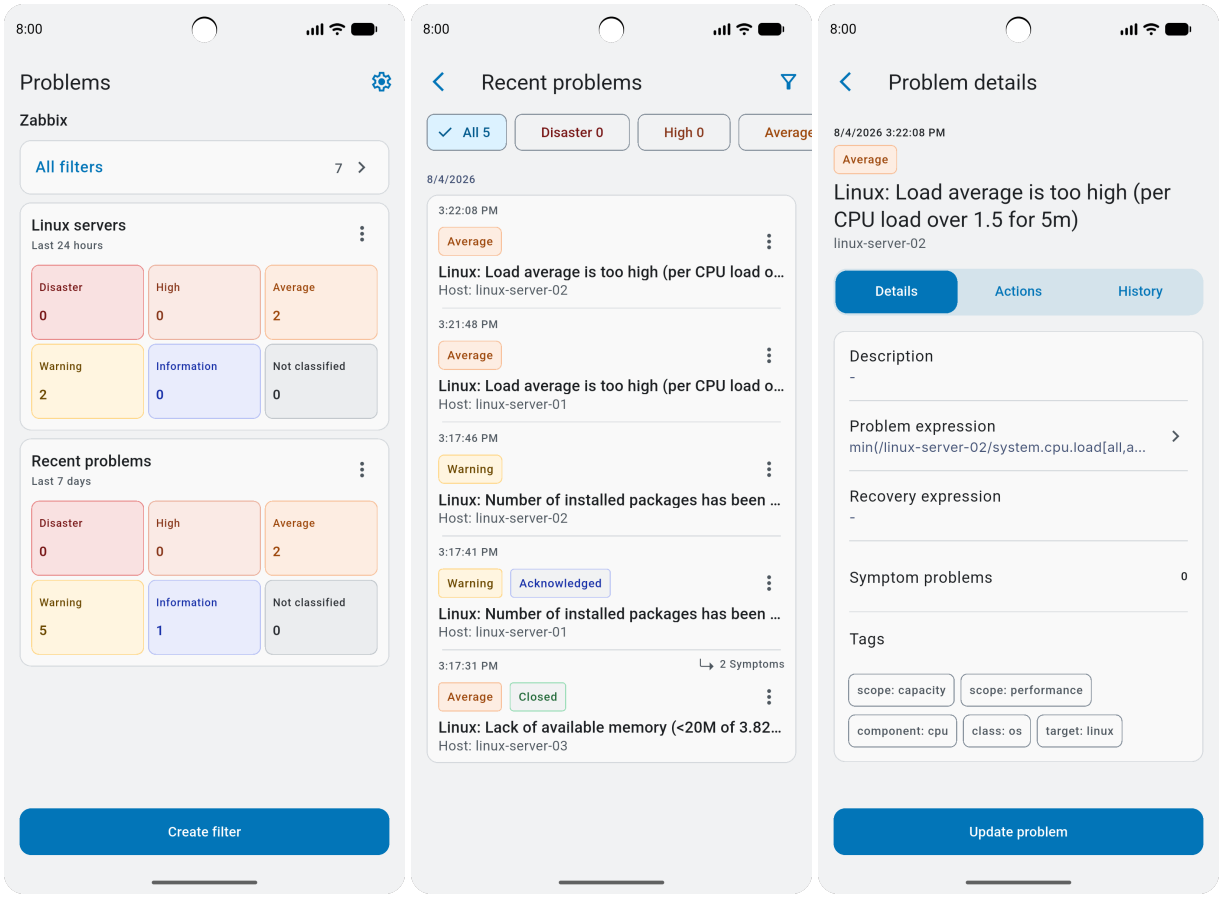
Detalizētāka dokumentācija būs pieejama, kad tiks izlaista stabilā 8.0 versija.

Mobilā lietotne Zabbix mobilā lietotne (drīzumā) ļauj pārraudzīt savu Zabbix vidi no mobilās ierīces.
Skatiet problēmas, pārbaudiet to detalizētu informāciju, atjauniniet problēmas un saņemiet pašpiegādes paziņojumus, kad nepieciešama uzmanība.

Lietotnē Zabbix 8.0 ir pieejami šādi jauni līdzekļi:

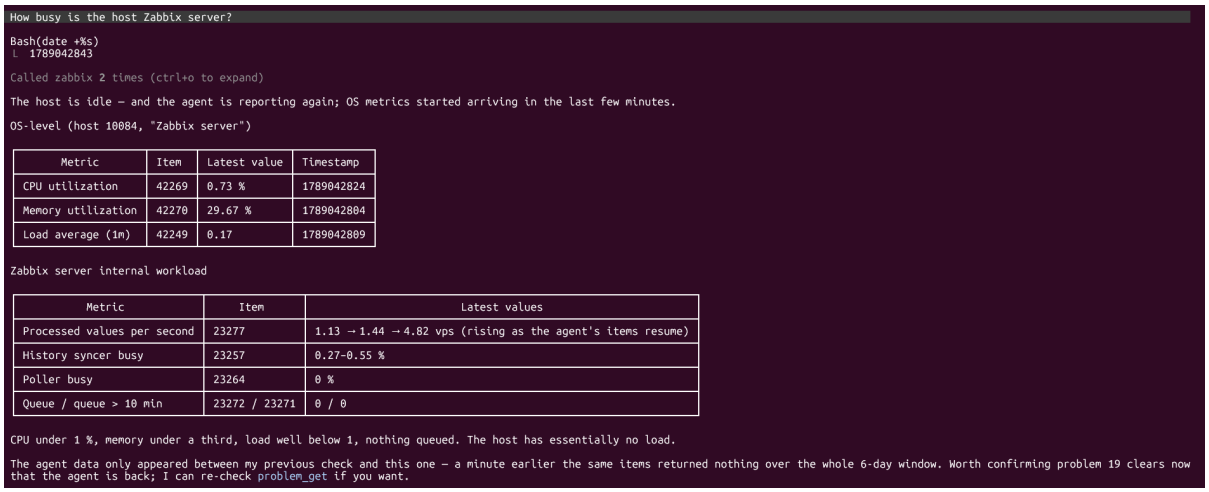
- Lietotāja saskarnes sadaļas mobilo ierīču pārvaldībai: *Users* > *Devices* un *User settings* > *Devices*.
- Multivides tips problēmu paziņojumu sūtīšanai no Zabbix servera uz mobilo lietotni: Push notification.
- Komponents, kas ļauj jūsu Zabbix serverim sazināties ar mobilo lietotni: Bridge adapter.
- Zabbix servera konfigurācijas parametri: BridgeAdapterURL, EnableMobileDevices.

Detalizētu informāciju par Zabbix iestatīšanu lietošanai ar mobilo lietotni skatiet sadaļā Mobile app.



MCP serveris Jaunais Zabbix MCP serveris savieno MI aģentus ar Zabbix, izmantojot Model Context Protocol (MCP). Tagad varat uzdot MI asistentam jautājumus par savu uzraudzību dabiskā valodā, piemēram:

- *Kuras problēmas datubāzu serveros vēl nav apstiprinātas?*
- *Kāpēc web-01 ziņo par problēmām kopš šī rīta?*
- *Ievietot visus Linux serveru grupas hostus apkopē šonakt no plkst. 22:00 līdz 02:00.*



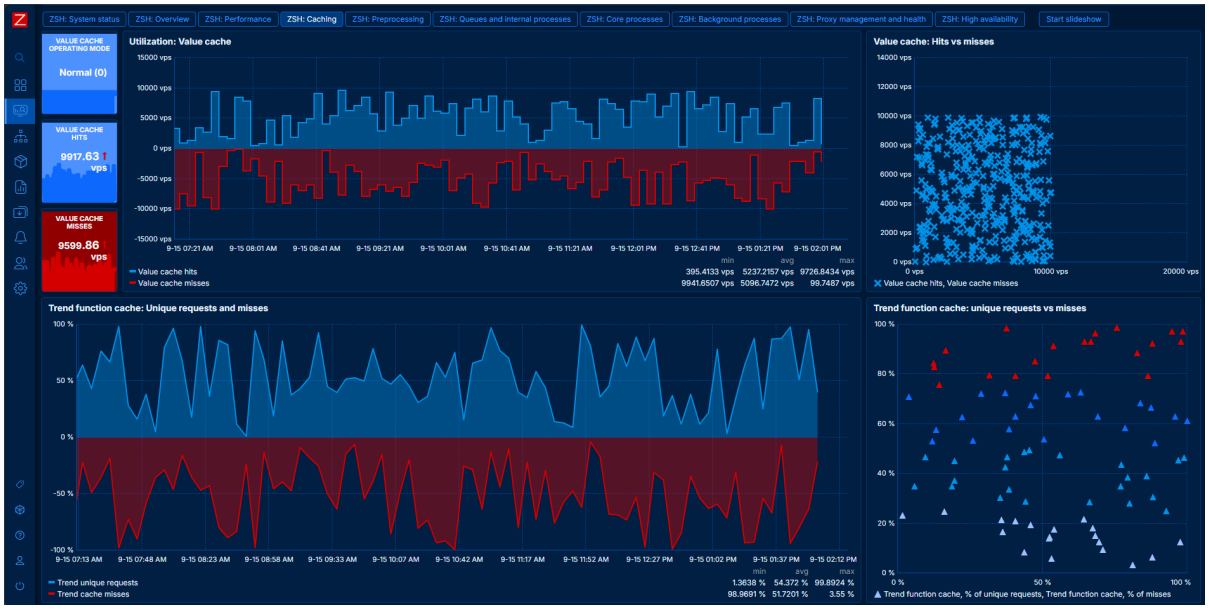
Aģents izveido savienojumu, izmantojot Zabbix API pilnvaru, tāpēc tas var piekļūt tikai tam, kam ir atļauts piekļūt pilnvaras īpašniekam, un visas izmaiņas tiek reģistrētas audita žurnālā pie šī lietotāja. Pieejamo darbību sarakstu skatiet sadaļā Rīki un resursi.

Trigeru uzturēšana pēc pieprasījuma Uzturēšanas periodus tagad var definēt trigeriem vai notikumu nosaukumiem.

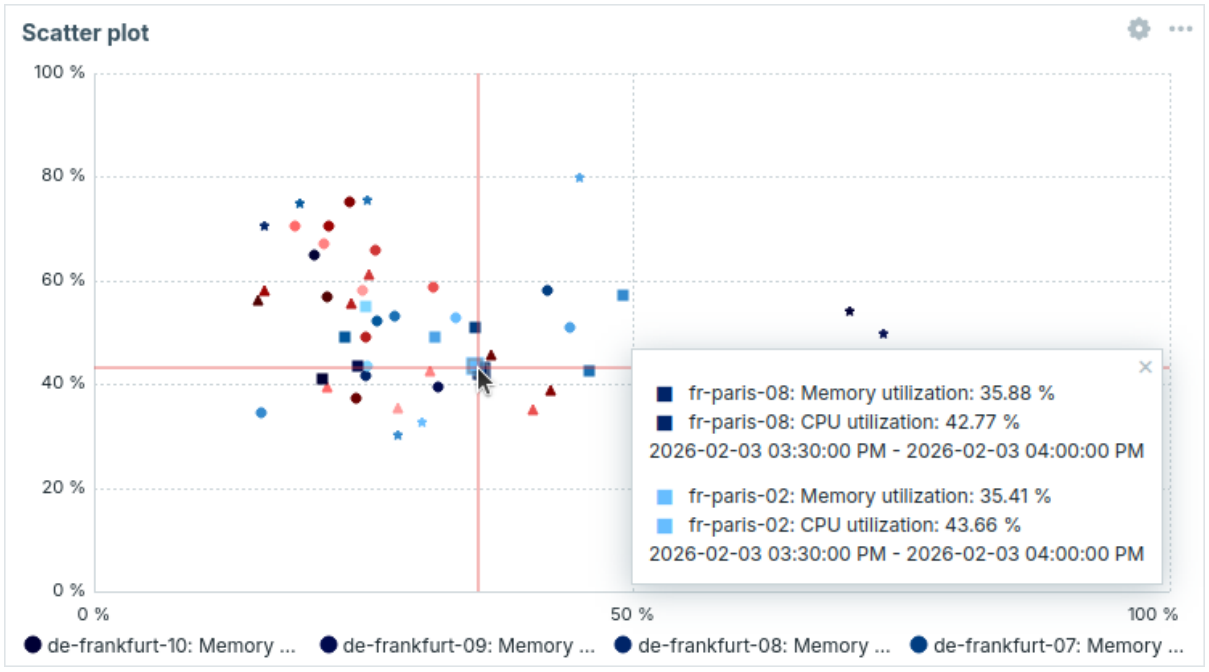
Uzturēšanas periodu var izveidot arī pēc pieprasījuma notikumu izvēlnē. Uzturēšanu pēc pieprasījuma var sākt hostam, trigerim, notikuma nosaukumam vai tagiem.

Vēl viens veids, kā izveidot uzturēšanu trigerim, ir izmantot problēmas atjaunināšanas veidlapu.

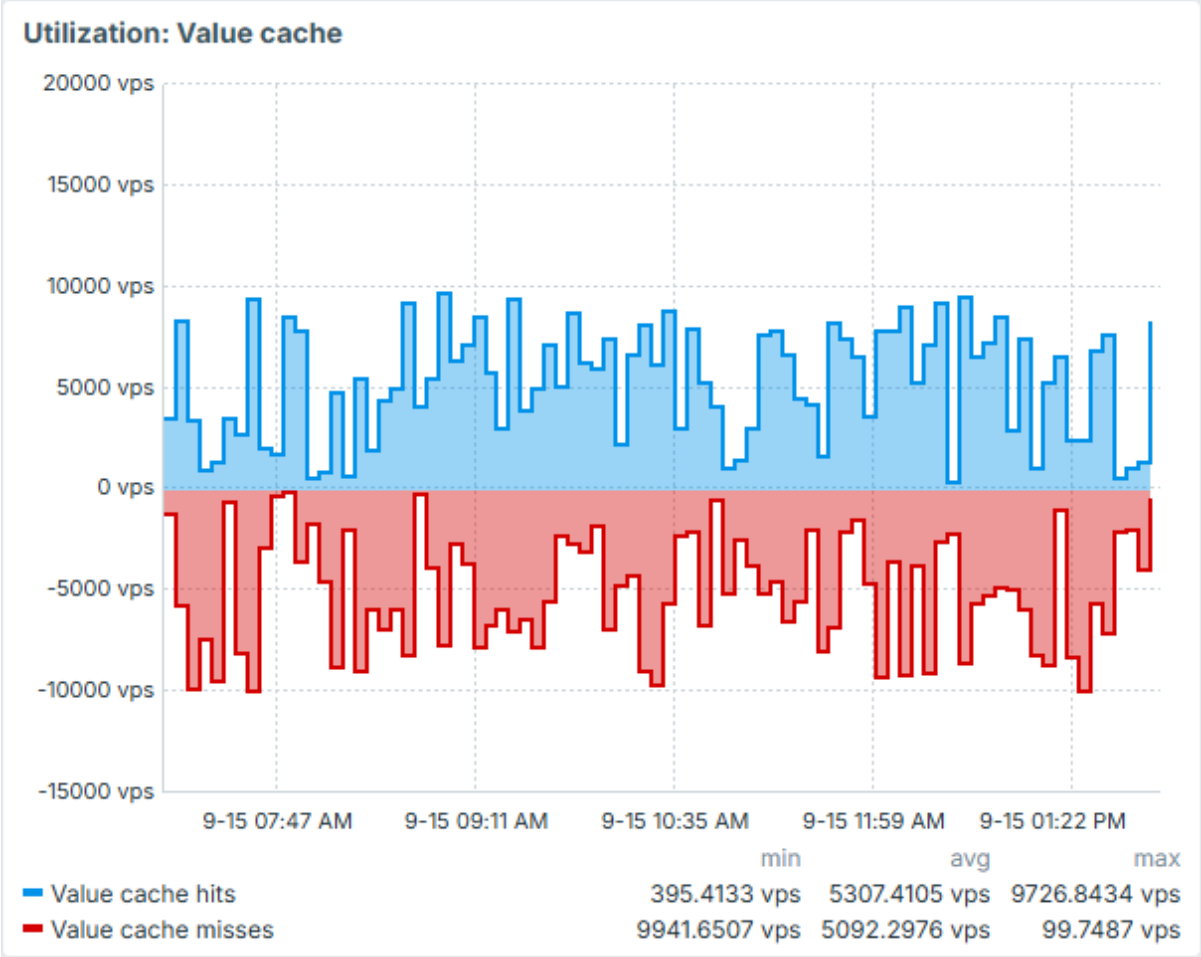
Tumši zilā tēma Zabbix lietotāja saskarne tagad piedāvā jaunu tumši zilu tēmu.
Tā ir pieejama lietotāja profilā un kā noklusējuma tēma:



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 ass inversija grafika sīkrīkā Grafiki grafika sīkrīkā tagad var tikt attēloti ar apgrieztām Y ass vērtībām. Jauns iestatījums *Invert values* ļauj reizināt Y ass vērtības ar −1, nemainot sākotnējos datus.



Skatiet arī: Grafika sīkrīka uzlabojumi.

JSON datu tips Zabbix tagad atbalsta JSON kā datu tipu vienumu vērtībām.

Iepriekš JSON vērtības tika iegūtas ar teksta vienumiem un glabātas kā virknes ar 64 KB ierobežojumu. Tagad Zabbix var glabāt JSON vērtības natively ar 128 MiB ierobežojumu, kā arī noraidīt nederīgas JSON vērtības (piem., ar neiekļautiem pēdiņās atslēgu nosaukumiem, liekām komatiem, nesakritīgām iekavām).

JSON datu tips ir atbalstīts visiem vienumu un vienumu prototipu tipiem, izņemot *Calculated*, un ir pieejams reāllaika datu eksportā un savienotājos. JSON vērtības var glabāt visās atbalstītajās datubāzēs un Elasticsearch. Ja izmantojat TimescaleDB, skatiet jaunināšanas piezīmes.

Ņemiet vērā, ka vienumus ar JSON datu tipu nevar izmantot trigeros; tomēr jūs varat izvilkt JSON laukus ar atkarīgajiem vienumiem, kuriem ir ne-JSON datu tips, un izmantot tos trigeros.

Vienumi, kas atgriež JSON virkni (`net.if.discovery`, `vfs.file.get` u.c.), joprojām ir teksta vienumi; tomēr, ja nepieciešams, jūs varat tos mainīt uz JSON.

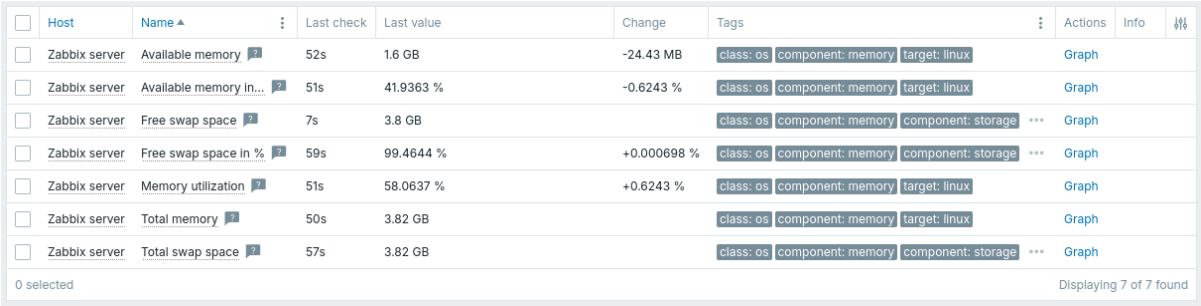
Plašāku informāciju un JSON datu ierobežojumus skatiet vienuma konfigurācijā.

Pielāgots tabulu izkārtojums Vairākās lietotāja saskarnes tabulās tagad varat paslēpt/parādīt, pārkārtot un mainīt kolonnu platumu. Ir arī iespējams pievienot *Pielāgots teksts* kolonnu ar pielāgotu tekstu, makro vērtībām vai abiem.

Pielāgotie izkārtojumi ir pieejami šādās vietās:

- *Monitoring* > *Hosts*, *Latest data* un *Problems*
- *Data collection* > *Templates* un *Hosts*

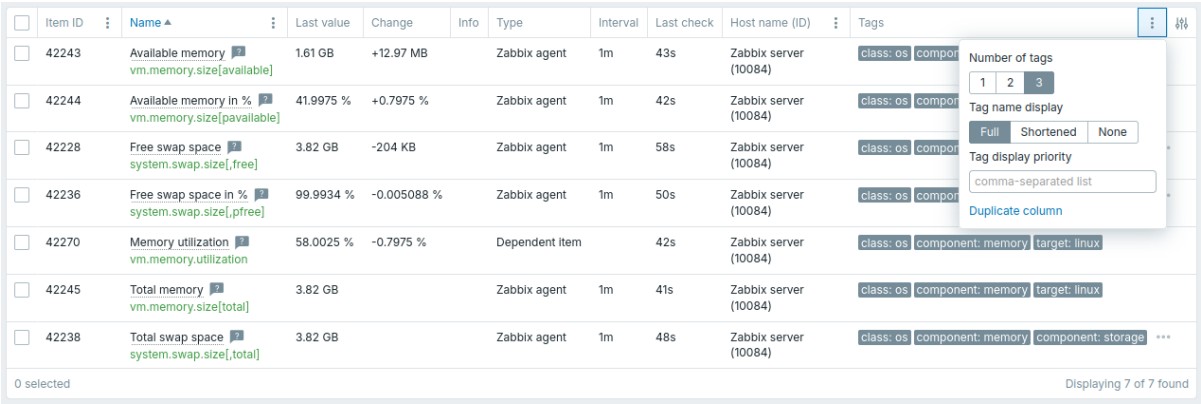
Piemēram, noklusējuma tabulas izkārtojums sadaļā *Monitoring* > *Latest data*:



Host	Name	Last check	Last value	Change	Tags	Actions	Info
Zabbix server	Available memory	52s	1.6 GB	-24.43 MB	class: os, component: memory, target: linux	Graph	
Zabbix server	Available memory in...	51s	41.9363 %	-0.6243 %	class: os, component: memory, target: linux	Graph	
Zabbix server	Free swap space	7s	3.8 GB		class: os, component: memory, component: storage	Graph	
Zabbix server	Free swap space in %	59s	99.4644 %	+0.000698 %	class: os, component: memory, component: storage	Graph	
Zabbix server	Memory utilization	51s	58.0637 %	+0.6243 %	class: os, component: memory, target: linux	Graph	
Zabbix server	Total memory	50s	3.82 GB		class: os, component: memory, target: linux	Graph	
Zabbix server	Total swap space	57s	3.82 GB		class: os, component: memory, component: storage	Graph	

0 selected — Displaying 7 of 7 found

Pielāgota izkārtojuma piemērs:



Item ID	Name	Last value	Change	Info	Type	Interval	Last check	Host name (ID)	Tags
42243	Available memory vm.memory.size[available]	1.61 GB	+12.97 MB		Zabbix agent	1m	43s	Zabbix server (10084)	class: os, compon
42244	Available memory in % vm.memory.size[pavailable]	41.9975 %	+0.7975 %		Zabbix agent	1m	42s	Zabbix server (10084)	class: os, compon
42228	Free swap space system.swap.size[,free]	3.82 GB	-204 KB		Zabbix agent	1m	58s	Zabbix server (10084)	class: os, compon
42236	Free swap space in % system.swap.size[,pfree]	99.9934 %	-0.005088 %		Zabbix agent	1m	50s	Zabbix server (10084)	class: os, compon
42270	Memory utilization vm.memory.utilization	58.0025 %	-0.7975 %		Dependent item		42s	Zabbix server (10084)	class: os, component: memory, target: linux
42245	Total memory vm.memory.size[total]	3.82 GB			Zabbix agent	1m	41s	Zabbix server (10084)	class: os, component: memory, target: linux
42238	Total swap space system.swap.size[,total]	3.82 GB			Zabbix agent	1m	48s	Zabbix server (10084)	class: os, component: memory, component: storage

Number of tags: 1 | 2 | 3
Tag name display: Full | Shortened | None
Tag display priority: comma-separated list
Duplicate column

0 selected — Displaying 7 of 7 found

Dažām kolonnām ir arī poga *Options*, kas nodrošina papildu vadīklas tam, kā šī kolonna attēlo datus. Tāpēc daži filtra parametri ir pārvietoti no filtra cilnes uz kolonnu opcijām. Ja izmantojat cilnes izlases filtriem, lūdzu, skatiet jaunināšanas piezīmes.

Globālās regulārās izteiksmes Globālajām regulārajām izteiksmēm ir pievienoti vairāki lietojamības uzlabojumi:

- Globālās regulārās izteiksmes konfigurācijas veidlapa tagad tiek atvērta modālā logā:
- Izteiksmju tipi ir pārdēvēti.
- Veidlapā tagad ir iekļauts lauks *Apraksts*.
- Testēšana vairs nenotiek atsevišķā cilnē.
- Globālo regulāro izteiksmju sarakstu tagad var filtrēt pēc nosaukuma un apraksta.
- Globālās regulārās izteiksmes tagad var eksportēt un importēt.

Veidnes

Ir pievienotas jaunas veidnes, un daudzas veidnes ir atjauninātas.

Veidnes, kas izveido savienojumu ar HTTPS galapunktiem, pēc noklusējuma tagad pārbauda galapunkta TLS sertifikātu. Tas attiecas uz visām mākoņpakalpojumu un API uzraudzības veidnēm, piemēram, *Claude API by HTTP*, *Cloudflare by HTTP*, *Kubernetes Cluster by HTTP* u. c. Pirms jebkuru uz HTTP balstītu veidņu jaunināšanas skatiet jaunināšanas piezīmes.

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

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

Jaunas veidnes MI lietojuma uzraudzība

- Claude API by HTTP — veidne Claude API lietojuma un izmaksu uzraudzībai.
- OpenAI Platform by HTTP — veidne OpenAI izstrādātāju platformas uzraudzībai.

Mākonis

- AWS by HTTP veidņu kopa papildināta ar veidni *AWS Backup Vault by HTTP*.
- Azure by HTTP veidņu kopa papildināta ar veidnēm *Azure Sentinel by HTTP* un *Azure Container Apps by HTTP*.
- GCP by HTTP veidņu kopa papildināta ar veidnēm *GCP Cloud Run Service by HTTP*, *GCP Cloud Storage Bucket by HTTP*, *GCP Application Load Balancer by HTTP* un *GCP Cost Monitoring by HTTP*.
- Oracle Cloud by HTTP veidņu kopa papildināta ar veidnēm *Oracle Cloud Load Balancer by HTTP* un *Oracle Cloud Costs by HTTP*.

Virtualizācija un konteineri

- Kubernetes Cluster by HTTP — veidne Kubernetes klasteru uzraudzībai, kurā mezgli tiek automātiski atklāti un uzraudzīti, izmantojot īpašus hostus; veidne aizstāj iepriekšējās Kubernetes veidnes.
- 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.
- Podman by Zabbix agent, by Zabbix agent active un by HTTP — veidnes Podman konteineru un pakalpojumu darbspējas uzraudzībai, izmantojot Podman API.

Ņemiet vērā, ka Podman API neatbalsta autorizācijas metodes.

Datu bāzes

- MariaDB by ODBC — veidne MariaDB datu bāzu uzraudzībai, izmantojot ODBC.
- MariaDB by Zabbix agent un MariaDB by Zabbix agent active — veidnes MariaDB datu bāzu uzraudzībai, izmantojot Zabbix aģentu un aktīvo Zabbix aģentu.
- Percona by ODBC — veidne Percona datu bāzu uzraudzībai, izmantojot ODBC.
- Percona by Zabbix agent un Percona by Zabbix agent active — veidnes Percona datu bāzu uzraudzībai, izmantojot Zabbix aģentu un aktīvo Zabbix aģentu.

Tīkli, drošība un telekomunikācijas

- Aruba CX 8300s by SNMP — veidne Aruba CX 8300 komutatoru sērijas uzraudzībai, izmantojot SNMP.
- Ciena 3906 by SNMP — veidne Ciena 3906 ierīču, tostarp failu sistēmas un CPU noslodzes, uzraudzībai.
- Cisco Secure Firewall Threat Defense by HTTP — veidne Cisco Secure Firewall Threat Defense ierīču uzraudzībai, izmantojot REST API.
- Cradlepoint NCM v2 by HTTP — veidņu kopa Cradlepoint NCM v2 un tā ierīču uzraudzībai, izmantojot HTTP.
- Huawei AR600 by SNMP — veidne Huawei AR600 maršrutētāju uzraudzībai, tostarp CoS un NQA metrikām.
- 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.

Lietojumprogrammas un pakalpojumi

- Domain RDAP by HTTP — veidne domēnu reģistrācijas datu uzraudzībai, izmantojot RDAP protokolu.
- Envoy Proxy by OTLP — veidne Envoy Proxy OpenTelemetry trasējumu un žurnālu apkopošanai, izmantojot *Telemetry query* vienumus.
- Generic OpenTelemetry by OTLP — veidne OpenTelemetry trasējumu, metrikas datu un žurnālu apkopošanai, izmantojot *Telemetry query* vienumus.
- GLPI by HTTP — veidne GLPI uzraudzībai, izmantojot HTTP, REST API v2 un OAuth2 autentifikāciju.
- OpenRouter traces by OTLP — veidne OpenRouter trasējumu datu apkopošanai.
- Proxmox VE by OTLP — veidne Proxmox VE mezgla uzraudzībai, izmantojot tā OpenTelemetry metrikas straumi un *Telemetry query* vienumus.

Atjauninātas veidnes

- Uzlabots ierīces darbspējas laika trigeris SNMP veidnēs, lai apstrādātu SNMP darbspējas laika skaitītāja atiestatīšanu un novērstu kļūdainus restartēšanas brīdinājumus.

Operētājsistēmas

- AIX, FreeBSD, HP-UX, Linux, macOS, OpenBSD, Solaris un Windows veidnēs failu sistēmu uzraudzībā tagad tiek izmantoti jaunie `vfs.fs.get` parametri (skatiet Vieglā failu sistēmu atklāšana). Detalizētu informāciju skatiet sadaļā Veidņu izmaiņas.
- Linux by Zabbix agent un Linux by Zabbix agent active ir atjauninātas ar uzlabotām informācijas paneļiem un uzlabotu tīkla saskarņu un bloku ierīču uzraudzību.
- Linux by Prom un Linux by SNMP ir atjauninātas ar uzlabotām informācijas paneļiem.
- Windows by Zabbix agent un Windows by Zabbix agent active veidnes ir papildinātas ar jauniem rādītājiem, atklāšanas noteikumiem, prototipiem, lietotāja makro un informācijas paneļa logrīkiem, tostarp Windows ieplānoto uzdevumu, Windows ugunsmūra profilu un ar BitLocker šifrējamo sējumu atbalstu.

MI lietojuma uzraudzība

- GitHub organization by HTTP ir atjaunināta, iekļaujot vienumus Microsoft Copilot uzraudzībai.
- Microsoft 365 reports by HTTP ir atjaunināta, iekļaujot vienumus Microsoft Copilot uzraudzībai.

Mākonis

- AWS by HTTP veidņu kopas *AWS Cost Explorer by HTTP* un Azure by HTTP veidņu kopas *Azure Cost Management by HTTP* ir atjauninātas ar uzlabotiem informācijas paneļiem.

Virtualizācija

- Proxmox VE by HTTP ir atjaunināta ar ligzdotas LLD funkcionalitāti. Turklāt vienumu, kuros tiek rādītas procentuālās vērtības, mērvienību formāts ir pielāgots skaidrības uzlabošanai.
- VMware un VMware FQDN veidņu kopās VMware trauksmju atklāšanas vienumu un trigeru prototipu nosaukumi ir atjaunināti, iekļaujot trauksmes entītijas nosaukumu.

Datu bāzes

- 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). Turklāt MySQL by Zabbix agent, MySQL by Zabbix agent active un MySQL by ODBC ir atjauninātas ar jauniem rādītājiem, uzlabotu informācijas paneli un paplašinātiem tabulu atklāšanas noteikumiem (kā arī replikām *MySQL by ODBC* veidnē).
- PostgreSQL by ODBC ir atjaunināta ar jauniem rādītājiem, atklāšanas noteikumiem un informācijas paneļiem, kā arī vispārīgiem veiktspējas un novērojamības uzlabojumiem.
- PostgreSQL by Zabbix agent un PostgreSQL by Zabbix agent active ir atjauninātas ar papildu rādītājiem, atklāšanas noteikumiem, vienumu un trigeru prototipiem, kā arī uzlabotām uzraudzības iespējām.
- Redis by Zabbix agent 2 ir atjaunināta ar autentifikāciju, izmantojot lietotāja un paroles makro, kā arī uzlabotiem vienumu aprakstiem.

Tīkli, drošība un telekomunikācijas

- VeloCloud SD-WAN by HTTP, kas iepriekš tika dēvēta par *VMWare SD-WAN VeloCloud by HTTP*, ir pārdēvēta un atjaunināta, lai atbilstu jaunākajai VeloCloud SD-WAN platformai.

Krātuve un dublēšana

- Ceph by Zabbix agent 2 tagad atbalsta vietējā savienojuma režīmu. Detalizētu informāciju skatiet sadaļā Ceph spraudnis.
- Veeam Backup and Replication by HTTP un Veeam Backup Enterprise Manager by HTTP ir atjauninātas ar jaunām uzraudzības iespējām un Veeam Backup & Replication v13.1.1.18 atbalstu, tostarp REST API 1.3-rev2 versiju un atjauninātu licences derīguma termiņa uzraudzību.

Failu sistēmu uzraudzība OS veidnēs AIX, FreeBSD, HP-UX, Linux, macOS, OpenBSD, Solaris un Windows veidnēs (tostarp Linux un Windows veidņu aktīvās pārbaudes variantos) failu sistēmu uzraudzībā tagad tiek izmantoti jaunie `vfs.fs.get` parametri:

- vienums *Get filesystems* (`vfs.fs.get`) ir noņemts;
- atklāšanas noteikums *Mounted filesystem discovery* tagad ir Zabbix aģenta vienums ar atslēgu `vfs.fs.get[short]` (atjaunināšanas intervāls: 1h), nevis atkarīgs atklāšanas noteikums;
- vienuma prototips *FS [{#FSNAME}]: Get data* (Windows veidnēs *FS [{#FSLABEL}({#FSNAME})]: Get data*) tagad ir Zabbix aģenta vienums ar atslēgu `vfs.fs.get[full,{#FSNAME}]`, kas apkopo statistiku tikai vienai failu sistēmai. Pārējie failu sistēmu vienumu prototipi to izmanto kā galveno vienumu.

Tādējādi statistika tiek apkopota, izmantojot atsevišķu aģenta pārbaudi katrai failu sistēmai, un jaunās piemontētās failu sistēmas tiek atklātas atklāšanas atjaunināšanas intervāla laikā (pēc noklusējuma — 1h).

Failu sistēmu uzraudzībai šajās veidnēs ir nepieciešams Zabbix aģents 8.0 vai jaunāks.

Multivides veidi

Ir pievienots šāds multivides veids:

- IBM Maximo Service Request webhook multivides veids nodrošina Zabbix sasaisti ar IBM Maximo.

Ir atjaunināti šādi multivides veidi:

- GLPI webhook multivides veids ir atjaunināts, lai atbalstītu RESTful API v2 ar OAuth2 autorizāciju (iespējota pēc noklusējuma), pēc izvēles nodrošinātu mantotā REST API v1 atbalstu, izmantojot konfigurācijas karodziņu, izveidotu pieteikumus problēmu notikumiem un ļautu lietotājam konfigurēt steidzamības kartējumu.
- MS Teams Workflow webhook multivides veids tagad ir ieteicamā integrācija ar Microsoft Teams.

MS Teams webhook multivides veids, kas balstīts uz Office 365 Connector (Incoming Webhook) funkcionalitāti, kuru Microsoft pārtrauca 2026. gada 22. maijā, vairs nedarbojas. Esošās MS Teams webhook konfigurācijas ir jāmigrē uz MS Teams Workflow webhook multivides veidu.

Turklāt multivides veidi, kas izveido savienojumu ar HTTPS galapunktiem, tagad pēc noklusējuma pārbauda galapunkta TLS sertifikātu. Tas attiecas uz visiem webhook multivides veidiem, piemēram, *Slack*, *Jira*, *Telegram* u. c. Pirms webhook multivides veidu jauninašanas skatiet jauninašanas piezīmes.

Vienumi

Noklusējuma vērtība trapper vienuma atļautajiem hosts Lauks *Atļautie hosts* Zabbix trapper vienumiem un HTTP aģents vienumiem ar iespējotu *Enable trapping* tagad pēc noklusējuma izmanto lietotāja makro `{$TRAPPER.ALLOWED_HOSTS}`. Ja šī vērtība ir tukša, ienākošie savienojumi no visiem hosts tiek noraidīti. Lai atļautu datu iesniegšanu, norādiet vienu vai vairākas IP adreses vai DNS nosaukumus.

Trend funkcijas kešatmiņas izmantošanas uzraudzība Parametrs `paccessed` ir pievienots iekšējam vienumam `zabbix[tcache,cache,<parameter>]`. Tas atgriež trend funkcijas kešatmiņas ierakstu procentuālo daļu, kuriem piekļūts pēdējo 24 stundu laikā.

Novirzīšana web.page.* vienumos Zabbix aģenta un Zabbix aģenta 2 web.page.get, web.page.perf un web.page.regexp vienumi tagad var sekot novirzīšanām. Maksimālais novirzīšanu skaits tiek iestatīts jaunajā neobligātajā `redirect_limit` parametrā; pēc noklusējuma novirzīšanām netiek sekots.

Spraudņi

Ceph spraudnis Zabbix aģents 2 Ceph spraudnis tagad darbojas divos režīmos:

- **native** - Šis režīms izmanto go-ceph bibliotēku, lai tieši sazinātos ar Ceph klasteri, izmantojot vietējo Ceph API (msgr2 protokolu). Šis ir ieteicamais režīms mūsdienīgām Ceph instalācijām, taču tas tiek atbalstīts tikai **Linux** vidē un, sākot ar Ceph 16.
- **restful** (novecojis) - Šis režīms saziņai izmanto Ceph RESTful API. Tas ir noklusējuma režīms atpakaļsavietojamībai, taču tas nedarbosies ar Ceph versiju 20 (Tentacle) vai jaunāku, jo ir noņemts `mgr/restful` modulis.

Svarīgi: Uzraugot Ceph 20 (Tentacle) un jaunākas versijas ar Ceph spraudni Zabbix aģentam 2, izmantojiet native režīmu, jo restful režīms vairs netiek atbalstīts.

Izmantojamais režīms tiek noteikts pēc mode parametra vērtības (native/restful):

- `Plugins.Ceph.Default.Mode=native` - iestata spraudnim native režīmu
- `Plugins.Ceph.Sessions.<SessionName>.Mode=native` - iestata nosauktajai sesijai native režīmu

Ņemiet vērā, ka lietotāja akreditācijas datu kopums katram režīmam atšķiras un tie nav savstarpēji saderīgi. Parametrs `Plugins.Ceph.InsecureSkipVerify` native režīmā tiek ignorēts, jo savienojuma drošību Ceph klastera pusē nosaka msgr2 protokols (pēc noklusējuma drošs).

Ņemiet vērā, ka Ceph spraudnis Zabbix aģentam 2 tagad ir ielādējams spraudnis un tam 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 tradicionālai MongoDB URI parsēšanai, tostarp atbalsts gan `mongodb://`, gan `mongodb+srv://` shēmām. - Automātiska MongoDB replica set mezglu noteikšana, uzlabojot iespējas uzraudzīt MongoDB klasterus. - x509 autentifikācijas atbalsts, kas nodrošina 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 aizsargātus socket savienojumus, tādējādi uzlabojot attālinātās uzraudzības drošību.

Redis spraudnis — TLS atbalsts un palaišanas laika validācija Zabbix aģents 2 Redis spraudnim ir pievienots TLS atbalsts.

Ir ieviesta spraudņa TLS konfigurācijas validācija palaišanas laikā, kā arī uzlaboti validācijas/kļūdu ziņojumi. Nederīga konfigurācijas loģika (piemēram, savienojuma tipa `verify_full` izmantošana, nenorādot `TLSCAFile`) var neļaut Zabbix aģentam 2 startēties.

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.

Atklāšana un automātiskā reģistrācija

Starpniekserveru grupas nosacījums Starpniekserveru grupas nosacījums tagad tiek atbalstīts atklāšanas un automātiskās reģistrācijas darbībās.

Zema līmeņa atklāšana

Izvēles rūtiņa "Konvertēt uz JSON" atklāšanas veidlapās Zema līmeņa atklāšanas kārtulas un atklāšanas prototipa veidlapai tika pievienota jauna izvēles rūtiņa *Konvertēt uz JSON*, kas tiek parādīta, ja nolaižamajā sarakstā *Tips* ir atlasīts "HTTP aģents". Šī opcija ļauj atzīmēt izgūtos 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.

Bloka ierīču atklāšana Linux un AIX Jaunais `vfs.dev.get` vienuma atslēga tika pievienota bloka ierīču atklāšanai Linux un AIX. To var izmantot zemlīmeņa atklāšanā bloka ierīcēm. `vfs.dev.discovery` vienuma atslēga joprojām ir pieejama savietojamībai.

Tīkla saskarņu atklāšana operētājsistēmā Linux Zabbix aģentam un Zabbix aģentam 2 operētājsistēmā Linux ir pievienots jaunais `net.if.get` vienuma atslēga. To var izmantot zema līmeņa tīkla saskarņu atklāšanā, lai atgrieztu tīkla saskarņu sarakstu ar detalizētu informāciju. `net.if.discovery` vienuma atslēga joprojām ir pieejama savietojamībai.

Vieglāka failu sistēmu atklāšana Zabbix aģenta un Zabbix agent 2 `vfs.fs.get` vienums tagad atbalsta divus neobligātus parametrus, kas padara failu sistēmu atklāšanu vieglāku un uzticamāku:

- **mode** - `short` režīmā vienums atgriež tikai piemontēto failu sistēmu sarakstu ar to pamatīpašībām, neiekļaujot diska vietas un inode statistiku. Tā kā pašas failu sistēmas netiek vaicātas, atklāšana neizraisa automātisku piemontēšanu un to neietekmē nereaģējošas tīkla failu sistēmas. `full` režīms (noklusējums) atgriež tos pašus datus kā iepriekš.
- **mountpoint** - atgriež datus tikai par norādīto piemontēšanas punktu, lai statistiku varētu apkopot atsevišķi par katru atklāto failu sistēmu (piemēram, izmantojot `vfs.fs.get[full,{#FSNAME}]`).

OS veidnes tagad izmanto šo pieeju; skatiet Veidņu izmaiņas.

Papildinformāciju skatiet sadaļās vfs.fs.get un Piemontēto failu sistēmu atklāšana.

Datu bāzes

Atbalsts vairākiem PostgreSQL datubāzes hosts Zabbix tagad atbalsta vairāku PostgreSQL datubāzes hostu norādīšanu kā ar komatiem atdalītu `host:port` vērtību sarakstu, izmantojot PostgreSQL savienojuma atteices pārslēgšanas atbalstu. Hosts tiek mēģināti secīgi, līdz tiek izveidots lasīšanas-rakstīšanas savienojums. Šī darbība ir pieejama lietotāja saskarnes instalēšanas laikā laukā *Database host* un `DBHost` `zabbix_server.conf` un `zabbix_proxy.conf`.

Optimizēta Zabbix servera palaišana Zabbix serveris tagad startē ievērojami ātrāk. Tas to panāk, vienlaikus ielādējot vēstures datus no datubāzes vairākiem vienumiem, nevis pa vienam. Šis uzlabojums attiecas uz visiem datubāzu tipiem, ko atbalsta Zabbix serveris.

Optimizēta Elasticsearch veiktspēja Zabbix serveris tagad darbojas labāk ar Elasticsearch. Serveris efektīvāk izgūst datus, neizmantojot Elasticsearch *scroll* funkciju, un atkārtoti izmanto esošos savienojumus ar Elasticsearch, nevis atver jaunu savienojumu katram pieprasījumam. Serveris arī vairs neiestrēgst, vienlaikus nosūtot vairākus pieprasījumus uz Elasticsearch (izmantojot cURL *multi-handle*).

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 ātruma ierobežošana vēstures kešatmiņas atkopšanas laikā Starpniekservera ātruma ierobežošanas loģika ir pilnveidota, lai uzlabotu servera stabilitāti vēstures kešatmiņas atkopšanas laikā. Kad vēstures kešatmiņas lietojums sasniedz ātruma ierobežošanas slieksni, serveris turpina pārtraukt starpniekservera datu pieņemšanu, kā iepriekš. Kad kešatmiņas lietojums samazinās līdz 60%, serveris sāk apstrādāt ātruma ierobežošanas sarakstu, taču joprojām var noraidīt starpniekservera augšupielādes, kas satur ļoti lielas paketes (aptuveni vairāk nekā 10 tūkst. ierakstu), līdz kešatmiņas noslodze samazinās vēl vairāk. Šīs izmaiņas samazina atkārtotas kešatmiņas pārslodzes risku, kamēr serveris atkopjas.

Palielināts zabbix_get un zabbix_js maksimālais taimauts 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 — trigeru tagi, kas mantoti no atkopšanas notikumiem Atkopšanas notikumi, kas izveidoti pēc manuālas aizvēršanas, papildus vienuma un hosta tagiem pārmanto arī trigeru tagus. Šie tagi ir pieejami notikuma tagu masīvā un ir pieejami paziņojumu makro, 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.

Uzlabotas uz skaitu balstītas trigeru funkcijas Uz skaitu balstītu funkciju (count(), item_count(), changecount() u. c.) darbība trigeru izteiksmēs ir uzlabota, ļaujot ātrāk apstrādāt vairāk vērtību.

Izpildes laika vadības mērķu atjauninājumi Pieejamo izpildes laika vadības mērķu saraksts Zabbix servera un starpniekservera palīdzības ziņojumos un manpages ir atjaunināts, lai atbilstu atbalstītajiem procesu tipiem.

Ir pievienots `supervisor` mērķis, savukārt neatbalstītie `discovery worker` un `preprocessing worker` mērķi ir noņemti.

Drošība

Starpniekservera un servera piekļuves kontrole Jaunā cilne *Starpniekserveru piekļuves saraksts* lietotāju grupas konfigurācijā ļauj piešķirt lietotāju grupai piekļuvi starpniekserveriem — atsevišķi vai izmantojot starpniekserveru grupas.

Jaunā opcija *Atlasīt "Serveri" uzraudzībai un atklāšanai* lietotāja lomas konfigurācijā ļauj liegt lietotājiem izveidot hostus, kurus uzrauga serveris, un tīkla atklāšanas noteikumus, ko izpilda serveris, kā arī liegt vienumu testēšanu ar serveri.

Jauni parametri `AllowKeyRegexp` un `DenyKeyRegexp` Zabbix aģenta un Zabbix agent 2 konfigurācijā Zabbix aģenta konfigurācijai un Zabbix agent 2 konfigurācijai ir pievienoti jaunie parametri `AllowKeyRegexp` un `DenyKeyRegexp`. Šie parametri ļauj definēt uz regulārajām izteiksmēm balstītus modeļus precīzākai un elastīgākai atslēgu un to parametru atbilstības noteikšanai.

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.

Post-login URL per user role It is now possible to enforce a specific URL after login based on the user role. This is handy for enforcing a specific page after login for provisioned users.

For more details, see User profile settings in user role configuration.

HashiCorp vault autentifikācija ar AppRole metodi HashiCorp vault autentifikācija tagad atbalsta AppRole metodi:

- Servera/starpniekservera konfigurācijā, izmantojot jaunos `VaultAppRoleID` un `VaultAppSecretID` parametrus (skatiet piemēru).
- lietotāja saskarne konfigurācijā, norādot autentifikācijas lomas ID un slepenā ID vērtības.

Logdaļas

Izmaiņas pulksteņa logrīka opcijā "Laika josla" Opcija *Laika josla* pulksteņa logrīka konfigurācijas veidlapā ir pārvietota no sadaļas *Papildu konfigurācija* uz logrīka konfigurācijas galveno sadaļu un pārdēvēta par *Ignorēt laika joslu*.

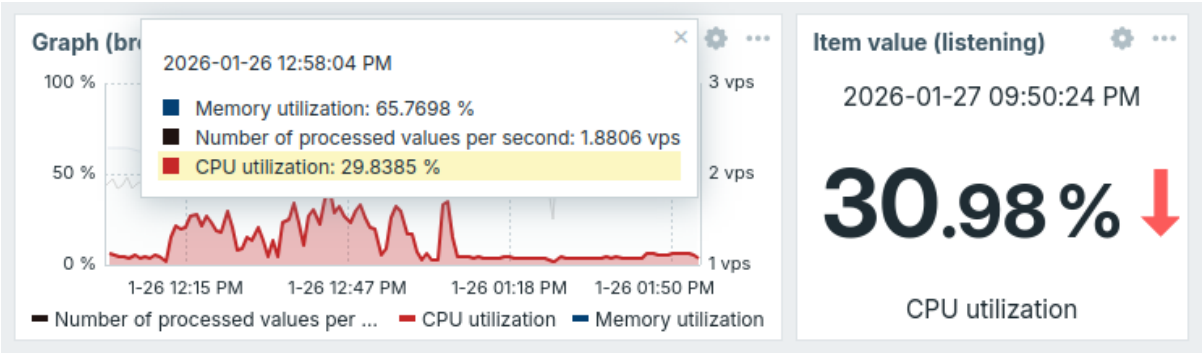
Šī opcija ir pieejama laika tipiem *Vietējais laiks* un *Servera laiks*.

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.

Vienumu apkopošana logrīkā "Populārākie vienumi" Logrīks Populārākie vienumi tagad var grupēt vienumu modeļus vienā apkopotā kolonnā vai rindā, tādējādi atvieglojot apvienotu metriku attēlošanu (piemēram, kopējo ienākošo datplūsmu visās hosts tīkla saskarnēs). Konfigurējot vienumu kolonnas, tika pievienotas trīs jaunas opcijas: *Apkopot kolonnas*, *Kolonnu apkopošanas funkcija* un *Apvienotās kolonnas nosaukums*.

Grafika logrīka uzlabojumi Grafika logrīkam ir pievienoti vairāki lietojamības uzlabojumi:

- Vienumi grafika datu rīka padomā, kas tiek parādīts, novietojot kursoru virs grafika, tiek kārtoti dilstošā secībā pēc vērtības.
- Novietojot kursoru virs vienuma rīka padomā, tiek izcelts attiecīgais grafiks, bet pārējie tiek aptumšoti.
- Atlasot vienumu rīka padomā, tā dati tiek pārraidīti citiem logrīkiem, kas atbalsta to saņemšanu.
- Ir noņemts ierobežojums rīka padomā parādāmo vienumu skaitam.
- Jauns iestatījums *Hosts nosaukumi etiķetēs* ļauj izvēlēties, vai hosts nosaukumi tiek parādīti rīka padā un grafika leģendā.
- Jauns iestatījums *Apgriezt vērtības* 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.

Lietotāja saskarne

Importējamas informācijas paneļa veidnes Informācijas paneļu repozitorijā ir pieejamas jaunas importējamas informācijas paneļa veidnes:

- Multicloud Cost Dashboard - informācijas panelis mākoņpakalpojumu izmaksu pārraudzībai AWS, Azure, Google Cloud un Oracle Cloud vidē.
- Multicloud Virtual Machines Dashboard - informācijas panelis AWS, Google Cloud, Oracle Cloud un Azure vidē darbojošos virtuālo mašīnu pārraudzībai.

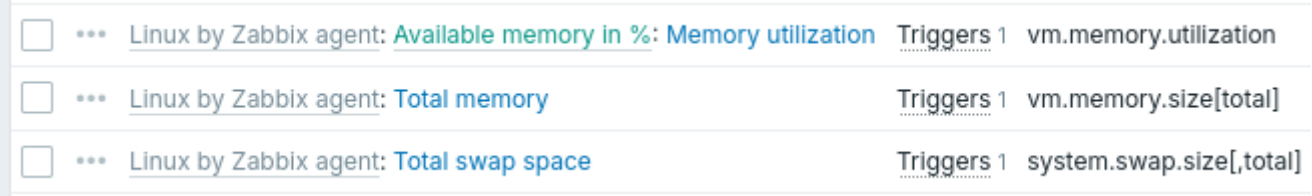
Šie informācijas paneļi ir paredzēti darbam ar esošajām Zabbix integrācijām, apvienojot datus no vairākiem hostiem un pārraudzības mērķiem vienotā globālā informācijas panelī.

Informācijas paneļa eksportēšana un importēšana Informācijas paneļus tagad var eksportēt un importēt, kas atvieglo informācijas paneļu migrēšanu starp Zabbix instancēm vai to atkārtotu izmantošanu dažādās vidēs.

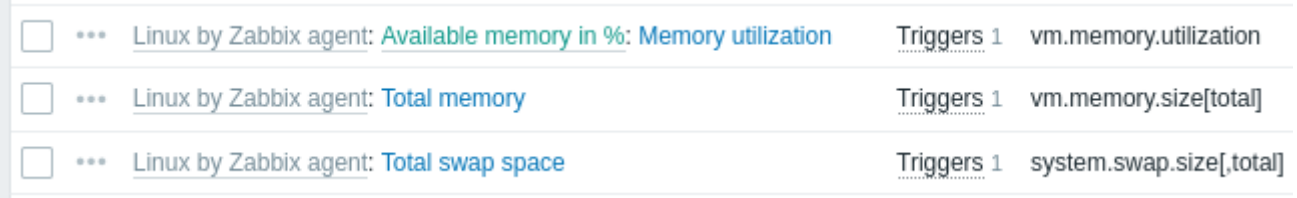
Jauna *Eksportēt* opcija ļauj saglabāt atlasītos informācijas paneļus YAML, JSON vai XML failos. Šos failus vēlāk var importēt, izmantojot *Importēt* funkcionalitāti, lai atjaunotu informācijas paneļus kopā ar to lapām, logrīkiem un konfigurāciju.

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:

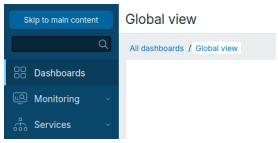


Jaunās fontu saimes tiek izmantotas gandrīz visās tēmās. Ja nepieciešams, ir pieejamas nesen pievienotās *Blue (classic)* un *Dark (classic)* tēmas ar mantotajām fontu saimēm.

Monospace satura un grafiku fontu saime paliek nemainīga.

Zabbix lietotāja saskarnes uzlabojumi Zabbix lietotāja saskarne ir modernizēta, lai uzlabotu pieejamību un lietojamību.

Pieejamības uzlabojums Lai uzlabotu navigāciju ar tastatūru un pieejamību, ir pievienota saite *Pāriet uz galveno saturu*.



Šī saite ļauj tastatūras un ekrānlasītāja lietotājiem izlaist atkārtotu sānjoslas navigāciju un pāriet tieši uz galveno saturu. Parastas lietošanas laikā tā paliek paslēpta un tiek parādīta tikai tad, kad tai tiek piešķirts fokuss, piemēram, kad pēc lapas ielādes tiek nospiests taustiņš Tab vai kad tai piekļūst, izmantojot ekrānlasītāju.

Iekšrindas validācija Tālāk norādītās formas lietotāja saskarnē ir pievienotas to formu kopai, kas atbalsta iekšrindas validāciju:

- API token
- Audit log
- Authentication
- Autoregistration
- Connector
- Discovery prototype
- Low-level discovery rule
- Network discovery rule
- Event correlation
- Geographical maps
- Graph prototypes
- Custom graphs
- GUI
- Host group

- Host prototypes
- Housekeeping
- Icon mapping
- Images
- Maintenance
- Media type
- Notifications
- Other configuration parameters
- Profile
- Proxy
- Proxy group
- Regular expressions
- Scheduled reports
- Script
- Service
- SLA
- Template group
- Test item
- Test media type
- Timeouts
- Trigger displaying options
- Trigger expression constructor
- Update problem
- User groups
- Global user macros
- User roles
- Users

Ievades kļūdas tiek parādītas uzreiz pēc lauku aizpildīšanas, uzlabojot lietojamību un samazinot konfigurācijas kļūdu skaitu.

Modālie formu logi Tālāk norādītās formas tagad tiek atvērtas modālajā logā:

- Atklāšanas prototipa konfigurācijas forma
- Zema līmeņa atklāšanas noteikuma konfigurācijas forma
- Hosta prototipa konfigurācijas forma
- Globālās regulārās izteiksmes konfigurācijas forma

Mantotie tagi, kas redzami hostos, veidnēs, vienumos un trigeros Mantotie tagi tagad tiek konsekventi parādīti un atgriezti veidnēs, hostos, vienumos, tīmekļa scenārijos un trigeros.
No veidņu un hostu ķēdēm mantotie tagi ir redzami sadaļā *Monitoring > Latest data*.
Filtrēšana pēc mantotajiem tagiem ir pieejama visur, kur tiek atbalstīta filtrēšana pēc tagiem, tostarp sadaļās *Monitoring > Latest data* un *Data collection*, kā arī visos informācijas paneļa logrīkos, kas ļauj filtrēt pēc hostu, vienumu, trigeru vai tīmekļa scenāriju tagiem. Tādējādi atlase un apakšfiltrēšana pēc tagiem darbojas vienādi neatkarīgi no tā, kur tags tika definēts.

Veidnes, hosta un hosta prototipa konfigurācijas veidlapu cilnē *Tags* tagad ir pieejams radio elements, ar kuru var izvēlēties tagu attēlošanas veidu: veidnēs tiek piedāvātas opcijas *Template tags* / *Inherited and template tags*, savukārt hostos un hostu prototipos — *Host tags* / *Inherited and host tags*.
Mantotie tagi tiek vizuāli izcelti ar jaunu konturēta dokumenta ikonu blakus taga nosaukumam.

Graph un Pie chart logrīkos ir pieejams jauns iestatījums *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.

Jauns $ZBX_FEATURE_FLAGS mainīgais lietotāja saskarnes konfigurācijā Lietotāja saskarnes konfigurācijā ir ieviests jaunais `$ZBX_FEATURE_FLAGS[]` mainīgais, lai kontrolētu piekļuvi lietotāju HTTP autentifikācijas, moduļu un datu nesēju tipu konfigurācijām. Zemāk norādītie funkciju karodziņi ir pievienoti `zabbix.conf.php` konfigurācijas failam:

- `$ZBX_FEATURE_FLAGS['http_auth_enabled'] = false` atspējo piekļuvi lietotāju HTTP autentifikācijas konfigurācijai (aizstāj mainīgo `$ALLOW_HTTP_AUTH`).
- `$ZBX_FEATURE_FLAGS['modules_config_enabled'] = false` atspējo piekļuvi moduļu konfigurācijai.
- `$ZBX_FEATURE_FLAGS['media_type_denylist'] = [<value1>, <value2>, ...]` ierobežo to datu nesēju tipu rediģēšanu, kuros tiek izmantotas norādītās piegādes metodes (`'email'`, `'script'`, `'sms'` vai `'webhook'`; var norādīt vienu vai vairākas vērtības).

Piekļuve `setup.php` ir ierobežota, ja pastāv `zabbix.conf.php` konfigurācijas fails.

Notikuma informācija rāda slāpēšanas iemeslu Notikuma informācijas lapa, kas tiek atvērta no *Monitoring* > *Problems*, tagad rāda, vai notikums tika slāpēts hosta apkopes dēļ vai manuāli slāpēts, ko veica lietotājs.

Oranžās ikonas  un  norāda uz ar apkopi saistītu slāpēšanu/neslāpēšanu, savukārt pelēkās ikonas  un  norāda uz manuālu slāpēšanu/neslāpēšanu.

Problēmu un notikumu vēsturē darbības, ko veikuši dzēsti lietotāji, tagad tiek rādītas kā *Nepieejams lietotājs*.

Sistēmas informācijas pārskata uzlabojumi Pārskats Sistēmas informācija tagad parāda servera ID (UUIDv7).
 Sistēmas informācijas pārskats tagad ir pieejams arī Zabbix *Admin* lietotājiem.

Izmaiņas Zabbix SELinux politikas pakotnē AlmaLinux, CentOS Stream, RHEL, Rocky Linux un Oracle Linux 10. versijai SELinux politika tiek nodrošināta ar jauno `zabbix-selinux-policy-main` pakotni, kurā ir iekļauta SELinux politika no augšupstrādes SELinux repozitorija.

Esošajai `zabbix-selinux-policy` pakotnei tagad ir atkarība no `zabbix-selinux-policy-main` vai `selinux-policy-targeted-extr`
`zabbix-selinux-policy-main` pakotne konfliktē ar `selinux-policy-targeted-extra`; tādēļ abas pakotnes nevar instalēt vienlaikus.

Ja jums jau ir esoša `zabbix-selinux-policy` instalācija, manuālas izmaiņas nav nepieciešamas. Pakotnes jaunināšanas laikā `zabbix-selinux-policy-main` tiks instalēta automātiski. Ja `selinux-policy-targeted-extra` jau ir instalēta, papildu pakotne netiks instalēta.

Informācijas paneļa slaidrādes iestatījums Veidojot informācijas paneli, izvēles rūtiņa *Sākt slaidrādi* pēc noklusējuma ir atspējota, un tā ir jāiespējo manuāli.

`slideshow` URL parametrs atbalsta vērtības `on` vai `off`, ļaujot ātri ignorēt informācijas paneļa slaidrādes iestatījumu.

Parametrs `slideshow=1` ir novecojis, un turpmāk vairs netiks atbalstīts.

Saite uz Zabbix Cloud starpniekserveriem Starpniekserveru sarakstā, kas atrodas sadaļā *Administration* > *Proxies*, tagad ir pieejama poga *Create proxy in Cloud*. Noklikšķinot uz tās, jaunā pārlūkprogrammas cilnē tiek atvērts Zabbix Cloud, kur varat izvietot starpniekserveri, to neinstalējot savā infrastruktūrā. Plašāku informāciju skatiet sadaļā Zabbix Cloud starpniekservera iestatīšana.

Dokumentācija

Konsolidētas dokumentācijas lapas mazajiem laidieniem Mazāko versiju laidienu dokumentācija galvenajām Zabbix laidienu versijām tagad tiks apkopota vienā dokumentācijas lapā gan jaunajām funkcijām, gan jaunināšanas piezīmēm.

Atjaunināšanas piezīmes

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

Visas piezīmes ir sagrupētas šādi:

- **Pārtraucošas izmaiņas** - izmaiņas, kas var pārtraukt esošu instalāciju darbību, un cita būtiska informācija, kas saistīta ar atjaunināšanas procesu
- **Cits** - visa pārējā informācija, kas apraksta izmaiņas Zabbix funkcionalitātē

Skatiet arī:

- Atjaunināšanas procedūra visai būtiskajai informācijai par atjaunināšanu no versijām, kas ir vecākas par Zabbix 8.0.0;
- HA klastera atjaunināšana norādījumiem par serveru atjaunināšanu **augstas pieejamības** (HA) klasterī.

Būtiskas izmaiņas

Hostu saskarņu API metodes noņemtas no lietotāju lomām Pēc Zabbix jaunināšanas `hostinterface.create`, `hostinterface.update`, `hostinterface.delete`, `hostinterface.massadd` un `hostinterface.massremove` API metodes tiks noņemtas no esošo lietotāju lomu *Access to API* sarakstiem.

Pielāgotas tēmas un multivides tips Zabbix pievieno trīs jaunas iepriekš definētas lietotāja saskarnes tēmas: *Blue (classic)*, *Dark (classic)* un *Dark blue*, kā arī jaunu Push notification multivides tipu. Ja jums ir pielāgota tēma ar kādu no šiem nosaukumiem (`blue-classic-theme`, `dark-classic-theme`, `dark-blue-theme`) vai multivides tips ar nosaukumu `Push notification`, pirms jauninašanas pārdēvējiet tēmu vai multivides tipu. Pretējā gadījumā pēc jauninašanas:

- Iepriekš definētā tēma aizstās jūsu pielāgotās tēmas stilu lapu, bet jūsu pielāgotās grafiku krāsas tiks saglabātas, tādējādi izveidojot abu tēmu apvienojumu.
- Jūsu esošais multivides tips paliks nemainīgs, bet jaunais multivides tips netiks izveidots.

Jauninašana neietekmē pielāgotās tēmas un multivides tipus ar citiem nosaukumiem.

Iekšējie starpniekservera vienumi ir ierobežoti ar servera uzraudzītajiem hostiem Iekšējie vienumi `zabbix[proxy,*]` un `zabbix[proxy group,*]` vairs netiek atbalstīti hostos, kurus uzrauga starpniekserveris vai starpniekserveru grupa.

Pārtraukts atbalsts Zabbix aģentiem, kas izdoti pirms 1.6. versijas Tā kā no Zabbix servera/starpniekservera trapera ir noņemts mantoto vienkāršā teksta un XML protokolu atbalsts, Zabbix aģenti, kas izdoti pirms 1.6. versijas, vairs netiek atbalstīti.

Minimālā Logback bibliotēkas versija Minimālā Logback bibliotēkas versija ir paaugstināta no 1.5.16 uz 1.5.25.

Iekļautā validācija Tālāk norādītās formas lietotāja saskarnē ir pievienotas formu kopai, kas atbalsta iekļauto validāciju:

- API token
- Audit log
- Authentication
- Autoregistration
- Connector
- Event correlation
- Geographical maps
- Graph prototypes
- Custom graphs
- GUI
- Host group
- Host prototypes
- Housekeeping
- Icon mapping
- Images
- Maintenance
- Media type
- Notifications
- Other configuration parameters
- Profile
- Proxy
- Proxy group
- Regular expressions
- Scheduled reports
- Script
- Service
- SLA
- Template group
- Test item
- Test media type
- Timeouts
- Trigger displaying options
- Trigger expression constructor
- Update problem
- User groups
- Global user macros
- User roles
- Users

Ievades kļūdas tiek parādītas uzreiz pēc lauku aizpildīšanas, uzlabojot lietojamību un samazinot konfigurācijas kļūdu skaitu.

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

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

Parametram `Plugins.Oracle.CallTimeout` Oracle spraudņa konfigurācijas failā vairs nav nekādas ietekmes. Maksimālais gaidīšanas laiks tagad tiek noteikts pēc vienuma taimauta.

MongoDB spraudņa versiju atbalsts Šī spraudņa versija vairs neatbalsta MongoDB 2.6–4.0. Tā izmanto MongoDB Go Driver v2, kam nepieciešama MongoDB versija 4.2 vai jaunāka.

Lai pārraudzītu MongoDB 2.6–4.0, izmantojiet iepriekšējo spraudņa versiju. Iepriekšējās versijas atbalsta MongoDB 2.6–5.3, un tās var izmantot ar Zabbix aģenta 2 versiju 6.0.0 vai jaunāku.

Minimālā nepieciešamā PHP versija Minimālā nepieciešamā PHP versija ir paaugstināta no 8.0.0 uz 8.2.0.

Paplašinātais UnsafeUserParameters saraksts Rakstzīme `%` ir pievienota `UnsafeUserParameters` sarakstam Zabbix aģents un Zabbix aģents 2.

Pārvietoti filtra parametri Kā daļa no pielāgojamām tabulām daži filtra parametri ir pārvietoti no filtra cilnes uz kolonnu opcijām. Ja esat saglabājis cilnes iecienītākajiem filtriem, kurās tiek izmantots kāds no tālāk norādītajiem filtra parametriem, tās tiks atiestatītas un pēc jauninašanas būs jākonfigurē manuāli:

- *Monitoring* > *Hosts*: *Rādīt apslāpētās problēmas*
- *Monitoring* > *Latest data*: *Rādīt tagus*, *Taga nosaukums*, *Taga rādīšanas prioritāte*, *Rādīt detalizētu informāciju*
- *Monitoring* > *Problems*: *Rādīt tagus*, *Taga nosaukums*, *Taga rādīšanas prioritāte*, *Rādīt operatīvos datus*, *Kompakts skats*, *Rādīt detalizētu informāciju*, *Rādīt laika skalu*, *Izcelt visu rindu*

JavaScript priekšapstrāde: iebūvētās objektu metodes tagad ir tikai lasāmas Iebūvētās JavaScript objektu metodes priekšapstrādes skriptos tagad ir tikai lasāmas. Ja vienumu priekšapstrādes skriptos jau esat modificējis iebūvētās metodes JavaScript prototipos, varat saskarties ar problēmām, jo iebūvēto prototipa metožu modificēšana vairs nav atļauta.

MySQL datubāzes rakstzīmju kopa un salīdzināšana Ja Zabbix MySQL datubāze tika izveidota ar utf8mb3 rakstzīmju kopu un utf8mb3_bin salīdzināšanu, pirms Zabbix jauninašanas ir stingri ieteicams to pārveidot uz utf8mb4 rakstzīmju kopu un utf8mb4_bin salīdzināšanu, kā aprakstīts sadaļā Zabbix datubāzes rakstzīmju kopas un salīdzināšanas labošanas.

Citi

SAML parakstīšana pēc noklusējuma ir iespējota Lai uzlabotu drošību, visas SAML parakstīšanas opcijas (*Messages*, *Assertions*, *AuthN requests*, *Logout requests* un *Logout responses*) tagad pēc noklusējuma ir iespējotas, konfigurējot SAML autentifikāciju sadaļā *Users* > *Authentication*. Tādēļ jaunām SAML konfigurācijām tagad ir nepieciešama SP privātā atslēga un SP sertifikāts, ja vien parakstīšanas opcijas netiek notīrītas.

Esošās SAML konfigurācijas pēc jauninašanas netiek ietekmētas.

Novecojušas Kubernetes veidnes Kubernetes veidnes ir novecojušas, un tās ir aizstātas ar veidni *Kubernetes Cluster by HTTP*. Vecās veidnes tiek saglabātas atpakaļsaderībai ar esošajām konfigurācijām.

Novecojušie makrosi ir noņemti Atbalsts šādiem iebūvētajiem makro tagad ir noņemts:

Noņemts	Lietot 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}

Novecojuši parametri vēstures datu glabāšanas nodrošinātājiem Tālāk norādītie parametri tagad ir novecojuši:

- Zabbix servera konfigurācijas fails: `HistoryStorageDateIndex`, `HistoryStorageURL` un `HistoryStorageTypes`. Tā vietā izmantojiet jauno parametru `HistoryProvider`.
- Zabbix lietotāja saskarnes konfigurācijas fails: `$HISTORY`. Tā vietā izmantojiet jauno parametru `$HISTORY_PROVIDERS`.

Abi jaunie parametri ļauj izvēlēties vēstures datu glabāšanas nodrošinātāju: Elasticsearch, ClickHouse vai SQL (noklusējuma opcija). Ņemiet vērā, ka jaunos un novecojušos parametrus nevar izmantot kopā.

No HTTP aģenta vienumiem noņemta JSON un XML validācija JSON un XML validācija ir noņemta no HTTP aģents vienumu lauka *Request body*. Šis lauks vairs nepārbauda, vai vērtība ir derīgs XML vai JSON, un, izvēloties *XML data* kā *Request body type*, vairs nav nepieciešama libxml2 bibliotēka.

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

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

Noņemta radio poga jaunās tagu filtra formas sadaļā Radio poga, kas, izveidojot jaunu tagu filtru, prasīja izvēlēties starp *Visi tagi* un *Tagu saraksts*, ir noņemta.

Vienumu vērtības, kas tiek attēlotas kā HTML, izmantojot iframe Vienumu dati Item history logrīkā, kad tie tiek attēloti kā HTML formatēts teksts, tagad ir izolēti iframe.

TLS pārbaude HTTP veidnēm un multivides tipiem Veidnes un multivides tipi, kas izveido savienojumu ar HTTPS galapunktiem, tagad pēc noklusējuma pārbauda galapunkta TLS sertifikātu.

Šīs izmaiņas attiecas uz visām mākoņpakalpojumu un API uzraudzības veidnēm (piemēram, *Claude API by HTTP*, *Cloudflare by HTTP*, *Kubernetes Cluster by HTTP*) un visiem webhook multivides tipiem (piemēram, *Slack*, *Jira*, *Telegram*). Šo darbību var mainīt, izmantojot jauno lietotāja makro `{$HTTP.TLS.VERIFY}`, kas pievienots šīm veidnēm un multivides tipiem. Iestatiet makro vērtību uz `none`, lai atspējotu pārbaudi, uz `peer`, lai pārbaudītu tikai sertifikātu ķēdi un derīguma termiņu, vai uz `full`, lai veiktu pilnu pārbaudi. Plašāku informāciju skatiet katras HTTP veidnes vai webhook README failā.

Atjauninot Zabbix, veidnes un multivides tipi netiek atjaunināti automātiski. Tas novērš jūsu veikto pielāgoto izmaiņu pārrakstīšanu. Lai manuāli atjauninātu vai pievienotu veidnes un multivides tipus, skatiet veidņu atjaunināšanas norādījumus.

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

Konfigurācijas lauku automātiska izmēra pielāgošana Lielākā daļa konfigurācijas lauku lietotāja saskarnē tagad automātiski pielāgo savu izmēru, lai atbilstu to saturam.

Aizstātā lietotāja saskarnes konfigurācijas mainīgā vērtība Mainīgais `$ALLOW_HTTP_AUTH` ir aizstāts ar funkcionalitātes karogu `$ZBX_FEATURE_FLAGS['http_auth_enabled']` lietotāja saskarnes konfigurācijas failā (`zabbix.conf.php`).

SNMPv1 un SNMPv2 pārbaužu vienlaicīga apstrāde SNMPv1 un SNMPv2 asinhronie servisa vienumu pārbaudes tagad tiek apstrādātas vienlaicīgi, izmantojot pieejamos atklāšanas darbiniekus.

Izmaiņas Zabbix SELinux politikas pakotnē AlmaLinux, CentOS Stream, RHEL, Rocky Linux un Oracle Linux 10. versijai esošajai `zabbix-selinux-policy` pakotnei tagad ir atkarība no `zabbix-selinux-policy-main` vai `selinux-policy-targeted-extra`.
Ja jums jau ir instalēta `zabbix-selinux-policy`, manuālas izmaiņas nav nepieciešamas. Pakotnes jaunināšanas laikā `zabbix-selinux-policy-main` tiks instalēta automātiski. Ja `selinux-policy-targeted-extra` jau ir instalēta, papildu pakotne netiks instalēta.

APM support in Zabbix proxy packages To use application performance monitoring after upgrading Zabbix proxy, choose the option for your operating system:

- RHEL 10 and its derivatives - install the EPEL repository, enable the CRB repository, and install the Zabbix proxy APM package (see details).
- Debian 12 or later, Raspberry Pi OS (Raspbian) 12 or later, Ubuntu 24.04 or later - upgrade Zabbix proxy packages as usual; APM is included.
- Other systems - compile Zabbix proxy from sources with the `--with-apm` option.

Pārdēvēts apkopes perioda stāvoklis Apkopes perioda stāvoklis ir pārdēvēts no *Approaching* uz *Upcoming*.

Novecojis informācijas paneļa slaidrādes parametrs Atbalsts parametram `slideshow=1` tiks noņemts. Parametram `slideshow` ir ieviestas jaunas vērtības, lai palaistu un apturētu slaidrādi:

- `slideshow=on` - palaist slaidrādi
- `slideshow=off` - apturēt slaidrādi

Izmaiņas globālajās regulārajās izteiksmēs

- Lauks *Description* ir pievienots regulāro izteiksmju konfigurācijas formai.
- Opciju *Expression type* nosaukumi ir mainīti:

Before	Now
Character string included	*Contains string*
Any character string included	*Contains any substring from list*
Character string not included	*Does not contain string*
Result is TRUE	*Matches regular expression*
Result is FALSE	*Does not match regular expression*

Noklusējuma piekļuve API Jaunās instalācijās piekļuve API pēc noklusējuma ir atspējota visām lomām, tostarp jaunizveidotām lomām un noklusējuma lietotāju lomām.

Esošajās instalācijās piekļuve API pēc noklusējuma ir atspējota tikai jaunizveidotām lomām.

Empty API allow list In user roles, an *Allow list* with no API methods specified now denies access to all API methods. Previously, the empty list was ignored, and all API methods were allowed.

To preserve existing access, the upgrade adds the * wildcard (any method) to all user roles that have *Allow list* selected but no methods specified. After the upgrade, review these roles in *Users* > *User roles*: replace * with the methods the role actually needs, or clear the *Enabled* checkbox in the *Access to API* section if the role does not need API access.

Jaunumi un jauninašanas piezīmes: 8.0.x

Šajā lapā ir aprakstītas visas jaunās funkcijas, kas iekļautas galvenās Zabbix versijas mazākajos laidienos. Tajā ir iekļautas arī veidņu izmaiņas un jauninašanas piezīmes.

Skatiet arī galvenās versijas sadaļu Jaunumi.

Kas jauns Zabbix 8.0.1 Šī versija vēl nav izlaista.

Skatiet jauninašanas piezīmes šai versijai.

Jauninašanas piezīmes

Zabbix ierīce

Pārskats Zabbix appliance nodrošina iespēju nekavējoties izvietot Zabbix serveri un lietotāja saskarni, nevis tos iestatīt manuāli vai atkārtoti izmantot esošu serveri Zabbix vajadzībām.

Zabbix 8.0 appliance ir balstīts uz tādiem Red Hat Enterprise Linux 10 atvasinājumiem kā AlmaLinux vai Rocky Linux, un tajā ir iepriekš konfigurēts Zabbix serveris, kas darbojas ar MySQL, kā arī lietotāja saskarne, kas darbojas NGINX tīmekļa serverī.

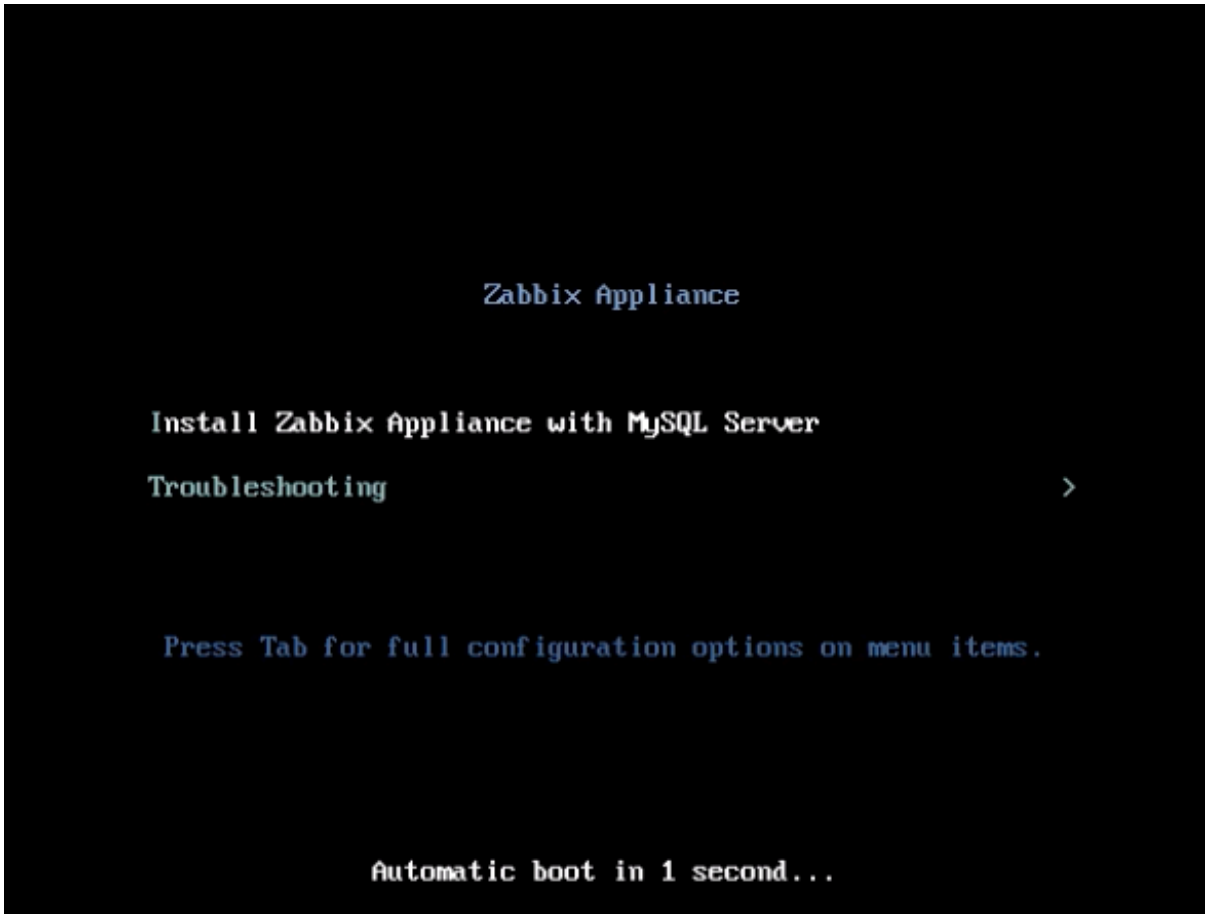
> **Attention:**
> Šis appliance ir paredzēts Zabbix novērtēšanai. Tā izmantošana nopietnās ražošanas vidēs nav ieteicama.

Appliance attēli ir pieejami lejupielādei šādos formātos:

- Instalācijas CD/DVD (.iso)

- VMware (.vmx) - skatiet piezīmes
- Open virtualization format (.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 ierīces virtuālo mašīnu no lejupielādētā attēla. Norādījumus skatiet savas virtualizācijas programmatūras dokumentācijā, piemēram, VirtualBox dokumentācijā.
2. Konfigurējiet virtuālās mašīnas tīkla iestatījumus, lai nodrošinātu piekļuvi no pārlūkprogrammas uz hosta datora. To var panākt, iespējojot *Bridged mode*.
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 uz hosta datora un norādiet tajā IP adresi, ko ierīce ir saņēmusi, 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.

Datubāze

Visu datubāzes lietotāju paroles instalēšanas procesa laikā tiek ģenerētas nejauši. Datubāzei ir definēti šādi lietotāji:

Root:

- Lietotājvārds: `root`
- Parole: parole tiek glabāta failā `/root/.my.cnf` Parole nav jāievada, izmantojot root kontu.

Zabbix serveris:

- Lietotājvārds: `zabbix_srv`
- Parole: parole tiek glabāta servera konfigurācijas failā

Zabbix lietotāja saskarne:

- Lietotājvārds: `zabbix_web`
- Parole: parole tiek glabāta lietotāja saskarnes konfigurācijas failā

Lai mainītu datubāzes lietotāja paroli, mainiet to MySQL un atbilstošajā konfigurācijas failā.

Lietotāja saskarnes piekļuve

Zabbix lietotāja saskarnei var piekļūt, izmantojot adresi *http://<virtuālās mašīnas IP adrese>*.

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, izmantojot SSH, kā root lietotājs, un izpildot:

```
systemctl restart nginx
```

Statiskā IP adrese

Pēc noklusējuma ierīce izmanto DHCP, lai iegūtu IP adresi. Lai iestatītu statisku IP adresi:

1. Piesakieties kā root lietotājs.
2. Izpildiet šādas komandas, aizstājot vērtības ar savu pielāgoto IP adresi:

```
nmcli con mod eth0 ipv4.addresses 192.168.1.10/24    # Ierīces IP adrese/CIDR prefikss
nmcli con mod eth0 ipv4.gateway 192.168.1.1     # Vārtejas IP adrese
nmcli con mod eth0 ipv4.dns 8.8.8.8    # DNS servera IP adrese
nmcli con mod eth0 ipv4.method manual
nmcli con up eth0
```

Alternatīvi, varat atvērt failu `/etc/NetworkManager/system-connections/eth0.nmconnection` un pievienot šādas rindas:

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

Pēc rediģētā faila saglabāšanas izpildiet komandas:

```
nmcli con reload
nmcli con up eth0
```

Lai pārbaudītu, vai tīkla konfigurācijas izmaiņas ir piemērotas, izpildiet šādu komandu:

```
nmcli dev show eth0
```

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

- Zabbix servera un aģenta konfigurācijas faili atrodas `/etc/zabbix`.
- Zabbix servera un aģenta žurnālfaili atrodas `/var/log/zabbix`.
- Zabbix lietotāja saskarnes konfigurācijas fails atrodas `/etc/zabbix/web`.
- Zabbix lietotāja saskarne atrodas `/usr/share/zabbix`.
- Lietotāja *zabbix* mājas direktorija 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

Vmdk formāta attēlus var tieši izmantot VMware Player, Server un Workstation produktos.
Lai tos izmantotu ESX, ESXi un vSphere vidē, tie ir jākonvertē, izmantojot VMware vCenter Converter (lejupielādei nepieciešama autentifikācija).
Ja izmantojat VMware vCenter Converter, var rasties problēmas ar hibrīdo tīkla adapteri.
Šādā gadījumā konvertēšanas procesa laikā varat mēģināt norādīt E1000 adapteri.
Varat arī pēc konvertēšanas pabeigšanas izdzēst esošo adapteri un pievienot E1000 adapteri.

HDD/flash attēls (raw)

Lai sāknētu attēlu, izpildiet:

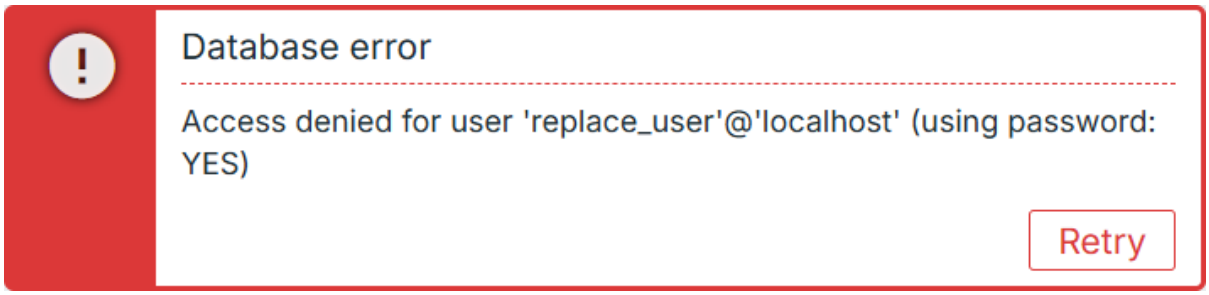
```
dd if=./zabbix_appliance_8.0.0.raw of=/dev/sdc bs=4k conv=fdatasync
```

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, tiek parādīts kļūdas ziņojums *Access denied for user 'replace_user'@'localhost' (using password: YES)*, tas var norādīt, ka instalēšana joprojām notiek.



Ja kļūda saglabājas arī pēc dažu minūšu gaidīšanas vai ja pamanāt kādu citu negaidītu darbību, tas, visticamāk, nozīmē, ka instalēšanas process nav pabeigts veiksmīgi. Šādā gadījumā iesakām dzēst pašreizējo ierīci un izvietot to atkārtoti, ievērojot tās pašas instalēšanas instrukcijas. Šis solis parasti novērš problēmu.

Ņemiet vērā, ka nav ieteicams mēģināt manuāli labot bojātu instalāciju, jo tas var radīt papildu sarežģījumus.

Procesi

Pārskats Zabbix sastāv no vairākiem procesiem, kas darbojas kopā, lai uzraudzītu sistēmas un nosūtītu paziņojumus. Katram procesam uzraudzības infrastruktūrā ir noteikta loma.

Pamatprocesi

- Serveris - centrālais process, kas veic datu aptauju un uztveršanu, aprēķina trigerus un nosūta lietotājiem paziņojumus.
- Aģents - apkopo datus no uzraudzītajiem hostiem.
- Aģents 2 - jaunās paaudzes aģents ar spraudņu arhitektūru un paplašinātām iespējām.
- Starpniekserveris - Zabbix servera vārdā apkopo uzraudzības datus no uzraudzītajām ierīcēm un izplatītā iestatījumā nosūta šo informāciju centralizētam Zabbix serverim.
- Java gateway - nodrošina Java lietojumprogrammu uzraudzību.
- MCP serveris - ļauj MI aģentiem strādāt ar Zabbix, izmantojot Model Context Protocol.

Lai ērti piekļūtu Zabbix no jebkuras vietas un platformas, ir pieejama tīmekļa lietotāja saskarne. Lietotāja saskarne ir Zabbix servera daļa un parasti (bet ne obligāti) darbojas tajā pašā fiziskajā datorā, kurā darbojas serveris.

Visa konfigurācijas informācija, kā arī Zabbix apkopotie dati tiek glabāti datubāzē.

Virsraksts Galvene ir iekļauta visos pieprasījuma un atbildes ziņojumos starp Zabbix komponentēm.
Tā ir nepieciešama, lai noteiktu ziņojuma garumu, to, vai tas ir saspiests, un to, vai tas ir liels pakotnes formāts.

Zabbix saziņas protokolam ir 1 GB pakotnes lieluma ierobežojums vienam savienojumam. 1 GB ierobežojums tiek piemērots gan saņemtās pakotnes datu garumam, gan nesaspiesto datu garumam.

Nosūtot konfigurāciju uz Zabbix starpniekserveris, pakotnes lieluma ierobežojums tiek palielināts līdz 4 GB, lai ļautu sinhronizēt lielas konfigurācijas. Ja datu garums pirms saspiešanas pārsniedz 4 GB, Zabbix serveris automātiski sāk izmantot lielas pakotnes formātu (0x04 karodziņš), kas palielina pakotnes lieluma ierobežojumu līdz 16 GB.

Ņemiet vērā, ka, lai gan lielas pakotnes formātu var izmantot jebkuru datu nosūtīšanai, pašlaik tikai Zabbix starpniekservera konfigurācijas sinhronizētājs var apstrādāt pakotnes, kas ir lielākas par 1 GB.

Galvenes struktūra Galvene sastāv no četriem laukiem. Visi skaitļi galvenē ir formatēti kā little-endian.

Field	Size	Size (large packet)	Description
<PROTOCOL>	4	4	"ZBXD" vai 5A 42 58 44
<FLAGS>	1	1	Protokola karodziņi: 0x01 - Zabbix sakaru protokols 0x02 - saspiešana 0x04 - liels pakotnes izmērs
<DATALEN>	4	8	Datu garums.

Field	Size	Size (large packet)	Description
<RESERVED>	4	8	Ja tiek izmantota saspiešana (0x02 karodziņš) - nesaspiesto datu garums Ja saspiešana netiek izmantota - 00 00 00 00

Galvenes 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 interpretēta pareizi. Šie koda fragmenti pieņem, ka datu apjoms nav lielāks par 1 GB, tāpēc lielās paketes 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"
```

serveris

Pārskats Zabbix serveris ir Zabbix programmatūras centrālais process.

Serveris veic datu aptauju un uztveršanu, aprēķina trigerus un nosūta paziņojumus lietotājiem. Tas ir centrālais komponents, kuram Zabbix aģenti un starpniekserveri ziņo datus par sistēmu pieejamību un integritāti. Serveris pats var 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 tas ir Zabbix elements, kas aktīvi brīdina administratorus, kad jebkurā no uzraudzītajām sistēmām rodas problēmas.

Pamata Zabbix servera darbība ir sadalīta trīs atsevišķos komponentos; tie ir: Zabbix serveris, lietotāja saskarne un datubāzes krātuve.

Visa Zabbix konfigurācijas informācija tiek glabāta datubāzē, ar kuru mijiedarbojas gan serveris, gan lietotāja saskarne. Piemēram, kad jūs izveidojat jaunu vienumu, izmantojot lietotāja saskarni (vai API), tas tiek pievienots datubāzes items tabulai. Pēc tam aptuveni reizi minūtē Zabbix serveris vaicās items tabulu, lai iegūtu aktīvo vienumu sarakstu, kas pēc tam tiek saglabāts kešatmiņā Zabbix serverī. Tāpēc var paiet līdz pat divām minūtēm, līdz jebkuras Zabbix lietotāja saskarnē veiktās izmaiņas parādās jaunāko datu sadaļā.

Skatiet arī:

- Augstas pieejamības iestatīšana Zabbix serveriem
- Konfigurācijas parametri: Zabbix serveris

Servera darbība 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, iespējams, būs jāizpilda:

```
/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
```

Startēt manuāli

Ja iepriekš minētais nedarbojas, jums tas jāstartē manuāli. Atrodiet `zabbix_server` binārā faila ceļu un izpildiet:

```
zabbix_server
```

Ar Zabbix serveri varat izmantot šādus komandrindas parametrus:

Parameter	Description
`-c --config <config-file>`	Ceļš uz konfigurācijas failu.
`-f --foreground`	Palaist Zabbix serveri priekšplānā.
`-R --runtime-control <option>`	Veikt administratīvās funkcijas.
`-T --test-config`	Pārbaudīt konfigurācijas failu un iziet.
`-h --help`	Parādīt palīdzības informāciju un iziet.
`-V --version`	Parādīt versijas numuru.

Zabbix servera palaišanas piemēri ar komandrindas parametriem:

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

Darbības vadība

Darbības vadības opcijas:

Opcija	Apraksts	Mērķis
`config_cache_reload`	Atkārtoti ielādēt konfigurācijas kešatmiņu. Tiek ignorēts, ja kešatmiņa pašlaik tiek ielādēta.	
`history_cache_clear=<target>`	Notīrīt vēstures kešatmiņu vienumam, kas norādīts pēc 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` - mutex saraksts (BSD sistēmās tas ir tukšs); `connector` - to savienotāju statistika, kuriem ir visgarākā rinda.
`ha_status`	Ierakstīt augstas pieejamības (HA) klastera statusu žurnālā.	
`ha_remove_node=<target>`	Noņemt augstas pieejamības (HA) mezglu, kas norādīts pēc tā nosaukuma vai ID. Ņemiet vērā, ka aktīvos/rezerves mezglus nevar noņemt.	**target** - mezgla nosaukums vai ID (to var iegūt, izpildot `ha_status`).
`ha_set_failover_delay=<delay>`	Iestatīt augstas pieejamības (HA) pārslēgšanās aizkavi. Tiek atbalstīti laika sufiksi, piemēram, 10s, 1m.	
`proxy_config_cache_reload[=<target>]`	Atkārtoti ielādēt starpniekservera konfigurācijas kešatmiņu.	**target** - ar komatiem atdalīts starpniekserveru nosaukumu saraksts. Ja mērķis nav norādīts, konfigurācija tiek atkārtoti ielādēta visiem starpniekserveriem.
`secrets_reload`	Atkārtoti ielādēt noslēpumus no Vault.	
`service_cache_reload`	Atkārtoti ielādēt pakalpojumu pārvaldnieka kešatmiņu.	

Opcija	Apraksts	Mērķis
`snmp_cache_reload`	Atkārtoti ielādēt SNMP kešatmiņu — notīrīt SNMP dzinēja rekvizītus (dzinēja laiku, dzinēja palaišanas reižu skaitu, dzinēja ID, akreditācijas datus) visiem hostiem. Izmantojiet šo opciju, lai SNMP problēmu novēršanas laikā piespiedu kārtā notīrītu globālo kešatmiņu.	
`housekeeper_execute`	Sākt uzkopšanas procedūru. Tiek ignorēts, ja uzkopšanas procedūra pašlaik jau notiek.	
`trigger_housekeeper_execute`	Sākt trigeru uzkopšanas procedūru. Tiek ignorēts, ja trigeru uzkopšanas procedūra pašlaik jau notiek.	
`log_level_increase[=<target>]`	Palielināt žurnāla līmeni; ja mērķis nav norādīts, tas ietekmē visus procesus. BSD sistēmās netiek atbalstīts.	**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 formātā "process type,N".
`log_level_decrease[=<target>]`	Samazināt žurnāla līmeni; ja mērķis nav norādīts, tas ietekmē visus procesus. BSD sistēmās netiek atbalstīts.	
`prof_enable[=<target>]`	Iespējot profilēšanu. Ja mērķis nav norādīts, tas ietekmē visus procesus. Iespējota profilēšana nodrošina detalizētu informāciju par visiem rwlock/mutex objektiem 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`, `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 formātā "process type,N". **scope** - `rwlock`, `mutex`, `processing` var izmantot kopā ar procesa tipu un numuru (piemēram, `history syncer,1,processing`) vai visiem attiecīgā tipa procesiem (piemēram, `history syncer,rwlock`).
`prof_disable[=<target>]`	Atspējot profilēšanu. Ja mērķis nav norādīts, tas ietekmē visus procesus.	**process type** - visi norādītā tipa procesi (piemēram, `history syncer`). Atbalstītos procesu tipus kā profilēšanas mērķus skatiet pie `prof_enable`. **process type,N** - procesa tips un numurs (piemēram, `history syncer,1`). **pid** - procesa identifikators (1 līdz 65535). Lielākām vērtībām norādiet mērķi formātā "process type,N".

Ja procesa tips ir norādīts kā runtime control mērķis, komanda tiek piemērota tikai šī tipa darbojošajiem procesiem. Ja pašlaik nedarbojas neviens norādītā tipa process, komanda atgriež kļūdu:

```
zabbix_server -R log_level_increase='discovery manager'
Cannot redirect signal: "discovery manager" process does not exist
```

Runtime control izmantošanas piemērs servera konfigurācijas kešatmiņas pārlādēšanai:

```
zabbix_server -c /usr/local/etc/zabbix_server.conf -R config_cache_reload
```

Runtime control izmantošanas piemēri starpniekservera konfigurācijas pārlādēšanai:

```
### 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
```

Runtime control izmantošanas piemērs vienuma vēstures kešatmiņas notīrīšanai:

```
zabbix_server -c /usr/local/etc/zabbix_server.conf -R history_cache_clear=42243
```

Runtime control izmantošanas piemēri diagnostikas informācijas iegūšanai:

```
### Iegūt visu pieejamo diagnostikas informāciju servera žurnāla failā:
zabbix_server -R diaginfo

### Iegūt vēstures kešatmiņas statistiku servera žurnāla failā:
zabbix_server -R diaginfo=historycache
```

Runtime control izmantošanas piemērs SNMP kešatmiņas pārlādēšanai:

```
zabbix_server -R snmp_cache_reload
```

Attention:
Kad SNMPv3 interfeiss tiek atjaunināts, izmantojot Zabbix lietotāja saskarni, vairumā gadījumu Zabbix automātiski pārlādēs šī interfeisa jaunās SNMPv3 akreditācijas; izmantojiet `-R snmp_cache_reload` tikai tad, ja aptauja joprojām neizdodas pēc akreditācijas datu maiņas (piemēram, engineBoots/engineID neatbilstību vai ierīču, kas neatbilst RFC, dēļ), vai arī tad, ja problēmu novēršanai ir nepieciešams piespiedu kārtā notīrīt globālo SNMP kešatmiņu.

Runtime control izmantošanas piemērs housekeeper izpildes aktivizēšanai:

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

Runtime control izmantošanas piemēri žurnāla līmeņa maiņai:

```
### Palielināt visu procesu žurnāla līmeni:
zabbix_server -c /usr/local/etc/zabbix_server.conf -R log_level_increase

### Palielināt otrā poller procesa žurnāla līmeni:
zabbix_server -c /usr/local/etc/zabbix_server.conf -R log_level_increase=poller,2

### Palielināt procesa ar PID 1234 žurnāla līmeni:
zabbix_server -c /usr/local/etc/zabbix_server.conf -R log_level_increase=1234

### Samazināt visu http poller procesu žurnāla līmeni:
zabbix_server -c /usr/local/etc/zabbix_server.conf -R log_level_decrease="http poller"
```

HA kļūmjpārlēces aizkaves iestatīšanas piemērs uz minimālajām 10 sekundēm:

```
zabbix_server -R ha_set_failover_delay=10s
```

Procesa lietotājs

Zabbix serveris ir izstrādāts tā, lai tas darbotos kā lietotājs, kas nav root. Tas darbosies ar jebkuru lietotāju, kas nav root, ar kuru tas tiek palaists. Tāpēc jūs varat palaist serveri kā jebkuru lietotāju, kas nav root, bez jebkādām problēmām.

Ja mēģināsiet to palaist kā `root`, tas pārslēgsies uz iepriekš noteiktu `zabbix` lietotāju, kuram jābūt pieejamam jūsu sistēmā. Jūs varat palaist serveri kā `root` tikai tad, ja attiecīgi modificējat `AllowRoot` parametru servera konfigurācijas failā.

Ja Zabbix serveris un aģents darbojas vienā un tajā pašā datorā, ieteicams servera palaišanai izmantot citu lietotāju nekā aģenta palaišanai. Pretējā gadījumā, ja abi darbojas kā viens un tas pats lietotājs, aģents var piekļūt servera konfigurācijas failam, un jebkurš Zabbix Admin līmeņa lietotājs var diezgan viegli iegūt, piemēram, datubāzes paroli.

Konfigurācijas fails

Servera konfigurācijas faila noklusējuma ceļš ir atkarīgs no operētājsistēmas un Zabbix instalēšanas metodes:

- Instalējot no pirmkoda: `/usr/local/etc/zabbix_server.conf` vai ceļš, kas iestatīts ar kompilēšanas laikā norādītajiem mainīgajiem `--sysconfdir` vai `--prefix`.
- Instalējot no pakotnēm vai appliance: `/etc/zabbix/zabbix_server.conf`.

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 veidi un pavedieni

- `agent poller` - asinhrons pollētāja process pasīvajām pārbaudēm ar darba pavedienu;
- `alert manager` - brīdinājumu rindas pārvaldnieks;
- `alert syncer` - brīdinājumu datubāzes rakstītājs;
- `alerter` - process paziņojumu nosūtīšanai;
- `availability manager` - process hostu pieejamības atjaunināšanai;
- `browser poller` - pollētājs pārlūkprogrammas vienumu pārbaudēm;
- `configuration syncer` - process konfigurācijas datu atmiņas kešatmiņas pārvaldībai;
- `configuration syncer worker` - process lietotāja makro vērtību atrisināšanai un sinhronizēšanai vienumu nosaukumos;
- `connector manager` - savienotāju pārvaldības process;
- `connector worker` - process pieprasījumu apstrādei no savienotāju pārvaldnieka;
- `discovery manager` - ierīču atklāšanas pārvaldības process;
- `discovery worker` - process atklāšanas uzdevumu apstrādei no atklāšanas pārvaldnieka;
- `escalator` - process darbību eskalācijai;
- `ha manager` - augstas pieejamības pārvaldības process;
- `history poller` - process aprēķināto pārbaužu apstrādei, kurām nepieciešams savienojums ar datubāzi;
- `history syncer` - vēstures datubāzes rakstītājs;
- `housekeeper` - process novecojušu datu (vienumu vēstures un tendenču, lietotāju sesiju, notikumu u. c.), kā arī dzēsto objektu atstāto datu noņemšanai;
- `http agent poller` - asinhrons pollētāja process HTTP pārbaudēm ar darba pavedienu;
- `http poller` - tīmekļa uzraudzības pollētājs;
- `icmp pinger` - pollētājs icmpping pārbaudēm;
- `internal poller` - pollētājs iekšējām pārbaudēm;
- `ipmi manager` - IPMI pollētāja pārvaldnieks;
- `ipmi poller` - pollētājs IPMI pārbaudēm;
- `java poller` - pollētājs Java pārbaudēm;
- `lld manager` - zema līmeņa atklāšanas uzdevumu pārvaldības process;
- `lld worker` - zema līmeņa atklāšanas uzdevumu darba process;
- `odbc poller` - pollētājs ODBC pārbaudēm;
- `poller` - parastais pollētājs pasīvajām pārbaudēm;
- `preprocessing manager` - priekšapstrādes uzdevumu pārvaldnieks ar priekšapstrādes darba pavedieniem;
- `preprocessing worker` - pavediens datu priekšapstrādei;
- `proxy poller` - pollētājs pasīvajiem starpniekserveriem;
- `proxy group manager` - starpniekserveru slodzes līdzsvarošanas un augstas pieejamības pārvaldnieks;
- `report manager`- plānoto pārskatu ģenerēšanas uzdevumu pārvaldnieks;
- `report writer` - process plānoto pārskatu ģenerēšanai;
- `self-monitoring` - process iekšējās servera statistikas apkopošanai;
- `service manager` - process pakalpojumu pārvaldībai, saņemot informāciju par problēmām, problēmu tagiem un problēmu atkopšanu no vēstures sinhronizētāja, uzdevumu pārvaldnieka un brīdinājumu pārvaldnieka;
- `snmp poller` - asinhrons pollētāja process SNMP pārbaudēm ar darba pavedienu (tikai `walk[OID]` un `get[OID]` vienumiem);
- `snmp trapper` - SNMP trapu trapperis;
- `supervisor` - koordinē servera procesu un pavedienu palaišanu un apturēšanu, kā arī uzrauga to gatavību;
- `task manager` - process citu komponentu pieprasītu uzdevumu attālai izpildei (piemēram, problēmas aizvēršanai, problēmas apstiprināšanai, vienuma vērtības tūlītējai pārbaudei, attālās komandas funkcionalitātei);
- `telemetry query poller` - pollētājs telemetrijas vaicājumu vienumu pārbaudēm;
- `timer` - taimeris uzturēšanas periodu apstrādei;
- `trapper` - trapperis aktīvajām pārbaudēm, trapēm un saziņai ar starpniekserveri;
- `trigger housekeeper` - process trigeru ģenerēto problēmu un notikumu noņemšanai pēc šo trigeru dzēšanas;
- `unreachable poller` - pollētājs nesasniedzamām ierīcēm;

- `vmware collector` - VMware datu savācējs, kas atbild par datu apkopošanu no VMware pakalpojumiem.

Šos procesu veidus var novērot servera žurnālfailā.

Servera žurnālfails tiek izveidots ar lasīšanas un rakstīšanas atļaujām tikai faila īpašniekam. Turklāt failu var lasīt īpašnieka grupa. Visas pārējās atļaujas ir liegtas.

Dažādus Zabbix servera procesu veidus 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 triggers":

- A - trigeri, kas apstrādāti vēstures vērtību dēļ;
- B - trigeri, kas apstrādāti taimeru dēļ.

Laiki formātā `processed...in N (<timings>) sec` ir:

- Laiks, kas pavadīts vienumu vērtību ierakstīšanai datubāzē;
- Laiks, kas pavadīts vienumu datu atjaunināšanai (stāvoklis, kļūdas, hosta inventārs u. c.);
- Laiks, kas pavadīts tendenču iztukšošanai uz datubāzi;
- Laiks, kas pavadīts trigeru aprēķināšanai;
- Laiks, kas pavadīts notikumu un darbību apstrādei.

Housekeeping procedūra

`housekeeper` process periodiski noņem novecojušus datus (vienumu vēsturi un tendences, lietotāju sesijas, notikumus u. c.), kā arī datus, kas palikuši pēc dzēstiem objektiem. Tas darbojas ciklos, un biežumu, kā arī dzēšanas limitu vienā ciklā nosaka `HousekeepingFrequency` un `MaxHousekeeperDelete`. Jebkuri dati, kas netiek noņemti vienā ciklā, tiek pārcelti uz nākamo. Automātisko housekeeping var iespējot un konfigurēt sadaļā *Administration* > *Housekeeping*.

Kad `housekeeper` process noņem datus, kas palikuši pēc dzēstiem objektiem, tas paļaujas uz uzdevumiem no `housekeeper` tabulas, kas tiek aizpildīta ikreiz, kad objekts tiek dzēsts. Piemēram, dzēšot hostu, Zabbix dzēš tā vienumus, bet ne to vēsturi, tendences vai problēmas. Tā vietā datubāzes trigeri aizpilda `housekeeper` tabulu ar uzdevumiem, kas sastāv no šādiem laukiem:

- `housekeeperid` - uzdevuma ID
- `object` - objekta tips (0 - vienums; 1 - trigeris; 2 - pakalpojums; 3 - atklāts hosts; 4 - atklāts 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 ar uzdevumiem, nepārbaudot ar objektu saistītos datus; šo pārbaudi veic `housekeeper` process, kad tas noņem datus:

- Dzēstiem vienumiem (tostarp LLD noteikumiem) - noņem to datus no visām history un trends tabulām (`history`, `history_str`, `history_log`, `history_uint`, `history_text`, `history_bin`, `history_json`, `trends`, `trends_uint`). Tāpat noņem to iekšējos notikumus no `problem` tabulas.
- Dzēstiem trigeriem - noņem to notikumus no ar notikumiem saistītajām tabulām (`problem`, `event_symptom`, `events`; pati datubāze arī noņem notikumus no `event_recovery` tabulas). Tāpat paziņo `service manager` procesam par noņemtajiem notikumiem.

Note:
Atsevišķs `trigger housekeeper` process apstrādā šaurāku uzdevumu — noņem problēmas un notikumus, kuriem nav zināma avota trigera. Tā darbības biežumu nosaka `ProblemHousekeepingFrequency`.
Līdz brīdim, kad tiek palaista trigger housekeeping procedūra, problēmas, ko radījuši trigeri, kas kopš tā laika ir dzēsti, joprojām var ģenerēt pakalpojumu problēmas un piešķirt tās pakalpojumiem. Ja jūsu konfigurācijā ir daudz pakalpojumu statusa aprēķina noteikumu, kas balstīti uz bieži atklātiem/neatklātiem trigeriem, apsveriet iespēju palielināt housekeeping procedūras biežumu, pielāgojot servera konfigurācijas parametru `ProblemHousekeepingFrequency`.

- Dzēstiem pakalpojumiem - noņem to problēmas no `problem` tabulas; to notikumi no `events` tabulas tiek noņemti tikai pēc konfigurētā datu glabāšanas perioda.
- Dzēstiem atklātajiem hostiem/pakalpojumiem (network discovery) - noņem to notikumus no `events` tabulas.

`housekeeper` process noņem tikai tos notikumus, kas nav saistīti ar problēmām. Piemēram, novecojis problēmas/atjaunošanās notikums netiks noņemts, ja tas ir saistīts ar atvērtu problēmu. Kad `housekeeper` noņem novecojušus objektus, tas vispirms noņem problēmas, pēc tam notikumus.

Note:
Tabulas, kas izmanto `partition` režīmu (TimescaleDB sadalītās tabulas), tiek izlaistas; tiek apstrādātas tikai tabulas, kas izmanto `regular` režīmu.

Atbalstītās platformas Ņemot vērā drošības prasības un servera darbības kritiski svarīgo raksturu, UNIX ir vienīgā operētājsistēma, kas var konsekventi nodrošināt nepieciešamo veiktspēju, kļūdu toleranci un noturību. Zabbix darbojas uz tirgū vadošajām versijām.

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.

Lokāle Ņemiet vērā, ka serverim ir nepieciešama UTF-8 lokāle, lai dažus teksta vienumus varētu pareizi interpretēt. Lielākajai daļai mūsdienu Unix tipa sistēmu UTF-8 lokāle ir noklusējuma iestatījums, tomēr dažās sistēmās tā var būt jāiestata īpaši.

Zabbix serveru augstas pieejamības iestatīšana

Pārskats

Augsta pieejamība (HA) parasti ir nepieciešama kritiskās infrastruktūrās, kurās praktiski nav pieļaujama dīkstāve. Tāpēc jebkuram pakalpojumam, kas var atteikt, ir jābūt ieviestai pārslēgšanās uz rezerves sistēmu iespējai, lai pārņemtu darbību, ja pašreizējais pakalpojums neizdodas.

Zabbix piedāvā **vietējo** augstas pieejamības risinājumu, kuru ir viegli iestatīt un kuram nav nepieciešama iepriekšēja HA pieredze. Zabbix vietējā HA var būt noderīga kā papildu aizsardzības slānis pret Zabbix servera programmatūras/aparatūras kļūmēm vai lai samazinātu dīkstāvi apkopes dēļ.

Zabbix augstas pieejamības režīmā vairāki Zabbix serveri darbojas kā mezgli klasterī. Kamēr viens Zabbix serveris klasterī ir aktīvs, pārējie ir gaidstāves režīmā un gatavi nepieciešamības gadījumā pārņemt darbību.



Pāreja uz Zabbix HA nav saistoša. Jebkurā brīdī varat atgriezties pie atsevišķas darbības režīma.

Skatiet arī: Īstenošanas detaļas

Augstas pieejamības iespējošana

Zabbix servera palaišana kā klastera mezglam

Lai palaistu Zabbix serveri kā klastera mezglu, servera konfigurācijā ir nepieciešami divi parametri:

- Parametrs `HANodeName` ir jānorāda katram Zabbix serverim, kas būs HA klastera mezgls.

Tas ir unikāls mezgla identifikators (piemēram, `zabbix-node-01`), ar kuru serveris tiks identificēts aģenta un starpniekservera konfigurācijās. Ja nenorādīsiet `HANodeName`, serveris tiks palaists savrupajā režīmā.

- Parametrs `NodeAddress` ir jānorāda katram mezglam.

Parametru `NodeAddress` (`address:port`) Zabbix lietotāja saskarne izmantos, lai izveidotu savienojumu ar aktīvo servera mezglu. `NodeAddress` ir 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. Serveru jauno statusu var skatīt sadaļā *Reports* > *System information*, kā arī izpildot:

```
zabbix_server -R ha_status
```

Šī izpildlaika komanda ierakstīs pašreizējo HA klastera statusu Zabbix servera žurnālā (un standarta izvadē):

```
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 `address:port` nav definēts Zabbix lietotāja saskarnes konfigurācijas failā (`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 mezgla adrese tiks izmantota kā Zabbix servera adrese.

Starpniekservera konfigurācija

HA klastera mezgli (serveri) ir jānorāda vai nu pasīvā, vai aktīvā Zabbix starpniekservera konfigurācijā.

Pasīvam starpniekserverim mezglu nosaukumi ir jānorāda starpniekservera parametrā Server, atdalot tos ar **komatu**.

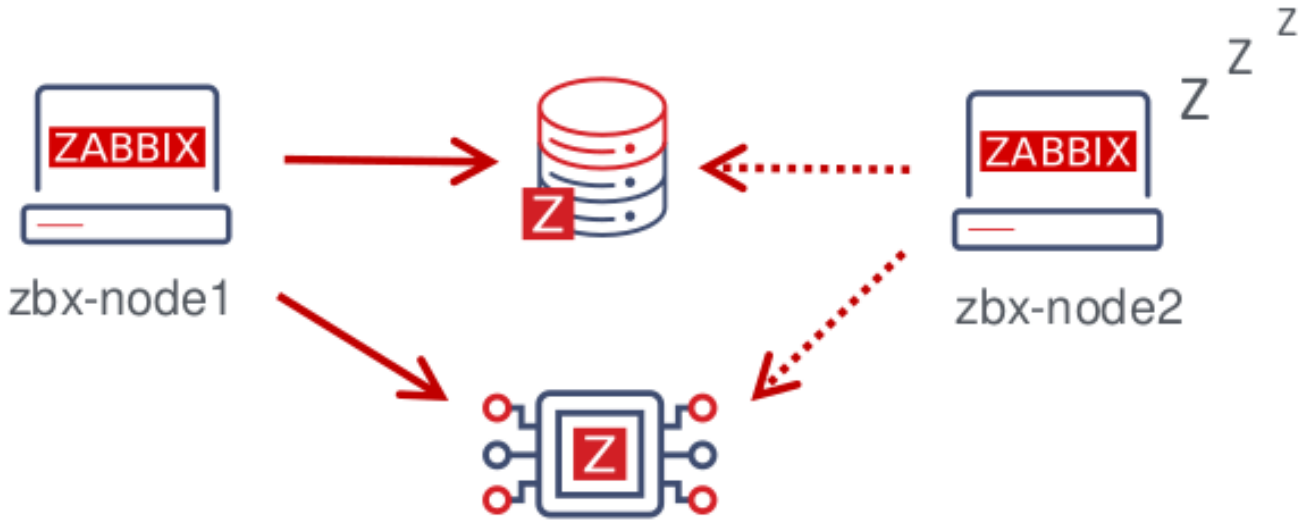
```
Server=zabbix-node-01,zabbix-node-02
```

Aktīvam starpniekserverim mezglu nosaukumi ir jānorāda starpniekservera parametrā Server, atdalot tos ar **semikolu**.

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

Aģenta konfigurācija

HA klastera mezgli (serveri) ir jānorāda Zabbix aģenta vai Zabbix aģenta 2 konfigurācijā.



Lai iespējotu pasīvos pārbaužu veidus, mezglu nosaukumi ir jānorāda Server parametrā, atdalot tos ar **komatu**.

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

Lai iespējotu aktīvos pārbaužu veidus, mezglu nosaukumi ir jānorāda ServerActive parametrā. Ņemiet vērā, ka aktīvajām pārbaudēm mezgliem no citiem serveriem jābūt atdalītiem ar komatu, savukārt pašiem mezgliem savā starpā jābūt atdalītiem ar **semikolu**, piemēram:

```
ServerActive=zabbix-node-01;zabbix-node-02
```

Pārslēgšanās uz gaidstāves mezglu

Zabbix automātiski pārslēgsies uz citu mezglu, ja aktīvais mezgls pārtrauks darbību. Lai pārslēgšanās notiktu, vismaz vienam mezglam jābūt gaidstāves statusā.

Cik ātra būs pārslēgšanās? Visi mezgli ik pēc 5 sekundēm atjaunina savu pēdējās piekļuves laiku (un statusu, ja tas ir mainījies). Tātad:

- Ja aktīvais mezgls tiek izslēgts un paspēj paziņot savu statusu kā "apturēts", cits mezgls pārņems vadību 5 sekunžu laikā.
- Ja aktīvais mezgls tiek izslēgts/kļūst nepieejams, nespējot atjaunināt savu statusu, gaidstāves mezgli gaidīs **pārslēgšanās aizkavi** + 5 sekundes, lai pārņemtu vadību.

Pārslēgšanās aizkave ir konfigurējama, atbalstītais diapazons ir no 10 sekundēm līdz 15 minūtēm (pēc noklusējuma viena minūte). Lai mainītu pārslēgšanās aizkavi, varat palaist:

```
zabbix_server -R ha_set_failover_delay=5m
```

HA klastera pārvaldība

HA klastera pašreizējo statusu var pārvaldīt, izmantojot īpašās izpildlaika vadības opcijas:

- `ha_status` - reģistrēt HA klastera statusu Zabbix servera žurnālā (un standarta izvadē);
- `ha_remove_node=target` - noņemt HA mezglu, kas identificēts pēc tā `<target>` - mezgla nosaukuma vai ID (nosaukumu/ID var iegūt no komandas `ha_status` 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, `30s`, `1m`)).

Mezgla statusu var pārraudzīt:

- Sadaļā *Pārskati* > *Sistēmas informācija*.
- Informācijas paneļa logrīkā *Sistēmas informācija*.
- Izmantojot servera izpildlaika vadības opciju `ha_status` (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:

1. Izveidojiet konfigurācijas failu rezerves kopijas.
2. Apturiet gaidstāves mezglus.
3. No aktīvā primārā servera noņemiet parametru HANodeName.
4. Pārstartējiet primāro serveri (tas tiks palaists patstāvīgā režīmā).

HA klastera jaunināšana

Lai veiktu HA mezglu galvenās versijas jaunināšanu:

1. Apturiet visus mezglus.
2. Izveidojiet pilnu datubāzes dublējumu.
3. Ja datubāze izmanto replikāciju, pārliecinieties, ka visi mezgli ir sinhronizēti un tiem nav problēmu. Neveiciet jaunināšanu, ja replikācija ir bojāta.
4. Atlasiet vienu mezglu, kas veiks datubāzes jaunināšanu, mainiet tā konfigurāciju uz atsevišķu režīmu, komentējot `HANodeName`, un jauniniet to.
5. Pārliecinieties, ka datubāzes jaunināšana ir pilnībā pabeigta (*Sistēmas informācijai* jāparāda, ka Zabbix serveris darbojas).
6. Restartējiet mezglu HA režīmā.
7. Jauniniet un palaidiet pārējos mezglus (nav nepieciešams tos pārslēgt uz atsevišķu režīmu, jo šajā brīdī datubāze jau ir jaunināta).

Nelielas versijas jaunināšanas gadījumā pietiek jaunināt pirmo mezglu, pārliecināties, ka tas ir jaunināts un darbojas, un pēc tam sākt jaunināšanu nākamajā mezglā.

Īstenošanas detaļas

Augstas pieejamības (HA) klasteris ir izvēles risinājums, un tas tiek atbalstīts Zabbix serverim. Dzimtais HA risinājums ir izstrādāts tā, lai to būtu vienkārši lietot, tas darbosies starp vietnēm un tam nav īpašu prasību attiecībā uz datubāzēm, ko Zabbix atpazīst. Lietotāji var brīvi izmantot Zabbix dzimto HA risinājumu vai trešās puses HA risinājumu atkarībā no tā, kas vislabāk atbilst augstas pieejamības prasībām viņu vidē.

Risinājums sastāv no vairākiem `zabbix_server` instancēm vai mezgliem. Katrs mezgls:

- Tiek konfigurēts atsevišķi.
- Izmanto to pašu datubāzi.
- Var darboties vairākos režīmos: aktīvs, gaidstāves, nepieejams, apturēts.

Vienlaikus aktīvs (strādājošs) var būt tikai viens mezgls. Gaidstāves mezgls palaiž tikai vienu procesu - HA pārvaldnieku. 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 ik pēc 5 sekundēm atjaunina savu pēdējās piekļuves laiku. Katrs gaidstāves mezgls uzrauga aktīvā mezgla pēdējās piekļuves laiku. Ja aktīvā mezgla pēdējās piekļuves laiks pārsniedz "failover delay" sekundes, gaidstāves mezgls pārslēdzas uz aktīva mezgla lomu un iepriekš aktīvajam mezglam piešķir statusu `unavailable`.

Aktīvais mezgls uzrauga savu datubāzes savienojamību - ja tā tiek zaudēta ilgāk nekā `failover delay`−5 sekundes, 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 Zabbix mazajām versijām.

Konfigurācijas parametri: Zabbix serveris

Pārskats

Konfigurācijas parametri ļauj pielāgot Zabbix servera darbību.

Norādiet parametru vērtības Zabbix servera konfigurācijas failā (`zabbix_server.conf`). Visi parametri ir **neobligāti**, ja vien nav skaidri norādīts, ka parametrs ir obligāts.

Šī lapa ir strukturēta šādi:

- Parametru pārskats (lai skatītu pilnu informāciju par parametru, noklikšķiniet uz tā nosaukuma)
- Parametru detalizēts apraksts

Parametrs	Apraksts
AlertScriptsPath	Norādiet brīdinājumu skriptu atrašanās vietu.
AllowRoot	Atļaujiet serverim darboties kā `root`.
AllowSoftwareUpdateCheck	Atļaujiet saņemt informāciju par programmatūras atjauninājumiem no `zabbix.com`.

Parametrs	Apraksts
AllowUnsupportedDBVersions	Atļaujiet serverim darboties ar neatbalstītām datubāzu versijām.
BridgeAdapterURL	Iestatiet Zabbix tilta adaptera HTTP(S) URL.
CacheSize	Iestatiet konfigurācijas kešatmiņas lielumu (baitos).
CacheUpdateFrequency	Iestatiet konfigurācijas kešatmiņas atjaunināšanas biežumu (sekundēs).
DBHost	Norādiet datubāzes hostu (vai ligzdas direktoriju).
DBName	Norādiet datubāzes nosaukumu. **Obligāts**.
DBPassword	Norādiet datubāzes paroli.
DBPort	Norādiet datubāzes hosta portu.
DBSchema	Norādiet datubāzes shēmas nosaukumu. Izmanto PostgreSQL.
DBSocket	Norādiet ceļu uz MySQL ligzdas failu.
DBUser	Norādiet datubāzes lietotāju.
DBTLSConnect	Norādiet vērtību, lai piespiestu TLS savienojumu ar datubāzi.
DBTLSCAFile	Norādiet pilnu ceļu uz failu, kurā ir augstākā līmeņa CA sertifikāti datubāzes sertifikāta pārbaudei.
DBTLSCertFile	Norādiet pilnu ceļu uz failu, kurā ir Zabbix servera sertifikāts autentifikācijai datubāzē.
DBTLSKeyFile	Norādiet pilnu ceļu uz failu, kurā ir privātā atslēga autentifikācijai datubāzē.
DBTLSCipher	Norādiet šifrēšanas algoritmu sarakstu, ko Zabbix serveris atļauj TLS protokoliem līdz TLS v1.2. Atbalstīts tikai MySQL.
DBTLSCipher13	Norādiet šifrēšanas algoritmu komplektu sarakstu, ko Zabbix serveris atļauj TLS v1.3 protokolam. Atbalstīts tikai MySQL, sākot ar versiju 8.0.16.
DebugLevel	Iestatiet atkļūdošanas līmeni.
EnableGlobalScripts	Iespējojiet globālos skriptus Zabbix serverī.
EnableMobileDevices	Iespējojiet mobilo ierīču atbalstu Zabbix serverī.
ExportDir	Norādiet direktoriju, lai iespējotu notikumu, vēstures un tendenču eksportēšanu reāllaikā.
ExportFileSize	Iestatiet maksimālo eksporta faila lielumu (baitos).
ExportType	Norādiet entītiju tipus eksportēšanai reāllaikā (notikumi, vēsture, tendences).
ExternalScripts	Norādiet ārējo skriptu atrašanās vietu.
Fping6Location	Norādiet fping6 atrašanās vietu.
FpingLocation	Norādiet fping atrašanās vietu.
FrontendAllowedIP	Norādiet IP adreses vai CIDR diapazonus, no kuriem atļauts izveidot savienojumu ar lietotāja saskarni.
HANodeName	Norādiet mezgla nosaukumu augstas pieejamības klasterī.
HistoryCacheSize	Iestatiet vēstures kešatmiņas lielumu.
HistoryIndexCacheSize	Iestatiet vēstures indeksu kešatmiņas lielumu.
HistoryProvider	Iestatiet vēstures glabāšanas nodrošinātāju (Elasticsearch, ClickHouse vai SQL).
HistoryStorageDateIndex	Šis parametrs ir **novecojis** (tā vietā izmantojiet HistoryProvider). Iespējojiet vēstures vērtību glabāšanu dažādos indeksos atkarībā no datuma.
HistoryStorageURL	Šis parametrs ir **novecojis** (tā vietā izmantojiet HistoryProvider). Norādiet vēstures glabāšanas URL.
HistoryStorageTypes	Šis parametrs ir **novecojis** (tā vietā izmantojiet HistoryProvider). Norādiet vērtību tipus, kas jānosūta uz vēstures glabātuvi.
HousekeepingFrequency	Iestatiet uzturēšanas procedūras biežumu (stundās).
Include	Norādiet direktoriju vai atsevišķus failus, kas jāiekļauj konfigurācijas failā.
JavaGateway	Norādiet Zabbix Java gateway IP adresi (vai resursdatora nosaukumu).
JavaGatewayPort	Norādiet Zabbix Java gateway portu.
ListenBacklog	Iestatiet maksimālo gaidošo savienojumu skaitu TCP rindā.
ListenIP	Norādiet trappera klausīšanās IP adresi.
ListenPort	Norādiet trappera klausīšanās portu.
LoadModule	Norādiet moduli, kas jāielādē servera palaišanas laikā.
LoadModulePath	Norādiet pilnu ceļu uz servera moduļu atrašanās vietu.
LogFile	Norādiet servera žurnāla failu.
LogFileSize	Iestatiet maksimālo žurnāla faila lielumu.
LogSlowQueries	Iestatiet maksimālo datubāzes vaicājuma izpildes ilgumu, pēc kura tas tiek ierakstīts žurnālā (milisekundēs).
LogType	Norādiet žurnāla izvades tipu.
MaxConcurrentChecksPerPoller	Iestatiet maksimālo asinhrono pārbaužu skaitu, ko vienlaikus var izpildīt katrs HTTP aģenta polleris, aģenta polleris vai SNMP polleris.
MaxHousekeeperDelete	Iestatiet maksimālo vienā uzturēšanas ciklā katram uzdevumam dzēšamo rindu skaitu.
NodeAddress	Norādiet IP adresi vai resursdatora nosaukumu ar neobligātu portu, lai aizstātu veidu, kā lietotāja saskarne savienojas ar serveri.
PidFile	Norādiet PID failu.
ProblemHousekeepingFrequency	Iestatiet dzēsto trigeru problēmu dzēšanas biežumu.

Parametrs	Apraksts
ProxyConfigFrequency	Iestatiet, cik bieži serveris nosūta konfigurācijas datus Zabbix starpniekserverim.
ProxyDataFrequency	Iestatiet, cik bieži serveris pieprasa vēstures datus no Zabbix starpniekservera.
ServiceManagerSyncFrequency	Iestatiet pakalpojumu pārvaldnieka konfigurācijas sinhronizēšanas biežumu.
SMSDevices	Norādiet modema failus, kurus atļauts izmantot Zabbix serverim.
SNMPTrapperFile	Norādiet pagaidu failu, kas tiek izmantots datu nodošanai no SNMP trapera dēmona serverim.
SocketDir	Norādiet direktoriju iekšējo Zabbix pakalpojumu izmantoto IPC ligzdu glabāšanai.
SourceIP	Norādiet avota IP adresi.
SSHKeyLocation	Norādiet publisko un privāto atslēgu atrašanās vietu SSH pārbaudēm un darbībām.
SSLCertLocation	Norādiet SSL klienta sertifikātu failu atrašanās vietu klienta autentifikācijai.
SSLKeyLocation	Norādiet SSL privāto atslēgu failu atrašanās vietu klienta autentifikācijai.
SSLCALocation	Norādiet sertifikācijas iestādes (CA) failu atrašanās vietu SSL servera sertifikāta pārbaudei.
StartAgentPollers	Iestatiet asinhrono Zabbix aģenta polleru iepriekš izveidoto instanču skaitu.
StartAlerters	Iestatiet alerteru iepriekš izveidoto instanču skaitu.
StartBrowserPollers	Iestatiet pārlūka vienumu polleru iepriekš izveidoto instanču skaitu.
StartConnectors	Iestatiet savienotāju darbplūsmu iepriekš izveidoto instanču skaitu.
StartDBSyncers	Iestatiet vēstures sinhronizētāju iepriekš izveidoto instanču skaitu.
StartDiscoverers	Iestatiet atklāšanas darbplūsmu iepriekš izveidoto instanču skaitu.
StartEscalators	Iestatiet eskalatoru iepriekš izveidoto instanču skaitu.
StartHistoryPollers	Iestatiet vēstures polleru iepriekš izveidoto instanču skaitu.
StartHTTPAgentPollers	Iestatiet asinhrono HTTP aģenta polleru iepriekš izveidoto instanču skaitu.
StartHTTPPollers	Iestatiet HTTP polleru iepriekš izveidoto instanču skaitu.
StartIPMIPollers	Iestatiet IPMI polleru iepriekš izveidoto instanču skaitu.
StartJavaPollers	Iestatiet Java polleru iepriekš izveidoto instanču skaitu.
StartLLDProcessors	Iestatiet zema līmeņa atklāšanas (LLD) darbplūsmu iepriekš izveidoto instanču skaitu.
StartODBCPollers	Iestatiet ODBC polleru iepriekš izveidoto instanču skaitu.
StartPingers	Iestatiet ICMP pingeru iepriekš izveidoto instanču skaitu.
StartPollersUnreachable	Iestatiet nepieejamu hostu polleru iepriekš izveidoto instanču skaitu (tostarp IPMI un Java).
StartPollers	Iestatiet polleru iepriekš izveidoto instanču skaitu.
StartPreprocessors	Iestatiet iepriekš palaisto priekšapstrādes darbplūsmu instanču skaitu.
StartProxyPollers	Iestatiet pasīvo starpniekserveru polleru iepriekš izveidoto instanču skaitu.
StartReportWriters	Iestatiet pārskatu rakstītāju iepriekš izveidoto instanču skaitu.
StartSNMPPollers	Iestatiet asinhrono SNMP polleru iepriekš izveidoto instanču skaitu.
StartSNMPTrapper	Iestatiet vērtību ”1”, lai palaistu SNMP trappera procesu.
StartTelemetryQueryPollers	Iestatiet asinhrono telemetrijas vaicājumu polleru iepriekš izveidoto instanču skaitu.
StartTimers	Iestatiet taimeru iepriekš izveidoto instanču skaitu.
StartTrappers	Iestatiet trapperu iepriekš izveidoto instanču skaitu.
StartVMwareCollectors	Iestatiet iepriekš izveidoto VMware kolektoru instanču skaitu.
StatsAllowedIP	Norādiet ārējo Zabbix instanču IP adreses vai DNS nosaukumus, kurām atļauts pieprasīt statistiku.
TelemetryProvider	Iestatiet telemetrijas datu avotu.
Timeout	Norādiet, cik ilgi (sekundēs) gaidīt savienojuma izveidi un datu apmaiņu ar Zabbix starpniekserveri, aģentu, tīmekļa pakalpojumu un SNMP pārbaužu veikšanu (izņemot SNMP `walk[OID]` un `get[OID]` vienumus).
TLSCAFile	Norādiet pilnu ceļu uz failu, kurā ir augstākā līmeņa CA sertifikāti partnera sertifikāta pārbaudei; to izmanto šifrētai saziņai starp Zabbix komponentiem.
TLSCertFile	Norādiet pilnu ceļu uz failu, kurā ir servera sertifikāts vai sertifikātu ķēde; to izmanto šifrētai saziņai starp Zabbix komponentiem.
TLSCipherAll	Norādiet GnuTLS prioritāšu virkni vai OpenSSL (TLS 1.2) šifru virkni. Ignorējiet noklusējuma šifru komplekta atlases kritērijus sertifikātu un PSK šifrēšanai.
TLSCipherAll13	Norādiet šifru virkni OpenSSL 1.1.1 vai jaunākai versijai TLS 1.3 režīmā. Ignorējiet noklusējuma šifru komplekta atlases kritērijus sertifikātu un PSK šifrēšanai.
TLSCipherCert	Norādiet GnuTLS prioritāšu virkni vai OpenSSL (TLS 1.2) šifru virkni. Ignorējiet noklusējuma šifru komplekta atlases kritērijus sertifikātu šifrēšanai.
TLSCipherCert13	Norādiet šifru virkni OpenSSL 1.1.1 vai jaunākai versijai TLS 1.3 režīmā. Ignorējiet noklusējuma šifru komplekta atlases kritērijus sertifikātu šifrēšanai.
TLSCipherPSK	Norādiet GnuTLS prioritāšu virkni vai OpenSSL (TLS 1.2) šifru virkni. Ignorējiet noklusējuma šifru komplekta atlases kritērijus PSK šifrēšanai.
TLSCipherPSK13	Norādiet šifru virkni OpenSSL 1.1.1 vai jaunākai versijai TLS 1.3 režīmā. Ignorējiet noklusējuma šifru komplekta atlases kritērijus PSK šifrēšanai.
TLSCRLFile	Norādiet pilnu ceļu uz failu, kurā ir atsauktie sertifikāti. Šo parametru izmanto šifrētai saziņai starp Zabbix komponentiem.

Parametrs	Apraksts
TLSFrontendAccept	Norādiet ienākošo savienojumu no lietotāja saskarnes šifrēšanas līmeni.
TLSFrontendCertIssuer	Norādiet atļauto lietotāja saskarnes sertifikāta izdevēju.
TLSFrontendCertSubject	Norādiet atļauto lietotāja saskarnes sertifikāta subjektu.
TLSKeyFile	Norādiet pilnu ceļu uz failu, kurā ir servera privātā atslēga; to izmanto šifrētai saziņai starp Zabbix komponentiem.
TLSListen	Pieprasiet tikai TLS savienojumus trappera ligzdā.
TmpDir	Norādiet pagaidu direktoriju.
TrapperTimeout	Norādiet trappera taimautu sekundēs šādām darbībām: - vēstures datu izgūšanai no Zabbix starpniekservera; - konfigurācijas datu nosūtīšanai Zabbix starpniekserverim; - globālā skripta izpildei vai attālās komandas izpildei Zabbix serverī.
TrendCacheSize	Iestatiet tendenču kešatmiņas lielumu.
TrendFunctionCacheSize	Iestatiet tendenču funkciju kešatmiņas lielumu.
UnavailableDelay	Iestatiet hosta pieejamības pārbaudes biežumu nepieejamības periodā.
UnreachableDelay	Iestatiet hosta pieejamības pārbaudes biežumu nesasniedzamības periodā.
UnreachablePeriod	Iestatiet, pēc cik nesasniedzamības sekundēm hosts jāuzskata par nepieejamu.
User	Atceliet privilēģijas un pārslēdzieties uz konkrētu, sistēmā esošu lietotāju.
ValueCacheSize	Iestatiet vēstures vērtību kešatmiņas lielumu.
Vault	Norādiet vault nodrošinātāju.
VaultAppRoleID	Norādiet HashiCorp vault AppRole ID.
VaultAppSecretID	Norādiet HashiCorp vault AppRole slepeno ID.
VaultDBPath	Norādiet atrašanās vietu, no kuras pēc atslēgām izgūt datubāzes akreditācijas datus.
VaultPrefix	Norādiet pielāgotu prefiksu vault ceļam vai vaicājumam.
VaultTLSCertFile	Norādiet klienta autentifikācijai izmantotā SSL sertifikāta faila nosaukumu.
VaultTLSKeyFile	Norādiet klienta autentifikācijai izmantotā SSL privātās atslēgas faila nosaukumu.
VaultToken	Norādiet HashiCorp vault autentifikācijas pilnvaru.
VaultURL	Norādiet vault servera URL.
VMwareCacheSize	Iestatiet koplietojamās atmiņas lielumu VMware datu glabāšanai.
VMwareFrequency	Iestatiet datu apkopošanas biežumu no viena VMware pakalpojuma sekundēs.
VMwarePerfFrequency	Iestatiet veiktspējas skaitītāju statistikas izgūšanas biežumu no viena VMware pakalpojuma sekundēs.
VMwareTimeout	Iestatiet maksimālo sekunžu skaitu, ko `vmware collector` gaidīs atbildi no VMware pakalpojuma.
WebDriverURL	Norādiet WebDriver saskarnes URL.
WebServiceURL	Norādiet tīmekļa pakalpojuma URL formātā `http[s]://host:port/report`.

Ņemiet vērā:

- Noklusējuma vērtības atspoguļo dēmona 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.

Warning:
Pārāk liels procesu skaits var izsmelt priekšapstrādes pārvaldniekam pieejamo failu deskriptoru limitu.

Šādi procesi ir datu apkopošana, ko veic polleri, nesasniedzamu hostu polleri, ODBC polleri, HTTP polleri, Java polleri, pingeri, trapperi un starpniekserveru polleri, kā arī IPMI pārvaldnieks, SNMP trapperis, priekšapstrādes darbplūsmas un atklāšanas darbplūsmas. Failu deskriptoru limita izsmelšana izraisīs Zabbix servera **apturēšanu**, parasti neilgi pēc palaišanas, taču dažkārt tas var aizņemt vairāk laika.

Lai izvairītos no šādām problēmām, pārskatiet Zabbix servera konfigurācijas failu, lai optimizētu vienlaicīgo pārbaužu un procesu skaitu. Turklāt, ja nepieciešams, pārbaudiet un pielāgojiet sistēmas limitus, lai nodrošinātu pietiekami augstu failu deskriptoru limitu.

Parametra informācija

AlertScriptsPath

Norādiet brīdinājuma skripta atrašanās vietu. Atkarīgs no `datadir` kompilēšanas laikā iestatītā instalācijas mainīgā.

Noklusējums: `/usr/local/share/zabbix/alertscripts`

AllowRoot

Atļaut serverim darboties kā `root`. Ja šī opcija ir atspējota un serveris tiek palaists ar `root` lietotāju, serveris mēģinās pārslēgties uz `zabbix` lietotāju. Nav ietekmes, ja tas tiek 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

Atļaut neatbalstītas DB versijas

Atļaut serverim darboties ar neatbalstītām datubāzes versijām.

Noklusējums: 0
 Vērtības: 0 - neatļaut; 1 - atļaut

BridgeAdapterURL

Iestatiet Zabbix tilta adaptera HTTP(S) URL.

Piemērs:

```
BridgeAdapterURL=https://bridge-adapter.example.com:10054/rpc
```

CacheSize

Iestatiet konfigurācijas kešatmiņas lielumu (baitos). Koplietotās atmiņas apjoms, kas tiek izmantots hosts, vienumu un trigeru datu glabāšanai.

Noklusējums: 32M
 Diapazons: 128K-64G

CacheUpdateFrequency

Iestatiet konfigurācijas kešatmiņas atjaunināšanas biežumu (sekundēs). Skatiet arī izpildlaika vadības opcijas.

Noklusējums: 10
 Diapazons: 1-3600

DBHost

Norādiet datubāzes hostu (vai ligzdas direktoriju).

MySQL gadījumā:

- `localhost` vai tukša virkne izmanto noklusējuma UNIX domēna ligzdu.

PostgreSQL gadījumā:

- `localhost` tiek atrisināts, izmantojot DNS (parasti uz 127.0.0.1).
- Tukša virkne izmanto noklusējuma UNIX domēna ligzdu.
- Ceļš (piemēram, `/var/run/pgbouncer`) nozīmē UNIX ligzdas izmantošanu šajā ceļā.
- Ar komatiem atdalīts saraksts var saturēt vairākas `host:port` vērtības. Katrs hosts tiek mēģināts pēc kārtas, līdz tiek izveidots lasīšanas-rakstīšanas savienojums.

Piemērs: `DBHost=localhost:5431,127.0.0.1:20051,zabbix.domain,[::1]:30051,[12fc::1]`

Noklusējums: `localhost`

DBName

Norādiet datubāzes nosaukumu.

Obligāts: **Jā**

DBPassword

Norādiet datubāzes paroli. Ja parole netiek izmantota, komentējiet šo rindu.

DBPort

Norādiet Zabbix datubāzes hosta portu (skatiet `DBHost`).
 `DBSocket` un `DBPort` servera konfigurācijā ir savstarpēji izslēdzoši. Norādiet tikai vienu no tiem vai atstājiet abus nedefinētus.

Noklusējums MySQL: 3306

Noklusējums PostgreSQL: 5432

Diapazons: 1024-65535

DBSchema

Norādiet datubāzes shēmas nosaukumu. Izmanto PostgreSQL.

DBSocket

Norādiet MySQL socket faila ceļu.
 `DBSocket` un `DBPort` servera konfigurācijā ir savstarpēji izslēdzoši. Norādiet tikai vienu vai atstājiet abus nedefinētus.

DBUser

Norādiet datubāzes lietotāju.

DBTLSConnect

Norādiet vērtību, lai piespiestu 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, ka datubāzes identitāte, kas norādīta ar `DBHost`, atbilst tās sertifikātam.

MySQL gadījumā, sākot no 5.7.11, un PostgreSQL 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

Norādiet pilnu ceļu uz failu, kurā ir datubāzes sertifikātu verifikācijai paredzētie augstākā līmeņa CA sertifikāti.

Obligāti: nē (jā, ja DBTLSConnect ir iestatīts uz `verify_ca` vai `verify_full`)

DBTLSCertFile

Norādiet pilnu ceļu uz failu, kurā atrodas Zabbix servera sertifikāts autentifikācijai datubāzē.

DBTLSKeyFile

Norādiet pilnu faila ceļu uz failu, kas satur privāto atslēgu autentifikācijai datubāzē.

DBTLSCipher

Norādiet šifrēšanas algoritmu sarakstu, ko Zabbix serveris atļauj TLS protokoliem līdz TLS v1.2 ieskaitot. Atbalstīts tikai MySQL.

DBTLSCipher13

Norādiet šifrēšanas ciphersuite sarakstu, ko Zabbix serveris atļauj TLS v1.3 protokolam. Atbalstīts tikai MySQL, sākot no versijas 8.0.16.

DebugLevel

Iestatiet 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, начиная ar Zabbix 7.0, EnableGlobalScripts ir skaidri iestatīts uz 0 (atspējots).

Noklusējums: 1
 Vērtības: 0 - atspējot; 1 - iespējot

EnableMobileDevices

Iespējot mobilo ierīču atbalstu Zabbix serverī.

Ja šī opcija ir iespējota, Zabbix serveris apstrādās pieprasījumus mobilo ierīču pievienošanai vai noņemšanai un nosūtīs push paziņojumus uz šīm ierīcēm.

Noklusējuma vērtība: 1
 Vērtības: 0 - atspējot; 1 - iespējot

ExportDir

Norādiet direktoriju, lai iespējotu notikumu, vēstures un tendenču reāllaika eksportu.

ExportFileSize

Iestatiet maksimālo eksportējamā faila lielumu (baitos). Tiek izmantots rotācijai, ja ir iestatīts `ExportDir`.

Noklusējums: 1G
 Diapazons: 1M-1G

ExportType

Norādiet entītiju tipus reāllaika eksportam (notikumi, vēsture, tendences). Ar komatiem atdalīts saraksts. Pēc noklusējuma tiek eksportēti visi tipi. Derīgs tikai tad, ja ir iestatīts ExportDir.
 Piezīme: ja ir norādīts ExportType, bet ExportDir nav norādīts, tā ir konfigurācijas kļūda, un serveris netiks palaists.

Piemērs vēstures un tendenču eksportam:

```
ExportType=history,trends
```

Piemērs tikai notikumu eksportam:

```
ExportType=events
```

Ārējie skripti

Norādiet ārējo skriptu atrašanās vietu. Atkarīgs no kompilēšanas laikā iestatītā instalācijas mainīgā datadir.

Noklusējums: /usr/local/share/zabbix/externalscripts

Fping6Location

Norādiet fping6 atrašanās vietu. Pārliecinieties, ka fping6 binārajam failam ir root īpašumtiesības un ir iestatīts SUID karodziņš. Atstājiet tukšu (Fping6Location=), ja jūsu fping utilīta spēj apstrādāt IPv6 adreses.

Noklusējums: /usr/sbin/fping6

FpingLocation

Norādiet fping atrašanās vietu. Pārliecinieties, ka fping binārajam failam ir root īpašumtiesības un iestatīts SUID karodziņš.

Noklusējums: /usr/sbin/fping

FrontendAllowedIP

Norādiet IP adreses vai CIDR diapazonus, kuriem ir atļauts izveidot savienojumu no lietotāja saskarnes. Ar komatiem atdalīts saraksts. Pēc noklusējuma visi savienojumi tiek pieņemti lietotāja saskarnes pieprasījumiem. Ja ir iespējots IPv6 atbalsts, tad 127.0.0.1, ::127.0.0.1, ::ffff:127.0.0.1 tiek uzskatīti par ekvivalentiem, 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

Norādiet mezgla nosaukumu augstas pieejamības klasterī. Ja lauks ir tukšs, serveris darbojas atsevišķā režīmā un tiek izveidots mezgls ar tukšu nosaukumu.

HistoryCacheSize

Iestatiet vēstures kešatmiņas lielumu (baitos). Koplietotās atmiņas apjoms vēstures datu glabāšanai.

Noklusējums: 16M
 Diapazons: 128K-16G

HistoryIndexCacheSize

Iestatiet history-index kešatmiņas lielumu (baitos). Koplietotās atmiņas apjoms vēstures datu indeksēšanai, kas tiek glabāti history kešatmiņā. Indeksa kešatmiņas lielumam vienas vienības kešošanai nepieciešami aptuveni 100 baiti.

Noklusējums: 4M
 Diapazons: 128K-16G

HistoryProvider

Iestatiet vēstures glabāšanas nodrošinātāju, izmantojot formātu <provider>[;<options>], kur:

- <provider> - viens no atbalstītajiem vēstures glabāšanas nodrošinātājiem: sql (noklusējums), elasticsearch vai clickhouse.
- <options> - ar komatiem atdalīti key=value pāri. Vērtības var būt pēdiņās.

Opcijas visiem vēstures glabāšanas nodrošinātājiem:

- value_types - iestata glabājamo vērtību tipus (uint, dbl, str, log, text, json). Katru var izmantot vienu reizi un tikai vienam nodrošinātājam.
- precache - iestata 1 (noklusējums), lai masveidā ielādētu datus no vēstures glabāšanas nodrošinātāja vērtību kešatmiņā. Tas var būtiski uzlabot vērtību kešatmiņas uzsildīšanu pēc restartēšanas.

Opcijas Elasticsearch un ClickHouse:

- `url` - iestata HTTP vai HTTPS URL, ko izmanto savienojuma izveidei ar vēstures glabāšanas nodrošinātāju.
- `log_slow_queries` - iestata maksimālo ilgumu (milisekundēs), cik ilgi vēstures glabāšanas nodrošinātāja vaicājums drīkst darboties, pirms tas tiek reģistrēts žurnālā. Šī opcija kļūst pieejama, sākot ar `DebugLevel=3`.
- `source_ip` - iestata avota IP adresi, ko izmanto izejošajiem savienojumiem ar vēstures glabāšanas nodrošinātāju.

Opcijas Elasticsearch:

- `date_index` - iestata 1, lai iespējotu vēstures vērtību glabāšanu dažādos indeksos, pamatojoties uz datumu.

Opcijas ClickHouse:

- `db` - iestata datubāzes nosaukumu.
- `username` - iestata lietotājvārdu, ko izmanto pieteikšanās datubāzē.
- `password` - iestata paroli, ko izmanto pieteikšanās datubāzē.
- `ssl_cert_file`, `ssl_key_file`, `ssl_key_password`, `ssl_verify_peer`, `ssl_verify_host`, `ssl_ca_location`, `ssl_cert_location`, `ssl_key_location` - iestatiet šīs opcijas, lai konfigurētu drošu (TLS/SSL) savienojumu ar datubāzi.

ClickHouse piemērs:

```
HistoryProvider=clickhouse;value_types="uint,dbl,str,log,text,json",url=http://localhost:8123,db=zabbix,us
```

Elasticsearch piemērs:

```
HistoryProvider=elasticsearch;value_types="uint,dbl,str,log,text,json",url=http://localhost:9200,date_inde
```

ClickHouse un Elasticsearch piemērs:

```
HistoryProvider=clickhouse;value_types="uint,dbl,str",url=http://localhost:8123,db=zabbix,username=zabbix,
HistoryProvider=elasticsearch;value_types="log,text,json",url=http://localhost:9200,date_index=0
```

HistoryStorageDateIndex

Šis parametrs ir novecojis (tā vietā izmantojiet HistoryProvider).
 Iespējot vērtību vēstures glabāšanu dažādos indeksos, pamatojoties uz datumu. Šis parametrs tiek izmantots Elasticsearch konfigurācijā.

Noklusējums: 0
 Vērtības: 0 - atspējot; 1 - iespējot

HistoryStorageURL

Šis parametrs ir novecojis (tā vietā izmantojiet HistoryProvider).
 Norādiet vēstures glabāšanas URL. Šis parametrs tiek izmantots Elasticsearch iestatīšanai.

HistoryStorageTypes

Šis parametrs ir novecojis (tā vietā izmantojiet HistoryProvider).
 Norādiet vērtību tipus, kas jānosūta uz vēstures krātuvi. Komatiem atdalīts saraksts. Šis parametrs tiek izmantots Elasticsearch iestatīšanai.

Noklusējums: `uint,dbl,str,log,text,json`

HousekeepingFrequency

Iestata housekeeper procedūras biežumu (stundās). Housekeeper no datubāzes noņem novecojušu informāciju.
Piezīme: Lai novērstu housekeeper pārslodzi (piemēram, ja vēstures un tendenču periodi ir būtiski samazināti), vienā housekeeper ciklā katram vienumam tiek dzēsts ne vairāk kā 4 reizes `HousekeepingFrequency` stundu novecojušas informācijas. 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 palaišanas laikā, housekeeper darbība tiek atlikta uz 30 minūtēm pēc servera starta. Tādējādi, ja `HousekeepingFrequency` ir 1, pati pirmā housekeeper procedūra pēc servera starta tiks palaista pēc 30 minūtēm un pēc tam atkārtosies ar vienas stundas intervālu.
Automātisko housekeeper var atspējot, iestatot `HousekeepingFrequency` uz 0. Šādā gadījumā housekeeper procedūru var palaist tikai ar `housekeeper_execute` izpildlaika vadības opciju, un vienā housekeeper ciklā dzēstais novecojušas informācijas periods ir 4 reizes lielāks par periodu kopš pēdējā housekeeper cikla, bet ne mazāks par 4 stundām un ne lielāks par 4 dienām.
Skatiet arī izpildlaika vadības opcijas un informāciju par housekeeper procedūru.

Noklusējums: 1
 Diapazons: 0-24

Iekļaut

Norādiet direktoriju vai atsevišķus failus, kas jāiekļauj konfigurācijas failā.
 Zabbix instalēšanas laikā šis parametrs tiek iestatīts uz `/usr/local/etc`, ja vien jūs to nemaināt kompilēšanas laikā.

Vispārīgie noteikumi parametram `Include`:

- Ceļš var būt relatīvs attiecībā pret `zabbix_server.conf` faila atrašanās vietu.
- Vienā konfigurācijas failā var norādīt vairākus `Include` parametrus.
- Zabbix apstrādā `Include` parametru uzreiz pēc tā nolasīšanas, tāpēc ieteicams to novietot konfigurācijas faila beigās.

Iekļauto failu noteikumi:

- Katram failam jābūt lasāmam.
- Katram failam jābūt ar pareizu sintaksi. Pretējā gadījumā komponents netiks palaists.

Direktoriju vai failu rakstu iekļaušanas noteikumi:

- Nepieņemiet, ka iekļaušana notiek noteiktā secībā (piemēram, faili netiek iekļauti alfabētiskā secībā).
- Nenorādiet vienu un to pašu parametru vairāk nekā vienā iekļautajā failā (piemēram, lai pārrakstītu vispārīgu iestatījumu ar specifisku), jo nevar paredzēt, kura vērtība tiks izmantota.
- Uzmanieties no rezerves kopijām, ko daži teksta redaktori izveido automātiski (piemēram, rediģējot `include/my_specific.conf`, var tikt izveidots `include/my_specific.conf.BAK`), jo abi faili tiks iekļauti.

Lai no direktorija iekļautu tikai atbilstošos failus, izmantojiet aizstājējzīmi (*) raksta atbilstībai.

Piemērs:

```
Include=/absolute/path/to/config/files/*.conf
```

JavaGateway

Norādiet Zabbix Java gateway IP adresi vai resursdatora nosaukumu. Nepieciešams tikai tad, ja ir palaisti Java aptaujātāji.

JavaGatewayPort

Norādiet Zabbix Java gateway portu.

Noklusējums: 10052
 Diapazons: 1024-32767

ListenBacklog

Iestata maksimālo TCP rindā gaidošo savienojumu skaitu.
Noklusējuma vērtība ir cieti kodēta konstante, kas ir atkarīga no sistēmas.
Arī maksimālā atbalstītā vērtība ir atkarīga no sistēmas; pārāk lielas vērtības var tikt klusi saīsinātas līdz "implementation-specified maximum".

Noklusējums: SOMAXCONN
 Diapazons: 0 - INT_MAX

ListenIP

Norādiet trapper klausīšanās IP adresi. Ar komatiem atdalīts saraksts.
 Trapper klausīsies uz visām tīkla saskarnēm, ja šis parametrs nav norādīts.

Noklusējums: 0.0.0.0

ListenPort

Norādiet trapper klausīšanās portu.

Noklusējums: 10051
 Diapazons: 1024-32767

LoadModule

Norādiet moduli, kas jāielādē servera palaišanas laikā.
Moduļi tiek izmantoti, lai paplašinātu servera funkcionalitāti.
Modulim jāatrodas direktorijā, kas norādīta ar LoadModulePath, vai arī ceļam jābūt norādītam pirms moduļa nosaukuma.
Ja norādītais ceļš ir absolūts (sākas ar /), LoadModulePath tiek ignorēts.
 Formāti:

- LoadModule=<module.so>
- LoadModule=<path/module.so>
- LoadModule=</abs_path/module.so>

Var norādīt vairākus LoadModule parametrus.

LoadModulePath

Norādiet pilnu ceļu uz servera moduļu atrašanās vietu. Noklusējums ir atkarīgs no kompilēšanas opcijām.

LogFile

Norādiet servera žurnāla failu.

Obligāts: Jā, ja LogType ir iestatīts uz file; citādi nē

LogFileSize

Iestatiet maksimālo žurnāla faila lielumu (MB).
0 - atspējot automātisku žurnāla rotāciju.
Piezīme: Ja tiek sasniegts žurnāla faila lieluma ierobežojums un faila rotācija kāda iemesla dēļ neizdodas, esošais žurnāla fails tiek saīsināts un izveidots no jauna.

Noklusējums: 1
 Diapazons: 0-1024
 Obligāts: Jā, ja LogType ir iestatīts uz file; citos gadījumos nē

LogSlowQueries

Iestatiet maksimālo ilgumu, kādu datubāzes vaicājums drīkst aizņemt, pirms tas tiek reģistrēts žurnālā (milisekundēs).
0 - nereģistrēt lēnos vaicājumus.
Šī opcija kļūst pieejama, sākot ar DebugLevel=3.

Noklusējums: 0
 Diapazons: 0-3600000

LogType

Norādiet žurnāla izvades tipu: - *file* - ierakstīt žurnālu failā, kas norādīts ar `LogFile` parametru; - *system* - ierakstīt žurnālu syslog; - *console* - ierakstīt žurnālu standarta izvadei.

Noklusējums: `file`

MaxConcurrentChecksPerPoller

Iestatiet maksimālo asinhrono pārbaužu skaitu, ko katrs HTTP aģenta aptaujātājs, aģenta aptaujātājs vai SNMP aptaujātājs var izpildīt vienlaikus. Skatiet StartHTTPAgentPollers, StartAgentPollers un StartSNMPPollers.

Noklusējums: 1000
 Diapazons: 1-1000

MaxHousekeeperDelete

Iestata maksimālo rindu skaitu, kas vienā apkopes ciklā tiks dzēsts katram uzdevumam no `history`, `history_str`, `history_log`, `history_uint`, `history_text`, `history_bin`, `history_json`, `trends`, `trends_uint` un `problem` tabulām.
Apkopes uzdevumi atrodas `housekeeper` tabulā šādā formātā: `[housekeeperid]`, `[object]`, `[objectid]`.
 Šis parametrs attiecas tikai uz datu dzēšanu, kas palikuši pāri pēc jau dzēstiem vienumiem.
 Ja iestatīts 0, tad netiek izmantots nekāds ierobežojums.
 Šādā gadījumā jums precīzi jāzina, ko darāt, lai nepārslogotu datubāzi.
 Ja tiek dzēsts liels skaits vienumu, tas palielina slodzi datubāzei, jo housekeeper būs jānoņem visi šo vienumu vēsturiskie dati.
 Piemēram, ja no veidnes jānoņem tikai 1 vienuma prototips, bet šī veidne ir saistīta ar 50 hosts un katram hostam prototips tiek izvērsts 100 reālos vienumos, kopā jānoņem 5000 vienumi (1*50*100).
 Ja `MaxHousekeeperDelete` ir iestatīts uz 500 (`MaxHousekeeperDelete=500`), housekeeper process vienā ciklā no vēstures un tendenču tabulām būs jānoņem līdz pat 2500000 vērtībām (5000*500) dzēstajiem vienumiem.

Skatiet arī detalizētu informāciju par apkopes procedūru.

Noklusējums: 5000
 Diapazons: 0-1000000

NodeAddress

Norādiet IP adresi vai resursdatora nosaukumu ar izvēles portu, lai pārrakstītu, kā lietotāja saskarne savienojas ar serveri.
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 pārrakstīt ar adresi, kas norādīta lietotāja saskarnes konfigurācijā.

Skatiet arī: HANodeName parametru; Augstas pieejamības iespējošana.

Noklusējums: `localhost:10051`

PidFile

Norādiet PID failu.

Noklusējums: `/tmp/zabbix_server.pid`

ProblemHousekeepingFrequency

Iestata problēmu dzēšanas biežumu dzēstajiem trigeriem (sekundēs).
 Skatiet arī informāciju par housekeeping procedūru.

Noklusējums: 60
 Diapazons: 1-3600

ProxyConfigFrequency

Iestata konfigurācijas datu nosūtīšanas biežumu uz Zabbix starpniekserveri no servera. Tiek izmantots tikai starpniekserveriem pasīvajā režīmā.

Noklusējums: 10
 Diapazons: 1-604800

ProxyDataFrequency

Iestata servera pieprasījumu biežumu pēc vēstures datiem no Zabbix starpniekservera. Tiek izmantots tikai starpniekserveriem pasīvajā režīmā.

Noklusējums: 1
 Diapazons: 1-3600

ServiceManagerSyncFrequency

Iestatiet pakalpojumu pārvaldnieka konfigurācijas sinhronizācijas biežumu (sekundēs).

Noklusējums: 60
 Diapazons: 1-3600

SMSDevices

Norādiet modema failus, kuriem ir atļauts izmantot Zabbix serveris. Ar komatiem atdalīts saraksts.
 SMS sūtīšana nav iespējama, ja šis parametrs nav iestatīts.

Piemērs:

```
SMSDevices=/dev/ttyUSB0,/dev/ttyUSB1
```

SNMPTrapperFile

Norādiet pagaidu failu, ko izmanto datu nodošanai no SNMP trap dēmona uz serveri.
Tam jābūt tādam pašam kā `zabbix_trap_receiver.pl` vai SNMPTT konfigurācijas failā.

Noklusējums: `/tmp/zabbix_traps.tmp`

SocketDir

Norādiet direktoriju IPC ligzdu glabāšanai, ko izmanto iekšējie Zabbix pakalpojumi.

Noklusējums: `/tmp`

SourceIP

Norādiet avota IP adresi šādiem savienojumiem:

- izejošie savienojumi ar Zabbix starpniekserveri un Zabbix aģentu
- savienojumi bez aģenta (VMware, SSH, JMX, SNMP, Telnet un vienkāršās pārbaudes)
- HTTP aģenta savienojumi
- skripta vienuma JavaScript HTTP pieprasījumi
- pirmapstrādes JavaScript HTTP pieprasījumi
- paziņojumu e-pasta sūtīšana (savienojumi ar SMTP serveri)
- webhook paziņojumi (JavaScript HTTP savienojumi)
- savienojumi ar Vault

SSHKeyLocation

Norādiet publisko un privāto atslēgu atrašanās vietu SSH pārbaudēm un darbībām.

SSLCertLocation

Norādiet SSL klienta sertifikātu failu atrašanās vietu klienta autentifikācijai.
 Šis parametrs tiek izmantots tīmekļa uzraudzībā, saziņai ar Vault, vēstures glabāšanas nodrošinātājam ClickHouse un telemetrijas datu avotam ClickHouse.

SSLKeyLocation

Norādiet SSL privāto atslēgu failu atrašanās vietu klienta autentifikācijai.
 Šis parametrs tiek izmantots tīmekļa uzraudzībā, saziņai ar Vault, vēstures glabāšanas pakalpojumu sniedzējam ClickHouse un telemetrijas datu avotam ClickHouse.

SSLCALocation

Norādiet sertificēšanas iestādes (CA) failu atrašanās vietu SSL servera sertifikāta pārbaudei. Ja atrašanās vieta nav norādīta, tiks izmantots visas sistēmas direktorijs.
 Ņemiet vērā, ka šī parametra vērtība tiks iestatīta kā libcurl opcija `CURLOPT_CAPATH`. libcurl versijām, kas vecākas par 7.42.0, tas darbojas tikai tad, ja libcurl ir kompilēts OpenSSL izmantošanai. Plašāku informāciju skatiet cURL tīmekļa lapā.
 Šis parametrs tiek izmantots tīmekļa uzraudzībā, SMTP autentifikācijā, HTTP aģenta vienumos, saziņai ar Vault, vēstures glabāšanas nodrošinātājam ClickHouse un telemetrijas datu avotam ClickHouse.

StartAgentPollers

Iestata iepriekš sazaroto Zabbix aģenta polleru instanču skaitu. Skatiet MaxConcurrentChecksPerPoller.

Noklusējums: 1
 Diapazons: 0-1000

StartAlerters

Iestata iepriekš sazaroto alerters instanču skaitu.

Noklusējums: 3
 Diapazons: 1-100

StartBrowserPollers

Iestatiet iepriekš sazaroto browser vienuma pollers instanču skaitu.

Noklusējums: 1
 Diapazons: 0-1000

StartConnectors

Iestata iepriekš sazaroto savienotāju darbu izpildītāju instanču skaitu. Savienotāju pārvaldnieka process tiek palaists automātiski, kad tiek palaists savienotāju darbu izpildītājs.

Noklusējums: 0
 Diapazons: 0-1000

StartDBSyncers

Iestata iepriekš atzaroto history syncers instanču skaitu.
Piezīme: Esiet uzmanīgi, mainot šo vērtību, jo 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

Iestatiet iepriekš sazaroto atklāšanas darbu izpildītāju[1] skaitu.

Noklusējums: 5
 Diapazons: 0-1000

StartEscalators

Iestata iepriekš sazaroto eskalatoru instanču skaitu.

Noklusējums: 1
 Diapazons: 1-100

StartHistoryPollers

Iestata iepriekš sazaroto history pollers instanču skaitu.
Tikai nepieciešams aprēķinātajām pārbaudēm.

Noklusējums: 5
 Diapazons: 0-1000

StartHTTPAgentPollers

Iestata iepriekš sazaroto HTTP aģenta aptaujātāju instanču skaitu. Skatiet MaxConcurrentChecksPerPoller.

Noklusējums: 1
 Diapazons: 0-1000

StartHTTPPollers

Iestata iepriekš sazaroto HTTP pollers[1] instanču skaitu.

Noklusējums: 1
 Diapazons: 0-1000

StartIPMIPollers

Iestata iepriekš sazaroto IPMI aptaujātāju instanču skaitu.

Noklusējums: 0
 Diapazons: 0-1000

StartJavaPollers

Iestata iepriekš sazaroto Java pollers[1] instanču skaitu.

Noklusējums: 0
 Diapazons: 0-1000

StartLLDProcessors

Iestata iepriekš sazaroto zema līmeņa atklāšanas (LLD) darbinieku[1] skaitu.
 LLD pārvaldnieka process tiek automātiski palaists, kad tiek palaists LLD darbinieks.

Noklusējums: 2
 Diapazons: 1-100

StartODBCPollers

Iestata iepriekš sazaroto ODBC aptaujātāju[1] instanču skaitu.

Noklusējums: 1
 Diapazons: 0-1000

StartPingers

Iestata iepriekš sazaroto ICMP pingeru[1] instanču skaitu.

Noklusējums: 1
 Diapazons: 0-1000

StartPollersUnreachable

Iestata iepriekš atzaroto nepieejamu hostu aptaujātāju instanču skaitu (tostarp IPMI un Java)[1].
 Ja tiek palaisti parastie, IPMI vai Java aptaujātāji, ir jādarbojas vismaz vienam nepieejamu hostu aptaujātājam.

Noklusējums: 1
 Diapazons: 0-1000

StartPollers

Iestata iepriekš atzaroto pollers[1] instanču skaitu.

Noklusējums: 5
 Diapazons: 0-1000

StartPreprocessors

Iestatiet iepriekš palaisto priekšapstrādes worker[1] pavedienu skaitu ne mazāku par pieejamo CPU kodolu skaitu. Ja priekšapstrāde nav saistīta ar CPU noslodzi un tajā ir daudz tīkla pieprasījumu, jāiestata vairāk worker.

Noklusējums: 16
 Diapazons: 1-1000

StartProxyPollers

Iestata iepriekš sazaroto pasīvo starpniekserveru aptaujātāju[1] instanču skaitu.

Noklusējums: 1
 Diapazons: 0-250

StartReportWriters

Iestata iepriekš sazaroto report writers instanču skaitu.
Ja iestatīts uz 0, plānoto atskaišu ģenerēšana ir atspējota.
Atskaites pārvaldnieka process tiek automātiski palaists, kad tiek palaists report writer.

Noklusējums: 0
 Diapazons: 0-100

StartSNMPPollers

Iestata iepriekš sazaroto SNMP aptaujātāju instanču skaitu. Skatiet `MaxConcurrentChecksPerPoller`.

Noklusējums: 1
 Diapazons: 0-1000

StartSNMPTrapper

Iestatiet uz 1, lai palaistu SNMP trapper procesu.

Noklusējums: 0
 Diapazons: 0-1

StartTelemetryQueryPollers

Iestatiet asinhrono Telemetry vaicājumu pollers iepriekš izveidoto instanču skaitu. Skatiet MaxConcurrentChecksPerPoller.

Noklusējuma vērtība: 1
 Diapazons: 0-1000

StartTimers

Iestatiet iepriekš atzaroto taimeru instanču skaitu.
Taimeri apstrādā uzturēšanas periodus.

Noklusējums: 1
 Diapazons: 1-1000

StartTrappers

Iestata iepriekš sazaroto trapperu[1] instanču skaitu.
 Trapperi pieņem ienākošos savienojumus no Zabbix sender, aktīvajiem aģentiem un aktīvajiem starpniekserveriem.

Noklusējums: 5
 Diapazons: 0-1000

StartVMwareCollectors

Iestata iepriekš sazaroto VMware kolektora instanču skaitu.

Noklusējums: 0
 Diapazons: 0-250

StatsAllowedIP

Norādiet IP adreses (pēc izvēles CIDR notācijā) vai ārējo Zabbix instanču DNS nosaukumus, kurām ir atļauts pieprasīt statistiku. Ar komatiem atdalīts saraksts.
 Ja šis parametrs nav iestatīts, statistikas pieprasījumi netiks pieņemti.
 Ja ir iespējots IPv6 atbalsts, tad `127.0.0.1`, `::127.0.0.1`, `::ffff:127.0.0.1` tiek uzskatīti par ekvivalentiem, un `::/0` ļaus 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
```

TelemetryProvider

Iestatiet telemetrijas datu avotu, izmantojot formātu `<provider>[;<options>]`, kur:

- `<provider>` - viens no atbalstītajiem telemetrijas datu avotiem: `clickhouse`
- `<options>` - ar komatiem atdalīti `key=value` pāri. Vērtības var būt ietvertas pēdiņās.

ClickHouse opcijas:

- `url` - iestatiet HTTP vai HTTPS URL, kas tiek izmantots savienojuma izveidei ar telemetrijas datu avotu.
- `db` - iestatiet datubāzes nosaukumu.
- `username` - iestatiet lietotājvārdu, kas tiek izmantots, lai pieteiktos datubāzē.
- `password` - iestatiet paroli, kas tiek izmantota, lai pieteiktos datubāzē.
- `ssl_cert_file`, `ssl_key_file`, `ssl_key_password`, `ssl_verify_peer`, `ssl_verify_host`, `ssl_ca_location`, `ssl_cert_location`, `ssl_key_location` - iestatiet šīs opcijas, lai konfigurētu drošu (TLS/SSL) savienojumu ar datubāzi.
- `source_ip` - iestatiet avota IP adresi, kas tiek izmantota izejošajiem savienojumiem ar telemetrijas datu avotu.
- `vault_path` - iestatiet ceļu uz Vault noslēpumu, kurā glabājas `username` un `password` vērtības. Neizmantojiet kopā ar opcijām `username` un `password`.

ClickHouse piemērs:

```
TelemetryProvider=clickhouse;url=http://localhost:8123,db=zabbix,username=zabbix,password=zabbix
```

ClickHouse piemērs ar Vault:

```
TelemetryProvider=clickhouse;url=http://localhost:8123,db=zabbix,vault_path=clickhouse
```

Noildze

Norādiet, cik ilgi jāgaida (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 dažādu saziņas darbību ilgumu:

- attālinātas komandas izpilde Zabbix aģentā
- SSH/Telnet komandas izpilde
- pieprasījumi Zabbix tīmekļa pakalpojumam
- saziņas noildze mediatype testa pieprasījumam un `history.push` metodei
- vienumu pārplānošana, kad IPMI interfeiss kļūst nepieejams
- atbildes nosūtīšana Zabbix starpniekserverim, ja datu apmaiņa neizdodas tiesību vai šifrēšanas problēmu dēļ
- termiņš IPC asinhronajiem socketiem un izpildlaika vadības opcijām
- JMX savienojumi
- statistikas iegūšana no attālināta Zabbix starpniekservera vai servera
- atbilžu nosūtīšana Zabbix lietotāja saskarnei
- asinhrono poller DNS pieprasījumi
- atbilde aktīvās pārbaudes heartbeat
- Zabbix aģenta datu (vērtību) izgūšana no aktīvajiem aģentiem
- datu izgūšana no Zabbix sender
- atbilde, ja trapper neizdodas apstrādāt pieprasījumu
- aktīvās pārbaudes saraksta nosūtīšana Zabbix aģentam

Šī noildze **netiks** izmantota tām pārbaudēm, kurām lietotāja saskarnē ir konfigurēti elastīgās noildzes iestatījumi (globālā, 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

Norādiet pilnu ceļu uz failu, kurā ir augstākā līmeņa CA sertifikāti, kas tiek izmantoti vienaudžu sertifikātu verifikācijai un šifrētai saziņai starp Zabbix komponentiem.

TLSCertFile

Norādiet pilnu ceļu uz failu, kurā atrodas servera sertifikāts vai sertifikātu ķēde, ko izmanto šifrētai saziņai starp Zabbix komponentiem.

TLSCipherAll

Norādiet GnuTLS prioritātes virkni vai OpenSSL (TLS 1.2) šifru virkni. Pārrakstiet noklusējuma šifru komplekta atlases kritērijus sertifikātu un PSK balstītai šifrēšanai.

Piemērs:

```
TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSCipherAll13

Norādiet šifru virkni OpenSSL 1.1.1 vai jaunākai versijai TLS 1.3. Pārrakstiet noklusējuma šifru kopas atlases kritērijus sertifikātu un uz PSK balstītai šifrēšanai.

GnuTLS piemērs:

```
NONE:+VERS-TLS1.2:+ECDHE-RSA:+RSA:+ECDHE-PSK:+PSK:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL
```

OpenSSL piemērs:

```
EECDH+aRSA+AES128:RSA+aRSA+AES128:kECDHEPSK+AES128:kPSK+AES128
```

TLSCipherCert

Norādiet GnuTLS prioritātes virkni vai OpenSSL (TLS 1.2) šifru virkni. Pārrakstiet noklusējuma šifru kopas atlases kritērijus sertifikātu bāzētai šifrēšanai.

GnuTLS piemērs:

```
NONE:+VERS-TLS1.2:+ECDHE-RSA:+RSA:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL:+COMP-NULL:+SIG
```

OpenSSL piemērs:

```
EECDH+aRSA+AES128:RSA+aRSA+AES128
```

TLSCipherCert13

Norādiet OpenSSL 1.1.1 vai jaunākas versijas šifru virkni TLS 1.3 režīmā. Pārrakstiet noklusējuma šifru komplekta atlases kritērijus sertifikātā balstītai šifrēšanai.

TLSCipherPSK

Norādiet GnuTLS prioritātes virkni vai OpenSSL (TLS 1.2) šifru virkni. Pārrakstiet noklusējuma šifru komplekta atlases kritērijus PSK balstītai šifrēšanai.

GnuTLS piemērs:

```
NONE:+VERS-TLS1.2:+ECDHE-PSK:+PSK:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL:+COMP-NULL:+SIG
```

OpenSSL piemērs:

```
kECDHEPSK+AES128:kPSK+AES128
```

TLSCipherPSK13

Norādiet šifru virkni OpenSSL 1.1.1 vai jaunākai versijai TLS 1.3 režīmā. Pārrakstiet noklusējuma šifru komplekta atlases kritērijus PSK balstītai šifrēšanai.

Piemērs:

```
TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSCRLFile

Norādiet pilnu ceļu uz failu, kurā ir atsauktie sertifikāti. Šis parametrs tiek izmantots šifrētai saziņai starp Zabbix komponentiem.

TLSFrontendAccept

Norādiet šifrēšanas līmeni ienākošajiem savienojumiem no lietotāja saskarnes.

Var norādīt vairākas vērtības, atdalot tās ar komatu:

- unencrypted - pieņemt savienojumus bez šifrēšanas.
- cert - pieņemt savienojumus, kas ir aizsargāti ar TLS un sertifikātu.

Noklusējums: `unencrypted`

TLSFrontendCertIssuer

Norādiet atļauto lietotāja saskarnes sertifikāta izdevēju.

TLSFrontendCertSubject

Norādiet atļauto lietotāja saskarnes sertifikāta subjektu.

TLSKeyFile

Norādiet pilnu ceļu uz failu, kurā atrodas servera privātā atslēga, kas tiek izmantota šifrētai saziņai starp Zabbix komponentēm.

TLSListen

Pieprasīt tikai TLS savienojumus trapper ligzdā.

Atbalstītās vērtības:

- required - pieņemt tikai TLS savienojumus

TmpDir

Norādiet pagaidu direktoriju.

Noklusējums: `/tmp`

TrapperTimeout

Norādiet trapper taimautu sekundēs šādiem gadījumiem:

- vēsturisko datu izgūšanai no Zabbix starpniekserveris
- konfigurācijas datu nosūtīšanai uz Zabbix starpniekserveris
- globāla skripta izpildei vai attālinātas komandas izpildei Zabbix serveris

Noklusējums: 300
 Diapazons: 1-300

TrendCacheSize

Iestatiet tendenču kešatmiņas lielumu baitos.
Koplietotās atmiņas apjoms tendenču glabāšanai.

Noklusējums: 4M
 Diapazons: 128K-16G

TrendFunctionCacheSize

Iestatiet trendu funkciju kešatmiņas lielumu baitos.
Koplietotās atmiņas apjoms aprēķināto trendu funkciju datu kešošanai.

Noklusējums: 4M
 Diapazons: 128K-2G

Nav nepieejams aizkave

Iestatiet host pieejamības pārbaudes biežumu nepieejamības periodā sekundēs.

Noklusējums: 60
 Diapazons: 1-3600

UnreachableDelay

Iestatiet host pieejamības pārbaudes biežumu nepieejamības periodā sekundēs.

Noklusējums: 15
 Diapazons: 1-3600

Nesasniedzamības periods

Norāda, pēc cik sekundēm nesasniedzamības gadījumā hostu uzskatīt par nepieejamu.

Noklusējums: 45
 Diapazons: 1-3600

Lietotājs

Nolaiž privilēģijas līdz konkrētam, sistēmā jau esošam lietotājam.
Stājas spēkā tikai tad, ja tiek palaists kā 'root' un AllowRoot ir atspējots.

Noklusējums: `zabbix`

ValueCacheSize

Iestatiet vēstures vērtību kešatmiņas lielumu (baitos).
 Koplietojamās atmiņas lielums vienuma vēstures datu pieprasījumu kešatmiņai.
 Iestatot 0, vērtību kešatmiņa tiek atspējota (nav ieteicams).
 Kad vērtību kešatmiņā beidzas koplietojamā atmiņa, ik pēc 5 minūtēm servera žurnālā tiek ierakstīts brīdinājuma ziņojums.

Noklusējums: 8M
 Diapazons: 0, 128K-64G

Vault

Norādiet vault nodrošinātāju: - *HashiCorp* - HashiCorp KV Secrets Engine versija 2; - *CyberArk* - CyberArk Central Credential Provider
Jāatbilst vault nodrošinātājam, kas iestatīts lietotāja saskarnē.

Noklusējums: `HashiCorp`

VaultAppRoleID

Norādiet HashiCorp vault AppRole ID.
 Ja ir iestatīts, iespējo AppRole autentifikāciju. To nevar izmantot kopā ar VaultToken.
 Atbalstīts tikai HashiCorp vault.

VaultAppSecretID

Norādiet HashiCorp vault AppRole slepeno ID.
 Atbalstīts tikai HashiCorp vault.

VaultDBPath

Norādiet atrašanās vietu datubāzes akreditācijas datu izgūšanai pēc atslēgām. Tā ir Vault ceļa vai vaicājuma vērtība atkarībā no Vault.

Atslēgas, ko izmanto HashiCorp, ir `password` un `username`.

Ceļa piemērs ar `VaultPrefix=/v1/secret/data/zabbix/`:

```
database
```

Ceļa piemērs bez VaultPrefix:

```
secret/zabbix/database
```

Atslēgas, ko izmanto 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

Norādiet pielāgotu prefiksu vault ceļam vai vaicājumam atkarībā no Vault. Ja tas nav norādīts, tiks izmantotas vispiemērotākās noklusējuma vērtības.
 Ņemiet vērā, ka `data` tiek automātiski pievienots pēc mountpoint HashiCorp gadījumā, ja `VaultPrefix` nav norādīts.

Prefiksa piemērs HashiCorp gadījumā:

```
/v1/secret/data/zabbix/
```

Prefiksa piemērs Cyberark gadījumā:

```
/AIMWebService/api/Accounts?
```

VaultTLSCertFile

Norādiet SSL sertifikāta faila nosaukumu, ko izmanto klienta autentifikācijai. Sertifikāta failam jābūt PEM1 formātā.
 Ja sertifikāta fails satur arī privāto atslēgu, atstājiet SSL atslēgas faila lauku tukšu.
 Direktorijs, kurā atrodas šis fails, ir norādīts ar konfigurācijas parametru SSLCertLocation.
 Šo opciju var neizmantot, taču tā ir ieteicama CyberArkCCP vault.

VaultTLSKeyFile

Norādiet SSL privātās atslēgas faila nosaukumu, ko izmanto klienta autentifikācijai. Privātās atslēgas failam jābūt PEM1 formātā.
Šī faila saturošais katalogs ir norādīts ar konfigurācijas parametru SSLKeyLocation.
Šo opciju var neiekļaut, taču tā ir ieteicama CyberArkCCP vault.

VaultToken

Norādiet HashiCorp Vault autentifikācijas marķieri. HashiCorp Vault autentifikācijas marķierim jābūt ģenerētam tikai Zabbix serverim ar tikai lasīšanas tiesībām uz ceļiem, kas norādīti Vault macros, un tikai lasīšanas tiesībām uz ceļu, kas norādīts izvēles VaultDBPath konfigurācijas parametrā.
 Tā ir kļūda, ja `VaultToken` un `VAULT_TOKEN` vides mainīgais ir definēti vienlaikus.
 Nevar izmantot kopā ar VaultAppRoleID.
 Atbalstīts tikai HashiCorp Vault.

VaultURL

Norādiet vault servera URL. Ja SSLCALocation nav norādīts, tiks izmantots sistēmas CA sertifikātu katalogs.

Noklusējums: `https://127.0.0.1:8200`

VMwareCacheSize

Iestatiet koplietotās atmiņas lielumu VMware datu glabāšanai. VMware iekšējo pārbaudi `zabbix[vmware,buffer,...]` var izmantot, lai uzraudzītu VMware kešatmiņas lietojumu (skatiet Iekšējās pārbaudes).
Ņemiet vērā, ka koplietotā atmiņa netiek piešķirta, ja nav konfigurēti `vmware collector` instanču palaišanai.

Noklusējums: 8M
 Diapazons: 256K-2G

VMwareFrequency

Iestatiet datu apkopošanas biežumu no viena VMware pakalpojuma (sekundēs). Šis biežums jāiestata kā mazākais atjaunināšanas intervāls jebkuram VMware uzraudzības vienumam.

Noklusējums: 60
 Diapazons: 10-86400

VMwarePerfFrequency

Iestatiet veiktspējas skaitītāju statistikas iegūšanas biežumu no viena VMware pakalpojuma (sekundēs).
Šis biežums jāiestata uz mazāko atjaunināšanas intervālu jebkuram VMware uzraudzības vienumam, kas izmanto VMware veiktspējas skaitītājus.

Noklusējums: 60
 Diapazons: 10-86400

VMwareTimeout

Iestata maksimālo sekunžu skaitu, cik ilgi `vmware collector` gaidīs atbildi no VMware pakalpojuma (vCenter vai ESX hipervizora).

Noklusējums: 10
 Diapazons: 1-300

WebDriverURL

Norādiet WebDriver saskarnes URL.

Piemērs (izmantošanai ar Selenium WebDriver standalone serveri):

```
WebDriverURL=http://localhost:4444
```

WebServiceURL

Norādiet tīmekļa pakalpojuma URL formātā `http[s]://host:port/report`.

Piemērs:

```
WebServiceURL=http://localhost:10053/report
```

`http://` shēmu var izlaist tikai savienojumiem bez TLS (HTTP); ja ir konfigurēts TLS, jāizmanto `https://`.

Piezīmes

[1] Ņemiet vērā, ka pārāk daudz datu vākšanas procesu (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 priekšapstrādes pārvaldniekam noteikto failu deskriptoru limita vērtību katram procesam.

> **Warning:**
> Failu deskriptoru limita izsmelšana izraisīs Zabbix serveris apstāšanos, parasti neilgi pēc palaišanas, bet dažkārt tas var aizņemt ilgāku laiku. Lai izvairītos no šādām problēmām, pārskatiet Zabbix servera konfigurācijas failu, lai optimizētu vienlaicīgo pārbaužu un procesu skaitu. Turklāt, ja nepieciešams, pārliecinieties, ka failu deskriptoru limits ir iestatīts pietiekami augsts, pārbaudot un pielāgojot sistēmas limitus.

Starpniekserveris

Pārskats Zabbix starpniekserveris ir process, kas var apkopot uzraudzības datus no vienas vai vairākām uzraudzītām ierīcēm un nosūtīt šo informāciju Zabbix serverim, būtībā darbojoties servera vārdā. Visi apkopotie dati tiek lokāli buferēti un pēc tam pārsūtīti Zabbix serverim, kuram pieder starpniekserveris.

Starpniekservera izvietošana nav obligāta, taču tā var būt ļoti noderīga, lai sadalītu viena Zabbix servera slodzi. Ja datus apkopo tikai starpniekserveri, servera apstrādei ir nepieciešams mazāk CPU resursu un diska I/O.

Zabbix starpniekserveris ir ideāls risinājums attālu atrašanās vietu, filiāļu un tīklu centralizētai uzraudzībai, ja tajos nav vietējo administratoru.

Zabbix starpniekserverim ir nepieciešama atsevišķa datubāze.

> **Attention:**
> Ņemiet vērā, ka Zabbix starpniekservera atbalstītās datubāzes ir SQLite, MySQL un PostgreSQL.

Skatiet arī:

- Izplatīta uzraudzība, izmantojot starpniekserverus
- Starpniekservera slodzes balansēšana un augsta pieejamība
- Uzraudzības konfigurācijas sinhronizācija
- Datu apmaiņas protokols starp serveri un starpniekserveri
- Konfigurācijas parametri: Zabbix starpniekserveris
- Zabbix Cloud starpniekservera iestatīšana

Darbojošs starpniekserveris 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
```

Startēt manuāli

Ja iepriekš minētais nedarbojas, jums tas jāstartē manuāli. Atrodiet `zabbix_proxy` binārā faila ceļu un izpildiet:

```
zabbix_proxy
```

Ar Zabbix starpniekserveri varat izmantot šādus komandrindas parametrus:

Parameter	Description
`-c --config <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 palīdzības informāciju un iziet.
`-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:

Option	Description	Target
`config_cache_reload`	Pārlādēt konfigurācijas kešatmiņu. Tiek ignorēts, ja kešatmiņa pašlaik tiek ielādēta. Aktīvais Zabbix starpniekserveris izveidos savienojumu ar Zabbix serveri un pieprasīs konfigurācijas datus. Pasīvais Zabbix starpniekserveris pieprasīs konfigurācijas datus no Zabbix servera nākamreiz, kad serveris izveidos savienojumu ar starpniekserveri.	
`history_cache_clear=<target>`	Notīrīt vēstures kešatmiņu vienumam, kas norādīts pēc tā ID. Ietekmē visas vienuma vērtības, izņemot pirmo un pēdējo vērtību.	**target** - vienuma ID.
`diaginfo[=<section>]`	Saņemt diagnostikas informāciju starpniekservera žurnāla failā.	`historycache` - vēstures kešatmiņas statistika; `preprocessing` - pirmapstrādes pārvaldnieka statistika; `locks` - mutexu saraksts (sistēmās *BSD* tas ir tukšs).
`snmp_cache_reload`	Pārlādēt SNMP kešatmiņu - notīrīt SNMP dzinēja īpašības (dzinēja laiks, dzinēja palaišanas reižu skaits, dzinēja ID, akreditācijas dati) visiem hosts. Izmantojiet, lai piespiestu globālu kešatmiņas notīrīšanu, novēršot SNMP problēmas.	
`housekeeper_execute`	Palaist housekeeping procedūru. Tiek ignorēts, ja housekeeping procedūra pašlaik jau notiek.	

Option	Description	Target
`log_level_increase[=<target>]`	Palielināt žurnāla līmeni, ietekmē visus procesus, ja target nav norādīts. Nav 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 (1 līdz 65535). Lielākām vērtībām norādiet target kā 'process type,N'.
`log_level_decrease[=<target>]`	Samazināt žurnāla līmeni, ietekmē visus procesus, ja target nav norādīts. Nav atbalstīts *BSD* sistēmās.	
`prof_enable[=<target>]`	Iespējot profilēšanu. Ietekmē visus procesus, ja target nav norādīts. Iespējota profilēšana sniedz informāciju par visiem rwlocks/mutexiem 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 (1 līdz 65535). Lielākām vērtībām norādiet target kā 'process type,N'. **scope** - `rwlock`, `mutex`, `processing` var izmantot kopā ar procesa tipu un numuru (piemēram, `history syncer,1,processing`) vai visiem tipa procesiem (piemēram, `history syncer,rwlock`).
`prof_disable[=<target>]`	Atspējot profilēšanu. Ietekmē visus procesus, ja target 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 (1 līdz 65535). Lielākām vērtībām norādiet target kā 'process type,N'.

Ja procesa tips ir norādīts kā izpildlaika vadības target, komanda attiecas tikai uz šī tipa aktīvajiem procesiem. Ja pašlaik nav aktīvs neviens norādītā tipa process, komanda atgriež kļūdu:

```
zabbix_proxy -R log_level_increase='data sender'
Cannot redirect signal: "data sender" process does not exist
```

Izpildlaika vadības izmantošanas piemērs, lai pārlādētu starpniekservera konfigurācijas kešatmiņu:

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

Izpildlaika vadības izmantošanas piemērs, lai notīrītu vienuma vēstures kešatmiņu:

```
zabbix_proxy -c /usr/local/etc/zabbix_proxy.conf -R history_cache_clear=42243
```

Izpildlaika vadības izmantošanas piemēri diagnostikas informācijas savākšanai:

```
### Savākt visu pieejamo diagnostikas informāciju starpniekservera žurnāla failā:
zabbix_proxy -R diaginfo

### Savākt vēstures kešatmiņas statistiku starpniekservera žurnāla failā:
zabbix_proxy -R diaginfo=historycache
```

Izpildlaika vadības izmantošanas piemērs, lai pārlādētu SNMP kešatmiņu:

```
zabbix_proxy -R snmp_cache_reload
```

Attention:
Kad SNMPv3 interfeiss tiek atjaunināts, izmantojot Zabbix lietotāja saskarni, vairumā gadījumu Zabbix automātiski pārlādēs šī interfeisa jaunos SNMPv3 akreditācijas datus; izmantojiet `-R snmp_cache_reload` tikai tad, ja aptauja joprojām neizdodas pēc akreditācijas datu maiņas (piemēram, engineBoots/engineID neatbilstību vai ierīču, kas neatbilst RFC, dēļ), vai arī tad, ja nepieciešams piespiest globālu SNMP kešatmiņas notīrīšanu problēmu novēršanai.

Izpildlaika vadības izmantošanas piemērs, lai aktivizētu housekeeper izpildi:

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

Izpildlaika vadības izmantošanas piemēri žurnāla līmeņa maiņai:

```
### Palielināt visu procesu žurnāla līmeni:
zabbix_proxy -c /usr/local/etc/zabbix_proxy.conf -R log_level_increase

### Palielināt otrā poller procesa žurnāla līmeni:
zabbix_proxy -c /usr/local/etc/zabbix_proxy.conf -R log_level_increase=poller,2

### Palielināt procesa ar PID 1234 žurnāla līmeni:
zabbix_proxy -c /usr/local/etc/zabbix_proxy.conf -R log_level_increase=1234

### Samazināt visu http poller procesu žurnāla līmeni:
zabbix_proxy -c /usr/local/etc/zabbix_proxy.conf -R log_level_decrease="http poller"
```

Procesa lietotājs

Zabbix starpniekserveris ir paredzēts darbībai kā lietotājam, kas nav root lietotājs. Tas darbosies kā jebkurš lietotājs, kas nav root lietotājs un ar kura tiesībām tas tiek palaists. Tāpēc starpniekserveri varat palaist kā jebkuru lietotāju, kas nav root lietotājs, bez problēmām.

Ja mēģināsiet to palaist kā root lietotāju, tas pārslēgsies uz kodā fiksēto lietotāju `zabbix`, kuram jābūt jūsu sistēmā. Starpniekserveri kā root lietotāju varat palaist tikai tad, ja attiecīgi modificējat parametru `AllowRoot` starpniekservera konfigurācijas failā.

Konfigurācijas fails

Noklusējuma ceļš uz konfigurācijas failu ir atkarīgs no operētājsistēmas un Zabbix instalēšanas metodes:

- Instalācijai no avota kodiem: `/usr/local/etc/zabbix_proxy.conf`, vai kā noteikts ar kompilēšanas laikā mainīgajiem `--sysconfdir` vai `--prefix`.
- Instalācijai no pakotnēm: `/etc/zabbix/zabbix_proxy.conf`.

Skatiet konfigurācijas faila opcijas, lai iegūtu informāciju par `zabbix_proxy` konfigurēšanu.

Starpniekservera procesu veidi un pavedieni

- `agent poller` - asinhrons pollera process pasīvajām pārbaudēm ar darba pavedienu;
- `apm manager` - pārvaldnieka process lietojumprogrammu veiktspējas uzraudzībai (APM);
- `apm worker` - process lietojumprogrammu veiktspējas uzraudzības (APM) uzdevumu apstrādei no `apm manager`;
- `availability manager` - process hostu pieejamības atjaunināšanai;
- `browser poller` - polleris pārlūkprogrammas vienumu pārbaudēm;
- `configuration syncer` - process konfigurācijas datu atmiņas kešatmiņas pārvaldībai;
- `data sender` - starpniekservera datu sūtītājs;
- `discovery manager` - pārvaldnieka process ierīču atklāšanai;
- `discovery worker` - process atklāšanas uzdevumu apstrādei no `discovery manager`;
- `history syncer` - vēstures datubāzes rakstītājs;
- `housekeeper` - process novecojušas vienumu vēstures noņemšanai;
- `http agent poller` - asinhrons pollera process HTTP pārbaudēm ar darba pavedienu;
- `http poller` - tīmekļa uzraudzības polleris;
- `icmp pinger` - polleris icmpping pārbaudēm;
- `internal poller` - polleris iekšējām pārbaudēm;
- `ipmi manager` - IPMI pollera pārvaldnieks;
- `ipmi poller` - polleris IPMI pārbaudēm;
- `java poller` - polleris Java pārbaudēm;
- `odbc poller` - polleris ODBC pārbaudēm;
- `poller` - parastais polleris 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 pollera process SNMP pārbaudēm ar darba pavedienu (tikai `walk[OID]` un `get[OID]` vienumiem);
- `snmp trapper` - trapperis SNMP trapu apstrādei;
- `supervisor` - koordinē servera procesu un pavedienu palaišanu un apturēšanu, kā arī uzrauga to gatavību;
- `task manager` - process citu komponentu pieprasītu uzdevumu attālai izpildei (piemēram, problēmas aizvēršanai, problēmas apstiprināšanai, vienuma vērtības tūlītējai pārbaudei un attālās komandas funkcionalitātei);
- `telemetry query poller` - polleris telemetrijas vaicājumu vienumu pārbaudēm;
- `trapper` - trapperis aktīvajām pārbaudēm, trapēm un saziņai ar starpniekserveri;
- `unreachable poller` - polleris nesasniedzamām ierīcēm;
- `vmware collector` - VMware datu savācējs, kas atbild par datu apkopošanu no VMware pakalpojumiem.

Šos procesu veidus var novērot starpniekservera žurnālfailā.

Starpniekservera žurnālfails tiek izveidots ar lasīšanas un rakstīšanas atļaujām tikai faila īpašniekam. Turklāt failu var lasīt īpašnieka grupa. Visas pārējās atļaujas ir liegtas.

Dažādus Zabbix starpniekservera procesu veidus var uzraudzīt, izmantojot iekšējo vienumu `zabbix[process,<type>,<mode>,<state>]`.

Vēstures sinhronizētāja transakciju statistika

Vēstures sinhronizētāja procesa nosaukums parāda 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 formātā `processed...in N (<timings>) sec` ir:

- Laiks, kas pavadīts, rakstot vienuma vērtības datubāzē.
- Laiks, kas pavadīts, atjauninot vienuma 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 dzēšanas ierobežojumu vienā ciklā 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 uz tā paša atbalstīto platformu saraksta kā Zabbix serveris.

Atmiņas buferis Atmiņas buferis ļauj saglabāt jaunus datus (vienumu vērtības, tīkla atklāšanu, automātisku hostu reģistrāciju) buferī un augšupielādēt tos Zabbix serverī, nepiekļūstot datubāzei. Atmiņas buferis starpniekserverim tika ieviests Zabbix 7.0 versijā.

Instalācijās pirms Zabbix 7.0 apkopotie dati pirms augšupielādes Zabbix serverī tika saglabāti datubāzē. Šajās instalācijās šāda darbība arī pēc jaunināšanas uz Zabbix 7.0 paliek noklusējuma darbība.

Lai nodrošinātu optimālu veiktspēju, ieteicams konfigurēt atmiņas bufera izmantošanu starpniekserverī. To var izdarīt, mainot `ProxyBufferMode` vērtību no `disk` (esošām instalācijām iepriekš iestatītā noklusējuma vērtība) uz `hybrid` (ieteicams) vai `memory`. Ir jāiestata arī atmiņas bufera lielums (parametrs `ProxyMemoryBufferSize`).

Hibrīdajā režīmā buferis ir aizsargāts pret datu zudumu, nosūtot nenosūtītos datus uz datubāzi, ja starpniekserveris tiek apturēts, buferis ir pilns vai dati ir pārāk veci. Kad visas vērtības ir nosūtītas uz datubāzi, starpniekserveris atkal sāk izmantot atmiņas buferi.

Atmiņas režīmā tiks izmantots atmiņas buferis, taču nav aizsardzības pret datu zudumu. Ja starpniekserveris tiek apturēts vai atmiņa tiek pārpildīta, nenosūtītie dati tiks atmesti.

Hibrīdais 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 vecumu buferī.

Ņemiet vērā, ka konfliktējošas konfigurācijas gadījumā starpniekserveris izvadīs kļūdu un neizdosies startēt, piemēram, ja:

- `ProxyBufferMode` ir iestatīts uz `hybrid` vai `memory` un `ProxyMemoryBufferSize` ir 0.
- `ProxyBufferMode` ir iestatīts uz `hybrid` vai `memory` un `ProxyLocalBuffer` nav 0.

Lokāle Ņemiet vērā, ka starpniekserverim ir nepieciešama UTF-8 lokāle, lai dažus teksta vienumus varētu pareizi interpretēt. Lielākajai daļai mūsdienu Unix tipa sistēmu UTF-8 lokāle ir noklusējuma iestatījums, tomēr dažās sistēmās tā var būt jāiestata īpaši.

Rindu aprēķins apkopes laikā

> **Attention:**
> Zabbix starpniekserveris nav informēts par apkopes periodiem; skatiet Rindu aprēķins apkopes laikā, informāciju.

Zabbix lietotāja saskarnes izmantošana aiz starpniekservera Ja Zabbix lietotāja saskarne darbojas aiz starpniekservera, starpniekservera konfigurācijas failā ir jāpārraksta sīkfaila ceļš, lai tas atbilstu reversā starpniekservera ceļam. Skatiet tālāk sniegtos piemērus. Ja sīkfaila ceļš netiek pārrakstīts, lietotājiem, mēģinot pieteikties Zabbix lietotāja saskarnē, var rasties autorizācijas problēmas.

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
### ..
```

Izkliedētā uzraudzība, izmantojot starpniekserverus

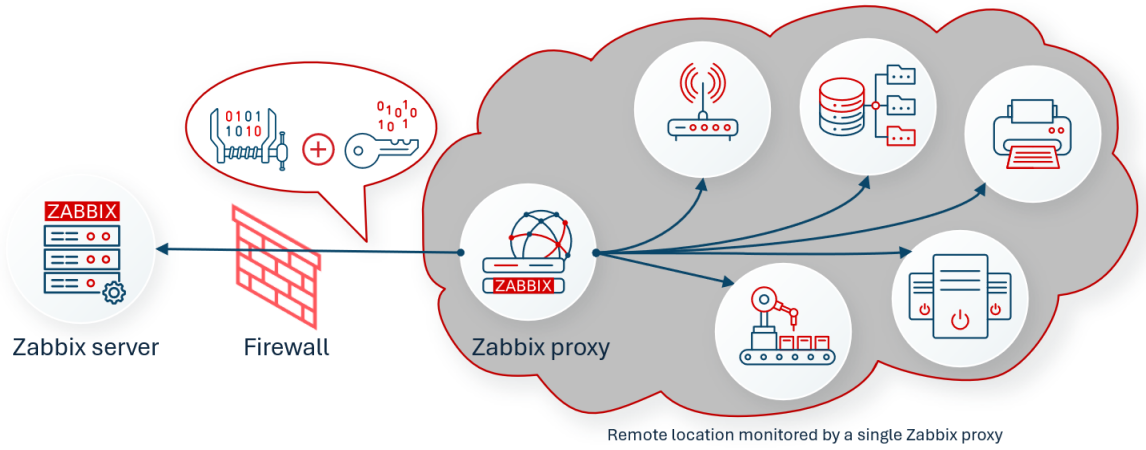
Pārskats

Zabbix starpniekserverus var izmantot efektīvas un uzticamas izkliedētas uzraudzības sistēmas izveidei.

Starpniekserveri tiek izvietoti lokāli, lai apkopotu veiktspējas un pieejamības datus. Pēc tam dati tiek nosūtīti centralizētam serverim.

Zabbix starpniekserveri var izmantot, lai:

- Uzraudzītu attālas atrašanās vietas
- Uzraudzītu atrašanās vietas ar neuzticamiem sakariem
- Atslogotu Zabbix serveri, uzraugot tūkstošiem ierīču
- Apkopotu telemetrijas datus lietojumprogrammu veiktspējas uzraudzībai (APM)



Starpniekserverim ir nepieciešams tikai viens TCP savienojums ar Zabbix serveri. Tādējādi ir vieglāk apiet ugunsmūri, jo jā konfigurē tikai viens ugunsmūra noteikums.

> **Attention:**
> Zabbix starpniekserverim jāizmanto atsevišķa datubāze. Norādot Zabbix servera datubāzi, konfigurācija tiks sabojāta.

Visi starpniekservera apkopotie dati tiek glabāti lokāli, pirms tie tiek nosūtīti serverim. Tādējādi īslaicīgu sakaru problēmu ar serveri dēļ dati netiek zaudēti. Parametri `ProxyLocalBuffer` un `ProxyOfflineBuffer` starpniekservera konfigurācijas failā nosaka, cik ilgi dati tiek glabāti lokāli.

Attention:
Var gadīties, ka starpniekserverim, kas jaunākās konfigurācijas izmaiņas saņem tieši no Zabbix servera datubāzes, ir jaunāka konfigurācija nekā Zabbix serverim, kura konfigurācija var netikt atjaunināta tik ātri CacheUpdateFrequency vērtības dēļ. Tā rezultātā starpniekserveris var sākt apkopot datus un nosūtīt tos Zabbix serverim, kas šos datus ignorē.

Zabbix starpniekserveris ir datu apkopotājs. Tas neaprēķina trigerus, neapstrādā notikumus un nesūta brīdinājumus. Lai uzzinātu, kādas funkcijas atbalsta starpniekserveris, 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 slazdi	**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**
	Skripta vienumi	**Jā**
	Pārlūkprogrammas vienumi	**Jā**
	Telemetrijas vaicājumu vienumi	**Jā**
Iebūvētā tīmekļa uzraudzība		**Jā**
Vienuma vērtības priekšapstrāde		**Jā**
Tīkla atklāšana		**Jā**
Aktīva aģenta automātiskā reģistrācija		**Jā**
Zema līmeņa atklāšana		**Jā** [2]
Attālās komandas		**Jā**
Telemetrijas datu apkopošana (APM)		**Jā** [3]
Trigeru aprēķināšana		*Nē*
Notikumu apstrāde		*Nē*
Notikumu korelācija		*Nē*
Brīdinājumu nosūtīšana		*Nē*

Note:
[1] Lai nodrošinātu, ka aģents aktīvo pārbaužu veikšanai vaicā starpniekserverim, nevis serverim, starpniekserverim jābūt norādītam aģenta konfigurācijas faila parametrā **ServerActive**.
 [2] Zema līmeņa atklāšanas gadījumā Zabbix starpniekserveris tikai apkopo un priekšapstrādā datus, pēc tam nosūtot tos Zabbix serverim turpmākai apstrādei.
 [3] Telemetrijas dati tiek apkopoti tikai tad, ja starpniekserverī ir iespējots APM (skatiet cilni *APM* sadaļā starpniekservera konfigurācija); Zabbix serveris nevar saņemt telemetrijas datus.

Starpniekservera funkcijas

Izvēloties, vai izmantot starpniekserveri vai ne, ir jāņem vērā vairāki apsvērumi.

	Starpniekserveris
Viegls	**Jā**
GUI	Nē
Darbojas neatkarīgi	**Jā**
Vienkārša uzturēšana	**Jā**
Automātiska DB izveide	**Jā**[1]

	Starpniekserveris
Lokāla administrēšana	Nē
Gatavs iegultai aparatūrai	**Jā**
Vienvirziena TCP savienojumi	**Jā**
Centralizēta konfigurācija	**Jā**
Ģenerē paziņojumus	Nē

[1] Automātiskās DB izveides funkcija darbojas tikai ar SQLite. Citām atbalstītajām datubāzēm ir nepieciešama manuāla iestatīšana.

Attention:
Zabbix starpniekserveris nezina par uzturēšanas periodiem; sīkāku informāciju skatiet sadaļā Rindu aprēķins uzturēšanas laikā.

Aizsardzība pret pārslodzi

Ja Zabbix serveris kādu laiku bija dīkstāvē un starpniekserveri bija savākuši daudz datu, tad, serverim startējoties, tas var tikt pārslogots (history cache lietojums kādu laiku saglabājas 95-100% līmenī). Šī pārslodze var izraisīt veiktspējas pasliktināšanos, kad pārbaudes tiek apstrādātas lēnāk, nekā vajadzētu. Lai izvairītos no problēmām, kas rodas history cache pārslodzes dēļ, tika ieviesta aizsardzība pret šo scenāriju.

Kad Zabbix servera history cache ir pilna, rakstīšanas piekļuve history cache tiek ierobežota, aizturot servera datu apkopošanas procesus. Visbiežākais history cache pārslodzes gadījums ir pēc servera dīkstāves, kad starpniekserveri augšupielādē savāktos datus. Lai no tā izvairītos, tika pievienota starpniekserveru ierobežošana (pašlaik to nevar atspējot).

Kad history cache lietojums sasniedz 80%, Zabbix serveris pāriet ierobežošanas režīmā. Ierobežošanas režīmā serveris pieņem starpniekserveru datus tikai tad, ja history cache lietojums ir zem 60%, pārmaiņus apkalpojot pieņemtos starpniekserverus. Kad history cache lietojums nokrītas zem 20%, serveris atgriežas normālajā režīmā.

Papildus tam normālajā režīmā Zabbix serveris ierobežo atsevišķus starpniekserverus, kas sūta ļoti lielas paketes (10 000+ ierakstu), ja history cache lietojums pārsniedz 60%. Šis lēmums tiek piemērots brīdī, kad serveris izvērtē starpniekservera augšupielādi, un tāpēc tas ne vienmēr uzreiz atspoguļojas history cache lietojuma grafikā (iekšējais vienums `zabbix[wcache,history,pused]` un tā atjaunināšanas intervāls var nepamanīt īslaicīgus pīķus).

Šis ierobežošanas režīms turpināsies līdz brīdim, kad vai nu kešatmiņas lietojums atkal sasniegs 80%, vai nokritīsies 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 var ilustrēt šādā tabulā:

History write cache usage	Zabbix server mode	Zabbix server action
Sasniedz 80%	Gaida	Pārtrauc pieņemt starpniekserveru datus, bet saglabā *ierobežošanas sarakstu* (prioritizētu starpniekserveru sarakstu, ar kuriem vēlāk sazināties).
Sasniedz 60%	Normāls, bet sagatavots ierobežošanai	Var atteikt ļoti lielas starpniekserveru augšupielādes (vairāk nekā 10 tūkst. ierakstu), lemjot par datu pieņemšanu; turpina pieņemt citus starpniekserveru datus.
Nokrīt līdz 20%	Normāls	Atmet ierobežošanas sarakstu un sāk pieņemt starpniekserveru datus normāli.

Lai sasaistītu šo Zabbix servera uzvedību ar metrikām, varat izmantot iekšējo vienumu zabbix[wcache,history,pused].

Konfigurācija

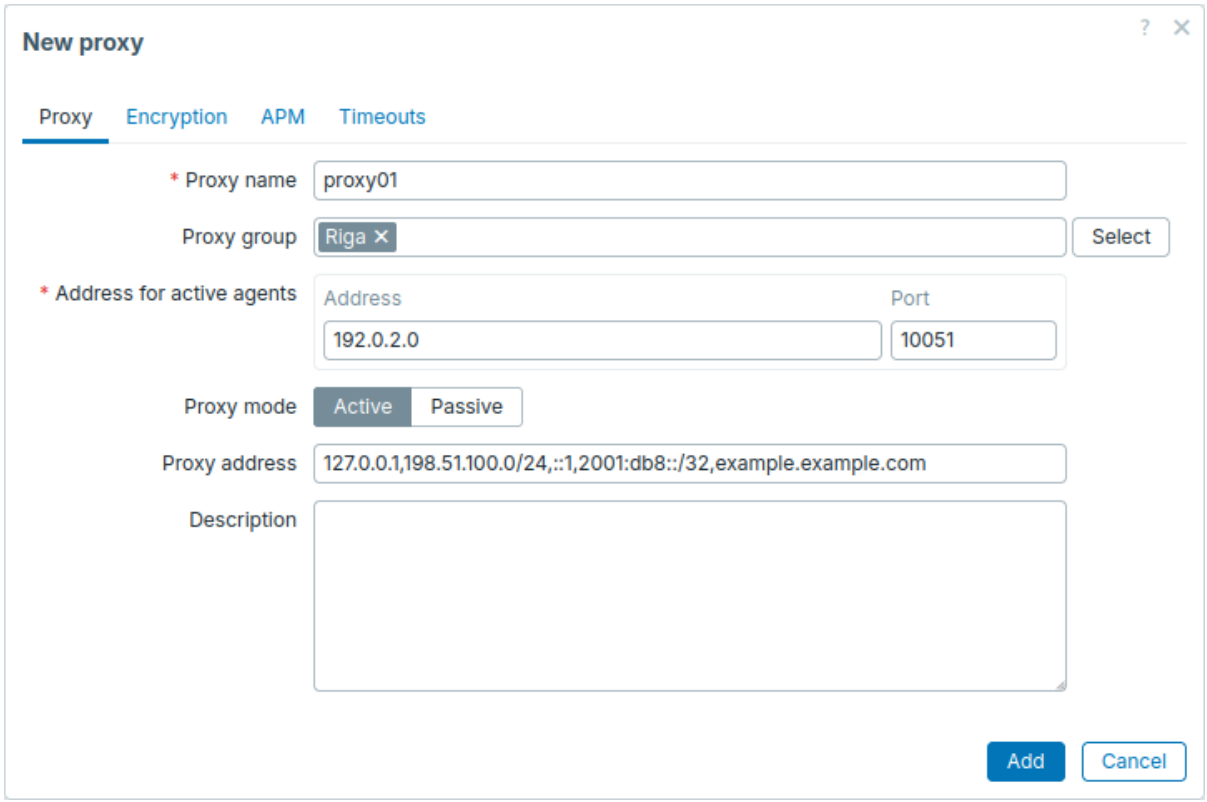
Kad esat instalējis un konfigurējis starpniekserveri, ir pienācis laiks to konfigurēt Zabbix lietotāja saskarnē.

Starpniekserveru pievienošana

Note:
Tā vietā, lai instalētu starpniekserveri savā infrastruktūrā, varat to izvietot Zabbix Cloud. Lai to izdarītu, sadaļā *Administration* > *Proxies* noklikšķiniet uz *Create proxy in Cloud*; Zabbix Cloud tiks atvērts jaunā pārlūkprogrammas cilnē. Pēc starpniekservera mezgla izveides konfigurējiet starpniekserveri Zabbix lietotāja saskarnē, kā aprakstīts tālāk. Plašāku informāciju skatiet sadaļā Zabbix Cloud starpniekservera iestatīšana.

Lai konfigurētu starpniekserveri Zabbix lietotāja saskarnē:

1. Dodieties uz *Administration* > *Proxies*.
2. Noklikšķiniet uz *Create proxy*.
3. Veidlapā ievadiet starpniekservera parametrus.



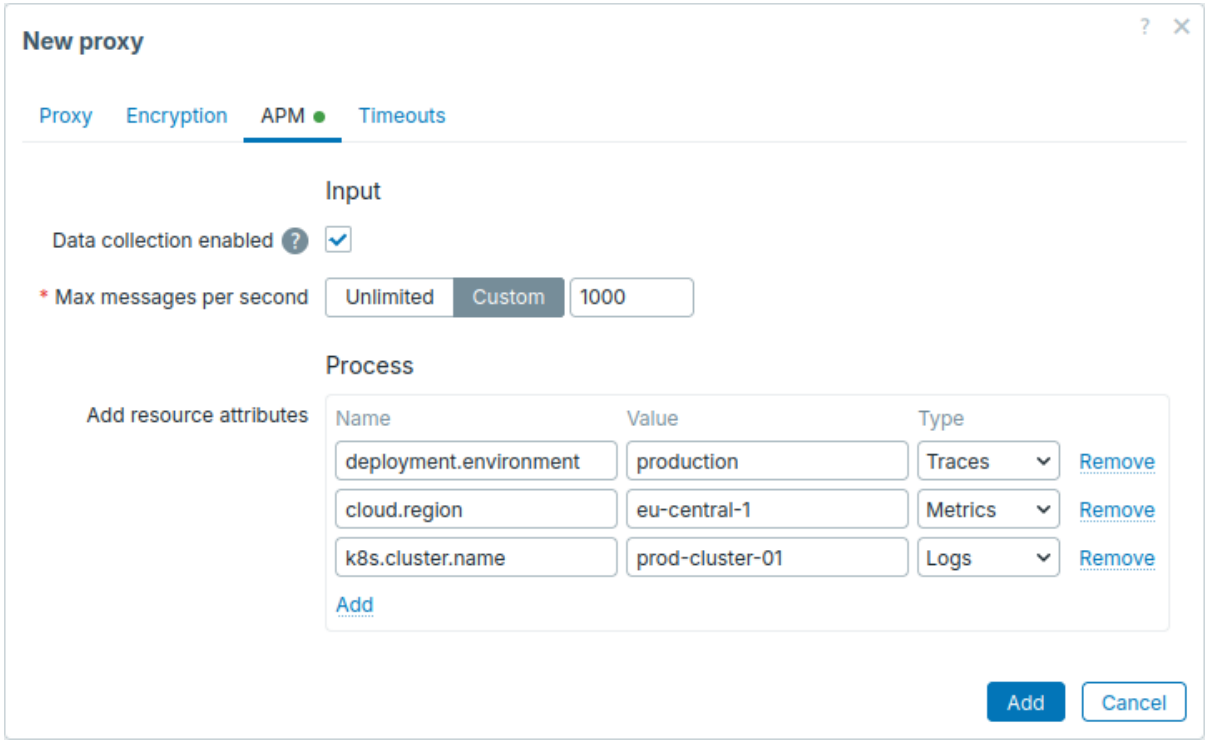
Parameter	Description
Proxy name	Ievadiet starpniekservera nosaukumu. Tam jābūt tādam pašam kā *Hostname* parametrā starpniekservera konfigurācijas failā.
Proxy group	Atlasiet vienu starpniekserveru grupu starpniekserveru slodzes balansēšanai/augstai pieejamībai.
Address for active agents	Ievadiet adresi, kurai jāizveido savienojums uzraugāmajiem aktīvajiem aģentiem vai sūtītājiem. Atbalstīts **tikai** Zabbix 7.0 vai jaunāku versiju aģentiem.
Address for active agents	Šī adrese tiek izmantota savienojuma izveidei gan ar aktīviem, gan pasīviem starpniekserveriem. Šis lauks ir pieejams tikai tad, ja laukā *Proxy group* ir atlasīta starpniekserveru grupa.
Address	IP adrese/DNS nosaukums, ar kuru izveidot savienojumu.
Port	TCP porta numurs (pēc noklusējuma 10051), ar kuru izveidot savienojumu. Tiek atbalstīti lietotāja makro.
Proxy mode	Atlasiet starpniekservera režīmu. **Active** - starpniekserveris izveidos savienojumu ar Zabbix serveri un pieprasīs konfigurācijas datus **Passive** - Zabbix serveris izveidos savienojumu ar starpniekserveri *Ņemiet vērā*, ka, izmantojot aktīvu starpniekserveri bez šifrētas saziņas (sensitīviem) starpniekservera konfigurācijas datiem var piekļūt puses, kurām ir piekļuve Zabbix servera trapper portam. Tas ir iespējams, jo ikviens var uzdoties par aktīvu starpniekserveri un pieprasīt konfigurācijas datus, ja autentifikācija nenotiek vai ja laukā *Proxy address* nav ierobežotas starpniekserveru adreses.
Proxy address	Ja ir norādīts, aktīva starpniekservera pieprasījumi tiek pieņemti tikai no šī ar komatiem atdalīto IP adrešu saraksta (pēc izvēles CIDR apzīmējumā) vai aktīva Zabbix starpniekservera DNS nosaukumiem. Šis lauks ir pieejams tikai tad, ja laukā *Proxy mode* ir atlasīts aktīvs starpniekserveris. Makro netiek atbalstīti.
Interface	Ievadiet pasīva starpniekservera saskarnes informāciju. 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 makro.

Parameter	Description
Description	Ievadiet starpniekservera aprakstu.

Cilne **Encryption** ļauj pieprasīt šifrētus savienojumus ar starpniekserveri.

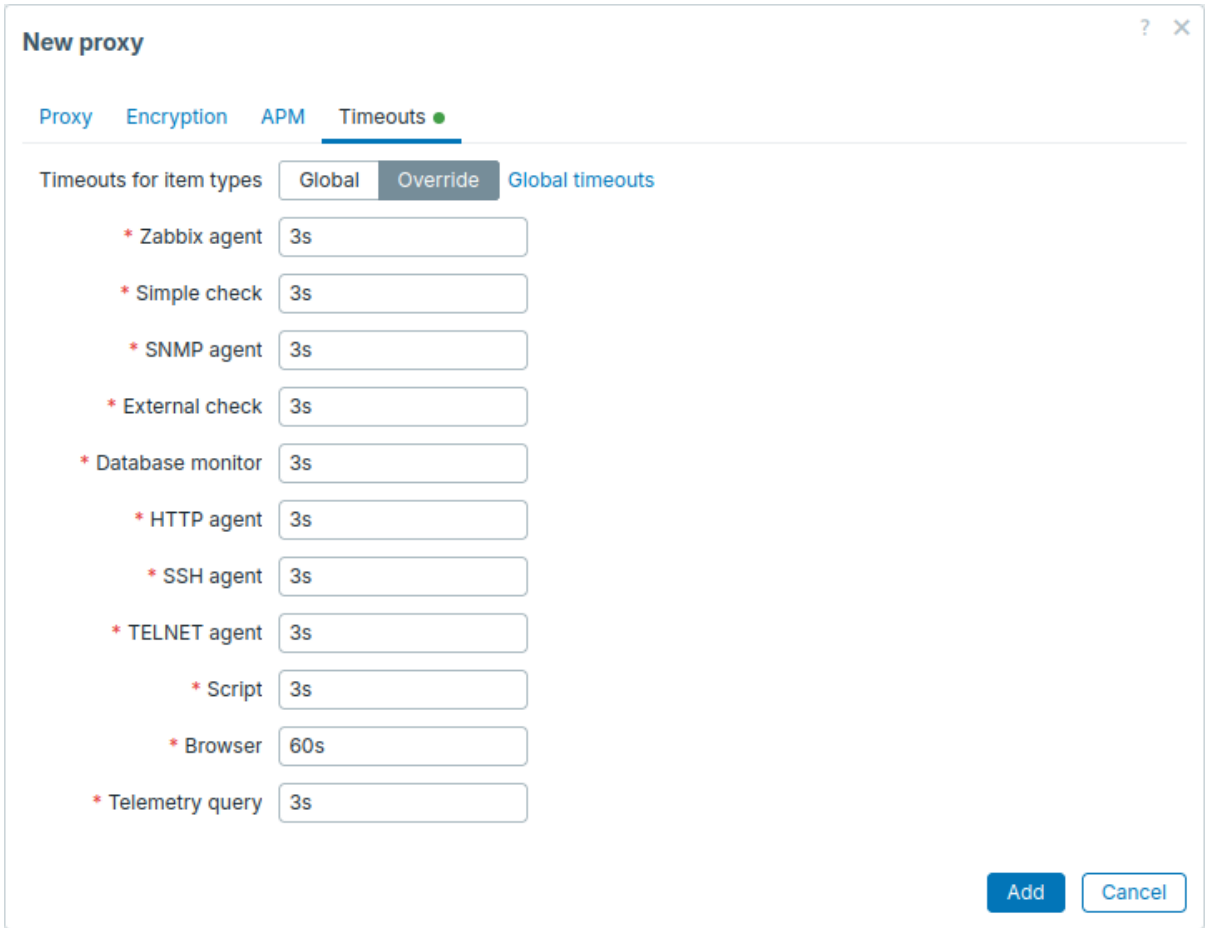
Parameter	Description
Connections to proxy	Kā serveris izveido savienojumu ar pasīvo starpniekserveri: bez šifrēšanas (pēc noklusējuma), izmantojot PSK (iepriekš koplietotu atslēgu) vai sertifikātu.
Connections from proxy	Atlasiet, kāda veida savienojumi ir atļauti no aktīvā starpniekservera. Vienlaikus var atlasīt vairākus savienojumu veidus (noderīgi testēšanai un pārejai uz citu savienojuma veidu). Pēc noklusējuma ir atlasīts "No encryption".
Issuer	Atļautais sertifikāta izdevējs. Vispirms sertifikāts tiek validēts ar CA (sertificēšanas iestādi). Ja tas ir derīgs un CA parakstīts, lauku *Issuer* var izmantot, lai vēl vairāk ierobežotu atļauto CA. Šis lauks nav obligāts; to paredzēts izmantot, ja jūsu Zabbix instalācijā tiek izmantoti vairāku CA sertifikāti.
Subject	Atļautais sertifikāta subjekts. Vispirms sertifikāts tiek validēts ar CA. Ja tas ir derīgs un CA parakstīts, lauku *Subject* var izmantot, lai atļautu tikai vienu *Subject* virknes vērtību. Ja šis lauks ir tukšs, tiek pieņemts jebkurš derīgs sertifikāts, ko parakstījusi konfigurētā CA.
PSK identity	Iepriekš koplietotas atslēgas identitātes virkne. Neievietojiet sensitīvu informāciju PSK identitātē, jo tā tiek pārsūtīta nešifrētā veidā tīklā, lai informētu saņēmēju, kuru PSK izmantot.
PSK	Iepriekš koplietota atslēga (heksadecimāla virkne). Maksimālais garums: 512 heksadecimālās zīmes (256 baitu PSK), ja Zabbix izmanto GnuTLS vai OpenSSL bibliotēku, vai 64 heksadecimālās zīmes (32 baitu PSK), ja Zabbix izmanto mbed TLS (PolarSSL) bibliotēku. Piemērs: 1f87b595725ac58dd977beef14b97461a7c1045b9a1c963065002c5473194952

Cilne **APM** ļauj konfigurēt lietojumprogrammu veiktspējas uzraudzību (APM) starpniekserverī. Starpniekserveris var apkopot lietojumprogrammu vai OpenTelemetry kolektoru nosūtītos telemetrijas datus (izsekojumus, metriku un žurnālus), izmantojot OpenTelemetry protokolu (OTLP/gRPC), un ierakstīt šos datus telemetrijas datu avotā.



Parameter	Description
Data collection enabled	Atzīmējiet izvēles rūtiņu, lai starpniekserverī iespējotu telemetrijas datu apkopošanu. Pirms telemetrijas datu apkopošanas iespējošanas starpniekservera konfigurācijas failā iestatiet `StartAPMCollectors` parametru uz 1.
Max messages per second	Iestatiet maksimālo ziņojumu skaitu sekundē, ko starpniekserveris pieņem: **Unlimited** - pieņemt visus ienākošos ziņojumus. **Custom** - pieņemt līdz norādītajam ziņojumu skaitam sekundē. Ziņojums ir OTLP pieprasījums, ko nosūta lietojumprogramma, OpenTelemetry kolektors vai cits telemetrijas avots. Tā lielums ir atkarīgs no sūtītāja; piemēram, OpenTelemetry kolektors vienā ziņojumā var apvienot tūkstošiem span, savukārt vienkārša lietojumprogramma var nosūtīt mazākus ziņojumus (piemēram, vienu katram izsekojumam). Ja ierobežojums tiek pārsniegts, starpniekserveris atbild ar gRPC statusa kodu `UNAVAILABLE` (īslaicīga kļūda), tāpēc sūtītājs atkarībā no konfigurācijas var mēģināt nosūtīt ziņojumu vēlāk.
Add resource attributes	Norādiet resursa atribūtus, kas jāpievieno atlasītā tipa ienākošajiem telemetrijas datiem (piemēram, lai atzīmētu visus šī starpniekservera saņemtos datus ar tā vidi vai reģionu). Pēc tam šos atribūtus varat izmantot telemetrijas datu filtrēšanai, piemēram, telemetrijas vaicājuma vienuma nosacījumos.
Name	Ievadiet atribūta nosaukumu (piemēram, `deployment.environment`, `cloud.region`). Tam jābūt unikālam atlasītajam *Type*.
Value	Ievadiet atribūta vērtību (piemēram, `production`, `eu-central-1`). Var būt tukša.
Type	Atlasiet ienākošo datu tipu, uz kuru attiecas atribūts: *Traces*, *Metrics* vai *Logs*.

Cilne **Timeouts** ļauj ignorēt globālos noildzes iestatījumus vienumu tipiem, kas tos atbalsta.



Parameter	Description
Timeouts for item types	Iestatiet vienuma noildzi (atkarībā no tā tipa): **Global** - izmantot globālo noildzi (kas tiek parādīta pelēkajā *Timeout* laukā pie katra vienuma tipa); **Override** - iestatīt pielāgotu noildzi (katra vienuma tipa *Timeout* laukā). Atļautais diapazons: 1–600 s (pēc noklusējuma: mantots no globālajiem noildzes iestatījumiem). Tiek atbalstīti laika sufiksi, piemēram, 30s, 1m, un lietotāja makro. Noklikšķinot uz saites *Global timeouts*, varat konfigurēt globālos noildzes iestatījumus. Ņemiet vērā, ka saite *Global timeouts* ir redzama tikai *Super admin* tipa lietotājiem, kuriem ir atļaujas sadaļai *Administration* > *General* lietotāja saskarnē. Ņemiet vērā, ka, lai gan starpniekservera līmeņa noildzes iestatījumi ignorē globālos iestatījumus, tos ignorēs atsevišķu vienumu noildzes iestatījumi, ja tie ir konfigurēti.

Note:

Ja starpniekservera galvenā versija neatbilst servera galvenajai versijai, 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 `Timeout` parametru no starpniekservera konfigurācijas faila.

Esoša starpniekservera rediģēšanas veidlapā ir pieejamas šādas papildu pogas:

- *Refresh configuration* - atsvaidzināt starpniekservera konfigurāciju;
- *Clone* - izveidot jaunu starpniekserveri, pamatojoties uz esošā starpniekservera rekvizītiem;
- *Delete* - dzēst starpniekserveri.

Hosts konfigurācija

Varat norādīt, ka atsevišķs hosts jāuzrauga starpniekserverim vai starpniekserveru grupai, izmantojot hosts konfigurācijas formu un lauku *Monitored by*.



Hosts masveida atjaunināšana ir vēl viens veids, kā norādīt, ka hosts jāuzrauga starpniekserverim vai starpniekserveru grupai.

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 hosts automātiska pārdale starp starpniekserveriem starpniekserveru grupas ietvaros:

- Ja starpniekserveris kļūst bezsaistē, tā hosts tiks pārvietoti uz citiem starpniekserveriem, tādējādi saglabājot augstu starpniekserveru pieejamību.
- Ja vienam starpniekserverim ir ievērojami lielāks/mazāks hosts skaits nekā citiem starpniekserveriem, tā hosts tiks pārvietoti uz citiem starpniekserveriem, lai līdzsvarotu starpniekserveru slodzi.

Hosts pārdale darbojas tikai starp grupā esošiem starpniekserveriem, kas atbilst šādiem nosacījumiem:

- Starpniekserveri darbojas ar Zabbix 7.0 vai jaunāku versiju.
- Starpniekservera versija atbilst Zabbix servera versijai. Ja tiek izmantots Zabbix aģents (pasīvs), starpniekservera versijai jāsakrīt ar aģenta versiju. Aktīvajiem aģentiem nepieciešams tikai Zabbix 7.0 vai jaunāks.
- Starpniekserveru grupai ir tiešsaistes statuss.
- Hosts ir konfigurēti tā, lai tos uzraudzītu starpniekserveru grupa, nevis atsevišķi starpniekserveri.

Note:
Starpniekserveru grupas veselību var uzraudzīt ar iekšējām pārbaudēm jebkurš hosts, kas piešķirts starpniekserveru grupai. Tomēr, lai uzraudzītu viena grupas starpniekservera veselību, piešķiriet hostu šim starpniekserverim; pretējā gadījumā rezultāti var būt nekonsekventi.

Hosts pārdale

Starpniekserveru slodzes balansēšanu un augstu pieejamību Zabbix serveris pārvalda, izmantojot starpniekserveru grupas pārvaldnieku, kas nepārtraukti uzrauga visu starpniekserveru stāvokli katrā starpniekserveru grupā un to hosts sadalījumu.

Starpniekserveru augstā pieejamība grupas ietvaros tiek nodrošināta ar starpniekservera atteici: kad starpniekserveris kļūst nepieejams, tā hosts nekavējoties tiek pārdalīti citiem starpniekserveriem. Notiek arī starpniekserveru slodzes balansēšana, jo hosts tiek pārdalīti uz starpniekserveriem ar vismazāko piešķirto hosts skaitu.

Papildus tam starpniekserveru slodzes balansēšana tiek aktivizēta, ja starpniekservera hosts skaits atšķiras no grupas vidējā vismaz par 10 hosts un 2 reizes (hosts pārpalikums vai hosts deficīts). Ja nelīdzsvarotība saglabājas pēc labvēlības perioda (10 x atteices aizkave), starpniekserveru grupa tiek ievietota rindā hosts pārdalei.

Starpniekserveru grupas pārvaldnieks pārdala hosts, izmantojot šādu loģiku:

1. Aprēķina vidējo hosts skaitu uz vienu starpniekserveri.
2. Starpniekserveriem ar hosts pārpalikumu — pārvieto liekos hosts uz starpniekserveru nepiešķirto kopu.
3. Starpniekserveriem ar hosts deficītu — aprēķina, cik hosts ir nepieciešami līdzsvara sasniegšanai.
4. Noņem nepieciešamo hosts skaitu no starpniekserveriem ar vislielāko hosts skaitu.
5. Pārvieto nepiešķirtos hosts uz starpniekserveriem ar vismazāko hosts skaitu.

Hosts pārdales piemēri:

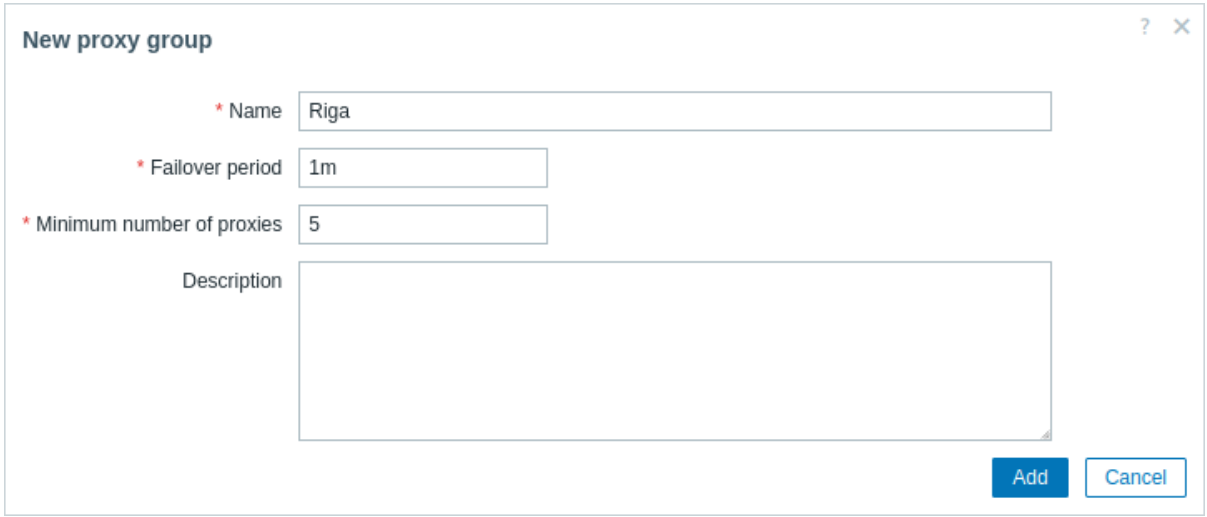
Hosts on proxy	Group average	Host reassignment
100	50	Yes
60	50	No
40	50	No
25	50	Yes
15	5	Yes
10	5	No

Attention:
Ja starpniekserveru grupā ir mazāk nekā 10 hosts uzraudzīti, tas var novest pie nevienmērīga hosts sadalījuma starp grupas starpniekserveriem.

Starpniekserveru grupas konfigurēšana

Lai konfigurētu starpniekserveru grupu Zabbix lietotāja saskarnē:

1. Atveriet *Administration* > *Proxy groups*.
2. Noklikšķiniet uz *Create proxy group*.
3. Veidlapā ievadiet starpniekserveru grupas parametrus.



Parameter	Description
Name	Starpniekserveru grupas nosaukums.
Failover period	Periods sekundēs, kura laikā grupas starpniekserverim ir jāsazinās ar Zabbix serveri, lai tas tiktu uzskatīts par tiešsaistē esošu (noklusējums: 1m; diapazons: 10s-15m). Ja starpniekserveris šajā periodā nesazinās, starpniekservera stāvoklis tiek mainīts uz *Offline*, un tā hosts tiek nekavējoties pārdalīti citiem starpniekserveriem. Starpniekserveru slodzes balansēšana sākas pēc 10 šādiem periodiem. Atbalsta laika sufiksus (piemēram, 30s, 1m) un lietotāja makro.
Minimum number of proxies	Minimālais tiešsaistē esošo starpniekserveru skaits, kas nepieciešams, lai starpniekserveru grupa būtu tiešsaistē (noklusējums: 1; diapazons: 1-1000). Atbalsta lietotāja makro. Šai vērtībai jābūt mazākai par kopējo starpniekserveru skaitu grupā. Piemēram, grupā ar 10 starpniekserveriem, iestatot minimālo skaitu uz 10, grupa pāries bezsaistes stāvoklī, ja kāds starpniekserveris pārstās darboties. Ņemiet vērā, ka tiešsaistē esošie starpniekserveri bezsaistes grupā turpina darboties normāli, taču slodzes balansēšana/augsta pieejamība nenotiks.
Description	Starpniekserveru grupas apraksts.
Proxies	Rediģējot grupu, kurā ir starpniekserveri, tiek parādīts saraksts ar ne vairāk kā pieciem starpniekserveriem (kā saites vai vienkāršā tekstā atkarībā no lietotāja atļaujām piekļūt starpniekserveriem).

Starpniekservera slodzes balansēšanas konfigurēšana

Lai izmantotu starpniekservera slodzes balansēšanu, Zabbix lietotāja saskarnē ir jākonfigurē starpniekserveru grupa (skatiet iepriekš) un jāpārliecinās, ka hosts tiek uzraudzīti ar starpniekserveru grupu, nevis ar atsevišķiem starpniekserveriem (varat izmantot hostu masveida atjaunināšanu, lai pārvietotu hostus no starpniekserveriem uz starpniekserveru grupu).

Ja tiek izmantots Zabbix aģents, konfigurējiet to arī šādi:

- Pasīvajām pārbaudēm parametram Server norādiet visus starpniekserveru grupas starpniekserverus.
- Aktīvajām pārbaudēm ieteicams parametram 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 parametrs `ServerActive` ietver tikai vienu starpniekserveri no starpniekserveru grupas (vai Zabbix serveri), aģents joprojām varēs izveidot savienojumu ar pareizo starpniekserveri. Kad aģenta pakalpojums 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 pāradresētas uz pareizo tiešsaistes starpniekserveri konkrētajam hostam, pamatojoties uz pašreizējo starpniekservera un hosta piešķīrumu 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 pieejams.

Note:
Izmantojot Zabbix sender, datu pieprasījumi arī tiek pāradresēti uz pareizo tiešsaistes starpniekserveri konkrētajam hostam, pamatojoties uz pašreizējo starpniekservera un hosta piešķīrumu starpniekserveru grupā. Tomēr, ja no ievades faila sūtāt vairāku hostu vērtības, izmantojiet `-g` opciju, lai novērstu datu nosūtīšanu uz nepareizo starpniekserveri.

Zabbix aģentam arī jāspēj caur ugunsmūri izveidot savienojumu ar visiem starpniekserveriem starpniekserveru grupā. Pretējā gadījumā aktīvās pārbaudes var iestrēgt vai neizdoties pāradresācijas vai atteices pārslēgšanas laikā. Piemēram:

- Aktīvo pārbaužu laikā starpniekserveris var pāradresēt aģentu uz citu starpniekserveri. Ja šis starpniekserveris ir bloķēts ar ugunsmūri, saziņa iestrēgs, gaidot atbildi.
- Stabilās augstas pieejamības konfigurācijās bez nesenas līdzsvarošanas aģenti, iespējams, nekad nesazināsies ar rezerves starpniekserveriem. Ja ugunsmūra noteikumi ir mainīti un nav pārbaudīti, atteices pārslēgšana var neizdoties.

Starpniekservera slodzes balansēšanas testēšana

Lai pārbaudītu starpniekservera slodzes balansēšanu:

1. Konfigurējiet starpniekserveru grupu.
2. Pārliecinieties, ka starpniekserveru grupai ir tiešsaistes statuss.
3. Pārliecinieties, ka hosts tiek uzraudzīti ar starpniekserveru grupu, nevis atsevišķiem starpniekserveriem (varat izmantot hostu masveida atjaunināšanu, lai pārvietotu hostus no starpniekserveriem uz starpniekserveru grupu).

4. Pagaidiet dažas sekundes konfigurācijas atjaunināšanai un hostu sadalei starp starpniekserveriem starpniekserveru grupā. Novērojiet izmaiņas, atsvaidzinot hostu sarakstu *Administrēšana* > *Starpniekserveri*.

Svarīgas piezīmes

- SNMP trapi netiek atbalstīti ar starpniekserveriem starpniekserveru grupā.
- Pārbaudēm, kas ir atkarīgas no ārējas konfigurācijas (piem., skriptiem ārējām pārbaudēm vai ODBC konfigurācijas datubāzes pārbaudēm), starpniekserveru grupas visos starpniekserveros jābūt vienādai konfigurācijai.
- Datubāzes pārbaudēm ir nepieciešamas paplašinātas atļaujas datubāzes objektam/serverim.
- VMware hosts, ko uzrauga starpniekserveru grupa, tiks nejauši sadalīti starp grupas starpniekserveriem. Tas liek katram starpniekserverim kešot visus VMware datus, radot papildu slodzi vCenter.
- Hosts, kas izveidoti, pamatojoties uz automātiskās reģistrācijas datiem no starpniekservera starpniekserveru grupā, tiek iestatīti kā uzraugāmi šajā starpniekserveru grupā. Tomēr hosts, kas izveidoti, pamatojoties uz tīkla atklāšanas datiem no starpniekservera starpniekserveru grupā, tiek iestatīti kā uzraugāmi šajā starpniekserverī.

Uzraudzības konfigurācijas sinhronizācija

Pārskats

Šī lapa sniedz informāciju par starpniekservera uzraudzības konfigurācijas atjaunināšanu, t. i., par to, kā serverī veiktās izmaiņas uzraudzības konfigurācijā tiek sinhronizētas ar starpniekserveri.

Inkremetāls atjauninājums

Starpniekservera konfigurācijas atjaunināšana ir inkrementāla. 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 atjaunināšanai.

Starpniekservera konfigurācijas izmaiņas tiek izsekotas, izmantojot pārskatīšanas numurus. Konfigurācijas datos, kas tiek nosūtīti starpniekserverim, tiek iekļautas tikai entītijas, kuru pārskatīšanas numuri ir lielāki par starpniekservera konfigurācijas pārskatīšanas numuru.

Konfigurācijas sinhronizācijai tiek izmantotas šādas entītijas:

Entītija	Detalizēta informācija
automātiskās reģistrācijas TLS dati	Visi automātiskās reģistrācijas TLS dati.
izteiksmes	Visas izteiksmes (regulārās izteiksmes, izteiksmju tabulas).
globālā konfigurācija	Globālā konfigurācija, kas definēta tabulā 'config'
hosts	Visi hosts rekvizīti, saskarnes, inventārs, vienumi, vienumu priekšapstrāde, vienumu parametri un hosts tīmekļa scenāriji.
host makro	Visi host definētie makro un visi ar to saistītie veidņu ID.
starpniekservera atklāšanas noteikums	Atklāšanas noteikumi un pārbaudes, kas piešķirtas starpniekserverim.

Tas nozīmē:

- Ja vienums tiek mainīts **host**, tiks sinhronizēta visa šī host konfigurācija.
- Ja tiek mainīta **regulārā izteiksme**, tiks sinhronizētas visas regulārās izteiksmes.

Izņēmums ir host makro, kas tiek nosūtīti arī tad, ja host ir veiktas jebkādas izmaiņas.

Komanda `-R config_cache_reload` starpniekserverī arī sāk inkrementālu atjaunināšanu.

Ņemiet vērā, ka pilna konfigurācijas sinhronizācija tiks veikta, startējot vai restartējot starpniekserveri, HA pārslēgšanās laikā, ja sesijas marķieris ir mainījies vai ja konfigurācijas atjaunināšana starpniekserverī 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

Aktīvajiem starpniekserveriem ProxyConfigFrequency ir jauns parametrs kopš Zabbix 6.4, un tas jāizmanto ConfigFrequency vietā, kas tagad ir novecojis.

Attention:
Ja tiek izmantoti gan ProxyConfigFrequency, gan ConfigFrequency, starpniekserveris reģistrēs kļūdu un tiks apturēts.

Rindas aprēķins apkopes laikā

Attention:
Zabbix starpniekserveris nav informēts par apkopes periodiem; sīkāku informāciju skatiet sadaļā Rindas aprēķins apkopes laikā.

Servera-starpniekservera datu apmaiņas protokols

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 nosūta tukšu `proxy config` pieprasījumu. Šis pieprasījums tiek nosūtīts ik pēc `ProxyConfigFrequency` (servera konfigurācijas parametra) sekundēm.

Starpniekserveris atbild ar pašreizējo starpniekservera versiju, sesijas pilnvaru un konfigurācijas revīziju. Serveris atbild ar konfigurācijas datiem, kas 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 pilnvara.
config_revision	*number*	Starpniekservera konfigurācijas revīzija.
server→proxy:		
full_sync	*number*	1 - ja tiek nosūtīti pilni konfigurācijas dati; citādi nav ietverts (neobligāts).
data	*array*	Tabulu datu objekts. Nav ietverts, ja konfigurācija nav mainīta (neobligāts).
<table>	*object*	Viens vai vairāki objekti ar <table> datiem (neobligāts, 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 makro informācija; nav ietverta, ja glabātavas makro nav mainīti (neobligāts).
config_revision	*number*	Konfigurācijas kešatmiņas revīzija - tiek nosūtīta kopā ar konfigurācijas datiem (neobligāta).
del_hostids	*array*	Noņemto hosts identifikatoru masīvs (neobligāts).
-	*number*	Hosts identifikators.
del_macro_hostids	*array*	Hosts identifikatoru masīvs, kuriem noņemti visi makro (neobligāts).
-	*number*	Hosts identifikators.
proxy→server:		
response	*string*	Informācija par pieprasījuma izpildi ('success' vai 'failed').

name	value type	description
version	*string*	Starpniekservera versija (<major>.<minor>.<build>).

Piemērs:

serveris→starpniekserveris:

```
{
  "request":"proxy config"
}
```

starpniekserveris→serveris:

```
{
  "version": "8.0.0",
  "session": "0033124949800811e5686dbfd9bcea98",
  "config_revision": 0
}
```

serveris→starpniekserveris:

```
{
    "full_sync": 1,
    "data": {
        "hosts": {
            "fields": ["hostid", "host", "status", "ipmi_authtype", "ipmi_privilege", "ipmi_username", "ip
            "data": [
                [10084, "Zabbix server", 0, -1, 2, "", "", "Zabbix server", 1, 1, "", "", "", ""]
            ]
        },
        "interface": {
            "fields": ["interfaceid", "hostid", "main", "type", "useip", "ip", "dns", "port", "available"]
            "data": [
                [1, 10084, 1, 1, 1, "127.0.0.1", "", "10053", 1]
            ]
        },
        "interface_snmp": {
            "fields": ["interfaceid", "version", "bulk", "community", "securityname", "securitylevel", "au
            "data": []
        },
        "host_inventory": {
            "fields": ["hostid", "type", "type_full", "name", "alias", "os", "os_full", "os_short", "seria
            "data": [
                [10084, "", "", ..., "56.95387", "24.22067", ..., ""]
            ]
        },
        "items": {
            "fields": ["itemid", "type", "snmp_oid", "hostid", "key_", "delay", "history", "status", "valu
            "data": [
                [44161, 7, "", 10084, "agent.hostmetadata", "10s", "90d", 0, 1, "", "", "", "", 0, "", "",
                [44162, 0, "", 10084, "agent.ping", "10s", "90d", 0, 3, "", "", "", "", 0, "", "", "", "",
            ]
        },
        "item_rtdata": {
            "fields": ["itemid", "lastlogsize", "mtime"],
            "data": [
                [44161, 0, 0],
                [44162, 0, 0]
            ]
        },
        "item_preproc": {
            "fields": ["item_preprocid", "itemid", "step", "type", "params", "error_handler", "error_handl
            "data": []
        },
```

```
        "item_parameter": {
            "fields": ["item_parameterid", "itemid", "name", "value"],
            "data": []
        },
        "globalmacro": {
            "fields": ["globalmacroid", "macro", "value", "type"],
            "data": [
                [2, "{$SNMP_COMMUNITY}", "public", 0]
            ]
        },
        "hosts_templates": {
            "fields": ["hosttemplateid", "hostid", "templateid", "link_type"],
            "data": []
        },
        "hostmacro": {
            "fields": ["hostmacroid", "hostid", "macro", "value", "type", "automatic"],
            "data": [
                [5676, 10084, "{$M}", "AppID=zabbix_server&Query=Safe=passwordSafe;Object=zabbix:Content",
            ]
        },
        "drules": {
            "fields": ["druleid", "name", "iprange", "delay"],
            "data": [
                [2, "Local network", "127.0.0.1", "10s"]
            ]
        },
        "dchecks": {
            "fields": ["dcheckid", "druleid", "type", "key_", "snmp_community", "ports", "snmpv3_securityn
            "data": [
                [2, 2, 9, "system.uname", "", "10052", "", 0, "", "", 0, 0, 0, "", 1, 0]
            ]
        },
        "regexps": {
            "fields": ["regexpid", "name"],
            "data": [
                [1, "File systems for discovery"],
                [2, "Network interfaces for discovery"],
                [3, "Storage devices for SNMP discovery"],
                [4, "Windows service names for discovery"],
                [5, "Windows service startup states for discovery"]
            ]
        },
        "expressions": {
            "fields": ["expressionid", "regexpid", "expression", "expression_type", "exp_delimiter", "case
            "data": [
                [1, 1, "^(btrfs|ext2|ext3|ext4|reiser|xfs|ffs|ufs|jfs|jfs2|vxfs|hfs|apfs|refs|ntfs|fat32|z
                [3, 3, "^(Physical memory|Virtual memory|Memory buffers|Cached memory|Swap space)$", 4, ",
                [5, 4, "^(MMCSS|gupdate|SysmonLog|clr_optimization_v2.0.50727_32|clr_optimization_v4.0.303
                [6, 5, "^(automatic|automatic delayed)$", 3, ",", 1],
                [7, 2, "^Software Loopback Interface", 4, ",", 1],
                [8, 2, "^(In)?[Ll]oop[Bb]ack[0-9._]*$", 4, ",", 1],
                [9, 2, "^NULL[0-9.]*$", 4, ",", 1],
                [10, 2, "^[Ll]o[0-9.]*$", 4, ",", 1],
                [11, 2, "^[Ss]ystem$", 4, ",", 1],
                [12, 2, "^Nu[0-9.]*$", 4, ",", 1]
            ]
        },
        "settings": {
            "fields": ["name", "type", "value_str", "value_int"],
            "data": [
                ["autoreg_tls_accept", 2, "", 1],
                ["hk_history_global", 2, "", 0],
```

```
                ["snmptrap_logging", 2, "", 1],
                ["proxy_secrets_provider", 2, "", 0],
                ["hk_history", 1, "31d", 0],
                ["timeout_db_monitor", 1, "3s", 0],
                ["timeout_external_check", 1, "3s", 0],
                ["timeout_http_agent", 1, "3s", 0],
                ["timeout_simple_check", 1, "3s", 0],
                ["timeout_snmp_agent", 1, "3s", 0],
                ["timeout_ssh_agent", 1, "3s", 0],
                ["timeout_telnet_agent", 1, "3s", 0],
                ["timeout_zabbix_agent", 1, "3s", 0],
                ["timeout_browser", 1, "30s", 0]
            ]
        },
        "httptest": {
            "fields": ["httptestid", "name", "delay", "agent", "authentication", "http_user", "http_passwo
            "data": []
        },
        "httptestitem": {
            "fields": ["httptestitemid", "httptestid", "itemid", "type"],
            "data": []
        },
        "httptest_field": {
            "fields": ["httptest_fieldid", "httptestid", "type", "name", "value"],
            "data": []
        },
        "httpstep": {
            "fields": ["httpstepid", "httptestid", "name", "no", "url", "timeout", "posts", "required", "s
            "data": []
        },
        "httpstepitem": {
            "fields": ["httpstepitemid", "httpstepid", "itemid", "type"],
            "data": []
        },
        "httpstep_field": {
            "fields": ["httpstep_fieldid", "httpstepid", "type", "name", "value"],
            "data": []
        },
        "config_autoreg_tls": {
            "fields": ["autoreg_tlsid", "tls_psk_identity", "tls_psk"],
            "data": [
                [1, "", ""]
            ]
        }
    },
    "macro.secrets": {
        "AppID=zabbix_server&Query=Safe=passwordSafe;Object=zabbix": {
            "Content": "738"
        }
    },
    "config_revision": 2
}
```


starpniekserveris→serveris:


```
{
  "response": "success",
  "version": "8.0.0"
}
```


Datu pieprasījums

`proxy data` pieprasījums tiek izmantots, lai iegūtu host interfeisa pieejamības, vēsturiskos, atklāšanas un automātiskās reģistrācijas datus no starpniekservera. Šis pieprasījums tiek nosūtīts ik pēc `ProxyDataFrequency` (servera konfigurācijas parametra)

sekundēm.

name	value type	description
server→proxy:		
request	*string*	'proxy data'
proxy→server:		
session	*string*	Datu sesijas marķieris.
interface	*array*	*(optional)* Interfeisa pieejamības datu objektu masīvs.
avail-		
abil-		
ity		
interfaceid	*number*	Interfeisa identifikators.
available	*number*	Interfeisa pieejamība: **0**, *INTERFACE_AVAILABLE_UNKNOWN* - nezināms **1**, *INTERFACE_AVAILABLE_TRUE* - pieejams **2**, *INTERFACE_AVAILABLE_FALSE* - nav pieejams
error	*string*	Interfeisa kļūdas ziņojums vai tukša virkne.
history	*array*	*(optional)* Vēsturisko datu objektu masīvs.
data		
itemid	*number*	Vienuma identifikators.
clock	*number*	Vienuma vērtības laikspiedols (sekundēs).
ns	*number*	Vienuma vērtības laikspiedols (nanosekundēs).
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)* Žurnāla tipa vienumu pēdējais žurnāla izmērs.
mtime	*number*	*(optional)* Žurnāla tipa vienumu modificēšanas laiks.
discovery	*array*	*(optional)* Atklāšanas datu objektu masīvs.
data		
clock	*number*	Atklāšanas datu laikspiedols.
druleid	*number*	Atklāšanas noteikuma identifikators.
dcheckid	*number*	Atklāšanas pārbaudes identifikators vai null atklāšanas noteikuma datiem.
type	*number*	Atklāšanas pārbaudes tips: **-1** atklāšanas noteikuma dati **0**, *SVC_SSH* - SSH pakalpojuma pārbaude **1**, *SVC_LDAP* - LDAP pakalpojuma pārbaude **2**, *SVC_SMTP* - SMTP pakalpojuma pārbaude **3**, *SVC_FTP* - FTP pakalpojuma pārbaude **4**, *SVC_HTTP* - HTTP pakalpojuma pārbaude **5**, *SVC_POP* - POP pakalpojuma pārbaude **6**, *SVC_NNTP* - NNTP pakalpojuma pārbaude **7**, *SVC_IMAP* - IMAP pakalpojuma 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 pakalpojuma pārbaude **15**, *SVC_TELNET* - Telnet pieejamības pārbaude
ip	*string*	Host IP adrese.
dns	*string*	Host DNS nosaukums.
port	*number*	*(optional)* Pakalpojuma porta numurs.
key_	*string*	*(optional)* Vienuma atslēga atklāšanas pārbaudei ar tipu **9** *SVC_AGENT*

name	value type	description
value	*string*	*(optional)* No pakalpojuma saņemtā vērtība, vairumam pakalpojumu var būt tukša.
status	*number*	*(optional)* Pakalpojuma statuss: **0**, *DOBJECT_STATUS_UP* - pakalpojums darbojas **1**, *DOBJECT_STATUS_DOWN* - pakalpojums nedarbojas
auto reg- is- tra- tion	*array*	*(optional)* Automātiskās reģistrācijas datu objektu masīvs.
clock	*number*	Automātiskās reģistrācijas datu laikspiedols.
host	*string*	Host nosaukums.
ip	*string*	*(optional)* Host IP adrese.
dns	*string*	*(optional)* No IP adreses atrisināts DNS nosaukums.
port	*string*	*(optional)* Host ports.
host_metadata	*string*	*(optional)* Host metadati, ko nosūta 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* - attālā komanda rezultāts
status	*number*	Attālās komandas izpildes statuss: **0**, *ZBX_TM_REMOTE_COMMAND_COMPLETED* - attālā komanda veiksmīgi pabeigta **1**, *ZBX_TM_REMOTE_COMMAND_FAILED* - attālā komanda neizdevās
error	*string*	*(optional)* Kļūdas ziņojums.
parent_taskid	*number*	Vecāka uzdevuma ID.
more	*number*	*(optional)* 1 - ir vēl vēsturiskie dati, ko nosūtīt.
clock	*number*	*(optional)* Datu pārsūtīšanas laikspiedols (sekundēs).
ns	*number*	*(optional)* Datu pārsūtīšanas laikspiedols (nanosekundēs).
version	*string*	Starpniekservera versija (<major>.<minor>.<build>).
server→proxy:		
response	*string*	Pieprasījuma izpildes informācija (’success’ vai ’failed’).
tasks	*array*	*(optional)* Uzdevumu masīvs.
type	*number*	Uzdevuma tips: **1**, *ZBX_TM_TASK_PROCESS_REMOTE_COMMAND* - attālā komanda
clock	*number*	Uzdevuma izveides laiks.
ttl	*number*	Laiks sekundēs, pēc kura uzdevums beidzas.
commandtype	*number*	Attālās 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 globālo skriptu (šobrīd funkcionāli ekvivalents pielāgotam skriptam)
command	*string*	Izpildāmā attālā komanda.
execute_on	*number*	Izpildes mērķis pielāgotiem skriptiem: **0**, *ZBX_SCRIPT_EXECUTE_ON_AGENT* - izpildīt skriptu uz aģents **1**, *ZBX_SCRIPT_EXECUTE_ON_SERVER* - izpildīt skriptu uz serveris **2**, *ZBX_SCRIPT_EXECUTE_ON_PROXY* - izpildīt skriptu uz starpniekserveris
port	*number*	*(optional)* Ports Telnet un SSH komandām.
authtype	*number*	*(optional)* SSH komandu autentifikācijas tips.
username	*string*	*(optional)* Lietotājvārds Telnet un SSH komandām.
password	*string*	*(optional)* Parole Telnet un SSH komandām.
publickey	*string*	*(optional)* Publiskā atslēga SSH komandām.

name	value type	description
privatekey	*string*	*(optional)* Privātā atslēga SSH komandām.
parent_taskid	*number*	Vecāka uzdevuma ID.
hostid	*number*	Mērķa host 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 attālo komandu uzdevumu piegādi un apstiprināšanu starp serveri un pasīvo starpniekserveri. Kad serveris aptaujā starpniekserveri par uzdevumu rezultātiem, tas nosūta tukšu `proxy tasks` pieprasījumu intervālā, kas definēts ar `ProxyDataFrequency`. Starpniekserveris atbild ar savu pašreizējo versiju un visiem gaidošajiem uzdevumu rezultātiem (tostarp laikspiedoliem). Pēc tam serveris apstiprina saņemšanu, atgriežot `response`, un var iekļaut jaunus uzdevumus, ko starpniekserverim izpildīt.

name	value type	description
server→proxy:		
request	*string*	'proxy tasks'
proxy→server:		
version	*string*	Starpniekservera versija (<major>.<minor>.<build>).
clock	*number*	*(optional)* Datu pārsūtīšanas laikspiedols (sekundes).
ns	*number*	*(optional)* Datu pārsūtīšanas laikspiedols (nanosekundes).
server→proxy:		
response	*string*	Pieprasījuma izpildes informācija ('success' or 'failed').

Piemērs:

server→proxy:

```
{
  "request":"proxy tasks"
}
```

proxy→server:

```
{
  "version":"7.0.0",
  "clock":1721059872,
  "ns":848141522
}
```

server→proxy:

```
{
  "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
starpniekserveris→serveris:		
request	*string*	'proxy config'
host	*string *	Hosts nosaukums.
version	*string*	Hosts versija (<major>.<minor>.<build>).
session	*string*	Hosts konfigurācijas sesijas marķieris.
config_revision	*number*	Hosts konfigurācijas revīzija.
hostmap_revision	*number*	Pašreizējā hosts un starpniekservera kartējuma revīzija.
serveris→starpniekserveris:		
full_sync	*number*	1 - ja tiek nosūtīti pilni konfigurācijas dati; pretējā gadījumā nav ietverts (neobligāts).
data	*array*	Tabulu datu objekts. Nav ietverts, ja konfigurācija nav mainīta (neobligāts).
<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 makro informācija; nav ietverta, ja glabātavas makro nav mainīti (neobligāts).
proxy_group	*string*	Starpniekserveru grupas nosaukums, kurai pieder starpniekserveris.
config_revision	*number*	Konfigurācijas kešatmiņas revīzija - tiek nosūtīta kopā ar konfigurācijas datiem (neobligāta).
del_hostids	*array*	Noņemto hostid masīvs (neobligāts).
-	*number*	Hosts identifikators.
del_macro_hostids	*array*	hostid masīvs, kuriem noņemti visi makro (neobligāts).
-	*number*	Hosts identifikators.

Piemērs:

starpniekserveris→serveris:

```
{
  "request": "proxy config",
  "host": "Zabbix proxy",
  "version":"8.0.0",
  "session": "fd59a09ff4e9d1fb447de1f04599bcf6",
  "config_revision": 0
}
```

serveris→starpniekserveris:

```
{
"full_sync": 1,
"data": {
"hosts": {
"fields": ["hostid", "host", "status", "ipmi_authtype", "ipmi_privilege", "ipmi_username", "ipmi_password"
"data": [
[10084, "Zabbix server", 0, -1, 2, "", "", "Zabbix server", 1, 1, "", "", "", ""]
]
},
"interface": {
"fields": ["interfaceid", "hostid", "main", "type", "useip", "ip", "dns", "port", "available"],
"data": [
[1, 10084, 1, 1, 1, "127.0.0.1", "", "10053", 1]
]
},
```

```
"interface_snmp": {
"fields": ["interfaceid", "version", "bulk", "community", "securityname", "securitylevel", "authpassphrase
"data": []
},
"host_inventory": {
"fields": ["hostid", "type", "type_full", "name", "alias", "os", "os_full", "os_short", "serialno_a", "ser
"data": [
[10084, "", "", ..., "56.95387", "24.22067", ..., ""]
]
},
"items": {
"fields": ["itemid", "type", "snmp_oid", "hostid", "key_", "delay", "history", "status", "value_type", "tr
"data": [
[44161, 7, "", 10084, "agent.hostmetadata", "10s", "90d", 0, 1, "", "", "", "", 0, "", "", "", "", 0, null
[44162, 0, "", 10084, "agent.ping", "10s", "90d", 0, 3, "", "", "", "", 0, "", "", "", "", 0, 1, 0, "", nu
]
},
"item_rtdata": {
"fields": ["itemid", "lastlogsize", "mtime"],
"data": [
[44161, 0, 0],
[44162, 0, 0]
]
},
"item_preproc": {
"fields": ["item_preprocid", "itemid", "step", "type", "params", "error_handler", "error_handler_params"],
"data": []
},
"item_parameter": {
"fields": ["item_parameterid", "itemid", "name", "value"],
"data": []
},
"globalmacro": {
"fields": ["globalmacroid", "macro", "value", "type"],
"data": [
[2, "{$SNMP_COMMUNITY}", "public", 0]
]
},
"hosts_templates": {
"fields": ["hosttemplateid", "hostid", "templateid", "link_type"],
"data": []
},
"hostmacro": {
"fields": ["hostmacroid", "hostid", "macro", "value", "type", "automatic"],
"data": [
[5676, 10084, "{$M}", "AppID=zabbix_server&Query=Safe=passwordSafe;Object=zabbix:Content", 2, 0]
]
},
"drules": {
"fields": ["druleid", "name", "iprange", "delay"],
"data": [
[2, "Local network", "127.0.0.1", "10s"]
]
},
"dchecks": {
"fields": ["dcheckid", "druleid", "type", "key_", "snmp_community", "ports", "snmpv3_securityname", "snmpv
"data": [
[2, 2, 9, "system.uname", "", "10052", "", 0, "", "", 0, 0, 0, "", 1, 0]
]
},
"regexps": {
"fields": ["regexpid", "name"],
```

```
"data": [
[1, "File systems for discovery"],
[2, "Network interfaces for discovery"],
[3, "Storage devices for SNMP discovery"],
[4, "Windows service names for discovery"],
[5, "Windows service startup states for discovery"]
]
},
"expressions": {
"fields": ["expressionid", "regexpid", "expression", "expression_type", "exp_delimiter", "case_sensitive"]
"data": [
[1, 1, "^(btrfs|ext2|ext3|ext4|reiser|xfs|ffs|ufs|jfs|jfs2|vxfs|hfs|apfs|refs|ntfs|fat32|zfs)$", 3, ",", 0
[3, 3, "^(Physical memory|Virtual memory|Memory buffers|Cached memory|Swap space)$", 4, ",", 1],
[5, 4, "^(MMCSS|gupdate|SysmonLog|clr_optimization_v2.0.50727_32|clr_optimization_v4.0.30319_32)$", 4, ","
[6, 5, "^(automatic|automatic delayed)$", 3, ",", 1],
[7, 2, "^Software Loopback Interface", 4, ",", 1],
[8, 2, "^(In)?[Ll]oop[Bb]ack[0-9._]*$", 4, ",", 1],
[9, 2, "^NULL[0-9.]*$", 4, ",", 1],
[10, 2, "^[Ll]o[0-9.]*$", 4, ",", 1],
[11, 2, "^[Ss]ystem$", 4, ",", 1],
[12, 2, "^Nu[0-9.]*$", 4, ",", 1]
]
},
"settings": {
"fields": ["name", "type", "value_str", "value_int"],
"data": [
["autoreg_tls_accept", 2, "", 1],
["hk_history_global", 2, "", 0],
["snmptrap_logging", 2, "", 1],
["proxy_secrets_provider", 2, "", 0],
["hk_history", 1, "31d", 0],
["timeout_db_monitor", 1, "3s", 0],
["timeout_external_check", 1, "3s", 0],
["timeout_http_agent", 1, "3s", 0],
["timeout_simple_check", 1, "3s", 0],
["timeout_snmp_agent", 1, "3s", 0],
["timeout_ssh_agent", 1, "3s", 0],
["timeout_telnet_agent", 1, "3s", 0],
["timeout_zabbix_agent", 1, "3s", 0],
["timeout_browser", 1, "30s", 0]
]
},
"httptest": {
"fields": ["httptestid", "name", "delay", "agent", "authentication", "http_user", "http_password", "hostid
"data": []
},
"httptestitem": {
"fields": ["httptestitemid", "httptestid", "itemid", "type"],
"data": []
},
"httptest_field": {
"fields": ["httptest_fieldid", "httptestid", "type", "name", "value"],
"data": []
},
"httpstep": {
"fields": ["httpstepid", "httptestid", "name", "no", "url", "timeout", "posts", "required", "status_codes"
"data": []
},
"httpstepitem": {
"fields": ["httpstepitemid", "httpstepid", "itemid", "type"],
"data": []
},
```

```
"httpstep_field": {
"fields": ["httpstep_fieldid", "httpstepid", "type", "name", "value"],
"data": []
},
"config_autoreg_tls": {
"fields": ["autoreg_tlsid", "tls_psk_identity", "tls_psk"],
"data": [
[1, "", ""]
]
}
},
"macro.secrets": {
"AppID=zabbix_server&Query=Safe=passwordSafe;Object=zabbix": {
"Content": "738"
}
},
"config_revision": 2
}
```


Datu pieprasījums

`proxy data` pieprasījumu starpniekserveris nosūta, lai nodrošinātu host interfeisu pieejamības, vēstures, atklāšanas un automātiskās reģistrācijas datus. Šis pieprasījums tiek nosūtīts ik pēc `DataSenderFrequency` (starpniekservera konfigurācijas parametrs) sekundēm. Ņemiet vērā, ka aktīvais starpniekserveris joprojām katru sekundi aptaujās Zabbix serveri par attālināto komandu uzdevumiem (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)* Interfeisu pieejamības datu objektu masīvs.
interfaceid	*number*	Interfeisa identifikators.
available	*number*	Interfeisa pieejamība: **0**, *INTERFACE_AVAILABLE_UNKNOWN* - nav zināma **1**, *INTERFACE_AVAILABLE_TRUE* - pieejams **2**, *INTERFACE_AVAILABLE_FALSE* - nav pieejams
error	*string*	Interfeisa kļūdas ziņojums vai tukša virkne.
history data	*array*	*(optional)* Vēstures datu objektu masīvs.
itemid	*number*	Vienuma identifikators.
clock	*number*	Vienuma vērtības laikspiedols (sekundēs).
ns	*number*	Vienuma vērtības laikspiedols (nanosekundēs).
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 smaguma pakāpes vērtība.
eventid	*number*	*(optional)* Eventlog vienuma eventid vērtība.
state	*string*	*(optional)* Vienuma stāvoklis: **0**, *ITEM_STATE_NORMAL* **1**, *ITEM_STATE_NOTSUPPORTED*
lastlogsize	*number*	*(optional)* Žurnāla tipa vienumu pēdējais žurnāla izmērs.
mtime	*number*	*(optional)* Žurnāla tipa vienumu modificēšanas laiks.
discovery data	*array*	*(optional)* Atklāšanas datu objektu masīvs.
clock	*number*	Atklāšanas datu laikspiedols.
druleid	*number*	Atklāšanas noteikuma identifikators.

name	value type	description
dcheckid	*number*	Atklāšanas pārbaudes identifikators vai null atklāšanas noteikuma datiem.
type	*number*	Atklāšanas pārbaudes tips: **-1** atklāšanas noteikuma dati **0**, *SVC_SSH* - SSH pakalpojuma pārbaude **1**, *SVC_LDAP* - LDAP pakalpojuma pārbaude **2**, *SVC_SMTP* - SMTP pakalpojuma pārbaude **3**, *SVC_FTP* - FTP pakalpojuma pārbaude **4**, *SVC_HTTP* - HTTP pakalpojuma pārbaude **5**, *SVC_POP* - POP pakalpojuma pārbaude **6**, *SVC_NNTP* - NNTP pakalpojuma pārbaude **7**, *SVC_IMAP* - IMAP pakalpojuma 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 pakalpojuma pārbaude **15**, *SVC_TELNET* - Telnet pieejamības pārbaude
ip	*string*	Host IP adrese.
dns	*string*	Host DNS nosaukums.
port	*number*	*(optional)* Pakalpojuma porta numurs.
key_	*string*	*(optional)* Vienuma atslēga atklāšanas pārbaudei ar tipu **9** *SVC_AGENT*
value	*string*	*(optional)* No pakalpojuma saņemtā vērtība, vairumam pakalpojumu var būt tukša.
status	*number*	*(optional)* Pakalpojuma statuss: **0**, *DOBJECT_STATUS_UP* - pakalpojums darbojas **1**, *DOBJECT_STATUS_DOWN* - pakalpojums nedarbojas
autoregistration	*array*	*(optional)* Automātiskās reģistrācijas datu objektu masīvs.
clock	*number*	Automātiskās reģistrācijas datu laikspiedols.
host	*string*	Host nosaukums.
ip	*string*	*(optional)* Host IP adrese.
dns	*string*	*(optional)* No IP adreses atrisināts DNS nosaukums.
port	*string*	*(optional)* Host ports.
host_metadata	*string*	*(optional)* Host 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* - attālinātas 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 vēstures dati, ko nosūtīt
clock	*number*	*(optional)* Datu pārsūtīšanas laikspiedols (sekundēs).
ns	*number*	*(optional)* Datu pārsūtīšanas laikspiedols (nanosekundēs).
version	*string*	Starpniekservera versija (<major>.<minor>.<build>).
server→proxy:		
response	*string*	Pieprasījuma izpildes informācija (’success’ vai ’failed’).

name	value type	description
upload	*string*	Vēsturisko datu augšupielādes kontrole (vēsture, automātiskā reģistrācija, host pieejamība, tīkla atklāšana). Iespējamās vērtības: **enabled** - normāla darbība **disabled** - serveris nepieņem datus (iespējams, iekšējās kešatmiņas pārsnieguma dēļ)
tasks	*array*	*(optional)* Uzdevumu masīvs.
type	*number*	Uzdevuma tips: **1**, *ZBX_TM_TASK_PROCESS_REMOTE_COMMAND* - attālināta komanda
clock	*number*	Uzdevuma izveides laiks.
ttl	*number*	Laiks sekundēs, pēc kura uzdevums 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 globālo skriptu (šobrīd funkcionāli ekvivalents pielāgotam skriptam)
command	*string*	Izpildāmā attālinātā komanda.
execute_on	*number*	Izpildes mērķis pielāgotiem skriptiem: **0**, *ZBX_SCRIPT_EXECUTE_ON_AGENT* - izpildīt skriptu aģentā **1**, *ZBX_SCRIPT_EXECUTE_ON_SERVER* - izpildīt skriptu serverī **2**, *ZBX_SCRIPT_EXECUTE_ON_PROXY* - izpildīt skriptu starpniekserverī
port	*number*	*(optional)* Ports Telnet un SSH komandām.
authtype	*number*	*(optional)* SSH komandu autentifikācijas tips.
username	*string*	*(optional)* Lietotājvārds Telnet un SSH komandām.
password	*string*	*(optional)* Parole Telnet un SSH komandām.
publickey	*string*	*(optional)* SSH komandu publiskā atslēga.
privatekey	*string*	*(optional)* SSH komandu privātā atslēga.
parent_taskid	*number*	Vecākuzdevuma ID.
hostid	*number*	Mērķa host 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
```

```
}]
}
```

Konfigurācijas parametri: Zabbix starpniekserveris

Pārskats

Konfigurācijas parametri ļauj pielāgot Zabbix starpniekservera darbību.

Norādiet parametru vērtības Zabbix starpniekservera konfigurācijas failā (`zabbix_proxy.conf`). Visi parametri ir **neobligāti**, ja vien nav skaidri norādīts, ka parametrs ir obligāts.

Šī lapa ir strukturēta šādi:

- Parametru pārskats (lai skatītu pilnu informāciju par parametru, noklikšķiniet uz tā nosaukuma)
- Parametru detalizēts apraksts

Parametrs	Apraksts
AllowRoot	Atļaut starpniekserverim darboties kā `root`.
AllowUnsupportedDBVersions	Atļaut starpniekserverim darboties ar neatbalstītām datubāzu versijām.
APMListenIP	Norādīt IP adresi, kurā APM kolektoram klausīties ienākošos telemetrijas datus.
APMListenPort	Norādīt portu, kurā APM kolektoram klausīties ienākošos telemetrijas datus.
APMTLSCAFile	Norādīt pilnu ceļu uz failu, kurā atrodas augstākā līmeņa CA sertifikāts(-i), kas nepieciešams(-i), lai pārbaudītu to lietojumprogrammu vai OpenTelemetry kolektoru sertifikātus, kuri nosūta datus APM kolektoram.
APMTLSCertFile	Norādīt pilnu ceļu uz failu, kurā atrodas APM kolektora sertifikāts vai sertifikātu ķēde.
APMTLSKeyFile	Norādīt pilnu ceļu uz failu, kurā atrodas APM kolektora privātā atslēga.
CacheSize	Iestatīt konfigurācijas kešatmiņas lielumu.
ConfigFrequency	Šis parametrs ir **novecojis** (izmantojiet ProxyConfigFrequency). Iestatīt konfigurācijas datu izgūšanas biežumu no Zabbix servera.
DataSenderFrequency	Iestatīt savākto datu nosūtīšanas biežumu Zabbix serverim.
DBHost	Norādīt datubāzes hostu (vai ligzdas direktoriju).
DBName	Norādīt datubāzes nosaukumu vai ceļu uz SQLite3 datubāzes failu.
DBPassword	Norādīt datubāzes paroli.
DBPort	Norādīt Zabbix datubāzes hosta portu.
DBSchema	Norādīt datubāzes shēmas nosaukumu. Izmanto PostgreSQL.
DBSocket	Norādīt ceļu uz MySQL ligzdas failu.
DBUser	Norādīt datubāzes lietotāju.
DBTLSConnect	Norādīt vērtību, lai piespiestu TLS savienojumu ar datubāzi.
DBTLSCAFile	Norādīt pilnu ceļu uz failu, kurā atrodas augstākā līmeņa CA sertifikāts(-i) datubāzes sertifikāta pārbaudei.
DBTLSCertFile	Norādīt pilnu ceļu uz failu, kurā atrodas Zabbix starpniekservera sertifikāts autentifikācijai datubāzē.
DBTLSKeyFile	Norādīt pilnu ceļu uz failu, kurā atrodas privātā atslēga autentifikācijai datubāzē.
DBTLSCipher	Norādīt šifrēšanas algoritmu sarakstu, ko Zabbix starpniekserveris atļauj TLS protokoliem līdz TLS v1.2. Atbalstīts tikai MySQL.
DBTLSCipher13	Norādīt šifrēšanas algoritmu komplektu sarakstu, ko Zabbix starpniekserveris atļauj TLS v1.3 protokolam. Atbalstīts tikai MySQL, sākot ar versiju 8.0.16.
DebugLevel	Iestatīt atkļūdošanas līmeni.
EnableRemoteCommands	Atļaut attālās komandas no Zabbix servera.
ExternalScripts	Norādīt ārējo skriptu atrašanās vietu.
Fping6Location	Norādīt fping6 atrašanās vietu.
FpingLocation	Norādīt fping atrašanās vietu.
HistoryCacheSize	Iestatīt vēstures kešatmiņas lielumu.
HistoryIndexCacheSize	Iestatīt vēstures indeksu kešatmiņas lielumu.
Hostname	Norādīt starpniekservera nosaukumu.
HostnameItem	Norādīt vienumu, kas tiek izmantots Hostname iestatīšanai (ja tas nav definēts).
HousekeepingFrequency	Iestatīt uzkopšanas procedūras biežumu (stundās).
Include	Norādīt direktoriju vai atsevišķus failus, kas jāiekļauj konfigurācijas failā.
JavaGateway	Norādīt Zabbix Java gateway IP adresi (vai resursdatora nosaukumu).
JavaGatewayPort	Norādīt Zabbix Java gateway portu.
ListenBacklog	Iestatīt maksimālo gaidošo savienojumu skaitu TCP rindā.
ListenIP	Norādīt trappera klausīšanās IP adresi.

Parametrs	Apraksts
ListenPort	Norādīt trappera klausīšanās portu.
LoadModule	Norādīt moduli, kas jāielādē, startējot starpniekserveri.
LoadModulePath	Norādīt pilnu ceļu uz starpniekservera moduļu atrašanās vietu.
LogFile	Norādīt starpniekservera žurnālfailu.
LogFileSize	Iestatīt maksimālo žurnālfaila lielumu.
LogRemoteCommands	Iespējot izpildīto čaulas komandu reģistrēšanu kā brīdinājumus.
LogSlowQueries	Iestatīt maksimālo datubāzes vaicājuma izpildes ilgumu, pēc kura tas tiek reģistrēts (milisekundēs).
LogType	Norādīt žurnāla izvades tipu.
MaxConcurrentChecksPerPoller	Iestatīt maksimālo asinhrono pārbaužu skaitu, ko vienlaikus var izpildīt katrs HTTP aģenta polleris, aģenta polleris vai SNMP polleris.
PidFile	Norādīt PID failu.
ProxyBufferMode	Norādīt vēstures, atklāšanas un automātiskās reģistrācijas datu glabāšanas mehānismu (diskā/atmiņā/hibrīds).
ProxyConfigFrequency	Iestatīt konfigurācijas datu izgūšanas biežumu no Zabbix servera sekundēs.
ProxyLocalBuffer	Iestatīt datu lokālās glabāšanas ilgumu (stundās), pat ja dati jau ir sinhronizēti ar serveri.
ProxyMemoryBufferAge	Iestatīt maksimālo datu vecumu starpniekservera atmiņas buferī sekundēs.
ProxyMemoryBufferSize	Iestatīt koplietojamās atmiņas kešatmiņas lielumu savāktajiem vēstures, atklāšanas un automātiskās reģistrācijas datiem.
ProxyMode	Iestatīt starpniekservera darbības režīmu (aktīvs/pasīvs).
ProxyOfflineBuffer	Iestatīt datu glabāšanas ilgumu (stundās), ja nav savienojuma ar Zabbix serveri.
Server	Ja ProxyMode ir iestatīts uz `active`: norādīt Zabbix servera IP adresi vai DNS nosaukumu (address:port) vai klasteri (address:port;address2:port), no kura iegūt konfigurācijas datus un uz kuru nosūtīt datus. Ja ProxyMode ir iestatīts uz `passive`: norādīt Zabbix servera IP adreses vai DNS nosaukumus.
SNMPTrapperFile	Norādīt pagaidu failu, kas tiek izmantots datu pārsūtīšanai no SNMP trapu dēmona uz starpniekserveri.
SocketDir	Norādīt direktoriju iekšējo Zabbix pakalpojumu izmantoto IPC ligzdu glabāšanai.
SourceIP	Norādīt avota IP adresi.
SSHKeyLocation	Norādīt publisko un privāto atslēgu atrašanās vietu SSH pārbaudēm un darbībām.
SSLCertLocation	Norādīt SSL klienta sertifikātu failu atrašanās vietu klienta autentifikācijai.
SSLKeyLocation	Norādīt SSL privāto atslēgu failu atrašanās vietu klienta autentifikācijai.
SSLCALocation	Norādīt sertificēšanas iestādes (CA) failu atrašanās vietu SSL servera sertifikāta pārbaudei.
StartAgentPollers	Iestatīt iepriekš izveidoto asinhrono Zabbix aģenta polleru eksemplāru skaitu.
StartAPMCollectors	Iestatīt uz 1, lai palaistu APM kolektoru.
StartBrowserPollers	Iestatīt iepriekš izveidoto pārlūkprogrammas vienumu polleru eksemplāru skaitu.
StartDBSyncers	Iestatīt iepriekš izveidoto vēstures sinhronizētāju eksemplāru skaitu.
StartDiscoverers	Iestatīt iepriekš izveidoto atklāšanas procesu eksemplāru skaitu.
StartHTTPAgentPollers	Iestatīt iepriekš izveidoto asinhrono HTTP aģenta polleru eksemplāru skaitu.
StartHTTPPollers	Iestatīt iepriekš izveidoto HTTP polleru eksemplāru skaitu.
StartIPMIPollers	Iestatīt iepriekš izveidoto IPMI polleru eksemplāru skaitu.
StartJavaPollers	Iestatīt iepriekš izveidoto Java polleru eksemplāru skaitu.
StartODBCPollers	Iestatīt iepriekš izveidoto ODBC polleru eksemplāru skaitu.
StartPingers	Iestatīt iepriekš izveidoto ICMP pingeru eksemplāru skaitu.
StartPollersUnreachable	Iestatīt iepriekš izveidoto polleru eksemplāru skaitu nesasniedzamiem hostiem (tostarp IPMI un Java).
StartPollers	Iestatīt iepriekš izveidoto polleru eksemplāru skaitu.
StartPreprocessors	Iestatīt iepriekš palaisto priekšapstrādes procesu eksemplāru skaitu.
StartSNMPPollers	Iestatīt iepriekš izveidoto asinhrono SNMP polleru eksemplāru skaitu.
StartSNMPTrapper	Iestatīt uz ”1”, lai palaistu SNMP trappera procesu.
StartTelemetryQueryPollers	Iestatīt iepriekš izveidoto asinhrono telemetrijas vaicājumu polleru eksemplāru skaitu.
StartTrappers	Iestatīt iepriekš izveidoto trapperu eksemplāru skaitu.
StartVMwareCollectors	Iestatīt iepriekš izveidoto VMware kolektoru eksemplāru skaitu.
StatsAllowedIP	Norādīt ārējo Zabbix instanču IP adreses vai DNS nosaukumus, kurām atļauts pieprasīt statistiku.
TelemetryProvider	Iestatīt telemetrijas datu avotu.
Timeout	Norādīt, cik ilgi (sekundēs) gaidīt savienojuma izveidi un datu apmaiņu ar Zabbix starpniekserveri, aģentu, tīmekļa pakalpojumu un SNMP pārbaužu veikšanu (izņemot SNMP `walk[OID]` un `get[OID]` vienumus).
TLSAccept	Norādīt ienākošajiem savienojumiem no Zabbix servera nepieciešamo šifrēšanas līmeni.
TLSCAFile	Norādīt pilnu ceļu uz failu, kurā atrodas augstākā līmeņa CA sertifikāts(-i) partnera sertifikāta pārbaudei, ko izmanto šifrētai saziņai starp Zabbix komponentiem.

Parametrs	Apraksts
TLSCertFile	Norādīt pilnu ceļu uz failu, kurā atrodas servera sertifikāts vai sertifikātu ķēde, ko izmanto šifrētai saziņai starp Zabbix komponentiem.
TLSCipherAll	Norādīt GnuTLS prioritātes virkni vai OpenSSL (TLS 1.2) šifrēšanas virkni. Ignorēt noklusējuma šifrēšanas algoritmu komplekta atlases kritērijus sertifikātu un PSK šifrēšanai.
TLSCipherAll13	Norādīt OpenSSL 1.1.1 vai jaunākas versijas šifrēšanas virkni TLS 1.3. Ignorēt noklusējuma šifrēšanas algoritmu komplekta atlases kritērijus sertifikātu un PSK šifrēšanai.
TLSCipherCert	Norādīt GnuTLS prioritātes virkni vai OpenSSL (TLS 1.2) šifrēšanas virkni. Ignorēt noklusējuma šifrēšanas algoritmu komplekta atlases kritērijus sertifikātu šifrēšanai.
TLSCipherCert13	Norādīt OpenSSL 1.1.1 vai jaunākas versijas šifrēšanas virkni TLS 1.3. Ignorēt noklusējuma šifrēšanas algoritmu komplekta atlases kritērijus sertifikātu šifrēšanai.
TLSCipherPSK	Norādīt GnuTLS prioritātes virkni vai OpenSSL (TLS 1.2) šifrēšanas virkni. Ignorēt noklusējuma šifrēšanas algoritmu komplekta atlases kritērijus PSK šifrēšanai.
TLSCipherPSK13	Norādīt OpenSSL 1.1.1 vai jaunākas versijas šifrēšanas virkni TLS 1.3. Ignorēt noklusējuma šifrēšanas algoritmu komplekta atlases kritērijus PSK šifrēšanai.
TLSConnect	Norādīt savienojumiem ar Zabbix serveri nepieciešamo šifrēšanas līmeni.
TLSCRLFile	Norādīt pilnu ceļu uz failu, kurā atrodas atsauktie sertifikāti. Šo parametru izmanto šifrētai saziņai starp Zabbix komponentiem.
TLSKeyFile	Norādīt pilnu ceļu uz failu, kurā atrodas starpniekservera privātā atslēga, ko izmanto šifrētai saziņai starp Zabbix komponentiem.
TLSListen	Pieprasīt tikai TLS savienojumus trappera ligzdā.
TLSPSKFile	Norādīt pilnu ceļu uz failu, kurā atrodas starpniekservera iepriekš koplietotā atslēga, ko izmanto šifrētai saziņai ar Zabbix serveri.
TLSPSKIdentity	Norādīt iepriekš koplietotās atslēgas identitātes virkni, ko izmanto šifrētai saziņai ar Zabbix serveri.
TLSServerCertIssuer	Norādīt atļauto servera sertifikāta izdevēju.
TLSServerCertSubject	Norādīt atļauto servera sertifikāta subjektu.
TmpDir	Norādīt pagaidu direktoriju.
TrapperTimeout	Norādīt taimautu sekundēs: - konfigurācijas datu izgūšanai no Zabbix servera; - globālā skripta izpildei vai attālās komandas izpildei.
UnavailableDelay	Iestatīt hosta pieejamības pārbaudes biežumu nepieejamības periodā.
UnreachableDelay	Iestatīt hosta pieejamības pārbaudes biežumu nesasniedzamības periodā.
UnreachablePeriod	Iestatīt, pēc cik sekundēm hostu uzskatīt par nepieejamu, ja tas nav sasniedzams.
User	Atteikties no privilēģijām un izmantot konkrētu, sistēmā esošu lietotāju.
Vault	Norādīt vault nodrošinātāju.
VaultAppRoleID	Norādīt HashiCorp vault AppRole ID.
VaultAppSecretID	Norādīt HashiCorp vault AppRole slepeno ID.
VaultDBPath	Norādīt atrašanās vietu, no kuras pēc atslēgām izgūt datubāzes akreditācijas datus.
VaultPrefix	Norādīt pielāgotu prefiksu vault ceļam vai vaicājumam.
VaultTLSCertFile	Norādīt klienta autentifikācijai izmantotā SSL sertifikāta faila nosaukumu.
VaultTLSKeyFile	Norādīt klienta autentifikācijai izmantotā SSL privātās atslēgas faila nosaukumu.
VaultToken	Norādīt HashiCorp vault autentifikācijas pilnvaru.
VaultURL	Norādīt vault servera URL.
VMwareCacheSize	Iestatīt koplietojamās atmiņas lielumu VMware datu glabāšanai.
VMwareFrequency	Iestatīt datu apkopošanas biežumu no viena VMware pakalpojuma sekundēs.
VMwarePerfFrequency	Iestatīt veiktspējas skaitītāju statistikas izgūšanas biežumu no viena VMware pakalpojuma sekundēs.
VMwareTimeout	Iestatīt maksimālo sekunžu skaitu, ko `vmware collector` gaidīs atbildi no VMware pakalpojuma.
WebDriverURL	Norādīt WebDriver saskarnes URL.

Ņemiet vērā:

- Noklusējuma vērtības atspoguļo dēmona noklusējuma vērtības, nevis vērtības piegādātajos konfigurācijas failos.
- Vērtības atbalsta vides mainīgos.
- Zabbix atbalsta konfigurācijas failus tikai UTF-8 kodējumā bez BOM.
- Komentāri, kas sākas ar #, tiek atbalstīti tikai rindas sākumā.

Parametra informācija

AllowRoot

Atļaut starpniekserverim darboties kā `root`. Ja šī opcija ir atspējota un starpniekserveris tiek palaists ar `root`, starpniekserveris mēģinās pārslēgties uz `zabbix` lietotāju. Nav ietekmes, ja tas tiek palaists parasta lietotāja ietvaros.

Noklusējums: 0
 Vērtības: 0 - neatļaut; 1 - atļaut

Atļaut neatbalstītas DB versijas

Atļaut starpniekserverim darboties ar neatbalstītām datubāzes versijām.

Noklusējums: 0
 Vērtības: 0 - neatļaut; 1 - atļaut

APMListenIP

Norādiet IP adresi, kurā APM collector jāgaida ienākošie telemetrijas dati.
 Ja nav norādīts, APM collector gaidīs datus visās tīkla saskarnēs.

Noklusējums: `0.0.0.0`

APMListenPort

Norādiet portu, kurā APM collector jāgaida ienākošie telemetrijas dati.
 Noklusējuma ports ir standarta OpenTelemetry (OTLP/gRPC) ports, ko pēc noklusējuma izmanto OpenTelemetry SDK un kolektori.

Noklusējums: 4317
 Diapazons: 1024-32767

APMTLSCAFile

Norādiet pilnu ceļu uz failu, kurā ir augstākā līmeņa CA sertifikāti to lietojumprogrammu vai OpenTelemetry kolektoru sertifikātu pārbaudei, kuri sūta datus uz APM kolektoru.
 Ja šis parametrs ir iestatīts, APM kolektors pieņem tikai šifrētus savienojumus; pretējā gadījumā tas pieņem tikai nešifrētus savienojumus.

APMTLSCertFile

Norādiet pilnu ceļu uz failu, kurā atrodas APM collector sertifikāts vai sertifikātu ķēde.
 Šis parametrs jāiestata kopā ar `APMTLSKeyFile` un tikai tad, ja ir iestatīts `APMTLSCAFile`.

APMTLSKeyFile

Norādiet pilnu ceļu uz failu, kurā atrodas APM kolektora privātā atslēga.
 Šis parametrs ir jāiestata kopā ar `APMTLSCertFile` un tikai tad, ja ir iestatīts `APMTLSCAFile`.

CacheSize

Iestatiet konfigurācijas kešatmiņas lielumu baitos. Koplietotās atmiņas apjoms, kas tiek izmantots hosts un vienumu datu glabāšanai.

Noklusējums: 32M
 Diapazons: 128K-64G

ConfigFrequency

Šis parametrs ir **novecojis** (tā vietā izmantojiet `ProxyConfigFrequency`).
Iestata konfigurācijas datu izgūšanas biežumu no Zabbix serveris (sekundēs).
Aktīvs starpniekserveris parametrs. Tiek ignorēts pasīvajiem starpniekserveriem (skatiet `ProxyMode` parametru).

Noklusējums: 3600
 Diapazons: 1-604800

DataSenderFrequency

Iestata savākto datu nosūtīšanas biežumu no starpniekservera uz Zabbix serveri. Ņemiet vērā, ka aktīvs starpniekserveris joprojām katru sekundi aptaujās Zabbix serveri attālo komandu uzdevumiem.
Aktīva starpniekservera parametrs. Tiek ignorēts pasīvajiem starpniekserveriem (skatiet `ProxyMode` parametru).

Noklusējums: 1
 Diapazons: 1-3600

DBHost

Norādiet datubāzes hostu (vai socket direktoriju).

MySQL gadījumā:

- `localhost` vai tukša virkne izmanto noklusējuma UNIX domēna socket.

PostgreSQL gadījumā:

- `localhost` tiek atrisināts, izmantojot DNS (parasti uz `127.0.0.1`).
- Tukša virkne izmanto noklusējuma UNIX domēna socket.
- Ceļš (piemēram, `/var/run/pgbouncer`) nozīmē UNIX socket izmantošanu šajā ceļā.

- Ar komatiem atdalīts saraksts var saturēt vairākas `host:port` vērtības. Katrs host tiek mēģināts pēc kārtas, līdz tiek izveidots lasīšanas-rakstīšanas savienojums. Piemērs: `DBHost=localhost:5431,127.0.0.1:20051,zabbix.domain,[::1]:30051`

Noklusējums: `localhost`

DBName

Norādiet datubāzes nosaukumu vai ceļu uz SQLite3 datubāzes failu (Zabbix daudzprocesu arhitektūra neļauj izmantot atmiņas datubāzi, 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

Norādiet datubāzes paroli. Ja parole netiek izmantota, komentējiet šo rindu. SQLite gadījumā tiek ignorēts.

DBPort

Norādiet Zabbix datubāzes hosta portu (skatiet `DBHost`). SQLite gadījumā tiek ignorēts.
 DBSocket un DBPort starpniekservera konfigurācijā ir savstarpēji izslēdzoši. Norādiet tikai vienu vai atstājiet abus nedefinētus.

Noklusējums MySQL: 3306

Noklusējums PostgreSQL: 5432

Diapazons: 1024-65535

DBSchema

Norādiet datubāzes shēmas nosaukumu. Izmanto PostgreSQL.

DBSocket

Norādiet ceļu uz MySQL socket failu.
DBSocket un DBPort starpniekservera konfigurācijā ir savstarpēji izslēdzoši. Norādiet tikai vienu vai atstājiet abus nedefinētus.

DBUser

Norādiet datubāzes lietotāju. SQLite gadījumā tiek ignorēts.

DBTLSConnect

Norādiet vērtību, lai piespiestu 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, ka datubāzes identitāte, kas norādīta ar DBHost, atbilst tās sertifikātam
MySQL, sākot no 5.7.11, un PostgreSQL tiek atbalstītas šādas vērtības: "required", "verify", "verify_full".
MariaDB, sākot no versijas 10.2.6, tiek atbalstītas vērtības "required" un "verify_full".
Pēc noklusējuma neviena opcija nav iestatīta, un darbība ir atkarīga no datubāzes konfigurācijas.

DBTLSCAFile

Norādiet pilnu ceļu uz failu, kurā atrodas augstākā līmeņa CA sertifikāti datubāzes sertifikātu verifikācijai.

Obligāts: nē (jā, ja DBTLSConnect ir iestatīts uz *verify_ca* vai *verify_full*)

DBTLSCertFile

Norādiet pilnu ceļu uz failu, kurā ir Zabbix starpniekservera sertifikāts autentifikācijai datubāzē.

DBTLSKeyFile

Norādiet pilnu ceļu uz failu, kurā atrodas privātā atslēga autentifikācijai datubāzē.

DBTLSCipher

Norādiet šifrēšanas algoritmu sarakstu, ko Zabbix starpniekserveris atļauj TLS protokoliem līdz TLS v1.2 ieskaitot. Atbalstīts tikai MySQL.

DBTLSCipher13

Norādiet šifrēšanas ciphersuite sarakstu, ko Zabbix starpniekserveris atļauj TLS v1.3 protokolam. Atbalstīts tikai MySQL, sākot no versijas 8.0.16.

DebugLevel

Iestatiet 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

EnableRemoteCommands

Atļaut attālinātās komandas no Zabbix serveris.

Noklusējums: 0
 Vērtības: 0 - neatļaut; 1 - atļaut

Ārējie skripti

Norādiet ārējā skripta atrašanās vietu (atkarīgs no `datadir` kompilēšanas laikā iestatītās instalācijas mainīgās).

Noklusējums: `/usr/local/share/zabbix/externalscripts`

Fping6Location

Norādiet fping6 atrašanās vietu. Pārliecinieties, ka fping6 binārajam failam ir root īpašumtiesības un ir iestatīts SUID karodziņš. Atstājiet tukšu (`Fping6Location=`), ja jūsu `fping` utilīta spēj apstrādāt IPv6 adreses.

Noklusējums: `/usr/sbin/fping6`

FpingLocation

Norādiet fping atrašanās vietu. Pārliecinieties, ka fping binārajam failam ir root īpašumtiesības un iestatīts SUID karodziņš.

Noklusējums: `/usr/sbin/fping`

HistoryCacheSize

Iestatiet vēstures kešatmiņas lielumu baitos. Koplietotās atmiņas apjoms vēstures datu glabāšanai.

Noklusējums: `16M`
 Diapazons: 128K-16G

HistoryIndexCacheSize

Iestatiet vēstures indeksa kešatmiņas lielumu baitos. Koplietotās atmiņas apjoms vēstures datu indeksēšanai, kas glabājas vēstures kešatmiņā. Indeksa kešatmiņas lielumam viena vienuma kešošanai nepieciešami aptuveni 100 baiti.

Noklusējums: `4M`
 Diapazons: 128K-16G

Hostname

Norādiet unikālu, reģistrjutīgu starpniekservera nosaukumu. Pārliecinieties, ka serveris zina starpniekservera nosaukumu.
Atļautās rakstzīmes: burti un cipari, '.', ' ', '_' un '-'. Maksimālais garums: 128

Noklusējums: iestatīts ar HostnameItem

HostnameItem

Norādiet vienumu `Hostname` iestatīšanai, ja tas nav definēts.
 Šis vienums tiks izpildīts uz starpniekservera tāpat kā uz aģenta.
 Tiek ignorēts, ja `Hostname` ir iestatīts.
Neatbalsta lietotāja parametrus, veiktspējas skaitītājus vai aizstājvārdus; atbalsta `system.run[]`.

Noklusējums: `system.hostname`

HousekeepingFrequency

Iestatiet apkopes procedūras biežumu (stundās). Apkope ir novecojušas informācijas noņemšana no datubāzes.
Piezīme: lai samazinātu slodzi starpniekservera startēšanas laikā, apkope tiek atlikta uz 30 minūtēm pēc starpniekservera palaišanas. Tādējādi, ja `HousekeepingFrequency` ir 1, pati pirmā apkopes procedūra pēc starpniekservera startēšanas tiks palaista pēc 30 minūtēm un pēc tam atkārtosies katru stundu.
Automātisko apkopi ir iespējams atspējot, iestatot HousekeepingFrequency uz 0. Šādā gadījumā apkopes procedūru var palaist tikai ar `housekeeper_execute` izpildlaika vadības opciju.
Skatiet arī izpildlaika vadības opcijas un informāciju par apkopes procedūru.

Noklusējums: `1`
 Diapazons: 0-24

Iekļaut

Norādiet direktoriju vai atsevišķus failus, kas jāiekļauj konfigurācijas failā.
 Zabbix instalēšanas laikā šis parametrs tiek iestatīts uz `/usr/local/etc`, ja vien to nemaināt kompilēšanas laikā.

Vispārīgie noteikumi parametram `Include`:

- Ceļš var būt relatīvs attiecībā pret `zabbix_proxy.conf` faila atrašanās vietu.
- Vienā konfigurācijas failā var norādīt vairākus `Include` parametrus.
- Zabbix apstrādā `Include` parametru uzreiz pēc tā nolasīšanas, tāpēc ieteicams to izvietot konfigurācijas faila beigās.

Iekļauto failu noteikumi:

- Katram failam jābūt lasāmam.
- Katram failam jābūt ar pareizu sintaksi. Pretējā gadījumā komponents netiks palaists.

Direktoriju vai failu paraugu iekļaušanas noteikumi:

- Nepieņemiet, ka iekļaušana notiek noteiktā secībā (piemēram, faili netiek iekļauti alfabētiskā secībā).
- Nenorādiet vienu un to pašu parametru vairāk nekā vienā iekļautajā failā (piemēram, lai pārrakstītu vispārīgu iestatījumu ar konkrētu), jo nevar paredzēt, kura vērtība tiks izmantota.
- Uzmanieties no rezerves kopijām, ko daži teksta redaktori izveido automātiski (piemēram, rediģējot `include/my_specific.conf`, var tikt izveidots `include/my_specific.conf.BAK`), jo tiks iekļauti abi faili.

Lai no direktorija iekļautu tikai atbilstošos failus, izmantojiet aizstājējzīmi (`*`) paraugu atbilstības noteikšanai.

Piemērs:

```
Include=/absolute/path/to/config/files/*.conf
```

JavaGateway

Norādiet Zabbix Java gateway IP adresi vai resursdatora nosaukumu. Nepieciešams tikai tad, ja ir palaisti Java aptaujātāji.

JavaGatewayPort

Norādiet Zabbix Java gateway portu.

Noklusējums: 10052
 Diapazons: 1024-32767

ListenBacklog

Iestata maksimālo TCP rindā gaidošo savienojumu skaitu.
Noklusējuma vērtība ir cieti kodēta konstante, kas ir atkarīga no sistēmas.
Atbalstītā maksimālā vērtība arī ir atkarīga no sistēmas; pārāk lielas vērtības var tikt klusi saīsinātas līdz ”implementation-specified maximum”.

Noklusējums: SOMAXCONN
 Diapazons: 0 - INT_MAX

ListenIP

Norādiet trapper klausīšanās IP adresi. Komatatdalīts saraksts.
Ja šis parametrs nav norādīts, trapper klausīsies visās tīkla saskarnēs.

Noklusējums: `0.0.0.0`

ListenPort

Norādiet trapper klausīšanās portu.

Noklusējums: 10051
 Diapazons: 1024-32767

LoadModule

Norādiet moduli, kas jāielādē, startējot starpniekserveri. 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 `/`), `LoadModulePath` tiek ignorēts.
 Formāti: - `LoadModule=<module.so>` - `LoadModule=<path/module.so>` - `LoadModule=</abs_path/module.so>`

Var norādīt vairākus `LoadModule` parametrus.

LoadModulePath

Pilns ceļš uz starpniekservera moduļu atrašanās vietu. Noklusējums ir atkarīgs no kompilēšanas opcijām.

LogFile

Norādiet starpniekservera žurnāla failu.

Obligāts: Jā, ja `LogType` ir iestatīts uz `file`; citādi nē

LogFileSize

Iestatiet maksimālo žurnāla faila lielumu (MB).
0 - atspējot automātisku žurnāla rotāciju.
Piezīme: Ja tiek sasniegts žurnāla faila lieluma ierobežojums un faila rotācija neizdodas jebkāda iemesla dēļ, esošais žurnāla fails tiek saīsināts un izveidots no jauna.

Noklusējums: 1
 Diapazons: 0-1024

LogRemoteCommands

Iespējot izpildīto čaulas komandu reģistrēšanu kā brīdinājumus.

Noklusējums: 0
 Vērtības: 0 - atspējots, 1 - iespējots

LogSlowQueries

Iestatiet maksimālo datubāzes vaicājuma izpildes ilgumu, pēc kura tas tiek reģistrēts žurnālā (milisekundēs).
0 - nereģistrēt lēnos vaicājumus.
Šī opcija kļūst pieejama, sākot ar DebugLevel=3.

Noklusējums: 0
 Diapazons: 0-3600000

LogType

Norādiet žurnāla izvades tipu: - *file* - rakstīt žurnālu failā, kas norādīts ar LogFile parametru; - *system* - rakstīt žurnālu syslog; - *console* - rakstīt žurnālu standarta izvadei.

Noklusējums: file

MaxConcurrentChecksPerPoller

Iestatiet maksimālo asinhrono pārbaužu skaitu, ko vienlaikus var izpildīt katrs HTTP aģenta aptaujātājs, aģenta aptaujātājs vai SNMP aptaujātājs. Skatiet StartHTTPAgentPollers, StartAgentPollers un StartSNMPPollers.

Noklusējums: 1000
 Diapazons: 1-1000

PidFile

Norādiet PID failu.

Noklusējums: /tmp/zabbix_proxy.pid

ProxyBufferMode

Norādiet vēstures, 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 buferim beidzas atmiņa, vecie dati tiks atmesti. Izslēgšanas laikā buferis tiek atmests. - *hybrid* - starpniekservera buferis parasti darbojas kā atmiņas režīmā, līdz tam beidzas atmiņa vai vecākais ieraksts pārsniedz konfigurēto vecumu. Ja tas notiek, buferis tiek iztukšots datubāzē un tas darbojas kā diska režīmā, līdz visi dati ir augšupielādēti, un pēc tam tas atkal sāk darboties ar atmiņu. Izslēgšanas laikā atmiņas buferis tiek iztukšots datubāzē.

Skatiet arī: Starpniekservera atmiņas buferis.

Noklusējums: disk
 Vērtības: disk; memory; hybrid

ProxyConfigFrequency

Iestata konfigurācijas datu izgūšanas biežumu no Zabbix serveris sekundēs.
Aktīvs starpniekserveris parametrs. Tiek ignorēts pasīviem starpniekserveriem (skatiet ProxyMode parametru).

Noklusējums: 10
 Diapazons: 1-604800

ProxyLocalBuffer

Iestatiet datu glabāšanas ilgumu lokāli (stundās), pat ja dati jau ir sinhronizēti ar serveri..
Šo parametru var izmantot, ja lokālos datus izmantos trešo pušu lietojumprogrammas.

Noklusējums: 0
 Diapazons: 0-720

ProxyMemoryBufferAge

Iestata maksimālo datu vecumu starpniekservera atmiņas buferī sekundēs.
 Ja šis iestatījums ir iespējots (nav nulle) un ieraksti starpniekservera atmiņas buferī ir vecāki, tas piespiež starpniekservera buferi pārslēgties uz datubāzes režīmu, līdz visi ieraksti ir augšupielādēti serveris. Šim parametram jābūt mazākam vai vienādam ar ProxyOfflineBuffer parametru.

Noklusējums: 0
 Diapazons: 0;600-864000

ProxyMemoryBufferSize

Iestatiet koplietotās atmiņas kešatmiņas lielumu savāktajiem vēstures, atklāšanas un automātiskās reģistrācijas datiem (baitos). Ja ir iespējots (nav nulle), starpniekserveris glabās vēstures, atklāšanas un automātiskās reģistrācijas datus atmiņā, līdz kešatmiņa ir pilna vai saglabātie ieraksti ir vecāki par definēto ProxyMemoryBufferAge. Šo parametru nevar izmantot kopā ar parametru ProxyLocalBuffer.

Noklusējums: 0
 Diapazons: 0;128K-2G

ProxyMode

Iestatiet starpniekservera darbības režīmu.
0 - starpniekserveris aktīvajā režīmā
1 - starpniekserveris pasīvajā režīmā
Piezīme ka (sensitīvi) starpniekservera konfigurācijas dati var kļūt pieejami pusēm, kurām ir piekļuve Zabbix servera trapper porta, ja tiek izmantots aktīvais starpniekserveris. Tas ir iespējams, jo ikviens var uzdoties par aktīvo starpniekserveri un pieprasīt konfigurācijas datus; autentifikācija netiek veikta.

Noklusējums: 0
 Diapazons: 0-1

ProxyOfflineBuffer

Iestatiet datu saglabāšanas ilgumu (stundās) gadījumā, ja nav savienojuma ar Zabbix serveri.
Vecāki dati tiks zaudēti.

Noklusējums: 1
 Diapazons: 1-720

Serveris

Ja `ProxyMode` ir iestatīts uz `active`: norādiet Zabbix servera IP adresi vai DNS nosaukumu (adrese:ports) vai klasteri (adrese:ports;adrese2:ports), no kura saņemt konfigurācijas datus un uz kuru sūtīt datus.
Ja ports nav norādīts, tiek izmantots noklusējuma ports.
Klastera mezgliem jābūt atdalītiem ar semikolu.

Ja `ProxyMode` ir iestatīts uz `passive`: norādiet Zabbix servera IP adreses, pēc izvēles CIDR notācijā, vai DNS nosaukumus. Vērtības jānorāda ar komatiem atdalītā sarakstā.
 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 uzskatīti par vienādiem.
`::/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

Norādiet pagaidu failu, 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

Norādiet direktoriju IPC ligzdu glabāšanai, ko izmanto iekšējie Zabbix pakalpojumi.

Noklusējums: `/tmp`

SourceIP

Norādiet avota IP adresi šādiem savienojumiem:

- izejošie savienojumi ar Zabbix serveri
- aģentless savienojumi (VMware, SSH, JMX, SNMP, Telnet un vienkāršās pārbaudes)
- HTTP aģenta savienojumi
- skripta vienuma JavaScript HTTP pieprasījumi
- pirmapstrādes JavaScript HTTP pieprasījumi
- savienojumi ar Vault

SSHKeyLocation

Norādiet publisko un privāto atslēgu atrašanās vietu SSH pārbaudēm un darbībām.

SSLCertLocation

Norādiet SSL klienta sertifikātu failu atrašanās vietu klienta autentifikācijai.
Šis parametrs tiek izmantots tīmekļa uzraudzībā, saziņai ar Vault un telemetrijas datu avotam ClickHouse.

SSLKeyLocation

Norādiet SSL privāto atslēgu failu atrašanās vietu klienta autentifikācijai.
Šis parametrs tiek izmantots tīmekļa uzraudzībā, saziņai ar Vault un telemetrijas datu avotam ClickHouse.

SSLCALocation

Norādiet sertificēšanas iestādes (CA) failu atrašanās vietu SSL servera sertifikāta pārbaudei.
Ņemiet vērā, ka šī parametra vērtība tiks iestatīta kā `CURLOPT_CAPATH` libcurl opcija. libcurl versijām, kas vecākas par 7.42.0, tas darbojas tikai tad, ja libcurl tika kompilēts lietošanai ar OpenSSL. Papildinformāciju skatiet cURL tīmekļa lapā.
Šis parametrs tiek izmantots tīmekļa uzraudzībā, SMTP autentifikācijā, HTTP aģenta vienumos, saziņai ar Vault un telemetrijas datu avotam ClickHouse.

StartAgentPollers

Iestata iepriekš atzaroto Zabbix aģents `pollers` instanču skaitu. Skatiet `MaxConcurrentChecksPerPoller`.

Noklusējums: 1
 Diapazons: 0-1000

StartAPMCollectors

Iestatiet uz 1, lai palaistu iepriekš izveidotu APM kolektora instanci (procesus `apm manager` un `apm worker`).

Noklusējuma vērtība: 0
 Diapazons: 0-1

StartBrowserPollers

Iestata iepriekš sazaroto browser vienuma `pollers` instanču skaitu.

Noklusējums: 1
 Diapazons: 0-1000

StartDBSyncers

Iestata iepriekš sazaroto `history syncers` instanču skaitu.
Piezīme: Esiet uzmanīgi, mainot šo vērtību, jo tās palielināšana var nodarīt vairāk ļauna nekā laba.

Noklusējums: 4
 Diapazons: 1-100

StartDiscoverers

Iestatiet iepriekš sazaroto `discovery workers` instanču skaitu.

Noklusējums: 5
 Diapazons: 0-1000

StartHTTPAgentPollers

Iestata iepriekš sazaroto `HTTP agent pollers` instanču skaitu. Skatiet `MaxConcurrentChecksPerPoller`.

Noklusējums: 1
 Diapazons: 0-1000

StartHTTPPollers

Iestata iepriekš sazaroto `HTTP pollers` instanču skaitu.

Noklusējums: 1
 Diapazons: 0-1000

StartIPMIPollers

Iestata iepriekš sazaroto `IPMI pollers` instanču skaitu.

Noklusējums: 0
 Diapazons: 0-1000

StartJavaPollers

Iestata iepriekš sazaroto `Java pollers` instanču skaitu.

Noklusējums: 0
 Diapazons: 0-1000

StartODBCPollers

Iestata iepriekš sazaroto `ODBC pollers` instanču skaitu.

Noklusējums: 1
 Diapazons: 0-1000

StartPingers

Iestata iepriekš sazaroto `ICMP pingers` instanču skaitu.

Noklusējums: 1
 Diapazons: 0-1000

StartPollersUnreachable

Iestata iepriekš atzaroto nepieejamu hostu aptaujātāju instanču skaitu (ieskaitot IPMI un Java). Ja ir palaisti parastie, IPMI vai Java aptaujātāji, jādarbojas vismaz vienam nepieejamu hostu aptaujātājam.

Noklusējums: 1
 Diapazons: 0-1000

StartPollers

Iestata iepriekš sazaroto `pollers` instanču skaitu.

Noklusējums: 5
 Diapazons: 0-1000

StartPreprocessors

Iestatiet iepriekš palaisto priekšapstrādes `worker` pavedienu instanču skaitu ne mazāku par pieejamo CPU kodolu skaitu. Ja priekšapstrāde nav ierobežota ar CPU jaudu un tajā ir daudz tīkla pieprasījumu, jāiestata vairāk worker.

Noklusējums: 16
 Diapazons: 1-1000

StartSNMPPollers

Iestata iepriekš atzaroto `SNMP pollers` instanču skaitu. Skatiet `MaxConcurrentChecksPerPoller`.

Noklusējums: 1
 Diapazons: 0-1000

StartTelemetryQueryPollers

Iestatiet asinhrono Telemetry vaicājumu aptaujātāju iepriekš izveidoto eksemplāru skaitu. Skatiet MaxConcurrentChecksPerPoller.

Noklusējums: 1
 Diapazons: 0-1000

StartSNMPTrapper

Iestatiet uz "1", lai palaistu `SNMP trapper` procesu.

Noklusējums: 0
 Diapazons: 0-1

StartTrappers

Iestata iepriekš atzaroto `trappers` instanču skaitu.
Trappers pieņem ienākošos savienojumus no Zabbix sender un aktīvajiem aģentiem.

Noklusējums: 5
 Diapazons: 0-1000

StartVMwareCollectors

Iestatiet iepriekš sazaroto `vmware collector` instanču skaitu.

Noklusējums: 0
 Diapazons: 0-250

StatsAllowedIP

Norādiet IP adreses (pēc izvēles CIDR notācijā) vai ārējo Zabbix instanču DNS nosaukumus, kurām ir atļauts pieprasīt statistiku. Ar komatiem atdalīts saraksts.
 Ja šis parametrs nav iestatīts, statistikas pieprasījumi netiks pieņemti.
Ja ir iespējots IPv6 atbalsts, tad `127.0.0.1`, `::127.0.0.1`, `::ffff:127.0.0.1` tiek uzskatīti par ekvivalentiem, 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
```

TelemetryProvider

Iestatiet telemetrijas datu avotu formātā `<provider>[;<options>]`, kur:

- `<provider>` - viens no atbalstītajiem telemetrijas datu avotiem: `clickhouse`
- `<options>` - ar komatiem atdalīti `key=value` pāri. Vērtības var būt ietvertas pēdiņās.

ClickHouse opcijas:

- `url` - iestatiet HTTP vai HTTPS URL, kas tiek izmantots savienojuma izveidei ar telemetrijas datu avotu.
- `db` - iestatiet datubāzes nosaukumu.
- `username` - iestatiet lietotājvārdu, kas tiek izmantots, lai pieteiktos datubāzē.
- `password` - iestatiet paroli, kas tiek izmantota, lai pieteiktos datubāzē.
- `ssl_cert_file`, `ssl_key_file`, `ssl_key_password`, `ssl_verify_peer`, `ssl_verify_host`, `ssl_ca_location`, `ssl_cert_location`, `ssl_key_location` - iestatiet šīs opcijas, lai konfigurētu drošu (TLS/SSL) savienojumu ar datubāzi.
- `source_ip` - iestatiet avota IP adresi, kas tiek izmantota izejošajiem savienojumiem ar telemetrijas datu avotu.
- `vault_path` - iestatiet ceļu uz Vault noslēpumu, kurā glabājas `username` un `password` vērtības. Neizmantojiet kopā ar `username` un `password` opcijām.

ClickHouse piemērs:

```
TelemetryProvider=clickhouse;url=http://localhost:8123,db=zabbix,username=zabbix,password=zabbix
```

ClickHouse piemērs ar Vault:

```
TelemetryProvider=clickhouse;url=http://localhost:8123,db=zabbix,vault_path=clickhouse
```

Noildze

Norādiet, cik ilgi jāgaida (sekundēs), lai izveidotu savienojumu un apmainītos ar datiem ar Zabbix serveri, aģentu, tīmekļa pakalpojumu un mantotajām SNMP pārbaudēm (viens OID numurs vai virkne).

Šis parametrs nosaka dažādu saziņas darbību ilgumu:

- Attālinātas komandas izpilde Zabbix aģentā
- SSH / Telnet komandas izpilde
- Vienumu pārplānošana, kad IPMI saskarne kļūst nepieejama
- Atbildes nosūtīšana Zabbix serverim, ja datu apmaiņa neizdodas tiesību vai šifrēšanas problēmu dēļ
- IPC asinhrono ligzdu un izpildlaika vadības opciju termiņš
- Asinhrono aptaujātāju DNS pieprasījumi

- Atbilde aktīvās pārbaudes sirdspukstam
- Zabbix aģenta datu (vērtību) izgūšana no aktīvajiem aģentiem
- Datu izgūšana no Zabbix sender
- Aktīvās pārbaudes saraksta nosūtīšana Zabbix aģentam

Šis noildzes iestatījums **netiks** izmantots tām pārbaudēm, kurām lietotāja saskarnē ir konfigurēti elastīgā noildze iestatījumi (globālā, 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

TLSAccept

Norādiet šifrēšanas līmeni, kas nepieciešams ienākošajiem savienojumiem no Zabbix serveris. Tiek izmantots pasīvajam starpniekserveris, aktīvajam starpniekserveris tiek ignorēts. 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.

Obligāti: jā pasīvajam starpniekserveris, ja ir definēti TLS sertifikāta vai PSK parametri (pat *unencrypted* savienojumam); citādi nē

TLSCAFile

Norādiet pilnu ceļu uz failu, kurā ir augstākā līmeņa CA sertifikāti, kas tiek izmantoti vienaudžu sertifikātu verifikācijai un šifrētai saziņai starp Zabbix komponentiem.

TLSCertFile

Norādiet pilnu ceļu uz failu, kurā atrodas starpniekservera sertifikāts vai sertifikātu ķēde, kas tiek izmantota šifrētai saziņai starp Zabbix komponentēm.

TLSCipherAll

Norādiet GnuTLS prioritātes virkni vai OpenSSL (TLS 1.2) šifru virkni. Pārrakstiet noklusējuma šifru komplekta atlases kritērijus sertifikātu un PSK balstītai šifrēšanai.

Piemērs:

```
TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSCipherAll13

Norādiet šifru virkni OpenSSL 1.1.1 vai jaunākai versijai TLS 1.3 režīmā. Pārrakstiet noklusējuma ciphersuite atlases kritērijus sertifikātu un PSK balstītai šifrēšanai.

GnuTLS piemērs:

```
NONE:+VERS-TLS1.2:+ECDHE-RSA:+RSA:+ECDHE-PSK:+PSK:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL
```

OpenSSL piemērs:

```
EECDH+aRSA+AES128:RSA+aRSA+AES128:kECDHEPSK+AES128:kPSK+AES128
```

TLSCipherCert

Norādiet GnuTLS prioritātes virkni vai OpenSSL (TLS 1.2) šifru virkni. Pārrakstiet noklusējuma šifru komplekta atlases kritērijus sertifikātu balstītai šifrēšanai.

GnuTLS piemērs:

```
NONE:+VERS-TLS1.2:+ECDHE-RSA:+RSA:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL:+COMP-NULL:+SIG
```

OpenSSL piemērs:

```
EECDH+aRSA+AES128:RSA+aRSA+AES128
```

TLSCipherCert13

Norādiet šifru virkni OpenSSL 1.1.1 vai jaunākai versijai TLS 1.3 režīmā. Pārrakstiet noklusējuma šifru kopas atlases kritērijus sertifikātā balstītai šifrēšanai.

TLSCipherPSK

Norādiet GnuTLS prioritātes virkni vai OpenSSL (TLS 1.2) šifru virkni. Pārrakstiet noklusējuma šifru komplekta atlases kritērijus PSK balstītai šifrēšanai.

GnuTLS piemērs:

```
NONE:+VERS-TLS1.2:+ECDHE-PSK:+PSK:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL:+COMP-NULL:+SIG
```

OpenSSL piemērs:

```
kECDHEPSK+AES128:kPSK+AES128
```

TLSCipherPSK13

Norādiet OpenSSL 1.1.1 vai jaunākas versijas šifru virkni TLS 1.3 režīmā. Pārrakstiet noklusējuma šifru kopas atlases kritērijus PSK balstītai šifrēšanai.

Piemērs:

```
TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSConnect

Norādiet šifrēšanas līmeni, kas nepieciešams starpniekservera savienojumiem ar Zabbix serveri. Tiek izmantots aktīvam starpniekserverim, tiek ignorēts pasīvam starpniekserverim. Var norādīt tikai vienu vērtību: - *unencrypted* - savienoties bez šifrēšanas (noklusējums); - *psk* - savienoties, izmantojot TLS un iepriekš koplietotu atslēgu (PSK); - *cert* - savienoties, izmantojot TLS un sertifikātu.

Obligāti: jā aktīvam starpniekserverim, ja ir definēti TLS sertifikāta vai PSK parametri (pat *unencrypted* savienojumam); citādi nē

TLSCRLFile

Norādiet pilnu ceļu uz failu, kurā ir atsauktie sertifikāti. Šis parametrs tiek izmantots šifrētai saziņai starp Zabbix komponentiem.

TLSKeyFile

Norādiet pilnu ceļu uz failu, kurā atrodas starpniekservera privātā atslēga, ko izmanto šifrētai saziņai starp Zabbix komponentēm.

TLSListen

Pieprasīt tikai TLS savienojumus trapper ligzdā.

Atbalstītās vērtības:

- required - pieņemt tikai TLS savienojumus

TLSPSKFile

Norādiet pilnu ceļu uz failu, kurā ir starpniekservera iepriekš koplietotā atslēga, kas tiek izmantota šifrētai saziņai ar Zabbix serveri.

TLSPSKIdentity

Norādiet iepriekš koplietotās atslēgas identitātes virkni, ko izmanto šifrētai saziņai ar Zabbix serveri.

TLSServerCertIssuer

Norādiet atļauto servera sertifikāta izdevēju.

TLSServerCertSubject

Norādiet atļauto servera sertifikāta subjektu.

TmpDir

Norādiet pagaidu direktoriju.

Noklusējums: `/tmp`

TrapperTimeout

Norādiet taimautu sekundēs šādiem procesiem:
 - konfigurācijas datu izgūšanai no Zabbix serveris;
 - globāla skripta izpildei vai attālinātas komandas izpildei.

Noklusējums: 300
 Diapazons: 1-300

Nav nepieejams aizkave

Iestatiet host pieejamības pārbaudes biežumu nepieejamības periodā sekundēs.

Noklusējums: 60
 Diapazons: 1-3600

UnreachableDelay

Iestatiet host pieejamības pārbaudes biežumu nepieejamības periodā sekundēs.

Noklusējums: 15
 Diapazons: 1-3600

Nesasniedzamības periods

Norāda, pēc cik sekundēm nesasniedzamības gadījumā hostu uzskatīt par nepieejamu.

Noklusējums: 45
 Diapazons: 1-3600

Lietotājs

Samazina privilēģijas līdz konkrētam, sistēmā jau esošam lietotājam.
Stājas spēkā tikai tad, ja tiek palaists kā `root` un `AllowRoot` ir atspējots.

Noklusējums: `zabbix`

Vault

Norādiet vault nodrošinātāju: - *HashiCorp* - HashiCorp KV Secrets Engine versija 2; - *CyberArk* - CyberArk Central Credential Provider
Jāatbilst vault nodrošinātājam, kas iestatīts lietotāja saskarnē.

Noklusējums: `HashiCorp`

VaultAppRoleID

Norādiet HashiCorp vault AppRole ID.
 Ja ir iestatīts, tiek iespējota AppRole autentifikācija. To nevar izmantot kopā ar VaultToken.
 Atbalstīts tikai HashiCorp vault.

VaultAppSecretID

Norādiet HashiCorp Vault AppRole slepeno ID.
 Atbalstīts tikai HashiCorp Vault.

VaultDBPath

Norādiet atrašanās vietu datubāzes piekļuves datu izgūšanai pēc atslēgām. Tā ir Vault ceļš vai vaicājums atkarībā no Vault.

Atslēgas, ko izmanto HashiCorp, ir `password` un `username`.

Ceļa piemērs ar `VaultPrefix=/v1/secret/data/zabbix/`:

```
database
```

Ceļa piemērs bez VaultPrefix:

```
secret/zabbix/database
```

Atslēgas, ko izmanto 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

Norādiet pielāgotu prefiksu Vault ceļam vai vaicājumam atkarībā no Vault. Ja tas nav norādīts, tiks izmantotas vispiemērotākās noklusējuma vērtības.
Piezīme: HashiCorp gadījumā pēc mountpoint automātiski tiek pievienots 'data', ja `VaultPrefix` nav norādīts.
Piezīme: HashiCorp gadījumā pēc mountpoint automātiski tiek pievienots 'data', ja `VaultPrefix` nav norādīts.

Prefiksa piemērs HashiCorp:

```
/v1/secret/data/zabbix/
```

Prefiksa piemērs Cyberark:

```
/AIMWebService/api/Accounts?
```

VaultTLSCertFile

Norādiet SSL sertifikāta faila nosaukumu, ko izmanto klienta autentifikācijai. Sertifikāta failam jābūt PEM1 formātā.
Ja sertifikāta fails satur arī privāto atslēgu, atstājiet SSL atslēgas faila lauku tukšu.
Direktoriju, kurā atrodas šis fails, nosaka SSLCertLocation konfigurācijas parametrs.
Šo opciju var neizmantot, taču tā ir ieteicama CyberArkCCP vault.

VaultTLSKeyFile

Norādiet SSL privātās atslēgas faila nosaukumu, ko izmanto klienta autentifikācijai. Privātās atslēgas failam jābūt PEM1 formātā.
Šī faila atrašanās vietas direktorijs ir norādīts ar konfigurācijas parametru SSLKeyLocation.
Šo opciju var neiekļaut, taču tā ir ieteicama CyberArkCCP vault.

VaultToken

Norādiet HashiCorp Vault autentifikācijas tokenu, kas jābūt ģenerētam tikai Zabbix starpniekserverim ar tikai lasīšanas tiesībām uz ceļu, kas norādīts izvēles VaultDBPath konfigurācijas parametrā.
Ja `VaultToken` un `VAULT_TOKEN` vides mainīgais ir definēti vienlaikus, tā ir kļūda.
 Nevar izmantot kopā ar VaultAppRoleID.
 Atbalstīts tikai HashiCorp Vault.

VaultURL

Norādiet vault servera URL. Ja SSLCALocation nav norādīts, tiks izmantots sistēmas mēroga CA sertifikātu direktorijs.

Noklusējums: `https://127.0.0.1:8200`

VMwareCacheSize

Iestatiet koplietotās atmiņas lielumu VMware datu glabāšanai.
VMware iekšējo pārbaudi `zabbix[vmware,buffer,...]` var izmantot, lai uzraudzītu VMware kešatmiņas lietojumu (skatiet Iekšējās pārbaudes).
Ņemiet vērā, ka koplietotā atmiņa netiek piešķirta, ja nav konfigurētu vmware savācēja instanču, kas jāstartē.

Noklusējums: 8M
 Diapazons: 256K-2G

VMwareFrequency

Iestatiet datu apkopošanas biežumu no viena VMware pakalpojuma (sekundēs).
Šis biežums jāiestata kā mazākais atjaunināšanas intervāls jebkuram VMware uzraudzības vienumam.

Noklusējums: 60
 Diapazons: 10-86400

VMwarePerfFrequency

Iestatiet veiktspējas skaitītāju statistikas iegūšanas biežumu no viena VMware pakalpojuma (sekundēs).
Šis biežums jāiestata uz mazāko atjaunināšanas intervālu jebkuram VMware monitoringa vienumam, kas izmanto VMware veiktspējas skaitītājus.

Noklusējums: 60
 Diapazons: 10-86400

VMwareTimeout

Iestata maksimālo sekunžu skaitu, cik ilgi `vmware collector` gaidīs atbildi no VMware pakalpojuma (vCenter vai ESX hipervizora).

Noklusējums: 10
 Diapazons: 1-300

WebDriverURL

Norādiet WebDriver saskarnes URL.

Piemērs (izmantošanai ar Selenium WebDriver standalone serveri):

```
WebDriverURL=http://localhost:4444
```

Aģents

Pārskats Zabbix aģents tiek izvietots uz uzraudzības mērķa, lai aktīvi uzraudzītu lokālos resursus un lietojumprogrammas (cietos diskus, atmiņu, procesora statistiku u. c.).

Aģents lokāli apkopo operatīvo informāciju un nosūta datus Zabbix serverim turpmākai apstrādei. Ja rodas kļūmes (piemēram, cietais disks ir pilns vai pakalpojuma process ir avarējis), Zabbix serveris var aktīvi brīdināt konkrētās ierīces administratorus, kas ziņoja par kļūmi.

Zabbix aģenti ir ļoti efektīvi, jo statistiskās informācijas apkopošanai tiek izmantoti vietējie sistēmas izsaukumi.

Vienā hosts ir iespējams palaist vairākus aģenta eksemplārus.

Skatiet arī:

- Pasīvo un aktīvo pārbaužu protokoli
- Aģenta palaišana kā root
- Konfigurācijas parametri: Zabbix aģents
- Konfigurācijas parametri: Zabbix aģents (Windows)
- Zabbix aģents operētājsistēmā Microsoft Windows
- Zabbix aģents operētājsistēmā IBM AIX

Pasīvie un aktīvie pārbaudes veidi

Zabbix aģenti var veikt pasīvās un aktīvās pārbaudes:

- Pasīvās pārbaudes - Zabbix aģents atbild uz Zabbix servera (vai starpniekservera) pieprasījumu. Piemēram, serveris pieprasa datus (piem., CPU noslodzi), un aģents atgriež rezultātu.
- Aktīvās pārbaudes - Zabbix aģents apkopo un nosūta datus, negaidot Zabbix servera (vai starpniekservera) pieprasījumu. Vispirms tas izgūst uzraudzības vienumu sarakstu no servera (CPU noslodzi, pieejamo atmiņu utt.), pēc tam apkopo nepieciešamos datus un periodiski nosūta tam jaunās vērtības.

Aģenta pārbaudes veids tiek konfigurēts, izvēloties attiecīgo uzraudzības vienuma tipu. Zabbix aģents apstrādā tipa *Zabbix agent* vai *Zabbix agent (active)* vienumus.

Atbalstītās platformas Par atbalstītajām platformām skatiet lapu Prasības.

Aģents Unix tipa sistēmās Zabbix aģents Unix tipa sistēmās tiek palaists uz uzraudzītā hosta.

Instalēšana

Zabbix aģentu Linux sistēmās var instalēt, izmantojot kādu no tālāk norādītajām metodēm:

- Zabbix pakotnes - atlasiet komponentu Agent (pēc Zabbix versijas, operētājsistēmas distribūcijas un operētājsistēmas versijas atlasīšanas) un izpildiet sniegtos norādījumus;
- Zabbix pirmkods - lejupielādējiet pirmkoda failus un kompilējiet Zabbix aģentu, konfigurējot pirmkodu ar opciju `--enable-agent`.

Attention:
Parasti 32 bitu Zabbix aģenti darbosies 64 bitu sistēmās, taču dažos gadījumos 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 servera/starpniekservera versiju, ir pieejami operētājsistēmām NetBSD un HP-UX. Informāciju par aģenta instalēšanu un palaišanu IBM AIX skatiet sadaļā Zabbix aģents IBM AIX sistēmā.

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
```

Ja instalēts no binārā arhīva

Pēc arhīva izvilkšanas atrodiet `zabbix_agentd` bināro failu un `zabbix_agentd.conf` konfigurācijas failu. To atrašanās vieta arhīvā ir atkarīga no platformas (piemēram, `sbin/` un `conf/` vai `usr/sbin/` un `etc/zabbix/`). Palaidiet bināro failu tieši no izvilktās direktorijas un norādiet konfigurācijas failu, izmantojot opciju `-c`. Piemēram:

```
/opt/zabbix/sbin/zabbix_agentd -c /opt/zabbix/etc/zabbix/zabbix_agentd.conf
```

UNIX līdzīgās sistēmās Zabbix aģents ir paredzēts darbībai ar lietotāja, kas nav root lietotājs, tiesībām. Ja tas tiek palaists ar root lietotāja tiesībām, tas pārslēdzas uz lietotāju `zabbix` (vai lietotāju, kas norādīts ar parametru `[User](<PARAMS>#user)`), ja vien konfigurācijas failā nav iestatīts `[AllowRoot=1](<PARAMS>#allowroot)`. Šim lietotājam sistēmā ir jābūt; pretējā gadījumā aģents netiek palaists.

IBM AIX sistēmā aģentu var palaist arī, izmantojot System Resource Controller (SRC), un konfigurēt tā palaišanu sistēmas startēšanas laikā; skatiet Zabbix aģents IBM AIX sistēmā.

Manuāla palaišana

Zabbix aģentu var palaist, tieši izpildot `zabbix_agentd` bināro failu, piemēram:

```
zabbix_agentd
```

Tas darbojas, ja direktorija, kurā atrodas `zabbix_agentd`, ir iekļauta mainīgajā `PATH`.

Aģents Windows sistēmās Zabbix aģents operētājsistēmā Windows darbojas kā Windows pakalpojums.

Instalēšana

Zabbix aģentu var instalēt operētājsistēmā Windows, izmantojot vienu no šīm metodēm:

- Iepriekš kompilēti Zabbix aģenta binārie faili - lejupielādējiet Zabbix aģenta MSI instalēšanas pakotni un izpildiet norādījumus lapā Windows aģenta instalēšana no MSI;
- Zabbix avota faili - lejupielādējiet avota failus un izpildiet norādījumus lapā Zabbix aģenta kompilēšana operētājsistēmā Windows.

Papildu informāciju par Zabbix aģenta instalēšanu (no ZIP arhīva) kā Windows pakalpojumu skatiet lapā Zabbix aģents Microsoft Windows.

Opcijas Ar Zabbix aģentu var izmantot šādus komandrindas parametrus:

Parametrs	Apraksts
UNIX un Windows aģents	
`-c --config <config-file>`	Konfigurācijas faila ceļš. Šo opciju var izmantot, lai norādītu konfigurācijas failu, kas nav noklusējuma fails.
`-f --foreground`	Palaist Zabbix aģentu priekšplānā (noklusējums: `true`).
`-p --print`	Izdrukāt zināmos vienumus un iziet. Lai atgrieztu arī lietotāja parametra rezultātus, jānorāda konfigurācijas fails (ja tas neatrodas noklusējuma atrašanās vietā).
`-t --test <item key>`	Pārbaudīt norādīto vienumu un iziet. Lai atgrieztu arī lietotāja parametra rezultātus, jānorāda konfigurācijas fails (ja tas neatrodas noklusējuma atrašanās 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 sadaļu izpildlaika vadība.
Tikai Windows aģents	
`-m --multiple-agents`	Izmantot vairākus aģenta eksemplārus (ar opcijām `-i`, `-d`, `-s`, `-x`). Lai atšķirtu eksemplāru pakalpojumu nosaukumus, katra pakalpojuma nosaukumā tiks iekļauta Hostname vērtība no norādītā konfigurācijas faila.
`-S --startup-type <value>`	Iestatīt Zabbix Windows aģenta pakalpojuma startēšanas tipu. Atļautās vērtības: `automatic` - *(noklusējums)* automātiski palaist pakalpojumu, startējot Windows; `delayed` - atlikt pakalpojuma startēšanu, līdz automātiski palaistie pakalpojumi ir pabeiguši startēšanu (pieejams operētājsistēmā Windows Server 2008/Vista un jaunākās versijās); `manual` - palaist pakalpojumu manuāli (lietotājam vai lietojumprogrammai); `disabled` - atspējot pakalpojumu, lai to nevarētu palaist lietotājs vai lietojumprogramma. Šo opciju var izmantot kopā ar opciju `-i` vai atsevišķi, lai mainītu jau instalēta pakalpojuma startēšanas tipu.
`-i --install`	Instalēt Zabbix Windows aģentu kā pakalpojumu.
`-d --uninstall`	Atinstalēt Zabbix Windows aģenta pakalpojumu.
`-s --start`	Palaist Zabbix Windows aģenta pakalpojumu.
`-x --stop`	Apturēt Zabbix Windows aģenta pakalpojumu.

Konkrēti komandrindas parametru izmantošanas piemēri:

- Izdrukāt visus iebūvētos aģenta vienumus ar vērtībām.
- Pārbaudīt lietotāja parametru ar `mysql.ping` atslēgu, kas definēta norādītajā konfigurācijas failā.
- Instalēt Zabbix Agent pakalpojumu operētājsistēmai Windows, izmantojot noklusējuma ceļu uz konfigurācijas failu.
- Instalēt Zabbix Agent [Hostname] pakalpojumu operētājsistēmai Windows, izmantojot konfigurācijas failu, kas atrodas tajā pašā mapē, kur aģenta izpildāmais fails, un padarīt pakalpojuma nosaukumu unikālu, paplašinot to ar `Hostname` vērtību no konfigurācijas faila.
- Mainīt instalēta Zabbix Agent pakalpojuma operētājsistēmai Windows startēšanas tipu, izmantojot konfigurācijas failu, kas atrodas tajā pašā mapē, kur aģenta izpildāmais fails.

```
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
```

Darbības laika vadība

Izmantojot darbības laika vadības opcijas, varat mainīt aģenta procesu žurnalēšanas līmeni.

Option	Description	Target
`log_level_increase[=<target>]`	Palielināt žurnalēšanas līmeni. Ja mērķis nav norādīts, tiek ietekmēti visi procesi.	Mērķi var norādīt šādi: **procesa tips** - visi norādītā tipa procesi (piemēram, `listener`) Skatiet visus aģenta procesu tipus. **procesa tips,N** - procesa tips un numurs (piemēram, `listener,3`) **pid** - procesa identifikators (1 līdz 65535). Lielākām vērtībām norādiet mērķi formātā "procesa tips,N".
`log_level_decrease[=<target>]`	Samazināt žurnalēšanas līmeni. Ja mērķis nav norādīts, tiek ietekmēti visi procesi.	
`userparameter_reload`	Atkārtoti ielādē opciju `UserParameter` un `Include` vērtības no pašreizējā konfigurācijas faila.	

Piemēri:

- Visu procesu žurnalēšanas līmeņa palielināšana
- Trešā listener procesa žurnalēšanas līmeņa palielināšana
- Procesa ar PID 1234 žurnalēšanas līmeņa palielināšana
- Visu aktīvās pārbaudes procesu žurnalēšanas 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:
Darbības laika vadība netiek atbalstīta operētājsistēmās OpenBSD, NetBSD un Windows.

Aģenta procesu tipi Aģenta procesu tipi ir šādi:

- `active checks` - process aktīvo pārbaužu veikšanai;
- `collector` - datu vākšanas process;
- `listener` - process pasīvo pārbaužu uzklausīšanai.

Aģenta žurnāla failu var izmantot, lai novērotu šos procesu tipus.

Aģenta žurnāla fails tiek izveidots ar lasīšanas un rakstīšanas atļaujām tikai faila īpašniekam. Turklāt failu var lasīt īpašnieka grupa. Visas pārējās atļaujas ir liegtas.

Procesa lietotājs Zabbix aģents UNIX sistēmā ir paredzēts darbībai kā lietotājam, kas nav root lietotājs. Tas darbosies kā tas lietotājs, kas nav root lietotājs un ar kura tiesībām tas ir palaists. Tāpēc aģentu varat palaist kā jebkuru lietotāju, kas nav root lietotājs, bez problēmām.

Ja mēģināsiet to palaist kā root lietotāju, tas pārslēgsies uz cieti kodētu zabbix lietotāju, kuram jābūt pieejamam jūsu sistēmā. Aģentu varat palaist kā root lietotāju tikai tad, ja attiecīgi modificējat parametru `AllowRoot` aģenta konfigurācijas failā.

Konfigurācijas fails Konfigurācijas faila noklusējuma ceļš ir atkarīgs no operētājsistēmas un Zabbix instalēšanas metodes:

- UNIX sistēmā noklusējuma ceļš ir:
  - Instalējot no pirmkoda: `/usr/local/etc/zabbix_agentd.conf` vai ceļš, kas iestatīts ar kompilēšanas laikā izmantotajiem mainīgajiem `--sysconfdir` vai `--prefix`.

    - Instalējot no pakotnēm vai appliance: `/etc/zabbix/zabbix_agentd.conf`.
  - Windows sistēmā noklusējuma ceļš ir `C:\Program Files\Zabbix Agent\zabbix_agentd.conf`.

Ja hosts darbojas viena aģenta instance, tā var izmantot noklusējuma konfigurācijas failu vai konfigurācijas failu, kas norādīts komandrindā. Ja darbojas vairākas instances, katrai aģenta instancei ir jābūt savam konfigurācijas failam (viena no instancēm var izmantot noklusējuma konfigurācijas failu).

Plašāku informāciju par Zabbix aģenta konfigurēšanu skatiet konfigurācijas faila opcijās Unix Zabbix aģentam vai Windows Zabbix aģentam.

Lokāle Ņemiet vērā, ka aģentam ir nepieciešama UTF-8 lokāle, lai daži teksta aģenta vienumi varētu atgriezt paredzēto saturu. Lielākajai daļai mūsdienu Unix tipa sistēmu UTF-8 lokāle ir iestatīta kā noklusējuma, tomēr dažās sistēmās tā var būt jāiestata īpaši.

Iziešanas kods Zabbix aģents atgriež 0 veiksmīgas iziešanas gadījumā un 1 kļūmes gadījumā.

Pasīvo un aktīvo pārbaužu protokoli

Šī sadaļa sniedz protokola informāciju par pasīvajām un aktīvajām pārbaudēm, ko veic Zabbix aģents un Zabbix aģents 2.

Zabbix saziņai ar aģentiem izmanto uz JSON balstītu saziņas protokolu.

Kopš Zabbix 7.0 Zabbix aģents un Zabbix aģents 2 protokoli ir apvienoti. Atšķirība starp Zabbix aģents un Zabbix aģents 2 pieprasījumiem/atbildēm tiek izteikta ar "variant" taga vērtību.

Pasīvie pārbaužu pieprasījumi

Pasīvā pārbaude ir vienkāršs datu pieprasījums. Zabbix serveris vai starpniekserveris pieprasa noteiktus datus (piemēram, CPU noslodzi), un Zabbix aģents nosūta rezultātu atpakaļ uz serveri.

Pasīvās pārbaudes tiek izpildītas asinhroni - nav nepieciešams saņemt viena pieprasījuma atbildi, pirms tiek sākta cita pārbaude. DNS atrisināšana arī ir asinhrona.

Aģenta aptaujātājs mēģinās izveidot savienojumu ar visām adresēm, ko atgriež DNS uzmeklēšana. Tas nodrošina, ka, ja viena IP adrese nav sasniedzama, aptaujātājs mēģinās izmantot nākamo pieejamo adresi, palielinot veiksmīga savienojuma iespējamību. Šis uzlabojums attiecas gan uz Zabbix serveri, gan starpniekserveri.

Asinhrono pārbaužu maksimālais paralēlais skaits ir 1000 (definēts ar MaxConcurrentChecksPerPoller).

Asinhrono aģenta aptaujātāju skaits ir definēts ar parametru StartAgentPollers.

Servera pieprasījums

Galvenes un datu garuma definīciju skatiet protokola detaļās.

```
{
  "request": "passive checks",
  "data": [
    {
      "key": "agent.version",
      "timeout": 3
    }
  ]
}
```

Field		Type	Mandatory	Value
request		*string*	yes	`"passive checks"`
data		*array of object*	yes	Pasīvās pārbaudes vienums.
	key	*string*	yes	Vienuma atslēga ar izvērstiem makro.
	timeout	*number*	yes	Sakaru noildze.

Aģenta atbilde

```
{
  "version": "8.0.0",
  "variant": 2,
  "data": [
    {
```

```
      "value": "8.0.0"
    }
  ]
}
```

Field		Type	Mandatory	Value
version		*string*	yes	Aģenta versijas numurs.
variant		*number*	yes	Aģenta variants (*1* - Zabbix aģents, *2* - Zabbix agent 2).
data		*array of object*	yes	Satur pārbaudes rezultātu.
	value	*string*	no	Vienuma vērtība, ja pārbaude bija veiksmīga.
	error	*string*	no	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

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 neatbalstītu 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, kurām ir vienkāršteksta protokols, ir ieviesta pārslēgšanās uz veco protokolu.

Pasīvās pārbaudes pēc restartēšanas vai pēc saskarnes konfigurācijas izmaiņām tiek veiktas, izmantojot JSON protokolu (7.0 un jaunākas versijas).
Ja atbildē netiek saņemts derīgs JSON (aģents nosūtīja "ZBX_NOTSUPPORTED"), Zabbix saglabās saskarni kā vecā protokola saskarni un **atkārtos** pārbaudi, nosūtot tikai vienuma atslēgu.

Ņemiet vērā, ka katru stundu Zabbix serveris/starpniekserveris atkal mēģinās strādāt ar jauno protokolu ar visām saskarnēm, vajadzības gadījumā atgriežoties pie vecā protokola.

Aktīvās pārbaudes

Aktīvajām pārbaudēm nepieciešama sarežģītāka apstrāde.
Aģentam vispirms no servera/starpniekservera jāiegūst vienumu un/vai attālo komandu saraksts neatkarīgai apstrādei.

Serveri/starpniekserveri, no kuriem jāiegūst 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, tā tiek atkārtota pēc iepriekš definētām 60 sekundēm.

Note:
Kopš Zabbix 6.4 aģents (aktīvajā režīmā) vairs nesaņem no servera/starpniekservera pilnu konfigurācijas kopiju reizi divās minūtēs (pēc noklusējuma). Tā vietā, lai samazinātu tīkla trafiku un resursu patēriņu, 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 ja ir mainījusies hosts konfigurācija, globālie makro vai globārie regulārie izteicieni.

Aģents pēc tam periodiski nosūta jaunās vērtības uz serveri(-iem). Ja aģents ir saņēmis kādas attālās komandas izpildei, tiks nosūtīts arī izpildes rezultāts. Ņemiet vērā, ka attālo komandu izpilde aktīvajā aģentā tiek atbalstīta kopš Zabbix aģents 7.0.

Note:
Ja aģents atrodas aiz ugunsmūra, varat apsvērt tikai aktīvo pārbaužu izmantošanu, jo šajā gadījumā jums nebūtu jā maina ugunsmūra iestatījumi, lai atļautu sākotnējos ienākošos savienojumus.

Vienumu saraksta iegūšana

Aģenta pieprasījums

Aktīvo pārbaužu pieprasījums tiek izmantots, lai iegūtu aktīvās pārbaudes, kuras apstrādās aģents.
Šo pieprasījumu aģents nosūta startēšanas brīdī un pēc tam ar RefreshActiveChecks intervāliem.

```
{
  "request": "active checks",
  "host": "Zabbix serveris",
  "host_metadata": "mysql,nginx",
  "interface": "zabbix.server.lan",
  "ip": "159.168.1.1",
  "port": 12050,
  "version": "8.0.0",
  "variant": 2,
  "config_revision": 1,
  "session": "e3dcbd9ace2c9694e1d7bbd030eeef6e"
}
```

Field	Type	Mandatory	Value
request	*string*	yes	`active checks`
host	*string*	yes	Host name.
host_metadata	*string*	no	The configuration parameter HostMetadata or HostMetadataItem metric value.
interface	*string*	no	The configuration parameter HostInterface or HostInterfaceItem metric value.
ip	*string*	no	The configuration parameter ListenIP first IP if set.
port	*number*	no	The configuration parameter ListenPort value if set and not default agent listening port.
version	*string*	yes	The agent version number.
variant	*number*	yes	The agent variant (*1* - Zabbix agent, *2* - Zabbix agent 2).
config_revision	*number*	no	Configuration identifier for incremental configuration sync.
session	*string*	no	Session identifier for incremental configuration sync.

Servera atbilde

Aktīvo pārbaužu atbilde tiek nosūtīta no servera 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	Aktīvās pārbaudes vienumi. Netiek iekļauts, ja host konfigurācija nav mainījusies.
key	*string*	no	Vienuma atslēga ar izvērstiem makro.
itemid	*number*	no	Vienuma identifikators.
delay	*string*	no	Vienuma atjaunināšanas intervāls. Elastīgie/plānošanas intervāli tiek atbalstīti gan Zabbix agent, gan Zabbix agent 2 kopš 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. Iekļauts, ja attālinātās komandas izpildi ir aktivizējusi darbības operācija vai manuāla skripta izpilde. Ņemiet vērā, ka attālināto komandu izpilde uz aktīva aģenta tiek atbalstīta kopš Zabbix agent 7.0. Vecāki aktīvie aģenti ignorēs jebkuras attālinātās komandas, kas iekļautas aktīvo pārbaužu servera 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 manuāla skripta izpildes).
timeout	*number*	no	Attālinātās komandas izpildes noildze serveris/starpniekserveris konfigurācijā.
config_revision	*number*	no	Konfigurācijas identifikators inkrementālai konfigurācijas sinhronizācijai. Netiek iekļauts, ja host konfigurācija nav mainījusies. Tiek palielināts, ja host konfigurācija ir mainīta.

Serverim jāatbild ar success.

Piemēram:

1. Aģents atver TCP savienojumu
2. Aģents pieprasa pārbaužu sarakstu
3. Serveris atbild ar vienumu un izpildāmo attālināto komandu sarakstu
4. Aģents parsē atbildi
5. TCP savienojums tiek aizvērts
6. Aģents sāk periodisku datu vākšanu un izpilda attālinātās komandas (atbalstīts kopš Zabbix agent 7.0)

Attention:
Ņemiet vērā, ka (sensitīvi) konfigurācijas dati var kļūt pieejami pusēm, kurām ir piekļuve Zabbix servera trapper portam, izmantojot aktīvo pārbaudi. Tas ir iespējams, jo jebkurš var izlikties par aktīvu aģentu un pieprasīt vienuma konfigurācijas datus; autentifikācija nenotiek, ja vien netiek izmantotas šifrēšanas opcijas.

Nosūtīšana ar savāktajiem datiem

Aģents nosūta

Aģenta datu pieprasījums satur savāktās vienumu vērtības un izpildīto attālo 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
}
```


Field	Type	Mandatory	Value
request	*string*	yes	`agent data`
data	*array of objects*	yes	Vienumu vērtības.
id	*number*	yes	Vērtības identifikators (pieaugošs skaitītājs, ko izmanto, lai pārbaudītu dublētas vērtības tīkla problēmu gadījumā).
itemid	*number*	yes	Vienuma identifikators.
value	*string*	no	Vienuma vērtība.
lastlogsize	*number*	no	Vienuma lastlogsize.
mtime	*number*	no	Vienuma mtime.
state	*number*	no	Vienuma state.
source	*string*	no	Vērtības notikumu žurnāla avots.
eventid	*number*	no	Vērtības notikumu žurnāla eventid.
severity	*number*	no	Vērtības notikumu žurnāla severity.
timestamp	*number*	no	Vērtības notikumu žurnāla timestamp.
clock	*number*	yes	Vērtības laikspiedols (sekundes kopš Epoch).
ns	*number*	yes	Vērtības laikspiedola nanosekundes.
commands	*array of objects*	no	Attālo komandu izpildes rezultāts. Ņemiet vērā, ka attālo komandu izpilde aktīvajā aģentā tiek atbalstīta kopš Zabbix aģenta 7.0. Vecāki aktīvie aģenti ignorēs jebkuras attālās komandas, kas iekļautas aktīvo pārbaužu servera atbildē.
id	*number*	no	Attālās komandas identifikators.
value	*string*	no	Attālās komandas izpildes rezultāts, ja izpilde bija veiksmīga.
error	*string*	no	Attālās komandas izpildes kļūdas ziņojums, ja izpilde neizdevās.
session	*string*	yes	Unikāls sesijas identifikators, kas tiek ģenerēts katru reizi, kad aģents tiek palaists (32 heksadecimālu ciparu garš).
host	*string*	yes	Hosts nosaukums.
version	*string*	yes	Aģenta versijas numurs.
variant	*number*	yes	Aģenta variants (*1* - Zabbix aģents, *2* - Zabbix aģents 2).

Katram vērtības ierakstam tiek piešķirts virtuāls ID. Vērtības ID ir vienkāršs augošs skaitītājs, kas ir unikāls vienas datu sesijas

ietvaros (identificēts pēc sesijas tokena). Šis ID tiek izmantots, lai atmestu dublētas vērtības, kas var tikt nosūtītas sliktas savienojamības vidēs.

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"
}
```

Field	Type	Mandatory	Value
response	*string*	yes	`success` | `failed`
info	*string*	yes	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.

Sirdspuksta ziņojums

Aģents nosūta

Sirdspuksta ziņojumu aktīvais aģents nosūta Zabbix serverim/starpniekserverim ik pēc `HeartbeatFrequency` sekundēm (konfigurēts Zabbix aģenta vai Zabbix agent 2 konfigurācijas failā).

To izmanto aktīvo pārbaužu pieejamības uzraudzībai.

```
{
  "request": "active check heartbeat",
  "host": "Zabbix server",
  "heartbeat_freq": 60,
  "version": "8.0.0",
  "variant": 2
}
```

Field	Type	Mandatory	Value
request	*string*	yes	`active check heartbeat`
host	*string*	yes	Hosts nosaukums.
heartbeat_freq	*number*	yes	Aģenta sirdspuksta biežums (`HeartbeatFrequency` konfigurācijas parametrs).
version	*string*	yes	Aģenta versijas numurs.
variant	*number*	yes	Aģenta variants (*1* - Zabbix aģents, *2* - Zabbix agent 2).

Novirzīšanas atbilde

Kad hosts ir piešķirts citam serverim, serveris var norādīt aģentam novirzīt sirdspuksta ziņojumus (un turpmākās aktīvās pārbaudes) uz citu starpniekserveri vai servera instanci.

```
  {
    "response": "failed",
    "redirect": {
      "revision": 2,
      "address": "192.0.2.1:10055"
```

```
    }
  }
```

Field		Type	Mandatory	Value
response		*string*	yes	`success` | `failed`
redirect		*object*	yes	Novirzīšanas norādījumi.
	revision	*number*	yes	Konfigurācijas revīzijas identifikators.
	address	*string*	yes	Mērķa servera/starpniekservera adrese.

Vecāks XML protokols

Note:
Zabbix pieņems līdz 16 MB XML Base64 kodētu datu, taču viena dekodētā vērtība nedrīkst pārsniegt 64 KB; pretējā gadījumā dekodēšanas laikā tā tiks saīsināta līdz 64 KB.

Aģenta palaišana kā root

Kopš Zabbix **5.0.0** oficiālo pakotņu Zabbix aģenta systemd pakalpojuma fails skaidri ietver `User` un `Group` direktīvas. Abas ir iestatītas uz `zabbix`.

Vairs nav iespējams konfigurēt, ar kādu lietotāju Zabbix aģents tiek palaists, izmantojot `zabbix_agentd.conf` failu, jo aģents ignorēs šo konfigurāciju un tiks palaists kā lietotājs, kas norādīts systemd pakalpojuma failā. Lai palaistu Zabbix aģentu kā root, jums jāveic tālāk aprakstītās izmaiņas.

Zabbix aģents

Lai ignorētu Zabbix aģenta noklusējuma lietotāju un grupu, izpildiet:

```
systemctl edit zabbix-agent
```

Pēc tam pievienojiet tālāk norādīto saturu:

```
[Service]
User=root
Group=root
```

Pārlādējiet dēmonus un restartējiet zabbix-agent servisu:

```
systemctl daemon-reload
systemctl restart zabbix-agent
```

Zabbix aģentam tādējādi tiek atkārtoti iespējota lietotāja konfigurēšanas funkcionalitāte aģenta konfigurācijas failā. Tagad konfigurācijas failā 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 pakalpojumu:

```
systemctl daemon-reload
systemctl restart zabbix-agent2
```

Zabbix aģentam 2 tas pilnībā nosaka lietotāju, ar kādu tas darbojas. Papildu izmaiņas nav nepieciešamas.

Konfigurācijas parametri: Zabbix aģents

Pārskats

Konfigurācijas parametri ļauj pielāgot Zabbix aģenta darbību.

Norādiet parametru vērtības Zabbix aģenta konfigurācijas failā (`zabbix_agentd.conf`). Visi parametri ir **neobligāti**, ja vien nav skaidri norādīts, ka parametrs ir obligāts.

Šī lapa ir strukturēta šādi:

- Parametru pārskats (lai skatītu pilnu informāciju par parametru, noklikšķiniet uz tā nosaukuma)
- Parametru detalizēts apraksts

Parametrs	Apraksts
Alias	Iestatīt vienuma atslēgas aizstājvārdu.
AllowKey	Atļaut to vienumu atslēgu izpildi, kas atbilst paraugam.
AllowKeyRegexp	Atļaut to vienumu atslēgu izpildi, kas atbilst uz regulāro izteiksmju balstītam paraugam.
AllowRoot	Atļaut aģentam darboties kā `root`.
BufferSend	Iestatīt vērtību nosūtīšanas no bufera uz Zabbix serveri/starpniekserveri biežumu.
BufferSize	Iestatīt maksimālo vērtību skaitu atmiņas buferī.
DebugLevel	Iestatīt atkļūdošanas līmeni.
DenyKey	Aizliegt to vienumu atslēgu izpildi, kas atbilst paraugam.
DenyKeyRegexp	Aizliegt to vienumu atslēgu izpildi, kas atbilst uz regulāro izteiksmju balstītam paraugam.
EnableRemoteCommands	Atļaut attālās komandas no Zabbix servera. Šis parametrs ir novecojis; tā vietā izmantojiet `AllowKey`, `AllowKeyRegexp`, `DenyKey` vai `DenyKeyRegexp`.
HeartbeatFrequency	Iestatīt sirdspukstu ziņojumu biežumu sekundēs.
HostInterface	Norādīt hosts saskarni.
HostInterfaceItem	Norādīt vienumu hosts saskarnes iegūšanai.
HostMetadata	Norādīt hosts metadatus.
HostMetadataItem	Norādīt Zabbix aģenta vienumu hosts metadatu iegūšanai.
Hostname	Norādīt hosts nosaukumu.
HostnameItem	Norādīt Zabbix aģenta vienumu hosts nosaukuma iegūšanai.
Include	Norādīt direktoriju vai atsevišķus failus, kas jāiekļauj konfigurācijas failā.
ListenBacklog	Iestatīt maksimālo neapstrādāto savienojumu skaitu TCP rindā.
ListenIP	Norādīt aģenta klausīšanās IP adresi savienojumiem no servera/starpniekservera.
ListenPort	Norādīt aģenta klausīšanās portu.
LoadModule	Norādīt moduli, kas jāielādē aģenta palaišanas laikā.
LoadModulePath	Norādīt pilnu ceļu uz aģenta moduļu atrašanās vietu.
LogFile	Norādīt aģenta žurnāla failu.
LogFileSize	Iestatīt maksimālo žurnāla faila lielumu.
LogRemoteCommands	Iespējot izpildīto čaulas komandu reģistrēšanu kā brīdinājumus.
LogType	Norādīt žurnāla izvades tipu.
MaxLinesPerSecond	Iestatīt maksimālo jauno rindu skaitu sekundē, kas tiek nosūtīts Zabbix serverim/starpniekserverim, apstrādājot aktīvās `log` vai `logrt` pārbaudes.
PidFile	Norādīt PID failu.
RefreshActiveChecks	Norādīt aktīvo pārbaužu saraksta atjaunināšanas biežumu.
Server	Norādīt Zabbix serveru/starpniekserveru IP adreses vai DNS nosaukumus.
ServerActive	Norādīt Zabbix servera/starpniekservera adresi vai klastera konfigurāciju, no kuras iegūt aktīvās pārbaudes.
SourceIP	Norādīt avota IP adresi.
StartAgents	Iestatīt iepriekš izveidoto `zabbix_agentd` eksemplāru skaitu pasīvo pārbaužu apstrādei.
Timeout	Norādīt, cik ilgi (sekundēs) gaidīt savienojuma izveidi un datu apmaiņu ar Zabbix starpniekserveri vai serveri.
TLSAccept	Norādīt ienākošajiem savienojumiem nepieciešamo šifrēšanas līmeni.
TLSCAFile	Norādīt pilnu ceļu uz failu, kurā ir augstākā līmeņa CA sertifikāti partnera sertifikāta pārbaudei; fails tiek izmantots šifrētai saziņai starp Zabbix komponentiem.
TLSCertFile	Norādīt pilnu ceļu uz failu, kurā ir aģenta sertifikāts vai sertifikātu ķēde; fails tiek izmantots šifrētai saziņai starp Zabbix komponentiem.
TLSCipherAll	Norādīt GnuTLS prioritāšu virkni vai OpenSSL (TLS 1.2) šifru virkni. Ignorēt noklusējuma šifru komplekta atlases kritērijus uz sertifikātiem un PSK balstītai šifrēšanai.
TLSCipherAll13	Norādīt šifru virkni OpenSSL 1.1.1 vai jaunākai versijai TLS 1.3 režīmā. Ignorēt noklusējuma šifru komplekta atlases kritērijus uz sertifikātiem un PSK balstītai šifrēšanai.
TLSCipherCert	Norādīt GnuTLS prioritāšu virkni vai OpenSSL (TLS 1.2) šifru virkni. Ignorēt noklusējuma šifru komplekta atlases kritērijus uz sertifikātiem balstītai šifrēšanai.
TLSCipherCert13	Norādīt šifru virkni OpenSSL 1.1.1 vai jaunākai versijai TLS 1.3 režīmā. Ignorēt noklusējuma šifru komplekta atlases kritērijus uz sertifikātiem balstītai šifrēšanai.
TLSCipherPSK	Norādīt GnuTLS prioritāšu virkni vai OpenSSL (TLS 1.2) šifru virkni. Ignorēt noklusējuma šifru komplekta atlases kritērijus uz PSK balstītai šifrēšanai.

Parametrs	Apraksts
TLSCipherPSK13	Norādīt šifru virkni OpenSSL 1.1.1 vai jaunākai versijai TLS 1.3 režīmā. Ignorēt noklusējuma šifru komplekta atlases kritērijus uz PSK balstītai šifrēšanai.
TLSConnect	Norādīt savienojumiem ar Zabbix serveri vai starpniekserveri nepieciešamo šifrēšanas līmeni.
TLSCRLFile	Norādīt pilnu ceļu uz failu, kurā ir atsaukto sertifikātu saraksts. Šis parametrs tiek izmantots šifrētai saziņai starp Zabbix komponentiem.
TLSKeyFile	Norādīt pilnu ceļu uz failu, kurā ir aģenta privātā atslēga; fails tiek izmantots šifrētai saziņai starp Zabbix komponentiem.
TLSPSKFile	Norādīt pilnu ceļu uz failu, kurā ir aģenta iepriekš koplietotā atslēga; fails tiek izmantots šifrētai saziņai ar Zabbix serveri vai starpniekserveri.
TLSPSKIdentity	Norādīt iepriekš koplietotās atslēgas identitātes virkni; tā tiek izmantota šifrētai saziņai ar Zabbix serveri vai starpniekserveri.
TLSServerCertIssuer	Norādīt atļauto servera (starpniekservera) sertifikāta izdevēju.
TLSServerCertSubject	Norādīt atļauto servera (starpniekservera) sertifikāta subjektu.
UnsafeUserParameters	Atļaut visus simbolus nodot lietotāja definēto parametru argumentos.
User	Atteikties no privilēģijām un izmantot konkrētu, sistēmā esošu lietotāju.
UserParameter	Norādīt uzraugāmu lietotāja definētu parametru.
UserParameterDir	Norādīt noklusējuma meklēšanas ceļu UserParameter komandām.

Ņemiet vērā:

- Noklusējuma vērtības atspoguļo dēmona noklusējuma iestatījumus, nevis piegādātajos konfigurācijas failos norādītās vērtības.
- Vērtības atbalsta vides mainīgos.
- Zabbix atbalsta tikai UTF-8 kodējumā esošus konfigurācijas failus bez BOM.
- Komentāri, kas sākas ar #, tiek atbalstīti tikai rindas sākumā.

Parametra informācija

Alias

Iestatiet vienuma atslēgas aizstājvārdu. To var izmantot, lai garu un sarežģītu vienuma atslēgu aizstātu ar īsāku un vienkāršāku.
 Var būt norādīti vairāki `Alias` parametri. Vairāki parametri ar vienādu `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 izgūšana.

```
Alias=zabbix.userid:vfs.file.regexp[/etc/passwd,"^zabbix:.:([0-9]+)",,,,\1]
```

Tagad datu izgūšanai var izmantot saīsināto atslēgu `zabbix.userid`.

2. piemērs: CPU izmantošanas 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 izmantošanas procentuālo vērtību ar noklusējuma parametriem, kā arī izmantot `cpu.util[all, idle, avg15]`, lai iegūtu konkrētus datus par CPU izmantošanu.

3. piemērs: vairāku zema 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]` u.c.

AllowKey

Atļauj izpildīt tos vienumu atslēgu nosaukumus, kas atbilst paraugam. Atslēgas paraugs ir aizstājējzīmju izteiksme, kas atbalsta * rakstzīmi, lai atbilstu jebkuram rakstzīmju skaitam.
 Vairākus atslēgu atbilstības noteikumus var definēt kopā ar `DenyKey`, `AllowKeyRegexp` un `DenyKeyRegexp`. Parametri tiek apstrādāti pa vienam atbilstoši to parādīšanās secībai. Secība nosaka, kuram no atbilstošajiem noteikumiem būs prioritāte. Skatiet arī: Restricting agent checks.

AllowKeyRegexp

Atļaut izpildīt tos vienumu taustiņus, kas atbilst paraugam.
Taustiņa paraugs ir regulāra izteiksme.
 Var definēt vairākus taustiņu atbilstības noteikumus kombinācijā ar `DenyKeyRegexp`,

AllowKey un DenyKey. Parametri tiek apstrādāti pa vienam atbilstoši to parādīšanās secībai. Secība nosaka, kuram atbilstības noteikumam būs prioritāte. Skatiet arī: Aģenta pārbaužu ierobežošana.

Piemērs:

```
AllowKeyRegexp=^system\.run\[(?:/usr)?/bin/hostname(?: -[fsd])?\]$
```

AllowRoot

Atļauj aģentam darboties kā root. Ja šī opcija ir atspējota un aģents tiek palaists ar root, aģents mēģinās pārslēgties uz lietotāju zabbix. Nav ietekmes, ja tas tiek palaists parasta lietotāja vārdā.

Noklusējums: 0
 Vērtības: 0 - neatļaut; 1 - atļaut

BufferSend

Iestatiet vērtību nosūtīšanas biežumu no bufera uz Zabbix serveri/starpniekserveri (sekundēs).

Noklusējums: 5
 Diapazons: 1-3600

BufferSize

Iestata maksimālo vērtību skaitu atmiņas buferī.
 Aģents nosūtīs visus savāktos datus uz Zabbix serveri vai starpniekserveri, ja buferis būs pilns.

Noklusējums: 100
 Diapazons: 2-65535

DebugLevel

Iestatiet 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

Liedz izpildīt tos vienumu taustiņus, kas atbilst paraugam. Taustiņa paraugs ir aizstājējzīmju izteiksme, kas atbalsta rakstzīmi *, lai atbilstu jebkuram skaitam jebkādu rakstzīmju.
 Vairākus taustiņu atbilstības noteikumus var definēt kopā ar AllowKey, AllowKeyRegexp un DenyKeyRegexp. Parametri tiek apstrādāti pa vienam to parādīšanās secībā. Secība nosaka, kuram atbilstības noteikumam būs prioritāte. Skatiet arī: Aģenta pārbaužu ierobežošana.

Noklusējums: DenyKey=system.run[*]

DenyKeyRegexp

Liedz izpildīt tos vienumu taustiņus, kas atbilst paraugam. Taustiņa paraugs ir regulārā izteiksme.
 Vairākus taustiņu atbilstības noteikumus var definēt kopā ar AllowKeyRegexp, AllowKey un DenyKey. Parametri tiek apstrādāti pa vienam atbilstoši to parādīšanās secībai. Secība nosaka, kuram atbilstības noteikumam būs prioritāte.
 Visi system.run[] vienumi pēc noklusējuma ir aizliegti (DenyKey=system.run[*]), ja vien nav norādīts cits system.run[*] noteikums.

Skatiet arī: Aģenta pārbaužu ierobežošana.

EnableRemoteCommands

Atļaut attālās komandas no Zabbix servera.
 Šis parametrs ir **novecojis**; tā vietā izmantojiet AllowKey=system.run[*], AllowKeyRegexp=^system\.run\[.*\]$, DenyKey=system.run[*] vai DenyKeyRegexp=^system\.run\[.*\]$.
 Tā ir iekšēja aizstājējvērtība parametriem AllowKey, AllowKeyRegexp, DenyKey un DenyKeyRegexp atkarībā no vērtības: - 0 - DenyKey=system.run[*] (DenyKeyRegexp=^system\.run\[.*\]$) - 1 - AllowKey=system.run[*] (AllowKeyRegexp=^system\.

Noklusējums: 0
 Vērtības: 0 - neatļaut, 1 - atļaut

HeartbeatFrequency

Iestatiet sirdspukstu ziņojumu biežumu sekundēs. Tiek izmantots aktīvo pārbaužu pieejamības uzraudzībai.
0 - sirdspukstu ziņojumi ir atspējoti.

Noklusējums: 60
 Diapazons: 0-3600

HostInterface

Norādiet hosta saskarni (IP adresi vai DNS nosaukumu), kas tiek izmantota hosta automātiskās reģistrācijas 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 izsniegs kļūdu un netiks palaists, ja vērtība pārsniegs 255 rakstzīmju ierobežojumu.

Diapazons: 0-255 rakstzīmes

HostInterfaceItem

Norādiet vienumu, lai noteiktu hosts saskarni (IP adresi vai DNS nosaukumu) hosts automātiskās reģistrācijas 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 pārsniegs 255 rakstzīmju ierobežojumu.

Vienums `system.run[]` tiek atbalstīts neatkarīgi no `AllowKey`, `AllowKeyRegexp`, `DenyKey` vai `DenyKeyRegexp` iestatījumiem.

HostMetadata

Norādiet metadatus, lai hosts tiktu identificēts vai atšķirts hostu automātiskās reģistrācijas laikā (aktīvais aģents). `HostMetadata` ļauj atšķirt hostus papildus resursdatora nosaukumam.

Ja nav definēts, vērtība tiks iegūta no `HostMetadataItem`.

Aģents parādīs kļūdu un netiks palaists, ja norādītā vērtība pārsniegs 2034 baitus vai nebūs UTF-8 virkne. Ja parametrs sagaida IP adresi vai DNS nosaukumu, arī derīgas UTF-8 vērtības, kas nav derīgas IP adreses vai DNS nosaukumi, tiks noraidītas un par tām tiks ziņots kā par nederīgām.

Vairākrindu metadati netiek atbalstīti — izvade tiks saīsināta pie pirmās rindiņas pārtraukuma.

Diapazons: 0–2034 baiti

HostMetadataItem

Norādiet Zabbix aģenta vienumu, 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 papildus pēc resursdatora nosaukuma.

Tiek atbalstīti lietotāja parametri un aizstājvārdi. `system.run[]` vienums tiek atbalstīts neatkarīgi no `AllowKey`, `AllowKeyRegexp`, `DenyKey` vai `DenyKeyRegexp` iestatījumiem.

Automātiskās reģistrācijas pieprasījuma laikā aģents reģistrēs brīdinājuma ziņojumu, ja norādītā vienuma atgrieztā vērtība pārsniedz 65535 UTF-8 koda punktu ierobežojumu. Vienuma atgrieztajai vērtībai ir jābūt UTF-8 virknei; pretējā gadījumā tā tiks ignorēta. Ja parametrs sagaida IP adresi vai DNS nosaukumu, tiks noraidītas un kā nederīgas paziņotas arī vērtības, kas ir derīgas UTF-8 virknes, bet nav derīgas IP adreses vai DNS nosaukumi.

Vairākrindu metadati netiek atbalstīti — izvade tiks saīsināta pie pirmās rindiņas pārtraukuma.

Hostname

Norādiet unikālus, reģistrjutīgus hostu nosaukumus. Ar komatiem atdalīts saraksts.
 Obligāts aktīvajām pārbaudēm, un tam jāsakrīt ar hostu nosaukumiem, kā tie konfigurēti serverī.
 Ja nav definēts, vērtība tiek iegūta no `HostnameItem`.
Atļautās rakstzīmes: burtciparu rakstzīmes, '.', ' ', '_' un '-'. Maksimālais garums: 128 rakstzīmes katram hosta nosaukumam, 2048 rakstzīmes visai rindai.

Noklusējums: iestatīts ar `HostnameItem`

HostnameItem

Norādiet Zabbix aģenta vienumu resursdatora nosaukuma iegūšanai.
 Šī opcija tiek izmantota tikai tad, ja Hostname nav definēts.
 Lietotāja parametri vai aizstājvārdi netiek atbalstīti, taču vienums `system.run[]` tiek atbalstīts neatkarīgi no `AllowKey`, `AllowKeyRegexp`, `DenyKey` vai `DenyKeyRegexp` vērtībām.

Noklusējums: `system.hostname`

Iekļaut

Norādiet direktoriju vai atsevišķus failus, kas jāiekļauj konfigurācijas failā.
 Zabbix instalēšanas laikā šis parametrs tiek iestatīts uz `/usr/local/etc`, ja vien to nemaināt kompilēšanas laikā.

Vispārīgie noteikumi parametram `Include`:

- Ceļš var būt relatīvs attiecībā pret `zabbix_agentd.conf` faila atrašanās vietu.
- Vienā konfigurācijas failā var norādīt vairākus `Include` parametrus.
- Zabbix apstrādā `Include` parametru uzreiz pēc tā nolasīšanas, tāpēc ieteicams to novietot konfigurācijas faila beigās.

Iekļauto failu noteikumi:

- Katram failam jābūt lasāmam.
- Katram failam jābūt ar pareizu sintaksi. Pretējā gadījumā komponents netiks palaists.

Direktoriju vai failu modeļu iekļaušanas noteikumi:

- Nepieņemiet, ka iekļaušana notiek noteiktā secībā (piemēram, faili netiek iekļauti alfabētiskā secībā).
- Nedefinējiet vienu un to pašu parametru vairāk nekā vienā iekļautajā failā (piemēram, lai pārrakstītu vispārīgu iestatījumu ar specifisku), jo nevar paredzēt, kura vērtība tiks izmantota.
- Uzmanieties no rezerves kopijām, ko daži teksta redaktori izveido automātiski (piemēram, rediģējot `include/my_specific.conf`, var tikt izveidots `include/my_specific.conf.BAK`), jo tiks iekļauti abi faili.

Lai no direktorija iekļautu tikai atbilstošos failus, izmantojiet aizstājējzīmi (*) modeļa atbilstībai.

Piemērs:

```
Include=/absolute/path/to/config/files/*.conf
```

ListenBacklog

Iestatiet maksimālo TCP rindā gaidošo savienojumu skaitu.
 Noklusējuma vērtība ir cieti kodēta konstante, kas ir atkarīga no sistēmas.
 Atbalstītā maksimālā vērtība arī ir atkarīga no sistēmas; pārāk lielas vērtības var tikt klusi saīsinātas līdz ”implementation-specified maximum”.

Noklusējums: SOMAXCONN
 Diapazons: 0 - INT_MAX

ListenIP

Norādiet aģenta klausīšanās IP adresi savienojumiem no servera/starpniekservera. Komatatdalīts saraksts.

Noklusējums: 0.0.0.0

ListenPort

Norādiet aģenta klausīšanās portu savienojumiem no servera/starpniekservera.

Noklusējums: 10050
 Diapazons: 1024-32767

LoadModule

Norādiet moduli, kas jāielādē aģenta startēšanas laikā.
 Moduļi tiek izmantoti, lai paplašinātu aģenta funkcionalitāti.
 Modulim jāatrodas direktorijā, kas norādīta ar LoadModulePath, vai arī ceļam ir jābūt norādītam pirms moduļa nosaukuma. Ja norādītais ceļš ir absolūts (sākas ar /), tad LoadModulePath tiek ignorēts.
 Formāti: - LoadModule=<module.so> - LoadModule=<path/module.so> - LoadModule=</abs_path/module.so>

Ir atļauts norādīt vairākus LoadModule parametrus.

LoadModulePath

Norādiet pilnu ceļu uz aģenta moduļu atrašanās vietu.
 Noklusējums ir atkarīgs no kompilēšanas opcijām.

LogFile

Norādiet aģenta žurnāla failu.

Obligāts: Jā, ja LogType ir iestatīts uz file; citādi nē

LogFileSize

Iestatiet maksimālo žurnāla faila lielumu (MB).
0 - atspējot automātisku žurnāla failu rotāciju.
Piezīme: Ja tiek sasniegts žurnāla faila lieluma ierobežojums un failu rotācija neizdodas jebkāda iemesla dēļ, esošais žurnāla fails 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, izmantojot HostMetadataItem, HostInterfaceItem vai HostnameItem parametrus.

Noklusējums: 0
 Vērtības: 0 - atspējots, 1 - iespējots

LogType

Norādiet žurnāla izvades tipu: - *file* - ierakstīt žurnālu failā, kas norādīts ar LogFile parametru; - *system* - ierakstīt žurnālu syslog; - *console* - ierakstīt žurnālu standarta izvades plūsmā.

Noklusējums: file

MaxLinesPerSecond

Iestatiet maksimālo jauno rindu skaitu sekundē, ko nosūtīt uz Zabbix serveri/starpniekserveri, apstrādājot `log` vai `logrt` aktīvās pārbaudes.
 Norādītā vērtība tiks pārrakstīta ar `maxlines` parametru, kas norādīts `log` vai `logrt` vienuma atslēgā.
Piezīme: Zabbix apstrādās 10 reizes vairāk jaunu rindu nekā iestatīts `MaxLinesPerSecond`, lai meklētu nepieciešamo virkni žurnāla vienumos.

Noklusējums: 20
 Diapazons: 1-1000

PidFile

Norādiet PID failu.

Noklusējums: `/tmp/zabbix_agentd.pid`

RefreshActiveChecks

Norādiet aktīvo pārbaužu saraksta atsvaidzināšanas biežumu sekundēs.
 Ņemiet vērā, ka pēc aktīvo pārbaužu atsvaidzināšanas neveiksmes nākamais atsvaidzināšanas mēģinājums tiks veikts pēc 60 sekundēm.

Noklusējums: 5
 Diapazons: 1-86400

Serveris

Norādiet Zabbix serveru/starpniekserveru IP adreses, pēc izvēles CIDR notācijā, vai DNS nosaukumus. Komatatdalīts saraksts.
 Ienākošie savienojumi tiks pieņemti tikai no šeit uzskaitītajiem hosts. Ja ir iespējota IPv6 atbalsta funkcija, tad `127.0.0.1`, `::127.0.0.1`, `::ffff:127.0.0.1` tiek uzskatīti par ekvivalentiem, un `::/0` ļaus jebkuru IPv4 vai IPv6 adresi. `0.0.0.0/0` var izmantot, lai atļautu jebkuru IPv4 adresi. Ņemiet vērā, ka "IPv4-compatible IPv6 addresses" (`0000::/96 prefix`) tiek atbalstītas, bet ir novecojušas saskaņā ar RFC4291. 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

Norādiet Zabbix servera/starpniekservera adresi vai klastera konfigurāciju, no kuras saņemt aktīvās pārbaudes.
 Servera/starpniekservera adrese ir IP adrese vai DNS nosaukums, un izvēles ports, kas atdalīts ar kolu.
Klastera konfigurācija ir viena vai vairāku servera vai starpniekservera grupas dalībnieku adrešu kopa, 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 šim starpniekserverim atbilstošo Zabbix serveri/klasteri norādīt nevajadzētu.
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 adreses jāiekļauj kvadrātiekavās, ja šim hostam ir norādīts ports. Ja ports nav norādīts, IPv6 adrešu kvadrātiekavas ir izvēles.
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

Norādiet avota IP adresi šādiem gadījumiem:

- izejošajiem savienojumiem ar Zabbix serveri vai Zabbix starpniekserveri;
- savienojumu izveidei, izpildot dažus vienumus (`web.page.get`, `net.tcp.port` u.c.).

StartAgents

Iestata iepriekš atzaroto `zabbix_agentd` instanču skaitu pasīvo pārbaužu apstrādei.
 Ja iestatīts uz 0, pasīvās pārbaudes tiek atspējotas un aģents neklausīsies nevienā TCP portā.

Noklusējums: 10
 Diapazons: 0-100

Noildze

Norādiet, cik ilgi jāgaida (sekundēs), lai izveidotu savienojumu un apmainītos ar datiem ar Zabbix starpniekserveri vai serveri.

Šis parametrs nosaka dažādu saziņas darbību ilgumu, tostarp:

- gaidot atbildi no Zabbix serveris;
- sūtot pieprasījumus uz Zabbix serveris, tostarp vienuma konfigurācijas pieprasījumus un vienuma datu pieprasījumus aktīvajās pārbaudēs;
- izgūstot žurnāla datus, izmantojot logfile;
- sūtot sirdspukstu ziņojumus;
- maksimālais ilgums `vfs.*` pārbaudēm;
- tiek izmantots Zabbix aģents moduļos;
- kalpo kā rezerves variants situācijās, kad serveris vai starpniekserveris, kas ir vecāks par 7.0 versiju, nosūta pārbaudes bez noildzēm.

Šī noildze **netiks** izmantota tām aģents pārbaudēm, kurām lietotāja saskarnē ir konfigurējami noildzes iestatījumi (globālā, starpniekservera vai atsevišķa vienuma līmenī).

Noklusējums: 3
 Diapazons: 1-30

TLSAccept

Norādiet šifrēšanas līmeni, kas nepieciešams ienākošajiem savienojumiem.
 Tiek izmantots 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); citādi nē

TLSCAFile

Norādiet pilnu ceļu uz failu, kurā ir augstākā līmeņa CA sertifikāti vienaudžu sertifikātu verifikācijai, kas tiek izmantota šifrētai saziņai starp Zabbix komponentiem.

TLSCertFile

Norādiet pilnu ceļu uz failu, kurā atrodas aģenta sertifikāts vai sertifikātu ķēde, kas tiek izmantota šifrētai saziņai ar Zabbix komponentiem.

TLSCipherAll

Norādiet GnuTLS prioritātes virkni vai OpenSSL (TLS 1.2) šifru virkni. Pārrakstiet noklusējuma šifru kopas atlases kritērijus sertifikātu un PSK balstītai šifrēšanai.

Piemērs:

```
TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSCipherAll13

Norādiet šifrēšanas virkni OpenSSL 1.1.1 vai jaunākai versijai TLS 1.3 režīmā. Pārrakstiet noklusējuma šifru kopas atlases kritērijus sertifikātu un PSK balstītai šifrēšanai.

GnuTLS piemērs:

```
NONE:+VERS-TLS1.2:+ECDHE-RSA:+RSA:+ECDHE-PSK:+PSK:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL
```

OpenSSL piemērs:

```
EECDH+aRSA+AES128:RSA+aRSA+AES128:kECDHEPSK+AES128:kPSK+AES128
```

TLSCipherCert

Norādiet GnuTLS prioritātes virkni vai OpenSSL (TLS 1.2) šifru virkni. Pārraksta noklusējuma šifru kopas atlases kritērijus sertifikātu balstītai šifrēšanai.

GnuTLS piemērs:

```
NONE:+VERS-TLS1.2:+ECDHE-RSA:+RSA:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL:+COMP-NULL:+SIG
```

OpenSSL piemērs:

```
EECDH+aRSA+AES128:RSA+aRSA+AES128
```

TLSCipherCert13

Norādiet šifru virkni OpenSSL 1.1.1 vai jaunākai versijai TLS 1.3 režīmā. Pārrakstiet noklusējuma šifru kopas atlases kritērijus sertifikātā balstītai šifrēšanai.

TLSCipherPSK

Norādiet GnuTLS prioritātes virkni vai OpenSSL (TLS 1.2) šifru virkni. Pārrakstiet noklusējuma šifru kopas atlases kritērijus PSK balstītai šifrēšanai.

GnuTLS piemērs:

```
NONE:+VERS-TLS1.2:+ECDHE-PSK:+PSK:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL:+COMP-NULL:+SIG
```

OpenSSL piemērs:

```
kECDHEPSK+AES128:kPSK+AES128
```

TLSCipherPSK13

Norādiet šifru virkni OpenSSL 1.1.1 vai jaunākai versijai TLS 1.3 režīmā. Pārrakstiet noklusējuma šifru komplekta atlases kritērijus PSK balstītai šifrēšanai.

Piemērs:

```
TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSConnect

Norādiet šifrēšanas līmeni, kas nepieciešams aģenta savienojumiem ar Zabbix serveri vai starpniekserveri.
 Tiek izmantots aktīvajām pārbaudēm.
 Var norādīt tikai vienu vērtību: - *unencrypted* - savienoties bez šifrēšanas (noklusējums); - *psk* - savienoties, izmantojot TLS un iepriekš koplietotu atslēgu (PSK); - *cert* - savienoties, izmantojot TLS un sertifikātu.

Obligāti: jā, ja ir definēti TLS sertifikāta vai PSK parametri (pat *unencrypted* savienojumam); citos gadījumos nē

TLSCRLFile

Norādiet pilnu ceļu uz failu, kurā ir atsauktie sertifikāti. Šis parametrs tiek izmantots šifrētai saziņai starp Zabbix komponentiem.

TLSKeyFile

Norādiet pilnu ceļu uz failu, kurā atrodas aģenta privātā atslēga, kas tiek izmantota šifrētai saziņai starp Zabbix komponentiem.

TLSPSKFile

Norādiet pilnu faila ceļu uz failu, kurā atrodas aģenta iepriekš koplietotā atslēga, kas tiek izmantota šifrētai saziņai ar Zabbix serveri vai starpniekserveri.

TLSPSKIdentity

Norādiet iepriekš koplietotās atslēgas identitātes virkni, ko izmanto šifrētai saziņai ar Zabbix serveri vai starpniekserveri.

TLSServerCertIssuer

Norādiet atļauto servera (starpniekservera) sertifikāta izdevēju.

TLSServerCertSubject

Norādiet atļauto servera (starpniekservera) sertifikāta subjektu.

UnsafeUserParameters

Atļaut visus simbolus nodot argumentiem lietotāja definētajos parametros. Šādi simboli nav atļauti: \ ' " ` * ? [ ] { } ~ $! & ; ( ) < > | # @ %
Turklāt rindiņas pārtraukuma simboli nav atļauti.

Noklusējums: 0
 Vērtības: 0 - neatļaut, 1 - atļaut

Lietotājs

Samazina privilēģijas līdz konkrētam, sistēmā jau esošam lietotājam.
Stājas spēkā tikai tad, ja tiek palaists kā `root` un `AllowRoot` ir atspējots.

Noklusējums: `zabbix`

UserParameter

Norādiet lietotāja definētu parametru, ko uzraudzīt. Var būt vairāki lietotāja definēti parametri.
Formāts: `UserParameter=<key>,<shel command>`
Ņemiet vērā, ka čaulas komandai 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

Norādiet noklusējuma meklēšanas ceļu `UserParameter` komandām. Ja tas tiek izmantots, aģents pirms komandas izpildes mainīs savu darba direktoriju uz šeit norādīto. Tādējādi `UserParameter` komandām var būt relatīvs `./` prefikss pilna ceļa vietā.
Atļauts tikai viens ieraksts.

Piemērs:

```
UserParameterDir=/opt/myscripts
```

Skatīt arī

1. Zabbix aģenta konfigurācijas atšķirības aktīvajām un pasīvajām pārbaudēm, начиная no versijas 2.0.0

Konfigurācijas parametri: Zabbix aģents (Windows)

Pārskats

Konfigurācijas parametri ļauj pielāgot Zabbix aģenta darbību operētājsistēmā Windows.

Norādiet parametru vērtības Zabbix aģenta konfigurācijas failā (`zabbix_agentd.conf`). Visi parametri ir **neobligāti**, ja vien nav skaidri norādīts, ka parametrs ir obligāts.

Šī lapa ir strukturēta šādi:

- Parametru pārskats (lai skatītu pilnu informāciju par parametru, noklikšķiniet uz tā nosaukuma).
- Parametru detalizēts apraksts.

Parametrs	Apraksts
Alias	Iestatīt vienuma atslēgas aizstājvārdu.
AllowKey	Atļaut to vienumu atslēgu izpildi, kas atbilst paraugam.
AllowKeyRegexp	Atļaut to vienumu atslēgu izpildi, kas atbilst uz regulāro izteiksmju balstītam paraugam.
BufferSend	Iestatīt vērtību nosūtīšanas biežumu no bufera uz Zabbix serveri/starpniekserveri.
BufferSize	Iestatīt maksimālo vērtību skaitu atmiņas buferī.
DebugLevel	Iestatīt atkļūdošanas līmeni.
DenyKey	Aizliegt to vienumu atslēgu izpildi, kas atbilst paraugam.
DenyKeyRegexp	Aizliegt to vienumu atslēgu izpildi, kas atbilst uz regulāro izteiksmju balstītam paraugam.
EnableRemoteCommands	Atļaut attālās komandas no Zabbix servera. Šis parametrs ir novecojis; tā vietā izmantojiet `AllowKey`, `AllowKeyRegexp`, `DenyKey` vai `DenyKeyRegexp`.
HeartbeatFrequency	Iestatīt sirdspukstu ziņojumu biežumu sekundēs.
HostInterface	Norādīt hosts saskarni.
HostInterfaceItem	Norādīt vienumu hosts saskarnes iegūšanai.
HostMetadata	Norādīt hosts metadatus.
HostMetadataItem	Norādīt Zabbix aģenta vienumu hosts metadatu iegūšanai.
Hostname	Norādīt resursdatora nosaukumu.
HostnameItem	Norādīt Zabbix aģenta vienumu resursdatora nosaukuma iegūšanai.
Include	Norādīt direktoriju vai atsevišķus failus, kas jāiekļauj konfigurācijas failā.
ListenBacklog	Iestatīt maksimālo gaidošo savienojumu skaitu TCP rindā.
ListenIP	Norādīt aģenta klausīšanās IP adresi savienojumiem no servera/starpniekservera.
ListenPort	Norādīt aģenta klausīšanās portu.
LogFile	Norādīt aģenta žurnālfailu.
LogFileSize	Iestatīt maksimālo žurnālfaila lielumu.
LogRemoteCommands	Iespējot izpildīto čaulas komandu reģistrēšanu kā brīdinājumus.
LogType	Norādīt žurnāla izvades tipu.
MaxLinesPerSecond	Iestatīt maksimālo jauno rindu skaitu sekundē, kas tiek nosūtīts Zabbix serverim/starpniekserverim, apstrādājot `log`, `logrt` vai `eventlog` aktīvās pārbaudes.
PerfCounter	Definēt jaunu parametru <parameter_name>, kas ir sistēmas veiktspējas skaitītāja <perf_counter_path> vidējā vērtība norādītajā laika periodā <period> (sekundēs).
PerfCounterEn	Definēt jaunu parametru <parameter_name>, kas ir sistēmas veiktspējas skaitītāja <perf_counter_path> vidējā vērtība norādītajā laika periodā <period> (sekundēs). Atšķirībā no PerfCounter veiktspējas skaitītāju ceļiem jābūt angļu valodā.
RefreshActiveChecks	Norādīt aktīvo pārbaužu saraksta atsvaidzināšanas biežumu.

Parametrs	Apraksts
Server	Norādīt Zabbix serveru/starpniekserveru IP adreses vai DNS nosaukumus.
ServerActive	Norādīt Zabbix servera/starpniekservera adresi vai klastera konfigurāciju, no kuras iegūt aktīvās pārbaudes.
SourceIP	Norādīt avota IP adresi.
StartAgents	Iestatīt iepriekš izveidoto `zabbix_agentd` instanču skaitu pasīvo pārbaužu apstrādei.
Timeout	Norādīt, cik ilgi (sekundēs) gaidīt savienojuma izveidi un datu apmaiņu ar Zabbix serveri vai starpniekserveri.
TLSAccept	Norādīt ienākošajiem savienojumiem nepieciešamo šifrēšanas līmeni.
TLSCAFile	Norādīt pilnu ceļu uz failu, kurā ir augstākā līmeņa CA sertifikāti partnera sertifikāta pārbaudei; tos izmanto šifrētai saziņai starp Zabbix komponentēm.
TLSCertFile	Norādīt pilnu ceļu uz failu, kurā ir aģenta sertifikāts vai sertifikātu ķēde; tos izmanto šifrētai saziņai starp Zabbix komponentēm.
TLSConnect	Norādīt savienojumiem ar Zabbix serveri vai starpniekserveri nepieciešamo šifrēšanas līmeni.
TLSCRLFile	Norādīt pilnu ceļu uz failu, kurā ir atsauktie sertifikāti. Šo parametru izmanto šifrētai saziņai starp Zabbix komponentēm.
TLSKeyFile	Norādīt pilnu ceļu uz failu, kurā ir aģenta privātā atslēga; to izmanto šifrētai saziņai starp Zabbix komponentēm.
TLSPSKFile	Norādīt pilnu ceļu uz failu, kurā ir aģenta iepriekš koplietotā atslēga; to izmanto šifrētai saziņai ar Zabbix serveri vai starpniekserveri.
TLSPSKIdentity	Norādīt iepriekš koplietotās atslēgas identitātes virkni; to izmanto šifrētai saziņai ar Zabbix serveri vai starpniekserveri.
TLSServerCertIssuer	Norādīt atļauto servera (starpniekservera) sertifikāta izdevēju.
TLSServerCertSubject	Norādīt atļauto servera (starpniekservera) sertifikāta subjektu.
UnsafeUserParameters	Atļaut lietotāja definētu parametru argumentos nodot visas rakstzīmes.
UserParameter	Norādīt uzraugāmu lietotāja definētu parametru.
UserParameterDir	Norādīt UserParameter komandu noklusējuma meklēšanas ceļu.

Ņemiet vērā:

- Noklusējuma vērtības atspoguļo dēmona noklusējuma iestatījumus, nevis vērtības piegādātajos konfigurācijas failos.
- Vērtības atbalsta vides mainīgos.
- Zabbix atbalsta tikai UTF-8 kodējumā veidotus konfigurācijas failus bez BOM.
- Komentāri, kas sākas ar #, tiek atbalstīti tikai rindas sākumā.

Parametra informācija

Aizstājvārds

Iestatiet vienuma atslēgas aizstājvārdu. To var izmantot, lai garu un sarežģītu vienuma atslēgu aizstātu ar īsāku un vienkāršāku.
 Var būt norādīti vairāki `Alias` parametri. Vairāki parametri ar vienādu `Alias` atslēgu nav atļauti.
 Dažādas `Alias` atslēgas var atsaukties uz vienu un to pašu vienuma atslēgu.
 Aizstājvārdus var izmantot parametrā `HostMetadataItem`, bet ne parametros `HostnameItem` vai `PerfCounter`.

1. piemērs. Peidžfaila lietojuma procentuāla izgūšana no servera.

```
Alias=pg_usage:perf_counter[\Paging File(_Total)\% Usage]
```

Tagad datu izgūšanai var izmantot saīsināto atslēgu `pg_usage`.

2. piemērs. CPU noslodzes 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 noslodzi ar noklusējuma parametriem, kā arī izmantot `cpu.load[percpu,avg15]`, lai iegūtu konkrētus datus par CPU noslodzi.

3. piemērs. Vairāku zema līmeņa atklāšanas noteikumu palaišana, apstrādājot 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

Atļauj izpildīt tos vienuma atslēgu nosaukumus, kas atbilst paraugam. Atslēgas paraugs ir aizstājējzīmju izteiksme, kas atbalsta * rakstzīmi, lai atbilstu jebkuram rakstzīmju skaitam.
 Vairākus atslēgu atbilstības noteikumus var definēt kopā ar DenyKey, AllowKeyRegexp un DenyKeyRegexp. Parametri tiek apstrādāti pa vienam atbilstoši to parādīšanās secībai. Secība nosaka, kuram no atbilstošajiem noteikumiem būs prioritāte. Skatiet arī: Aģenta pārbaužu ierobežošana.

AllowKeyRegexp

Atļaut izpildīt tos vienumu taustiņus, kas atbilst paraugam.
Taustiņa paraugs ir regulāra izteiksme.
 Vairākus taustiņu atbilstības noteikumus var definēt kopā ar DenyKeyRegexp, AllowKey un DenyKey. Parametri tiek apstrādāti pa vienam atbilstoši to parādīšanās secībai. Secība nosaka, kuram atbilstības noteikumam būs prioritāte. Skatiet arī: Aģenta pārbaužu ierobežošana.

Piemērs:

```
AllowKeyRegexp=^system\.run\[(?:/usr)?/bin/hostname(?: -[fsd])?\]$
```

BufferSend

Iestatiet vērtību nosūtīšanas biežumu no bufera uz Zabbix serveri/starpniekserveri (sekundēs).

Noklusējums: 5
 Diapazons: 1-3600

BufferSize

Iestata maksimālo vērtību skaitu atmiņas buferī.
 Aģents nosūtīs visus savāktos datus uz Zabbix serveri vai starpniekserveri, ja buferis būs pilns.

Noklusējums: 100
 Diapazons: 2-65535

DebugLevel

Iestatiet 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

Liedz izpildīt tos vienumu taustiņus, kas atbilst paraugam. Taustiņa paraugs ir aizstājējzīmju izteiksme, kas atbalsta rakstzīmi *, lai atbilstu jebkuram skaitam jebkādu rakstzīmju.
 Var definēt vairākus taustiņu atbilstības noteikumus kombinācijā ar AllowKey, AllowKeyRegexp un DenyKeyRegexp. Parametri tiek apstrādāti pa vienam to parādīšanās secībā. Secība nosaka, kuram atbilstības noteikumam būs prioritāte. Skatiet arī: Aģenta pārbaužu ierobežošana.

Noklusējums: DenyKey=system.run[*]

DenyKeyRegexp

Liedz izpildīt tos vienumu taustiņus, kas atbilst paraugam. Taustiņa paraugs ir regulārā izteiksme.
 Vairākus taustiņu atbilstības noteikumus var definēt kopā ar AllowKeyRegexp, AllowKey un DenyKey. Parametri tiek apstrādāti pa vienam atbilstoši to parādīšanās secībai. Secība nosaka, kuram atbilstības noteikumam būs prioritāte.
 Visi system.run[] vienumi pēc noklusējuma ir aizliegti (DenyKey=system.run[*]), ja vien nav norādīts cits system.run[*] noteikums.

Skatiet arī: Aģenta pārbaužu ierobežošana.

EnableRemoteCommands

Atļaut attālās komandas no Zabbix servera.
 Šis parametrs ir **novecojis**; tā vietā izmantojiet AllowKey=system.run[*], AllowKeyRegexp=^system\.run\[.*\]$, DenyKey=system.run[*] vai DenyKeyRegexp=^system\.run\[.*\]$.
 Tas ir iekšējs aizstājvārds parametriem AllowKey, AllowKeyRegexp, DenyKey un DenyKeyRegexp atkarībā no vērtības: - 0 - DenyKey=system.run[*] (DenyKeyRegexp=^system\.run\[.*\]$) - 1 - AllowKey=system.run[*] (AllowKeyRegexp=^system\.

Noklusējums: 0
 Vērtības: 0 - neatļaut, 1 - atļaut

HeartbeatFrequency

Iestatiet sirdspukstu ziņojumu biežumu sekundēs. Tiek izmantots aktīvo pārbaužu pieejamības uzraudzībai.
0 - sirdspukstu ziņojumi ir atspējoti.

Noklusējums: 60
 Diapazons: 0-3600

HostInterface

Norādiet hosta saskarni (IP adresi vai DNS nosaukumu), kas tiek izmantota host automātiskās reģistrācijas 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 izsniegs kļūdu un netiks palaists, ja vērtība pārsniegs 255 rakstzīmju ierobežojumu.

Diapazons: 0-255 rakstzīmes

HostInterfaceItem

Norādiet vienumu, lai noteiktu hosts saskarni (IP adresi vai DNS nosaukumu) hosta automātiskās reģistrācijas 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 pārsniegs 255 rakstzīmju ierobežojumu.

Vienums `system.run[]` tiek atbalstīts neatkarīgi no `AllowKey`, `AllowKeyRegexp`, `DenyKey` vai `DenyKeyRegexp` iestatījumiem.

HostMetadata

Norādiet metadatus, kas tiek izmantoti, lai identificētu vai atšķirtu hosts resursdatoru automātiskās reģistrācijas laikā (aktīvais aģents). `HostMetadata` ļauj atšķirt hosts resursdatorus papildus resursdatora nosaukumam.

Ja vērtība nav definēta, tā tiks iegūta no `HostMetadataItem`.

Aģents parādīs kļūdu un nesāks darbu, ja norādītā vērtība pārsniegs 2034 baitus vai nebūs UTF-8 virkne. Ja parametrs sagaida IP adresi vai DNS nosaukumu, par nederīgām tiks noraidītas un norādītas arī vērtības, kas ir derīgas UTF-8 virknes, bet nav derīgas IP adreses vai DNS nosaukumi.

Vairākrindu metadati netiek atbalstīti — izvade tiks saīsināta pie pirmās rindiņas pārtraukuma.

Diapazons: 0–2034 baiti

HostMetadataItem

Norādiet Zabbix aģenta vienumu, 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 papildus pēc resursdatora nosaukuma.

Tiek atbalstīti lietotāja parametri, veiktspējas skaitītāji un aizstājvārdi. Vienums `system.run[]` tiek atbalstīts neatkarīgi no `AllowKey`, `AllowKeyRegexp`, `DenyKey` vai `DenyKeyRegexp` iestatījumiem.

Automātiskās reģistrācijas pieprasījuma laikā aģents reģistrēs brīdinājuma ziņojumu, ja norādītā vienuma atgrieztā vērtība pārsniedz 65535 UTF-8 koda punktu ierobežojumu. Vienuma atgrieztajai vērtībai ir jābūt UTF-8 virknei; pretējā gadījumā tā tiks ignorēta. Ja parametrs sagaida IP adresi vai DNS nosaukumu, tiks noraidītas un kā nederīgas paziņotas arī vērtības, kas ir derīgas UTF-8 virknes, bet nav derīgas IP adreses vai DNS nosaukumi.

Vairākrindu metadati netiek atbalstīti — izvade tiks saīsināta pie pirmās rindiņas pārtraukuma.

Hostname

Norādiet unikālus, reģistrjutīgus hostu nosaukumus. Ar komatiem atdalīts saraksts.
 Obligāts aktīvajām pārbaudēm, un tam jāsakrīt ar hostu nosaukumiem, kā tie konfigurēti serverī. Ja nav definēts, vērtība tiek iegūta no `HostnameItem`.
Atļautās rakstzīmes: burtcipari, '.', ' ', '_' un '-'. Maksimālais garums: 128 rakstzīmes katram hosta nosaukumam, 2048 rakstzīmes visai rindai.

Noklusējums: iestatīts ar HostnameItem

HostnameItem

Norādiet Zabbix aģenta vienumu resursdatora nosaukuma iegūšanai.
 Šī opcija tiek izmantota tikai tad, ja `Hostname` nav definēts.
 Lietotāja parametri, veiktspējas skaitītāji vai aizstājvārdi netiek atbalstīti, taču vienums `system.run[]` tiek atbalstīts neatkarīgi no `AllowKey`, `AllowKeyRegexp`, `DenyKey` vai `DenyKeyRegexp` vērtībām.
Skatiet arī detalizētāku aprakstu.

Noklusējums: `system.hostname`

Include

Norādiet direktoriju vai atsevišķus failus, kas jāiekļauj konfigurācijas failā (pēc noklusējuma tas atrodas `C:\Program Files\Zabbix Agent`, ja Zabbix aģents ir instalēts, izmantojot Windows MSI instalācijas pakotnes; tas atrodas instalēšanas laikā norādītajā mapē, ja Zabbix aģents ir instalēts kā zip arhīvs).

Vispārīgie `Include` parametra noteikumi:

- Ceļš var būt relatīvs pret `zabbix_agentd.conf` faila atrašanās vietu.
- Vienā konfigurācijas failā var norādīt vairākus `Include` parametrus.
- Zabbix apstrādā `Include` parametru uzreiz pēc tā nolasīšanas, tāpēc apsveriet iespēju to ievietot konfigurācijas faila beigās.

Iekļauto failu noteikumi:

- Katram failam jābūt nolasāmam.
- Katram failam jābūt ar pareizu sintaksi. Pretējā gadījumā komponents netiks palaists.

Direktoriju vai failu šablonu iekļaušanas noteikumi:

- Neuzskatiet, ka iekļaušana notiks noteiktā secībā (piemēram, faili netiek iekļauti alfabētiskā secībā).
- Nedefinējiet vienu un to pašu parametru vairāk nekā vienā iekļautajā failā (piemēram, lai aizstātu vispārīgu iestatījumu ar konkrētu), jo nav iespējams paredzēt, kura vērtība tiks izmantota.
- Pievērsiet uzmanību rezerves kopijām, ko daži teksta redaktori izveido automātiski (piemēram, rediģējot `include/my_specific.conf` var tikt izveidots `include/my_specific.conf.BAK`), jo tiks iekļauti abi faili.

Lai no direktorijas iekļautu tikai atbilstošos failus, modeļa atbilstībai izmantojiet aizstājējzīmi (*).

Piemērs:

```
Include=C:\Program Files\Zabbix Agent\zabbix_agentd.d\*.conf
```

ListenBacklog

Iestata maksimālo TCP rindā gaidošo savienojumu skaitu.
 Noklusējuma vērtība ir cieti kodēta konstante, kas ir atkarīga no sistēmas.
 Atbalstītā maksimālā vērtība arī ir atkarīga no sistēmas; pārāk lielas vērtības var tikt klusi saīsinātas līdz ”implementation-specified maximum”.

Noklusējums: SOMAXCONN
 Diapazons: 0 - INT_MAX

ListenIP

Norādiet aģenta klausīšanās IP adresi savienojumiem no servera/starpniekservera. Komatatdalīts saraksts.

Noklusējums: `0.0.0.0`

ListenPort

Norādiet aģenta klausīšanās portu savienojumiem no servera/starpniekservera.

Noklusējums: 10050
 Diapazons: 1024-32767

LogFile

Norādiet aģenta žurnāla failu.

Noklusējums: `c:\zabbix_agentd.log`
 Obligāts: Jā, ja LogType ir iestatīts uz `file`; citādi nē

LogFileSize

Iestatiet maksimālo žurnāla faila lielumu (MB).
0 - atspējot automātisku žurnāla rotāciju.
Piezīme: Ja tiek sasniegts žurnāla faila lieluma ierobežojums un faila rotācija neizdodas jebkāda iemesla dēļ, esošais žurnāla fails 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, izmantojot `HostMetadataItem`, `HostInterfaceItem` vai `HostnameItem` parametrus.

Noklusējums: 0
 Vērtības: 0 - atspējots, 1 - iespējots

LogType

Norādiet žurnāla izvades tipu: - *file* - ierakstīt žurnālu failā, kas norādīts ar `LogFile` parametru; - *system* - ierakstīt žurnālu Windows notikumu žurnālā; - *console* - ierakstīt žurnālu standarta izvades plūsmā.

Noklusējums: `file`

MaxLinesPerSecond

Iestata maksimālo jauno rindu skaitu sekundē, ko sūtīt uz Zabbix serveri/starpniekserveri, apstrādājot `log`, `logrt` vai `eventlog` aktīvās pārbaudes.
 Norādītā vērtība tiks pārrakstīta ar `maxlines` parametru, kas norādīts `log`, `logrt` vai `eventlog` vienuma atslēgā.
Piezīme: Zabbix apstrādās 10 reizes vairāk jauno rindu nekā iestatīts `MaxLinesPerSecond`, lai meklētu nepieciešamo virkni žurnāla vienumos.

Noklusējums: 20
 Diapazons: 1-1000

PerfCounter

Definējiet jaunu parametru <parameter_name>, kas ir sistēmas veiktspējas skaitītāja <perf_counter_path> vidējā vērtība norādītajam laika periodam <period> (sekundēs).
Sintakse: <parameter_name>,"<perf_counter_path>",<period>

Piemēram, ja vēlaties saņemt vidējo procesora pārtraukumu skaitu sekundē pēdējās minūtes laikā, varat definēt jaunu parametru "interrupts" šādi:

```
PerfCounter = interrupts,"\Processor(0)\Interrupts/sec",60
```

Lūdzu, ņemiet vērā dubultpēdiņas ap veiktspējas skaitītāja ceļu. Parametra nosaukums (interrupts) ir 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

Definējiet jaunu parametru <parameter_name>, kas ir sistēmas veiktspējas skaitītāja <perf_counter_path> vidējā vērtība norādītajam laika periodam <period> (sekundēs). Salīdzinājumā 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 pēdējās minūtes laikā, varat definēt jaunu parametru "interrupts" šādi:

```
PerfCounterEn = interrupts,"\Processor(0)\Interrupts/sec",60
```

Lūdzu, ņemiet vērā dubultpēdiņas ap veiktspējas skaitītāja ceļu. Parametra nosaukums (interrupts) ir jāizmanto kā vienuma atslēga, veidojot vienumu. Paraugi vidējās vērtības aprēķināšanai tiks ņemti katru sekundi.
Angļu virkņu sarakstu var atrast, skatot šādu reģistra atslēgu: `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Perflib\009`.

RefreshActiveChecks

Norādiet aktīvo pārbaužu saraksta atsvaidzināšanas biežumu sekundēs.
 Ņemiet vērā, ka pēc aktīvo pārbaužu atsvaidzināšanas neveiksmes nākamais mēģinājums tiks veikts pēc 60 sekundēm.

Noklusējums: 5
 Diapazons: 1-86400

Serveris

Norādiet Zabbix serveru/starpniekserveru IP adreses, pēc izvēles CIDR notācijā, vai DNS nosaukumus. Komatatdalīts saraksts.
 Ienākošie savienojumi tiks pieņemti tikai no šeit norādītajiem hosts. Ja ir iespējota IPv6 atbalsts, tad `127.0.0.1`, `::127.0.0.1`, `::ffff:127.0.0.1` tiek uzskatīti par ekvivalentiem, 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" (`0000::/96 prefix`) tiek atbalstītas, bet ir novecojušas saskaņā ar RFC4291. 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 nepārprotami iestatīts uz 0

ServerActive

Norādiet Zabbix servera/starpniekservera adresi vai klastera konfigurāciju, no kuras saņemt aktīvās pārbaudes.
 Servera/starpniekservera adrese ir IP adrese vai DNS nosaukums, un izvēles 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 šim starpniekserverim atbilstošo Zabbix serveri/klasteri norādīt nedrīkst.
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 adreses jāiekļauj kvadrātiekavās, ja šim hosts ir norādīts ports. Ja ports nav norādīts, IPv6 adrešu kvadrātiekavas ir izvēles.
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: aktīvo serveru skaitam, kas norādīti `ServerActive`, plus pasīvo pārbaužu iepriekš atzaroto instanču skaitam, kas norādīts `StartAgents`, jābūt mazākam par 64.

SourceIP

Norādiet avota IP adresi šādiem gadījumiem:

- izejošajiem savienojumiem ar Zabbix serveri vai Zabbix starpniekserveri;
- savienojumu izveidei, izpildot dažus vienumus (`web.page.get`, `net.tcp.port` u.c.).

StartAgents

Iestata iepriekš sazaroto `zabbix_agentd` instanču skaitu pasīvo pārbaužu apstrādei.
 Ja iestatīts uz 0, pasīvās pārbaudes ir atspējotas, un aģents neklausīsies nevienā TCP portā.

Noklusējums: 10
 Diapazons: 0-100 (*)

Noildze

Norādiet, cik ilgi jāgaida (sekundēs), lai izveidotu savienojumu un apmainītos ar datiem ar Zabbix starpniekserveri vai serveri.

Šis parametrs nosaka dažādu saziņas darbību ilgumu, tostarp:

- gaidot atbildi no Zabbix servera;
- nosūtot pieprasījumus Zabbix serverim, tostarp vienuma konfigurācijas pieprasījumus un vienuma datus aktīvajās pārbaudēs;
- izgūstot žurnāla datus, izmantojot logfile vai Windows notikumu žurnāla uzraudzību;
- nosūtot sirdspukstu ziņojumus;
- maksimālais ilgums `vfs.*` pārbaudēm;
- ko izmanto Zabbix aģenta moduļi;
- kalpojot kā rezerves variants situācijās, kad serveris vai starpniekserveris, kas ir vecāks par 7.0 versiju, 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

Norādiet ienākošajiem savienojumiem nepieciešamo šifrēšanas līmeni.
 Tiek izmantots 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); citos gadījumos nē

TLSCAFile

Norādiet pilnu ceļu uz failu, kurā ir augstākā līmeņa CA sertifikāti vienaudžu sertifikātu verifikācijai, kas tiek izmantota šifrētai saziņai starp Zabbix komponentiem.

TLSCertFile

Norādiet pilnu ceļu uz failu, kurā atrodas aģenta sertifikāts vai sertifikātu ķēde, kas tiek izmantota šifrētai saziņai ar Zabbix komponentēm.

TLSConnect

Norādiet šifrēšanas līmeni, kas nepieciešams aģenta savienojumiem ar Zabbix serveri vai starpniekserveri.
 Tiek izmantots aktīvajām pārbaudēm.
 Var norādīt tikai vienu vērtību: - *unencrypted* - savienoties bez šifrēšanas (noklusējums); - *psk* - savienoties, izmantojot TLS un iepriekš koplietotu atslēgu (PSK); - *cert* - savienoties, izmantojot TLS un sertifikātu.

Obligāti: jā, ja ir definēti TLS sertifikāta vai PSK parametri (pat *unencrypted* savienojumam); citos gadījumos nē

TLSCRLFile

Norādiet pilnu ceļu uz failu, kurā ir atsauktie sertifikāti. Šis parametrs tiek izmantots šifrētai saziņai starp Zabbix komponentēm.

TLSKeyFile

Norādiet pilnu ceļu uz failu, kurā atrodas aģenta privātā atslēga, kas tiek izmantota šifrētai saziņai starp Zabbix komponentēm.

TLSPSKFile

Norādiet pilnu faila ceļu uz failu, kurā ir aģenta iepriekš koplietotā atslēga, kas tiek izmantota šifrētai saziņai ar Zabbix serveri vai starpniekserveri.

TLSPSKIdentity

Norādiet iepriekš koplietotās atslēgas identitātes virkni, ko izmanto šifrētai saziņai ar Zabbix serveri vai starpniekserveri.

TLSServerCertIssuer

Norādiet atļauto servera (starpniekservera) sertifikāta izdevēju.

TLSServerCertSubject

Norādiet atļauto servera (starpniekservera) sertifikāta subjektu.

UnsafeUserParameters

Atļaut visus simbolus nodot argumentiem lietotāja definētajos parametros. Šādi simboli nav atļauti: \ ' " ` * ? [ ] { } ~ $! & ; ( ) < > | # @ %
Turklāt rindu pārtraukuma simboli nav atļauti.

Noklusējums: 0
 Vērtības: 0 - neatļaut, 1 - atļaut

UserParameter

Norādiet lietotāja definētu parametru, ko uzraudzīt. Var būt vairāki lietotāja definēti parametri.
Formāts: `UserParameter=<key>,<shell command>`
Ņemiet vērā, ka čaulas komandai 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

Norādiet noklusējuma meklēšanas ceļu `UserParameter` komandām. Ja šis iestatījums tiek izmantots, aģents pirms komandas izpildes mainīs savu darba direktoriju uz šeit norādīto. Tādējādi `UserParameter` komandām var izmantot relatīvu `./` prefiksu pilna ceļa vietā. Atļauts tikai viens ieraksts.

Piemērs:

```
UserParameterDir=C:\opt\myscripts
```

Zabbix aģents operētājsistēmā Microsoft Windows

Aģenta konfigurēšana

Abas Zabbix aģentu paaudzes darbojas kā Windows pakalpojums. Zabbix aģentam 2 tālāk sniegtajos norādījumos aizstājiet `agentd` ar `agent2`.

Microsoft Windows hostā varat palaist vienu Zabbix aģenta instanci vai vairākas aģenta instances. Viena instance var izmantot:

- noklusējuma konfigurācijas failu, kas atrodas tajā pašā direktorijā, kur aģenta binārais fails;
- komandrindā norādītu konfigurācijas failu.

Ja tiek izmantotas vairākas instances, katrai aģenta instancei ir nepieciešams savs konfigurācijas fails (viena no instancēm var izmantot noklusējuma konfigurācijas failu).

Piemēra konfigurācijas fails ir pieejams Zabbix avota arhīvā:

- `conf/zabbix_agentd.conf` Zabbix aģentam;
- `conf/zabbix_agent2.conf` Zabbix agent2.

Ja vēlaties instalēt Zabbix aģentu/aģentu 2 operētājsistēmai Windows kā pakalpojumu no arhīva, skaidri nenorādot konfigurācijas failu, pirms aģenta instalēšanas:

- `conf/zabbix_agentd.conf` manuāli jānokopē uz direktoriju, kurā tiks instalēts `zabbix_agentd.exe`;
- `conf/zabbix_agent2.conf` un direktorija `conf/zabbix_agent2.d` manuāli jānokopē uz direktoriju, kurā tiks instalēts `zabbix_agent2.exe`.

Detalizētu informāciju par Zabbix Windows aģenta konfigurēšanu skatiet Zabbix aģenta konfigurācijas faila vai Zabbix aģenta 2 konfigurācijas faila opcijās.

Hostname parametrs

Lai veiktu aktīvās pārbaudes hosts, Zabbix aģentam ir jābūt definētam hostname. Turklāt aģenta pusē iestatītajai hostname vērtībai precīzi jāatbilst lietotāja saskarnē hosts konfigurētajam *Host name*.

Hostname vērtību aģenta pusē var definēt ar parametru `Hostname` vai `HostnameItem` aģenta konfigurācijas failā - ja neviens no šiem parametriem nav norādīts, tiek izmantotas noklusējuma vērtības.

Parametra `HostnameItem` noklusējuma vērtība ir aģenta atslēgas `system.hostname` atgrieztā vērtība. Operētājsistēmā Windows tā atgriež funkcijas `gethostname()` rezultātu, kas vaicā nosaukumvietu nodrošinātājiem, lai noteiktu lokālo hosts nosaukumu. Ja neviens nosaukumvietas nodrošinātājs neatbild, tiek atgriezts NetBIOS nosaukums.

`Hostname` noklusējuma vērtība ir parametra `HostnameItem` atgrieztā vērtība. Tādējādi, ja abi šie parametri nav norādīti, faktiskā hostname būs hosts NetBIOS nosaukums; Zabbix aģents izmantos NetBIOS hosts nosaukumu, lai no Zabbix servera izgūtu aktīvo pārbaužu sarakstu un nosūtītu tam rezultātus.

Atslēga `system.hostname` atbalsta divus neobligātus parametrus - *type* un *transform*.

Type nosaka tā nosaukuma veidu, kas jāatgriež vienumam:

- *netbios* (noklusējums) - atgriež NetBIOS hosts nosaukumu, kas ir ierobežots līdz 15 rakstzīmēm un sastāv tikai no LIELAJIEM BURTIEM;
- *host* - reģistrjutīgs; atgriež pilno, faktisko Windows hosts nosaukumu (bez domēna);
- *shorthost* - atgriež hostname daļu pirms pirmā punkta. Ja nosaukums nesatur punktu, tiek atgriezta visa virkne.
- *fqdn* - atgriež pilno kvalificēto domēna nosaukumu (bez beigu punkta).

Transform ļauj norādīt papildu hostname pārveidošanas noteikumu:

- *none* (noklusējums) - izmanto sākotnējo burtu reģistru;
- *lower* - pārveido tekstu mazajos burtos.

Tādējādi, lai vienkāršotu `zabbix_agentd.conf` faila konfigurēšanu un padarītu to vienotu, var izmantot dažādas pieejas:

1. Atstājiet parametrus `Hostname` vai `HostnameItem` nedefinētus, un Zabbix aģents izmantos NetBIOS hosts nosaukumu kā hostname.
2. Atstājiet parametru `Hostname` nedefinētu un definējiet `HostnameItem` šādi:
    - `HostnameItem=system.hostname[host]` - lai Zabbix aģents kā hostname izmantotu pilno, faktisko (reģistrjutīgo) Windows hosts nosaukumu;
    - `HostnameItem=system.hostname[shorthost,lower]` - lai Zabbix aģents izmantotu tikai hostname daļu pirms pirmā punkta, pārveidojot to mazajos burtos;
    - `HostnameItem=system.hostname[fqdn]` - lai Zabbix aģents kā hostname izmantotu pilno kvalificēto domēna nosaukumu.

Hosts nosaukums tiek izmantots arī kā daļa no Windows pakalpojuma nosaukuma, kas tiek izmantots Windows pakalpojuma instalēšanai, palaišanai, apturēšanai un atinstalēšanai. Piemēram, ja Zabbix aģenta konfigurācijas failā ir norādīts `Hostname=Windows_db_server`, aģents tiks instalēts kā Windows pakalpojums `Zabbix Agent [Windows_db_server]`. Tāpēc, lai katram Zabbix aģenta eksemplāram būtu atšķirīgs Windows pakalpojuma nosaukums, katram eksemplāram jāizmanto atšķirīgs hosts nosaukums.

Aģenta instalēšana kā Windows pakalpojumu

Pirms aģenta instalēšanas manuāli kopējiet `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, lai visi ar 64 bitu procesu darbību saistītie pārbaudes mehānismi darbotos pareizi, ir nepieciešama 64 bitu Zabbix aģenta versija.

Ja vēlaties izmantot citu konfigurācijas failu, nevis noklusējuma failu, 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 vajadzētu būt redzamam Vadības panelī.

Aģenta palaišana

Lai palaistu aģenta pakalpojumu, varat izmantot Control Panel vai to izdarīt 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 pakalpojumu, varat izmantot Vadības paneli vai to izdarīt 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
```

Aģenta Windows pakalpojuma 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 pakalpojumiem:

```
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, CPU veiktspējas metrikas dažiem CPU var nebūt pieejamas. Piemēram, ja ir 72 loģiskie CPU ar 2 NUMA mezgliem, abos mezglos jābūt pa 36 CPU katrā.

Zabbix aģents IBM AIX operētājsistēmā

Pārskats

Iepriekš kompilēti Zabbix aģenta binārie faili IBM AIX sistēmai ir pieejami lejupielādei `.tar.gz` arhīva formātā. Šajā lapā aprakstīts, kā instalēt aģentu no arhīva, palaist to manuāli vai ar System Resource Controller (SRC), palaist to sistēmas sāknēšanas laikā un noņemt to no SRC.

Aģenta instalēšana

AIX `tar` komanda neatbalsta opciju `-z`, tāpēc arhīvs ir jāatspiež, izmantojot `gunzip`. Izmantojiet `tar` opciju `-o`, lai izvilktie faili piederētu lietotājam, kurš tos izvelk, nevis arhīvā saglabātajiem skaitliskajiem lietotāja un grupas ID. Piemēram, lai ievietotu aģentu direktorijā `/opt/zabbix` (izpildiet kā root; `/opt/zabbix` vēl nedrīkst pastāvēt):

```
gunzip -c zabbix_agent-7.4.7-aix-7.2-powerpc-openssl.tar.gz | tar -xof -
mv zabbix_agent-7.4.7-aix-7.2-powerpc-openssl /opt/zabbix
```

AIX arhīvā aģenta binārais fails atrodas direktorijā `usr/sbin/`, bet konfigurācijas fails — direktorijā `etc/zabbix/`. Pirms aģenta palaišanas pēc nepieciešamības pielāgojiet `/opt/zabbix/etc/zabbix/zabbix_agentd.conf` (vismaz parametrus `Server`, `ServerActive` un `Hostname`).

Ja aģents tiek palaists kā root, tas pārslēdzas uz lietotāju zabbix (vai lietotāju, kas norādīts ar parametru User), ja vien konfigurācijas failā nav iestatīts AllowRoot=1. Šim lietotājam sistēmā ir jābūt; pretējā gadījumā aģents netiek palaists. Ja lietotājs zabbix vēl nepastāv, izveidojiet to kopā ar tāda paša nosaukuma grupu:

```
mkgroup zabbix
mkuser pgrp=zabbix login=false rlogin=false zabbix
```

Atribūti login=false un rlogin=false neļauj izmantot kontu pieteikšanās vajadzībām; aģents joprojām var darboties kā šis lietotājs.

Aģenta manuāla palaišana

Aģentu var palaist manuāli:

```
/opt/zabbix/usr/sbin/zabbix_agentd -c /opt/zabbix/etc/zabbix/zabbix_agentd.conf
```

Aģenta palaišana ar SRC

Aģenta reģistrēšana

Aģentu var reģistrēt arī System Resource Controller (SRC). Palaižot, izmantojot SRC, aģentam jādarbojas priekšplāna režīmā (-f opcija). Pretējā gadījumā SRC zaudē informāciju par aģenta procesu un nevar to apturēt. Piemēram (izpildiet kā root):

```
mkssys \
  -s zabbix_agentd \
  -p /opt/zabbix/usr/sbin/zabbix_agentd \
  -a "-c /opt/zabbix/etc/zabbix/zabbix_agentd.conf -f" \
  -u zabbix \
  -G tcpip \
  -S -n 15 -f 9 \
  -R
```

Šajā piemērā:

- -s iestata apakšsistēmas nosaukumu, kas tiek izmantots kopā ar SRC komandām;
- -p iestata pilno ceļu uz aģenta bināro failu;
- -a iestata aģenta komandrindas argumentus kā vienu virkni pēdiņās; SRC sadala šo virkni atsevišķos argumentos, kad palaiž aģentu (tiek pieņemta tikai viena -a opcija);
- -u iestata lietotāju, kura vārdā darbojas aģents; veidojot apakšsistēmu, lietotājvārds tiek pārveidots par skaitlisku lietotāja ID, tāpēc lietotājam jau ir jābūt izveidotam;
- -G pievieno aģentu tcpip apakšsistēmu grupai, tādējādi grupas komandas attiecas arī uz to (piemēram, stopsrc -g tcpip aptur arī aģentu);
- -S -n 15 -f 9 liek SRC kontrolēt aģentu, izmantojot signālus: stopsrc nosūta SIGTERM (15), lai veiktu parastu apturēšanu, un SIGKILL (9), lai veiktu piespiedu apturēšanu (stopsrc -f);
- -R liek SRC restartēt aģentu, ja tas tiek neparedzēti pārtraukts (piemēram, ja avarē kāds no tā procesiem); pēc stopsrc aģents netiek restartēts.

Tā kā lietotāja ID tiek saglabāts apakšsistēmas definīcijā, atjauniniet definīciju, ja lietotājs zabbix tiek izveidots no jauna ar citu lietotāja ID:

```
chssys -s zabbix_agentd -u zabbix
```

Aģenta palaišana un apturēšana

Aģentu var palaist, pārbaudīt un apturēt, izmantojot startsrc, lssrc un stopsrc:

```
startsrc -s zabbix_agentd
lssrc -s zabbix_agentd
stopsrc -s zabbix_agentd
```

Neizmantojiet piespiedu apturēšanu (stopsrc -f). Tā aptur tikai galveno aģenta procesu; pārējie aģenta procesi turpina darboties un saglabā atvērtu klausīšanās portu, tāpēc aģentu nevar palaist atkārtoti, kamēr tie nav manuāli pārtraukti (piemēram, izmantojot kill -9).

Aģenta palaišana sistēmas sāknēšanas laikā

Aģenta reģistrēšana SRC nepalaiž to sistēmas sāknēšanas laikā. Lai aģents tiktu automātiski palaists sāknēšanas laikā, pievienojiet ierakstu failam /etc/inittab:

```
mkitab "zabbix_agentd:2:once:/usr/bin/startsrc -s zabbix_agentd >/dev/console 2>&1"
```

Aģenta noņemšana no SRC

Lai noņemtu aģentu no sistēmas palaišanas un SRC:

```
stopsrc -s zabbix_agentd
rmitab zabbix_agentd
rmssys -s zabbix_agentd
```

Aģents 2

Pārskats Zabbix aģents 2 ir jaunas paaudzes Zabbix aģents, kas rakstīts Go valodā (ar daļu C koda, kas pārmantots no Zabbix aģents). Tas ir izstrādāts, lai:

- Samazinātu TCP savienojumu skaitu;
- Nodrošinātu uzlabotu pārbaužu paralēlismu;
- Būtu viegli paplašināms ar spraudņiem, kas nodrošina vienkāršas pārbaudes ar minimālu kodu, kā arī atbalsta sarežģītas pārbaudes, kas sastāv no ilgstoši darbojošiem skriptiem un patstāvīgas datu apkopošanas ar periodisku atskaišu sniegšanu;
- Darbotos kā Zabbix aģents aizstājējs, atbalstot visas iepriekšējās funkcijas.

Visas metrikas Zabbix aģents 2 tiek apkopotas ar spraudņu palīdzību.

Šīs funkcijas ir specifiskas spraudņiem:

- Plānoti un elastīgi intervāli gan pasīvajām, gan aktīvajām pārbaudēm
- Uzdevumu rindas pārvaldība, ņemot vērā grafiku un uzdevumu paralēlismu
- Spraudņa līmeņa noildzes
- Zabbix aģents 2 un tā spraudņu saderības pārbaude startēšanas laikā

Pastāv divu veidu spraudņi:

- Iebūvētie spraudņi
- Ielādējamie spraudņi

Jūs varat arī izveidot savus spraudņus.

Pašizveidotie spraudņi nav obligāti jāiebūvē aģents 2. Jūs varat tos pievienot kā ielādējamos spraudņus, tādējādi vienkāršojot spraudņu izveidi jaunām metrikām.

Ielādējamus spraudņus var pievienot izpildes laikā, nepārkompilējot Zabbix aģents 2.

Skatiet arī:

- Aģents un aģents 2 salīdzinājums
- Konfigurācijas parametri: Zabbix aģents 2
- Konfigurācijas parametri: Zabbix aģents 2 (Windows)
- Konfigurācijas parametri: Spraudņi
- Zabbix aģents 2 spraudņu protokols

Pasīvās un aktīvās pārbaudes

Zabbix aģents 2 atbalsta pasīvās un aktīvās pārbaudes, tāpat kā Zabbix aģents. Turklāt 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 ieplāno pirmo datu apkopošanu aktīvajām pārbaudēm nosacīti nejaušā laikā vienuma atjaunināšanas intervāla ietvaros, lai novērstu resursu lietojuma strauju pieaugumu. Lai aktīvās pārbaudes, kurām nav *Plānošanas* atjaunināšanas intervāla, tiktu veiktas nekavējoties pēc aģenta restartēšanas, konfigurācijas failā iestatiet parametru `ForceActiveChecksOnStart` (globālā līmeņa) vai `Plugins.<Plugin name>.System.ForceActiveChecksOnStart` spraudņa parametru (ietekmē tikai konkrēta spraudņa pārbaudes). Ja spraudņa līmeņa parametrs ir iestatīts, tas ignorēs globālo parametru.

Pārbaudes vienlaicīgums

Pārbaudes no dažādiem spraudņiem var tikt izpildītas vienlaicīgi. Vienlaicīgo pārbaužu skaits vienā spraudnī ir ierobežots ar spraudņa kapacitātes iestatījumu. Katram spraudnim var būt iebūvēts kapacitātes iestatījums (1000 ir noklusējums), ko var samazināt, izmantojot `Plugins.<PluginName>.System.Capacity=N` iestatījumu `Plugins` konfigurācijas parametrā.

Atbalstītās platformas Par atbalstītajām platformām skatiet lapu Prasības.

Aģents 2 Unix līdzīgās sistēmās Zabbix aģents 2 Unix līdzīgās sistēmās tiek palaists uz uzraudzītā hosta.

Instalēšana

Zabbix aģentu 2 var instalēt uz Linux bāzētām sistēmām, izmantojot vienu no šīm metodēm:

- Zabbix pakotnes - atlasiet Agent 2 komponentu (pēc tam, kad esat izvēlējies savu Zabbix versiju, OS distribūciju un OS versiju) un sekojiet norādījumiem;
- Zabbix avoti - lejupielādējiet avota failus un kompilējiet aģentu, konfigurējot ar `--enable-agent2` opciju.

Note:
Zabbix aģenta 2 uzraudzības iespējas var paplašināt ar ielādējamiem spraudņiem, kas ir pieejami atsevišķi. Sīkāku informāciju skatiet sadaļā Ielādējamie spraudņi.

Ja instalēts kā pakotne

Zabbix aģents 2 darbojas kā priekšplāna process un paļaujas uz ārēju pakalpojumu pārvaldnieku (piem., systemd) fona izpildei. Zabbix aģentam 2 nav iebūvēta dēmona režīma 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
```

Startēt manuāli

Zabbix aģentu var palaist, 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 palaist arī kā Windows pakalpojumu.

Instalēšana

Zabbix aģentu 2 var instalēt operētājsistēmā Windows, izmantojot vienu no šīm metodēm:

- Iepriekš kompilēti Zabbix aģenta binārie faili - lejupielādējiet aģenta MSI instalētāja pakotni un izpildiet norādījumus lapā Windows aģenta instalēšana no MSI;
- Zabbix avota faili - lejupielādējiet avota failus un izpildiet norādījumus lapā Zabbix aģenta 2 kompilēšana operētājsistēmā Windows.

Note:
Zabbix aģenta 2 uzraudzības iespējas var paplašināt ar ielādējamiem spraudņiem, kas ir pieejami atsevišķi. Sīkāku informāciju skatiet sadaļā Ielādējamie spraudņi.

Papildu informāciju par Zabbix aģenta 2 instalēšanu (no ZIP arhīva) kā Windows pakalpojumu skatiet lapā Zabbix aģents Microsoft Windows vidē.

Opcijas Ar Zabbix agent 2 var izmantot šādus komandrindas parametrus:

Parametrs	Apraksts
UNIX un Windows aģents	
`-c --config <config-file>`	Konfigurācijas faila ceļš. Šo opciju var izmantot, lai norādītu konfigurācijas failu, kas nav noklusējuma fails.
`-f --foreground`	Palaist Zabbix aģentu priekšplānā (noklusējums: true).
`-p --print`	Izvadīt zināmos vienumus un iziet. Lai tiktu atgriezti arī lietotāja parametra rezultāti, jānorāda konfigurācijas fails (ja tas neatrodas noklusējuma atrašanās vietā).
`-t --test <item key>`	Pārbaudīt norādīto vienumu un iziet. Lai tiktu atgriezti arī lietotāja parametra rezultāti, jānorāda konfigurācijas fails (ja tas neatrodas noklusējuma atrašanās vietā).
`-T --test-config`	Validēt konfigurācijas failu un iziet.

Parametrs	Apraksts
`-h --help`	Izvadīt palīdzības informāciju un iziet.
`-v --verbose`	Izvadīt atkļūdošanas informāciju. Izmantojiet šo opciju kopā ar opcijām `-p` un `-t`.
`-V --version`	Izvadīt aģenta versijas un licences informāciju.
`-R --runtime-control <option>`	Veikt administratīvās funkcijas. Skatiet sadaļu izpildlaika vadība.
Tikai Windows aģents	
`-m --multiple-agents`	Izmantot vairākas aģenta instances (ar opcijām `-i`, `-d`, `-s`, `-x`). Lai atšķirtu instanču pakalpojumu nosaukumus, katra pakalpojuma nosaukumā tiks iekļauta vērtība Hostname no norādītā konfigurācijas faila.
`-S --startup-type <value>`	Iestatīt Zabbix Windows aģenta pakalpojuma palaišanas tipu. Atļautās vērtības: `automatic` - *(noklusējums)* automātiski palaist pakalpojumu, startējot Windows; `delayed` - atlikt pakalpojuma palaišanu, līdz automātiski palaistie pakalpojumi ir pabeiguši palaišanu; `manual` - palaist pakalpojumu manuāli (lietotājam vai lietojumprogrammai); `disabled` - atspējot pakalpojumu, lai lietotājs vai lietojumprogramma to nevarētu palaist. Šo opciju var izmantot kopā ar opciju `-i` vai atsevišķi, lai mainītu jau instalēta pakalpojuma palaišanas tipu.
`-i --install`	Instalēt Zabbix Windows aģentu kā pakalpojumu.
`-d --uninstall`	Atinstalēt Zabbix Windows aģenta pakalpojumu.
`-s --start`	Palaist Zabbix Windows aģenta pakalpojumu.
`-x --stop`	Apturēt Zabbix Windows aģenta pakalpojumu.

Konkrēti komandrindas parametru izmantošanas piemēri:

- Izvadīt visus iebūvētos aģenta vienumus ar vērtībām.
- Pārbaudīt lietotāja parametru ar atslēgu `mysql.ping`, kas definēts norādītajā konfigurācijas failā.
- Instalēt Zabbix Agent pakalpojumu operētājsistēmai Windows, izmantojot noklusējuma ceļu uz konfigurācijas failu.
- Mainīt instalēta Zabbix Agent pakalpojuma palaišanas tipu operētājsistēmai Windows, izmantojot konfigurācijas failu, kas atrodas tajā pašā mapē, kur aģenta izpildāmais fails.

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

Darbības laika vadība

Darbības laika vadība nodrošina vairākas attālinātās vadības iespējas.

Option	Description
`log_level_increase`	Palielināt žurnāla detalizācijas līmeni.
`log_level_decrease`	Samazināt žurnāla detalizācijas līmeni.
`metrics`	Uzskaitīt pieejamos rādītājus.
`periodic_prof_enable`	Iespējot periodisku profilēšanu. Visi rādītāji apkopo datus attiecīgajā brīdī un tos izvada, izņemot CPU datus. Periodiskie CPU dati laika gaitā tiek apkopoti slēptā pagaidu failā. Kad intervāls beidzas, fails tiek publicēts kā CPU profils, un tiek saglabāts konfigurētais pabeigto failu skaits. Pagaidu fails netiek ieskaitīts `ProfilerMaxFilesPerProfile`.
`periodic_prof_disable`	Atspējot periodisku profilēšanu.
`periodic_prof_execute`	Pēc 5 sekunžu parauga izveidot profilus. Ja periodiskā profilēšana ir iespējota, šī komanda pabeidz pašreizējo CPU intervālu, izveido atsevišķu 5 sekunžu CPU profilu un sāk jaunu periodisko CPU failu. Komandas pabeigšanas laikā skaidri pieprasītie profili tiek rotēti atbilstoši `ProfilerMaxFilesPerProfile`. Kamēr šī komanda apkopo profilus, visas profilētāja darbības laika komandas tiek noraidītas ar ziņojumu "profiler is busy".
`periodic_prof_set_interval`	Iestatīt profilēšanas intervālu.
`version`	Parādīt aģenta versiju.
`userparameter_reload`	Atkārtoti ielādēt `UserParameter` un `Include` opciju vērtības no pašreizējā konfigurācijas faila.

Option	Description
`help`	Parādīt palīdzības informāciju par darbības laika vadību.

Piemēri:

- Palielināt aģenta 2 žurnāla detalizācijas līmeni.
- Izdrukāt darbības laika vadības opcijas.

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

Konfigurācijas fails Konfigurācijas faila noklusējuma ceļš ir atkarīgs no operētājsistēmas un Zabbix instalēšanas metodes:

- UNIX sistēmā noklusējuma ceļš ir:
  - Instalējot no pirmkoda: `/usr/local/etc/zabbix_agent2.conf` vai ceļš, kas iestatīts ar kompilēšanas laikā norādītajiem mainīgajiem `--sysconfdir` vai `--prefix`.
  - Instalējot no pakotnēm: `/etc/zabbix/zabbix_agent2.conf`.
- Windows sistēmā noklusējuma ceļš ir `C:\Program Files\Zabbix Agent 2\zabbix_agent2.conf`.

Ja hosts darbojas viena aģenta instance, tā var izmantot noklusējuma konfigurācijas failu vai konfigurācijas failu, kas norādīts komandrindā. Ja darbojas vairākas instances, katrai aģenta instancei ir nepieciešams savs konfigurācijas fails (viena no instancēm var izmantot noklusējuma konfigurācijas failu).

Zabbix aģenta 2 konfigurācijas parametri lielākoties atbilst Zabbix aģenta parametriem, izņemot dažus parametrus:

Parameters	Description
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 Zabbix aģentā 2 konfigurētu pastāvīgo krātuvi aktīvajiem vienumiem.
`ForceActiveChecksOnStart`	Nosaka, vai aģentam aktīvās pārbaudes jāveic uzreiz pēc restartēšanas vai arī tās vienmērīgi jāsadala laika gaitā.
`Plugins`	Spraudņiem var būt savi parametri formātā `Plugins.<Plugin name>.<Parameter>=<value>`. Izplatīts spraudņa parametrs ir `System.Capacity`, kas nosaka vienlaikus izpildāmo pārbaužu skaita ierobežojumu.
`StatusPort`	Ports, kurā Zabbix aģents 2 klausīsies HTTP statusa pieprasījumus un rādīs konfigurēto spraudņu un dažu iekšējo parametru sarakstu.

Parameters	Description
Zabbix aģen-tam speci-fiski konfig-urāci-jas parametri, kas netiek atbal-stīti Zab-bix aģentā 2	
`AllowRoot`, `User`	Zabbix aģents 2 tos neatbalsta, jo neatbalsta dēmona režīmu.
`EnableRemoteCommands`	Novecojis; abos aģentos tā vietā tiek izmantoti parametri `AllowKey`, `AllowKeyRegexp`, `DenyKey` un `DenyKeyRegexp`.
`ListenBacklog`	Zabbix aģentā tiek izmantots, lai definētu maksimālo gaidošo savienojumu skaitu TCP rindā. Zabbix aģents 2 to neatbalsta.
`LoadModule`, `LoadModulePath`	Zabbix aģents 2 neatbalsta ielādējamus moduļus.
`MaxLinesPerSecond`, `LogRemoteCommands`	Zabbix aģentā 2 tā vietā tiek izmantoti `Plugins.Log.MaxLinesPerSecond` un `Plugins.SystemRun.LogRemoteCommands`.
`StartAgents`	Zabbix aģentā šis parametrs tiek izmantots, 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 Agent 2 ir atspējotas, ja parametrs `Server` nav norādīts.

Plašāku informāciju skatiet konfigurācijas faila opcijās Zabbix aģentam 2.

Iziešanas kodi Zabbix aģentu 2 var arī kompilēt ar vecākām OpenSSL versijām (1.0.1, 1.0.2).

Šajā gadījumā Zabbix nodrošina mutexus bloķēšanai OpenSSL. Ja mutex bloķēšana vai atbloķēšana neizdodas, kļūdas ziņojums tiek izdrukāts standarta kļūdu straumē (STDERR), un Agent 2 iziet ar atgriešanās kodu 2 vai 3 attiecīgi.

Agenta un agent 2 salīdzinājums

Šajā sadaļā ir aprakstītas atšķirības starp Zabbix aģentu un Zabbix agent 2.

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

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

Skatīt arī:

- *Zabbix procesu apraksts*: Zabbix aģents, Zabbix agent 2
- *Konfigurācijas parametri*: Zabbix aģents UNIX / Windows, Zabbix agent 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 spraudņu repozitoriju ar papildu informāciju.

Spraudņa nosaukums	Apraksts	Atbalstītie vienumu atslēgas	Komentāri
Agent	Izmantotā Zabbix aģents metrikas.	agent.hostname, agent.ping, agent.version	Atbalstītajām atslēgām ir tādi paši parametri kā Zabbix aģents atslēgām.

Sprauduņa nosaukums	Apraksts	Atbalstītie vienumu atslēgas	Komentāri
CPU	Sistēmas CPU uzraudzība (CPU kodolu skaits, atklātie CPU, noslodzes procentuālā vērtība).	system.cpu.discovery, system.cpu.num, system.cpu.util	Atbalstītajām atslēgām ir tādi paši parametri kā Zabbix aģents 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 parametri
File	Failu metrikas 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 tādi paši parametri kā Zabbix aģents atslēgām.
Kernel	Kodola uzraudzība.	kernel.maxfiles, kernel.maxproc	Atbalstītajām atslēgām ir tādi paši parametri kā Zabbix aģents atslēgām.
Log	Žurnālfailu uzraudzība.	log, log.count, logrt, logrt.count	Atbalstītajām atslēgām ir tādi paši parametri kā Zabbix aģents atslēgām. Skatiet arī: Sprauduņa konfigurācijas parametri (Unix/Windows)
Memcached	Memcached serveris uzraudzība.	memcached.ping, memcached.stats	
Modbus	Nolasa Modbus datus.	modbus.get	Atbalstītajām atslēgām ir tādi paši parametri kā Zabbix aģents atslēgām.
MQTT	Saņem publicētās MQTT tēmu vērtības.	mqtt.get	Lai konfigurētu šifrētu savienojumu ar MQTT brokeri, norādiet TLS parametrus aģents konfigurācijas failā kā nosauktu sesiju 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 aģents konfigurācijas failā kā nosauktu sesiju vai noklusējuma parametrus. Pašlaik TLS parametrus nevar nodot kā vienuma atslēgas parametrus.
NetIf	Tīkla interfeisu uzraudzība.	net.if.collisions, net.if.discovery, net.if.get, net.if.in, net.if.out, net.if.total	Atbalstītajām atslēgām ir tādi paši parametri kā Zabbix aģents atslēgām.

Spraudņa nosaukums	Apraksts	Atbalstītie 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 spraudņa izmantošanas instalējiet Oracle Instant Client.
Proc	Procesa CPU noslodzes procentuālā vērtība.	proc.cpu.util	Atbalstītajai atslēgai ir tādi paši parametri kā Zabbix aģents atslēgai.
Redis	Redis serveris uzraudzība.	redis.config, redis.info, redis.ping, redis.slowlog.count	Lai konfigurētu šifrētu savienojumu ar Redis, norādiet TLS parametrus aģenta konfigurācijas failā kā nosauktu sesiju 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 startēt, 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 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 parametri

SW	Instalēto pakotņu saraksts.	system.sw.packages, system.sw.packages.get	Atbalstītajām atslēgām ir tādi paši parametri kā Zabbix aģents atslēgai.
Swap	Maiņatmiņas apjoms baitos/procentos.	system.swap.size	Atbalstītajai atslēgai ir tādi paši parametri kā Zabbix aģents atslēgai.

Sprauduņa nosaukums	Apraksts	Atbalstītie vienumu atslēgas	Komentāri
SystemRun	Izpilda norādīto komandu.	system.run	Atbalstītajai atslēgai ir tādi paši parametri kā Zabbix aģents atslēgai. Skatiet arī: Sprauduņa 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 tādi paši parametri kā Zabbix aģents atslēgai.
UDP	UDP pakalpojumu pieejamības un veiktspējas uzraudzība.	net.udp.service, net.udp.service.perf	Atbalstītajām atslēgām ir tādi paši parametri kā Zabbix aģents atslēgām.
Uname	Informācijas iegūšana par sistēmu.	system.hostname, system.sw.arch, system.uname	Atbalstītajām atslēgām ir tādi paši parametri kā Zabbix aģents atslēgām.
Uptime	Sistēmas darbspējas laika metrikas vākšana.	system.uptime	Atbalstītajai atslēgai ir tādi paši parametri kā Zabbix aģents atslēgai.
VFSDev	VFS metrikas vākšana.	vfs.dev.discovery, vfs.dev.get, vfs.dev.read, vfs.dev.write	Atbalstītajām atslēgām ir tādi paši parametri kā Zabbix aģents 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 tādi paši parametri kā Zabbix aģents atslēgām.
ZabbixAsync	Asinhrona metrikas 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 tādi paši parametri kā Zabbix aģents atslēgām.
ZabbixStats	Zabbix serveris/starpniekserveris iekšējās metrikas vai aizkavēto vienumu skaits rindā.	zabbix.stats	Atbalstītajām atslēgām ir tādi paši parametri kā Zabbix aģents atslēgām.

Spraudņa nosaukums	Apraksts	Atbalstītie vienumu atslēgas	Komentāri
ZabbixSync	Sinhrona metrikas 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 tādi paši parametri kā Zabbix aģents atslēgām.

Ielādējami spraudņi

Ielādējami spraudņi Zabbix aģentam 2 nav pieejami pēc noklusējuma, un tie jāinstalē atsevišķi:

- Linux sistēmā varat izmantot pakotnes (piemēram, instalēt Ember+ Ubuntu sistēmā ar `apt install zabbix-agent2-plugin-embe`
  vai izveidot spraudņus
- Windows sistēmā varat instalēt no MSI vai izveidot spraudņus

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

Noklikšķiniet uz spraudņa nosaukuma, lai pārietu uz spraudņa repozitoriju, kurā atrodas 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 (uz dokumentiem balstītas, izplatītas datubāzes) uzraudzība.	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 aģenta konfigurācijas failā kā nosauktas sesijas parametrus. Pašlaik TLS parametrus nevar nodot kā vienumu atslēgu 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 aģenta konfigurācijas failā kā nosauktas sesijas vai noklusējuma parametrus. Pašlaik TLS parametrus nevar nodot kā vienumu atslēgu 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 aģenta konfigurācijas failā kā nosauktas sesijas vai noklusējuma parametrus. Pašlaik TLS parametrus nevar nodot kā vienumu atslēgu parametrus. Skatiet arī PostgreSQL sprauduņa konfigurācijas parametrus.

Note:
Ielādējami sprauduņi, palaižot ar:
 - `-V --version` - izdrukā sprauduņa versiju un licences informāciju;
 - `-h --help` - izdrukā palīdzības informāciju.
 - `-t, --test <item key>` - palaiž spraudni testēšanai (sprauduņa konfigurācija tiek ignorēta).

Konfigurācijas parametri: Zabbix aģents 2

Pārskats

Konfigurācijas parametri ļauj pielāgot Zabbix aģenta 2 darbību.

Zabbix aģents 2 ir jaunas paaudzes Zabbix aģents, un to var izmantot Zabbix aģenta vietā.

Norādiet parametru vērtības Zabbix aģenta 2 konfigurācijas failā (`zabbix_agent2.conf`). Visi parametri ir **neobligāti**, ja vien nav skaidri norādīts, ka parametrs ir obligāts.

Šī lapa ir strukturēta šādi:

- Parametru pārskats (lai skatītu pilnu informāciju par parametru, noklikšķiniet uz tā nosaukuma)
- Parametru detalizēts apraksts

Parametrs	Apraksts
Alias	Iestatīt aizstājvārdu vienuma atslēgai.
AllowKey	Atļaut to vienumu atslēgu izpildi, kas atbilst paraugam.
AllowKeyRegexp	Atļaut to vienumu atslēgu izpildi, kas atbilst uz regulāro izteiksmi balstītam paraugam.
BufferSend	Iestatīt vērtību nosūtīšanas biežumu no bufera uz Zabbix serveri/starpniekserveri.
BufferSize	Iestatīt maksimālo vērtību skaitu atmiņas buferī.
ControlSocket	Norādīt vadības ligzdu izpildlaika komandu nosūtīšanai ar opciju '-R'.
DebugLevel	Iestatīt atkļūdošanas līmeni.
DenyKey	Aizliegt to vienumu atslēgu izpildi, kas atbilst paraugam.
DenyKeyRegexp	Aizliegt to vienumu atslēgu izpildi, kas atbilst uz regulāro izteiksmi balstītam paraugam.
EnablePersistentBuffer	Iespējot lokālās pastāvīgās krātuves izmantošanu aktīvajiem vienumiem.
EnableProfiler	Iespējot periodisku Go profilēšanu aģenta palaišanas laikā.

Parametrs	Apraksts
ForceActiveChecksOnStart	Veikt aktīvās pārbaudes tūlīt pēc restartēšanas, saņemot pirmo konfigurāciju.
HeartbeatFrequency	Iestatīt sirdspukstu ziņojumu biežumu sekundēs.
HostInterface	Norādīt hosts saskarni.
HostInterfaceItem	Norādīt vienumu hosts saskarnes iegūšanai.
HostMetadata	Norādīt hosts metadatus.
HostMetadataItem	Norādīt Zabbix aģenta vienumu hosts metadatu iegūšanai.
Hostname	Norādīt hosts nosaukumu.
HostnameItem	Norādīt Zabbix aģenta vienumu hosts nosaukuma iegūšanai.
Include	Norādīt direktoriju vai atsevišķus failus, kas jāiekļauj konfigurācijas failā.
ListenIP	Norādīt aģenta IP adresi savienojumu pieņemšanai no servera/starpniekservera.
ListenPort	Norādīt aģenta klausīšanās portu.
LogFile	Norādīt aģenta žurnālfailu.
LogFileSize	Iestatīt maksimālo žurnālfaila lielumu.
LogType	Norādīt žurnāla izvades tipu.
PersistentBufferFile	Norādīt failu SQLite datubāzes glabāšanai.
PersistentBufferPeriod	Iestatīt datu glabāšanas periodu, kad nav savienojuma ar Zabbix serveri vai starpniekserveri.
PidFile	Norādīt PID failu.
Plugins.<PluginName>.System.Capacity	Iestatīt vienlaikus izpildāmo pārbaužu ierobežojumu katram spraudnim.
Plugins.Log.MaxLinesPerSecond	Iestatīt maksimālo jauno rindu skaitu sekundē, kas tiek nosūtīts Zabbix serverim/starpniekserverim, apstrādājot `log` vai `logrt` aktīvās pārbaudes.
Plugins.SystemRun.LogRemoteCommands	Ieslēgt/izslēgt tālvadības čaulas komandu reģistrēšanu kā brīdinājumus.
PluginSocket	Norādīt ceļu uz UNIX ligzdu ielādējamu spraudņu saziņai.
PluginTimeout	Iestatīt savienojumu taimautu ar ielādējamiem spraudņiem.
ProfilerDir	Direktorija, kurā tiks ierakstīti profilēšanas dati.
ProfilerMaxFilesPerProfile	Maksimālais saglabājamo failu skaits katram profila tipam.
ProfilerInterval	Profilēšanas datu ierakstīšanas intervāls.
RefreshActiveChecks	Norādīt aktīvo pārbaužu saraksta atsvaidzināšanas biežumu.
Server	Norādīt Zabbix serveru/starpniekserveru IP adreses vai DNS nosaukumus.
ServerActive	Norādīt Zabbix servera/starpniekservera adresi vai klastera konfigurāciju, no kuras iegūt aktīvās pārbaudes.
SourceIP	Norādīt avota IP adresi.
StatusPort	Norādīt portu HTTP statusa pieprasījumu noklausīšanai (`http://localhost:<port>/status`).
Timeout	Norādīt, cik ilgi (sekundēs) gaidīt savienojuma izveidi un datu apmaiņu ar Zabbix serveri vai starpniekserveri.
TLSAccept	Norādīt ienākošajiem savienojumiem nepieciešamo šifrēšanas līmeni.
TLSCAFile	Norādīt pilnu ceļu uz failu, kurā ir augstākā līmeņa CA sertifikāti partnera sertifikāta pārbaudei, ko izmanto šifrētai saziņai starp Zabbix komponentēm.
TLSCertFile	Norādīt pilnu ceļu uz failu, kurā ir aģenta sertifikāts vai sertifikātu ķēde, ko izmanto šifrētai saziņai starp Zabbix komponentēm.
TLSCipherAll	Norādīt OpenSSL šifru virkni (TLS 1.2). Ignorēt noklusējuma šifru komplekta atlases kritērijus sertifikātu un PSK balstītai šifrēšanai.
TLSCipherAll13	Norādīt OpenSSL šifru virkni (TLS 1.3) operētājsistēmai OpenSSL 1.1.1 vai jaunākai versijai. Ignorēt noklusējuma šifru komplekta atlases kritērijus sertifikātu un PSK balstītai šifrēšanai.
TLSCipherCert	Norādīt OpenSSL šifru virkni (TLS 1.2). Ignorēt noklusējuma šifru komplekta atlases kritērijus sertifikātu balstītai šifrēšanai.
TLSCipherCert13	Norādīt OpenSSL šifru virkni (TLS 1.3) operētājsistēmai OpenSSL 1.1.1 vai jaunākai versijai. Ignorēt noklusējuma šifru komplekta atlases kritērijus sertifikātu balstītai šifrēšanai.
TLSCipherPSK	Norādīt OpenSSL šifru virkni (TLS 1.2). Ignorēt noklusējuma šifru komplekta atlases kritērijus PSK balstītai šifrēšanai.
TLSCipherPSK13	Norādīt OpenSSL šifru virkni (TLS 1.3) operētājsistēmai OpenSSL 1.1.1 vai jaunākai versijai. Ignorēt noklusējuma šifru komplekta atlases kritērijus PSK balstītai šifrēšanai.
TLSConnect	Norādīt savienojumiem ar Zabbix serveri vai starpniekserveri nepieciešamo šifrēšanas līmeni.
TLSCRLFile	Norādīt pilnu ceļu uz failu, kurā ir atsaukto sertifikātu saraksts. Šo parametru izmanto šifrētai saziņai starp Zabbix komponentēm.
TLSKeyFile	Norādīt pilnu ceļu uz failu, kurā ir aģenta privātā atslēga, ko izmanto šifrētai saziņai starp Zabbix komponentēm.
TLSPSKFile	Norādīt pilnu ceļu uz failu, kurā ir aģenta iepriekš koplietotā atslēga, ko izmanto šifrētai saziņai ar Zabbix serveri vai starpniekserveri.
TLSPSKIdentity	Norādīt iepriekš koplietotās atslēgas identitātes virkni, ko izmanto šifrētai saziņai ar Zabbix serveri vai starpniekserveri.
TLSServerCertIssuer	Norādīt atļauto servera (starpniekservera) sertifikāta izdevēju.

Parametrs	Apraksts
TLSServerCertSubject	Norādīt atļauto servera (starpniekservera) sertifikāta subjektu.
UnsafeUserParameters	Atļaut visus simbolus nodot lietotāja definētu parametru argumentos.
UserParameter	Norādīt uzraugāmu lietotāja definētu parametru.
UserParameterDir	Norādīt UserParameter komandu noklusējuma meklēšanas ceļu.

Ņemiet vērā:

- Noklusējuma vērtības atspoguļo procesa noklusējuma iestatījumus, nevis vērtības piegādātajos konfigurācijas failos.
- Vērtības atbalsta vides mainīgos.
- Zabbix atbalsta tikai UTF-8 kodējumā veidotus konfigurācijas failus bez BOM.
- Komentāri, kas sākas ar #, tiek atbalstīti tikai rindas sākumā.

Parametra informācija

Alias

Iestatiet vienuma atslēgas aizstājvārdu. To var izmantot, lai garu un sarežģītu vienuma atslēgu aizstātu ar īsāku un vienkāršāku.
 Var būt norādīti vairāki `Alias` parametri. Vairāki parametri ar vienādu `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 izgūšana.

```
Alias=zabbix.userid:vfs.file.regexp[/etc/passwd,"^zabbix:.:([0-9]+)",,,,\1]
```

Tagad datu izgūšanai var izmantot saīsināto atslēgu `zabbix.userid`.

2. piemērs: CPU izmantošanas 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 izmantošanas procentuālo vērtību ar noklusējuma parametriem, kā arī izmantot `cpu.util[all, idle, avg15]`, lai iegūtu konkrētus datus par CPU izmantošanu.

3. piemērs: vairāku zema līmeņa atklāšanas noteikumu izpilde, apstrādājot 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]` u. c.

AllowKey

Atļauj izpildīt tos vienumu atslēgu, kas atbilst paraugam. Atslēgas paraugs ir aizstājējzīmju izteiksme, kas atbalsta * rakstzīmi, lai atbilstu jebkuram rakstzīmju skaitam.
Vairākas atslēgu atbilstības kārtulas var definēt kopā ar `DenyKey`. Vairākas atslēgu atbilstības kārtulas var definēt kopā ar `DenyKey`, `AllowKeyRegexp` un `DenyKeyRegexp`. Parametri tiek apstrādāti pa vienam atbilstoši to parādīšanās secībai. Secība nosaka, kurai no atbilstošajām kārtulām būs prioritāte. Skatiet arī: Restricting agent checks.

AllowKeyRegexp

Atļaut izpildīt tos vienumu taustiņus, kas atbilst paraugam.
Taustiņa paraugs ir regulāra izteiksme.
 Var definēt vairākus taustiņu atbilstības noteikumus kombinācijā ar `DenyKeyRegexp`, `AllowKey` un `DenyKey`. Parametri tiek apstrādāti pa vienam atbilstoši to parādīšanās secībai. Secība nosaka, kuram atbilstības noteikumam būs prioritāte. Skatiet arī: Aģenta pārbaužu ierobežošana.

Piemērs:

```
AllowKeyRegexp=^system\.run\[(?:/usr)?/bin/hostname(?: -[fsd])?\]$
```

BufferSend

Iestatiet vērtību nosūtīšanas biežumu no bufera uz Zabbix serveri/starpniekserveri (sekundēs).
 Ņemiet vērā, ka, ja buferis ir pilns, dati tiks nosūtīti ātrāk.

Noklusējums: 5
 Diapazons: 1-3600

BufferSize

Iestata maksimālo vērtību skaitu 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

Norādiet vadības socket, lai ar `-R` opciju sūtītu izpildlaika komandas.

Noklusējums: `/tmp/agent.sock`

DebugLevel

Iestatiet 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

Liedz izpildīt tos vienumu taustiņus, kas atbilst paraugam.
 Taustiņa paraugs ir aizstājējzīmju izteiksme, kas atbalsta rakstzīmi `*`, lai atbilstu jebkuram skaitam jebkādu rakstzīmju.
Vairākus taustiņu atbilstības noteikumus var definēt kopā ar `AllowKey`. Vairākus taustiņu atbilstības noteikumus var definēt kopā ar `AllowKey`, `AllowKeyRegexp` un `DenyKeyRegexp`. Parametri tiek apstrādāti pa vienam to parādīšanās secībā. Secība nosaka, kuram atbilstības noteikumam būs prioritāte. Skatiet arī: Aģenta pārbaužu ierobežošana.

Noklusējums: `DenyKey=system.run[*]`

DenyKeyRegexp

Liedz izpildīt tos vienumu taustiņus, kas atbilst paraugam.
Taustiņa paraugs ir regulārā izteiksme.
 Vairākus taustiņu atbilstības noteikumus var definēt kopā ar `AllowKeyRegexp`, `AllowKey` un `DenyKey`. Parametri tiek apstrādāti pa vienam atbilstoši to parādīšanās secībai. Secība nosaka, kuram atbilstības noteikumam būs prioritāte.
 Visi `system.run[]` vienumi pēc noklusējuma ir aizliegti (`DenyKey=system.run[*]`), ja vien nav norādīts cits `system.run[*]` noteikums.

Skatiet arī: Aģenta pārbaužu ierobežošana.

Piemērs:

```
DenyKeyRegexp=^system\.run\[.*(?:rm -rf|poweroff|reboot).*\]$
```

EnablePersistentBuffer

Iespējot lokālās pastāvīgās krātuves izmantošanu aktīvajiem vienumiem.
 Ja pastāvīgā krātuve ir atspējota, tiks izmantots atmiņas buferis.

Noklusējums: 0
 Vērtības: 0 - atspējots, 1 - iespējots

EnableProfiler

Iespējot periodisku Go profilēšanu aģenta startēšanas laikā. Profilēšanu var arī iespējot vai atspējot izpildlaikā, izmantojot opciju `-R`. Kamēr `periodic_prof_execute` apkopo profilus, visas profilētāja izpildlaika komandas tiek noraidītas, jo tas ir aizņemts.

Noklusējums: 0
 Vērtības: 0 - atspējots, 1 - iespējots

ForceActiveChecksOnStart

Veic aktīvās pārbaudes nekavējoties pēc restartēšanas, lai saņemtu pirmo konfigurāciju.
 Pieejams arī kā katra spraudņa konfigurācijas parametrs, piemēram: `Plugins.Uptime.System.ForceActiveChecksOnStart=1`

Noklusējums: 0
 Vērtības: 0 - atspējots, 1 - iespējots

HeartbeatFrequency

Iestata heartbeat ziņojumu biežumu sekundēs.
 Izmanto aktīvo pārbaužu pieejamības uzraudzībai.
0 - heartbeat ziņojumi ir atspējoti.

Noklusējums: 60
 Diapazons: 0-3600

HostInterface

Norādiet hosta saskarni (IP adresi vai DNS nosaukumu), kas tiek izmantota hosta autoregistrācijas laikā.
 Šī vērtība tiks izmantota, lai aizpildītu saskarni jaunizveidotajā hostā, un ļauj nepārprotami 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 netiks palaists, ja vērtība pārsniegs 255 rakstzīmju limitu.

Diapazons: 0-255 rakstzīmes

HostInterfaceItem

Norādiet vienumu, lai noteiktu hosta saskarni (IP adresi vai DNS nosaukumu) hosta automātiskās reģistrācijas 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`, `AllowKeyRegexp`, `DenyKey` vai `DenyKeyRegexp` iestatījumiem.

HostMetadata

Norādiet metadatus, kas tiek izmantoti, lai identificētu vai atšķirtu hosts resursdatoru automātiskās reģistrācijas laikā.
 `HostMetadata` ļauj atšķirt hosts resursdatorus papildus resursdatora nosaukumam.

Ja vērtība nav definēta, tā tiks iegūta no `HostMetadataItem`.

aģents parādīs kļūdu un netiks palaists, ja norādītā vērtība pārsniegs 2034 baitus vai nebūs UTF-8 virkne. Ja parametrs sagaida IP adresi vai DNS nosaukumu, arī derīgas UTF-8 vērtības, kas nav derīgas IP adreses vai DNS nosaukumi, tiks noraidītas un parādītas kā nederīgas.

Vairākrindu metadati netiek atbalstīti — izvade tiks saīsināta pie pirmās rindiņas pārtraukuma.

Diapazons: 0–2034 baiti

HostMetadataItem

Norādiet vienumu, 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 papildus pēc resursdatora nosaukuma.

Tiek atbalstīti lietotāja parametri un aizstājvārdi. Vienums `system.run[]` tiek atbalstīts neatkarīgi no `AllowKey`, `AllowKeyRegexp`, `DenyKey` vai `DenyKeyRegexp` iestatījumiem.

Automātiskās reģistrācijas pieprasījuma laikā aģents reģistrēs brīdinājuma ziņojumu, ja norādītā vienuma atgrieztā vērtība pārsniegs 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, tiks noraidītas un kā nederīgas paziņotas arī vērtības, kas ir derīgas UTF-8 virknes, bet nav derīgas IP adreses vai DNS nosaukumi.

Vairākrindu metadati netiek atbalstīti — izvade tiks saīsināta pie pirmās rindiņas pārtraukuma.

Hostname

Norādiet unikālus, reģistrjutīgus hostu nosaukumus. Ar komatiem atdalīts saraksts.
 Nepieciešams aktīvajām pārbaudēm, un tam jāsakrīt ar hostu nosaukumiem, kā tie ir konfigurēti 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: iestatīts ar HostnameItem

HostnameItem

Norādiet vienumu resursdatora nosaukuma iegūšanai.
 Šī opcija tiek izmantota tikai tad, ja `Hostname` nav definēts.
 Lietotāja parametri vai aizstājvārdi netiek atbalstīti, taču vienums `system.run[]` tiek atbalstīts neatkarīgi no `AllowKey`, `AllowKeyRegexp`, `DenyKey` vai `DenyKeyRegexp` vērtībām.

Noklusējums: `system.hostname`

Iekļaut

Norādiet direktoriju vai atsevišķus failus, kas jāiekļauj konfigurācijas failā.
 Zabbix instalēšanas laikā šis parametrs tiek iestatīts uz `/usr/local/etc`, ja vien to nemaināt kompilēšanas laikā.

Vispārīgie noteikumi parametram `Include`:

- Ceļš var būt relatīvs attiecībā pret `zabbix_agent2.conf` faila atrašanās vietu.
- Vienā konfigurācijas failā var norādīt vairākus `Include` parametrus.
- Zabbix apstrādā `Include` parametru uzreiz pēc tā nolasīšanas, tāpēc ieteicams to ievietot konfigurācijas faila beigās.

Iekļauto failu noteikumi:

- Katram failam jābūt lasāmam.
- Katram failam jābūt ar pareizu sintaksi. Pretējā gadījumā komponents netiks palaists.

Direktoriju vai failu modeļu iekļaušanas noteikumi:

- Nepieņemiet, ka iekļaušanas secība ir noteikta (piemēram, faili netiek iekļauti alfabētiskā secībā).
- Nedefinējiet vienu un to pašu parametru vairāk nekā vienā iekļautajā failā (piemēram, lai pārrakstītu vispārīgu iestatījumu ar specifisku), jo nevar paredzēt, kura vērtība tiks izmantota.
- Ņemiet vērā rezerves kopijas, ko daži teksta redaktori izveido automātiski (piemēram, rediģējot `include/my_specific.conf`, var tikt izveidots `include/my_specific.conf.BAK`), jo abi faili tiks iekļauti.

Lai no direktorija iekļautu tikai atbilstošos failus, izmantojiet aizstājējzīmi (*) modeļa atbilstībai.

Piemērs:

```
Include=/absolute/path/to/config/files/*.conf
```

ListenIP

Norādiet aģenta klausīšanās IP adresi savienojumiem no servera/starpniekservera. Komatu atdalīts saraksts.
 Pirmā IP adrese tiek nosūtīta Zabbix serverim, ja savienojums tiek izveidots ar to, lai izgūtu aktīvo pārbaužu sarakstu.

Noklusējums: `0.0.0.0`

ListenPort

Norādiet aģenta klausīšanās portu savienojumiem no servera/starpniekservera.

Noklusējums: 10050
 Diapazons: 1024-32767

LogFile

Norādiet aģenta žurnāla failu.

Noklusējums: `/tmp/zabbix_agent2.log`
 Obligāts: Jā, ja `LogType` ir iestatīts uz `file`; citādi nē

LogFileSize

Iestatiet maksimālo žurnāla faila izmēru (MB).
0 - atspējot automātisku žurnāla rotāciju.
Piezīme: Ja tiek sasniegts žurnāla faila izmēra ierobežojums un faila rotācija neizdodas jebkāda iemesla dēļ, esošais žurnāla fails tiek saīsināts un izveidots no jauna.

Noklusējums: 1
 Diapazons: 0-1024

LogType

Norādiet žurnāla izvades tipu: - *file* - rakstīt žurnālu failā, kas norādīts ar `LogFile` parametru; - *system* - rakstīt žurnālu syslog; - *console* - rakstīt žurnālu standarta izvadei.

Noklusējums: `file`

PersistentBufferFile

Norādiet failu SQLite datubāzes glabāšanai. Jānorāda pilns faila nosaukums.
 Šis parametrs tiek izmantots tikai tad, ja pastāvīgais buferis ir iespējots (`EnablePersistentBuffer=1`).

Pastāvīgā bufera periods

Iestatiet laika periodu datu glabāšanai, ja nav savienojuma ar Zabbix serveri vai starpniekserveri. Vecākie dati tiks zaudēti.
 Žurnāla dati tiks saglabāti. Šis parametrs tiek izmantots tikai tad, ja pastāvīgais buferis ir iespējots (`EnablePersistentBuffer=1`).

Noklusējums: 1h
 Diapazons: 1m-365d

PidFile

Norādiet PID failu.

Noklusējums: `/tmp/zabbix_agent2.pid`

Spraudņi.<PluginName>.System.Capacity

Iestatiet vienlaicīgi izpildāmo pārbaužu limitu katram <PluginName> spraudnim.

Noklusējums: 1000 Diapazons: 1-1000

Plugins.Log.MaxLinesPerSecond

Iestata maksimālo jauno rindu skaitu sekundē, ko nosūtīt uz Zabbix serveri/starpniekserveri, apstrādājot `log` vai `logrt` aktīvās pārbaudes.
 Norādītā vērtība tiks pārrakstīta ar `maxlines` parametru, kas norādīts `log` un `logrt` vienuma atslēgā.
Piezīme: Zabbix apstrādās 10 reizes vairāk jauno rindu nekā iestatīts `MaxLinesPerSecond`, lai loga vienumos atrastu nepieciešamo virkni.

Noklusējums: 20
 Diapazons: 1-1000

Plugins.SystemRun.LogRemoteCommands

Iespējot izpildīto čaulas komandu reģistrēšanu kā brīdinājumus.
 Komandas tiks reģistrētas tikai tad, ja tās tiek izpildītas attālināti.
 Žurnāla ieraksti netiks izveidoti, ja system.run[] tiek palaists lokāli, izmantojot HostMetadataItem, HostInterfaceItem vai HostnameItem parametrus.

Noklusējums: 0
 Vērtības: 0 - atspējots, 1 - iespējots

PluginSocket

Norādiet UNIX ligzdas ceļu ielādējamo spraudņu saziņai.

Noklusējums: /tmp/agent.plugin.sock

PluginTimeout

Norādiet savienojumu noildzi ar ielādējamiem spraudņiem sekundēs.

Noklusējums: Timeout
 Diapazons: 1-30

ProfilerDir

Direktorija, kurā tiks ierakstīti profilēšanas dati.

Noklusējums: /tmp/zabbix_agent2_profiles

ProfilerMaxFilesPerProfile

Maksimālais saglabājamo failu skaits katram profila tipam. Ierobežojums tiek piemērots pabeigtajiem failiem pēc katras profila izdrukas. Aktīvais CPU profils tiek apkopots slēptā pagaidu failā un netiek ieskaitīts ierobežojumā. Uz failiem, kas izveidoti ar periodic_prof_execute, arī attiecas šis ierobežojums.

Noklusējums: 10
 Diapazons: 1-100

ProfilerInterval

Profilēšanas datu ierakstīšanas intervāls sekundēs.

Noklusējums: 3600
 Diapazons: 1-86400

RefreshActiveChecks

Norādiet aktīvo pārbaužu saraksta atsvaidzināšanas biežumu sekundēs.
 Ņemiet vērā, ka pēc aktīvo pārbaužu atsvaidzināšanas neveiksmes nākamais atsvaidzināšanas mēģinājums tiks veikts pēc 60 sekundēm.

Noklusējums: 5
 Diapazons: 1-86400

Serveris

Norādiet Zabbix serveru/starpniekserveru IP adreses, pēc izvēles CIDR notācijā, vai DNS nosaukumus. Ar komatiem atdalīts saraksts.
 Ienākošie savienojumi tiks pieņemti tikai no šeit norādītajiem hosts. Ja ir iespējota IPv6 atbalsts, tad 127.0.0.1, ::127.0.0.1, ::ffff:127.0.0.1 tiek uzskatīti par ekvivalentiem, 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, pasīvie pārbaužu veidi tiek atspējoti, 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

Norādiet Zabbix servera/starpniekservera adresi vai klastera konfigurāciju, no kuras saņemt aktīvās pārbaudes.
 Servera/starpniekservera adrese ir IP adrese vai DNS nosaukums, kā arī izvēles ports, kas atdalīts ar kolu.
Klastera konfigurācija ir viena vai vairāku servera vai starpniekservera grupas dalībnieku adrešu kopa, kas atdalīta 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 nedrīkst norādīt vairāk nekā vienu Zabbix starpniekserveri. Ja ir norādīts Zabbix starpniekserveris, tad šim starpniekserverim atbilstošo Zabbix serveri/klasteri norādīt nedrīkst.
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 adreses jāiekļauj 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

Norādiet avota IP adresi šādiem gadījumiem:

- izejošajiem savienojumiem ar Zabbix serveri vai Zabbix starpniekserveri.
- savienojumu izveidei, izpildot dažus vienumus (`web.page.get`, `net.tcp.port` u.c.).

StatusPort

Norādiet portu HTTP statusa pieprasījumu uzklausīšanai (`http://localhost:<port>/status`).

Diapazons: 1024-32767

Noildze

Norādiet, cik ilgi jāgaida (sekundēs), lai izveidotu savienojumu un apmainītos ar datiem ar Zabbix starpniekserveri vai serveri.

Šis parametrs nosaka dažādu saziņas darbību ilgumu, tostarp:

- gaidot atbildi no Zabbix servera;
- sūtot pieprasījumus Zabbix serverim, tostarp vienuma konfigurācijas pieprasījumus un vienuma datus aktīvajās pārbaudēs;
- izgūstot žurnāla datus, izmantojot logfile;
- sūtot sirdspukstu ziņojumus;
- maksimālais ilgums `vfs.*` pārbaudēm;
- kalpojot kā rezerves variants situācijās, kad serveris vai starpniekserveris, kas ir vecāks par 7.0 versiju, sūta pārbaudes bez noildzēm.

Šī noildze **netiks** izmantota tiem aģenta pārbaudījumiem, kuriem lietotāja saskarnē ir konfigurējami noildzes iestatījumi (globālā, starpniekservera vai atsevišķa vienuma līmenī).

Noklusējums: 3
 Diapazons: 1-30

TLSAccept

Norādiet ienākošajiem savienojumiem nepieciešamo šifrēšanas līmeni.
 Tiek izmantots 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.

Obligāti: jā, ja ir definēti TLS sertifikāta vai PSK parametri (pat *unencrypted* savienojumam); citādi nē

TLSCAFile

Norādiet pilnu ceļu uz failu, kurā ir augstākā līmeņa CA sertifikāti vienaudžu sertifikātu verifikācijai, kas tiek izmantota šifrētai saziņai starp Zabbix komponentiem.

TLSCertFile

Norādiet pilnu faila ceļu uz failu, kurā atrodas aģenta sertifikāts vai sertifikātu ķēde, kas tiek izmantota šifrētai saziņai ar Zabbix komponentēm.

TLSCipherAll

Norādiet OpenSSL šifru virkni (TLS 1.2). Pārrakstiet noklusējuma šifru kopas atlases kritērijus sertifikātu un PSK balstītai šifrēšanai.

Piemērs:

```
EECDH+aRSA+AES128:RSA+aRSA+AES128:kECDHEPSK+AES128:kPSK+AES128
```

TLSCipherAll13

Norādiet OpenSSL šifru virkni (TLS 1.3) OpenSSL 1.1.1 vai jaunākai versijai. Pārrakstiet noklusējuma šifru komplektu atlases kritērijus sertifikātu un PSK balstītai šifrēšanai.

Piemērs:

```
TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSCipherCert

Norādiet OpenSSL šifru virkni (TLS 1.2). Pārrakstiet noklusējuma šifru kopas atlases kritērijus 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`) izmantojiet `TLSCipherAll` vietā.

TLSCipherCert13

Norādiet OpenSSL šifru virkni (TLS 1.3) OpenSSL 1.1.1 vai jaunākai versijai. Pārrakstiet noklusējuma šifru kopas atlases kritērijus sertifikātu balstītai šifrēšanai.

Ņemiet vērā, ka šo parametru nevar izmantot kopā ar `TLSAccept=cert,psk`; sertifikāta savienojumam (`TLSConnect=cert`) izmantojiet `TLSCipherAll13` vietā.

TLSCipherPSK

Norādiet OpenSSL šifru virkni (TLS 1.2). Pārrakstiet noklusējuma šifru kopas atlases kritērijus PSK balstītai šifrēšanai.

Piemērs:

```
kECDHEPSK+AES128:kPSK+AES128
```

TLSCipherPSK13

Norādiet OpenSSL šifru virkni (TLS 1.3) OpenSSL 1.1.1 vai jaunākai versijai. Pārrakstiet noklusējuma šifru kopas atlases kritērijus PSK balstītai šifrēšanai.

Piemērs:

```
TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSConnect

Norādiet šifrēšanas līmeni, kas nepieciešams aģenta savienojumiem ar Zabbix serveri vai starpniekserveri.
 Tiek izmantots aktīvajām pārbaudēm.
 Var norādīt tikai vienu vērtību: - *unencrypted* - savienoties bez šifrēšanas (noklusējums); - *psk* - savienoties, izmantojot TLS un iepriekš koplietotu atslēgu (PSK); - *cert* - savienoties, izmantojot TLS un sertifikātu.

Obligāti: jā, ja ir definēti TLS sertifikāta vai PSK parametri (pat *unencrypted* savienojumam); citādi nē

TLSCRLFile

Norādiet pilnu ceļu uz failu, kurā ir atsauktie sertifikāti. Šis parametrs tiek izmantots šifrētai saziņai starp Zabbix komponentiem.

TLSKeyFile

Norādiet pilnu ceļu uz failu, kurā atrodas aģenta privātā atslēga, kas tiek izmantota šifrētai saziņai starp Zabbix komponentēm.

TLSPSKFile

Norādiet pilnu faila ceļu uz failu, kurā ir aģenta iepriekš koplietotā atslēga, kas tiek izmantota šifrētai saziņai ar Zabbix serveri vai starpniekserveri.

TLSPSKIdentity

Norādiet iepriekš koplietotās atslēgas identitātes virkni, ko izmanto šifrētai saziņai ar Zabbix serveri vai starpniekserveri.

TLSServerCertIssuer

Norādiet atļauto servera (starpniekservera) sertifikāta izdevēju.

TLSServerCertSubject

Norādiet atļauto servera (starpniekservera) sertifikāta subjektu.

UnsafeUserParameters

Atļaut visus simbolus nodot argumentiem lietotāja definētajos parametros. Šādi simboli nav atļauti: \ ' " ` * ? [ ] { } ~ $! & ; ( ) < > | # @ %
Turklāt rindiņas pārtraukuma simboli nav atļauti.

Noklusējums: 0
 Vērtības: 0 - neatļaut, 1 - atļaut

UserParameter

Norādiet lietotāja definētu parametru, ko uzraudzīt. Var būt vairāki lietotāja definēti parametri.
Formāts: `UserParameter=<key>,<shell command>`
Ņemiet vērā, ka shell komandai nedrīkst atgriezt tukšu virkni vai tikai EOL. Shell komandām var būt relatīvi ceļi, ja ir norādīts parametrs `UserParameterDir`.

Piemērs:

```
UserParameter=system.test,who|wc -l
UserParameter=check_cpu,./custom_script.sh
```

UserParameterDir

Norādiet noklusējuma meklēšanas ceļu `UserParameter` komandām. Ja šis iestatījums tiek izmantots, aģents pirms komandas izpildes mainīs savu darba direktoriju uz šeit norādīto. Tādējādi `UserParameter` komandām var izmantot relatīvu `./` prefiksu pilna ceļa vietā.
Atļauts tikai viens ieraksts.

Piemērs:

```
UserParameterDir=/opt/myscripts
```

Konfigurācijas parametri: Zabbix aģents 2 (Windows)

Pārskats

Konfigurācijas parametri ļauj pielāgot Zabbix aģenta 2 darbību operētājsistēmā Windows.

Norādiet parametru vērtības Zabbix aģenta 2 konfigurācijas failā (`zabbix_agent2.conf`). Visi parametri ir **neobligāti**, ja vien nav skaidri norādīts, ka parametrs ir obligāts.

Šī lapa ir strukturēta šādi:

- Parametru pārskats (lai skatītu pilnu informāciju par parametru, noklikšķiniet uz tā nosaukuma)
- Parametru detalizēts apraksts

Parametrs	Apraksts
Alias	Iestatīt vienuma atslēgas aizstājvārdu.
AllowKey	Atļaut to vienuma atslēgu izpildi, kas atbilst paraugam.
AllowKeyRegexp	Atļaut to vienuma atslēgu izpildi, kas atbilst uz regulāro izteiksmi balstītam paraugam.
BufferSend	Iestatīt vērtību nosūtīšanas biežumu no bufera uz Zabbix serveri.
BufferSize	Iestatīt maksimālo vērtību skaitu atmiņas buferī.
ControlSocket	Norādīt vadības ligzdu, kas tiek izmantota izpildlaika komandu nosūtīšanai ar opciju `-R`.
DebugLevel	Iestatīt atkļūdošanas līmeni.
DenyKey	Aizliegt to vienuma atslēgu izpildi, kas atbilst paraugam.
DenyKeyRegexp	Aizliegt to vienuma atslēgu izpildi, kas atbilst uz regulāro izteiksmi balstītam paraugam.
EnablePersistentBuffer	Iespējot lokālās pastāvīgās krātuves izmantošanu aktīvajiem vienumiem.
EnableProfiler	Iespējot periodisku Go profilēšanu aģenta palaišanas laikā.
ForceActiveChecksOnStart	Veikt aktīvās pārbaudes uzreiz pēc restartēšanas, saņemot pirmo konfigurāciju.
HeartbeatFrequency	Iestatīt sirdspukstu ziņojumu biežumu sekundēs.
HostInterface	Norādīt hosts saskarni.
HostInterfaceItem	Norādīt vienumu hosts saskarnes iegūšanai.
HostMetadata	Norādīt hosts metadatus.
HostMetadataItem	Norādīt Zabbix aģenta vienumu hosts metadatu iegūšanai.
Hostname	Norādīt hosts nosaukumu.
HostnameItem	Norādīt Zabbix aģenta vienumu hosts nosaukuma iegūšanai.
Include	Norādīt direktoriju vai atsevišķus failus, kas jāiekļauj konfigurācijas failā.
ListenIP	Norādīt aģenta klausīšanās IP adresi savienojumiem no servera/starpniekservera.
ListenPort	Norādīt aģenta klausīšanās portu.
LogFile	Norādīt aģenta žurnālfailu.
LogFileSize	Iestatīt maksimālo žurnālfaila lielumu.
LogType	Norādīt žurnāla izvades tipu.
PerfCounter	Norādīt jaunu parametru <parameter_name>, kas ir sistēmas veiktspējas skaitītāja <perf_counter_path> vidējā vērtība norādītajā laika periodā <period> (sekundēs).
PerfCounterEn	Norādīt jaunu parametru <parameter_name>, kas ir sistēmas veiktspējas skaitītāja <perf_counter_path> vidējā vērtība norādītajā laika periodā <period> (sekundēs). Atšķirībā no PerfCounter veiktspējas skaitītāju ceļiem jābūt angļu valodā.
PersistentBufferFile	Norādīt failu, kurā Zabbix aģentam 2 jāsaglabā SQLite datubāze.
PersistentBufferPeriod	Iestatīt datu glabāšanas ilgumu, ja nav savienojuma ar serveri vai starpniekserveri.

Parametrs	Apraksts
Plugins.<PluginName>.System.Capacity	Iestatīt vienam spraudnim vienlaikus izpildāmo pārbaužu skaita ierobežojumu.
Plugins.Log.MaxLinesPerSecond	Iestatīt maksimālo jauno rindu skaitu sekundē, ko aģents nosūtīs Zabbix serverim vai starpniekserverim, apstrādājot `log`, `logrt` un `eventlog` aktīvās pārbaudes.
Plugins.SystemRun.LogRemoteCommands	Iespējot izpildīto čaulas komandu reģistrēšanu kā brīdinājumus.
PluginSocket	Norādīt ceļu uz Windows nosaukto kanālu ielādējamu spraudņu saziņai.
PluginTimeout	Iestatīt savienojumu noildzi ar ielādējamiem spraudņiem sekundēs.
ProfilerDir	Direktorija, kurā tiks ierakstīti profilēšanas dati.
ProfilerMaxFilesPerProfile	Maksimālais saglabājamo failu skaits katram profila tipam.
ProfilerInterval	Profilēšanas datu ierakstīšanas intervāls.
RefreshActiveChecks	Norādīt aktīvo pārbaužu saraksta atsvaidzināšanas biežumu.
Server	Norādīt Zabbix serveru/starpniekserveru IP adreses vai DNS nosaukumus.
ServerActive	Norādīt Zabbix servera/starpniekservera adresi vai klastera konfigurāciju, no kuras saņemt aktīvās pārbaudes.
SourceIP	Norādīt avota IP adresi.
StatusPort	Iestatīt portu, kurā aģents klausās HTTP statusa pieprasījumus (`http://localhost:<port>/status`).
Timeout	Norādīt, cik ilgi (sekundēs) gaidīt savienojuma izveidi un datu apmaiņu ar Zabbix serveri vai starpniekserveri.
TLSAccept	Norādīt ienākošajiem savienojumiem nepieciešamo šifrēšanas līmeni.
TLSCAFile	Norādīt pilnu ceļu uz failu, kurā ir augstākā līmeņa CA sertifikāti partnera sertifikāta pārbaudei; fails tiek izmantots šifrētai saziņai starp Zabbix komponentēm.
TLSCertFile	Norādīt pilnu ceļu uz failu, kurā ir aģenta sertifikāts vai sertifikātu ķēde; fails tiek izmantots šifrētai saziņai starp Zabbix komponentēm.
TLSCipherAll13	Norādīt OpenSSL šifra virkni (TLS 1.3). Ignorēt noklusējuma šifru komplekta atlases kritērijus sertifikātos un PSK balstītai šifrēšanai.
TLSCipherCert13	Norādīt OpenSSL šifra virkni (TLS 1.3). Ignorēt noklusējuma šifru komplekta atlases kritērijus sertifikātos balstītai šifrēšanai.
TLSCipherPSK13	Norādīt OpenSSL šifra virkni (TLS 1.3). Ignorēt noklusējuma šifru komplekta atlases kritērijus PSK balstītai šifrēšanai.
TLSConnect	Norādīt savienojumiem ar Zabbix serveri vai starpniekserveri nepieciešamo šifrēšanas līmeni.
TLSCRLFile	Norādīt pilnu ceļu uz failu, kurā ir atsauktie sertifikāti. Šis parametrs tiek izmantots šifrētai saziņai starp Zabbix komponentēm.
TLSKeyFile	Norādīt pilnu ceļu uz failu, kurā ir aģenta privātā atslēga; fails tiek izmantots šifrētai saziņai starp Zabbix komponentēm.
TLSPSKFile	Norādīt pilnu ceļu uz failu, kurā ir aģenta iepriekš koplietotā atslēga; fails tiek izmantots šifrētai saziņai ar Zabbix serveri vai starpniekserveri.
TLSPSKIdentity	Norādīt iepriekš koplietotās atslēgas identitātes virkni; tā tiek izmantota šifrētai saziņai ar Zabbix serveri vai starpniekserveri.
TLSServerCertIssuer	Norādīt atļauto servera (starpniekservera) sertifikāta izdevēju.
TLSServerCertSubject	Norādīt atļauto servera (starpniekservera) sertifikāta subjektu.
UnsafeUserParameters	Atļaut visus simbolus nodot lietotāja definētu parametru argumentos.
UserParameter	Norādīt uzraugāmu lietotāja definētu parametru.
UserParameterDir	Norādīt noklusējuma meklēšanas ceļu `UserParameter` komandām.

Ņemiet vērā:

- Noklusējuma vērtības atspoguļo procesa noklusējuma iestatījumus, nevis vērtības piegādātajos konfigurācijas failos;
- Vērtības atbalsta vides mainīgos;
- Zabbix atbalsta tikai konfigurācijas failus UTF-8 kodējumā bez BOM;
- Komentāri, kas sākas ar #, tiek atbalstīti tikai rindas sākumā.

Parametra informācija

Alias

Iestatiet vienuma atslēgas aizstājvārdu.
To var izmantot, lai garu un sarežģītu vienuma atslēgu aizstātu ar īsāku un vienkāršāku atslēgu.
 Var būt norādīti vairāki `Alias` parametri.
Vairāki parametri ar vienādu `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. Peidžfaila lietojuma procentuāla izgūšana no servera.

```
Alias=pg_usage:perf_counter[\Paging File(_Total)\% Usage]
```

Tagad datu izgūšanai var izmantot saīsināto atslēgu `pg_usage`.

2. piemērs. CPU noslodzes 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 noslodzi ar noklusējuma parametriem, kā arī izmantot `cpu.load[percpu,avg15]`, lai iegūtu konkrētus datus par CPU noslodzi.

3. piemērs. Vairāku zema līmeņa atklāšanas noteikumu izpilde, apstrādājot 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]` u.c.

AllowKey

Atļauj izpildīt tos vienumu atslēgu nosaukumus, kas atbilst paraugam. Atslēgas paraugs ir aizstājējzīmju izteiksme, kas atbalsta * rakstzīmi, lai atbilstu jebkuram rakstzīmju skaitam.
 Vairākus atslēgu atbilstības noteikumus var definēt kopā ar `DenyKey`, `AllowKeyRegexp` un `DenyKeyRegexp`. Parametri tiek apstrādāti pa vienam atbilstoši to parādīšanās secībai. Secība nosaka, kuram atbilstošajam noteikumam būs prioritāte. Skatiet arī: Aģenta pārbaužu ierobežošana.

AllowKeyRegexp

Atļaut izpildīt tos vienumu taustiņus, kas atbilst paraugam.
Taustiņa paraugs ir regulārā izteiksme.
 Var definēt vairākus taustiņu atbilstības noteikumus kombinācijā ar `DenyKeyRegexp`, `AllowKey` un `DenyKey`. Parametri tiek apstrādāti pa vienam atbilstoši to parādīšanās secībai. Secība nosaka, kuram atbilstības noteikumam būs prioritāte. Skatiet arī: Aģenta pārbaužu ierobežošana.

Piemērs:

```
AllowKeyRegexp=^system\.run\[(?:/usr)?/bin/hostname(?: -[fsd])?\]$
```

BufferSend

Iestatiet vērtību nosūtīšanas biežumu no bufera uz Zabbix serveri/starpniekserveri (sekundēs).
Ņemiet vērā, ka, ja buferis ir pilns, dati tiks nosūtīti ātrāk.

Noklusējums: 5
 Diapazons: 1-3600

BufferSize

Iestata maksimālo vērtību skaitu atmiņas buferī.
 Ja buferis ir pilns, aģents nosūtīs visus savāktos datus uz Zabbix serveri vai starpniekserveri.
Šo parametru drīkst izmantot tikai tad, ja pastāvīgais buferis ir atspējots (`EnablePersistentBuffer=0`).

Noklusējums: 1000
 Diapazons: 2-65535

ControlSocket

Norādiet vadības ligzdu, ko izmanto izpildlaika komandu nosūtīšanai ar `-R` opciju.

Noklusējums: `\\.\pipe\agent.sock`

DebugLevel

Iestatiet 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

Liedz izpildīt tos vienumu taustiņus, kas atbilst paraugam. Taustiņa paraugs ir aizstājējzīmju izteiksme, kas atbalsta rakstzīmi *, lai atbilstu jebkuram skaitam jebkādu rakstzīmju.
 Var definēt vairākus taustiņu atbilstības noteikumus kombinācijā ar `AllowKey`, `AllowKeyRegexp` un `DenyKeyRegexp`. Parametri tiek apstrādāti pa vienam atbilstoši to parādīšanās secībai. Secība nosaka, kuram atbilstības noteikumam būs prioritāte.

Skatiet arī: Aģenta pārbaužu ierobežošana.

Noklusējums: `DenyKey=system.run[*]`

DenyKeyRegexp

Liedz izpildīt tos vienumu taustiņus, kas atbilst paraugam. Taustiņa paraugs ir regulārā izteiksme.
 Vairākus taustiņu atbilstības noteikumus var definēt kopā ar `AllowKeyRegexp`, `AllowKey` un `DenyKey`. Parametri tiek apstrādāti pa vienam atbilstoši to parādīšanās secībai. Secība nosaka, kuram atbilstības noteikumam būs prioritāte.
 Visi `system.run[]` vienumi pēc noklusējuma ir aizliegti (`DenyKey=system.run[*]`), ja vien nav norādīts cits `system.run[*]` noteikums.

Skatiet arī: Aģenta pārbaužu ierobežošana.

EnablePersistentBuffer

Iespējot lokālās pastāvīgās krātuves izmantošanu aktīvajiem vienumiem. Ja pastāvīgā krātuve ir atspējota, tiks izmantots atmiņas buferis.

Noklusējums: 0
 Vērtības: 0 - atspējots, 1 - iespējots

EnableProfiler

Iespējot periodisku Go profilēšanu aģenta startēšanas laikā.
Profilēšanu var arī iespējot vai atspējot izpildlaikā, izmantojot opciju `-R`.
Kamēr `periodic_prof_execute` apkopo profilus, visas profilētāja izpildlaika komandas tiek noraidītas, jo tas ir aizņemts.

Noklusējums: 0
 Vērtības: 0 - atspējots, 1 - iespējots

ForceActiveChecksOnStart

Veic aktīvās pārbaudes nekavējoties pēc restartēšanas, lai saņemtu pirmo konfigurāciju. Pieejams arī kā katra spraudņa konfigurācijas parametrs, piemēram: `Plugins.Uptime.System.ForceActiveChecksOnStart=1`

Noklusējums: 0
 Vērtības: 0 - atspējots, 1 - iespējots

HeartbeatFrequency

Iestatiet sirdspukstu ziņojumu biežumu sekundēs. Tiek izmantots aktīvo pārbaužu pieejamības uzraudzībai.
0 - sirdspukstu ziņojumi ir atspējoti.

Noklusējums: 60
 Diapazons: 0-3600

HostInterface

Norādiet hosta saskarni (IP adresi vai DNS nosaukumu), kas tiek izmantota hosta automātiskās reģistrācijas 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 izsniegs kļūdu un netiks palaists, ja vērtība pārsniegs 255 rakstzīmju ierobežojumu.

Diapazons: 0-255 rakstzīmes

HostInterfaceItem

Norādiet vienumu, lai noteiktu hosts saskarni (IP adresi vai DNS nosaukumu) hosta automātiskās reģistrācijas 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`, `AllowKeyRegexp`, `DenyKey` vai `DenyKeyRegexp` iestatījumiem.

HostMetadata

Norādiet metadatus, kas tiek izmantoti, lai identificētu vai atšķirtu hosts resursdatora automātiskās reģistrācijas laikā (aktīvais aģents). `HostMetadata` ļauj atšķirt hosts, izmantojot ne tikai resursdatora nosaukumu.

Ja vērtība nav definēta, tā tiks iegūta no `HostMetadataItem`.

Aģents parādīs kļūdu un netiks palaists, ja norādītā vērtība pārsniegs 2034 baitus vai nebūs UTF-8 virkne. Ja parametrs sagaida IP adresi vai DNS nosaukumu, tiks noraidītas un kā nederīgas paziņotas arī vērtības, kas ir derīgas UTF-8 virknes, bet nav derīgas IP adreses vai DNS nosaukumi.

Vairākrindu metadati netiek atbalstīti — izvade tiks saīsināta pie pirmās rindiņas pārtraukuma.

Diapazons: 0–2034 baiti

HostMetadataItem

Norādiet Zabbix aģenta vienumu, 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 papildus resursdatora nosaukumam.

Tiek atbalstīti lietotāja parametri un aizstājvārdi. Vienums system.run[] tiek atbalstīts neatkarīgi no AllowKey, AllowKeyRegexp, DenyKey vai DenyKeyRegexp iestatījumiem.

Automātiskās reģistrācijas pieprasījuma laikā aģents reģistrēs brīdinājuma ziņojumu, ja norādītā vienuma atgrieztā vērtība pārsniedz 65535 UTF-8 koda punktu ierobežojumu. Vienuma atgrieztajai vērtībai ir jābūt UTF-8 virknei; pretējā gadījumā tā tiks ignorēta. Ja parametrs sagaida IP adresi vai DNS nosaukumu, tiks noraidītas un kā nederīgas paziņotas arī vērtības, kas ir derīgas UTF-8 virknes, bet nav derīgas IP adreses vai DNS nosaukumi.

Vairākrindu metadati netiek atbalstīti — izvade tiks saīsināta pie pirmās rindiņas pārtraukuma.

Hostname

Norādiet unikālus, reģistrjutīgus hostu nosaukumus. Ar komatiem atdalīts saraksts.
 Obligāts aktīvajām pārbaudēm, un tam jāsakrīt ar hostu nosaukumiem, kā tie ir konfigurēti 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: iestatīts ar HostnameItem

HostnameItem

Norādiet vienumu resursdatora nosaukuma iegūšanai.
Šī opcija tiek izmantota tikai tad, ja Hostname nav definēts.
Lietotāja parametri vai aizstājvārdi netiek atbalstīti, taču vienums system.run[] tiek atbalstīts neatkarīgi no AllowKey, AllowKeyRegexp, DenyKey vai DenyKeyRegexp vērtībām.

Noklusējums: system.hostname

Iekļaut

Norādiet direktoriju vai atsevišķus failus, kas jāiekļauj konfigurācijas failā (pēc noklusējuma tas atrodas mapē C:\Program Files\Zabbix Agent 2, ja Zabbix aģents ir instalēts, izmantojot Windows MSI instalācijas pakotnes; tas atrodas instalēšanas laikā norādītajā mapē, ja Zabbix aģents ir instalēts kā zip arhīvs). Zabbix instalēšanas laikā šis parametrs tiek iestatīts uz /usr/local/etc, ja vien kompilēšanas laikā to nemaināt.

Vispārīgie Include parametra noteikumi:

- Ceļš var būt relatīvs pret zabbix_agent2.conf faila atrašanās vietu (piemēram, Include=.\zabbix_agent2.d\plugins.d*.co
- Vienā konfigurācijas failā var norādīt vairākus Include parametrus.
- Zabbix apstrādā Include parametru uzreiz pēc tā nolasīšanas, tāpēc apsveriet iespēju to ievietot konfigurācijas faila beigās.

Iekļauto failu noteikumi:

- Katram failam jābūt lasāmam.
- Katram failam jābūt ar pareizu sintaksi. Pretējā gadījumā komponents netiks palaists.

Direktoriju vai failu šablonu iekļaušanas noteikumi:

- Neuzskatiet, ka iekļaušana notiks noteiktā secībā (piemēram, faili netiek iekļauti alfabētiskā secībā).
- Nedefinējiet vienu un to pašu parametru vairāk nekā vienā iekļautajā failā (piemēram, lai aizstātu vispārīgu iestatījumu ar konkrētu), jo nav iespējams paredzēt, kura vērtība tiks izmantota.
- Pievērsiet uzmanību rezerves kopijām, ko daži teksta redaktori izveido automātiski (piemēram, rediģējot include/my_specific.conf var tikt izveidots include/my_specific.conf.BAK), jo tiks iekļauti abi faili.

Lai no direktorijas iekļautu tikai atbilstošos failus, izmantojiet aizstājējzīmi (*) atbilstības noteikšanai.

Piemērs:

```
Include=C:\Program Files\Zabbix Agent2\zabbix_agent2.d\*.conf
```

ListenIP

Norādiet aģenta klausīšanās IP adresi savienojumiem no servera/starpniekservera. Ar komatiem atdalīts saraksts.
 Pirmais IP adreses ieraksts tiek nosūtīts Zabbix serverim, ja tiek veidots savienojums ar to, lai izgūtu aktīvo pārbaužu sarakstu.

Noklusējums: 0.0.0.0

ListenPort

Norādiet aģenta klausīšanās portu savienojumiem no servera/starpniekservera.

Noklusējums: 10050
 Diapazons: 1024-32767

LogFile

Norādiet aģenta žurnāla failu.

Noklusējums: `c:\zabbix_agent2.log`
 Obligāti: Jā, ja `LogType` ir iestatīts uz `file`; citādi nē

LogFileSize

Iestatiet maksimālo žurnāla faila lielumu (MB).
0 - atspējot automātisku žurnāla rotāciju.
Piezīme: Ja tiek sasniegts žurnāla faila lieluma ierobežojums un faila rotācija neizdodas jebkāda iemesla dēļ, esošais žurnāla fails tiek saīsināts un izveidots no jauna.

Noklusējums: 1
 Diapazons: 0-1024

LogType

Norādiet žurnāla izvades tipu: - *file* - ierakstīt žurnālu failā, kas norādīts ar `LogFile` parametru; - *console* - ierakstīt žurnālu standarta izvades plūsmā.

Noklusējums: `file`

PerfCounter

Definējiet 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 vidējo procesora pārtraukumu skaitu sekundē pēdējās minūtes laikā, varat definēt jaunu parametru "interrupts" šādi:

```
PerfCounter = interrupts,"\Processor(0)\Interrupts/sec",60
```

Lūdzu, ņemiet vērā dubultpēdiņas ap veiktspējas skaitītāja ceļu. Parametra nosaukums (interrupts) ir 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

Definējiet jaunu parametru <parameter_name>, kas ir sistēmas veiktspējas skaitītāja <perf_counter_path> vidējā vērtība norādītajam laika periodam <period> (sekundēs). Salīdzinājumā 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 vidējo procesora pārtraukumu skaitu sekundē pēdējās minūtes laikā, varat definēt jaunu parametru "interrupts" šādi:

```
PerfCounterEn = interrupts,"\Processor(0)\Interrupts/sec",60
```

Lūdzu, ņemiet vērā dubultpēdiņas ap veiktspējas skaitītāja ceļu. Parametra nosaukums (interrupts) ir jāizmanto kā vienuma atslēga, veidojot vienumu. Paraugi vidējās vērtības aprēķināšanai tiks ņemti katru sekundi.
Angļu virkņu sarakstu var atrast, skatot šādu reģistra atslēgu: `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Perflib\009`.

Noturīgā bufera fails

Norādiet failu, kurā Zabbix aģents 2 glabās SQLite datubāzi. Tam jābūt pilnam faila nosaukumam.
 Šis parametrs tiek izmantots tikai tad, ja ir iespējots noturīgais buferis (`EnablePersistentBuffer=1`).

PersistentBufferPeriod

Iestata datu glabāšanas ilgumu, ja nav savienojuma ar serveri vai starpniekserveri. Vecākie dati tiks zaudēti. Žurnāla dati tiks saglabāti.
 Šis parametrs tiek izmantots tikai tad, ja ir iespējots pastāvīgais buferis (`EnablePersistentBuffer=1`).

Noklusējums: 1h
 Diapazons: 1m-365d

Spraudņi.<PluginName>.System.Capacity

Iestatiet vienlaicīgi izpildāmo pārbažu limitu katram `<PluginName>` spraudnim.

Noklusējums: 1000 Diapazons: 1-1000

Plugins.Log.MaxLinesPerSecond

Iestata maksimālo jauno rindu skaitu, ko aģents nosūtīs sekundē uz Zabbix serveris vai starpniekserveris, apstrādājot `log`, `logrt` un `eventlog` aktīvās pārbaudes. Norādīto vērtību pārrakstīs `maxlines` parametrs, kas norādīts `log`, `logrt` vai `eventlog` vienums atslēgā.
Piezīme: Zabbix apstrādās 10 reizes vairāk jauno rindu nekā iestatīts `MaxLinesPerSecond`, lai meklētu nepieciešamo virkni žurnāla vienumos.

Noklusējums: 20
 Diapazons: 1-1000

Plugins.SystemRun.LogRemoteCommands

Iespējot izpildīto čaulas komandu reģistrēšanu kā brīdinājumus. Komandas tiks reģistrētas tikai tad, ja tās tiek izpildītas attālināti. Žurnāla ieraksti netiks izveidoti, ja `system.run[]` tiek palaists lokāli, izmantojot `HostMetadataItem`, `HostInterfaceItem` vai `HostnameItem` parametrus.

Noklusējums: 0
 Vērtības: 0 - atspējots, 1 - iespējots

PluginSocket

Norādiet Windows nosauktās caurules ceļu ielādējamo spraudņu saziņai.

Noklusējums: `\\.\pipe\agent.plugin.sock`

PluginTimeout

Iestatiet savienojumu noildzi ar ielādējamiem spraudņiem sekundēs.

Noklusējums: `Timeout`
 Diapazons: 1-30

ProfilerDir

Direktorija, kurā tiks ierakstīti profilēšanas dati.

Noklusējums: `c:\zabbix_pprof`

ProfilerMaxFilesPerProfile

Maksimālais saglabājamo failu skaits katram profila tipam. Ierobežojums tiek piemērots pabeigtajiem failiem pēc katras profila izdrukas. Aktīvais CPU profils tiek apkopots slēptā pagaidu failā un netiek ieskaitīts ierobežojumā. Uz `periodic_prof_execute` izveidotajiem failiem arī attiecas šis ierobežojums.

Noklusējums: 10
 Diapazons: 1-100

ProfilerInterval

Profilēšanas datu ierakstīšanas intervāls sekundēs.

Noklusējums: 3600
 Diapazons: 1-86400

RefreshActiveChecks

Norādiet aktīvo pārbaužu saraksta atsvaidzināšanas biežumu sekundēs.
 Ņemiet vērā, ka pēc aktīvo pārbaužu atsvaidzināšanas neveiksmes nākamais atsvaidzināšanas mēģinājums tiks veikts pēc 60 sekundēm.

Noklusējums: 5
 Diapazons: 1-86400

serveris

Norādiet Zabbix serveru/starpniekserveru IP adreses, pēc izvēles CIDR notācijā, vai DNS nosaukumus. Ar komatiem atdalīts saraksts.
 Ja parametrs nav norādīts, pasīvie pārbaudes tiek atspējotas un aģents neklausīsies nevienā TCP portā.
 Ienākošie savienojumi tiks pieņemti tikai no šeit uzskaitītajiem hosts. Ja ir iespējots IPv6 atbalsts, tad `127.0.0.1`, `::127.0.0.1`, `::ffff:127.0.0.1` tiek uzskatīti par ekvivalentiem, un `::/0` atļaus jebkuru IPv4 vai IPv6 adresi. `0.0.0.0/0` var izmantot, lai atļautu jebkuru IPv4 adresi. Atstarpes ir atļautas.

Piemērs:

```
Server=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com
```

ServerActive

Norādiet Zabbix servera/starpniekservera adresi vai klastera konfigurāciju, no kuras saņemt aktīvās pārbaudes.
 Ja parametrs nav norādīts, aktīvās pārbaudes ir atspējotas.

Servera/starpniekservera adrese ir IP adrese vai DNS nosaukums, un izvēles ports, kas atdalīts ar kolu. 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.

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 šim starpniekserverim Zabbix serveris/klasteris nedrīkst būt norādīts.

Var norādīt vairākas ar komatiem atdalītas adreses, lai paralēli izmantotu vairākus neatkarīgus Zabbix serverus. Atstarpes ir atļautas.

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

Norādiet avota IP adresi šādiem gadījumiem:

- izejošajiem savienojumiem ar Zabbix serveri vai Zabbix starpniekserveri;
- savienojumu izveidei, izpildot dažus vienumus (`web.page.get`, `net.tcp.port` u.c.).

StatusPort

Iestatiet portu, kurā aģents klausās HTTP statusa pieprasījumus (`http://localhost:<port>/status`).

Diapazons: 1024-32767

Noildze

Norādiet, cik ilgi jāgaida (sekundēs), lai izveidotu savienojumu un apmainītos ar datiem ar Zabbix starpniekserveri vai serveri.

Šis parametrs nosaka dažādu saziņas darbību ilgumu, tostarp:

- gaidīšanu uz atbildi no Zabbix servera;
- pieprasījumu nosūtīšanu Zabbix serverim, tostarp vienuma konfigurācijas pieprasījumus un vienuma datus aktīvajās pārbaudēs;
- žurnāla datu izgūšanu, izmantojot logfile vai Windows notikumu žurnāla uzraudzību;
- sirdspukstu ziņojumu nosūtīšanu;
- maksimālo ilgumu `vfs.*` pārbaudēm;
- izmantošanu kā rezerves variantu situācijās, kad serveris vai starpniekserveris, kas ir vecāks par 7.0 versiju, nosūta 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

Norādiet ienākošo savienojumu prasīto šifrēšanas līmeni.
 Tiek izmantots 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.

Obligāti: jā, ja ir definēti TLS sertifikāta vai PSK parametri (pat *unencrypted* savienojumam); citādi nē

TLSCAFile

Norādiet pilnu ceļu uz failu, kurā ir augstākā līmeņa CA sertifikāti, kas tiek izmantoti vienaudžu sertifikātu verifikācijai un šifrētai saziņai starp Zabbix komponentiem.

TLSCertFile

Norādiet pilnu ceļu uz failu, kurā atrodas aģenta sertifikāts vai sertifikātu ķēde, kas tiek izmantota šifrētai saziņai ar Zabbix komponentiem.

TLSCipherAll13

Norādiet OpenSSL šifru virkni (TLS 1.3). Pārrakstiet noklusējuma šifru komplektu atlases kritērijus sertifikātu un PSK balstītai šifrēšanai.

Piemērs:

```
TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSCipherCert13

Norādiet OpenSSL šifru virkni (TLS 1.3). Pārrakstiet noklusējuma šifru kopas atlases kritērijus sertifikātu balstītai šifrēšanai.

Ņemiet vērā, ka šo parametru nevar izmantot kopā ar `TLSAccept=cert,psk`; sertifikāta savienojumam (`TLSConnect=cert`) izmantojiet `TLSCipherAll13` vietā.

TLSCipherPSK13

Norādiet OpenSSL šifra virkni (TLS 1.3). Pārrakstiet noklusējuma šifru kopas atlases kritērijus PSK balstītai šifrēšanai.

Piemērs:

```
TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSConnect

Norādiet šifrēšanas līmeni, kas nepieciešams aģenta savienojumiem ar Zabbix serveri vai starpniekserveri.
 Tiek izmantots aktīvajām pārbaudēm.
 Var norādīt tikai vienu vērtību: - *unencrypted* - savienoties bez šifrēšanas (noklusējums); - *psk* - savienoties, izmantojot TLS un iepriekš koplietotu atslēgu (PSK); - *cert* - savienoties, izmantojot TLS un sertifikātu.

Obligāts: jā, ja ir definēti TLS sertifikāta vai PSK parametri (pat *unencrypted* savienojumam); citos gadījumos nē

TLSCRLFile

Norādiet pilnu ceļu uz failu, kurā ir atsauktie sertifikāti. Šis parametrs tiek izmantots šifrētai saziņai starp Zabbix komponentiem.

TLSKeyFile

Norādiet pilnu ceļu uz failu, kurā atrodas aģenta privātā atslēga, kas tiek izmantota šifrētai saziņai starp Zabbix komponentiem.

TLSPSKFile

Norādiet pilnu faila ceļu uz failu, kurā atrodas aģenta iepriekš koplietotā atslēga, kas tiek izmantota šifrētai saziņai ar Zabbix serveri vai starpniekserveri.

TLSPSKIdentity

Norādiet iepriekš koplietotās atslēgas identitātes virkni, ko izmanto šifrētai saziņai ar Zabbix serveri vai starpniekserveri.

TLSServerCertIssuer

Norādiet atļauto servera (starpniekservera) sertifikāta izdevēju.

TLSServerCertSubject

Norādiet atļauto servera (starpniekservera) sertifikāta subjektu.

UnsafeUserParameters

Atļaut visus simbolus nodot argumentiem lietotāja definētajos parametros. Šādi simboli nav atļauti: \ ' " ` * ? [ ] { } ~ $! & ; ( ) < > | # @ %
Turklāt rindiņas pārtraukuma simboli nav atļauti.

Noklusējums: 0
 Vērtības: 0 - neatļaut, 1 - atļaut

UserParameter

Norādiet lietotāja definētu parametru, ko uzraudzīt. Var būt vairāki lietotāja definēti parametri.
Formāts: `UserParameter=<key>,<shell command>`
Ņemiet vērā, ka čaulas komandai nedrīkst atgriezt tukšu virkni vai tikai EOL. Čaulas komandās var izmantot relatīvus ceļus, ja ir norādīts parametrs `UserParameterDir`.

Piemērs:

```
UserParameter=system.test,dir /b | find /c /v ""
UserParameter=check_cpu,.\custom\_script.bat
```

UserParameterDir

Norādiet noklusējuma meklēšanas ceļu `UserParameter` komandām. Ja šis iestatījums tiek izmantots, aģents pirms komandas izpildes mainīs savu darba direktoriju uz šeit norādīto. Tādējādi `UserParameter` komandām var izmantot relatīvu `.\` prefiksu pilna ceļa vietā.
Atļauts tikai viens ieraksts.

Piemērs:

```
UserParameterDir=C:\opt\myscripts
```

Konfigurācijas parametri: spraudņi

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 aprakstīti spraudņu konfigurācijas principi un labākā prakse.

Visi spraudņi tiek konfigurēti, izmantojot parametru `Plugins.*`, kas var būt daļa no Zabbix aģenta 2 konfigurācijas faila vai spraudņa konfigurācijas faila. Ja spraudnis izmanto atsevišķu konfigurācijas failu, ceļš uz šo failu jānorāda Zabbix aģenta 2 konfigurācijas faila parametram `Include`.

Tipiskam spraudņa parametram ir šāda struktūra:

`Plugins.<PluginName>.<Parameter>=<Value>`

Turklāt pastāv divas īpašas 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 nosauktas sesijas.

Visiem parametru nosaukumiem jāatbilst šādām prasībām:

- Ieteicams spraudņu nosaukumus rakstīt ar lielo sākumburtu.
- Parametram jāsākas ar lielo burtu.
- Speciālās rakstzīmes nav atļautas.
- Ligzdošanas līmeņu maksimālais skaits nav ierobežots.
- Parametru skaits nav ierobežots.

Piemēram, lai veiktu aktīvās pārbaudes, kurām tūlīt pēc aģenta restartēšanas nav *Scheduling* atjaunināšanas intervāla, tikai Uptime spraudnim konfigurācijas failā konfigurācijas failā iestatiet `Plugins.Uptime.System.ForceActiveChecksOnStart=1`.

Līdzīgi, lai CPU spraudnim iestatītu pielāgotu vienlaicīgo pārbaužu ierobežojumu, konfigurācijas failā iestatiet `Plugins.CPU.System.Capaci`

Noklusējuma vērtības

Savienojuma parametru (URI, lietotājvārda, paroles u. c.) noklusējuma vērtības var iestatīt konfigurācijas failā, izmantojot šādu formātu:

```
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 kodā iekļautās noklusējuma vērtības.

> **Note:**
> Ja vienuma atslēgai nav parametru, Zabbix aģents 2 mēģinās iegūt metriku, izmantojot noklusējuma parametru sadaļā definētās vērtības.

Nosauktās sesijas

Nosauktās sesijas ir papildu spraudņa parametru līmenis, un tās var izmantot, lai norādītu atsevišķus autentifikācijas parametru kopumus katram no uzraudzītajiem instancēm. Katram nosauktās sesijas parametram jābūt šādai struktūrai:

```
Plugins.<PluginName>.Sessions.<SessionName>.<Parameter>=<Value>
```

Sesijas nosaukumu var izmantot kā connString vienums atslēgas parametru, nevis atsevišķi norādot URI, lietotājvārdu un/vai paroli.

Vienums 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 iegultu URI akreditācijas datu nodošana vienums atslēgā netiek atbalstīta; tā vietā izmantojiet nosauktās sesijas parametrus.

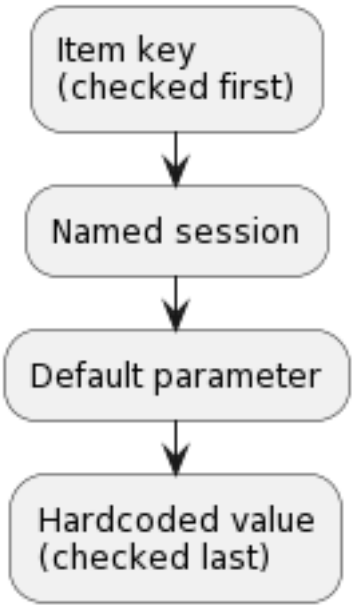
Pieejamo nosaukto sesijas parametru saraksts ir atkarīgs no spraudņa.

Sesijas parametrus ir iespējams pārrakstīt, vienums atslēgas parametros norādot jaunas vērtības (skatiet piemēru).

Ja parametrs nav definēts nosauktajai sesijai, Zabbix aģents 2 izmantos vērtību, kas definēta noklusējuma spraudņa parametrā.

Parametra 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 faktisku 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 sesijas parametrs arī nav norādīts, tiks izmantota atbilstošajā noklusējuma parametrā definētā vērtība.
5. Ja nekas cits nepalīdz, spraudnis izmantos cieti kodēto 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://192.0.2.1:3306
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 vienuma atslēgā ir izlaista): `session_password`
- Datubāzes nosaukums no vienuma atslēgas: `postgres`

Piemērs 3

Metrikas apkopošana, 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 metrikas vākšanu no vairākiem instancēm vienlaikus. Var uzraudzīt gan lokālās, gan attālās instances. Tiek atbalstīti TCP un Unix-socket savienojumi.

Ieteicams konfigurēt spraudņus tā, lai savienojumi ar instancēm tiktu uzturēti atvērtā stāvoklī. Tas sniedz priekšrocības, samazinot tīkla noslodzi, latentumu, kā arī CPU un atmiņas izmantošanu, jo tiek izmantots 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`

Ceph spraudnis

Pārskats

Konfigurācijas parametri ļauj pielāgot Ceph spraudņa darbību.

Ieteicams tos norādīt atsevišķā konfigurācijas failā (piem., `ceph.conf`), nevis aģents 2 konfigurācijas failā. Pēc tam izmantojiet direktīvu Include, lai iekļautu šo failu Zabbix aģents 2 konfigurācijā.

Visi parametri ir **neobligāti**, ja vien nav skaidri norādīts, ka parametrs ir obligāts.

Šī lapa ir strukturēta šādi:

- Parametru pārskats (lai skatītu pilnu parametra informāciju, noklikšķiniet uz tā nosaukuma)
- Parametru detalizēta informācija

Parameter	Description
Plugins.Ceph.Default.ApiKey	Norādiet noklusējuma API atslēgu savienojumam ar Ceph.
Plugins.Ceph.Default.Mode	Norādiet noklusējuma režīmu savienojumam ar Ceph.
Plugins.Ceph.Default.URI	Norādiet noklusējuma URI savienojumam ar Ceph.
Plugins.Ceph.Default.User	Norādiet noklusējuma lietotājvārdu savienojumam ar Ceph.
Plugins.Ceph.InsecureSkipVerify	Norādiet, vai HTTP klientam jāpārbauda serveris sertifikātu ķēde un resursdatora nosaukums.
Plugins.Ceph.KeepAlive	Iestatiet maksimālo gaidīšanas laiku pirms neizmantoto spraudņa savienojumu aizvēršanas.

Parameter	Description
Plugins.Ceph.Sessions.<SessionName>.ApiKey	Norādiet nosauktās sesijas API atslēgu savienojumam ar Ceph.
Plugins.Ceph.Sessions.<SessionName>.Mode	Norādiet nosauktās sesijas režīmu savienojumam ar Ceph.
Plugins.Ceph.Sessions.<SessionName>.URI	Norādiet nosauktās sesijas URI savienojumam ar Ceph.
Plugins.Ceph.Sessions.<SessionName>.User	Norādiet nosauktās sesijas lietotājvārdu savienojumam ar Ceph.
Plugins.Ceph.Timeout	Iestatiet pieprasījuma izpildes noildzi.

Ņemiet vērā:

- Noklusējuma vērtības atspoguļo procesa noklusējuma iestatījumus, nevis vērtības piegādātajos konfigurācijas failos;
- Vērtības atbalsta vides mainīgos;
- Zabbix atbalsta konfigurācijas failus tikai UTF-8 kodējumā bez BOM;
- Komentāri, kas sākas ar "#", tiek atbalstīti tikai rindas sākumā.

Parametra informācija

Plugins.Ceph.Default.ApiKey

Norādiet noklusējuma API atslēgu savienojumam ar Ceph.
 Tiek izmantota, ja vienums atslēgā vai nosauktā sesijā nav norādīta vērtība.

Plugins.Ceph.Default.Mode

Norādiet noklusējuma režīmu savienojumam ar Ceph.
 Dzimtais režīms tiek atbalstīts tikai Linux un, sākot ar Ceph 16. Restful režīms nedarbosies ar Ceph versiju 20 vai jaunāku.
 Ņemiet vērā, ka lietotāja akreditācijas datu kopums katram režīmam atšķiras, un tie nav savstarpēji saderīgi.

Noklusējums: `restful` (novecojis)
 Diapazons: native/restful

Plugins.Ceph.Default.URI

Norādiet noklusējuma URI savienojumam ar Ceph.
 Tiek izmantots, ja vērtība nav norādīta vienuma atslēgā vai nosauktā sesijā.

Nejāiekļauj iegultus akreditācijas datus (tie tiks ignorēti).
Jāatbilst URI formātam.
Tiek atbalstīta tikai `https` shēma; shēmu var nenorādīt.
Portu var nenorādīt (noklusējums=8003).

Noklusējums: `https://localhost:8003`

Piemēri:

```
https://127.0.0.1:8003
localhost
```

Plugins.Ceph.Default.User

Norādiet noklusējuma lietotājvārdu savienojumam ar Ceph.
 Tiek izmantots, ja vienums atslēgā vai nosauktā sesijā nav norādīta vērtība.

Plugins.Ceph.InsecureSkipVerify

Norādiet, vai HTTP klients pārbauda servera sertifikātu ķēdi un resursdatora nosaukumu.
 Ja *true*, TLS pieņem jebkuru servera uzrādīto sertifikātu un jebkuru šajā sertifikātā norādīto resursdatora nosaukumu. Šajā režīmā TLS ir pakļauts starpnieka uzbrukumiem (izmantojiet tikai testēšanai).

Noklusējums: `false`
 Diapazons: false/true

Plugins.Ceph.KeepAlive

Iestata maksimālo gaidīšanas laiku (sekundēs) pirms neizmantotie spraudņa savienojumi tiek aizvērti.

Noklusējums: 300
 Diapazons: 60-900

Plugins.Ceph.Sessions.<SessionName>.ApiKey

Norādiet nosauktās sesijas API atslēgu savienojumam ar Ceph.
 <SessionName> - norādiet sesijas nosaukumu, ko izmantot vienumu atslēgās.

Plugins.Ceph.Sessions.<SessionName>.Mode

Norādiet nosauktās sesijas režīmu savienojumam ar Ceph.
 <SessionName> - definējiet sesijas nosaukumu izmantošanai vienumu atslēgās.
 Native režīms tiek atbalstīts tikai Linux un, sākot ar Ceph 16. Restful režīms nedarbosies ar Ceph 20 vai jaunāku versiju.
 Ņemiet vērā, ka lietotāja akreditācijas datu kopums katram režīmam atšķiras, un tie nav savstarpēji saderīgi.

Noklusējums: `restful` (novecojis)
 Diapazons: native/restful

Plugins.Ceph.Sessions.<SessionName>.URI

Norādiet nosauktās sesijas URI, lai izveidotu savienojumu ar Ceph.
 <SessionName> - definējiet sesijas nosaukumu, ko izmantot vienumu atslēgās.
Nedrīkst ietvert iegultus akreditācijas datus (tie tiks ignorēti).
Jāatbilst URI formātam.
 Tiek atbalstīta tikai `https` shēma; shēmu var arī nenorādīt.
Portu var nenorādīt (noklusējums=8003).

Noklusējums: `https://localhost:8003`

Piemēri:

```
https://127.0.0.1:8003
localhost
```

Plugins.Ceph.Sessions.<SessionName>.User

Norādiet nosauktās sesijas lietotājvārdu, lai izveidotu savienojumu ar Ceph.
 <SessionName> - definējiet sesijas nosaukumu, ko izmantot vienumu atslēgās.

Plugins.Ceph.Timeout

Iestatiet pieprasījuma izpildes noildzi (ilgumu, cik ilgi gaidīt, līdz pieprasījums tiek pabeigts, pirms to pārtrauc) sekundēs.

Noklusējums: `global timeout<br>` Diapazons: 1-30

Skatīt arī

- Norādījumi par sprauduņu konfigurēšanu
- Vispārīgo Zabbix aģenta 2 konfigurācijas parametru apraksts: Zabbix aģents 2 (UNIX) / Zabbix aģents 2 (Windows)

Docker spraudnis

Pārskats

Konfigurācijas parametri ļauj pielāgot Docker sprauduņa darbību.

Ieteicams tos norādīt atsevišķā konfigurācijas failā (piem., `docker.conf`), nevis aģents 2 konfigurācijas failā. Pēc tam izmantojiet direktīvu Include, lai iekļautu šo failu Zabbix aģents 2 konfigurācijā.

Visi parametri ir **neobligāti**, ja vien nav skaidri norādīts, ka parametrs ir obligāts.

Šī lapa ir strukturēta šādi:

- Parametru pārskats (lai skatītu pilnu parametra informāciju, noklikšķiniet uz tā nosaukuma)
- Parametru informācija

Parameter	Description
Plugins.Docker.Endpoint	Norādiet Docker dēmona UNIX-soketa atrašanās vietu..
Plugins.Docker.Timeout	Norādiet pieprasījuma izpildes noildzi.

Ņemiet vērā, ka:

- Noklusējuma vērtības atspoguļo procesa noklusējumus, nevis vērtības piegādātajos konfigurācijas failos;
- Vērtības atbalsta vides mainīgos;
- Zabbix atbalsta konfigurācijas failus tikai UTF-8 kodējumā bez BOM;
- Komentāri, kas sākas ar "#", tiek atbalstīti tikai rindas sākumā.

Parametra informācija

Plugins.Docker.Endpoint

Norādiet Docker dēmona UNIX ligzdas atrašanās vietu.
 Jāietver shēma (tiek atbalstīts tikai `unix://`).

Noklusējums: `unix:///var/run/docker.sock`

Plugins.Docker.Timeout

Norādiet pieprasījuma izpildes noildzi (ilgumu sekundēs, cik ilgi gaidīt, līdz pieprasījums tiks pabeigts, pirms to pārtrauc).

Noklusējums: `global timeout<br>` Diapazons: 1–30

Skatīt arī

- Vispārīgo Zabbix aģenta 2 konfigurācijas parametru apraksts: Zabbix aģents 2 (UNIX) / Zabbix aģents 2 (Windows)
- Norādījumi par sprauduņu konfigurēšanu

Ember+ spraudnis

Pārskats

Konfigurācijas parametri ļauj pielāgot Ember+ spraudņa darbību.

Ieteicams tos norādīt atsevišķā konfigurācijas failā (piem., `emberplus.conf`), nevis aģents 2 konfigurācijas failā. Pēc tam izmantojiet direktīvu Include, lai iekļautu šo failu Zabbix aģents 2 konfigurācijā.

Ember+ spraudnis ir ielādējams spraudnis, un tas ir pieejams, kā arī pilnībā aprakstīts Ember+ spraudņa repozitorijā.

Šis spraudnis pašlaik ir pieejams tikai kompilēšanai no pirmkoda (gan Unix, gan Windows vidē).

Visi parametri ir **neobligāti**, ja vien nav skaidri norādīts, ka parametrs ir obligāts.

Šī lapa ir strukturēta šādi:

- Parametru pārskats (lai skatītu pilnu parametra informāciju, noklikšķiniet uz tā nosaukuma)
- Parametru detaļas

Parameter	Description
Plugins.EmberPlus.Default.Uri	Norādiet noklusējuma URI savienojumam ar Ember+.
Plugins.EmberPlus.KeepAlive	Iestatiet maksimālo gaidīšanas laiku pirms neizmantoto spraudņa savienojumu aizvēršanas.
Plugins.EmberPlus.Sessions.<SessionName>.Uri	Norādiet nosauktās sesijas URI savienojumam ar Ember+.
Plugins.EmberPlus.System.Path	Norādiet Ember+ spraudņa izpildāmā faila ceļu.
Plugins.EmberPlus.Timeout	Iestatiet pieprasījuma izpildes noildzi.

Ņemiet vērā, ka:

- Noklusējuma vērtības atspoguļo procesa noklusējuma iestatījumus, nevis vērtības piegādātajos konfigurācijas failos.
- Vērtības atbalsta vides mainīgos;
- Zabbix atbalsta konfigurācijas failus tikai UTF-8 kodējumā bez BOM.
- Komentāri, kas sākas ar "#", tiek atbalstīti tikai rindas sākumā.

Parametra informācija

Plugins.EmberPlus.Default.Uri

Norādiet noklusējuma URI savienojumam ar Ember+.
 Vienīgā atbalstītā shēma ir `tcp://`. Shēmu var arī nenorādīt. Iebūvētie akreditācijas dati tiks ignorēti.

Noklusējums: `tcp://localhost:9998`

Plugins.EmberPlus.KeepAlive

Iestata maksimālo gaidīšanas laiku (sekundēs) pirms neizmantotie spraudņa savienojumi tiek slēgti.

Noklusējums: 300
 Diapazons: 60-900

Plugins.EmberPlus.Sessions.<SessionName>.Uri

Norādiet nosauktās sesijas URI savienojumam ar Ember+.
 Vienīgā atbalstītā shēma ir `tcp://`. Shēmu var arī nenorādīt. Iegultie akreditācijas dati tiks ignorēti.
 <SessionName> - definējiet sesijas nosaukumu, ko izmantot vienumu atslēgās.

Noklusējums: `tcp://localhost:9998`

Plugins.EmberPlus.System.Path

Norādiet Ember+ spraudņa izpildāmā faila ceļu.

Piemērs: `Plugins.EmberPlus.System.Path=/usr/libexec/zabbix/zabbix-agent2-plugin-ember-plus`

Plugins.EmberPlus.Timeout

Iestatiet ilgumu sekundēs, cik ilgi gaidīt, līdz serveris atbild, veidojot pirmo savienojumu un veicot turpmākās darbības sesijā.

Noklusējums: `global timeout`
 Diapazons: 1-30

Opcijas

Parameter	Description
-V --version	Izdrukā spraudņa versiju un licences informāciju.
-h --help	Izdrukā palīdzības informāciju (saīsināti).

Parameter	Description
-t, --test <item key>	Palaiž spraudni testēšanai (spraudņa konfigurācija tiek ignorēta).

Skatiet arī:

- Vispārīgo Zabbix aģents 2 konfigurācijas parametru apraksts: Zabbix aģents 2 (UNIX) / Zabbix aģents 2 (Windows)
- Norādījumi par spraudņu konfigurēšanu

Memcached spraudnis

Pārskats

Konfigurācijas parametri ļauj pielāgot Memcached spraudņa darbību.

Ieteicams tos norādīt atsevišķā konfigurācijas failā (piem., `memcached.conf`), nevis aģents 2 konfigurācijas failā. Pēc tam izmantojiet direktīvu Include, lai iekļautu šo failu Zabbix aģents 2 konfigurācijā.

Visi parametri ir **neobligāti**, ja vien nav skaidri norādīts, ka parametrs ir obligāts.

Šī lapa ir strukturēta šādi:

- Parametru pārskats (lai skatītu pilnu parametra informāciju, noklikšķiniet uz tā nosaukuma)
- Parametru detaļas

Parameter	Description
Plugins.Memcached.Default.Password	Norādiet noklusējuma paroli savienojumam ar Memcached.
Plugins.Memcached.Default.Uri	Norādiet noklusējuma URI savienojumam ar Memcached.
Plugins.Memcached.Default.User	Norādiet noklusējuma lietotājvārdu savienojumam ar Memcached.
Plugins.Memcached.KeepAlive	Iestatiet maksimālo gaidīšanas laiku pirms neizmantoto spraudņa savienojumu aizvēršanas.
Plugins.Memcached.Session.Password	Norādiet nosaukta sesijas paroli savienojumam ar Memcached.
Plugins.Memcached.Session.Uri	Norādiet nosaukta sesijas URI savienojumam ar Memcached.
Plugins.Memcached.Session.User	Norādiet nosaukta sesijas lietotājvārdu savienojumam ar Memcached.
Plugins.Memcached.Timeout	Iestatiet pieprasījuma izpildes noildzi.

Ņemiet vērā:

- Noklusējuma vērtības atspoguļo procesa noklusējuma iestatījumus, nevis vērtības piegādātajos konfigurācijas failos;
- Vērtības atbalsta vides mainīgos;
- Zabbix atbalsta konfigurācijas failus tikai UTF-8 kodējumā bez BOM;
- Komentāri, kas sākas ar ”#”, tiek atbalstīti tikai rindas sākumā.

Parametra informācija

Plugins.Memcached.Default.Password

Norādiet noklusējuma paroli savienojumam ar Memcached.
 Tiek izmantota, ja vienums atslēgā vai nosauktajā sesijā nav norādīta vērtība.

Plugins.Memcached.Default.Uri

Norādiet noklusējuma URI savienojumam ar Memcached.
 Tiek izmantots, ja vienums atslēgā vai nosauktajā sesijā nav norādīta vērtība.
 Iebūvētie akreditācijas dati tiks ignorēti.
 Jāatbilst URI formātam. Atbalstītās shēmas: `tcp`, `unix`.
 Shēmu var nenorādīt.
 Portu var nenorādīt (noklusējums=11211).

Noklusējums: `tcp://localhost:11211`

Piemēri:

```
tcp://localhost:11211
localhost
unix:/var/run/memcached.sock
```

Plugins.Memcached.Default.User

Norādiet noklusējuma lietotājvārdu savienojumam ar Memcached.
 Tiek izmantots, ja vienums atslēgā vai nosauktajā sesijā nav norādīta vērtība.

Plugins.Memcached.KeepAlive

Iestata maksimālo gaidīšanas laiku (sekundēs) pirms neizmantotie spraudņa savienojumi tiek slēgti.

Noklusējums: 300
 Diapazons: 60–900

Plugins.Memcached.Sessions.<SessionName>.Password

Norādiet nosauktās sesijas paroli savienojumam ar Memcached.
 <SessionName> - definējiet sesijas nosaukumu izmantošanai vienumu atslēgās.

Plugins.Memcached.Sessions.<SessionName>.Uri

Norādiet nosauktās sesijas URI, lai izveidotu savienojumu ar Memcached.
 <SessionName> - definējiet sesijas nosaukumu, ko izmantot vienumu atslēgās.
 Iegultie akreditācijas dati 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.Memcached.Sessions.<SessionName>.User

Norādiet nosauktās sesijas lietotājvārdu savienojumam ar Memcached.
 <SessionName> - definējiet sesijas nosaukumu, ko izmantot vienumu atslēgās.

Plugins.Memcached.Timeout

Iestata pieprasījuma izpildes noildzi (ilgumu sekundēs, cik ilgi gaidīt, līdz pieprasījums tiks pabeigts, pirms to pārtrauc).

Noklusējums: `global timeout`
 Diapazons: 1–30

Skatīt arī:

- Vispārīgo Zabbix aģents 2 konfigurācijas parametru apraksts: Zabbix aģents 2 (UNIX) / Zabbix aģents 2 (Windows)
- Norādījumi spraudņu konfigurēšanai

Modbus spraudnis

Pārskats

Konfigurācijas parametri ļauj pielāgot Modbus spraudņa darbību.

Ieteicams tos norādīt atsevišķā konfigurācijas failā (piem., `modbus.conf`), nevis aģents 2 konfigurācijas failā. Pēc tam izmantojiet direktīvu Include, lai iekļautu šo failu Zabbix aģents 2 konfigurācijā.

Visi parametri ir **neobligāti**, ja vien nav skaidri norādīts, ka parametrs ir obligāts.

Šī lapa ir strukturēta šādi:

- Parametru pārskats (lai skatītu pilnu parametra informāciju, noklikšķiniet uz tā nosaukuma)
- Parametru detaļas

Parameter	Description
Plugins.Modbus.Sessions.<SessionName>.Endpoint	Norādiet nosauktās sesijas galapunktu savienojumam ar Modbus.
Plugins.Modbus.Sessions.<SessionName>.SlaveID	Norādiet nosauktās sesijas slave ID savienojumam ar Modbus.
Plugins.Modbus.Sessions.<SessionName>.Timeout	Iestatiet nosauktās sesijas noildzi sekundēs.

Ņemiet vērā:

- Noklusējuma vērtības atspoguļo procesa noklusējuma iestatījumus, nevis vērtības piegādātajos konfigurācijas failos;
- Vērtības atbalsta vides mainīgos;
- Zabbix atbalsta konfigurācijas failus tikai UTF-8 kodējumā bez BOM;
- Komentāri, kas sākas ar "#", tiek atbalstīti tikai rindas sākumā.

Parametra informācija

Plugins.Modbus.Sessions.<SessionName>.Endpoint

Norādiet nosauktās sesijas galapunktu savienojumam ar Modbus.
 Savienojuma virkne sastāv no protokola shēmas, host adreses un porta vai seriālā porta nosaukuma un atribūtiem.
 <SessionName> - definējiet sesijas nosaukumu izmantošanai vienumu atslēgās.

Plugins.Modbus.Sessions.<SessionName>.SlaveID

Norādiet nosauktās sesijas slave ID, lai izveidotu savienojumu ar Modbus.
 <SessionName> - definējiet sesijas nosaukumu, ko izmantot vienuma atslēgās.

Piemērs: `Plugins.Modbus.Sessions.MB1.SlaveID=20`

Piezīme - šis nosauktās sesijas parametrs tiek pārbaudīts tikai tad, ja vienuma atslēgā norādītā slave ID parametra vērtība ir tukša.

Plugins.Modbus.Sessions.<SessionName>.Timeout

Iestatiet nosauktās sesijas noildzi sekundēs.
 <SessionName> - definē sesijas nosaukumu izmantošanai vienumu atslēgās.

Piemērs: `Plugins.Modbus.Sessions.MB1.Timeout=2`

Ja nepieciešams iestatīt pieprasījuma izpildes noildzi (cik ilgi gaidīt, līdz pieprasījums tiks pabeigts, pirms to aptur), izmantojiet vienuma konfigurācijas formu.

Skatīt arī

- Vispārīgo Zabbix aģents 2 konfigurācijas parametru apraksts: Zabbix aģents 2 (UNIX) / Zabbix aģents 2 (Windows)
- Norādījumi par sprauduņu konfigurēšanu

MongoDB spraudnis

Pārskats

Konfigurācijas parametri ļauj pielāgot MongoDB sprauduņa darbību.

Ieteicams tos norādīt atsevišķā konfigurācijas failā (piem., `mongodb.conf`), nevis aģents 2 konfigurācijas failā. Pēc tam izmantojiet direktīvu Include, lai iekļautu šo failu Zabbix aģents 2 konfigurācijā.

MongoDB spraudnis ir ielādējams spraudnis, un tas ir pieejams, kā arī pilnībā aprakstīts MongoDB sprauduņa repozitorijā.

Visi parametri ir **neobligāti**, ja vien nav skaidri norādīts, ka parametrs ir obligāts.

Veidojot savienojumu ar MongoDB, spraudnis pārbauda savu konfigurāciju noteiktā secībā, kas nosaka, kura vērtība tiks izmantota. Secība no augstākās līdz zemākajai prioritātei ir šāda:

1. Parametri, kas norādīti savienojuma URI (piem., vaicājuma parametri, kas definēti aiz ? sadaļā `Plugins.MongoDB.Sessions.<sessi` ir ar visaugstāko prioritāti).
2. Parametri Zabbix aģents 2 vienuma atslēgā (piem., mongodb.collection.stats).
3. Parametri, kas definēti nosauktā sesijā (piem., `Plugins.MongoDB.Sessions.<SessionName>.User`, `Plugins.MongoDB.Session`
4. Noklusējuma vērtības (piem., `Plugins.MongoDB.Default.User`, `Plugins.MongoDB.Default.Password`).

Piemēram:

- Ja `*.<SessionName>.Uri` parametrā norādāt datubāzi, ko izmantot autentifikācijai savienojumam ar MongoDB serveris (piem., `Plugins.MongoDB.Sessions.<session_name>.Uri=mongodb://user:password@127.0.0.1:27017/?authSource=` tad spraudnis uzskatīs, ka `*.<SessionName>.AuthSource` parametrs ir iestatīts uz `admin`, pat ja `*.<SessionName>.AuthSource` parametrs ir iestatīts uz citu vērtību.
- Ja `*.<SessionName>.Uri` parametrā norādāt URI (piem., `Plugins.MongoDB.Sessions.<session_name>.Uri=mongodb://127` bet lietotājvārdu un paroli norādāt Zabbix aģents 2 vienuma atslēgā, spraudnis izmantos lietotājvārdu un paroli no vienuma atslēgas, pat ja ir konfigurēti `Plugins.MongoDB.Sessions.<SessionName>.User` un `Plugins.MongoDB.Sessions.<SessionN`

Attention:
Ja tiek izmantota shēma `mongodb+srv://`, URI var izgūt parametrus no DNS TXT ieraksta. No šī TXT ieraksta iegūtie parametri pārraksta visas pārējās konfigurācijas, tostarp URI parametrus, Zabbix Agent 2 vienuma atslēgas parametrus un sesijas parametrus.
 Savienojuma virknē izmantotajam resursdatora nosaukumam var pastāvēt tikai viens TXT ieraksts.

Šī lapa ir strukturēta šādi:

- Parametru pārskats (lai skatītu pilnu parametra informāciju, noklikšķiniet uz tā nosaukuma)
- Parametru detalizēts apraksts

Parameter	Description
Plugins.MongoDB.Default.Auth	Norādiet noklusējuma autentifikācijas mehānismu savienojumam ar MongoDB serveris.
Plugins.MongoDB.Default.AuthSource	Norādiet noklusējuma datubāzi autentifikācijai savienojumam ar MongoDB serveris.
Plugins.MongoDB.Default.Direct	Atspējojiet MongoDB repliku kopas mezglu automātisko noteikšanu.
Plugins.MongoDB.Default.Password	Norādiet noklusējuma paroli.

Parameter	Description
Plugins.MongoDB.Default.ReplicaSet	Norādiet noklusējuma repliku kopas nosaukumu, ko izmanto MongoDB repliku kopas mezglu automātiskai noteikšanai.
Plugins.MongoDB.Default.ReadPreference	Norādiet noklusējuma kārtību savienojumam ar MongoDB repliku kopas mezgliem vai rezerves variantu.
Plugins.MongoDB.Default.TLSConnect	Norādiet šifrēšanas līmeni, kas nepieciešams saziņai starp Zabbix aģents 2 un uzraudzītajām datubāzēm.
Plugins.MongoDB.Default.TLSCAFile	Norādiet pilnu ceļu uz failu, kurā ir augstākā līmeņa CA sertifikāti vienaudža sertifikāta pārbaudei.
Plugins.MongoDB.Default.TLSCertFile	Norādiet pilnu ceļu uz failu, kurā ir aģenta sertifikāts vai sertifikātu ķēde.
Plugins.MongoDB.Default.TLSKeyFile	Norādiet pilnu ceļu uz failu, kurā ir datubāzes privātā atslēga.
Plugins.MongoDB.Default.Uri	Norādiet noklusējuma URI.
Plugins.MongoDB.Default.User	Norādiet noklusējuma lietotājvārdu.
Plugins.MongoDB.KeepAlive	Iestatiet maksimālo gaidīšanas laiku, pirms neizmantotie spraudņa savienojumi tiek slēgti.
Plugins.MongoDB.Sessions.<SessionName>.AuthMechanism	Norādiet nosauktās sesijas autentifikācijas mehānismu savienojumam ar MongoDB serveris.
Plugins.MongoDB.Sessions.<SessionName>.AuthSource	Norādiet nosauktās sesijas datubāzi autentifikācijai savienojumam ar MongoDB serveris.
Plugins.MongoDB.Sessions.<SessionName>.Direct	Atspējot MongoDB repliku kopas mezglu automātisko noteikšanu nosauktajai sesijai.
Plugins.MongoDB.Sessions.<SessionName>.Password	Norādiet nosauktās sesijas paroli.
Plugins.MongoDB.Sessions.<SessionName>.ReplicaSet	Norādiet nosauktās sesijas repliku kopas nosaukumu, ko izmanto MongoDB repliku kopas mezglu automātiskai noteikšanai.
Plugins.MongoDB.Sessions.<SessionName>.ReadPreference	Norādiet nosauktās sesijas kārtību savienojumam ar MongoDB repliku kopas mezgliem vai rezerves variantu.
Plugins.MongoDB.Sessions.<SessionName>.TLSConnect	Norādiet šifrēšanas līmeni, kas nepieciešams saziņai starp Zabbix aģents 2 un uzraudzītajām datubāzēm nosauktajai sesijai.
Plugins.MongoDB.Sessions.<SessionName>.TLSCAFile	Norādiet pilnu ceļu uz failu, kurā ir augstākā līmeņa CA sertifikāti vienaudža sertifikāta pārbaudei nosauktajai sesijai.
Plugins.MongoDB.Sessions.<SessionName>.TLSCertFile	Norādiet pilnu ceļu uz failu, kurā ir aģenta sertifikāts vai sertifikātu ķēde nosauktajai sesijai.
Plugins.MongoDB.Sessions.<SessionName>.TLSKeyFile	Norādiet pilnu ceļu uz failu, kurā ir datubāzes privātā atslēga nosauktajai sesijai.
Plugins.MongoDB.Sessions.<SessionName>.Uri	Norādiet nosauktās sesijas URI.
Plugins.MongoDB.Sessions.<SessionName>.User	Norādiet nosauktās sesijas lietotājvārdu.
Plugins.MongoDB.System.Path	Norādiet ceļu uz spraudņa izpildāmo failu.
Plugins.MongoDB.Timeout	Iestatiet pieprasījuma izpildes noildzi.

Ņemiet vērā:

- Noklusējuma vērtības atspoguļo procesa noklusējuma iestatījumus, nevis vērtības piegādātajos konfigurācijas failos.
- Vērtības atbalsta vides mainīgos.
- Zabbix atbalsta konfigurācijas failus tikai UTF-8 kodējumā bez BOM.
- Komentāri, kas sākas ar "#", tiek atbalstīti tikai rindas sākumā.

Parametra informācija

Plugins.MongoDB.Default.AuthMechanism

Norādiet noklusējuma autentifikācijas mehānismu savienojumam ar MongoDB serveri.
 <SessionName> - definē sesijas nosaukumu izmantošanai vienumu atslēgās.

Diapazons: `SCRAM-SHA-1`, `SCRAM-SHA-256` vai `MONGODB-X509`

Plugins.MongoDB.Default.AuthSource

Norādiet noklusējuma datubāzi autentifikācijai, veidojot savienojumu ar MongoDB serveri.
 <SessionName> - definē sesijas nosaukumu izmantošanai vienumu atslēgās.

Noklusējums: `admin`

Ja tiek izmantots `MONGODB-X509` autentifikācijas mehānisms, šis parametrs automātiski tiek iestatīts uz `$external`; tā modificēšana uz jebkuru citu vērtību izraisīs kļūdu palaišanas laikā.

Plugins.MongoDB.Default.Direct

Atspējot MongoDB replica set mezglu automātisko noteikšanu.
 <SessionName> - definē sesijas nosaukumu izmantošanai vienumu atslēgās.

Noklusējums: `false`
 Diapazons: `true` - izveidot savienojumu tikai ar norādīto serveri, ignorējot citus replica set dalībniekus, `false` - automātiski noteikt visus replica set mezglus.

Ja tiek izmantota `tcp://` (novecojusi) shēma, parametrs pēc noklusējuma tiek iestatīts uz `true`.

Plugins.MongoDB.Default.Password

Norādiet noklusējuma paroli.
 <SessionName> - definējiet sesijas nosaukumu izmantošanai vienumu atslēgās.

Plugins.MongoDB.Default.ReplicaSet

Norādiet noklusējuma replica set nosaukumu, ko izmanto MongoDB replica set mezglu automātiskai atklāšanai.
 <SessionName> - definējiet sesijas nosaukumu izmantošanai vienumu atslēgās.
 Ja šis parametrs tiek izmantots kopā ar `mongodb+srv://` shēmu, tas darbosies tā, it kā būtu norādīti vairāki URL; DNS serverim var būt TXT ieraksts, kas var pārrakstīt šo (vai jebkuru citu) parametru.
 Nav pieejams, ja ir iestatīts parametrs `Plugins.MongoDB.Sessions.<SessionName>.Direct` uz `true`; aģents neizdosies startēties.

Plugins.MongoDB.Default.ReadPreference

Norādiet noklusējuma secību savienojuma izveidei ar MongoDB replikas kopas mezgliem vai atkāpšanās darbībai.
 <SessionName> - definējiet sesijas nosaukumu izmantošanai vienumu atslēgās.
 Nav pieejams, ja parametrs `Plugins.MongoDB.Sessions.<Se` ir iestatīts uz `true`; aģentu neizdosies palaist.

Diapazons: `primary`, `primaryPreferred`, `secondary`, `secondaryPreferred`, `nearest`, `PrimaryMode`, `PrimaryPreferredMode`, `SecondaryMode`, `SecondaryPreferredMode` vai `NearestMode`

Ieteicams izmantot vērtības `primary`, `primaryPreferred`, `secondary`, `secondaryPreferred` un `nearest`. Vērtības, kas beidzas ar `Mode`, tiek atbalstītas kā mantotie saderības aizstājvārdi.

Plugins.MongoDB.Default.TLSConnect

Norādiet šifrēšanas līmeni, kas nepieciešams saziņai starp Zabbix aģents 2 un uzraudzītajām datubāzēm.
 <SessionName> - definē sesijas nosaukumu, ko izmantot vienumu atslēgās.
 Atbalstīts kopš spraudņa versijas 1.2.1.

Diapazons: `required` - pieprasīt TLS savienojumu, `verify_ca` - pārbaudīt sertifikātus, `verify_full` - pārbaudīt sertifikātus un IP adresi

Plugins.MongoDB.Default.TLSCAFile

Norādiet pilnu ceļu uz failu, kurā ir augstākā līmeņa CA sertifikāti vienaudža sertifikāta verifikācijai, ko izmanto šifrētai saziņai starp Zabbix aģents 2 un uzraudzītajām datubāzēm.
 <SessionName> - definē sesijas nosaukumu izmantošanai vienumu atslēgās.

Obligāti: `yes`, ja `Plugins.MongoDB.Sessions.<SessionName>.TLSConnect` ir iestatīts uz `verify_ca` vai `verify_full`

Plugins.MongoDB.Default.TLSCertFile

Norādiet pilnu ceļu uz failu, kurā atrodas aģenta sertifikāts vai sertifikātu ķēde, kas tiek izmantota šifrētai saziņai starp Zabbix agent 2 un uzraudzītajām datubāzēm.
 <SessionName> - definējiet sesijas nosaukumu izmantošanai vienumu atslēgās.

Obligāti: `yes`, ja ir norādīts `Plugins.MongoDB.Sessions.<SessionName>.TLSKeyFile`

Plugins.MongoDB.Default.TLSKeyFile

Norādiet pilnu ceļu uz failu, kurā atrodas datubāzes privātā atslēga, kas tiek izmantota šifrētai saziņai starp Zabbix aģents 2 un uzraudzītajām datubāzēm.
 <SessionName> - definējiet sesijas nosaukumu izmantošanai vienumu atslēgās.

Obligāti: `yes`, ja ir norādīts `Plugins.MongoDB.Sessions.<SessionName>.TLSCertFile`

Plugins.MongoDB.Default.Uri

Norādiet noklusējuma URI.
 <SessionName> - definējiet sesijas nosaukumu izmantošanai vienumu atslēgās.
 Tam jāatbilst URI formātam.
 Atbalstītās shēmas: `mongodb://` un `mongodb+srv://`; shēma `tcp://` ir novecojusi un saglabāta atpakaļsavietojamībai ar esošajām konfigurācijām.
 Portu var nenorādīt (noklusējums=27017).
Piemēri: “

Noklusējums: `tcp://127.0.0.1:27017`

Piemēri:

```
mongodb://127.0.0.1:27017
mongodb+srv://example.com
localhost
```

Plugins.MongoDB.Default.User

Norādiet noklusējuma lietotājvārdu.
 <SessionName> - definējiet sesijas nosaukumu izmantošanai vienumu atslēgās.

Plugins.MongoDB.KeepAlive

Iestatiet maksimālo gaidīšanas laiku (sekundēs) pirms neizmantotie spraudņa savienojumi tiek slēgti.

Noklusējums: 300
 Diapazons: 60–900

Plugins.MongoDB.Sessions.<SessionName>.AuthMechanism

Norādiet nosauktās sesijas autentifikācijas mehānismu savienojumam ar MongoDB serveri.
 <SessionName> - definējiet sesijas nosaukumu izmantošanai vienumu atslēgās.

Diapazons: SCRAM-SHA-1, SCRAM-SHA-256 vai MONGODB-X509

Plugins.MongoDB.Sessions.<SessionName>.AuthSource

Norādiet nosauktās sesijas datubāzi, kas tiks izmantota autentifikācijai, veidojot savienojumu ar MongoDB serveri.
 <SessionName> - norādiet sesijas nosaukumu, ko izmantot vienumu atslēgās.

Noklusējums: admin

Ja tiek izmantots MONGODB-X509 autentifikācijas mehānisms, šis parametrs automātiski tiek iestatīts uz $external; tā modificēšana uz jebkuru citu vērtību izraisīs kļūdu palaišanas laikā.

Plugins.MongoDB.Sessions.<SessionName>.Direct

Atspējot MongoDB replica set mezglu automātisko noteikšanu nosauktajai sesijai.
 <SessionName> - definē sesijas nosaukumu izmantošanai vienumu atslēgās.

Noklusējums: false
 Diapazons: true - izveidot savienojumu tikai ar norādīto serveri, ignorējot citus replica set dalībniekus, false - automātiski noteikt visus replica set mezglus.

Ja tiek izmantota tcp:// (novecojusi) shēma, parametrs pēc noklusējuma tiek iestatīts uz true.

Plugins.MongoDB.Sessions.<SessionName>.Password

Norādiet nosauktās sesijas paroli.
 <SessionName> - norādiet sesijas nosaukumu, ko izmantot vienumu atslēgās.

Plugins.MongoDB.Sessions.<SessionName>.ReplicaSet

Norādiet nosauktās sesijas replica set nosaukumu, ko izmanto MongoDB replica set mezglu automātiskai atklāšanai.
 <SessionName> - definējiet sesijas nosaukumu izmantošanai vienumu atslēgās.
 Ja šis parametrs tiek izmantots kopā ar mongodb+srv:// shēmu, tas darbosies tā, it kā būtu norādīti vairāki URL; DNS serverim var būt TXT ieraksts, kas var pārrakstīt šo (vai jebkuru citu) parametru.
 Nav pieejams, ja parametrs Plugins.MongoDB.Sessions.<SessionName>.Direct ir iestatīts uz true; aģents neizdosies startēt.

Plugins.MongoDB.Sessions.<SessionName>.ReadPreference

Norādiet nosauktās sesijas secību savienojuma izveidei ar MongoDB replikas kopas mezgliem vai atkāpšanās režīmam.
 <SessionName> - definējiet sesijas nosaukumu izmantošanai vienumu atslēgās.
 Nav pieejams, ja parametrs Plugins.MongoDB.Sessions ir iestatīts uz true; aģentu neizdosies palaist.

Diapazons: primary, primaryPreferred, secondary, secondaryPreferred, nearest, PrimaryMode, PrimaryPreferredMode, SecondaryMode, SecondaryPreferredMode vai NearestMode

Ieteicams izmantot vērtības primary, primaryPreferred, secondary, secondaryPreferred un nearest. Vērtības, kas beidzas ar Mode, tiek atbalstītas kā mantoti saderības aizstājvārdi.

Plugins.MongoDB.Sessions.<SessionName>.TLSConnect

Norādiet šifrēšanas līmeni, kas nepieciešams saziņai starp Zabbix aģents 2 un uzraudzītajām datubāzēm norādītajai sesijai.
 <SessionName> - definējiet sesijas nosaukumu izmantošanai vienumu atslēgās.
 Atbalstīts kopš spraudņa versijas 1.2.1.

Diapazons: required - pieprasīt TLS savienojumu, verify_ca - pārbaudīt sertifikātus, verify_full - pārbaudīt sertifikātus un IP adresi

Plugins.MongoDB.Sessions.<SessionName>.TLSCAFile

Norādiet pilnu ceļu uz failu, kurā ir augstākā līmeņa CA sertifikāti vienaudža sertifikāta verifikācijai, kas tiek izmantota šifrētai saziņai starp Zabbix aģents 2 un uzraudzītajām datubāzēm norādītajai sesijai.
 <SessionName> - definējiet sesijas nosaukumu izmantošanai vienumu atslēgās.

Obligāti: yes, ja Plugins.MongoDB.Sessions.<SessionName>.TLSConnect ir iestatīts uz verify_ca vai verify_full

Plugins.MongoDB.Sessions.<SessionName>.TLSCertFile

Norādiet pilnu faila ceļu uz failu, kas satur aģenta sertifikātu vai sertifikātu ķēdi, ko izmanto šifrētai saziņai starp Zabbix aģents 2 un uzraudzītajām datubāzēm norādītajai sesijai.
 <SessionName> - definē sesijas nosaukumu, ko izmantot vienumu atslēgās.

Obligāti: yes, ja ir norādīts Plugins.MongoDB.Sessions.<SessionName>.TLSKeyFile

Plugins.MongoDB.Sessions.<SessionName>.TLSKeyFile

Norādiet pilnu ceļu uz failu, kurā atrodas datubāzes privātā atslēga, kas tiek izmantota šifrētai saziņai starp Zabbix aģents 2 un uzraudzītajām datubāzēm norādītajai sesijai.
 <SessionName> - definējiet sesijas nosaukumu, ko izmantot vienumu atslēgās.

Obligāti: yes, ja ir norādīts `Plugins.MongoDB.Sessions.<SessionName>.TLSCertFile`

Plugins.MongoDB.Sessions.<SessionName>.Uri

Norādiet nosauktās sesijas URI.
 <SessionName> - definējiet sesijas nosaukumu izmantošanai vienumu atslēgās.
 Tam jāatbilst URI formātam.
 Atbalstītās shēmas: `mongodb://` un `mongodb+srv://`; `tcp://` shēma ir novecojusi un saglabāta atpakaļsavietojamībai ar esošajām konfigurācijām.
 Portu var nenorādīt (noklusējums=27017).
Piemēri: “

Noklusējums: `tcp://127.0.0.1:27017`

Piemēri:

```
mongodb://127.0.0.1:27017
mongodb+srv://example.com
localhost
```

Plugins.MongoDB.Sessions.<SessionName>.User

Norādiet nosauktās sesijas lietotājvārdu.
 <SessionName> - definējiet sesijas nosaukumu izmantošanai vienumu atslēgās.

Plugins.MongoDB.System.Path

Norādiet MongoDB spraudņa izpildāmā faila ceļu.

Lietošanas piemērs: `Plugins.MongoDB.System.Path=/usr/libexec/zabbix/zabbix-agent2-plugin-mongodb`

Plugins.MongoDB.Timeout

Iestatiet pieprasījuma izpildes noildzi (ilgumu sekundēs, cik ilgi gaidīt, līdz pieprasījums tiks pabeigts, pirms to pārtrauc).

Noklusējums: `global timeout`
 Diapazons: 1-30

Opcijas

Parameter	Description
-V --version	Izdrukā spraudņa versiju un licences informāciju.
-h --help	Izdrukā palīdzības informāciju (saīsināti).
-t, --test <item key>	Palaiž spraudni testēšanai (spraudņa konfigurācija tiek ignorēta).

Skatiet arī

- Vispārīgo Zabbix aģenta 2 konfigurācijas parametru apraksts: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Norādījumi spraudņu konfigurēšanai

MQTT spraudnis

Pārskats

Konfigurācijas parametri ļauj pielāgot MQTT spraudņa darbību.

Ieteicams tos norādīt atsevišķā konfigurācijas failā (piem., `mqtt.conf`), nevis aģents 2 konfigurācijas failā. Pēc tam izmantojiet Include direktīvu, lai iekļautu šo failu Zabbix aģents 2 konfigurācijā.

Visi parametri ir **neobligāti**, ja vien nav skaidri norādīts, ka parametrs ir obligāts.

Šī lapa ir strukturēta šādi:

- Parametru pārskats (lai skatītu pilnu parametra informāciju, noklikšķiniet uz tā nosaukuma)
- Parametru detaļas

Parameter	Description
Plugins.MQTT.Default.Password	Norādiet noklusējuma paroli savienojumam ar MQTT.
Plugins.MQTT.Default.TLSCAFile	Norādiet pilnu ceļu uz failu, kurā ir augstākā līmeņa CA sertifikāti vienaudža sertifikāta verifikācijai.
Plugins.MQTT.Default.TLSCertFile	Norādiet pilnu ceļu uz failu, kurā ir aģents sertifikāts vai sertifikātu ķēde.
Plugins.MQTT.Default.TLSKeyFile	Norādiet pilnu ceļu uz failu, kurā ir MQTT privātā atslēga.

Parameter	Description
Plugins.MQTT.Default.Topic	Norādiet noklusējuma tēmu MQTT abonēšanai.
Plugins.MQTT.Default.Url	Norādiet noklusējuma MQTT starpniekservera savienojuma virkni.
Plugins.MQTT.Default.User	Norādiet noklusējuma lietotājvārdu savienojumam ar MQTT.
Plugins.MQTT.Sessions.<Session name>.Password	Norādiet nosauktās sesijas paroli savienojumam ar MQTT.
Plugins.MQTT.Sessions.<Session name>.TLSCAFile	Norādiet pilnu ceļu uz failu, kurā ir augstākā līmeņa CA sertifikāti vienaudža sertifikāta verifikācijai nosauktajai sesijai.
Plugins.MQTT.Sessions.<Session name>.TLSCertFile	Norādiet pilnu ceļu uz failu, kurā ir aģents sertifikāts vai sertifikātu ķēde nosauktajai sesijai.
Plugins.MQTT.Sessions.<Session name>.TLSKeyFile	Norādiet pilnu ceļu uz failu, kurā ir MQTT privātā atslēga nosauktajai sesijai.
Plugins.MQTT.Sessions.<Session name>.Topic	Norādiet nosauktās sesijas tēmu MQTT abonēšanai.
Plugins.MQTT.Sessions.<Session name>.Url	Norādiet nosauktās sesijas MQTT starpniekservera savienojuma virkni.
Plugins.MQTT.Sessions.<Session name>.User	Norādiet nosauktās sesijas lietotājvārdu savienojumam ar MQTT.

Ņemiet vērā:

- Noklusējuma vērtības atspoguļo procesa noklusējumus, nevis vērtības piegādātajos konfigurācijas failos;
- Vērtības atbalsta vides mainīgos;
- Zabbix atbalsta konfigurācijas failus tikai UTF-8 kodējumā bez BOM;
- Komentāri, kas sākas ar ”#”, tiek atbalstīti tikai rindas sākumā.

Parametra informācija

Plugins.MQTT.Default.Password

Norādiet noklusējuma paroli savienojumam ar MQTT.
 Tiek izmantota, ja vienums atslēgā vai nosauktajā sesijā nav norādīta vērtība.

Plugins.MQTT.Default.TLSCAFile

Norādiet pilnu ceļu uz failu, kurā ir augstākā līmeņa CA sertifikāti, kas paredzēti vienaudža sertifikāta pārbaudei šifrētai saziņai starp Zabbix aģentu 2 un MQTT brokeri.
 Tiek izmantots, ja nosauktajā sesijā nav norādīta vērtība.

Plugins.MQTT.Default.TLSCertFile

Norādiet pilnu ceļu uz failu, kurā atrodas aģenta sertifikāts vai sertifikātu ķēde šifrētai saziņai starp Zabbix aģentu 2 un MQTT brokeri.
 Tiek izmantots, ja nosauktajā sesijā nav norādīta vērtība.

Plugins.MQTT.Default.TLSKeyFile

Norādiet pilnu ceļu uz failu, kurā atrodas MQTT privātā atslēga šifrētai saziņai starp Zabbix aģentu 2 un MQTT brokeri.
 Tiek izmantots, ja nosauktajā sesijā nav norādīta neviena vērtība.

Plugins.MQTT.Default.Topic

Norādiet MQTT abonēšanas noklusējuma tēmu.
 Tiek izmantota, ja vienumā atslēgā vai nosauktajā sesijā nav norādīta vērtība.
 Tēma var saturēt aizstājējzīmes (”+”,”#”).

Piemēri:

```
path/to/file
path/to/#
path/+/topic
```

Plugins.MQTT.Default.Url

Norādiet noklusējuma MQTT brokera savienojuma virkni.
 Tiek izmantota, ja vienumā vai nosauktā sesijā nav norādīta neviena vērtība.
 Nedrīkst ietvert vaicājuma parametrus.
 Jāatbilst URL formātam.
 Atbalstītās shēmas: `tcp` (noklusējums), `ws`, `tls`
 Shēmu var nenorādīt.
 Portu var nenorādīt (noklusējums=1883)

Noklusējums: `tcp://localhost:1883`

Piemēri:

```
tcp://host:1883
localhost
ws://host:8080
```

Plugins.MQTT.Default.User

Norādiet noklusējuma lietotājvārdu savienojumam ar MQTT.
 Tiek izmantots, ja vienuma atslēgā vai nosauktā sesijā nav norādīta vērtība.

Plugins.MQTT.Sessions.<SessionName>.Password

Norādiet nosauktās sesijas paroli savienojumam ar MQTT.
 <SessionName> - definējiet sesijas nosaukumu, ko izmantot vienumu atslēgās.

Plugins.MQTT.Sessions.<SessionName>.TLSCAFile

Norādiet pilnu ceļu uz failu, kurā ir augstākā līmeņa CA sertifikāti, kas paredzēti vienaudža sertifikāta pārbaudei norādītajai sesijai un tiek izmantoti šifrētai saziņai starp Zabbix aģentu 2 un MQTT brokeri.
 <SessionName> - definējiet sesijas nosaukumu, ko izmantot vienumu atslēgās.

Plugins.MQTT.Sessions.<SessionName>.TLSCertFile

Norādiet pilnu ceļu uz failu, kurā ir aģenta sertifikāts vai sertifikātu ķēde norādītajai sesijai, kas tiek izmantota šifrētai saziņai starp Zabbix aģents 2 un MQTT brokeri.
 <SessionName> - definējiet sesijas nosaukumu, ko izmantot vienumu atslēgās.

Plugins.MQTT.Sessions.<SessionName>.TLSKeyFile

Norādiet pilnu ceļu uz failu, kurā atrodas MQTT privātā atslēga norādītajai sesijai, ko izmanto šifrētai saziņai starp Zabbix aģents 2 un MQTT brokeri.
 <SessionName> - norādiet sesijas nosaukumu, ko izmantot vienumu atslēgās.

Plugins.MQTT.Sessions.<SessionName>.Topic

Norādiet nosauktās sesijas tēmu MQTT abonēšanai.
 <SessionName> - definējiet sesijas nosaukumu, ko izmantot vienumu atslēgās.
 Tēma var saturēt aizstājējzīmes (”+”,”#”).

Piemēri:

```
path/to/file
path/to/#
path/+/topic
```

Plugins.MQTT.Sessions.<SessionName>.Url

Norādiet nosauktās sesijas MQTT brokera savienojuma virkni.
 <SessionName> - definējiet sesijas nosaukumu, ko izmantot vienumu atslēgās.
 Nedrīkst ietvert vaicājuma parametrus.
 Jāatbilst URL formātam.
 Atbalstītās shēmas: `tcp` (noklusējums), `ws`, `tls`
 Shēmu var nenorādīt.
 Portu var nenorādīt (noklusējums=1883)

Noklusējums: `tcp://localhost:1883`

Piemēri:

```
tcp://host:1883
localhost
ws://host:8080
```

Plugins.MQTT.Sessions.<SessionName>.User

Norādiet nosauktās sesijas lietotājvārdu savienojumam ar MQTT.
 <SessionName> - definējiet sesijas nosaukumu, ko izmantot vienumu atslēgās.

Ja nepieciešams iestatīt pieprasījuma izpildes noildzi (cik ilgi gaidīt, līdz pieprasījums tiks pabeigts, pirms to aptur), izmantojiet vienuma konfigurācijas formu.

Skatiet arī

- Vispārīgo Zabbix aģents 2 konfigurācijas parametru apraksts: Zabbix aģents 2 (UNIX) / Zabbix aģents 2 (Windows)
- Norādījumi par spraudņu konfigurēšanu

MSSQL spraudnis

Pārskats

Konfigurācijas parametri ļauj pielāgot MSSQL spraudņa darbību.

Ieteicams tos norādīt atsevišķā konfigurācijas failā (piem., `mssql.conf`), nevis aģents 2 konfigurācijas failā. Pēc tam izmantojiet direktīvu Include, lai iekļautu šo failu Zabbix aģents 2 konfigurācijā.

MSSQL spraudnis ir ielādējams spraudnis, un tas ir pieejams un pilnībā aprakstīts MSSQL spraudņa repozitorijā.

Visi parametri ir **neobligāti**, ja vien nav skaidri norādīts, ka parametrs ir obligāts.

Šī lapa ir strukturēta šādi:

- Parametru pārskats (lai skatītu pilnu parametra informāciju, noklikšķiniet uz tā nosaukuma)

- Parametru detaļas

Parameter	Description
Plugins.MSSQL.CustomQueriesDir	Norādiet ceļu uz direktoriju, kurā atrodas lietotāja definētie `.sql` faili ar pielāgotajiem vaicājumiem, ko spraudnis var izpildīt.
Plugins.MSSQL.CustomQueriesEnabled	Iespējot `mssql.custom.query` vienuma atslēgas izpildi.
Plugins.MSSQL.Default.CACertPath	Norādiet noklusējuma faila ceļu uz sertifikātu iestādes (CA) publiskās atslēgas sertifikātu, kas izdevusi MSSQL servera sertifikātu.
Plugins.MSSQL.Default.Database	Norādiet noklusējuma datubāzes nosaukumu.
Plugins.MSSQL.Default.Encrypt	Norādiet noklusējuma šifrēšanas līmeni.
Plugins.MSSQL.Default.HostNameInCertificate	Norādiet MSSQL servera sertifikāta vispārīgo nosaukumu (CN) pēc noklusējuma.
Plugins.MSSQL.Default.Password	Norādiet noklusējuma paroli.
Plugins.MSSQL.Default.TLSMinVersion	Norādiet noklusējuma minimālo TLS versiju.
Plugins.MSSQL.Default.TrustServerCertificate	Pēc noklusējuma uzticieties servera sertifikātam, to nepārbaudot.
Plugins.MSSQL.Default.Uri	Norādiet noklusējuma URI.
Plugins.MSSQL.Default.User	Norādiet noklusējuma lietotājvārdu.
Plugins.MSSQL.KeepAlive	Norādiet maksimālo gaidīšanas laiku (sekundēs) pirms neizmantoto spraudņa savienojumu aizvēršanas.
Plugins.MSSQL.Sessions.<SessionName>.CACertPath	Norādiet noklusējuma faila ceļu uz sertifikātu iestādes (CA) publiskās atslēgas sertifikātu, kas izdevusi MSSQL servera sertifikātu.
Plugins.MSSQL.Sessions.<SessionName>.Database	Norādiet noklusējuma datubāzes nosaukumu.
Plugins.MSSQL.Sessions.<SessionName>.Encrypt	Norādiet noklusējuma šifrēšanas līmeni.
Plugins.MSSQL.Sessions.<SessionName>.HostNameInCertificate	Norādiet MSSQL servera sertifikāta vispārīgo nosaukumu (CN) nosauktajai sesijai.
Plugins.MSSQL.Sessions.<SessionName>.Password	Norādiet noklusējuma paroli.
Plugins.MSSQL.Sessions.<SessionName>.TLSMinVersion	Norādiet noklusējuma minimālo TLS versiju.
Plugins.MSSQL.Sessions.<SessionName>.TrustServerCertificate	Uzticieties nosauktās sesijas serverim sertificēta sertifikātam, to nepārbaudot.
Plugins.MSSQL.Sessions.<SessionName>.Uri	Norādiet nosauktās sesijas URI.
Plugins.MSSQL.Sessions.<SessionName>.User	Norādiet nosauktās sesijas lietotājvārdu.
Plugins.MSSQL.System.Path	Norādiet ceļu uz MSSQL spraudņa izpildāmo failu.
Plugins.MSSQL.Timeout	Iestatiet ilgumu sekundēs, cik ilgi gaidīt servera atbildi, veidojot pirmo savienojumu un veicot turpmākās darbības sesijā.

Ņemiet vērā:

- Noklusējuma vērtības atspoguļo procesa noklusējumus, nevis vērtības piegādātajos konfigurācijas failos.
- Vērtības atbalsta vides mainīgos;
- Zabbix atbalsta konfigurācijas failus tikai UTF-8 kodējumā bez BOM.
- Komentāri, kas sākas ar "#", tiek atbalstīti tikai rindas sākumā.

Parametra informācija

Plugins.MSSQL.CustomQueriesDir

Norādiet ceļu uz direktoriju, kurā atrodas lietotāja definētie `.sql` faili ar pielāgotajiem vaicājumiem, ko spraudnis var izpildīt.
 Spraudnis startēšanas laikā ielādē visus pieejamos .sql failus konfigurētajā direktorijā. Tas nozīmē, ka jebkuras izmaiņas pielāgoto vaicājumu failos netiks atspoguļotas, līdz spraudnis tiks restartēts. Spraudnis tiek palaists un apturēts kopā ar Zabbix aģents 2.

Noklusējums:
`/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)

Plugins.MSSQL.CustomQueriesEnabled

Iespējot `mssql.custom.query` vienuma atslēgas izpildi.
 Ja ir atspējots, vaicājumi netiks ielādēti no pielāgoto vaicājumu direktorijas ceļa.

Noklusējums: `false`
 Diapazons: `true/false`

Plugins.MSSQL.Default.CACertPath

Norādiet noklusējuma faila ceļu uz sertifikātu iestādes (CA) publiskās atslēgas sertifikātu, kas izsniedza MSSQL servera sertifikātu.
 Sertifikātam jābūt PEM formātā.

Plugins.MSSQL.Default.Database

Norādiet noklusējuma datubāzes nosaukumu.

Plugins.MSSQL.Default.Encrypt

Norādiet noklusējuma šifrēšanas līmeni.

Diapazons:
true - datu sūtīšana starp spraudni un serveri ir šifrēta;
false - datu sūtīšana starp spraudni un serveri nav šifrēta, izņemot pieteikšanās pakotni;
strict - datu sūtīšana starp spraudni un serveri ir šifrēta no gala līdz galam, izmantojot TDS8;
disable - datu sūtīšana starp spraudni un serveri nav šifrēta.

Plugins.MSSQL.Default.HostNameInCertificate

Pēc noklusējuma norādiet MSSQL servera sertifikāta vispārīgo nosaukumu (CN).

Plugins.MSSQL.Default.Password

Norādiet aizsargāta MSSQL servera noklusējuma paroli.

Plugins.MSSQL.Default.TLSMinVersion

Pēc noklusējuma norādiet minimālo TLS versiju.

Diapazons: `1.0`, `1.1`, `1.2`, `1.3`

Plugins.MSSQL.Default.TrustServerCertificate

Pēc noklusējuma uzticēties servera sertifikātam, to nevalidējot.

Diapazons: `true`, `false`

Plugins.MSSQL.Default.Uri

Norādiet noklusējuma URI savienojumam ar MSSQL serveri.
 Iegultie akreditācijas dati tiks ignorēti.
 Vienīgā atbalstītā shēma ir `sqlserver://`.
 Shēmu var arī nenorādīt.

Noklusējums: `sqlserver://localhost:1433`

Plugins.MSSQL.Default.User

Norādiet noklusējuma lietotājvārdu aizsargātam MSSQL serverim.

Plugins.MSSQL.KeepAlive

Norādiet maksimālo gaidīšanas laiku (sekundēs) pirms neizmantotie spraudņa savienojumi tiek aizvērti.

Noklusējums: 300
 Diapazons: 60–900

Plugins.MSSQL.Sessions.<SessionName>.CACertPath

Norādiet nosauktās sesijas faila ceļu uz sertifikāta iestādes (CA) publiskās atslēgas sertifikātu, kas izsniedza MSSQL servera sertifikātu.
 <SessionName> - definējiet sesijas nosaukumu, ko izmantot vienumu atslēgās.
 Sertifikātam jābūt PEM formātā.

Plugins.MSSQL.Sessions.<SessionName>.Database

Norādiet nosauktās sesijas datubāzes nosaukumu.
 <SessionName> - definējiet sesijas nosaukumu, ko izmantot vienumu atslēgās.

Plugins.MSSQL.Sessions.<SessionName>.Encrypt

Norādiet nosauktās sesijas šifrēšanas līmeni.
 <SessionName> - definē sesijas nosaukumu izmantošanai vienumu atslēgās.

Diapazons:
true - datu sūtīšana starp spraudni un serveri ir šifrēta;
false - datu sūtīšana starp spraudni un serveri nav šifrēta tālāk par pieteikšanās pakotni;
strict - datu sūtīšana starp spraudni un serveri ir šifrēta E2E, izmantojot TDS8;
disable - datu sūtīšana starp spraudni un serveri nav šifrēta.

Plugins.MSSQL.Sessions.<SessionName>.HostNameInCertificate

Norādiet MSSQL servera sertifikāta kopējo nosaukumu (CN) nosauktajai sesijai.
 <SessionName> - definējiet sesijas nosaukumu, ko izmantot vienumu atslēgās.

Plugins.MSSQL.Sessions.<SessionName>.Password

Norādiet nosauktās sesijas paroli aizsargātam MSSQL serverim.
 <SessionName> - norādiet sesijas nosaukumu, ko izmantot vienumu atslēgās.

Plugins.MSSQL.Sessions.<SessionName>.TLSMinVersion

Norādiet nosauktās sesijas minimālo TLS versiju.
 <SessionName> - definē sesijas nosaukumu, ko izmantot vienumu atslēgās.

Diapazons: `1.0`, `1.1`, `1.2`, `1.3`

Plugins.MSSQL.Sessions.<SessionName>.TrustServerCertificate

Uzticēties norādītās sesijas servera sertifikātam, to nevalidējot.
 <SessionName> - norādiet sesijas nosaukumu, ko izmantot vienumu atslēgās.

Diapazons: `true`, `false`

Plugins.MSSQL.Sessions.<SessionName>.Uri

Norādiet nosauktās sesijas URI savienojumam ar MSSQL serveri.
 <SessionName> - definējiet sesijas nosaukumu, ko izmantot vienumu atslēgās.
 Iegultie akreditācijas dati tiks ignorēti.
 Vienīgā atbalstītā shēma ir `sqlserver://`.
 Shēmu var nenorādīt.

Noklusējums: `sqlserver://localhost:1433`

Plugins.MSSQL.Sessions.<SessionName>.User

Norādiet nosauktās sesijas lietotājvārdu aizsargātam MSSQL serverim.
 <SessionName> - definējiet sesijas nosaukumu, ko izmantot vienumu atslēgās.

Plugins.MSSQL.System.Path

Norādiet MSSQL spraudņa izpildāmā faila ceļu.
 Globāls MSSQL spraudņa iestatījums. Tiek piemērots visiem savienojumiem.

Piemērs: `Plugins.MSSQL.System.Path=/usr/libexec/zabbix/zabbix-agent2-plugin-mssql`

Plugins.MSSQL.Timeout

Iestatiet ilgumu sekundēs, cik ilgi gaidīt, līdz serveris atbild, veidojot sākotnējo savienojumu un veicot turpmākās darbības sesijā.

Noklusējums: `global timeout`
 Diapazons: 1–30

Opcijas

Parameter	Description
-V --version	Izdrukā spraudņa versiju un licences informāciju.
-h --help	Izdrukā palīdzības informāciju (saīsināti).
-t, --test <item key>	Palaiž spraudni testēšanai (spraudņa konfigurācija tiek ignorēta).

Skatiet arī

- Vispārīgo Zabbix aģenta 2 konfigurācijas parametru apraksts: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Norādījumi spraudņu konfigurēšanai

MySQL spraudnis

Pārskats

Konfigurācijas parametri ļauj pielāgot MySQL spraudņa darbību.

Ieteicams tos norādīt atsevišķā konfigurācijas failā (piem., `mysql.conf`), nevis aģents 2 konfigurācijas failā. Pēc tam izmantojiet direktīvu Include, lai iekļautu šo failu Zabbix aģents 2 konfigurācijā.

Visi parametri ir **neobligāti**, ja vien nav skaidri norādīts, ka parametrs ir obligāts.

Šī lapa ir strukturēta šādi:

- Parametru pārskats (lai skatītu pilnu parametra informāciju, noklikšķiniet uz tā nosaukuma)
- Parametru detalizēta informācija

Parameter	Description
Plugins.Mysql.CallTimeout	Norādiet maksimālo gaidīšanas laiku sekundēs, līdz pieprasījums tiek izpildīts.
Plugins.Mysql.CustomQueriesPath	Norādiet pilnu direktorijas ceļu, kurā atrodas `.sql` faili ar pielāgotiem vaicājumiem.
Plugins.Mysql.CustomQueriesEnabled	Iespējojiet `mysql.custom.query` vienums atslēgas izpildi.
Plugins.Mysql.Default.Password	Norādiet noklusējuma paroli.
Plugins.Mysql.Default.TLSConnect	Norādiet noklusējuma šifrēšanas līmeni saziņai starp Zabbix aģents 2 un uzraudzītajām datubāzēm.
Plugins.Mysql.Default.TLSCAFile	Norādiet pilnu faila ceļu, kurā atrodas augstākā līmeņa CA sertifikāti vienaudža sertifikāta verifikācijai.
Plugins.Mysql.Default.TLSCertFile	Norādiet pilnu faila ceļu, kurā atrodas aģents sertifikāts vai sertifikātu ķēde.
Plugins.Mysql.Default.TLSKeyFile	Norādiet pilnu faila ceļu, kurā atrodas datubāzes privātā atslēga.
Plugins.Mysql.Default.Uri	Norādiet noklusējuma URI.
Plugins.Mysql.Default.User	Norādiet noklusējuma lietotājvārdu.

Parameter	Description
Plugins.Mysql.KeepAlive	Norādiet maksimālo gaidīšanas laiku sekundēs pirms neizmantoto spraudņa savienojumu aizvēršanas.
Plugins.Mysql.Sessions.<SessionName>.Password	Norādiet nosauktās sesijas paroli.
Plugins.Mysql.Sessions.<SessionName>.TLSConnect	Norādiet nosauktās sesijas šifrēšanas līmeni saziņai starp Zabbix aģents 2 un uzraudzītajām datubāzēm.
Plugins.Mysql.Sessions.<SessionName>.TLSCAFile	Norādiet pilnu ceļu, kurā atrodas augstākā līmeņa CA sertifikāti nosauktās sesijas vienaudža sertifikāta verifikācijai.
Plugins.Mysql.Sessions.<SessionName>.TLSCertFile	Norādiet pilnu ceļu, kurā atrodas nosauktās sesijas aģents sertifikāts vai sertifikātu ķēde.
Plugins.Mysql.Sessions.<SessionName>.TLSKeyFile	Norādiet pilnu ceļu, kurā atrodas nosauktās sesijas datubāzes privātā atslēga.
Plugins.Mysql.Sessions.<SessionName>.Uri	Norādiet nosauktās sesijas URI.
Plugins.Mysql.Sessions.<SessionName>.User	Norādiet nosauktās sesijas lietotājvārdu.
Plugins.Mysql.Timeout	Iestatiet maksimālo gaidīšanas laiku sekundēs, kad ir jāizveido savienojums.

Ņemiet vērā, ka:

- Noklusējuma vērtības atspoguļo procesa noklusējuma iestatījumus, nevis vērtības piegādātajos konfigurācijas failos;
- Vērtības atbalsta vides mainīgos;
- Zabbix atbalsta konfigurācijas failus tikai UTF-8 kodējumā bez BOM;
- Komentāri, kas sākas ar "#", tiek atbalstīti tikai rindas sākumā.

Parametra informācija

Plugins.Mysql.CallTimeout

Norādiet maksimālo laiku sekundēs, cik ilgi gaidīt, līdz pieprasījums tiks izpildīts.

Noklusējums: `global timeout`
 Diapazons: 1-30

Plugins.Mysql.CustomQueriesPath

Norādiet pilnu ceļu uz direktoriju, kurā atrodas `.sql` faili ar pielāgotajiem vaicājumiem.

Noklusējums:
`/usr/local/share/zabbix/custom-queries/mysql` - Unix sistēmām,
`*:\Program Files\Zabbix Agent 2\Custom Queries\Mysql` - Windows sistēmām (kur `*` ir diska nosaukums no vides mainīgā `ProgramFiles`)

Plugins.Mysql.CustomQueriesEnabled

Iespējot `mysql.custom.query` vienuma atslēgas izpildi.
 Ja ir atspējots, vaicājumi netiks ielādēti no pielāgoto vaicājumu direktorijas ceļa.

Noklusējums: `false`
 Diapazons: `true/false`

Plugins.Mysql.Default.Password

Norādiet noklusējuma paroli aizsargātam MySQL serverim.

Plugins.Mysql.Default.TLSConnect

Norādiet noklusējuma šifrēšanas līmeni saziņai starp Zabbix aģentu 2 un uzraudzītajām datubāzēm.

Diapazons:
required - pieprasīt TLS savienojumu;
verify_ca - pārbaudīt sertifikātus;
verify_full - pārbaudīt sertifikātus un IP adresi

Plugins.Mysql.Default.TLSCAFile

Norādiet pilnu ceļu uz failu, kurā ir augstākā līmeņa CA sertifikāti, kas paredzēti vienaudža sertifikāta verifikācijai un tiek izmantoti šifrētai saziņai starp Zabbix aģents 2 un uzraudzītajām datubāzēm.

Obligāti: `yes`, ja `Plugins.Mysql.Default.TLSConnect` ir iestatīts uz `verify_ca` vai `verify_full`

Plugins.Mysql.Default.TLSCertFile

Norādiet pilnu ceļu uz failu, kurā atrodas aģenta sertifikāts vai sertifikātu ķēde, ko izmanto šifrētai saziņai starp Zabbix aģentu 2 un uzraudzītajām datubāzēm.

Obligāti: `yes`, ja ir norādīts `Plugins.Mysql.Default.TLSKeyFile`

Plugins.Mysql.Default.TLSKeyFile

Norādiet pilnu faila ceļu uz failu, kurā atrodas datubāzes privātā atslēga, kas tiek izmantota šifrētai saziņai starp Zabbix aģentu 2 un uzraudzītajām datubāzēm.

Obligāti: `yes`, ja ir norādīts `Plugins.Mysql.Default.TLSCertFile`

Plugins.Mysql.Default.Uri

Norādiet noklusējuma URI savienojumam ar MySQL serveri.
 Iegultie akreditācijas dati tiks ignorēti.
 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

Norādiet noklusējuma lietotājvārdu aizsargātam MySQL serverim.

Plugins.Mysql.KeepAlive

Norādiet maksimālo gaidīšanas laiku (sekundēs) pirms neizmantotie spraudņa savienojumi tiek slēgti.

Noklusējums: 300
 Diapazons: 60-900

Plugins.Mysql.Sessions.<SessionName>.Password

Norādiet nosauktās sesijas paroli aizsargātam MySQL serverim.
 <SessionName> - norādiet sesijas nosaukumu, ko izmantot vienumu atslēgās.

Plugins.Mysql.Sessions.<SessionName>.TLSConnect

Norādiet nosauktās sesijas šifrēšanas līmeni saziņai starp Zabbix aģentu 2 un uzraudzītajām datubāzēm.
 <SessionName> - definē sesijas nosaukumu, ko izmantot vienumu atslēgās.

Diapazons:
required - pieprasa TLS savienojumu;
verify_ca - pārbauda sertifikātus;
verify_full - pārbauda sertifikātus un IP adresi

Plugins.Mysql.Sessions.<SessionName>.TLSCAFile

Norādiet pilnu ceļu uz failu, kurā ir augstākā līmeņa CA sertifikāti nosauktās sesijas vienaudža sertifikāta verifikācijai, ko izmanto šifrētai saziņai starp Zabbix aģents 2 un uzraudzītajām datubāzēm.
 <SessionName> - definē sesijas nosaukumu izmantošanai vienumu atslēgās.

Obligāti: `yes`, ja `Plugins.Mysql.Sessions.<SessionName>.TLSConnect` ir iestatīts uz `verify_ca` vai `verify_full`

Plugins.Mysql.Sessions.<SessionName>.TLSCertFile

Norādiet pilnu faila ceļu uz failu, kurā atrodas nosauktās sesijas aģenta sertifikāts vai sertifikātu ķēde, kas tiek izmantota šifrētai saziņai starp Zabbix aģents 2 un uzraudzītajām datubāzēm.
 <SessionName> - definējiet sesijas nosaukumu, ko izmantot vienumu atslēgās.

Obligāti: `yes`, ja ir norādīts `Plugins.Mysql.Sessions.<SessionName>.TLSKeyFile`

Plugins.Mysql.Sessions.<SessionName>.TLSKeyFile

Norādiet pilnu ceļu uz failu, kurā atrodas nosauktās sesijas datubāzes privātā atslēga, kas tiek izmantota šifrētai saziņai starp Zabbix aģents 2 un uzraudzītajām datubāzēm.
 <SessionName> - definē sesijas nosaukumu, ko izmantot vienuma atslēgās.

Obligāts: `yes`, ja ir norādīts `Plugins.Mysql.Sessions.<SessionName>.TLSCertFile`

Plugins.Mysql.Sessions.<SessionName>.Uri

Norādiet nosauktās sesijas URI, lai izveidotu savienojumu ar MySQL serveri.
 <SessionName> - definējiet sesijas nosaukumu, ko izmantot vienumu atslēgās.
 Iegultie akreditācijas dati tiks ignorēti.
 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

Norādiet nosauktās sesijas lietotājvārdu aizsargātam MySQL serverim.
 <SessionName> - definējiet sesijas nosaukumu, ko izmantot vienumu atslēgās.

Plugins.Mysql.Timeout

Iestatiet maksimālo gaidīšanas laiku sekundēs, kad jāizveido savienojums.

Noklusējums: `global timeout<br>` Diapazons: 1–30

Skatīt arī

- Vispārīgo Zabbix aģents 2 konfigurācijas parametru apraksts: Zabbix aģents 2 (UNIX) / Zabbix aģents 2 (Windows)
- Norādījumi par sprauduņu konfigurēšanu

NVIDIA GPU spraudnis

Pārskats

Konfigurācijas parametri ļauj pielāgot NVIDIA GPU sprauduņa darbību.

Ieteicams tos norādīt atsevišķā konfigurācijas failā (piem., `nvidiagpu.conf`), nevis aģents 2 konfigurācijas failā. Pēc tam izmantojiet direktīvu Include, lai iekļautu šo failu Zabbix aģents 2 konfigurācijā.

NVIDIA GPU spraudnis ir ielādējams spraudnis, un tas ir pieejams un pilnībā aprakstīts NVIDIA GPU sprauduņa repozitorijā.

Visi parametri ir **neobligāti**, ja vien nav skaidri norādīts, ka parametrs ir obligāts.

Šī lapa ir strukturēta šādi:

- Parametru pārskats (lai skatītu pilnu parametra informāciju, noklikšķiniet uz tā nosaukuma)
- Parametru detaļas

Parameter	Description
Plugins.NVIDIA.System.Path	Norādiet NVIDIA GPU sprauduņa izpildāmā faila ceļu.
Plugins.NVIDIA.Timeout	Norādiet maksimālo laiku (sekundēs), cik ilgi gaidīt servera atbildi savienojuma mēģinājumu un turpmāko darbību laikā sesijā.

Ņemiet vērā:

- Noklusējuma vērtības atspoguļo procesa noklusējuma iestatījumus, nevis vērtības piegādātajos konfigurācijas failos;
- Vērtības atbalsta vides mainīgos;
- Zabbix atbalsta konfigurācijas failus tikai UTF-8 kodējumā bez BOM;
- Komentāri, kas sākas ar "#", tiek atbalstīti tikai rindas sākumā.

Parametra informācija

Plugins.NVIDIA.System.Path

Norādiet NVIDIA GPU sprauduņa izpildāmā faila ceļu.

Piemērs:

```
Plugins.NVIDIA.System.Path=/path/to/executable/nvidia
```

Plugins.NVIDIA.Timeout

Norādiet maksimālo laiku (sekundēs), cik ilgi gaidīt servera atbildi savienojuma mēģinājumu un turpmāko darbību laikā sesijā.
 Globālais vienuma tipa noildze vai atsevišķa vienuma noildze pārrakstīs šo vērtību, ja tā būs lielāka.

Noklusējums: `global timeout<br>` Diapazons: 1–30

Skatīt arī

- Vispārīgo Zabbix aģents 2 konfigurācijas parametru apraksts: Zabbix aģents 2 (UNIX) / Zabbix aģents 2 (Windows)
- Norādījumi sprauduņu konfigurēšanai

Oracle spraudnis

Pārskats

Konfigurācijas parametri ļauj pielāgot Oracle sprauduņa darbību.

Ieteicams tos norādīt atsevišķā konfigurācijas failā (piem., `oracle.conf`), nevis aģents 2 konfigurācijas failā. Pēc tam izmantojiet direktīvu Include, lai iekļautu šo failu Zabbix aģents 2 konfigurācijā.

Visi parametri ir **neobligāti**, ja vien nav skaidri norādīts, ka parametrs ir obligāts.

Šī lapa ir strukturēta šādi:

- Parametru pārskats (lai skatītu pilnu parametra informāciju, noklikšķiniet uz tā nosaukuma)
- Parametru detalizēta informācija

Parameter	Description
Plugins.Oracle.CallTimeout	Maksimālais gaidīšanas laiks sekundēs, līdz pieprasījums tiek izpildīts. Šim parametram pašlaik nav ietekmes; tā vietā tiek izmantots vienuma taimauts iestatījums.
Plugins.Oracle.ConnectTimeout	Norādiet maksimālo gaidīšanas laiku sekundēs, līdz tiek izveidots savienojums.
Plugins.Oracle.CustomQueriesPath	Norādiet pilnu direktorija ceļu, kurā atrodas `.sql` faili ar pielāgotiem vaicājumiem.
Plugins.Oracle.CustomQueriesEnabled	Iespējot `oracle.custom.query` vienuma atslēgas izpildi.
Plugins.Oracle.Default.Password	Norādiet noklusējuma paroli.
Plugins.Oracle.Default.Service	Norādiet noklusējuma pakalpojuma nosaukumu.
Plugins.Oracle.Default.URI	Norādiet noklusējuma URI.
Plugins.Oracle.Default.User	Norādiet noklusējuma lietotājvārdu.
Plugins.Oracle.KeepAlive	Iestatiet maksimālo gaidīšanas laiku (sekundēs) pirms neizmantotie spraudņa savienojumi tiek slēgti.
Plugins.Oracle.ResolveTNS	Norādiet, kā interpretēt savienojuma virkni (ConnString), veidojot savienojumu ar Oracle serveris.
Plugins.Oracle.Sessions.<SessionName>.Password	Norādiet nosauktās sesijas paroli.
Plugins.Oracle.Sessions.<SessionName>.Service	Norādiet nosauktās sesijas pakalpojuma nosaukumu.
Plugins.Oracle.Sessions.<SessionName>.URI	Norādiet nosauktās sesijas URI.
Plugins.Oracle.Sessions.<SessionName>.User	Norādiet nosauktās sesijas lietotājvārdu.

Ņemiet vērā:

- Noklusējuma vērtības atspoguļo procesa noklusējuma iestatījumus, nevis vērtības piegādātajos konfigurācijas failos.
- Vērtības atbalsta vides mainīgos.
- Zabbix atbalsta konfigurācijas failus tikai UTF-8 kodējumā bez BOM.
- Komentāri, kas sākas ar "#", tiek atbalstīti tikai rindas sākumā.

Parametra informācija

Plugins.Oracle.CallTimeout

Maksimālais gaidīšanas laiks sekundēs, līdz pieprasījums tiek pabeigts.
 Šim parametram pašlaik nav ietekmes. Maksimālais gaidīšanas laiks tiek noteikts ar vienuma taimautu.

Noklusējums: `Timeout`[1]
 Diapazons: 1-30

Spraudņi.Oracle.ConnectTimeout

Norādiet maksimālo gaidīšanas laiku sekundēs, lai izveidotu savienojumu.

Noklusējums: `Timeout`[1]
 Diapazons: 1-30

Plugins.Oracle.CustomQueriesPath

Norādiet pilnu ceļu uz direktoriju, kurā atrodas `.sql` faili ar pielāgotiem vaicājumiem.

Noklusējums:
`/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`)

Piemērs:

```
/etc/zabbix/oracle/sql
```

Spraudņi.Oracle.CustomQueriesEnabled

Iespējot `oracle.custom.query` vienuma atslēgas izpildi.
 Ja ir atspējots, no pielāgoto vaicājumu direktorijas ceļa netiks ielādēti nekādi vaicājumi.

Noklusējums: `false`
 Diapazons: `true`/`false`

Plugins.Oracle.Default.Password

Norādiet noklusējuma paroli savienojumam ar Oracle.
 Tiek izmantota, ja vien vienuma atslēgā vai nosauktā sesijā nav norādīta vērtība.

Plugins.Oracle.Default.Service

Norādiet noklusējuma pakalpojuma nosaukumu savienojumam ar Oracle (SID netiek atbalstīts).
 Tiek izmantots, ja vienumā atslēgā vai nosauktajā sesijā nav norādīta vērtība.

Plugins.Oracle.Default.Uri

Norādiet noklusējuma URI savienojumam ar Oracle.
 Tiek izmantots, ja nav norādīta vērtība vienuma atslēgā vai nosauktā sesijā.
 Iebūvētie akreditācijas dati tiks ignorēti.
 Tam jāatbilst URI formātam.
 Atbalstītās shēmas: `tcp`, `tcps`
 Shēmu var nenorādīt.
 Portu var nenorādīt (noklusējums=1521).
 Kā savienojuma virkni ir iespējams norādīt arī TNS atslēgu vai TNS vērtību. TNS vērtība jāveido bez atstarpēm.

Piemēri:

```
tcp://127.0.0.1:1521
localhost
zbx_tns_example (TNS key)
(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=localhost)(PORT=1521))(CONNECT_DATA=(SERVICE_NAME=xe))) (TNS val
```

Plugins.Oracle.Default.User

Norādiet noklusējuma lietotājvārdu savienojumam ar Oracle.
 Tiek izmantots, ja vienums atslēgā vai nosauktajā sesijā nav norādīta vērtība.

Plugins.Oracle.KeepAlive

Iestata maksimālo gaidīšanas laiku (sekundēs) pirms neizmantotie spraudņa savienojumi tiek aizvērti.

Noklusējums: 300
 Diapazons: 60–900

Plugins.Oracle.ResolveTNS

Norādiet, kā interpretēt savienojuma virkni (ConnString), veidojot savienojumu ar Oracle serveri.
 Ja tā ir iestatīta uz `false`, noklusējuma shēma un ports tiks pievienoti ConnString (ja tie nav norādīti).
 Ja opcija ir iestatīta uz `true`, noklusējuma shēma un ports netiks iekļauti (ja vien tie nav skaidri norādīti ConnString), un ConnString tiks nodota Oracle klientam tāda, kāda tā ir. Ja Oracle klients atradīs ConnString failā `tnsnames.ora`, tiks izmantots atrastais savienojuma apraksts, lai izveidotu savienojumu ar Oracle serveri.

Noklusējums: `true`
 Diapazons: `true`/`false`

Plugins.Oracle.Sessions.<SessionName>.Password

Norādiet nosauktās sesijas paroli savienojumam ar Oracle.
 <SessionName> - norādiet sesijas nosaukumu, ko izmantot vienumu atslēgās.

Plugins.Oracle.Sessions.<SessionName>.Service

Norādiet nosauktā sesijas pakalpojuma nosaukumu savienojumam ar Oracle (SID netiek atbalstīts).
 <SessionName> - definējiet sesijas nosaukumu, ko izmantot vienumu atslēgās.

Plugins.Oracle.Sessions.<SessionName>.Uri

Norādiet nosauktās sesijas URI savienojumam ar Oracle.
 <SessionName> - definējiet sesijas nosaukumu, ko izmantot vienumu atslēgās.
 Iegultie akreditācijas dati tiks ignorēti.
 Jāatbilst URI formātam.
 Atbalstītās shēmas: `tcp`, `tcps`
 Shēmu var nenorādīt.
 Portu var nenorādīt (noklusējums=1521).
 Kā savienojuma virkni ir iespējams norādīt arī TNS atslēgu vai TNS vērtību. TNS vērtība jāveido bez atstarpēm.

Piemēri:

```
tcp://127.0.0.1:1521
localhost
zbx_tns_example (TNS key)
(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=localhost)(PORT=1521))(CONNECT_DATA=(SERVICE_NAME=xe))) (TNS val
```

Plugins.Oracle.Sessions.<SessionName>.User

Norādiet nosauktās sesijas lietotājvārdu savienojumam ar Oracle.
 <SessionName> - definējiet sesijas nosaukumu, ko izmantot vienumu atslēgās.

Zemsvītras piezīmes

[1] Taimauts tiek definēts ar opciju `Timeout` Zabbix aģenta 2 (UNIX) vai Zabbix aģenta 2 (Windows) konfigurācijas failā (`zabbix_agent2.conf`).

Skatīt arī

- Vispārīgo Zabbix aģents 2 konfigurācijas parametru apraksts: Zabbix aģents 2 (UNIX) / Zabbix aģents 2 (Windows)
- Norādījumi spraudņu konfigurēšanai

PostgreSQL spraudnis

Pārskats

Konfigurācijas parametri ļauj pielāgot PostgreSQL spraudņa darbību.

Ieteicams tos norādīt atsevišķā konfigurācijas failā (piem., `postgresql.conf`), nevis aģenta 2 konfigurācijas failā. Pēc tam izmantojiet direktīvu Include, lai iekļautu šo failu Zabbix aģenta 2 konfigurācijā.

PostgreSQL spraudnis ir ielādējams spraudnis, un tas ir pieejams un pilnībā aprakstīts PostgreSQL spraudņa repozitorijā.

Visi parametri ir **neobligāti**, ja vien nav skaidri norādīts, ka parametrs ir obligāts.

Šī lapa ir strukturēta šādi:

- Parametru pārskats (lai skatītu pilnu parametra informāciju, noklikšķiniet uz tā nosaukuma)
- Parametru detalizēts apraksts

Parameter	Description
Plugins.PostgreSQL.CallTimeout	Iestata maksimālo gaidīšanas laiku sekundēs, līdz pieprasījums tiek izpildīts.
Plugins.PostgreSQL.CustomQueriesPath	Norādiet pilnu direktorija ceļu, kurā atrodas `.sql` faili ar pielāgotiem vaicājumiem.
Plugins.PostgreSQL.CustomQueriesPath.Log	Iespējo `postgresql.custom.query` vienuma atslēgas izpildi.
Plugins.PostgreSQL.Default.InitMode	Norādiet PostgreSQL savienojuma noklusējuma kešatmiņas režīmu.
Plugins.PostgreSQL.Default.Database	Norādiet noklusējuma datubāzes nosaukumu.
Plugins.PostgreSQL.Default.Password	Norādiet noklusējuma paroli.
Plugins.PostgreSQL.Default.TLSCAFile	Norādiet pilnu ceļu uz failu, kurā ir augstākā līmeņa CA sertifikāts(-i) vienaudža sertifikāta verifikācijai.
Plugins.PostgreSQL.Default.TLSCertFile	Norādiet pilnu ceļu uz failu, kurā ir PostgreSQL sertifikāts vai sertifikātu ķēde.
Plugins.PostgreSQL.Default.TLSCipher	Norādiet noklusējuma šifrēšanas līmeni saziņai starp Zabbix aģentu 2 un uzraudzītajām datubāzēm.
Plugins.PostgreSQL.Default.TLSKeyFile	Norādiet pilnu ceļu uz failu, kurā ir PostgreSQL privātā atslēga.
Plugins.PostgreSQL.Default.Uri	Norādiet noklusējuma URI.
Plugins.PostgreSQL.Default.User	Norādiet noklusējuma lietotājvārdu.
Plugins.PostgreSQL.KeepAlive	Iestata maksimālo gaidīšanas laiku (sekundēs) pirms neizmantotie spraudņa savienojumi tiek slēgti.
Plugins.PostgreSQL.Sessions.<SessionName>.InitMode	Norādiet nosaukumā sesijas kešatmiņas režīmu PostgreSQL savienojumam.
Plugins.PostgreSQL.Sessions.<SessionName>.Database	Norādiet nosaukumā sesijas datubāzes nosaukumu.
Plugins.PostgreSQL.Sessions.<SessionName>.Password	Norādiet nosaukumā sesijas paroli.
Plugins.PostgreSQL.Sessions.<SessionName>.TLSCAFile	Norādiet pilnu ceļu uz failu, kurā ir augstākā līmeņa CA sertifikāts(-i) vienaudža sertifikāta verifikācijai.
Plugins.PostgreSQL.Sessions.<SessionName>.TLSCertFile	Norādiet pilnu ceļu uz failu, kurā ir PostgreSQL sertifikāts vai sertifikātu ķēde.
Plugins.PostgreSQL.Sessions.<SessionName>.TLSCipher	Norādiet nosaukumā sesijas šifrēšanas līmeni saziņai starp Zabbix aģentu 2 un uzraudzītajām datubāzēm.
Plugins.PostgreSQL.Sessions.<SessionName>.TLSKeyFile	Norādiet pilnu ceļu uz failu, kurā ir PostgreSQL privātā atslēga.
Plugins.PostgreSQL.Sessions.<SessionName>.Uri	Norādiet nosaukumā sesijas URI.
Plugins.PostgreSQL.Sessions.<SessionName>.User	Norādiet nosaukumā sesijas lietotājvārdu.

|Plugins.PostgreSQL.System.Path|Norādiet ceļu uz PostgreSQL spraudņa izpildāmo failu.| |Plugins.PostgreSQL.Timeout|Iestata pieprasījuma izpildes noildzi.|

Ņemiet vērā:

- Noklusējuma vērtības atspoguļo procesa noklusējuma iestatījumus, nevis vērtības piegādātajos konfigurācijas failos.
- Vērtības atbalsta vides mainīgos;
- Zabbix atbalsta konfigurācijas failus tikai UTF-8 kodējumā bez BOM.
- Komentāri, kas sākas ar ”#”, tiek atbalstīti tikai rindas sākumā.

Parametra informācija

Plugins.PostgreSQL.CallTimeout

Iestata maksimālo gaidīšanas laiku sekundēs, lai pieprasījums tiktu pabeigts.

Noklusējums: `global timeout`
 Diapazons: 1-30

Plugins.PostgreSQL.CustomQueriesPath

Norādiet pilnu ceļa nosaukumu uz direktoriju, kurā atrodas `.sql` faili ar pielāgotiem vaicājumiem.

Noklusējums:
/usr/local/share/zabbix/custom-queries/postgreSQL - Unix sistēmām,
*:\Program Files\Zabbix Agent 2\Custom Queries\PostgreSQL - Windows sistēmām (kur * ir diska nosaukums no vides mainīgā ProgramFiles)

Piemērs:

```
/etc/zabbix/oracle/sql
```

Plugins.PostgreSQL.CustomQueriesEnabled

Iespējot postgresql.custom.query vienuma atslēgas izpildi.
 Ja šī opcija ir atspējota, vaicājumi netiks ielādēti no pielāgoto vaicājumu direktorijas ceļa.

Noklusējums: false
 Diapazons: true/false

Plugins.PostgreSQL.Default.CacheMode

Norādiet PostgreSQL savienojuma noklusējuma kešatmiņas režīmu.

Noklusējums: prepare
 Diapazons:
prepare - izveidot sagatavotos priekšrakstus PostgreSQL serverī;
describe - izmantot anonīmu sagatavoto priekšrakstu, lai aprakstītu priekšrakstu, neveidojot priekšrakstu serverī.

Ņemiet vērā, ka describe galvenokārt ir noderīgs tad, ja vide neļauj izmantot sagatavotos priekšrakstus, piemēram, ja tiek izmantots savienojumu pūlētājs, piemēram, PgBouncer.

Plugins.PostgreSQL.Default.Database

Norādiet noklusējuma datubāzi savienojumam ar PostgreSQL.
 Tiek izmantota, ja vienumā atslēgā vai nosauktajā sesijā nav norādīta vērtība.

Plugins.PostgreSQL.Default.Password

Norādiet noklusējuma paroli savienojumam ar PostgreSQL.
 Tiek izmantota, ja vienums atslēgā vai nosauktajā sesijā nav norādīta vērtība.

Plugins.PostgreSQL.Default.TLSCAFile

Norādiet pilnu faila ceļu uz failu, kurā atrodas augstākā līmeņa CA sertifikāts(-i) vienādranga sertifikāta verifikācijai šifrētai saziņai starp Zabbix aģents 2 un uzraudzītajām datubāzēm.
 Tiek izmantots, ja nosauktajā sesijā nav norādīta vērtība.

Obligāti: yes, ja Plugins.PostgreSQL.Default.TLSConnect ir iestatīts uz verify_ca vai verify_full

Plugins.PostgreSQL.Default.TLSCertFile

Norādiet pilnu ceļu uz failu, kurā ir PostgreSQL sertifikāts vai sertifikātu ķēde šifrētai saziņai starp Zabbix aģentu 2 un uzraudzītajām datubāzēm.
 Tiek izmantots, ja nosauktajā sesijā nav norādīta vērtība.

Obligāts: yes, ja ir norādīts Plugins.PostgreSQL.Default.TLSKeyFile

Plugins.PostgreSQL.Default.TLSConnect

Norādiet noklusējuma šifrēšanas līmeni saziņai starp Zabbix aģentu 2 un uzraudzītajām datubāzēm.
 Tiek izmantots, ja nosauktajā sesijā nav norādīta vērtība.

Diapazons:
required - savienoties, izmantojot TLS kā transporta režīmu bez identitātes pārbaudes;
verify_ca - savienoties, izmantojot TLS un pārbaudīt sertifikātu;
verify_full - savienoties, izmantojot TLS, pārbaudīt sertifikātu un pārbaudīt, ka datubāzes identitāte (CN), kas norādīta ar DBHost, atbilst tās sertifikātam.

Nedefinēts šifrēšanas tips nozīmē nešifrētu savienojumu.

Plugins.PostgreSQL.Default.TLSKeyFile

Norādiet pilnu ceļu uz failu, kurā atrodas PostgreSQL privātā atslēga šifrētai saziņai starp Zabbix aģentu 2 un uzraudzītajām datubāzēm.
 Tiek izmantots, ja nosauktajā sesijā nav norādīta vērtība.

Obligāti: yes, ja ir norādīts Plugins.PostgreSQL.Default.TLSCertFile

Plugins.PostgreSQL.Default.Uri

Norādiet noklusējuma URI savienojumam ar PostgreSQL.
 Tiek izmantots, ja vienums atslēgā vai nosauktā sesijā nav norādīta vērtība.
 Iekļautie akreditācijas dati tiks ignorēti.
 Tam 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

Norādiet noklusējuma lietotājvārdu savienojumam ar PostgreSQL.
 Tiek izmantots, ja vienums atslēgā vai nosauktā sesijā nav norādīta vērtība.

Plugins.PostgreSQL.KeepAlive

Iestata maksimālo gaidīšanas laiku (sekundēs) pirms neizmantotie spraudņa savienojumi tiek slēgti.

Plugins.PostgreSQL.Sessions.<SessionName>.CacheMode

Norādiet PostgreSQL savienojuma nosauktās sesijas kešatmiņas režīmu.
 <SessionName> - definējiet sesijas nosaukumu, ko izmantot vienumu atslēgās.

Noklusējums: `prepare`
 Diapazons:
`prepare` - izveido sagatavotos priekšrakstus PostgreSQL serverī;
`describe` - izmanto anonīmu sagatavoto priekšrakstu, lai aprakstītu priekšrakstu, neveidojot priekšrakstu serverī.

Ņemiet vērā, ka `describe` galvenokārt ir noderīgs tad, ja vide neļauj izmantot sagatavotos priekšrakstus, piemēram, darbinot savienojumu pūlētāju, piemēram, PgBouncer.

Plugins.PostgreSQL.Sessions.<SessionName>.Database

Norādiet nosauktās sesijas datubāzi savienojumam ar PostgreSQL.
 <SessionName> - definējiet sesijas nosaukumu, ko izmantot vienumu atslēgās.

Plugins.PostgreSQL.Sessions.<SessionName>.Password

Norādiet nosauktās sesijas paroli savienojumam ar PostgreSQL.
 <SessionName> - definējiet sesijas nosaukumu, ko izmantot vienumu atslēgās.
 Jāatbilst paroles formātam.

Plugins.PostgreSQL.Sessions.<SessionName>.TLSCAFile

Norādiet pilnu ceļu uz failu, kurā atrodas augstākā līmeņa CA sertifikāts(-i) vienaudža sertifikāta verifikācijai šifrētai saziņai starp Zabbix aģents 2 un uzraudzītajām datubāzēm.
 <SessionName> - definējiet sesijas nosaukumu, ko izmantot vienumu atslēgās.

Obligāti: `yes`, ja `Plugins.PostgreSQL.Sessions.<SessionName>.TLSConnect` ir iestatīts uz `verify_ca` vai `verify_full`

Plugins.PostgreSQL.Sessions.<SessionName>.TLSCertFile

Norādiet pilnu ceļu uz failu, kurā ir PostgreSQL sertifikāts vai sertifikātu ķēde šifrētai saziņai starp Zabbix aģents 2 un uzraudzītajām datubāzēm.
 <SessionName> - definējiet sesijas nosaukumu, ko izmantot vienumu atslēgās.

Obligāts: `yes`, ja ir norādīts `Plugins.PostgreSQL.Sessions.<SessionName>.TLSKeyFile`

Plugins.PostgreSQL.Sessions.<SessionName>.TLSConnect

Norādiet nosaukto sesijas līmeņa šifrēšanu saziņai starp Zabbix aģents 2 un uzraudzītajām datubāzēm.
 <SessionName> - definējiet sesijas nosaukumu, ko izmantot vienuma atslēgās.

Diapazons:
`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 datubāzes identitāte (CN), kas norādīta ar DBHost, atbilst tās sertifikātam.

Nenoteikts šifrēšanas tips nozīmē nešifrētu savienojumu.

Plugins.PostgreSQL.Sessions.<SessionName>.TLSKeyFile

Norādiet pilnu ceļu uz failu, kurā atrodas PostgreSQL privātā atslēga šifrētai saziņai starp Zabbix aģentu 2 un uzraudzītajām datubāzēm.
 <SessionName> - norādiet sesijas nosaukumu, ko izmantot vienumu atslēgās.

Obligāts: `yes`, ja ir norādīts `Plugins.PostgreSQL.Sessions.<SessionName>.TLSCertFile`.

Plugins.PostgreSQL.Sessions.<SessionName>.Uri

Norādiet nosauktās sesijas URI savienojumam ar PostgreSQL.
 <SessionName> - definējiet sesijas nosaukumu, ko izmantot vienumu atslēgās.
 Iegultie akreditācijas dati 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

Norādiet nosauktās sesijas lietotājvārdu savienojumam ar PostgreSQL.
 <SessionName> - definējiet sesijas nosaukumu, ko izmantot vienumu atslēgās.

Plugins.PostgreSQL.System.Path

Norādiet PostgreSQL spraudņa izpildāmā faila ceļu.

Obligāti: `yes`

Piemērs:

```
Plugins.PostgreSQL.System.Path=/usr/libexec/zabbix/zabbix-agent2-plugin-postgresql
```

Plugins.PostgreSQL.Timeout

Iestatiet pieprasījuma izpildes noildzi (ilgumu sekundēs, cik ilgi gaidīt, līdz pieprasījums tiek pabeigts, pirms to pārtrauc).

Noklusējums: `global timeout<br>` Diapazons: 1-30

Opcijas

Parameter	Description
-V --version	Izdrukā spraudņa versiju un licences informāciju.
-h --help	Izdrukā palīdzības informāciju (saīsināti).
-t, --test <item key>	Palaiž spraudni testēšanai (spraudņa konfigurācija tiek ignorēta).

Skatiet arī

- Vispārīgo Zabbix aģents 2 konfigurācijas parametru apraksts: Zabbix aģents 2 (UNIX) / Zabbix aģents 2 (Windows)
- Norādījumi par spraudņu konfigurēšanu

Redis spraudnis

Pārskats

Konfigurācijas parametri ļauj pielāgot Redis spraudņa darbību.

Ieteicams tos norādīt atsevišķā konfigurācijas failā (piem., `redis.conf`), nevis aģents 2 konfigurācijas failā. Pēc tam izmantojiet direktīvu Include, lai iekļautu šo failu Zabbix aģents 2 konfigurācijā.

Visi parametri ir **neobligāti**, ja vien nav skaidri norādīts, ka parametrs ir obligāts.

Šī lapa ir strukturēta šādi:

- Parametru pārskats (lai skatītu pilnu parametra informāciju, noklikšķiniet uz tā nosaukuma)
- Parametru detaļas

Parameter	Description
Plugins.Redis.Default.Password	Norādiet noklusējuma paroli savienojumam ar Redis.
Plugins.Redis.Default.TLSConnect	Norādiet Zabbix aģents 2 komunikācijai nepieciešamo šifrēšanas līmeni.
Plugins.Redis.Default.TLSCAFile	Norādiet pilnu ceļu uz failu, kurā ir augstākā līmeņa CA sertifikāti.
Plugins.Redis.Default.TLSCertFile	Norādiet pilnu ceļu uz failu, kurā ir aģents sertifikāts.
Plugins.Redis.Default.TLSKeyFile	Norādiet pilnu ceļu uz failu, kurā ir aģents privātā atslēga.
Plugins.Redis.Default.Uri	Norādiet noklusējuma URI savienojumam ar Redis.
Plugins.Redis.Default.User	Norādiet noklusējuma lietotājvārdu savienojumam ar Redis.
Plugins.Redis.KeepAlive	Iestatiet maksimālo gaidīšanas laiku (sekundēs) pirms neizmantoto spraudņa savienojumu aizvēršanas.
Plugins.Redis.Sessions.<SessionName>.Password	Norādiet nosauktās sesijas paroli savienojumam ar Redis.
Plugins.Redis.Sessions.<SessionName>.TLSConnect	Norādiet Zabbix aģents 2 komunikācijai nepieciešamo šifrēšanas līmeni nosauktajai sesijai.
Plugins.Redis.Sessions.<SessionName>.TLSCAFile	Norādiet pilnu ceļu uz failu, kurā ir augstākā līmeņa CA sertifikāti nosauktajai sesijai.
Plugins.Redis.Sessions.<SessionName>.TLSCertFile	Norādiet pilnu ceļu uz failu, kurā ir aģents sertifikāts nosauktajai sesijai.
Plugins.Redis.Sessions.<SessionName>.TLSKeyFile	Norādiet pilnu ceļu uz failu, kurā ir aģents privātā atslēga nosauktajai sesijai.
Plugins.Redis.Sessions.<SessionName>.Uri	Norādiet nosauktās sesijas URI savienojumam ar Redis.
Plugins.Redis.Sessions.<SessionName>.User	Norādiet nosauktās sesijas lietotājvārdu savienojumam ar Redis.
Plugins.Redis.Timeout	Iestatiet pieprasījuma izpildes noildzi.

Ņemiet vērā:

- Noklusējuma vērtības atspoguļo procesa noklusējuma iestatījumus, nevis vērtības piegādātajos konfigurācijas failos.
- Vērtības atbalsta vides mainīgos.
- Zabbix atbalsta konfigurācijas failus tikai UTF-8 kodējumā bez BOM.
- Komentāri, kas sākas ar "#", tiek atbalstīti tikai rindas sākumā.

Parametra informācija

Plugins.Redis.Default.Password

Norādiet noklusējuma paroli savienojumam ar Redis.
 Tiek izmantota, ja vienums atslēgā vai nosauktajā sesijā nav norādīta vērtība.

Plugins.Redis.Default.TLSConnect

Norādiet Zabbix aģents 2 saziņai nepieciešamo šifrēšanas līmeni.
 Tukša vai nenorādīta vērtība tiek uzskatīta par TLS neesamību.

Diapazons: `required` - šifrēts, nepārbaudīts (tikai testēšanai), `verify_ca` - šifrēts, servera sertifikāts, `verify_full` - šifrēts, servera sertifikāts un servera nosaukums pārbaudīts ar SAN

Plugins.Redis.Default.TLSCAFile

Norādiet pilnu ceļu uz failu, kurā ir augstākā līmeņa CA sertifikāti.

Plugins.Redis.Default.TLSCertFile

Norādiet pilnu ceļu uz failu, kurā atrodas aģenta sertifikāts.

Plugins.Redis.Default.TLSKeyFile

Norādiet pilnu ceļu uz failu, kurā atrodas aģenta privātā atslēga.

Plugins.Redis.Default.Uri

Norādiet noklusējuma URI savienojumam ar Redis.
 Tiek izmantots, ja vienums atslēgā vai nosauktajā sesijā nav norādīta vērtība.
 Iebūvētie akreditācijas dati tiks ignorēti. Jāatbilst URI formātam.
 Atbalstītās shēmas: `tcp`, `unix`
 Shēmu var nenorādīt.
 Portu var nenorādīt (noklusējums=6379).

Noklusējums: `tcp://localhost:6379`

Piemēri:

```
tcp://localhost:6379
localhost
unix:/var/run/redis.sock
```

Plugins.Redis.Default.User

Norādiet noklusējuma lietotājvārdu savienojumam ar Redis.
 Tiek izmantots, ja vienumā atslēgā vai nosauktā sesijā nav norādīta vērtība.

Noklusējums: `default`

Plugins.Redis.KeepAlive

Iestata maksimālo gaidīšanas laiku (sekundēs) pirms neizmantotie spraudņa savienojumi tiek slēgti.

Noklusējums: 300
 Diapazons: 60–900

Plugins.Redis.Sessions.<SessionName>.Password

Norādiet nosauktās sesijas paroli savienojumam ar Redis.
 <SessionName> - definējiet sesijas nosaukumu, kas jāizmanto vienumu atslēgās.

Plugins.Redis.Sessions.<SessionName>.TLSConnect

Norādiet Zabbix aģents 2 sakaru šifrēšanas līmeni norādītajai sesijai.
 <SessionName> - definējiet sesijas nosaukumu, kas jāizmanto vienumu atslēgās.
 Tukša vai nenorādīta vērtība tiek uzskatīta par TLS neesamību.

Diapazons: `required` - šifrēts, nepārbaudīts (tikai testēšanai), `verify_ca` - šifrēts, servera sertifikāts, `verify_full` - šifrēts, servera sertifikāts un servera nosaukums pārbaudīts ar SAN

Plugins.Redis.Sessions.<SessionName>.TLSCAFile

Norādiet pilnu ceļu uz failu, kurā ir nosauktās sesijas augšējā līmeņa CA sertifikāti.
 <SessionName> - definējiet sesijas nosaukumu, kas jāizmanto vienumu atslēgās.

Obligāti: `yes`, ja `Plugins.Redis.Sessions.<SessionName>.TLSConnect` ir iestatīts uz `verify_ca` vai `verify_full`

Plugins.Redis.Sessions.<SessionName>.TLSCertFile

Norādiet pilnu faila ceļu uz failu, kurā atrodas aģenta sertifikāts norādītajai sesijai.
 <SessionName> - norādiet sesijas nosaukumu, kas tiks izmantots vienumu atslēgās.

Plugins.Redis.Sessions.<SessionName>.TLSKeyFile

Norādiet pilnu ceļu uz failu, kurā atrodas aģenta privātā atslēga norādītajai sesijai.
 <SessionName> - definējiet sesijas nosaukumu, kas jāizmanto vienumu atslēgās.

Plugins.Redis.Sessions.<SessionName>.Uri

Norādiet nosauktās sesijas URI savienojumam ar Redis.
 <SessionName> - definējiet sesijas nosaukumu, kas tiks izmantots vienumu atslēgās.
 Iegultie akreditācijas dati tiks ignorēti. Jāatbilst URI formātam.
 Atbalstītās shēmas: `tcp`, `unix`
 Shēmu var nenorādīt.
 Portu var nenorādīt (noklusējums=6379).

Noklusējums: `tcp://localhost:6379`

Piemēri:

```
tcp://localhost:6379
localhost
unix:/var/run/redis.sock
```

Plugins.Redis.Sessions.<SessionName>.User

Norādiet nosauktās sesijas lietotājvārdu savienojumam ar Redis.
 <SessionName> - definējiet sesijas nosaukumu, kas jāizmanto vienumu atslēgās.

Plugins.Redis.Timeout

Iestatiet pieprasījuma izpildes noildzi (ilgumu sekundēs, cik ilgi gaidīt, līdz pieprasījums tiek pabeigts, pirms to pārtrauc).

Noklusējums: `global timeout`
 Diapazons: 1–30

Skatiet arī

- Vispārīgo Zabbix aģenta 2 konfigurācijas parametru apraksts: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Norādījumi par sprauduņu konfigurēšanu

SMART spraudnis

Pārskats

Konfigurācijas parametri ļauj pielāgot SMART sprauduņa darbību.

Ieteicams tos norādīt atsevišķā konfigurācijas failā (piem., `smart.conf`), nevis aģents 2 konfigurācijas failā. Pēc tam izmantojiet direktīvu Include, lai iekļautu šo failu Zabbix aģents 2 konfigurācijā.

Visi parametri ir **neobligāti**, ja vien nav skaidri norādīts, ka parametrs ir obligāts.

Šī lapa ir strukturēta šādi:

- Parametru pārskats (lai skatītu pilnu parametra informāciju, noklikšķiniet uz tā nosaukuma)
- Parametru detaļas

Parameter	Description
Plugins.Smart.Path	Norādiet smartctl izpildāmā faila ceļu.
Plugins.Smart.Timeout	Iestatiet pieprasījuma izpildes noildzi (ilgumu sekundēs, cik ilgi gaidīt, līdz pieprasījums tiks pabeigts, pirms to apturēt).

Ņemiet vērā, ka:

- Noklusējuma vērtības atspoguļo procesa noklusējumus, nevis vērtības piegādātajos konfigurācijas failos;
- Vērtības atbalsta vides mainīgos;
- Ceļš uz `smartctl` izpildāmo failu ir jāiestata 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.

Parametra informācija

Plugins.Smart.Path

Norādiet ceļu uz smartctl izpildāmo failu.

Noklusējums: `smartctl`

Plugins.Smart.Timeout

Iestata pieprasījuma izpildes noildzi (ilgumu sekundēs, cik ilgi gaidīt, līdz pieprasījums tiek pabeigts, pirms to pārtrauc).

Noklusējums: `global timeout<br>` Diapazons: 1–30

Skatīt arī

- Vispārīgo Zabbix aģents 2 konfigurācijas parametru apraksts: Zabbix aģents 2 (UNIX) / Zabbix aģents 2 (Windows)
- Norādījumi spraudņu konfigurēšanai

Zabbix aģents 2 spraudņa protokols

Zabbix aģents 2 protokols ir balstīts uz kodu, izmēru un datu modeli.

Kods

Type	Size	Comments
Byte	4	Ielādes tips, pašlaik tiek atbalstīts tikai JSON.

Izmērs

Type	Size	Comments
Byte	4	Pašreizējā datu apjoma izmērs baitos.

Ielādes dati

Type	Size	Comments
Byte	Definēts ar lauku *Size*	JSON formāta dati.

Datu kopas definīcija

Kopējie dati

Šie parametri ir pieejami visos pieprasījumos/atbildēs:

Name	Type	Comments
id	uint32	Pieprasījumiem - pieaugošs identifikators, ko izmanto pieprasījumu sasaistīšanai ar atbildēm. Unikāls vienā pieprasījuma virzienā (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.

direction	spraudnis → aģents
response	nē

Žurnāla pieprasījumiem specifiskie parametri:

Name	Type	Comments
severity	uint32	Ziņojuma svarīgums (žurnāla līmenis).
message	string	Ziņojums, ko ierakstīt žurnālā.

Piemērs:

```
{"id":0,"type":1,"severity":3,"message":"message"}
```

Reģistrācijas pieprasījums

Pieprasījums, ko aģents nosūta aģenta startēšanas fāzē, lai iegūtu sniegtos metriskos datus un reģistrētu spraudni.

direction	aģents → spraudnis
response	yes

Reģistrācijas pieprasījumiem specifiskie parametri:

Name	Type	Comments
version	string	Protokola versija <major>.<minor>

Piemērs:

```
{"id":1,"type":2,"version":"1.0"}
```

Reģistrācijas atbilde

Spraudņa atbilde uz reģistrācijas pieprasījumu.

direction	spraudnis → aģents
response	n/a

Reģistrācijas atbildēm specifiskie parametri:

Name	Type	Comments
name	string	Spraudņa nosaukums.
metrics	array of strings (optional)	Metrikas ar aprakstiem, kā tās tiek izmantotas spraudnī. Atgriež RegisterMetrics(). Nav, ja tiek atgriezta kļūda.
interfaces	uint32 (optional)	Spraudņa atbalstīto saskarņu bitu maska. Nav, ja tiek atgriezta kļūda.
error	string (optional)	Kļūdas ziņojums, kas tiek atgriezts, ja spraudni nevar palaist. Nav, 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"}
```

Sākuma pieprasījums

Pieprasījums izpildīt Runner saskarnes funkciju Start.

direction	aģents → plugin
response	no

Pieprasījumam nav specifisku parametru, tas satur tikai kopīgo datu parametrus.

Piemērs:

```
{"id":3,"type":4}
```

Pieprasījuma pārtraukšana

Pieprasījums, ko aģents nosūta, lai apturētu spraudni.

direction	aģents → spraudnis
response	no

Šim pieprasījumam nav specifisku parametru, tas satur tikai kopīgo datu parametrus.

Piemērs:

```
{"id":3,"type":5}
```

Eksporta pieprasījums

Pieprasījums izpildīt Export funkciju Exporter interfeisā.

direction	aģents → plugin
response	no

Eksporta pieprasījumiem specifiskie parametri:

Name	Type	Comments
key	string	Spraudņa atslēga.
parameters	array of strings (optional)	Export funkcijas parametri.

Piemērs:

```
{"id":4,"type":6,"key":"test.key","parameters":["foo","bar"]}
```

Eksporta atbilde

Atbilde no Export funkcijas Exporter saskarnē.

direction	spraudnis → aģents
response	n/a

Eksporta atbildēm specifiskie parametri:

Name	Type	Comments
value	string (optional)	Atbildes vērtība no Export funkcijas. Nav norādīts, ja tiek atgriezta kļūda.
error	string (optional)	Kļūdas ziņojums, ja Export funkcija 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*.

direction	aģents → spraudnis
response	n/a

Parametri, kas raksturīgi *Configure* pieprasījumiem:

Name	Type	Comments
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 tās ir norādītas.

Piemērs:

```
{"id":6,"type":8,"global_options":{...},"private_options":{...}}
```

Validēt pieprasījumu

Pieprasījums izpildīt *Validate* funkciju no *Configurator* saskarnes.

direction	aģents → plugin
response	yes

Parametri, kas raksturīgi *Validate* pieprasījumiem:

Name	Type	Comments
private_options	JSON object (optional)	JSON object, kas satur privātās spraudņa konfigurācijas opcijas, ja tās ir norādītas.

Piemērs:

```
{"id":7,"type":9,"private_options":{...}}
```

Validēt atbildi

Atbilde no *Validate* funkcijas *Configurator* interfeisā.

direction	spraudnis → aģents
response	n/a

Parametri, kas raksturīgi *Validate* atbildēm:

Name	Type	Comments
error	string (optional)	Kļūdas ziņojums, kas tiek atgriezts, ja funkcija Validate netiek izpildīta veiksmīgi. Nav pieejams, ja izpilde ir veiksmīga.

Piemērs:

```
{"id":8,"type":10}
```

vai

```
{"id":8,"type":10,"error":"error message"}
```

Java gateway

Pārskats Zabbix Java gateway var instalēt no avota koda vai pakotnēm.

JMX lietojumprogrammu sākotnējais pārraudzības atbalsts ir pieejams Zabbix dēmona "Zabbix Java gateway" veidā. Zabbix Java gateway ir Java valodā rakstīts dēmons. Lai noskaidrotu konkrēta JMX skaitītāja vērtību hostā, Zabbix serveris nosūta vaicājumu Zabbix Java gateway, kas izmanto JMX pārvaldības API, lai attālināti nosūtītu vaicājumu par interesējošo lietojumprogrammu. Lietojumprogrammai nav nepieciešama papildu programmatūra — tā tikai jāpalaiž komandrindā ar opciju `-Dcom.sun.management.jmxremote`.

Java gateway pieņem ienākošos savienojumus no Zabbix servera vai starpniekservera, un to var izmantot tikai kā "pasīvu starpniekserveri". Atšķirībā no Zabbix starpniekservera to var izmantot arī no Zabbix starpniekservera (Zabbix starpniekserverus nevar savienot ķēdē). Piekļuve katram Java gateway tiek konfigurēta tieši Zabbix servera vai Zabbix starpniekservera konfigurācijas failā, tādēļ katram Zabbix serverim vai Zabbix starpniekserverim var konfigurēt tikai vienu Java gateway. Ja hostam būs vienumi ar tipu *JMX aģents* un vienumi ar citu tipu, izgūšanai uz Java gateway tiks nosūtīti tikai *JMX aģenta* vienumi.

Kad vienums ir jāatjaunina, izmantojot Java gateway, Zabbix serveris vai starpniekserveris izveidos savienojumu ar Java gateway un pieprasīs vērtību, kuru Java gateway savukārt izgūst un nosūta atpakaļ serverim vai starpniekserverim. Tādējādi Java gateway neveic vērtību kešatmiņošanu.

Zabbix serverim vai starpniekserverim ir īpaša tipa procesi, kas izveido savienojumu ar Java gateway; tos kontrolē opcija `StartJavaPollers`. Iekšēji Java gateway palaiž vairākus pavedienus, kurus kontrolē opcija `START_POLLERS` option. Servera pusē, ja savienojuma izveide ilgst vairāk nekā "Timeout" sekundes, tas tiek pārtraukts, taču Java gateway joprojām var būt aizņemts ar vērtības izgūšanu no JMX skaitītāja. Lai to novērstu, Java gateway nodrošina opciju `TIMEOUT`, kas ļauj iestatīt taimautu JMX tīkla darbībām.

Skatiet arī:

- Configuration parameters: Zabbix Java gateway
- Setup from Debian/Ubuntu packages
- Setup from RHEL packages
- Setup from sources

Zabbix serveris vai starpniekserveris centīsies pēc iespējas vairāk apvienot pieprasījumus vienam JMX mērķim (to ietekmē vienumu intervāli) un nosūtīt tos uz Java gateway vienā savienojumā, lai nodrošinātu labāku veiktspēju.

Ieteicams, lai `StartJavaPollers` būtu mazāks vai vienāds ar `START_POLLERS`; pretējā gadījumā var rasties situācijas, kad Java gateway nav pieejamu pavedienu ienākošo pieprasījumu apkalpošanai. Šādos gadījumos Java gateway izmanto `ThreadPoolExecutor.CallerRunsPolicy`, kas nozīmē, ka galvenais pavediens apkalpos ienākošo pieprasījumu un īslaicīgi nepieņems jaunus pieprasījumus.

Ja mēģināt uzraudzīt uz Wildfly balstītas Java lietotnes ar Zabbix Java gateway, lūdzu, instalējiet jaunāko `jboss-client.jar`, kas pieejams Wildfly lejupielādes lapā.

Konfigurācijas parametri: Zabbix Java gateway

Ja Zabbix Java gateway palaišanai izmantojat `startup.sh` un `shutdown.sh` skriptus, tad nepieciešamos konfigurācijas parametrus varat norādīt `settings.sh` failā. Palaišanas un apturēšanas skripti nolasa iestatījumu failu un parūpējas par čaulas mainīgo (uzskaitīti pirmajā kolonnā) pārveidošanu Java īpašībās (uzskaitītas otrajā kolonnā).

Ja Zabbix Java gateway palaižat 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, uz kuras klausīties.
LISTEN_PORT	zabbix.listenPort	no	1024-32767	10052	Ports, uz kura klausīties.
PID_FILE	zabbix.pidFile	no		/tmp/zabbix_java.pid	PID faila nosaukums. Ja tas netiek norādīts, Zabbix Java Gateway tiek palaists kā konsoles lietojumprogramma.

Variable	Property	Mandatory	Range	Default	Description
PROPERTIES_FILE	zabbix.propertiesFile	no			Īpašību faila nosaukums. To var izmantot, lai iestatītu papildu īpašības, izmantojot atslēgas-vērtības formātu, tādā veidā, lai tās nebūtu redzamas komandrindā, vai lai pārrakstītu esošās. Piemēram: ”javax.net.ssl.trustStorePassword=<p
START_POLLERS	zabbix.startPollers	no	1-1000	5	Darba pavedienu skaits, ko palaist.
TIMEOUT	zabbix.timeout	no	1-30	3	Cik ilgi gaidīt tīkla operācijas (sekundēs).

Warning:
Ports 10052 nav IANA reģistrēts.

Iestatīšana no avotiem

Pārskats

Ja instalēts no avota kodiem, tālāk sniegtā informācija palīdzēs jums konfigurēt Zabbix Java gateway.

Failu pārskats

Ja Java gateway ieguvāt no pirmkoda, jums vajadzētu būt čaulas skriptu, JAR un konfigurācijas failu kopumam direktorijā `$PREFIX/sbin/zabbix_java`. Šo failu nozīme ir apkopota tālāk.

`bin/zabbix-java-gateway-$VERSION.jar`

Pats Java gateway JAR fails.

```
lib/logback-core-1.5.25.jar
lib/logback-classic-1.5.25.jar
lib/slf4j-api-2.0.16.jar
lib/android-json-4.3_r3.1.jar
```

Java gateway atkarības: Logback, SLF4J un Android JSON bibliotēka.

```
lib/logback.xml
lib/logback-console.xml
```

Logback konfigurācijas faili.

```
shutdown.sh
startup.sh
```

Ērtības skripti Java gateway palaišanai un apturēšanai.

`settings.sh`

Konfigurācijas fails, kas tiek ielādēts iepriekš minētajos palaišanas un apturēšanas skriptos.

Java gateway konfigurēšana un palaišana

Pēc noklusējuma Java gateway klausās 10052. portā. Ja plānojat palaist Java gateway citā portā, to var norādīt `settings.sh` skriptā. Skatiet Java gateway konfigurācijas faila aprakstu, lai uzzinātu, kā norādīt šo un citas opcijas.

Warning:
10052. ports nav IANA reģistrēts.

Kad iestatījumi ir sagatavoti, varat palaist Java gateway, izpildot `startup.sh` skriptu:

```
./startup.sh
```

Tāpat, kad Java gateway vairs nav nepieciešams, izpildiet `shutdown.sh` 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 lietošanai ar Java gateway

Kad Java gateway ir palaists un darbojas, jums jānorāda Zabbix serverim, kur atrast Zabbix Java gateway. To veic, norādot parametrus `JavaGateway` un `JavaGatewayPort` servera konfigurācijas failā. Ja hosts, kurā darbojas JMX lietojumprogramma, tiek uzraudzīts ar Zabbix starpniekserveri, tad savienojuma parametrus norādāt starpniekservera konfigurācijas failā.

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

Pēc noklusējuma serveris nepalaiž nekādus procesus, kas saistīti ar JMX uzraudzību. Tomēr, ja vēlaties to izmantot, jums jānorāda Java polleru iepriekš atzaroto instanču skaits. To veicat tāpat, kā norādāt parastos pollerus un traperus.

```
StartJavaPollers=5
```

Neaizmirstiet pēc konfigurēšanas pabeigšanas restartēt serveri vai starpniekserveri.

Java gateway atkļūdošana

Ja rodas kādas problēmas ar Java gateway vai kļūdas ziņojums, ko redzat par vienumu lietotāja saskarnē, nav pietiekami informatīvs, iespējams, vēlēsieties apskatīt Java gateway žurnāla failu.

Pēc noklusējuma Java gateway reģistrē savu darbību failā `/tmp/zabbix_java.log` ar žurnāla līmeni "info". Dažkārt šī informācija nav pietiekama, un ir nepieciešama informācija ar žurnāla līmeni "debug". Lai paaugstinātu žurnāla līmeni, modificējiet failu `lib/logback.xml` un mainiet `<root>` taga `level` atribūtu uz `debug`:

```
<root level="debug">
  <appender-ref ref="FILE" />
</root>
```

Ņemiet vērā, ka atšķirībā no Zabbix serveris vai Zabbix starpniekserveris, pēc `logback.xml` faila izmaiņām Zabbix Java gateway nav jāpārstartē - izmaiņas `logback.xml` failā tiks automātiski ņemtas vērā. Kad atkļūdošana ir pabeigta, varat atgriezt žurnāla līmeni uz "info".

Ja vēlaties reģistrēt datus citā failā vai pilnīgi citā datu nesējā, piemēram, datubāzē, pielāgojiet `logback.xml` failu atbilstoši savām vajadzībām. Plašāku informāciju skatiet Logback Manual.

Dažkārt atkļūdošanas nolūkos ir noderīgi palaist Java gateway kā konsoles lietojumprogrammu, nevis kā dēmonu. Lai to izdarītu, komentējiet `PID_FILE` mainīgo failā `settings.sh`. Ja `PID_FILE` tiek izlaists, `startup.sh` skripts palaiž Java gateway kā konsoles lietojumprogrammu un liek Logback izmantot failu `lib/logback-console.xml`, kas ne tikai reģistrē datus konsolē, bet arī iespējo žurnāla līmeni "debug".

Visbeidzot, ņemiet vērā, ka, tā kā Java gateway žurnālošanai izmanto SLF4J, jūs varat aizstāt Logback ar jebkuru sev vēlamo ietvaru, ievietojot atbilstošu JAR failu `lib` direktorijā. Plašāku informāciju skatiet SLF4J Manual.

JMX monitorings

Skatiet lapu JMX monitoring, lai iegūtu plašāku informāciju.

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 Zabbix Java gateway automātiski startētu sistēmas palaišanas laikā:

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

Servera konfigurēšana lietošanai 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 lietojumprogramma, tiek uzraudzīts ar Zabbix starpniekserveri, tad savienojuma parametrus norādiet starpniekservera konfigurācijas failā.

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

Pēc noklusējuma serveris nepalaiž nekādus procesus, kas saistīti ar JMX uzraudzību. Tomēr, ja vēlaties to izmantot, jums jānorāda Java polleru iepriekš atzaroto instanču skaits. To izdara tāpat, kā norādāt parastos pollerus un trapperus.

```
StartJavaPollers=5
```

Neaizmirstiet restartēt serveri vai starpniekserveri, kad konfigurēšana ir pabeigta.

Java gateway atkļūdošana

Zabbix Java gateway žurnāla fails ir:

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

Ja vēlaties palielināt reģistrēšanas detalizācijas līmeni, rediģējiet failu:

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

un mainiet `info` sadaļā `level="info"` uz debug vai pat `trace` (dziļai problēmu novēršanai):

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

</configuration>
```

JMX monitorings

Skatiet JMX monitoring lapu, lai iegūtu vairāk informācijas.

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
```

Plašāku informāciju skatiet Zabbix Java gateway konfigurācijas parametros.

Lai palaistu Zabbix Java gateway:

```
systemctl restart zabbix-java-gateway
```

Lai Zabbix Java gateway automātiski startētu sistēmas palaišanas laikā:

RHEL 7 un jaunākās versijas:

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

RHEL pirms 7:

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

Servera konfigurēšana lietošanai 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 lietojumprogramma, tiek uzraudzīts ar Zabbix starpniekserveri, tad savienojuma parametrus norādiet starpniekservera konfigurācijas failā.

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

Pēc noklusējuma serveris nepalaiž nekādus procesus, kas saistīti ar JMX uzraudzību. Tomēr, ja vēlaties to izmantot, jums jānorāda Java polleru iepriekš atzaroto instanču skaits. To izdara tāpat kā norādot parastos pollerus un trapperus.

```
StartJavaPollers=5
```

Neaizmirstiet restartēt serveri vai starpniekserveri, kad konfigurēšana ir pabeigta.

Java gateway atkļūdošana

Zabbix Java gateway žurnāla fails 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 `info` sadaļā `level="info"` uz debug vai pat `trace` (dziļākai problēmu novēršanai):

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

</configuration>
```

JMX monitorings

Skatiet JMX monitoring lapu, lai iegūtu sīkāku informāciju.

Sūtītājs

Pārskats Zabbix sender ir komandrindas utilīta, ko var izmantot veiktspējas datu nosūtīšanai uz Zabbix serveri vai starpniekserveri apstrādei.

Utilītu parasti izmanto ilgstoši darbojošos lietotāja skriptos, lai periodiski nosūtītu pieejamības un veiktspējas datus.

Lai rezultātus nosūtītu tieši uz Zabbix serveri vai starpniekserveri, ir jākonfigurē *Zabbix trapper* vienuma tips.

Skatiet arī:

- Zabbix sender protokols
- Zabbix sender dinamiskā saišu bibliotēka operētājsistēmai Windows
- Python bibliotēka operētājsistēmai Zabbix, kurai ir iebūvēta funkcionalitāte darbībai 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 hosts (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, jāiekļauj dubultajās pēdiņās.

Zabbix sender var izmantot, lai nosūtītu vairākas vērtības no ievades faila. Plašāku informāciju skatiet Zabbix sender manlapā.

Ja ir norādīts konfigurācijas fails, Zabbix sender datu nosūtīšanai izmanto visas aģenta `ServerActive` konfigurācijas parametrā definētās adreses. Ja nosūtīšana uz vienu adresi neizdodas, sender mēģina nosūtīt datus uz citām adresēm. Ja pakešdatu nosūtīšana uz vienu adresi neizdodas, nākamās paketes uz šo adresi netiek nosūtītas.

Zabbix sender pieņem UTF-8 kodējuma virknes (gan Unix līdzīgās sistēmās, gan Windows), ja faila sākumā nav baitu secības marķiera (BOM).

Zabbix sender operētājsistēmā Windows var palaist līdzīgi:

```
zabbix_sender.exe [options]
```

`zabbix_sender` reāllaika nosūtīšanas scenāriji apkopos vairākas tam nodotās vērtības, kas tiek nodotas ar nelielu intervālu, un nosūtīs tās serverim viena savienojuma ietvaros. Vērtību, kas no iepriekšējās vērtības atšķiras ne vairāk kā par 0,2 sekundēm, var ievietot vienā stekā, taču maksimālais aptaujas laiks joprojām ir 1 sekunde.

> **Note:**
> Zabbix sender tiks pārtraukts, ja norādītajā konfigurācijas failā būs nederīgs parametrs (kas neatbilst apzīmējumam `parameter=value`).

Zabbix sender palaišana ar zema līmeņa atklāšanu Zabbix sender palaišanas piemērs, lai nosūtītu JSON formātā vērtību zema līmeņa atklāšanai:

```
./zabbix_sender -z 192.168.1.113 -s "Zabbix serveris" -k trapper.discovery.item -o '[{"{#FSNAME}":"/","{#F
```

Lai tas darbotos, zema līmeņa atklāšanas noteikumam jābūt ar *Zabbix trapper* vienuma tipu (šajā piemērā ar `trapper.discovery.item` atslēgu).

Zabbix sender protokols

Zabbix serveris un Zabbix starpniekserveris izmanto uz JSON balstītu saziņas protokolu datu saņemšanai no Zabbix sender. Datus var saņemt, izmantojot trapper vienumu vai HTTP aģenta vienumu ar iespējotu trapping.

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 servera 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 servera atbilde

```
{
    "response": "success",
    "info": "processed: 2; failed: 0; total: 2; seconds spent: 0.060904"
}
```

Zabbix sender dinamiskā saišu bibliotēka operētājsistēmai Windows

Windows vidē lietojumprogrammas var nosūtīt datus uz Zabbix serveri/starpniekserveri, izmantojot Zabbix sender dinamisko saišu bibliotēku (`zabbix_sender.dll`), nevis palaižot ārēju procesu (`zabbix_sender.exe`).

Lai kompilētu lietotāja 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 `zabbix_sender.h`, `zabbix_sender.lib` un `zabbix_sender.dll` failus ZIP arhīvā.

Izvēloties lejupielādes opcijas, pārliecinieties, ka sadaļā *Encryption* ir atlasīts "No encryption", bet sadaļā *Packaging* - "Archive". Pēc tam lejupielādējiet Zabbix aģents (nevis Zabbix agent 2).

`zabbix_sender.h`, `zabbix_sender.lib` un `zabbix_sender.dll` faili atradīsies lejupielādētā ZIP arhīva `bin\dev` direktorijā. Atarhivējiet failus vietā, kur tie ir nepieciešami.

2. Izveidojiet `zabbix_sender.dll` no pirmkoda (skatiet instrukcijas).

Dinamiskā saišu bibliotēka kopā ar izstrādes failiem atradīsies `bin\winXX\dev` direktorijā. Lai to izmantotu, iekļaujiet `zabbix_sender.h` galvenes failu un piesaistiet `zabbix_sender.lib` bibliotēku.

Skatiet arī

- piemēru vienkāršam Zabbix sender utilītam, kas ieviests ar Zabbix sender dinamisko saišu bibliotēku, lai ilustrētu bibliotēkas lietošanu;
- `zabbix_sender.h` failu ar Zabbix sender dinamiskās saišu bibliotēkas saskarnes funkcijām. Šis fails satur dokumentāciju, kas izskaidro katras saskarnes funkcijas mērķi, tās argumentus un atgrieztās vērtības.

Saņemt

Pārskats Zabbix get ir komandrindas utilīta, ko var izmantot saziņai ar Zabbix aģentu un nepieciešamās informācijas iegūšanai no aģenta.

Šo utilītu parasti izmanto Zabbix aģentu problēmu novēršanai.

Skatiet arī Python bibliotēku Zabbix, kurai ir iebūvēta funkcionalitāte, kas darbojas līdzīgi Zabbix get.

Zabbix get palaišana Piemērs, kā UNIX vidē palaist Zabbix get, lai iegūtu procesora noslodzes vērtību no aģenta:

```
cd bin
./zabbix_get -s 127.0.0.1 -p 10050 -k system.cpu.load[all,avg1]
```

Vēl viens Zabbix get palaišanas piemērs, 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 vienuma atslēgā šeit ir atstarpe, tāpēc čaulas interpretācijai vienuma atslēga tiek norādīta pēdiņās. 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>             Norādīt hosta nosaukumu vai hosta IP adresi
-p --port <port number>                 Norādīt hostā darbojošā aģenta porta numuru (noklusējums: 10050)
-I --source-address <IP address>        Norādīt avota IP adresi
-t --timeout <seconds>                  Norādīt noildzi. Derīgais diapazons: 1-600 sekundes (noklusējums:
-k --key <item key>                     Norādīt vienuma atslēgu, kuras vērtību iegūt
-P --protocol <value>                   Protokols, ko izmantot saziņai ar aģentu. Vērtības:
                                            auto - izveidot savienojumu, izmantojot JSON protokolu, ar atk
                                            json - izveidot savienojumu, izmantojot JSON protokolu
                                            plaintext - izveidot savienojumu, izmantojot plaintext protoko
-h --help                               Parādīt šo palīdzības ziņojumu
-V --version                            Parādīt versijas numuru

--tls-connect <value>                   Kā izveidot savienojumu ar aģentu. Vērtības:
                                            unencrypted - izveidot savienojumu bez šifrēšanas (noklusējums
                                            psk - izveidot savienojumu, izmantojot TLS un iepriekš koplie
                                            cert - izveidot savienojumu, izmantojot TLS un sertifikātu
--tls-ca-file <CA file>                 Pilns ceļš uz failu, kurā ir augstākā līmeņa CA sertifikāti vienau
--tls-crl-file <CRL file>               Pilns ceļš uz failu, kurā ir atsauktie sertifikāti
--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, kurā ir sertifikāts vai sertifikātu ķēde
--tls-key-file <key file>               Pilns ceļš uz failu, kurā ir privātā atslēga
--tls-psk-identity <PSK-identity>       Unikāla, reģistrjutīga virkne, ko izmanto iepriekš koplietotās ats
--tls-psk-file <PSK-file>               Pilns ceļš uz failu, kurā ir iepriekš koplietotā atslēga
--tls-cipher13 <cipher-string>          Šifrēšanas virkne OpenSSL 1.1.1 vai jaunākai versijai TLS 1.3. Pār
--tls-cipher <cipher-string>            GnuTLS prioritātes virkne (TLS 1.2 un jaunākām versijām) vai OpenS
```

Papildinformāciju skatiet arī Zabbix get rokasgrāmatas lapā.

Zabbix get operētājsistēmā Windows var palaist līdzīgā veidā:

```
zabbix_get.exe [options]
```

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 skriptēšanas dzinēju.

Kompilācijas vai izpildes kļūdu gadījumā zabbix_js izdrukās kļūdu stderr un izies ar kodu 1.

```
zabbix_js -s script-file -p input-param [-l log-level] [-t timeout]
zabbix_js -s script-file -i input-file [-l log-level] [-t timeout]
zabbix_js -h
zabbix_js -V
```

Lietošana zabbix_js pieņem šādus komandrindas parametrus:

```
-s, --script script-file          Norādiet izpildāmā skripta faila nosaukumu
                                  Ja kā faila nosaukums ir norādīts "-", skripts tiks nolasīts no stdin
-i, --input input-file            Norādiet ievades informācijas faila nosaukumu
                                  Saturs tiks nodots skriptam kā vērtība
                                  Savstarpēji izslēdzošs ar -p
                                  Ja kā faila nosaukums ir norādīts "-", skripts tiks nolasīts no stdin
-p, --param input-param           Norādiet ievades parametru: mainīgo, kas tiks nodots skriptam kā vērtība
                                  Savstarpēji izslēdzošs ar -i
-l, --loglevel log-level          Norādiet žurnāla līmeni
                                  Diapazons: 0-5
-t, --timeout timeout             Norādiet noildzi sekundēs
```

```
                                Derīgais diapazons: 1-600 sekundes (noklusējums: 10 sekundes)
-h, --help                      Parādīt palīdzības informāciju
-V, --version                   Parādīt versijas numuru
-w <webdriver url>              Iespējo pārlūkprogrammas 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`

Tīmekļa pakalpojums

Pārskats Zabbix tīmekļa pakalpojums ir process, kas tiek izmantots saziņai ar ārējiem tīmekļa pakalpojumiem. Pašlaik Zabbix tīmekļa pakalpojums tiek izmantots plānoto atskaišu ģenerēšanai un nosūtīšanai, 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; dažos izplatījumos 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 konfigurēšanas opciju `--enable-webservice`.

Lai konfigurētu Zabbix tīmekļa servisu, atjauniniet tīmekļa servisa konfigurācijas faila 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, tādēļ var rasties nesankcionētas piekļuves risks.

Konfigurācijas parametri: Zabbix tīmekļa pakalpojums

Pārskats

Zabbix tīmekļa pakalpojums ir process, kas tiek izmantots 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.

Parameter	Description
AllowedIP	Ar komatiem atdalītu IP adrešu saraksts, pēc izvēles CIDR notācijā, vai Zabbix serveru un Zabbix starpniekserveru DNS nosaukumi.
DebugLevel	Atkļūdošanas līmenis.
IgnoreURLCertErrors	Norāda TLS sertifikāta validācijas kļūdu apstrādi, piekļūstot lietotāja saskarnes URL.
Include	Konfigurācijas failā var iekļaut atsevišķus failus vai visus failus direktorijā.
ListenPort	Pakalpojums klausīsies šajā portā uz savienojumiem no servera.
LogFile	Žurnālfaila nosaukums.
LogFileSize	Maksimālais žurnālfaila izmērs.
LogType	Žurnāla izvades veids.
Timeout	Maksimālais laiks (sekundēs), kas pavadīts paneļa PDF report formatēšanai.
TLSAccept	Kādus ienākošos savienojumus pieņemt.
TLSCAFile	Pilns ceļš uz failu, kurā ir augstākā līmeņa CA sertifikāti vienaudžu sertifikātu verifikācijai, ko izmanto šifrētai saziņai starp Zabbix komponentēm.
TLSCertFile	Pilns ceļš uz failu, kurā ir pakalpojuma sertifikāts vai sertifikātu ķēde, ko izmanto šifrētai saziņai starp Zabbix komponentēm.
TLSKeyFile	Pilns ceļš uz failu, kurā ir pakalpojuma privātā atslēga, ko izmanto šifrētai saziņai starp Zabbix komponentēm.

Visi parametri nav obligāti, ja vien nav skaidri norādīts, ka parametrs ir obligāts.

Ņemiet vērā:

- Noklusējuma vērtības atspoguļo procesa noklusējumus, nevis vērtības piegādātajos konfigurācijas failos;
- Vērtības atbalsta vides mainīgos;
- Zabbix atbalsta konfigurācijas failus tikai UTF-8 kodējumā bez BOM;
- Komentāri, kas sākas ar #, tiek atbalstīti tikai rindas sākumā.

Parametra informācija

AllowedIP

Ar komatiem atdalīts IP adrešu saraksts, pēc izvēles CIDR notācijā, vai Zabbix serveru un Zabbix starpniekserveru DNS nosaukumi. Ienākošie savienojumi tiks pieņemti tikai no šeit norādītajiem hosts.
Ja ir iespējots IPv6 atbalsts, tad `127.0.0.1`, `::127.0.0.1`, `::ffff:127.0.0.1` tiek uzskatīti par ekvivalentiem, un `::/0` ļaus 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 saskarnes URL:
0 - neignorēt sertifikāta kļūdas;
1 - ignorēt sertifikāta kļūdas.

Noklusējums: 0
 Diapazons: 0-1

Iekļaut

Norādiet direktoriju vai atsevišķus failus, kas jāiekļauj konfigurācijas failā.
 Zabbix instalēšanas laikā šis parametrs tiek iestatīts uz `/usr/local/etc`, ja vien to nemaināt kompilēšanas laikā.

Vispārīgie noteikumi parametram `Include`:

- Ceļš var būt relatīvs attiecībā pret `zabbix_web_service.conf` faila atrašanās vietu.
- Vienā konfigurācijas failā var norādīt vairākus `Include` parametrus.
- Zabbix apstrādā `Include` parametru uzreiz pēc tā nolasīšanas, tāpēc ieteicams to novietot konfigurācijas faila beigās.

Iekļauto failu noteikumi:

- Katram failam jābūt lasāmam.
- Katram failam jābūt ar pareizu sintaksi. Pretējā gadījumā komponents netiks palaists.

Direktoriju vai failu rakstu iekļaušanas noteikumi:

- Nepieņemiet, ka iekļaušana notiek noteiktā secībā (piemēram, faili netiek iekļauti alfabētiskā secībā).
- Nenorādiet vienu un to pašu parametru vairāk nekā vienā iekļautajā failā (piemēram, lai pārrakstītu vispārīgu iestatījumu ar specifisku), jo nevar paredzēt, kura vērtība tiks izmantota.
- Pievērsiet uzmanību rezerves kopijām, ko daži teksta redaktori izveido automātiski (piemēram, rediģējot `include/my_specific.conf` var tikt izveidots `include/my_specific.conf.BAK`), jo abi faili tiks iekļauti.

Lai no direktorija iekļautu tikai atbilstošos failus, izmantojiet aizstājējzīmi (`*`) raksta atbilstībai.

Piemērs:

```
Include=/absolute/path/to/config/files/*.conf
```

ListenPort

Pakalpojums klausīsies šajā portā servera savienojumiem.

Noklusējums: 10053
 Diapazons: 1024-32767

LogFile

Žurnālfaila nosaukums.

Piemērs:

`/tmp/zabbix_web_service.log`

Obligāts: Jā, ja LogType ir iestatīts uz `file`; citādi nē

LogFileSize

Maksimālais žurnāla faila lielums MB.
0 - atspējot automātisku žurnāla rotāciju.
Piezīme: Ja tiek sasniegts žurnāla faila lieluma ierobežojums un failu rotācija kāda iemesla dēļ neizdodas, esošais žurnāla fails tiek saīsināts un izveidots no jauna.

Noklusējums: 1
 Diapazons: 0-1024

LogType

Žurnāla izvades tips: - *file* - ierakstīt žurnālu failā, kas norādīts ar parametru LogFile; - *system* - ierakstīt žurnālu syslog; - *console* - ierakstīt žurnālu standarta izvadei.

Noklusējums: `file`

Noildze

Maksimālais laiks (sekundēs), kas tiek pavadīts, formatējot paneļa PDF atskaiti.

Noklusējums: 10
 Diapazons: 1-30

TLSAccept

Kādas ienākošās savienojumus pieņemt: - *unencrypted* - pieņemt savienojumus bez šifrēšanas (noklusējums); - *cert* - pieņemt savienojumus ar TLS un sertifikātu.

Noklusējums: `unencrypted`

TLSCAFile

Fails, kurā ir augstākā līmeņa CA sertifikāti vienaudžu sertifikātu verifikācijai, kas tiek izmantota šifrētai saziņai starp Zabbix komponentiem.

TLSCertFile

Pilns ceļa nosaukums uz failu, kurā atrodas pakalpojuma sertifikāts vai sertifikātu ķēde, kas tiek izmantota šifrētai saziņai ar Zabbix komponentēm.

TLSKeyFile

Fails, kas satur pakalpojuma privāto atslēgu, pilnais ceļš, ko izmanto šifrētai saziņai starp Zabbix komponentēm.

Bridge adapter

Zabbix bridge adapter ļauj jūsu Zabbix serverim sazināties ar Zabbix mobilo lietotni.

Zabbix mobilā lietotne neveido tiešu savienojumu ar jūsu Zabbix serveri. Tā vietā dati starp tiem tiek pārsūtīti, izmantojot divus komponentus:

- Zabbix Bridge — centralizēti mitināts pakalpojums, kas apmainās ar datiem ar Zabbix mobilo lietotni jūsu mobilajā ierīcē.
- Zabbix bridge adapter — komponents, ko instalējat un palaižat. Tas savieno jūsu Zabbix serveri ar Zabbix Bridge.

Bridge adapter var instalēt tajā pašā sistēmā, kurā darbojas jūsu Zabbix serveris, vai citā sistēmā, izmantojot pakotnes.

Lai konfigurētu Zabbix bridge adapter, atjauniniet tā konfigurācijas failu (noklusējuma ceļš: `/etc/zabbix/zabbix_bridge_adapter.conf`

Lai nodrošinātu Zabbix bridge adapter drošību, skatiet sadaļu Droša piekļuve mobilajām ierīcēm.

Datu plūsma Izmantojot Zabbix mobilo lietotni, dati tiek pārsūtīti šādi:

1. Jūs veicat darbību Zabbix mobilajā lietotnē, kas pēc tam nosūta vaicājumu uz Zabbix Bridge.
2. Zabbix Bridge pārsūta vaicājumu Zabbix bridge adapterim kā JSON-RPC pieprasījumu.
3. Zabbix bridge adapteris pārbauda, vai pieprasītā Zabbix API metode ir atļauta.
   Ja metode nav atļauta, bridge adapteris noraida pieprasījumu.
   Ja metode ir atļauta, bridge adapteris nosūta pieprasījumu uz Zabbix API.
4. Zabbix API apstrādā pieprasījumu un atgriež atbildi Zabbix bridge adapterim.
5. Zabbix bridge adapteris atgriež atbildi Zabbix Bridge.
6. Zabbix Bridge pārsūta atbildi Zabbix mobilajai lietotnei, kas parāda datus jūsu mobilajā ierīcē.

Konfigurācijas parametri: Zabbix bridge adapteris

Pārskats

Konfigurācijas parametri ļauj pielāgot Zabbix bridge adapter darbību.

Norādiet parametru vērtības konfigurācijas failā (`zabbix_bridge_adapter.conf`). Visi parametri ir **neobligāti**, ja vien nav skaidri norādīts, ka parametrs ir obligāts.

Šī lapa ir strukturēta šādi:

- Parametru pārskats (lai skatītu pilnu informāciju par parametru, noklikšķiniet uz tā nosaukuma)
- Parametru detalizēts apraksts

Parametrs	Apraksts
AllowedAPIMethods	Norādiet, kuras Zabbix API metodes var izsaukt, izmantojot Zabbix bridge adapter.
DataDir	Norādiet direktoriju, kurā tiek izveidots un glabāts bridge adapter stāvokļa fails un šifrēšanas atslēgas.
DebugLevel	Iestatiet atkļūdošanas līmeni.
Include	Norādiet direktoriju vai atsevišķus failus, kas jāiekļauj konfigurācijas failā.
ListenIP	Norādiet bridge adapter IP adresi, kurā tas klausās ienākošos HTTP(S) savienojumus no Zabbix servera.
ListenHealthPort	Norādiet bridge adapter portu, kurā tas klausās ienākošos HTTP savienojumus veselības pārbaudei.
ListenPort	Norādiet bridge adapter portu, kurā tas klausās ienākošos HTTP(S) savienojumus no Zabbix servera.
LogFile	Norādiet pilnu ceļu uz bridge adapter žurnālfailu.
LogFileSize	Iestatiet maksimālo žurnālfaila lielumu (MB).
LogType	Norādiet žurnāla izvades tipu.

Parametrs	Apraksts
PidFile	Norādiet PID failu.
TLSConfig.AllowedCN	Norādiet ar komatiem atdalītu atļauto klientu Common Names sarakstu (tikai `mTLS`).
TLSConfig.CAFile	Norādiet pilnu ceļu uz failu, kurā ir augstākā līmeņa CA sertifikāts(-i), kas tiek izmantots(-i), lai pārbaudītu klientu sertifikātus `mTLS` režīmā.
TLSConfig.CertFile	Norādiet pilnu ceļu uz failu, kurā ir sertifikāts vai sertifikātu ķēde.
TLSConfig.KeyFile	Norādiet pilnu ceļu uz failu, kurā ir privātā atslēga.
TLSConfig.Mode	Iestatiet TLS režīmu ienākošajiem savienojumiem portā `ListenPort`.
ServerId	Norādiet Zabbix servera unikālo identifikatoru. **Obligāts**.
ZabbixAPITLSCAFile	Norādiet pilnu ceļu uz failu, kurā ir augstākā līmeņa CA sertifikāts(-i), ko bridge adapter izmanto Zabbix API sertifikāta pārbaudei.
ZabbixAPITLSConnect	Iestatiet TLS režīmu izejošajiem savienojumiem ar Zabbix API.
ZabbixAPIURL	Norādiet Zabbix API galapunkta URL. **Obligāts**.
ZabbixBridgeURL	Norādiet Zabbix Bridge URL Zabbix Bridge konfigurācijas izgūšanai (tā tiek izgūta, kad bridge adapter tiek palaists).

Ņemiet vērā:

- Noklusējuma vērtības atspoguļo procesa noklusējuma vērtības, nevis vērtības piegādātajos konfigurācijas failos.
- 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

AllowedAPIMethods

Norādiet, kuras Zabbix API metodes var izsaukt, izmantojot Zabbix bridge adapter.
 Vairākām metodēm atkārtojiet parametru, katrai metodei izmantojot atsevišķu rindu.

Katra parametra vērtībai ir jābūt vai nu precīzam metodes nosaukumam (piemēram, `problem.get`), vai aizstājējzīmes paraugam, kas aizstāj tikai visu API nosaukumu (piemēram, `*.get`) vai visu metodes nosaukumu (piemēram, `problem.*`).

Jebkura cita vērtība nav derīga un neļauj Zabbix bridge adapter palaist. Tas ietver atsevišķu aizstājējzīmi (`*`), daļēju aizstājējzīmes paraugu (`host*` vai `host*.*`) vai tukšu vērtību (`AllowedAPIMethods=`).

Piemērs:

```
AllowedAPIMethods=apiinfo.version
AllowedAPIMethods=problem.*
AllowedAPIMethods=*.get
```

Minimālais metožu kopums, kas nepieciešams Zabbix mobilajai lietotnei:

```
AllowedAPIMethods=apiinfo.version
AllowedAPIMethods=device.onboard
AllowedAPIMethods=device.offboard
AllowedAPIMethods=user.get
AllowedAPIMethods=role.get
AllowedAPIMethods=problem.get
AllowedAPIMethods=event.get
AllowedAPIMethods=event.acknowledge
AllowedAPIMethods=trigger.get
AllowedAPIMethods=alert.get
AllowedAPIMethods=host.get
AllowedAPIMethods=hostgroup.get
```

Noklusējums: nav iestatīts (visas Zabbix API metodes ir liegtas)

DataDir

Norādiet direktoriju, kurā tiek izveidots un glabāts bridge adaptera stāvokļa fails un šifrēšanas atslēgas.
 Direktorijai ir jābūt ar 0700 atļaujām, un visiem tajā glabātajiem failiem ir jābūt ar 0600 atļaujām.

Noklusējums: `/var/lib/zabbix-bridge-adapter`

DebugLevel

Iestatiet atkļūdošanas līmeni:

- 0 - pamatinformācija par Zabbix procesu palaišanu un apturēšanu
- 1 - kritiska informācija
- 2 - informācija par kļūdām
- 3 - brīdinājumi
- 4 - atkļūdošanai (ģenerē daudz informācijas)
- 5 - paplašināta atkļūdošana (ģenerē vēl vairāk informācijas)

Noklusējums: 3
 Diapazons: 0-5

Iekļaušana

Norādiet konfigurācijas failā iekļaujamo direktoriju vai atsevišķus failus.
 Zabbix instalēšanas laikā šis parametrs tiek iestatīts uz `/usr/local/etc`, ja vien kompilēšanas laikā to nemaināt.

Vispārīgi `Include` parametra noteikumi:

- Ceļš var būt relatīvs pret `zabbix_bridge_adapter.conf` faila atrašanās vietu.
- Vienā konfigurācijas failā varat norādīt vairākus `Include` parametrus.
- Zabbix apstrādā `Include` parametru uzreiz pēc tā nolasīšanas, tāpēc apsveriet iespēju to ievietot konfigurācijas faila beigās.

Iekļaujamo failu noteikumi:

- Katram failam jābūt nolasāmam.
- Katram failam jābūt ar pareizu sintaksi. Pretējā gadījumā komponents netiks palaists.

Direktoriju vai failu modeļu iekļaušanas noteikumi:

- Neuzskatiet, ka iekļaušana notiks noteiktā secībā (piemēram, faili netiek iekļauti alfabētiskā secībā).
- Nedefinējiet vienu un to pašu parametru vairāk nekā vienā iekļautajā failā (piemēram, lai aizstātu vispārīgu iestatījumu ar konkrētu), jo nevarat paredzēt, kura vērtība tiks izmantota.
- Pievērsiet uzmanību rezerves kopijām, ko daži teksta redaktori izveido automātiski (piemēram, rediģējot `include/my_specific.conf` var tikt izveidots `include/my_specific.conf.BAK`), jo tiks iekļauti abi faili.

Lai no direktorijas iekļautu tikai atbilstošos failus, modeļu atbilstībai izmantojiet aizstājējzīmi (`*`).

Piemērs:

```
Include=/absolute/path/to/config/files/*.conf
```

ListenIP

Norādiet tilta adaptera klausīšanās IP adresi ienākošajiem HTTP(S) savienojumiem no Zabbix servera.
 Ja nav norādīta, tilta adapteris klausās visās IP adresēs.

ListenHealthPort

Norādiet tilta adaptera klausīšanās portu ienākošajiem HTTP veselības pārbaudes savienojumiem.
 Apkalpo `/health` un `/ready` galapunktus.
 Ja nav norādīts, veselības pārbaudes pakalpojums netiek palaists.

Noklusējums: 8081
 Diapazons: 1024-32767

ListenPort

Norādiet tilta adaptera klausīšanās portu ienākošajiem HTTP(S) savienojumiem no Zabbix servera.

Noklusējums: 10054
 Diapazons: 1024-32767

LogFile

Norādiet pilnu ceļu uz bridge adaptera žurnālfailu.

Obligāti: **Jā**, ja LogType ir iestatīts uz `file`

Noklusējums: `/tmp/zabbix_bridge_adapter.log`

LogFileSize

Iestatiet maksimālo žurnālfaila lielumu (MB).
 0 - atspējot automātisku žurnālfaila rotāciju.

Noklusējums: 1
 Diapazons: 0-1024

LogType

Norādiet žurnāla izvades tipu:

- `file` - rakstīt žurnālu failā, kas norādīts ar parametru `LogFile`.
- `system` - rakstīt žurnālu sistēmas žurnālā.
- `console` - rakstīt žurnālu standarta izvadē.

Noklusējums: `file`

PidFile

Norādiet PID failu.

Noklusējums: `/tmp/zabbix_bridge_adapter.pid`

ServerId

Norādiet Zabbix servera unikālo identifikatoru.
 Pieejams Zabbix lietotāja saskarnē sadaļā *Pārskati* > *Sistēmas informācija*.

Obligāts: **Jā**

TLSConfig.AllowedCN

Norādiet ar komatiem atdalītu atļauto klienta Common Name vērtību sarakstu (tikai `mTLS`).
 Ja vērtība nav norādīta, tilta adapteris nepārbauda klienta Common Name.

TLSConfig.CAFile

Norādiet pilnu ceļu uz failu, kurā ir augstākā līmeņa CA sertifikāts(-i), kas tiek izmantots(-i), lai `mTLS` režīmā pārbaudītu klientu sertifikātus.

TLSConfig.CertFile

Norādiet pilnu ceļu uz failu, kurā atrodas sertifikāts vai sertifikātu ķēde.

TLSConfig.KeyFile

Norādiet pilnu ceļu uz failu, kurā atrodas privātā atslēga.

TLSConfig.Mode

Iestatiet TLS režīmu ienākošajiem savienojumiem portā `ListenPort`:

- `none` - pieņemt savienojumus bez šifrēšanas.
- `tls` - standarta TLS ar vismaz v1.2 versiju.
- `mtls` - paplašina TLS ar savstarpēju autentifikāciju.

Noklusējums: `none`

ZabbixAPITLSCAFile

Norādiet pilnu ceļu uz failu, kurā ir augstākā līmeņa CA sertifikāts(-i), ko tilta adapteris izmanto Zabbix API sertifikāta pārbaudei.

Obligāts: **Jā**, ja `ZabbixAPITLSConnect` ir iestatīts uz `verify_ca` vai `verify_full`.

ZabbixAPITLSConnect

Iestatiet TLS režīmu izejošajiem savienojumiem ar Zabbix API:

- `disabled` (vai tukšs) - izmantot vienkāršu HTTP; `ZabbixAPIURL` jāizmanto `http://`.
- `required` - izmantot HTTPS bez sertifikāta pārbaudes (tikai testēšanai).
- `verify_ca` - izmantot HTTPS un pārbaudīt Zabbix API sertifikātu.
- `verify_full` - izmantot HTTPS un pārbaudīt Zabbix API sertifikātu un resursdatora nosaukumu.

ZabbixAPIURL

Norādiet Zabbix API galapunkta URL.

Obligāts: **Jā**

Piemērs:

`ZabbixAPIURL=https://example.com/zabbix/api_jsonrpc.php`

ZabbixBridgeURL

Norādiet Zabbix bridge URL Zabbix Bridge konfigurācijas izgūšanai (tā tiek izgūta, kad tiek startēts bridge adapteris).

Noklusējums: `ZabbixBridgeURL=https://test.aws.zabbix-lab.win/.well-known/bridge-configuration`

MCP server

Pārskats Zabbix MCP serveris ir process, kas ļauj MI aģentiem strādāt ar Zabbix, izmantojot Model Context Protocol (MCP). MCP serveris sazinās ar MCP klientiem un Zabbix API.

Izmantojot MCP serveri, MI aģents var:

- Iegūt hosts grupas, veidņu grupas, hosts, veidnes, vienumus, LLD noteikumus, saskarnes, makro, trigerus, problēmas, uzturēšanas periodus un vienumu vēstures datus (izņemot bināros vienumus);
- Atjaunināt problēmas: pievienot komentārus, apstiprināt vai atcelt apstiprinājumu, mainīt prioritāti, apspiest vai atcelt apspiešanu, kā arī aizvērt problēmas;
- Izveidot, atjaunināt un dzēst uzturēšanas periodus.

Autorizācijai tiek izmantoti Zabbix API marķieri; marķieri tiek nosūtīti ar katru pieprasījumu. Piekļuvi ierobežo marķiera īpašnieka atļaujas: ir redzami tikai dati no tām hosts grupām, kas ir pieejamas šim lietotājam, un darbības, kuras lietotājam nav atļauts veikt, tiek noraidītas.

MCP serveris ir bezstāvokļa: starp pieprasījumiem sesija netiek saglabāta.

> **Note:**
> Lai gan Zabbix MCP serveris var darboties ar vecākām vai jaunākām Zabbix versijām, tā oficiālā saderība tiek atbalstīta tikai ar to pašu Zabbix versiju, kuru izmanto pats MCP serveris.

Skatiet arī:

- Konfigurācijas parametri: MCP serveris
- Rīki un resursi

Instalēšana Oficiālā zabbix-mcp-server pakotne ir pieejama Zabbix repozitorijā.

Lai instalētu Zabbix MCP serveri no pirmkoda, izveidojiet to, izmantojot `make`.

Lai konfigurētu Zabbix MCP serveri, atjauniniet MCP servera konfigurācijas faila parametrus. Jāiestata vismaz parametri `FrontendURL` un `AllowedIP`.

Noklusējuma konfigurācijas fails (`zabbix_mcp_server.conf`) atrodas `/etc/zabbix/`; tajā ir aizliegti rīki, kas visbiežāk atklāj sensitīvus datus. Lai iespējotu kādu rīku, noņemiet atbilstošo rindu `DenyTool`.

Skatiet arī: Drošības paraugprakse

Darbināšana Lai palaistu MCP serveri, izpildiet:

```
zabbix_mcp_server -c /etc/zabbix/zabbix_mcp_server.conf
```

Lai pārbaudītu konfigurācijas failu un izietu, nepalaižot serveri, izmantojiet opciju `-T`:

```
zabbix_mcp_server -c /etc/zabbix/zabbix_mcp_server.conf -T
```

Tiek atbalstītas šādas komandrindas opcijas:

Opcija	Apraksts
`-c <file>`	Konfigurācijas faila ceļš. Noklusējuma vērtība ir `/etc/zabbix/zabbix_mcp_server.conf`.
`-T`	Validēt konfigurācijas failu un iziet.
`-h`	Parādīt palīdzības informāciju.
`-V`	Parādīt versijas numuru.

Ja MCP serveris ir instalēts no pirmkoda, systemd vienība ir jāizveido manuāli, piemēram:

```
[Unit]
Description=Zabbix MCP server
After=network-online.target

[Service]
User=zabbix
ExecStart=/usr/local/sbin/zabbix_mcp_server -c /etc/zabbix/zabbix_mcp_server.conf
Restart=on-failure

[Install]
WantedBy=multi-user.target
```

MCP klienta savienošana MCP serverim var pievienot jebkuru MCP klientu, kas atbalsta Streamable HTTP transportu. Ir nepieciešami šādi iestatījumi:

Iestatījums	Vērtība
Transports	Streamable HTTP
URL	`https://<host>:<ListenPort>/mcp`
Galvene	`Authorization: Bearer <Zabbix API token>`, nosūtīta ar katru pieprasījumu

Katram lietotājam ir jākonfigurē savs API tokens.

Lai pārbaudītu iestatījumus, nosūtiet vienu pieprasījumu MCP serverim:

```
curl -sS https://mcp.example.com:8443/mcp \
  -H "Authorization: Bearer $ZABBIX_API_TOKEN" \
  -H "Content-Type: application/json" \
  -H "Accept: application/json, text/event-stream" \
  -d '{"jsonrpc":"2.0","id":1,"method":"tools/list"}'
```

HTTP statusa kods 200 un pieejamo rīku saraksts atbildē apstiprina, ka MCP serveris ir sasniedzams un pieņem tokenu; sarakstā ir redzams, kuri rīki ir pieejami. Citi statusa kodi norāda uz šādām problēmām:

- 400 - `Accept` galvenē nav iekļauti abi transportam nepieciešamie multivides tipi;
- 401 - trūkst `Authorization` galvenes ar tokenu;
- 403 - klienta adrese nav norādīta `AllowedIP` parametrā.

Apstiprināšana un rīku piekļuve Ja MCP klients atbalsta veidlapu izsaukšanu, lietotājam ir jāapstiprina katras izmaiņas pirms tās nosūtīšanas uz Zabbix. MCP klients parāda lietotājam, kas tiks mainīts, un izmaiņa tiek lietota tikai tad, ja lietotājs to apstiprina; ja lietotājs pieprasījumu noraida vai aizver, Zabbix nekas netiek mainīts. Apstiprinājums ir derīgs piecas minūtes un tikai tieši parādītajai izmaiņai; šajā laikā apstiprinājumu var izmantot atkārtoti.

Ja MCP klients neatbalsta veidlapu izsaukšanu (tostarp klienti, kas atbalsta tikai URL izsaukšanu), izmaiņas tiek lietotas nekavējoties, bez apstiprināšanas. Šādā gadījumā ierobežojiet rakstīšanas rīkus un Zabbix API atļaujas.

Apstiprināšana ir īpaši svarīga, jo rīka izvade var saturēt neuzticamu tekstu no uzraudzītajām sistēmām, kas var ietekmēt aģentu.

Izmantojiet AllowTool, DenyTool, AllowToolRegexp un DenyToolRegexp, lai kontrolētu, kurus rīkus MCP serveris padara pieejamus. Liegtos rīkus nevar izsaukt vai uzskaitīt, un to saistītajiem resursiem nevar piekļūt.

Warning:
AllowTool, DenyTool, AllowToolRegexp un DenyToolRegexp noteikumi ierobežo MCP piekļuvi, bet nemaina Zabbix pilnvaras marķiera atļaujas. Ja lietotāja saskarnes adrese ir pieejama aģentam, aģents var mēģināt nosūtīt tiešus API pieprasījumus. Nodrošiniet, lai Zabbix lietotāja saskarnes atrašanās vieta un MCP servera konfigurācija nebūtu pieejama aģentam, tādējādi pieprasījumiem būs jāiet caur MCP serveri un tā drošības vadīklām.

Drošības paraugprakse API marķieri

Autentifikāciju un autorizāciju pilnībā nodrošina Zabbix. MCP klients ar katru pieprasījumu nosūta Zabbix API marķieri; MCP serveris to pārsūta, neuzglabājot un nereģistrējot, savukārt Zabbix validē marķieri, piemēro atļaujas un reģistrē darbības audita žurnālā.

Katram aģentam un lietotājam izmantojiet atsevišķu, ar minimālajām nepieciešamajām privilēģijām apveltītu marķieri ar derīguma termiņu. Glabājiet marķierus drošā vietā un regulāri nomainiet API marķierus.

TLS šifrēšana

Lai novērstu tokenu nosūtīšanu nešifrētā veidā, iestatiet `TLSAccept=cert` un norādiet sertifikātu un atslēgu (`TLSCertFile` un `TLSKeyFile`) vai palaidiet MCP serveri aiz uzticama TLS terminējoša reversā starpniekservera un iestatiet `TLSAccept=unencrypted`. Nešifrēts HTTP ir pieļaujams tikai uzticamā tīklā.

Piekļuves ierobežojumi

Ierobežojiet parametru `AllowedIP`, atļaujot tikai uzticamu MCP klientu adreses, un izmantojiet Zabbix atļaujas, lai ierobežotu API pilnvaru iespējas.

MCP serveris nenodrošina pieprasījumu ātruma ierobežošanu. Katrs MCP pieprasījums parasti izraisa vienu Zabbix API izsaukumu, tomēr cilpā darbojošs aģents joprojām var radīt ievērojamu slodzi.

Sensitīvu datu aizsardzība

Vienumu atslēgas var saturēt akreditācijas datus (piemēram, SSH, JMX, ODBC un HTTP aģenta vienumos), makro vērtības var saturēt paroles un kopienas virknes, saskarnes atklāj iekšējās adreses, bet žurnāla vai teksta vienumu vērtības var saturēt jebkādus datus. Šī iemesla dēļ rīki `item_get`, `lld_get`, `interface_get`, `macro_get` un `history_get` piegādes konfigurācijas failā pēc noklusējuma ir liegti.

Lai nepieļautu sensitīvu makro vērtību atklāšanu pievienotā aģenta LLM pakalpojumu sniedzējam, izmantojiet slepenos makro; to vērtības API nekad neatgriež.

Konfigurācijas parametri: Zabbix MCP serveris

Pārskats

Zabbix MCP serveris ir process, kas ļauj AI aģentiem strādāt ar Zabbix, izmantojot Model Context Protocol (MCP).

Šajā sadaļā ir uzskaitīti Zabbix MCP servera konfigurācijas faila (`zabbix_mcp_server.conf`) atbalstītie parametri.

Parametri ir uzskaitīti bez papildu informācijas. Noklikšķiniet uz parametra, lai skatītu pilnu informāciju.

Parameter	Description
AllowedIP	Ar komatiem atdalītu IP adrešu (pēc izvēles CIDR notācijā) vai MCP klientu DNS nosaukumu saraksts.
AllowTool	Aizstājējzīmes noteikums, kas atļauj rīku.
AllowToolRegexp	Regulārās izteiksmes noteikums, kas atļauj rīkus.
DebugLevel	Atkļūdošanas līmenis.
DenyTool	Aizstājējzīmes noteikums, kas liedz izmantot rīku.
DenyToolRegexp	Regulārās izteiksmes noteikums, kas liedz izmantot rīkus.
FrontendURL	Zabbix API galapunkts.
IgnoreURLCertErrors	Norāda TLS sertifikāta validācijas kļūdu apstrādi, piekļūstot Zabbix API.
ListenIP	IP adrese, kurā klausīties.
ListenPort	Ports, kurā klausīties.
LogFile	Žurnālfaila nosaukums.
LogFileSize	Žurnālfaila maksimālais lielums.
LogType	Žurnāla izvades veids.
MaxResponseSize	Atbildes maksimālais lielums, ko atgriež lasīšanas rīks.
PidFile	PID faila nosaukums.
Timeout	Maksimālais laiks (sekundēs), ko gaidīt atbildi no Zabbix API.
TLSAccept	Kādus ienākošos savienojumus pieņemt.
TLSCertFile	Pilns ceļš uz failu, kurā atrodas servera sertifikāts vai sertifikātu ķēde.
TLSKeyFile	Pilns ceļš uz failu, kurā atrodas servera privātā atslēga.
ToolPrefix	Prefikss, kas pievienots visu rīku nosaukumiem.

Visi parametri nav obligāti, ja vien nav skaidri norādīts, ka parametrs ir obligāts.

Ņemiet vērā:

- Noklusējuma vērtības atspoguļo procesa noklusējuma iestatījumus, nevis vērtības piegādātajos konfigurācijas failos;
- Vērtības atbalsta vides mainīgos formātā `${NAME}`;
- Zabbix atbalsta konfigurācijas failus tikai UTF-8 kodējumā bez BOM;
- Komentāri, kas sākas ar #, tiek atbalstīti tikai rindas sākumā.

Parametru informācija

AllowedIP

Ar komatiem atdalīts IP adrešu saraksts, pēc izvēles CIDR notācijā, vai MCP klientu DNS nosaukumi. Ienākošie savienojumi tiks pieņemti tikai no šeit norādītajiem hostiem.

Piemērs:

```
AllowedIP=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,mcp-client.example.com
```

Obligāts: jā

AllowTool

Aizstājējzīmes noteikums, kas atļauj rīku. Parametru var norādīt vairākas reizes.
 Aizstājējzīmju raksti izmanto vienuma atslēgas sintaksi, kur * atbilst jebkādām rakstzīmēm. Noteikums tiek piemērots vienkāršajam rīka nosaukumam un ignorē ToolPre-

fix.
 Visi `AllowTool`, `DenyTool`, `AllowToolRegexp` un `DenyToolRegexp` noteikumi veido vienu sarakstu to norādīšanas secībā; lēmumu pieņem pirmais atbilstošais noteikums. Rīks, kuram neatbilst neviens noteikums, ir atļauts.

Piemērs:

```
AllowTool=host_get
```

AllowToolRegexp

Regulārās izteiksmes noteikums, kas atļauj rīkus. Parametru var norādīt vairākas reizes.
 Regulārās izteiksmes izmanto RE2 sintaksi un atbilst jebkurai vienkāršā rīka nosaukuma daļai, ja vien izteiksme nav piesaistīta virknes sākumam vai beigām.

Piemērs:

```
AllowToolRegexp=^(host|hostgroup|trigger|problem)_get$
```

DebugLevel

Norādiet atkļūdošanas līmeni: - *0* - pamatinformācija par Zabbix procesu palaišanu un apturēšanu; - *1* - kritiska informācija; - *2* - informācija par kļūdām; - *3* - brīdinājumi; - *4* - atkļūdošanai (ģenerē daudz informācijas); - *5* - paplašinātā atkļūdošana (ģenerē vēl vairāk informācijas).

Noklusējums: 3
 Diapazons: 0-5

DenyTool

Aizstājējzīmes noteikums, kas liedz izmantot rīku. Parametru var norādīt vairākas reizes.
 Liegtie rīki netiek iekļauti rīku sarakstā, un tos nevar izsaukt. Detalizētu informāciju skatiet sadaļā AllowTool.

Piemērs:

```
DenyTool=maintenance_delete
```

DenyToolRegexp

Regulāras izteiksmes noteikums, kas liedz rīku izmantošanu. Parametru var norādīt vairākas reizes.
 Detalizētu informāciju skatiet sadaļā AllowToolRegexp.

Piemērs:

```
DenyToolRegexp=^problem_(close|suppress)$
```

FrontendURL

Zabbix API galapunkts: Zabbix lietotāja saskarnes URL, kas beidzas ar `api_jsonrpc.php`.

Piemērs:

```
FrontendURL=https://zabbix.example.com/api_jsonrpc.php
```

Obligāts: jā

IgnoreURLCertErrors

Norāda TLS sertifikāta validācijas kļūdu apstrādi, piekļūstot Zabbix API:
 false - neignorēt sertifikāta kļūdas;
 true - ignorēt sertifikāta kļūdas (izmantot tikai izstrādes vidē).

Noklusējuma vērtība: `false`

ListenIP

IP adrese, kurā klausīties.

Noklusējums: `0.0.0.0`

ListenPort

Ports, kurā klausīties. MCP galapunkta ceļš ir `/mcp`.

Noklusējums: 8443
 Diapazons: 1024-32767

LogFile

Žurnālfaila nosaukums.

Noklusējums: `/tmp/zabbix_mcp_server.log`

Obligāts: Nē

LogFileSize

Maksimālais žurnālfaila lielums MB.
0 - atspējot automātisku žurnālfaila rotāciju.

Noklusējums: 1
 Diapazons: 0-1024

LogType

Žurnāla izvades veids: - *file* - rakstīt žurnālu failā, kas norādīts ar parametru LogFile; - *system* - rakstīt žurnālu sistēmas žurnālā; - *console* - rakstīt žurnālu standarta izvadē.

Noklusējums: `file`

MaxResponseSize

Maksimālais atbildes lielums, ko atgriež lasīšanas rīks. Atbildes izveide tiek pārtraukta, tiklīdz tiek sasniegts šis ierobežojums vai rīka argumenta `limit` vērtība — atkarībā no tā, kurš nosacījums tiek sasniegts pirmais.
 Atbalsta mērvienību sufiksus.

Noklusējums: `64K`
 Diapazons: 16K-256K

PidFile

PID faila nosaukums.

Noklusējums: `/tmp/zabbix_mcp_server.pid`

Taimauts

Maksimālais laiks (sekundēs), cik ilgi gaidīt atbildi no Zabbix API.

Noklusējums: 10
 Diapazons: 3-30

TLSAccept

Kādus ienākošos savienojumus pieņemt: - *unencrypted* - pieņemt savienojumus bez šifrēšanas (noklusējums); - *cert* - pieņemt savienojumus ar TLS un sertifikātu.

Noklusējums: `unencrypted`

TLSCertFile

Pilns ceļš uz failu, kurā atrodas servera sertifikāts vai sertifikātu ķēde.

Obligāts: Jā, ja `TLSAccept` ir iestatīts uz `cert`; pretējā gadījumā — nē

TLSKeyFile

Pilns faila ceļš, kurā atrodas servera privātā atslēga.

Obligāts: Jā, ja `TLSAccept` ir iestatīts uz `cert`; pretējā gadījumā — nē

ToolPrefix

Prefikss, kas tiek pievienots visu rīku nosaukumiem, piemēram, lai izvairītos no nosaukumu konfliktiem ar citu MCP serveru rīkiem. Piemēram, ja `ToolPrefix=zbx_`, rīks `host_get` tiek eksponēts kā `zbx_host_get`.
 Atļautās rakstzīmes: burti, cipari, pasvītrojums (`_`), defise (`-`) un punkts (`.`).
 Rīku piekļuves noteikumi atbilst rīku nosaukumiem bez prefiksa.

Piemērs:

`ToolPrefix=zbx_`

Rīki un resursi

Pārskats

Zabbix MCP serveris nodrošina Zabbix funkcionalitāti MCP klientiem kā *rīkus* un *resursus*.

Rīku nosaukumi apraksta darbību. Ja ir iestatīts parametrs ToolPrefix, prefikss tiek pievienots katram rīka nosaukumam; rīku piekļuves noteikumi atbilst tālāk norādītajiem vienkāršajiem nosaukumiem.

Lasīšanas rīki

Rīks	Zabbix API metode	Argumenti	Kārtots pēc	Resurss
`hostgroup_get`	hostgroup.get	`groupids`, `names`, `search`, `limit`	name	`zabbix://hostgroup/<i`
`templategroup_get`	templategroup.get	`groupids`, `names`, `search`, `limit`	name	`zabbix://templategrou`
`host_get`	host.get	`hostids`, `groupids`, `search`, `limit`	name	`zabbix://host/<id>`
`template_get`	template.get	`templateids`, `groupids`, `search`, `limit`	name	`zabbix://template/<id`
`item_get`	item.get	`hostids`, `itemids`, `search`, `limit`	name	`zabbix://item/<id>`

Rīks	Zabbix API metode	Argumenti	Kārtots pēc	Resurss
lld_get	discoveryrule.get	hostids, ruleids, limit	name	-
interface_get	hostinterface.get	hostids, interfaceids, limit	interface ID	-
macro_get	usermacro.get	hostids vai global, macro_names, limit	macro	-
problem_get	problem.get	eventids, groupids, hostids, severities, acknowledged, suppressed, recent, limit	event ID, dilstoši	-
maintenance_get	maintenance.get	maintenanceids, groupids, hostids, limit	name	-
history_get	history.get	itemids, time_from, time_till, type, limit	clock, dilstoši	-
trigger_get	trigger.get	triggerids, hostids, search, limit	description	zabbix://trigger/<id>

Ņemiet vērā:

- host_get un template_get atgriež arī katra objekta trigerus (ierobežotu lauku kopu), tagus un saistītās veidnes. Vienumi, atklāšanas noteikumi, saskarnes un makro tiek atgriezti, izmantojot atbilstošos rīkus.
- history_get ir nepieciešams laika diapazons, kas nav garāks par 7 dienām un sākas ne agrāk kā pirms 7 dienām. Argumentam type jābūt vienam no float, unsigned, character, text, log vai json, un tam jāatbilst vienumu vērtības tipam; ja tips ir nepareizs, tiek atgriezts tukšs rezultāts, ko nevar atšķirt no vienuma bez datiem. Bināro vienumu (vērtības tips 5) vēsturi nevar izgūt, izmantojot MCP serveri.
- Lasīšanas rīki pieprasa fiksētu lauku sarakstu; visi lauki (output: extend) nekad netiek pieprasīti.
- To rīku rezultāti, kuriem ir resurss, satur lauku uri, lai aģents varētu pāriet no saraksta uz atsevišķu entītiju.

Atbildes formāts

Visi lasīšanas rīki atgriež atbildi tādā pašā formātā:

```
{
  "results": [ ... ],
  "count": 2,
  "truncated": true,
  "truncated_by": "count",
  "limit": 2,
  "note": "The result count limit was reached ..."
}
```

Atbildi ierobežo arguments limit (noklusējuma vērtība: 100, maksimālā vērtība: 1000) un atbildes serializētais lielums (MaxResponseSize, noklusējuma vērtība: 64K). Atbildes veidošana tiek pārtraukta, kad tiek sasniegts kāds no šiem ierobežojumiem. Atbilde tiek saīsināta tikai starp rezultātiem, nekad rezultāta vidū, un saīsināta atbilde norāda, kurš ierobežojums tika sasniegts. Katram rīkam ir noteikta kārtošanas secība, tāpēc saīsināta atbilde vienmēr ir pilnas rezultātu kopas sākums. Lapu veidošana netiek atbalstīta; lai iegūtu citus rezultātus, sašauriniet vaicājumu.

Problēmu rīki

Visi problēmu rīki izsauc event.acknowledge ar atbilstošajiem darbības karodziņiem.
Arguments eventids ir obligāts; notikumu ID var izgūt, izmantojot problem_get.
Izmantojot jebkuru problēmu rīku, var pievienot ziņojumu (komentāru).
Arguments message nav obligāts, izņemot problem_annotate, kur tas ir obligāts; ja tas ir norādīts, ziņojums tiek reģistrēts tajā pašā Zabbix API izsaukumā.

Rīks	Darbība	Papildu argumenti
problem_acknowledge	Apstiprināt problēmu.	-
problem_unacknowledge	Noņemt apstiprinājumu.	-
problem_close	Manuāli aizvērt problēmu (tikai trigeriem, kas atļauj manuālu aizvēršanu).	-
problem_annotate	Reģistrēt ziņojumu.	message (obligāts)
problem_change_severity	Mainīt problēmas nopietnības pakāpi.	severity (0-5)
problem_suppress	Apturēt problēmas paziņojumus līdz norādītajam laikam vai uz nenoteiktu laiku.	suppress_until
problem_unsuppress	Noņemt paziņojumu apturēšanu.	-

Problēmas nevar izveidot vai dzēst; visi problēmu rīki atjaunina esošās problēmas.

Uzturēšanas rīki

Rīks	Zabbix API metode	Argumenti
`maintenance_create`	maintenance.create	`name`, `groupids` un/vai `hostids`, `active_since`, `active_till`, `timeperiods`, `description`, `tags`, `tags_evaltype`
`maintenance_update`	maintenance.update	`maintenanceid` un jebkurš no iepriekš minētajiem argumentiem. Saraksta argumenti aizstāj visu vērtību kopu; tukšs saraksts šo kopu notīra.
`maintenance_delete`	maintenance.delete	`maintenanceids`

Atjauninot uzturēšanas periodu, saraksta argumenti (piemēram, `hostids`, `groupids` un `timeperiods`) aizstāj esošās vērtības, nevis pievieno tām jaunas. Piemēram, ja uzturēšanas periods attiecas uz hostiem A un B un `maintenance_update` tiek izsaukts ar `hostids`, kurā ir tikai hosts C, uzturēšanas periods attieksies tikai uz hostu C.

`maintenance_create` ir vienīgā darbība, kas nav idempotenta, bet `maintenance_delete` ir vienīgā darbība, kuru nevar atsaukt. Lai novērstu šo darbību izmantošanu, aizliedziet šos rīkus ar parametru DenyTool.

Rakstīšanas rīku darbība

Problēmu un uzturēšanas rīki ir *rakstīšanas rīki*. Tie darbojas šādi:

- Argumenti tiek validēti pirms jebkura Zabbix API izsaukuma; ja validācija neizdodas, kļūdas ziņojumā tiek norādīts nosacījums, kas netika izpildīts.
- Ja klients to atbalsta, izmaiņa tiek lietota tikai pēc tam, kad lietotājs to ir apstiprinājis.
- Izmaiņu veic Zabbix, izmantojot izsaucēja tokenu. Zabbix pārbauda atļaujas un reģistrē izmaiņu audita žurnālā, piesaistot to tokena īpašniekam.
- Zabbix kļūdas, tostarp atļauju kļūdas, tiek atgrieztas nemainītas kā kļūdas objekts ar laukiem `code`, `message` un `data`.
- Rakstīšanas rīki tieši atgriež Zabbix rezultātu; lasīšanas rīku atbildes ierobežojumi uz tiem neattiecas.
- MCP serveris nekad nemēģina atkārtoti izpildīt rakstīšanas izsaukumus.

Resursi

Resursi izmanto `zabbix://<kind>/<id>` URI shēmu.
Atbilde `resources/list` ietver tikai `zabbix://about`; entītiju ID nevar uzskaitīt, un tos iegūst no rīku rezultātiem.
Atbilde `resources/templates/list` uzskaita entītiju veidus.

URI	Saturs
`zabbix://about`	API URL, lietotāja saskarnes URL, Zabbix versija, identitāte, kurai tiek atrisināts izsaucēja tokens, un eksponēto rīku saraksts.
`zabbix://host/<hostid>`	Hosts bez vienumiem, trigeriem, saskarnēm un makro.
`zabbix://hostgroup/<groupid>`	Hostu grupas ieraksts bez dalībnieku saraksta.
`zabbix://template/<templateid>`	Veidnes ieraksts bez vienumiem, trigeriem un atklāšanas noteikumiem.
`zabbix://templategroup/<groupid>`	Veidņu grupas ieraksts bez dalībnieku saraksta.
`zabbix://item/<itemid>`	Vienuma definīcija bez pašreizējās vērtības.
`zabbix://trigger/<triggerid>`	Trigera definīcija bez hostiem, atkarībām, funkcijām un tagiem.

Resurss, kuram atbilstošais rīks ir liegts, nepastāv.
Neeksistējoša entītija un entītija, kurai tokena īpašniekam nav piekļuves, abos gadījumos tiek norādīta kā *resource not found*; šie gadījumi tiek apzināti padarīti neatšķirami.
Piekļuves rīku noteikumi neietekmē `zabbix://about` resursu.

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 iestatījumu kombinācijām.

Vienkāršākajā gadījumā, iestatot Zabbix servera DebugLevel konfigurācijas parametra vērtību kā vides mainīgo, jūs pēc tam varat to izmantot servera konfigurēšanai startēšanas laikā:

```
### Zabbix servera konfigurācijas fails:
DebugLevel=${NEW_DEBUG_LEVEL}

### Zabbix servera palaišana:
NEW_DEBUG_LEVEL=5 /usr/sbin/zabbix_server
```

Vides mainīgie tiek atbalstīti šādos Zabbix komponentos:

- Serveris
- Starpniekserveris
- Aģents (UNIX vai Windows)
- Aģents 2 (UNIX vai Windows), tostarp spraudņi
- Tīmekļa pakalpojums
- Zabbix sender (izmantojot -c, --config opciju)

Svarīgas piezīmes

- Ja konfigurācijas parametrs ir iestatīts uz vides mainīgo, kas nav norādīts, palaižot komponentu, tiek izmantota parametra noklusējuma vērtība.
- Izmantojot izpildlaika komandas (piem., lai palielinātu aģenta žurnāla līmeni), ir jānorāda visi iepriekš izmantotie vides mainīgie. Tas ir tāpēc, ka Zabbix komponenti izpildlaika komandu veikšanai izmanto savu konfigurācijas failu; ja vides mainīgie tiek izlaisti, tiks izmantotas konfigurācijas parametru noklusējuma vērtības. Skatiet Piemēri.
- `userparameter_reload` izpildlaika komanda neatbalsta vides mainīgo pārlādi. Pārlādes laikā mainīgie tiek ignorēti, un tiek pārlādēti tikai parametri ar parastām vērtībām.
- Procesa pašreizējie vides mainīgie, kas tika izmantoti konfigurācijas failos, tiek notīrīti pēc Zabbix komponenta palaišanas. Tas nodrošina, ka bērnprocesi (piem., attālinātie skripti, ko izpilda Zabbix) nevar piekļūt šiem mainīgajiem. Tomēr ņemiet vērā, ka procesa sākotnējos mainīgos joprojām var izgūt (piem., izmantojot failu `/proc/<PID>/environ`).

Sintakse Vides mainīgajiem jāizmanto šāda sintakse: `${burtcipari/apakšsvītras}`.

Mainīgā nosaukumā drīkst būt tikai burti (a-z, A-Z), apakšsvītras (_), un cipari (0-9), un tas nedrīkst sākties ar ciparu.

Mainīgie, kas neatbilst prasītajai sintaksei vai ir apvienoti ar parastu vērtību, tiks uzskatīti par parastām vērtībām, kas var izraisīt kļūdas.

Pareiza mainīgā sintakse:

```
DebugLevel=${NEW_DEBUG_LEVEL}
Hostname=${ZBX_HOSTNAME}
LogFile=${LogFile_001}
```

Nepareiza mainīgā sintakse:

```
DebugLevel=${5_DebugLevel}
Hostname=${ZBX.HOSTNAME 1}
LogFile=/${HOME}/zabbix/zabbix_server.log
```

> **Note:**
> Operētājsistēmā Windows vides mainīgo nosaukumi nav reģistrjutīgi.

Piemēri Tālāk redzamie piemēri parāda, kā konfigurēt un izmantot vides mainīgos ar Zabbix komponentiem.

1. piemērs: 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 izpildlaika komandu izpildei izmanto savu konfigurācijas failu; ja vides mainīgie netiek norādīti, tiks izmantotas konfigurācijas parametru noklusējuma vērtības.

Varat arī pēc vides mainīgo iestatīšanas aģenta konfigurācijas failā padarīt tos pieejamus procesiem (piemēram, izmantojot komandu `export`). Tas samazina neparedzētas darbības risku, ko var izraisīt trūkstoši vai nepareizi iestatīti mainīgie.

```
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
```

2. piemērs: Zabbix aģenta konfigurēšana konteineram

Ja veidojat un konfigurējat savu pielāgoto Zabbix komponentu (piemēram, Zabbix aģenta) attēlu, 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), izmantojot vides mainīgos:

```
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 atkārtotu ielādi. Atkārtotas ielādes laikā mainīgie tiek ignorēti, un tiek atkārtoti ielādēti tikai parametri ar parastām vērtībām.

Konfigurācija

Pārskats Zabbix konfigurācija ietver to, kuri hosts un sistēmas tiks uzraudzīti, kādi dati tiks vākti un kā tiks piegādāti paziņojumi, kad rodas problēmas.

Izmantojiet sānjoslu, lai pārietu uz attiecīgajiem konfigurācijas aspektiem.

Galvenie konfigurācijas uzdevumi

- Iestatiet jaunu hostu vai hostu grupu → Hosts and host groups
- Definējiet, kādus datus vākt no hostiem → Vienumi
- Izveidojiet noteikumus problēmu noteikšanai → Problēmu noteikšana ar trigeriem
- Konfigurējiet, kā tiek piegādāti paziņojumi → Paziņojumi notikumu laikā
- Attēlojiet monitoringa datus, izmantojot paneļus, grafikus un kartes → Vizualizācija
- Atkārtoti izmantojiet konfigurācijas vairākos hostos → Veidnes un veidņu grupas, Noklusējuma veidnes
- Kontrolējiet lietotāju piekļuvi un atļaujas → Lietotāji un lietotāju grupas

Konfigurācijas labākā prakse Konfigurācijas izvēles ietekmē veiktspēju un uzturējamību. Lai gan katra Zabbix konfigurācija ir atšķirīga, šos labākās prakses ieteikumus var izmantot kā pamatu un pielāgot pēc vajadzības.

Atsevišķs hosts katrai uzraudzības instancei

Hosts Zabbix ir loģiska vienība, nevis fiziska iekārta vai ierīce. Izveidojiet atsevišķu hosts katrai neatkarīgai instancei, piemēram, virtuālajai mašīnai, datubāzei, konteineram vai tīkla komutatoram.

Šī pieeja nodrošina:

1. Tīrākus uzraudzības datus ar atsevišķiem vienumiem, trigeriem un brīdinājumu paziņojumiem katram hosts.
2. Detalizētu lietotāju piekļuves kontroli, izmantojot user-roles, kas atbilst mazāko privilēģiju principam.

Izmantojiet veidnes un masveida atjaunināšanu līdzīgiem hostiem

Līdzīgiem hostiem, piemēram, *Network switch 1* un *Network switch 2*, varat izveidot veidni ar nepieciešamajiem metrikiem (*Network switch template*) un grupēt hostus hostu grupā (*Network switches*).

Pievienojot jaunus slēdžus, varat izmantot masveida atjaunināšanu sadaļā *Data Collection* > *Hosts*, lai saistītu veidni, nevis klonētu hostus manuāli.

Konfigurējiet galveno un atkarīgos vienumus, lai samazinātu pieprasījumu skaitu

Lai samazinātu pieprasījumu skaitu uz mērķa entītiju, konfigurējiet galveno vienumu, kas apkopo datus vienā pieprasījumā, un pēc tam izmantojiet atkarīgos vienumus ar pirmapstrādi, lai iegūtu konkrētas vērtības.

Piemēram, galvenais vienums varētu apkopot JSON vai XML atbildi ar vairākiem rādītājiem vai izpildīt datubāzes vaicājumu, kas atgriež vairākas kolonnas (piem., atvērto savienojumu skaitu, pārtraukto savienojumu skaitu, maksimāli atļauto vienlaicīgo savienojumu skaitu un kopējo uzkrāto savienojumu skaitu kopš palaišanas), bet atkarīgie vienumi parsēs un saglabās katru vērtību atsevišķi.

Pēc datu apkopošanas atmetiet galvenā vienuma vēsturi un saglabājiet tikai atkarīgo vienumu datus.

Ja nepieciešams, izvietojiet Zabbix starpniekserveri

Ja visi jūsu hosti atrodas tajā pašā lokālajā tīklā, kur Zabbix serveris, un nav bažu par mērogojamību vai veiktspēju, starpniekserveris var nebūt nepieciešams. Lielākās vai sarežģītākās vidēs starpniekserveris nodrošina vienmērīgāku slodzes sadalījumu.

Pievienojiet Zabbix starpniekserveri, ja:

1. Izmantojat dažādas metriku apkopošanas metodes, lai uzraudzītu vairākus hostus aiz ugunsmūra. Starpniekserveris apkopo datus no hostiem un pārsūta tos Zabbix serverim, samazinot nepieciešamo atvērto ugunsmūra portu skaitu.
2. Uzraugāt attālas atrašanās vietas, filiāles vai tīklus. Attālās atrašanās vietās izvietotie Zabbix starpniekserveri turpina datu apkopošanu tīkla pārtraukumu laikā un nosūta datus atpakaļ, kad savienojums tiek atjaunots.
3. Pārvaldāt liela mēroga izvietojumus. Ja jums ir daudz hostu un/vai tiek apkopots liels vērtību skaits sekundē, izvietojiet starpniekserveri, lai samazinātu servera slodzi un uzlabotu veiktspēju.

Konfigurēt neatkarīgu slepeno makro izgūšanu

Jūs varat izmantot secret lietotāja makro kā slepenā teksta vai slepenā glabātuves makro.

Pēc noklusējuma slepeno makro vērtības izgūst Zabbix serveris un tās tiek nodotas starpniekserveriem. Lai uzlabotu drošību ar slepenās glabātuves makro, konfigurējiet makro vērtības, lai tās tiktu izgūtas neatkarīgi no Zabbix servera un starpniekserveriem.

Veidnes konfigurēšana

Pārskats

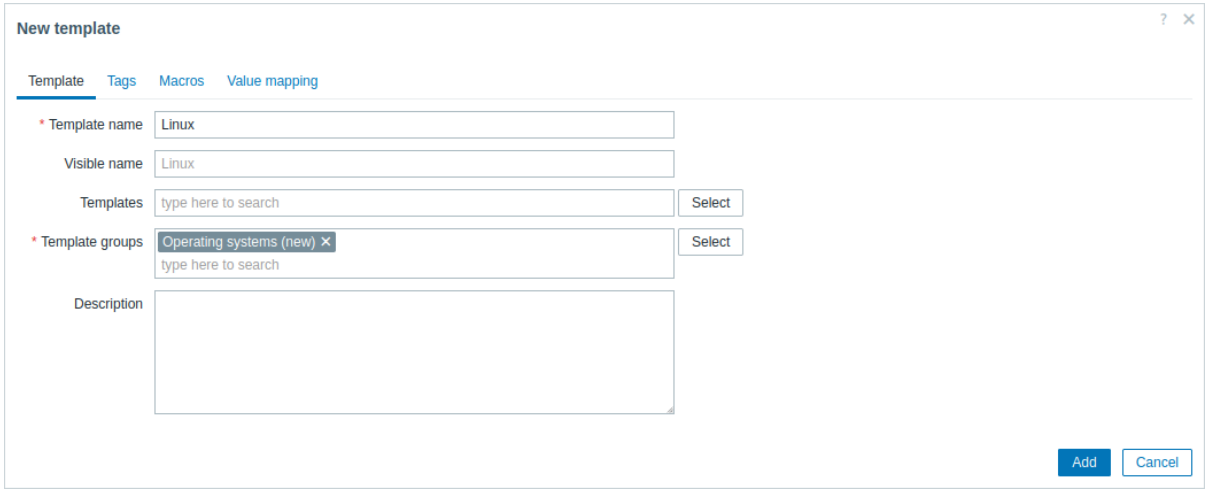
Veidnes konfigurēšanai vispirms ir jāizveido veidne, definējot tās vispārīgos parametrus, un pēc tam tai jāpievieno entītijas (vienumi, trigeri, grafiki u.c.).

Veidnes izveide

Lai izveidotu veidni, rīkojieties, kā norādīts tālāk:

1. Dodieties uz *Data collection* > *Templates*.
2. Noklikšķiniet uz *Create template*.
3. Rediģējiet veidnes atribūtus.

Cilnē **Template** ir ietverti vispārīgie veidnes atribūti.

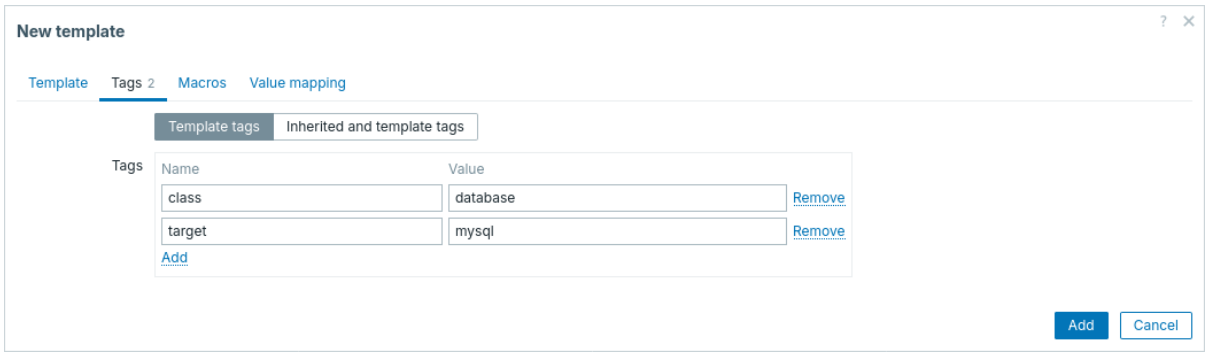


Visi obligātie ievades lauki ir atzīmēti ar sarkanu zvaigznīti.

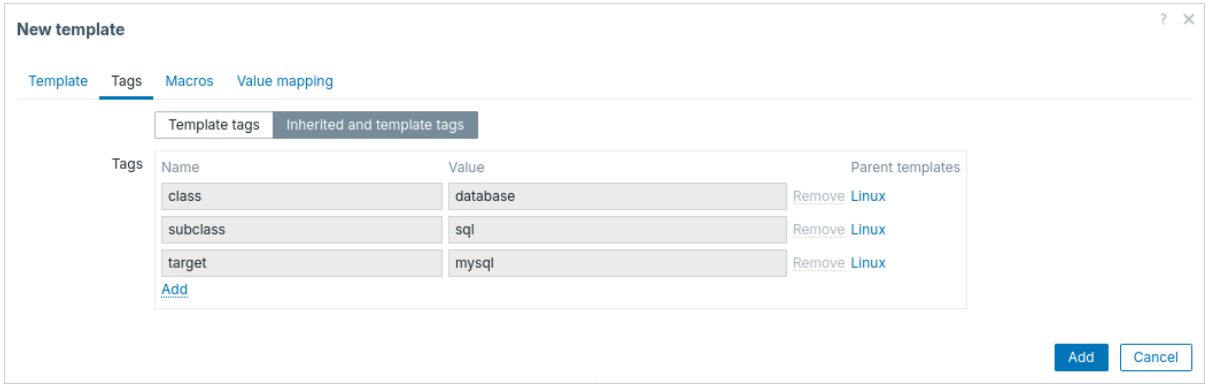
Veidnes atribūti:

Parameter	Description
Template name	Unikāls veidnes nosaukums. Ir atļauti burti un cipari, atstarpes, punkti, defises un pasvītrojuma zīmes. Atstarpes nosaukuma sākumā un beigās nav atļautas.
Visible name	Ja iestatāt šo nosaukumu, tas būs redzams sarakstos, kartēs u. c.
Templates	Saistiet ar šo veidni vienu vai vairākas veidnes. Visas entītijas (vienumi, trigeri u. c.) tiks mantotas no saistītajām veidnēm. Lai saistītu jaunu veidni, laukā *Templates* ievadiet tās nosaukumu — tiks parādīts atbilstošo veidņu nolaižamais saraksts. Varat arī noklikšķināt uz *Select*, lai atvērtu uznirstošo logu *Templates*, kurā sākotnēji nav nevienas veidnes. Lai aizpildītu sarakstu, laukā *Template group* ievadiet veidņu grupas nosaukumu (un atlasiet to ieteikumu sarakstā) vai noklikšķiniet uz pogas *Select* blakus laukam *Template group*, lai atvērtu uznirstošo logu *Template groups*. Kad esat izvēlējies veidņu grupu, uznirstošais logs *Templates* tiek atjaunināts un tajā tiek parādītas šai grupai piederošās veidnes. Atlasītās veidnes tiks saistītas, kad konfigurācija tiks saglabāta vai atjaunināta. Lai atsaistītu veidni, izmantojiet vienu no divām iespējām blokā *Templates*: *Unlink* — atsaistīt veidni, bet saglabāt tās entītijas (vienumus, trigerus u. c.); *Unlink and clear* — atsaistīt veidni un noņemt visas tās entītijas (vienumus, trigerus u. c.).
Template groups	Veidņu grupas, kurām pieder veidne.
Description	Veidnes apraksts.
Vendor and version	Veidnes piegādātājs un versija; tiek parādīti tikai tad, kad tiek atjauninātas esošās veidnes (iepriekš sagatavotās veidnes, ko nodrošina Zabbix, importētās veidnes vai veidnes, kas modificētas, izmantojot Template API), ja veidnes konfigurācijā ir ietverta šāda informācija. Zabbix lietotāja saskarnē tos nevar modificēt. Iepriekš sagatavotām veidnēm versija tiek parādīta šādi: Zabbix galvenā versija, atdalītājs (”-”), pārskatījuma numurs (tiek palielināts ar katru jaunu veidnes versiju un atiestatīts ar katru jaunu Zabbix galveno versiju). Piemēram, 7.0-0, 7.0-5, 8.0-0, 8.0-3.

Cilnē **Tags** varat definēt veidnes līmeņa tagus. Visas ar šo veidni saistīto hostu problēmas tiks marķētas ar šeit ievadītajām vērtībām.



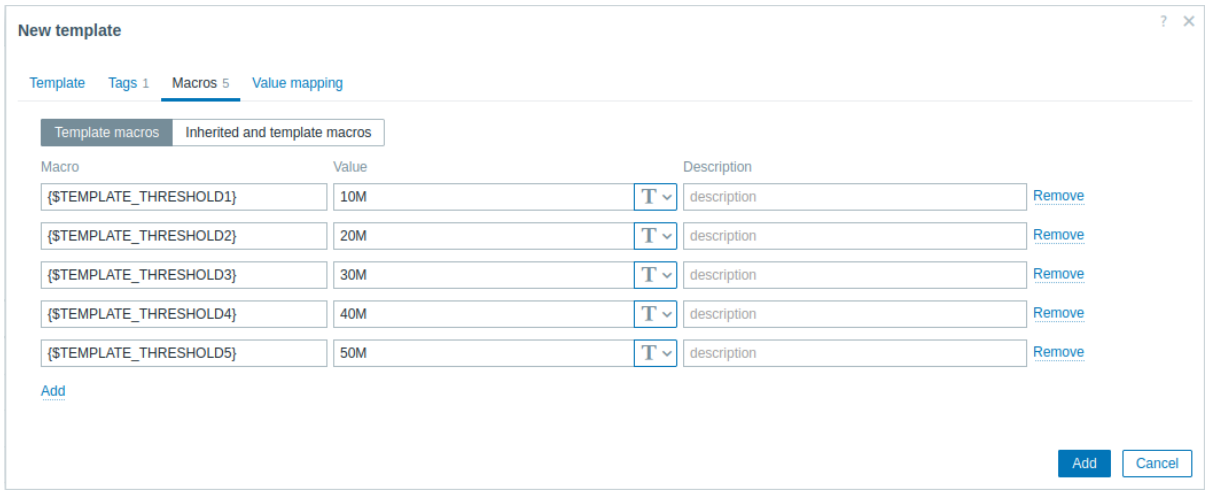
Ja atlasāt opciju *Inherited and template tags*, varat skatīt arī tagus no saistītajām veidnēm, kurus veidne mantos.



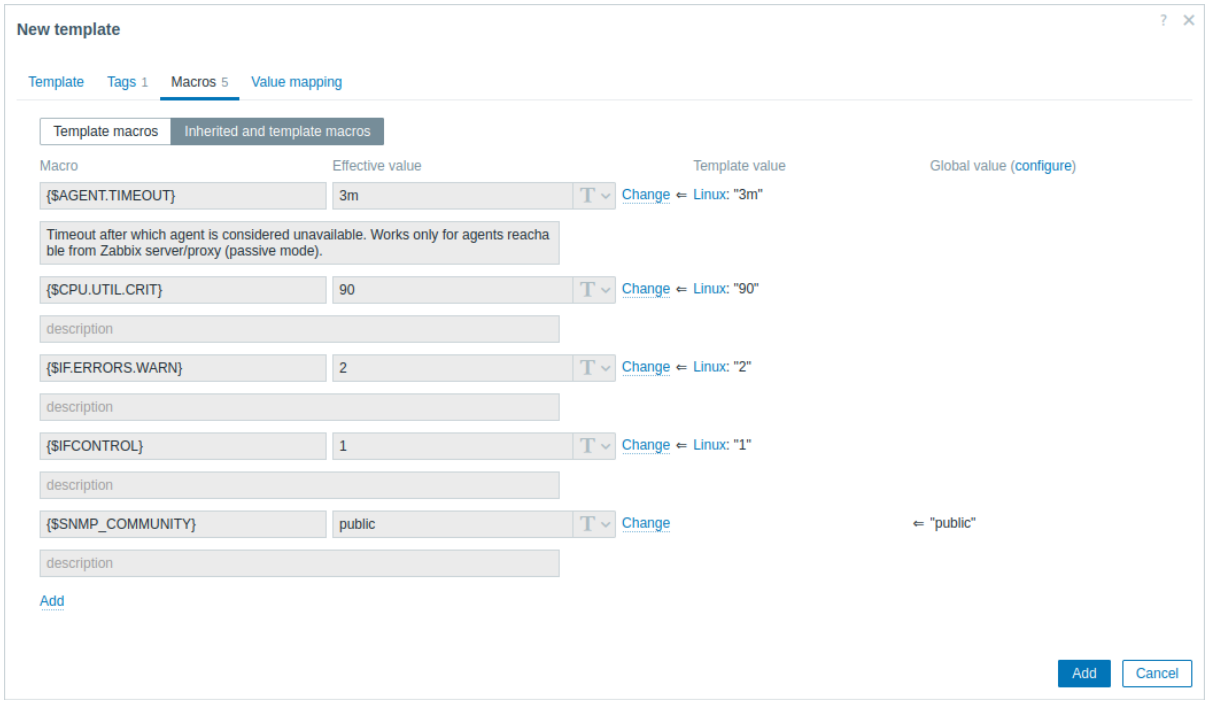
Ērtības labad ir nodrošinātas saites uz attiecīgajām veidnēm.

Tagos tiek atbalstīti lietotāja makro, {INVENTORY.*} makro, {HOST.HOST}, {HOST.NAME}, {HOST.CONN}, {HOST.DNS}, {HOST.IP}, {HOST.PORT} un {HOST.ID} makro.

Cilnē **Macros** varat definēt veidnes līmeņa lietotāja makro kā nosaukuma un vērtības pārus. Ņemiet vērā, ka makro vērtības var tikt saglabātas kā vienkāršs teksts, slepens teksts vai Vault noslēpums. Tiek atbalstīta arī apraksta pievienošana.



Ja atlasāt opciju *Inherited and template macros*, varat skatīt arī makro no saistītajām veidnēm un globālos makro, kurus veidne mantos, kā arī vērtības, uz kurām makro tiks izvērsti.



Ērtības labad ir nodrošinātas saites uz attiecīgajām veidnēm, kā arī saite uz globālo makro konfigurāciju. Saistītās veidnes makro vai globālo makro var arī rediģēt veidnes līmenī, tādējādi efektīvi izveidojot makro kopiju veidnē.

Cilnē **Value mapping** varat konfigurēt vienumu datu cilvēkiem saprotamu attēlojumu vērtību kartējumos.

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 u. c.), kas mantotas no saistītajām veidnēm un tieši pievienotas pašreizējai veidnei, bet **neietver** pašreizējās veidnes piegādātāju un versiju, lai klonēto veidni varētu atšķirt no oriģināla.
Delete	Dzēst veidni; veidnes entītijas (vienumi, trigeri u. c.) paliek saistītajos hostos.
Delete and clear	Dzēst veidni un visas tās entītijas no saistītajiem hostiem.
Cancel	Atcelt veidnes rekvizītu rediģēšanu.

Vienumu, trigeru, grafiku pievienošana

Attention:
Vienumi vispirms jāpievieno veidnei. Trigerus un grafikus nevar pievienot bez atbilstošā vienuma.

Jūs varat pievienot esošos vienumus veidnei vai izveidot jaunus vienumus.

Lai pievienotu vienumus veidnei:

1. Dodieties uz *Datu vākšana* > *Hosts* (vai *Veidnes*).
2. Rindā ar nepieciešamo hostu/veidni noklikšķiniet uz *Vienumi*.
3. Atzīmējiet to vienumu izvēles rūtiņas, kurus vēlaties pievienot veidnei.
4. Noklikšķiniet uz *Kopēt* zem vienumu saraksta.
5. Atlasiet veidni (vai veidņu grupu), uz kuru vienumi jākopē, un noklikšķiniet uz *Kopēt*.
 Visiem atlasītajiem vienumiem jābūt nokopētiem uz veidni.

Trigeru un grafiku pievienošana notiek līdzīgā veidā (attiecīgi no trigeru un grafiku saraksta), atkal ņemot vērā, ka tos var pievienot tikai tad, ja vispirms ir pievienoti nepieciešamie vienumi.

Paneļu pievienošana

Lai pievienotu paneļus veidnei sadaļā *Datu vākšana* > *Veidnes*, rīkojieties šādi:

1. Noklikšķiniet uz *Paneļi* veidnes rindā.
2. Konfigurējiet paneli, ievērojot paneļu konfigurēšanas norādījumus.

Attention:
Konfigurējot logrīkus veidnes panelī (nevis globālajā panelī), ar hostu saistītie parametri nav pieejami, un dažiem parametriem ir atšķirīgs nosaukums. Tas ir tāpēc, ka veidnes paneļi rāda datus tikai no hosta, ar kuru veidne ir saistīta. Piemēram, parametri *Host groups*, *Exclude host groups* un *Hosts Problems* logrīkā nav pieejami, parametrs *Host groups Host availability* logrīkā nav pieejams, un parametrs *Show hosts in maintenance* ir pārdēvēts par *Show data in maintenance* utt. Papildinformāciju par parametru pieejamību veidnes paneļa logrīkos skatiet konkrētajos katra paneļa logrīka parametros.

Note:
Sīkāku informāciju par piekļuvi hostu paneļiem, kas izveidoti no veidnes paneļiem, skatiet sadaļā hostu paneļi.

Zema līmeņa atklāšanas noteikumu konfigurēšana

Skatiet rokasgrāmatas sadaļu zema līmeņa atklāšana.

Web scenāriju pievienošana

Lai pievienotu web scenārijus veidnei sadaļā *Datu vākšana* > *Veidnes*, rīkojieties šādi:

1. Noklikšķiniet uz *Web* veidnes rindā.
2. Konfigurējiet web scenāriju, ievērojot parasto web scenāriju konfigurēšanas metodi.

Veidņu grupas konfigurēšana

Pārskats

Veidņu grupas tiek izmantotas veidņu loģiskai grupēšanai un lietotāju atļauju piešķiršanai tām.

Katrai veidnei ir jābūt piešķirtai vismaz vienai veidņu grupai. Veidne var piederēt vairākām veidņu grupām, un katra veidņu grupa var saturēt vairākas veidnes.

Ņemiet vērā, ka Zabbix visas atļaujas ir balstītas uz grupām: lietotāju grupām, host grupām un veidņu grupām. Tāpēc pat tad, ja vienam lietotājam ir nepieciešama piekļuve vienai veidnei, tā tiek piešķirta, pievienojot lietotāju lietotāju grupai un piešķirot šai lietotāju grupai atļaujas veidņu grupai, kurā atrodas attiecīgā veidne.

Konfigurācija

> **Attention:**
> Tikai Super admin lietotāji var izveidot veidņu grupas.

Zabbix lietotāja saskarnē ir divas veidņu grupas izveides iespējas.

Pirmais variants:

1. Dodieties uz *Data collection* > *Template groups*.
2. Ekrāna augšējā labajā stūrī noklikšķiniet uz *Create template group*.
3. Formā ievadiet grupas nosaukumu.



Otrais variants: konfigurējot veidni, laukā *Template groups* ievadiet neeksistējošas grupas nosaukumu.

Kad veidņu grupa ir izveidota, sarakstā zem *Data collection* > *Template groups* varat noklikšķināt uz veidnes nosaukuma, 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`.

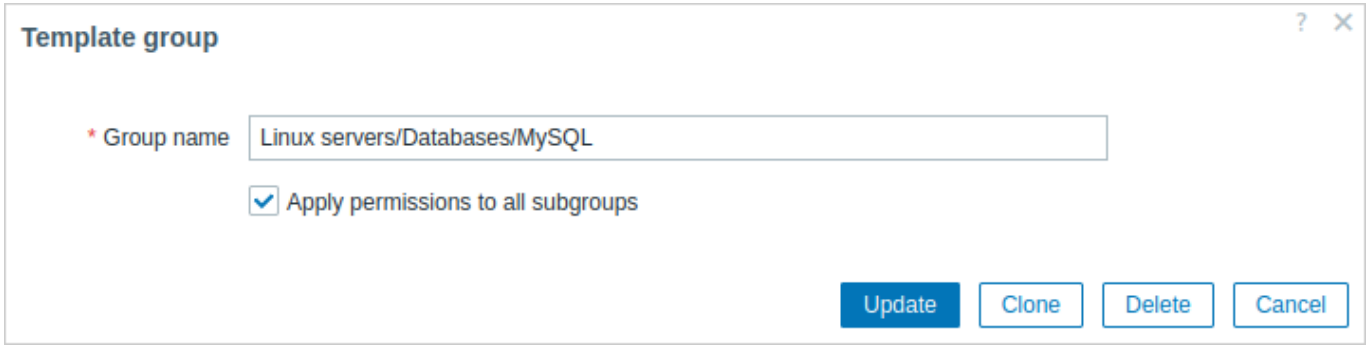
Veidojot apakšgrupu, nav atļauts izmantot slīpsvītras sākumā vai beigās, kā arī vairākas slīpsvītras pēc kārtas. '/' ekrānēšana netiek atbalstīta.

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`, ja `Linux servers` jau pastāv), lietotāju grupas atļaujas apakšgrupai tiek mantotas no vecākās grupas.
- Veidojot vecāko veidņu grupu esošai apakšgrupai (piemēram, izveidojot `Linux servers`, ja `Linux servers/Databases` jau pastāv), vecākajai grupai atļaujas netiek iestatītas.

Rediģējot jebkuru veidņu grupu, varat iestatīt arī papildu opciju *Piemērot atļaujas visām apakšgrupām*.



Atzīmējot šo izvēles rūtiņu un noklikšķinot *Atjaunināt*, tiks piemērots tāds pats atļauju līmenis visām pašreizējām un turpmākajām rediģējamās veidņu grupas apakšgrupām.

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. Jebkuras izmaiņas, kas veiktas, izmantojot šo opciju, var atcelt tikai manuāli.

Saistīšana/atsaistīšana

Pārskats

Piesaistīšana ir process, kurā veidnes tiek piemērotas hostiem, savukārt atsaistīšana noņem hosta saistību ar veidni.

Veidnes saistīšana

Lai saistītu veidni ar hostu, rīkojieties, kā norādīts tālāk:

1. Atveriet *Data collection* > *Hosts*.
2. Noklikšķiniet uz nepieciešamā hosta.
3. Sāciet rakstīt veidnes nosaukumu laukā *Templates*. Tiks parādīts atbilstošo veidņu saraksts; ritiniet uz leju, lai atlasītu veidni. Varat arī noklikšķināt uz *Select* blakus laukam un uznirstošajā logā atlasīt vienu vai vairākas veidnes no saraksta.
4. Hosta atribūtu veidlapā noklikšķiniet uz *Add/Update*.

Tagad hostam būs visas veidnes entītijas. Tas ietver vienumus, trigerus, grafikus, zema līmeņa atklāšanas noteikumus, tīmekļa scenārijus, kā arī informācijas paneļus.

Attention:
Vairāku veidņu saistīšana ar vienu un to pašu hostu neizdosies, ja šajās veidnēs ir vienumi ar vienādu vienuma atslēgu. Tā kā trigeri un grafiki izmanto vienumus, arī tos nevar saistīt ar vienu hostu no vairākām veidnēm, ja tiek izmantotas identiskas vienumu atslēgas.

Kad no veidnes tiek pievienotas entītijas (vienumi, trigeri u. c.):

- iepriekš hostā esošās identiskās entītijas tiek atjauninātas kā veidnes entītijas, un **visi esošie pielāgojumi, kas hosta līmenī veikti entītijai, tiek zaudēti**;
- entītijas no veidnes tiek pievienotas;
- visas tieši saistītās entītijas, kas pirms veidnes saistīšanas pastāvēja tikai hostā, paliek nemainītas.

Sarakstos visām veidnes entītijām tagad pirms nosaukuma ir veidnes nosaukums, kas norāda, ka tās pieder konkrētajai veidnei. Pats veidnes nosaukums (attēlots pelēkā krāsā) 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 saistīšanas laikā hostam nav definēta saskarne, šie vienumi tiks pievienoti bez saskarnes. Ja hosta saskarni pievienosiet vēlāk, tā netiks automātiski piešķirta jau esošajiem vienumiem. Lai jaunpievienoto hosta saskarni vienlaikus piešķirtu visiem veidnes vienumiem, atceliet veidnes saistīšanu ar hostu un pēc tam saistiet to vēlreiz. Lai saglabātu vienuma vēsturi, izmantojiet opciju *Unlink*; neizmantojiet *Unlink and clear*.

Ja kādai entītijai pirms nosaukuma nav veidnes nosaukuma, tas nozīmē, ka tā hostā pastāvēja jau iepriekš un veidne to nepievienoja.

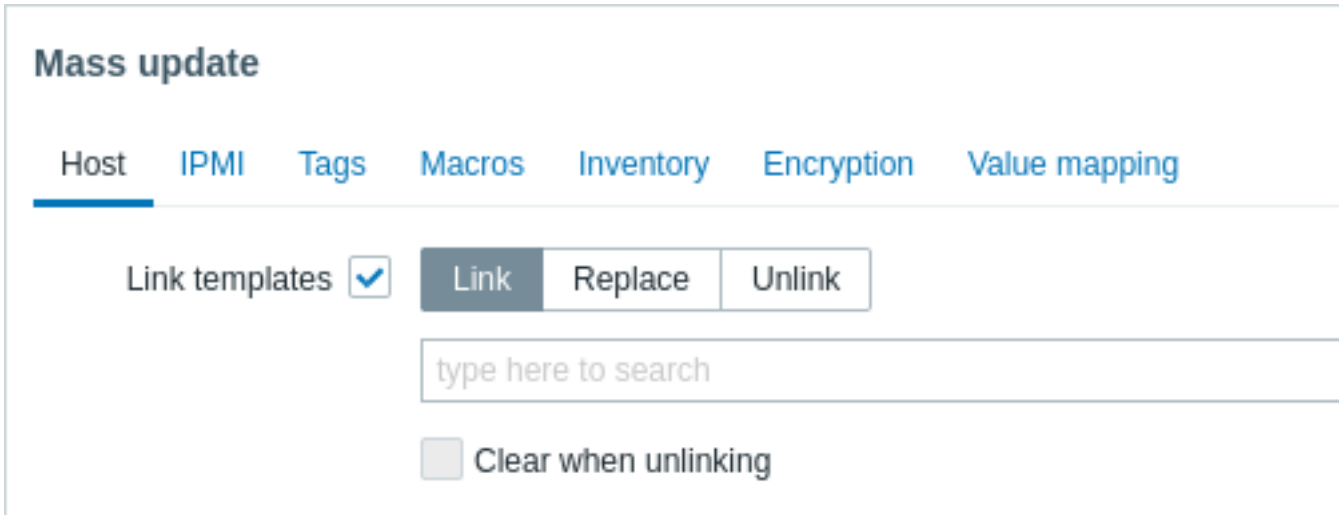
Entītiju unikalitātes kritēriji

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 piesaistīšana vairākiem hostiem

Lai atjauninātu daudzu hostu veidņu piesaisti, sadaļā *Datu apkopošana* > *Hosti* atzīmējiet vajadzīgo hostu izvēles rūtiņas, pēc tam zem saraksta noklikšķiniet uz **Masveida atjaunināšana** un atlasiet *Piesaistīt veidnes*:



Lai piesaistītu papildu veidnes, automātiskās pabeigšanas laukā sāciet rakstīt veidnes nosaukumu, līdz tiek parādīts nolaižamais saraksts ar atbilstošajām veidnēm. Ritiniet sarakstu uz leju, lai atlasītu piesaistāmo veidni.

Opcija *Aizstāt* ļauj piesaistīt jaunu veidni, vienlaikus atsaistot visas veidnes, kas iepriekš bija piesaistītas hostiem. Opcija *Atsaistīt* ļauj norādīt, kuras veidnes atsaistīt. Opcija *Notīrīt atsaistot* ļauj ne tikai atsaistīt iepriekš piesaistītās veidnes, bet arī noņemt visas no tām mantotās entītijas (vienumus, trigerus u. c.).

Note:
Zabbix piedāvā plašu iepriekš definētu veidņu kopu. Varat tās izmantot kā atsauci, taču neizmantojiet tās nemainītā veidā produkcijas vidē, jo tajās var būt pārāk daudz vienumu un dati var tikt iegūti pārāk bieži. Ja vēlaties tās izmantot, pielāgojiet tās savām faktiskajām 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 *Data collection* > *Hosts*.
2. Noklikšķiniet uz vajadzīgā hosta un atrodiet lauku *Templates*.
3. Noklikšķiniet uz *Unlink* vai *Unlink and clear* blakus veidnei, lai atsaistītu veidni.
4. Hosta atribūtu formā noklikšķiniet uz *Update*.

Izvēloties opciju *Unlink*, tiks vienkārši noņemta saistība ar veidni, bet visas tās entītijas paliks hostam. Tas ietver vienumus, trigerus, grafikus, zema līmeņa atklāšanas noteikumus un tīmekļa scenārijus, bet neietver paneļus. Ņemiet vērā, ka tiks noņemti arī vērtību kartējumi un tagi, kas mantoti no saistītajām veidnēm.

Izvēloties opciju *Unlink and clear*, tiks noņemta gan saistība ar veidni, gan visas tās entītijas (vienumi, trigeri utt.).

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 saistītu veidnes, jums ir jāņem 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 tagad veidnes konfigurācijā būs redzamas visas konfigurētās veidnes entītijas, kā arī visu saistīto veidņu entītijas. Tas ietver vienumus, trigerus, grafikus, zema līmeņa atklāšanas noteikumus un tīmekļa scenārijus, bet neietver informācijas paneļus. Tomēr saistīto veidņu informācijas paneļi, neskatoties uz to, tiks mantoti hosts.

Lai atsaistītu jebkuru no saistītajām veidnēm, veidnes konfigurācijas formā noklikšķiniet uz *Unlink* vai *Unlink and clear* un pēc tam noklikšķiniet uz *Update*.

Opcija *Unlink* vienkārši noņems saistību ar saistīto veidni, bet nenoņems nevienu no tās entītijām (vienumus, trigerus utt.).

Opcija *Unlink and clear* noņems gan saistību ar saistīto veidni, gan visas tās entītijas (vienumus, trigerus utt.).

Masveida atjaunināšana

Pārskats

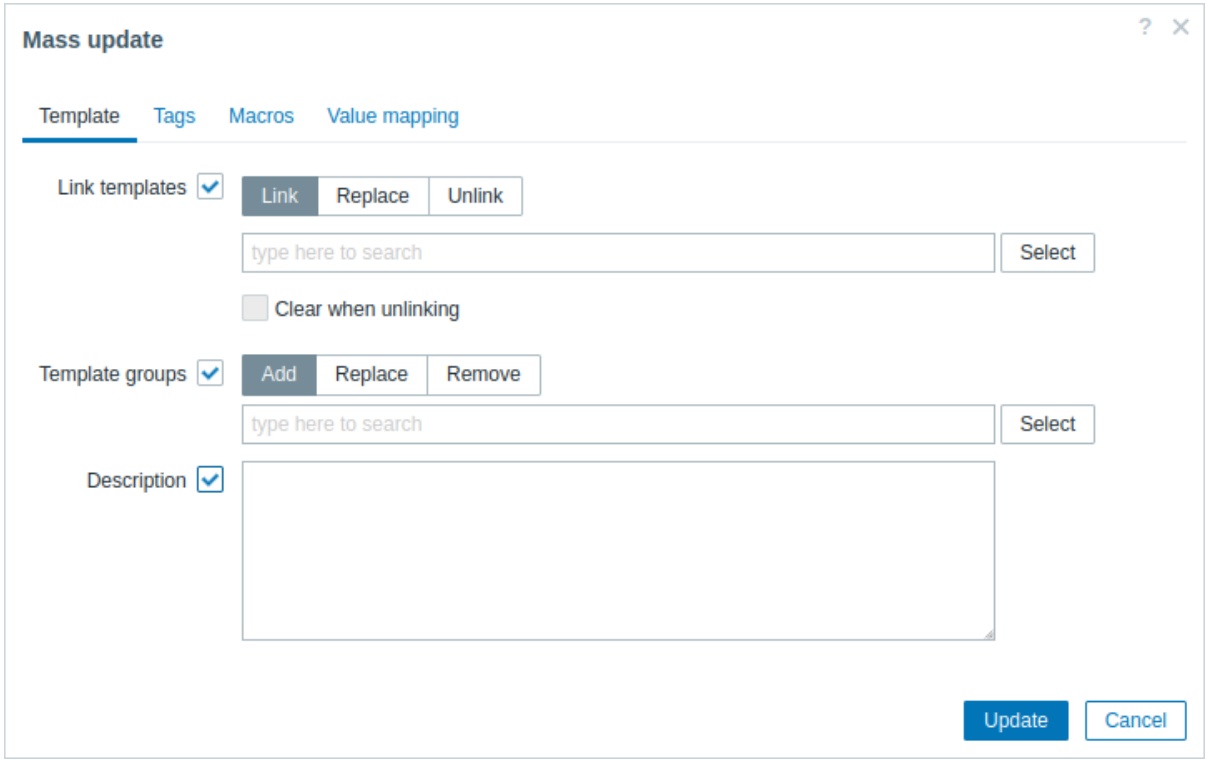
Dažreiz var būt nepieciešams vienlaikus mainīt kādu atribūtu vairākām veidnēm. Tā vietā, lai atvērtu katru atsevišķo veidni rediģēšanai, šim nolūkam varat izmantot masveida atjaunināšanas funkciju.

Masveida atjaunināšanas izmantošana

Lai masveidā atjauninātu dažas veidnes, rīkojieties šādi:

1. Atzīmējiet izvēles rūtiņas pie veidnēm, kuras vēlaties atjaunināt veidņu sarakstā.
2. Noklikšķiniet uz *Masveida atjaunināšana* zem saraksta.
3. Pārejiet uz cilni ar nepieciešamajiem atribūtiem (*Veidne*, *Tagi*, *Makro* vai *Vērtību kartēšana*).
4. Atzīmējiet to atribūtu izvēles rūtiņas, kurus vēlaties atjaunināt, un ievadiet tiem jaunu vērtību.

Cilnē **Veidne** ir pieejamas vispārīgas veidņu masveida atjaunināšanas opcijas.



Atlasot attiecīgo pogu atjauninājumam *Saistīt veidnes*, ir pieejamas šādas opcijas:

- *Saistīt* - norādīt, kuras papildu veidnes saistīt;
- *Aizstāt* - norādīt, kuras veidnes saistīt, vienlaikus atsaistot visas iepriekš saistītās veidnes;
- *Atsaistīt* - norādīt, kuras veidnes atsaistīt.

Lai norādītu saistāmās vai atsaistāmās veidnes, automātiskās pabeigšanas laukā sāciet rakstīt veidnes nosaukumu, līdz tiek parādīts nolaižamais saraksts ar atbilstošajām veidnēm. Ritiniet sarakstu, lai atlasītu nepieciešamās veidnes.

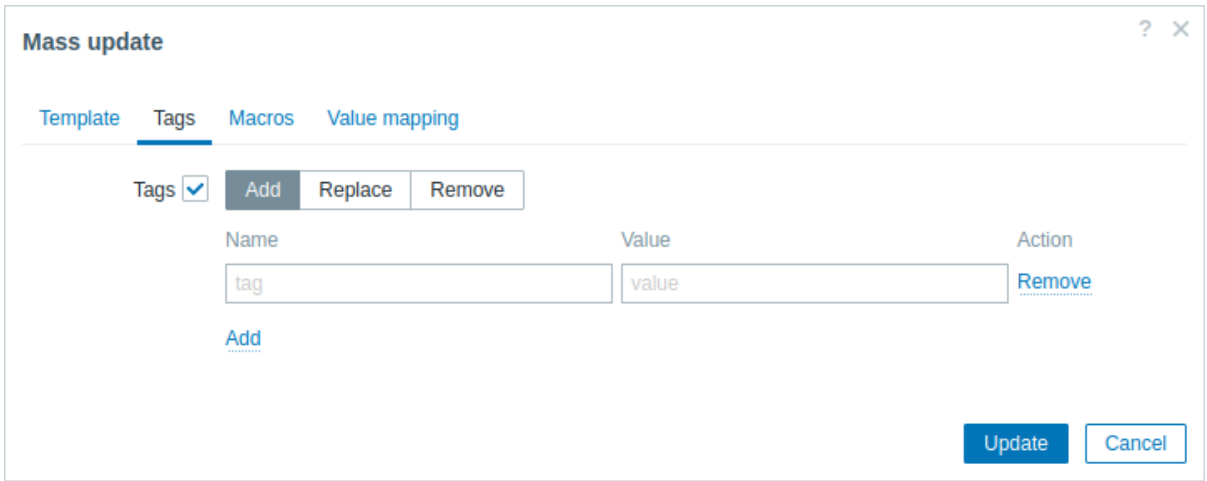
Opcija *Notīrīt, atsaistot* ļauj ne tikai atsaistīt iepriekš saistītās veidnes, bet arī noņemt visas no tām mantotās entītijas (vienumus, trigerus, grafikus u. c.).

Atlasot attiecīgo pogu atjauninājumam *Veidņu grupas*, ir pieejamas šādas opcijas:

- *Pievienot* - norādīt papildu veidņu grupas no esošajām grupām vai ievadīt veidnēm pilnīgi jaunas veidņu grupas;
- *Aizstāt* - noņemt veidni no visām esošajām veidņu grupām un aizstāt tās ar šajā laukā norādīto grupu vai grupām (esošām vai jaunām veidņu grupām);
- *Noņemt* - noņemt konkrētas veidņu grupas no veidnēm.

Šie ir automātiskās pabeigšanas lauki - sākot tajos rakstīt, tiek parādīts nolaižamais saraksts ar atbilstošajām veidņu grupām. Ja veidņu grupa ir jauna, tā arī tiek parādīta nolaižamajā sarakstā, un aiz virknes tiek norādīts *(jauna)*. Ritiniet sarakstu, lai atlasītu nepieciešamo grupu.

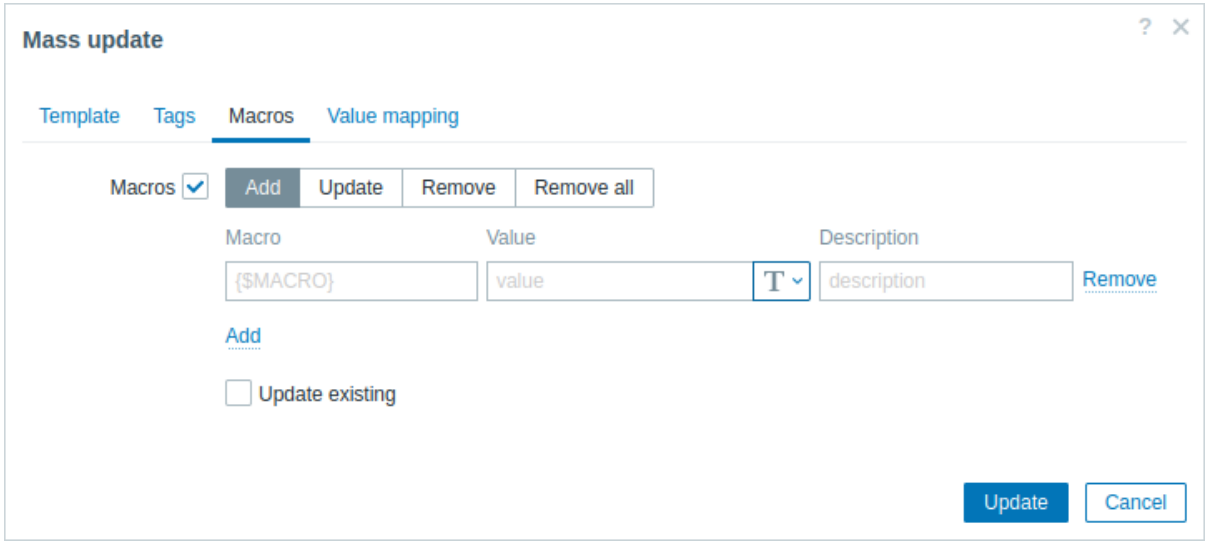
Cilne **Birkas** ļauj masveidā atjaunināt veidnes līmeņa birkas.



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.

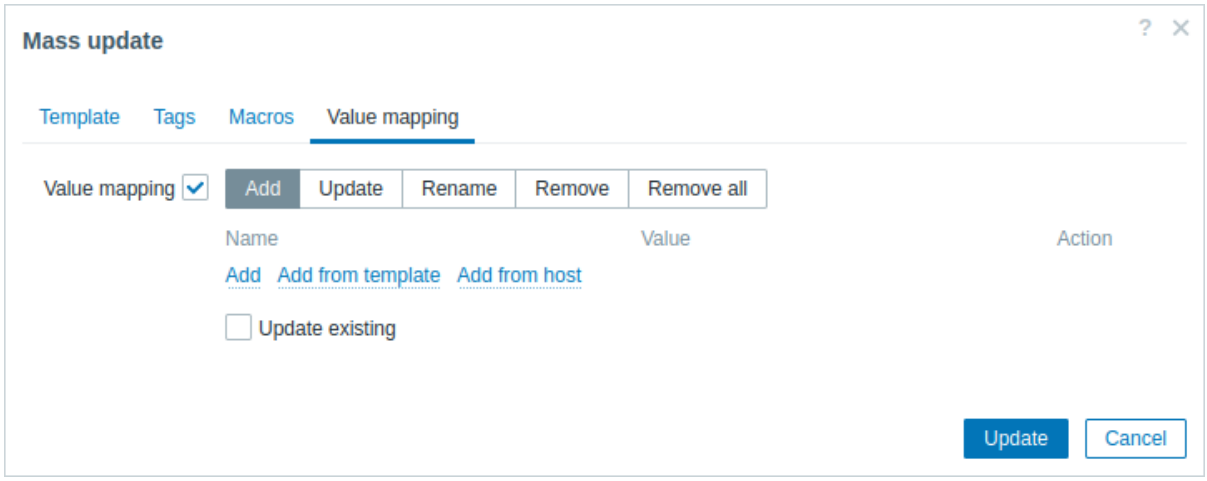
Cilne **Makro** ļauj masveidā atjaunināt veidņu līmeņa makro.



Atlasot attiecīgo pogu makro atjaunināšanai, ir pieejamas šādas opcijas:

- *Pievienot* - norādīt veidnēm papildu lietotāja makro. Ja ir atzīmēta izvēles rūtiņa *Atjaunināt esošos*, norādītā makro nosaukuma vērtība, tips un apraksts tiks atjaunināti. Ja tā nav atzīmēta un veidnē(-ēs) jau pastāv makro ar šādu nosaukumu, tas netiks atjaunināts.
- *Atjaunināt* - aizstāt šajā sarakstā norādīto makro vērtības, tipus un aprakstus. Ja ir atzīmēta izvēles rūtiņa *Pievienot trūkstošos*, makro, kas iepriekš veidnē nepastāvēja, tiks pievienots kā jauns makro. Ja tā nav atzīmēta, tiks atjaunināti tikai makro, kas veidnē jau pastāv.
- *Noņemt* - noņemt no veidnēm norādītos makro. Ja ir atzīmēta izvēles rūtiņa *Izņemot atlasītos*, tiks noņemti visi makro, izņemot sarakstā norādītos. Ja tā nav atzīmēta, tiks noņemti tikai sarakstā norādītie makro.
- *Noņemt visus* - noņemt no veidnēm visus lietotāja makro. Ja izvēles rūtiņa *Es apstiprinu visu makro noņemšanu* nav atzīmēta, tiks atvērts jauns uznirstošais logs ar lūgumu apstiprināt visu makro noņemšanu.

Cilne **Vērtību kartēšana** ļauj masveidā atjaunināt vērtību kartējumus.



Vērtību karšu atjaunināšanai ir pieejamas pogas ar tālāk norādītajām opcijām:

- *Pievienot* - pievienot vērtību kartes veidnēm. Ja atzīmēsiet *Atjaunināt esošās*, tiks atjaunināti visi vērtību kartes ar šo nosaukumu rekvizīti. Pretējā gadījumā, ja vērtību karte ar šādu nosaukumu jau pastāv, tā netiks atjaunināta.
- *Atjaunināt* - atjaunināt esošās vērtību kartes. Ja atzīmēsiet *Pievienot trūkstošās*, vērtību karte, kas iepriekš veidnē nepastāvēja, tiks pievienota kā jauna vērtību karte. Pretējā gadījumā tiks atjauninātas tikai tās vērtību kartes, kas veidnē jau pastāv.
- *Pārdēvēt* - piešķirt jaunu nosaukumu esošai vērtību kartei.
- *Noņemt* - noņemt norādītās vērtību kartes no veidnēm. Ja atzīmēsiet *Izņemot atlasītās*, tiks noņemtas visas vērtību kartes, **izņemot** norādītās.
- *Noņemt visas* - noņemt visas vērtību kartes no veidnēm. Ja izvēles rūtiņa *Es apstiprinu visu vērtību karšu noņemšanu* nav atzīmēta, tiks atvērts jauns uznirstošais logs ar lūgumu apstiprināt noņemšanu.

Opcijas *Pievienot no veidnes* un *Pievienot no hosts* ir pieejamas vērtību kartēšanas pievienošanas/atjaunināšanas darbībām. Tās ļauj atlasīt vērtību kartējumus attiecīgi no veidnes vai hosts.

Kad visas nepieciešamās izmaiņas ir veiktas, noklikšķiniet uz *Update*. Atribūti tiks attiecīgi atjaunināti visām atlasītajām veidnēm.

Hosts 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.

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 jūs izved cauri šādiem soļiem:

- Veidnes izvēle
- Hosts izveide vai izvēle
- Zabbix aģenta vai agent 2 instalēšana
- Host interfeisa pievienošana
- Papildu konfigurācijas piemērošana jūsu uzraudzības mērķim vai Zabbix hostam (ja to pieprasa veidne)

Lai piekļūtu Host Wizard Zabbix lietotāja saskarnē, rīkojieties šādi:

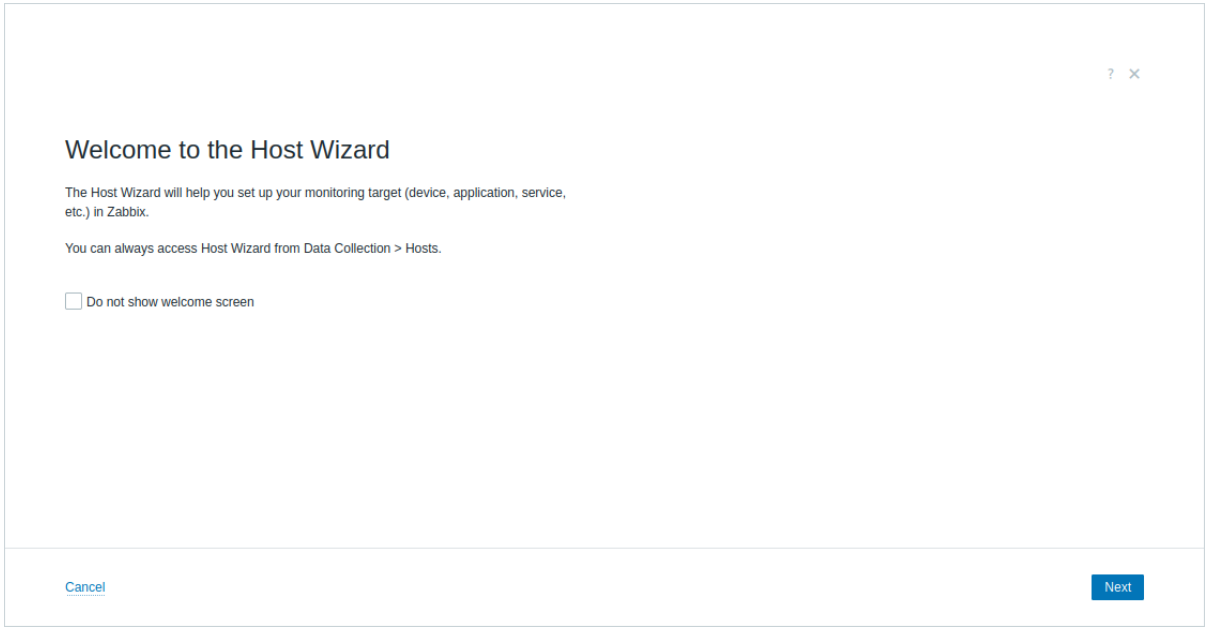
1. Dodieties uz: *Data collection* > *Hosts*.
2. Noklikšķiniet uz *Host Wizard* ekrāna augšējā labajā stūrī.
3. Izpildiet vedņa norādījumus.

Varat arī rediģēt esošu hostu, noklikšķinot uz tā nosaukuma (vai trīs punktu ikonas blakus tam) dažādās lietotāja saskarnes sadaļās, lai piekļūtu Host Wizard no host menu. Ņemiet vērā, ka, rediģējot esošu hostu, daži soļi tiek izlaisti, ja tajā jau ir iekļauta atlasītajai veidnei nepieciešamā konfigurācija.

Alternatīvi varat izmantot klasisko pieeju hosta izveidei un konfigurēšanai.

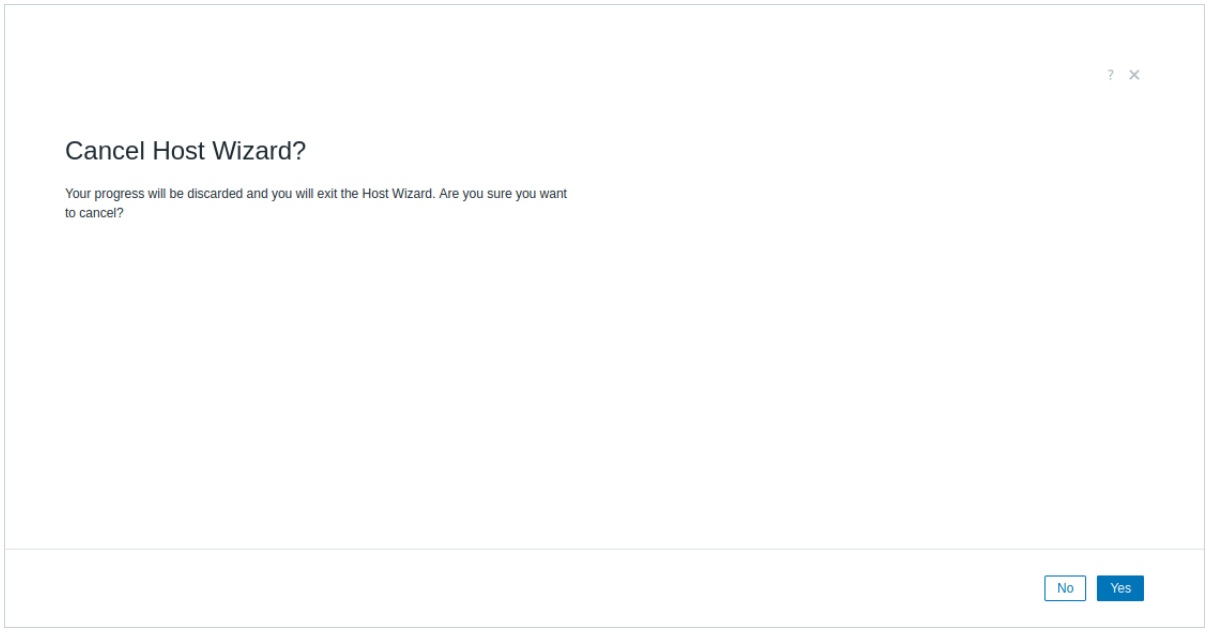
Laipni lūdzam Host Wizard

Kad tiek palaists Host Wizard, tiek parādīts sveiciena ekrāns:



Lai turpmākajās sesijās šo ekrānu izlaistu, varat atzīmēt izvēles rūtiņu *Do not show welcome screen* un noklikšķināt *Next*.

Ja nākamajās Host Wizard darbībās sākat konfigurēt iestatījumus un mēģināt aizvērt vedni, noklikšķinot uz aizvēršanas ikonas vai nospiežot Esc, tiks parādīts apstiprinājuma ekrāns:



Noklikšķinot *Yes*, jūs iziesiet no Host Wizard, nesaglabājot paveikto. Noklikšķinot *No*, Esc vai aizvēršanas ikonas, tiks atgriezts pēdējais solis.

Atlasiet veidni

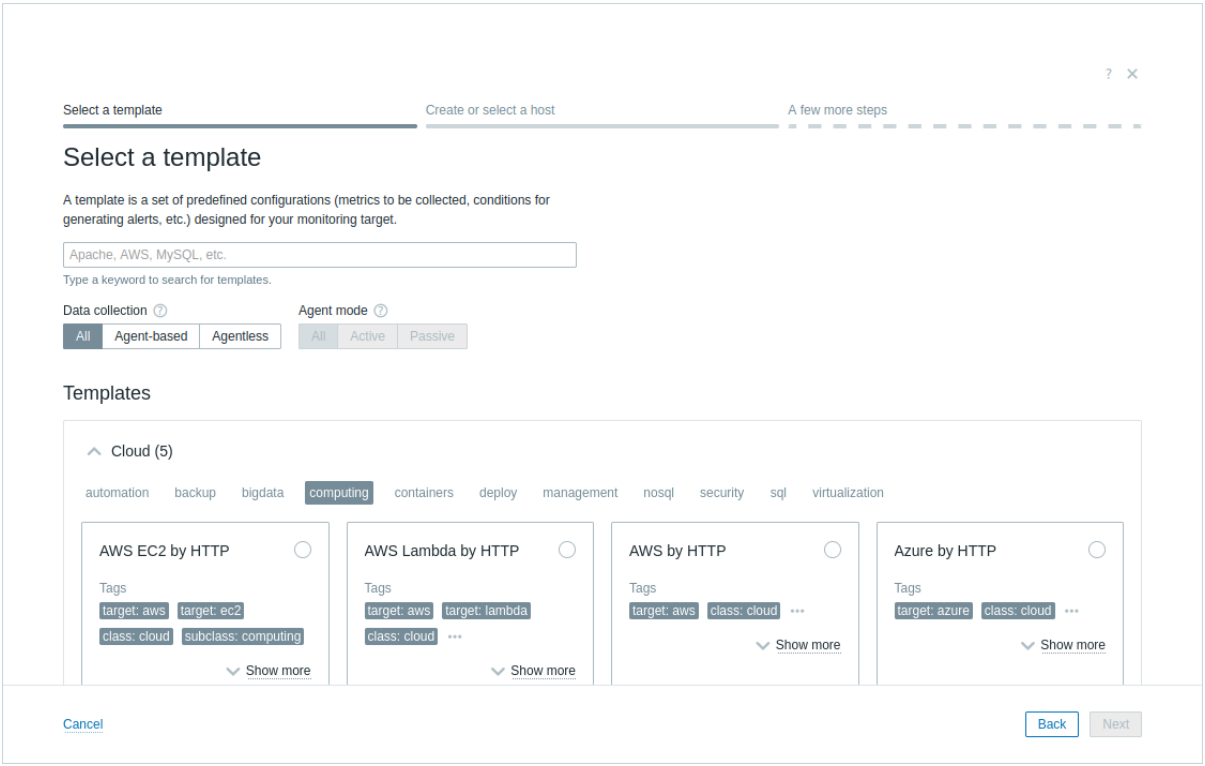
Pirmais jūsu uzraudzības mērķa iestatīšanas solis ir veidnes atlasīšana. Tā ir iepriekš definētu konfigurāciju kopa (apkopojamie rādītāji, brīdinājumu ģenerēšanas nosacījumi u. c.), kas paredzēta jūsu uzraudzības mērķim.

Zabbix gatavās veidnes piedāvā dažādas iepriekš definētas uzraudzības konfigurācijas operētājsistēmām, lietojumprogrammām, datubāzēm, tīkla ierīcēm, pakalpojumiem un citām sistēmām.

Šajā darbībā varat:

- Pārlūkot veidnes pēc klases (mākonis, datubāze, tīkls u. c.) un apakšklases (piemēram, automatizācija, dublēšana), kas balstītas uz veidnes tagiem.
- Meklēt veidnes pēc atslēgvārdiem veidnes nosaukumā vai veidnes tagu nosaukumos un vērtībās.
- Filtrēt veidnes pēc datu apkopošanas metodes (izmantojot aģentu vai bez aģenta; veidnēs, kurās tiek izmantots aģents, ir vismaz viens Zabbix aģenta vienums).
- Filtrēt veidnes pēc aģenta režīma (aktīvs vai pasīvs; aktīvās veidnes ietver vismaz vienu Zabbix aģenta (aktīvo) vienumu, savukārt pasīvās veidnes ietver vismaz vienu pasīvo vienumu; plašāku informāciju skatiet sadaļā Pasīvās un aktīvās aģenta pārbaudes).

- Izmantojot resursdatora vedni esoša resursdatora konfigurēšanai, filtrēt arī veidnes, kas jau ir piesaistītas resursdatoram.



Attention:
Atlasot veidni, var tikt parādīts šāds ziņojums: *Dažas veidnes (n) nav saderīgas ar resursdatora vedni. Uzziniet, kā tās atjaunināt. Pielāgotas veidnes netiek atbalstītas.* Tas nozīmē, ka dažas gatavās veidnes vēl nav saderīgas ar resursdatora vedni un tās ir jāatjaunina; skatiet veidnes jaunināšanas norādījumus.

Atlasītā veidne noteiks nākamās darbības resursdatora vednī. Piemēram, ja atlasāt veidni *MySQL by Zabbix agent*, kurā datu apkopošanai tiek izmantots Zabbix aģents, vednis palīdzēs jums veikt aģenta instalēšanas procesu.

Izveidojiet vai atlasiet hostu

Veidne ir jāsaista ar hostu, kas ir Zabbix entītija, kura attēlo jūsu pārraudzības mērķi. Pēc saistīšanas hosts saņem visas veidnes entītijas, piemēram, vienumus (apkopojamos rādītājus) un trigerus (nosacījumus brīdinājumu ģenerēšanai), kas nepieciešami pārraudzībai.

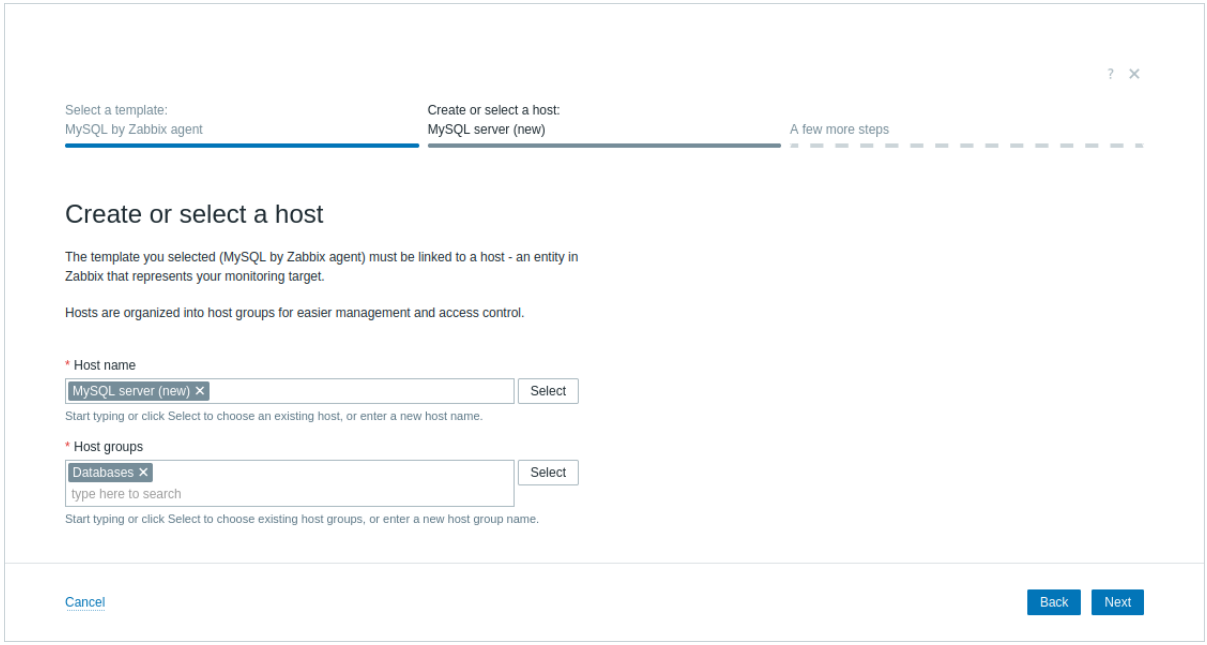
Katram hostam ir jāpieder vismaz vienai hostu grupai, ko izmanto hostu organizēšanai un lietotāju atļauju piešķiršanai tiem.

Šajā darbībā varat:

- Izveidot jaunu hostu un hostu grupu, ievadot abu nosaukumus.
- 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 pašreizējām grupām.

Note:
Esoša hosta atlasīšana nenoņem un nepārraksta tā pašreizējo konfigurāciju, ja vien tas nav skaidri norādīts turpmākajās darbībās. Atklātos hostus nevar atlasīt.

Piemēram, varat izveidot jaunu hostu ar nosaukumu *MySQL server*, lai attēlotu lokāli instalēto MySQL serveri, kuru vēlaties pārraudzīt. Iepriekš atlasītā veidne *MySQL by Zabbix agent* tiks saistīta ar šo hostu. Turklāt varat atlasīt esošo hostu grupu *Databases* (vai izveidot jaunu, piemēram, *MySQL servers*), lai organizētu šo 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 izvietots jūsu uzraudzības mērķī, lai aktīvi uzraudzītu lokālos resursus un lietojumprogrammas (ja to pieprasa veidne).

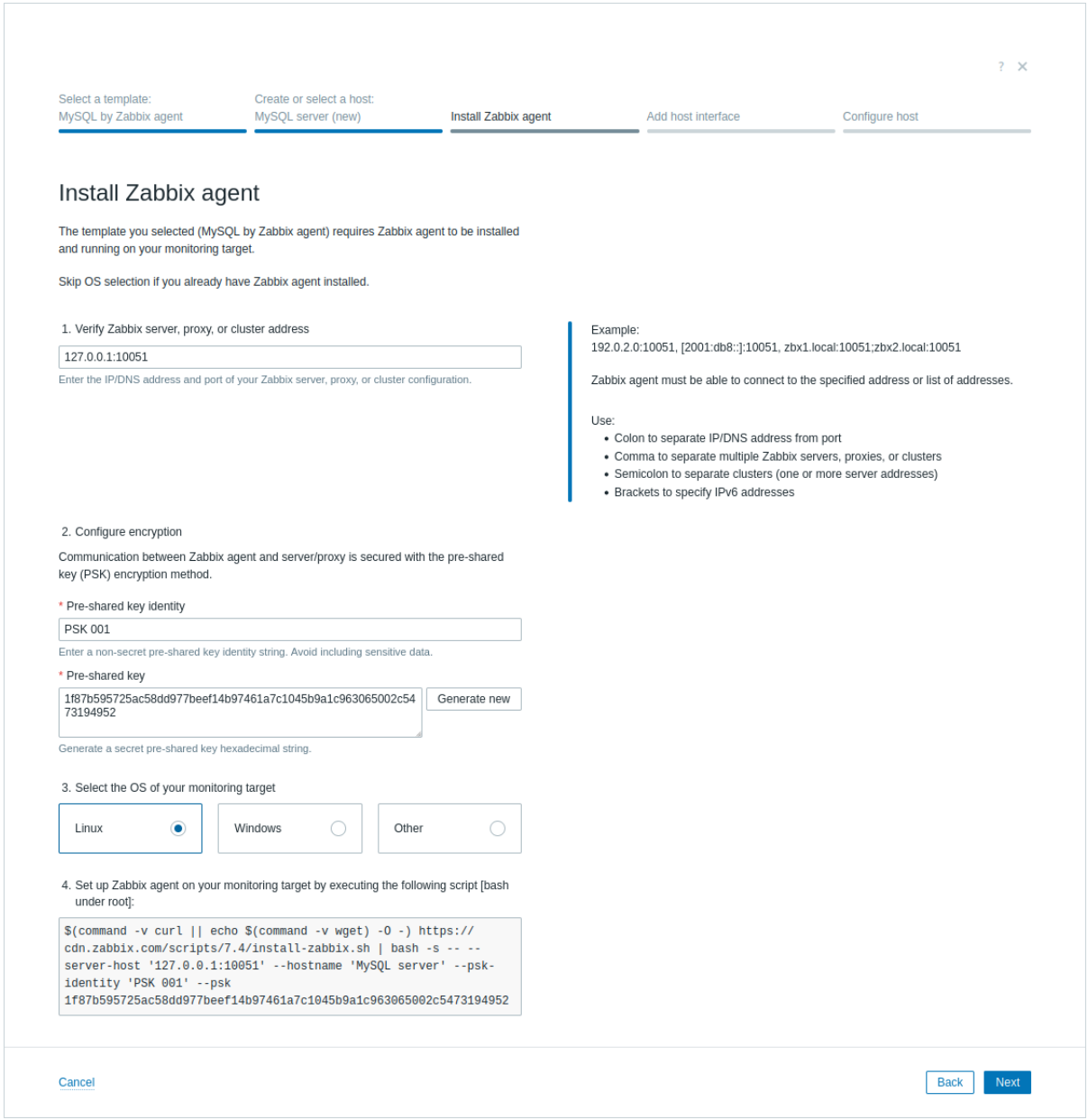
Šajā darbībā:

- Pārbaudiet Zabbix servera, starpniekservera vai klastera adresi (piemēram, 192.0.2.1:10051). Šī vērtība tiks izmantota, lai konfigurētu Server un ServerActive parametrus Zabbix aģentam.
- Ievadiet jaunu, unikālu un neizpaužamu 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 serverim*) un Zabbix aģentam.

Attention:
Rediģējot esošu hostu, PSK konfigurācija pārrakstīs visus esošos šifrēšanas iestatījumus hostā.

- Atlasiet tās iekārtas operētājsistēmu, kurā atrodas jūsu uzraudzības mērķis.
- Instalējiet Zabbix aģentu šajā sistēmā, tajā palaižot nodrošināto skriptu vai izpildot instalēšanas norādījumus.

Piemēram, varat iestatīt *Pre-shared key identity* uz *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 pie Host Wizard.

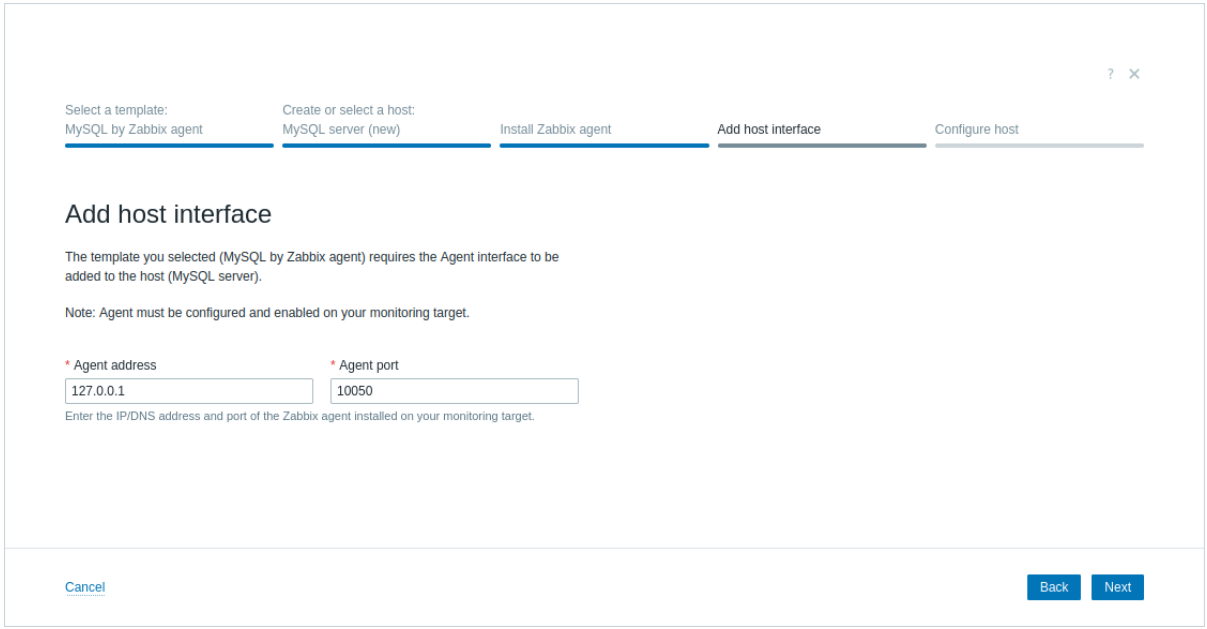


Pievienot hosta saskarni

Hosta saskarne nosaka, kā Zabbix serveris savienojas ar jūsu uzraudzības mērķi tīklā. Tā nosaka savienojuma parametrus, piemēram, IP adresi, DNS nosaukumu, portu un saskarnes tipu (aģents, SNMP, JMX vai IPMI), atkarībā no datu vākšanas metodes, ko pieprasa atlasītā veidne.

Šajā solī ievadiet atlasītajai veidnei nepieciešamo hosta saskarnes informāciju.

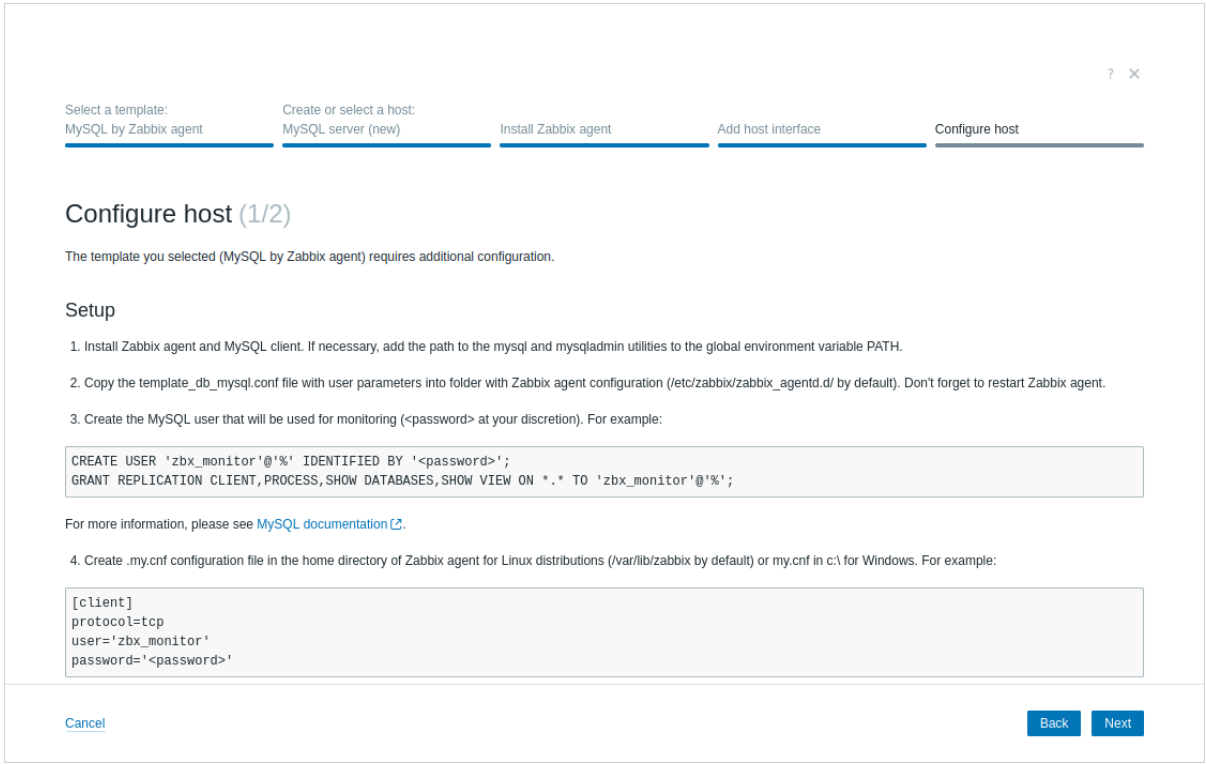
Piemēram, varat izmantot noklusējuma aģenta adresi (127.0.0.1) un portu (10050), ja Zabbix serveris, Zabbix aģents un MySQL serveris visi darbojas lokāli vienā un tajā pašā datorā.



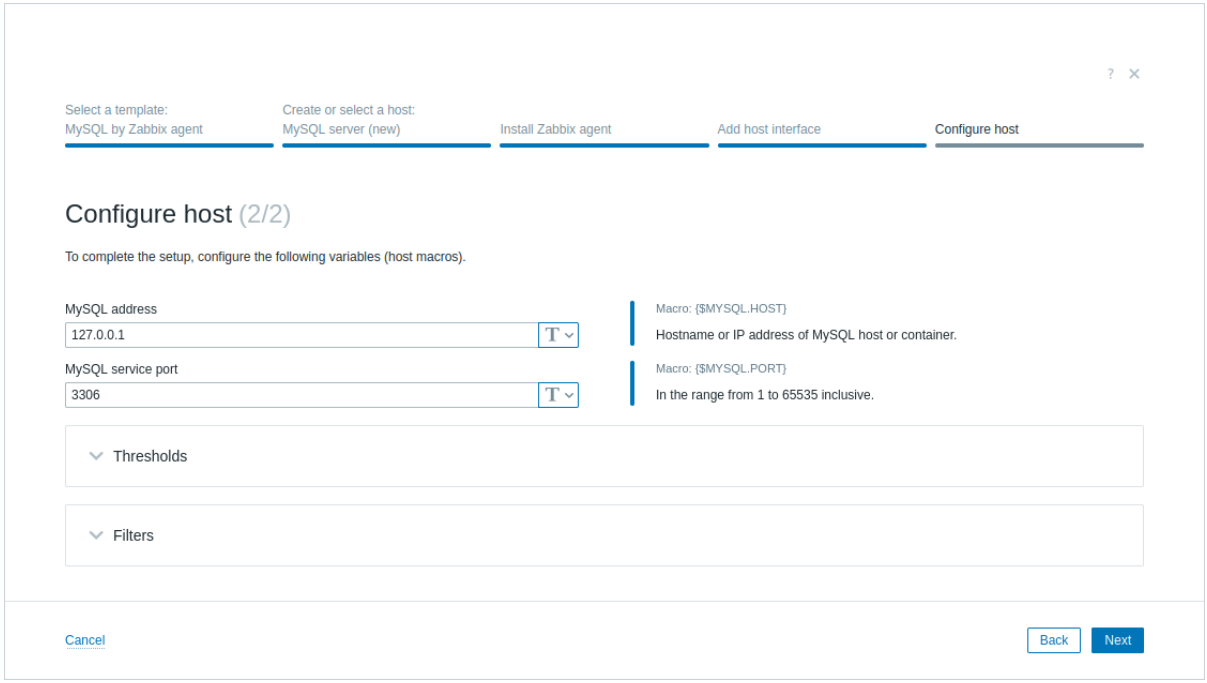
Hosts konfigurēšana

Dažām veidnēm pirms hosta izveides ir nepieciešama papildu konfigurēšana. Tā var ietvert:

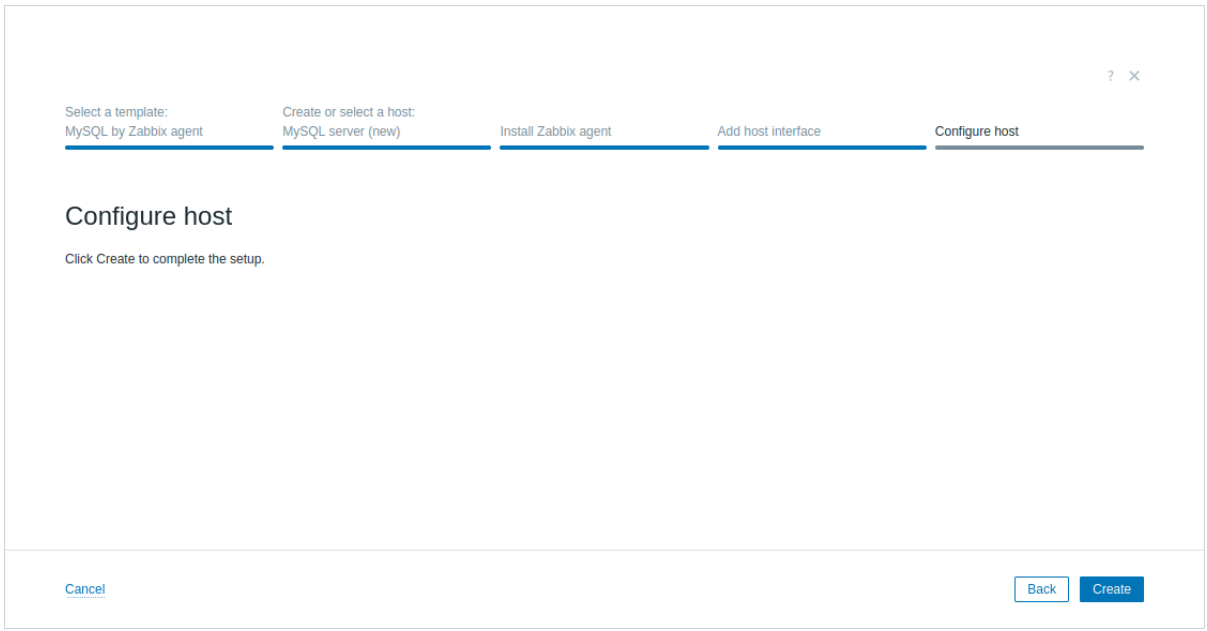
- Pārraudzības mērķa manuālu konfigurēšanu (piemēram, noteiktu pakalpojumu iespējošanu, pakalpojumu lietotāju izveidi vai atļauju piešķiršanu):



- Hosta makros definēšanu — mainīgie, kas kontrolē vienumu darbību, savienojuma iestatījumus, autentifikācijas akreditācijas datus un citus parametrus:



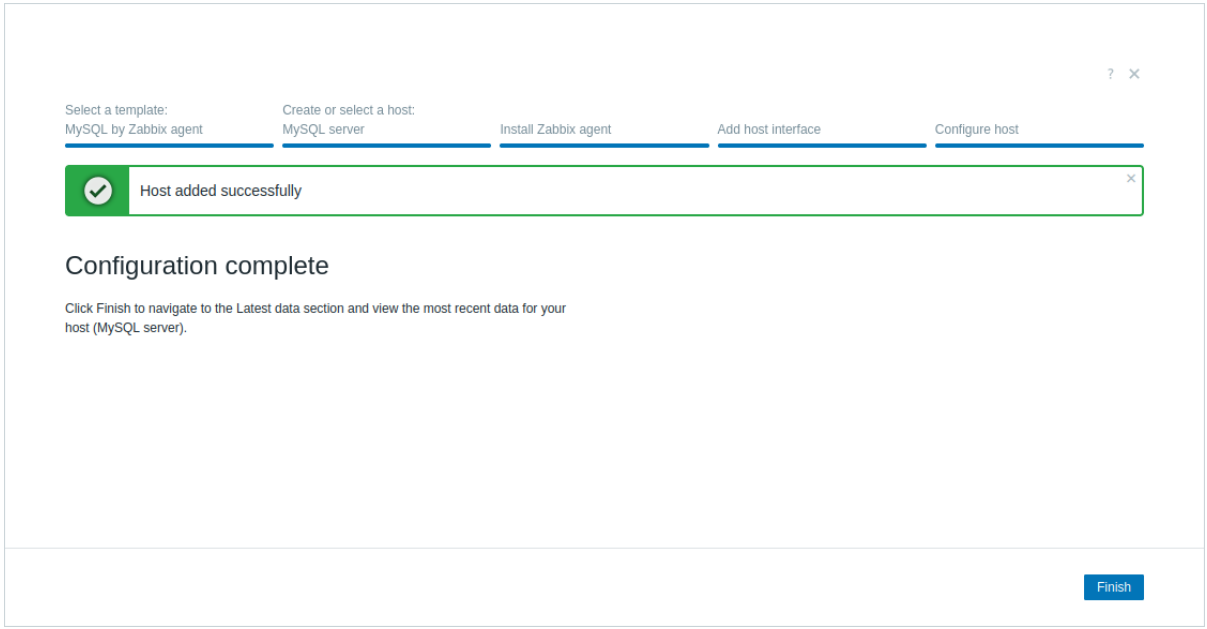
Kad visas nepieciešamās konfigurēšanas darbības ir pabeigtas, noklikšķiniet uz *Create*, lai pievienotu hostu Zabbix (vai uz *Update*, lai atjauninātu esoša hosta konfigurāciju):



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.



Hosts konfigurēšana

Pārskats

Lai konfigurētu hostu Zabbix lietotāja saskarnē, rīkojieties šādi:

1. Dodieties uz *Datu vākšana* > *Hosts* vai *Uzraudzība* > *Hosts*.
2. Noklikšķiniet uz *Izveidot hostu* ekrāna augšējā labajā stūrī (vai uz hosta nosaukuma, lai rediģētu esošu hostu).
3. Formā ievadiet hosta parametrus.

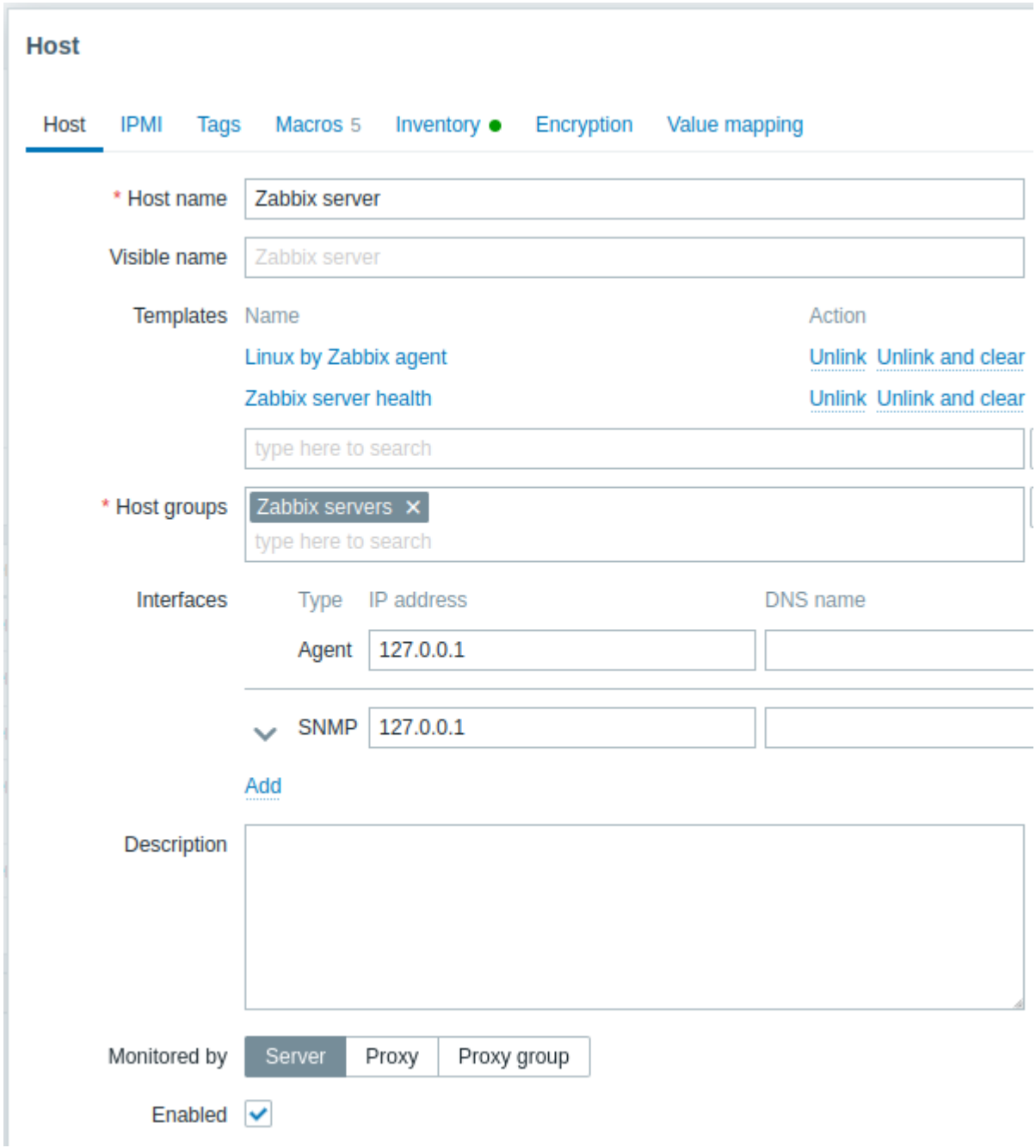
Varat arī izmantot pogu *Klonēt* esoša hosta konfigurācijas formā, lai izveidotu jaunu hostu. Šim hostam būs visas esošā hosta īpašības, tostarp saistītās veidnes, no šīm veidnēm mantotās entītijas (vienumi, trigeri utt.), kā arī entītijas, kas tieši pievienotas esošajam hostam.

Ņemiet vērā, ka, klonējot hostu, tas saglabās visas veidnes entītijas tādas, kādas tās sākotnēji ir veidnē. Jebkuras izmaiņas šajās entītijās, kas veiktas esošā hosta līmenī (piemēram, mainīts vienuma intervāls, modificēta regulārā izteiksme vai pievienoti prototipi zemā līmeņa atklāšanas noteikumam), netiks klonētas uz jauno hostu; tā vietā tās būs tādas pašas kā veidnē.

Alternatīvi varat izmantot Host Wizard, lai konfigurētu hostu, izmantojot vadītu, soli pa solim saskarni.

Konfigurācija

Cilnē **Hosts** ir ietverti vispārīgie hosts atribūti:



Visi obligātie ievades lauki ir atzīmēti ar sarkanu zvaigznīti.

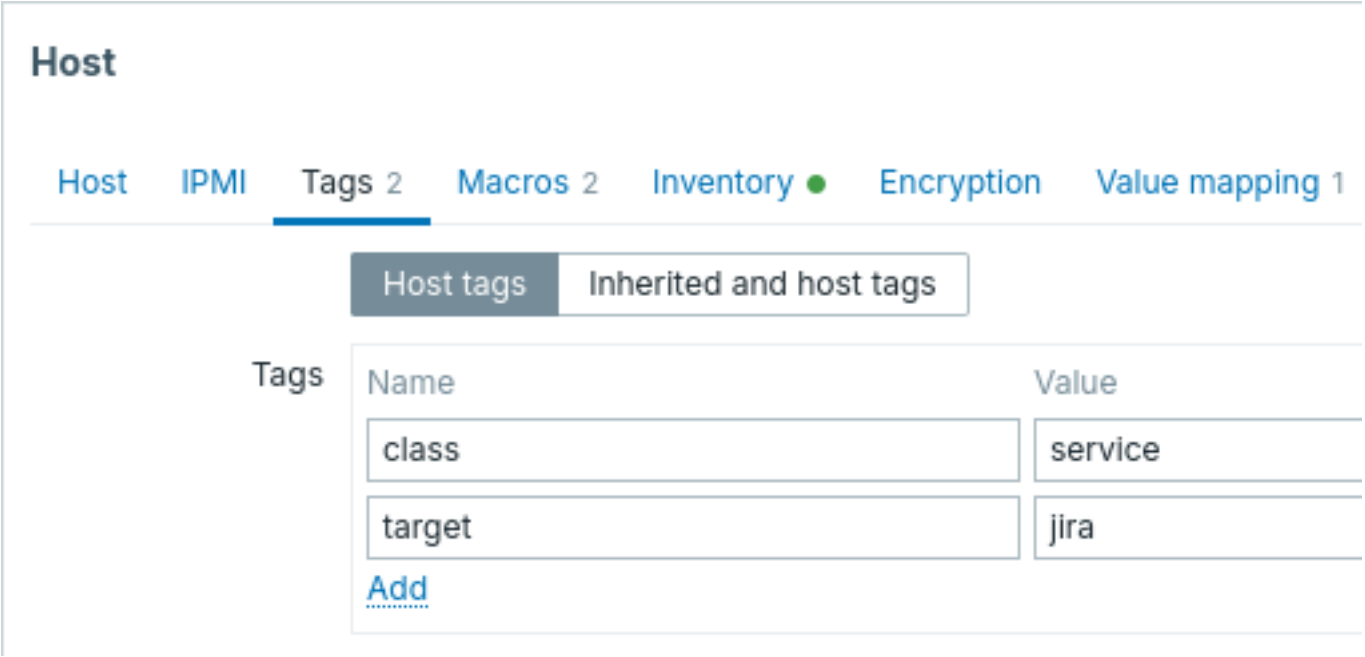
Parametrs	Apraksts
Hosts nosaukums	Ievadiet unikālu hosts nosaukumu. Ir atļauti burti un cipari, atstarpes, punkti, defises un pasvītras. Tomēr sākuma un beigu atstarpes nav atļautas. *Piezīme:* Ja konfigurējamajā hosts darbojas Zabbix aģents, aģenta konfigurācijas faila parametram *Hostname* ir jābūt tādai pašai vērtībai kā šeit ievadītajam hosts nosaukumam. Parametrā norādītais nosaukums ir nepieciešams aktīvo pārbaužu apstrādei.
Redzamais nosaukums	Ievadiet unikālu hosts redzamo nosaukumu. Ja iestatīsiet šo nosaukumu, tas tehniskā hosts nosaukuma vietā būs redzams sarakstos, kartēs u. c. Šis atribūts atbalsta UTF-8.

Parametrs	Apraksts
Veidnes	Piesaistiet veidnes hosts. Visas entītijas (vienumi, trigeri u. c.) tiks mantotas no veidnes. Lai piesaistītu jaunu veidni, sāciet teksta ievades laukā rakstīt veidnes nosaukumu. Tiks parādīts atbilstošo veidņu saraksts; ritiniet to uz leju, lai veiktu atlasi. Varat arī noklikšķināt uz *Atlasīt* blakus laukam un uznirstošajā logā atlasīt veidnes no saraksta. Visas atlasītās veidnes tiks piesaistītas hosts, kad hosts konfigurācijas veidlapa tiks saglabāta vai atjaunināta. Lai atsaistītu veidni, izmantojiet vienu no divām opcijām blokā *Piesaistītās veidnes*: *Atsaistīt* - atsaistīt veidni, bet saglabāt tās entītijas (vienumus, trigerus u. c.); *Atsaistīt un notīrīt* - atsaistīt veidni un noņemt visas tās entītijas (vienumus, trigerus u. c.). Sarakstā norādītie veidņu nosaukumi ir noklikšķināmas saites, kas ved uz veidnes konfigurācijas veidlapu.
Hosts gru-pas	Atlasiet hosts grupas, kurām hosts pieder. Hosts ir jāpieder vismaz vienai hosts grupai. Jaunas grupas izveidi un piesaisti hosts var veikt, pievienojot neesošas grupas nosaukumu.
Saskarnes	Hosts var izmantot vairākus hosts saskarņu tipus: *Aģents*, *SNMP*, *JMX* un *IPMI*. Pēc noklusējuma saskarnes nav definētas. Lai pievienotu jaunu saskarni, blokā *Saskarnes* noklikšķiniet uz *Pievienot*, atlasiet saskarnes tipu un ievadiet informāciju *IP/DNS*, *Savienoties ar* un *Ports*. *Piezīme:* Saskarnes, kas tiek izmantotas kādā vienumā, nevar noņemt, un saite *Noņemt* tām ir neaktīva. SNMP saskarnes "IP" vai "DNS" tiek izmantots arī SNMP trapu saņemšanai. Atbilstības noteikšanas laikā tiek izmantots tikai hosts saskarnē atlasītais "IP" vai "DNS". Papildinformāciju par SNMP saskarnes (v1, v2 un v3) konfigurēšanu skatiet sadaļā SNMP uzraudzības konfigurēšana.
IP adrese	Hosts IP adrese (neobligāti).
DNS nosaukums	Hosts DNS nosaukums (neobligāti).
Savienoties ar	Noklikšķinot uz attiecīgās pogas, tiks norādīts Zabbix serverim, ko izmantot datu iegūšanai no aģentiem: **IP** - savienoties ar hosts IP adresi (ieteicams) **DNS** - savienoties ar hosts DNS nosaukumu
Ports	TCP/UDP porta numurs. Noklusējuma vērtības ir: 10050 Zabbix aģentam, 161 SNMP aģentam, 12345 JMX un 623 IPMI.
Noklusējuma	Atzīmējiet radiopogu, lai iestatītu noklusējuma saskarni.
Apraksts	Ievadiet hosts aprakstu.
Uzrauga	Atlasiet, kas uzrauga hosts: **Serveris** - hosts tiek uzraudzīts ar Zabbix serveri; **Starpniekserveris** - hosts tiek uzraudzīts ar atsevišķu starpniekserveri; **Starpniekserveru grupa** - hosts tiek uzraudzīts ar starpniekserveru grupu.
Starpniekserveris	Tiek parādīts piešķirtā starpniekservera nosaukums (tikai tad, ja Zabbix serveris to ir piešķīris no atlasītās starpniekserveru grupas). Šis lauks tiek parādīts tikai tad, ja hosts uzrauga starpniekserveru grupa.
Iespējots	Ja izvēles rūtiņa ir atzīmēta, hosts ir iespējots - gatavs uzraudzībai. Ja izvēles rūtiņa nav atzīmēta, hosts ir atspējots - tas netiek uzraudzīts: Pasīviem datu pieprasījumiem, ko sāk Zabbix serveris/starpniekserveris (piemēram, Zabbix aģentam, SNMP aģentam, vienkāršajām pārbaudēm), uzraudzība tiek atspējota pēc konfigurācijas sinhronizācijas. Ar hosts saistītie trigeri un darbības tiek atspējotas tikai pēc konfigurācijas atkārtotas ielādes. Zabbix aģenta aktīvajām pārbaudēm uzraudzība tiek pārtraukta laika periodā (aptuveni 5 sekundēs), kurā Zabbix aģents saņem informāciju par hosts atspējošanu. Šajā īsajā intervālā hosts turpinās lokāli vākt aktīvajām pārbaudēm paredzētos datus un mēģinās tos nosūtīt serverim/starpniekserverim; tomēr, tā kā hosts ir atzīmēts kā *Atspējots*, serveris/starpniekserveris datus noraidīs. Atspējojot hosts, tā vienumi tiek nekavējoties noņemti no vēstures kešatmiņas (izņemot to pēdējās vērtības, kas tiek saglabātas žurnāliem).

Cilnē **IPMI** ir ietverti IPMI pārvaldības atribūti.

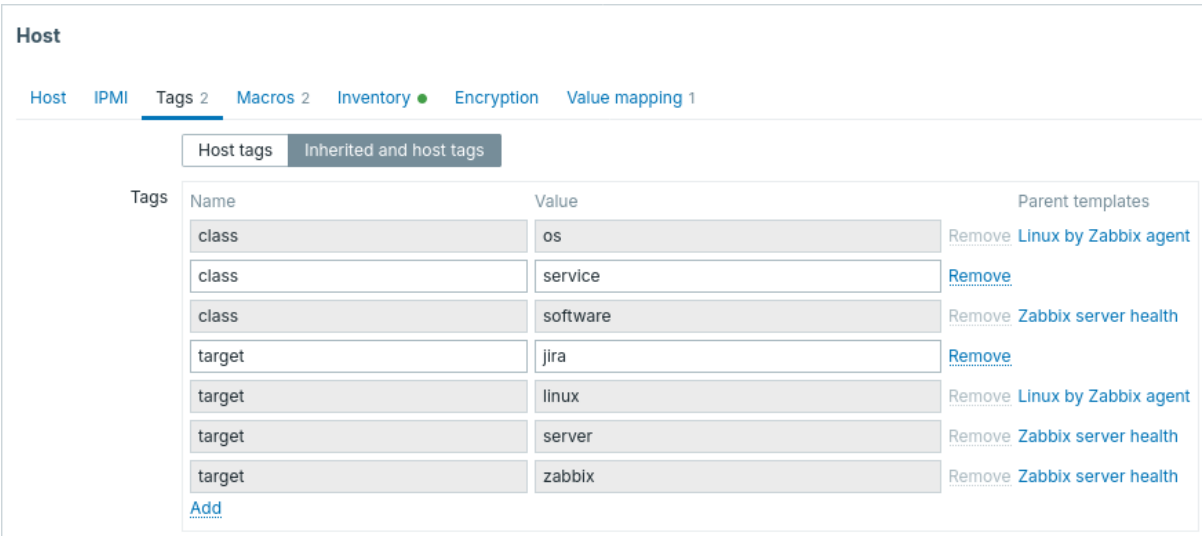
Parametrs	Apraksts
Autentifikācijas algoritms	Atlasiet autentifikācijas algoritmu.
Privilēģiju līmenis	Atlasiet privilēģiju līmeni.
Lietotājvārds	Lietotājvārds autentifikācijai. Var izmantot lietotāja makro.
Parole	Parole autentifikācijai. Var izmantot lietotāja makro.

Cilnē **Tagi** varat definēt hosts līmeņa tagus. Visas šī hosts problēmas tiks marķētas ar šeit ievadītajām vērtībām.



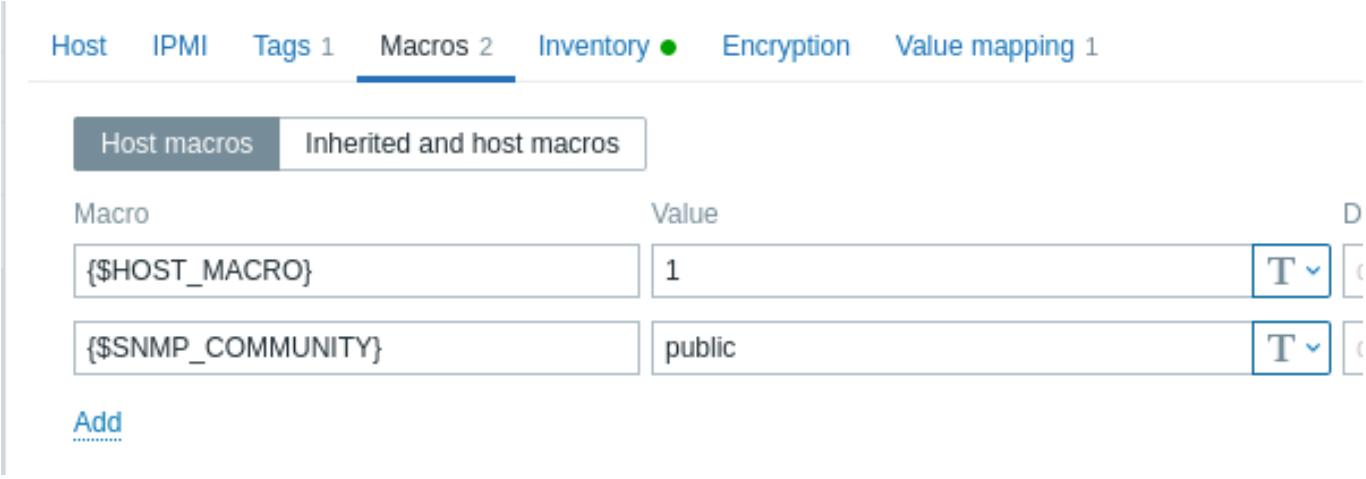
Tagos tiek atbalstīti lietotāja makro, `{INVENTORY.*}` makro, `{HOST.HOST}`, `{HOST.NAME}`, `{HOST.CONN}`, `{HOST.DNS}`, `{HOST.IP}`, `{HOST.PORT}` un `{HOST.ID}` makro.

Šeit varat skatīt arī veidnes līmeņa tagus, ja atlasāt opciju *Mantotie un hosts tagi*. Šeit tiek parādīti visi hosts definētie lietotāja tagi, kā arī to izcelsme.

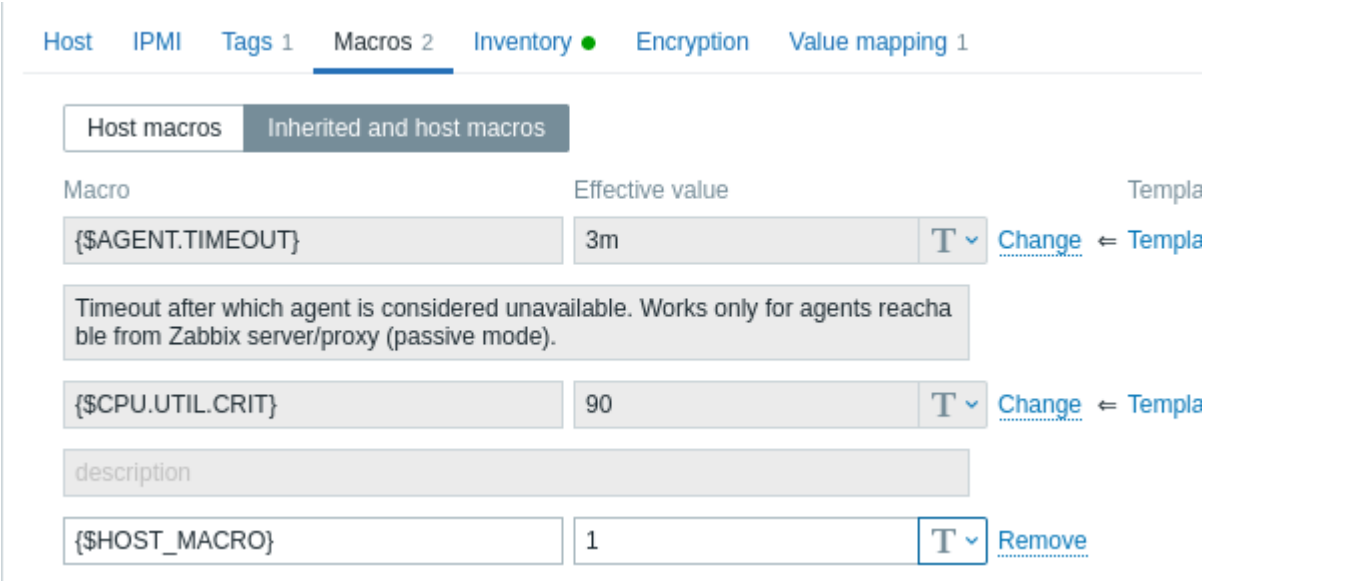


Ērtākai lietošanai ir pieejamas saites uz attiecīgajām veidnēm.

Cilnē **Makro** varat definēt hosts līmeņa lietotāja makro kā nosaukuma un vērtības pārus. Ņemiet vērā, ka makro vērtības var tikt saglabātas kā vienkāršs teksts, slepens teksts vai Vault noslēpums. Varat pievienot arī aprakstu.

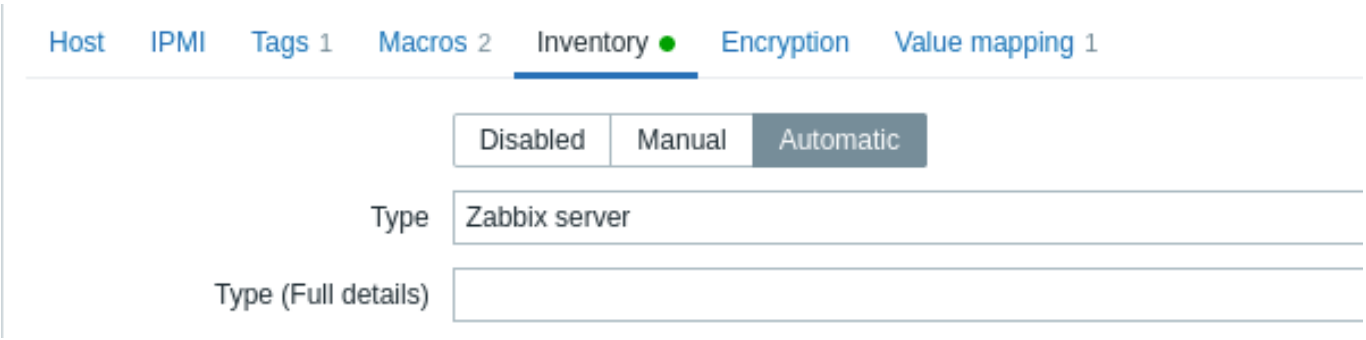


Šeit varat skatīt arī veidnes līmeņa un globālos lietotāja makro, ja atlasāt opciju *Mantotie un hosts makro*. Šeit tiek parādīti visi hosts definētie lietotāja makro, to atrisinātās vērtības un izcelsme.



Ērtākai lietošanai ir pieejamas saites uz attiecīgajām veidnēm un globālo makro konfigurāciju. Veidnes vai globālo makro var arī rediģēt hosts līmenī, tādējādi efektīvi izveidojot makro kopiju hosts.

Cilnē **Inventārs** varat manuāli ievadīt hosts inventāra informāciju. Varat arī izvēlēties iespējot inventāra aizpildīšanu *Automātiski* vai atspējot inventāra aizpildīšanu šim hosts.



Ja inventārs ir iespējots (manuāli vai automātiski), blakus cilnes nosaukumam tiek parādīts zaļš punkts.

Šifrēšana

Cilne **Šifrēšana** ļauj pieprasīt šifrētus savienojumus ar hostu.

Parameter	Description
Connections to host	Kā Zabbix serveris vai starpniekserveris savienojas ar Zabbix aģentu uz hosta: bez šifrēšanas (noklusējums), izmantojot PSK (iepriekš koplietota atslēga) vai sertifikātu.

Parameter	Description
Connections from host	Atlasiet, kāda veida savienojumi ir atļauti no hosta (t. i., no Zabbix aģenta un Zabbix sender). Vienlaikus var atlasīt vairākus savienojumu veidus (noderīgi testēšanai un pārslēgšanai uz citu savienojuma veidu). Noklusējums ir "No encryption".
Issuer	Atļautais sertifikāta izdevējs. Sertifikāts vispirms tiek validēts ar CA (sertifikātu centru). Ja tas ir derīgs un parakstīts ar CA, tad lauku *Issuer* var izmantot, lai tālāk ierobežotu atļauto CA. Šis lauks ir paredzēts lietošanai, ja jūsu Zabbix instalācijā tiek izmantoti sertifikāti no vairākiem CA. Ja šis lauks ir tukšs, tiek pieņemts jebkurš CA.
Subject	Atļautais sertifikāta subjekts. Sertifikāts vispirms tiek validēts ar CA. Ja tas ir derīgs un parakstīts ar CA, tad lauku *Subject* var izmantot, lai atļautu tikai vienu *Subject* virknes vērtību. Ja šis lauks ir tukšs, tiek pieņemts jebkurš derīgs sertifikāts, ko parakstījis konfigurētais CA.
PSK identity	Iepriekš koplietotās atslēgas identitātes virkne. Nevietojiet sensitīvu informāciju PSK identitātē, jo tā tiek pārsūtīta pa tīklu nešifrētā veidā, lai saņēmējam norādītu, kuru PSK izmantot.
PSK	Iepriekš koplietota atslēga (heksadecimāla virkne). Maksimālais garums: 512 heksadecimālie cipari (256 baitu PSK), ja Zabbix izmanto GnuTLS vai OpenSSL bibliotēku, 64 heksadecimālie cipari (32 baitu PSK), ja Zabbix izmanto mbed TLS (PolarSSL) bibliotēku. Piemērs: 1f87b595725ac58dd977beef14b97461a7c1045b9a1c963065002c5473194952

Vērtību kartēšana

Cilne **Vērtību kartēšana** ļauj konfigurēt vienuma datu lietotājam saprotamu attēlojumu sadaļā vērtību kartējumi.

Hosts grupas konfigurēšana

Pārskats

Host grupas tiek izmantotas hostu loģiskai grupēšanai un lietotāju atļauju piešķiršanai tiem.

Katram hostam ir jābūt piešķirtai vismaz vienai host grupai. Hosts var piederēt vairākām host grupām, un katrā host grupā var būt vairāki hosts.

Ņemiet vērā, ka Zabbix visas atļaujas ir balstītas uz grupām: lietotāju grupām, host grupām un veidņu grupām. Tāpēc pat tad, ja vienam lietotājam ir nepieciešama piekļuve vienam hostam, tā tiek piešķirta, pievienojot lietotāju lietotāju grupai un piešķirot šai lietotāju grupai atļaujas host grupai, kurā ietilpst šis hosts.

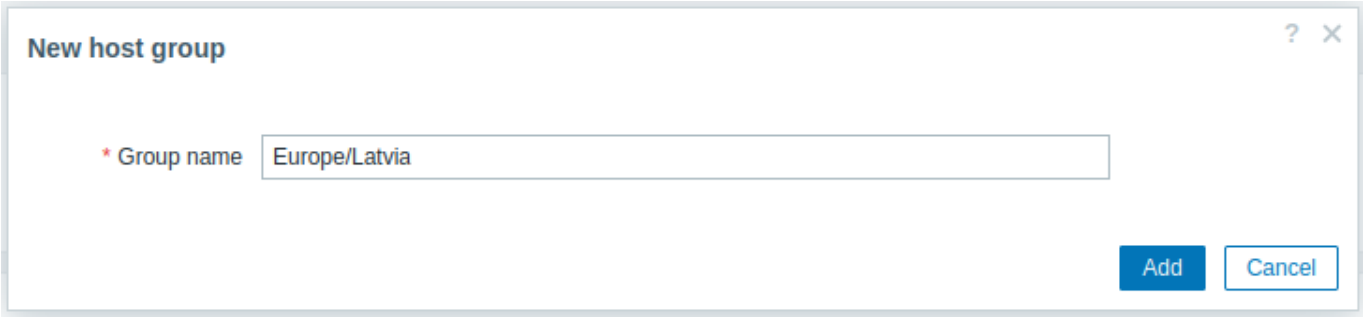
Konfigurācija

> **Attention:**
> Tikai Super admin lietotāji var izveidot hostu grupas.

Zabbix lietotāja saskarnē ir divas hostu grupas izveides iespējas.

Pirmā iespēja:

1. Dodieties uz *Datu vākšana* > *Hostu grupas*.
2. Ekrāna augšējā labajā stūrī noklikšķiniet uz *Izveidot hostu grupu*.
3. Formā ievadiet grupas nosaukumu.



Otrā iespēja: konfigurējot hostu, laukā *Hostu grupas* ievadiet neesošas grupas nosaukumu.

Kad hostu grupa ir izveidota, sarakstā zem *Datu vākšana* > *Hostu grupas* varat noklikšķināt uz grupas nosaukuma, lai rediģētu grupas nosaukumu, klonētu grupu vai dzēstu grupu.

Dzēšot hostu grupu, tiek dzēsta tikai loģiskā grupa, nevis grupā esošie hosts. 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`.

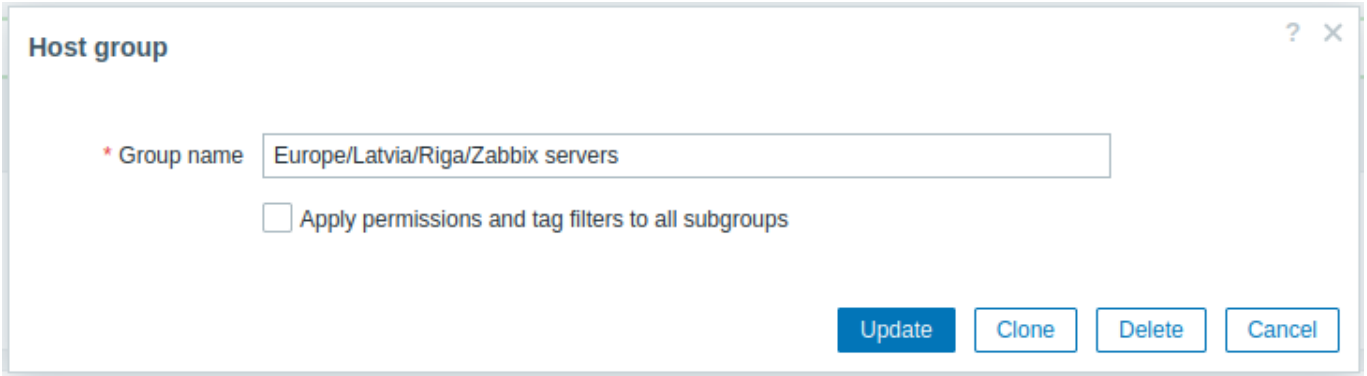
Veidojot apakšgrupu, nav atļauts izmantot slīpsvītras sākumā vai beigās, kā arī vairākas slīpsvītras pēc kārtas. '/' ekrānēšana netiek atbalstīta.

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 vecākajai hostu grupai (piemēram, veidojot `Europe/Latvia`, ja `Europe` jau pastāv), lietotāju grupas atļaujas apakšgrupai tiek mantotas no vecākās grupas.
- Veidojot vecāko hostu grupu esošai apakšgrupai (piemēram, veidojot `Europe`, ja `Europe/Latvia` jau pastāv), vecākajai grupai atļaujas netiek iestatītas.

Rediģējot jebkuru hostu grupu, varat iestatīt arī papildu opciju *Piemērot atļaujas un tagu filtrus visām apakšgrupām*.



Atzīmējot šo izvēles rūtiņu un noklikšķinot *Atjaunināt*, tiks piemērots tāds pats atļauju un tagu filtru līmenis visām pašreizējām un nākotnē izveidotajām rediģējamās hostu grupas apakšgrupām.

Tātad, ja kādām lietotāju grupām ir piešķirtas atšķirīgas atļaujas rediģējamās hostu grupas apakšgrupām, izvēles rūtiņas atzīmēšana piešķirs visām pašreizējām un nākotnē izveidotajām apakšgrupām tādas pašas lietotāju atļaujas un uz tagiem balstītās atļaujas kā rediģējamajai grupai.

Ņemiet vērā, ka šī opcija netiek saglabāta datubāzē un pārrakstīs esošās atļaujas. Jebkuras izmaiņas, kas veiktas, izmantojot šo opciju, var atsaukt tikai manuāli.

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 tipu, sērijas numuru, atrašanās vietu, atbildīgo personu, URL u. c. — datus, kas aizpildīs inventāra informāciju.

Ja hosta inventāra informācijā ir iekļauts URL, kas sākas ar "http" vai "https", sadaļā *Inventārs* tas tiks attēlots kā noklikšķināma saite.

Automātiskais režīms

Hosts inventāru var aizpildīt arī automātiski. Lai tas darbotos, konfigurējot hostu, inventāra režīmam cilnē *Inventārs* jābūt iestatītam uz *Automātisks*.

Pēc tam varat konfigurēt hosta vienumus, lai aizpildītu jebkuru hosta inventāra lauku ar to vērtību, vienumu konfigurācijā norādot mērķa lauku ar atbilstošo atribūtu (sauktu *Vienums aizpildīs hosta inventāra lauku*).

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 regulāra 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 veidlapā.

Jauniem hostiem inventarizācijas režīms pēc noklusējuma tiek izvēlēts, pamatojoties uz iestatījumu *Default host inventory mode* sadaļā *Administration* > *General* > *Other*.

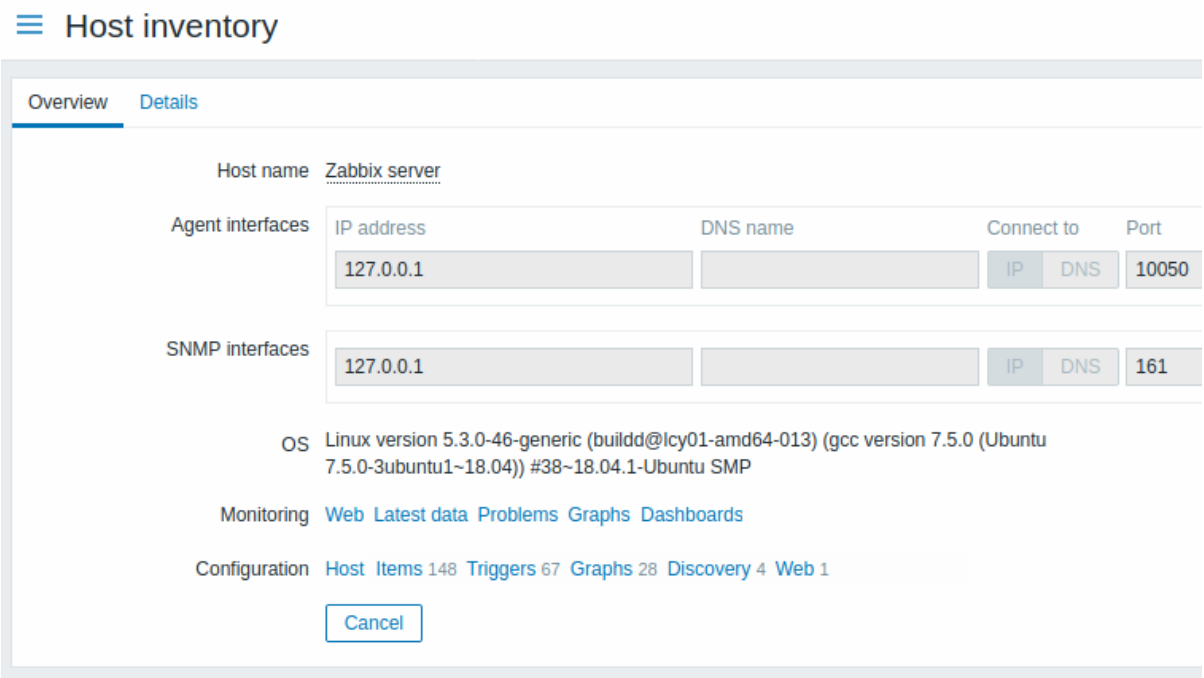
Hostiem, kas pievienoti, izmantojot tīkla atklāšanas vai automātiskās reģistrācijas darbības, ir iespējams definēt darbību *Set host inventory mode*, izvēloties manuālo vai automātisko režīmu. Šī darbība ignorē iestatījumu *Default host inventory mode*.

Inventāra pārskats

Visu esošo inventāra datu informācija ir pieejama izvēlnē *Inventory*.

Sadaļā *Inventory* > *Overview* varat iegūt hosts skaitu pēc dažādiem inventāra laukiem.

Sadaļā *Inventory* > *Hosts* varat redzēt visus hosts, kuriem ir inventāra informācija. Noklikšķinot uz hosta nosaukuma, tiks parādīta inventāra informācija formā.



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 ir apkopē.
Visible name	Hosta redzamais nosaukums (ja definēts).
Host (Agent, SNMP, JMX, IPMI) interfaces	Šis bloks sniedz informāciju par hostam konfigurētajām saskarnēm.
OS	Hosta operētājsistēmas inventāra lauks (ja definēts).

Parameter	Description
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 šī hosta konfigurācijas sadaļām: *Host*, *Items*, *Triggers*, *Graphs*, *Discovery*, *Web*. Konfigurēto entītiju skaits ir norādīts pēc katras saites.

Cilnē **Details** tiek rādīti visi aizpildītie inventāra lauki (tukšie lauki netiek rādīti).

Inventāra makrokomandas

Ir pieejamas host inventāra makrokomandas `{INVENTORY.*}`, ko var izmantot paziņojumos, piemēram:

”Serveris {INVENTORY.LOCATION1} ir radusies problēma, atbildīgā persona ir {INVENTORY.CONTACT1}, tālruņa numurs {INVENTORY.POC.PRIMARY.PHONE.A1}.”

Plašāku informāciju skatiet lapā Atbalstītās makrokomandas.

Masveida atjaunināšana

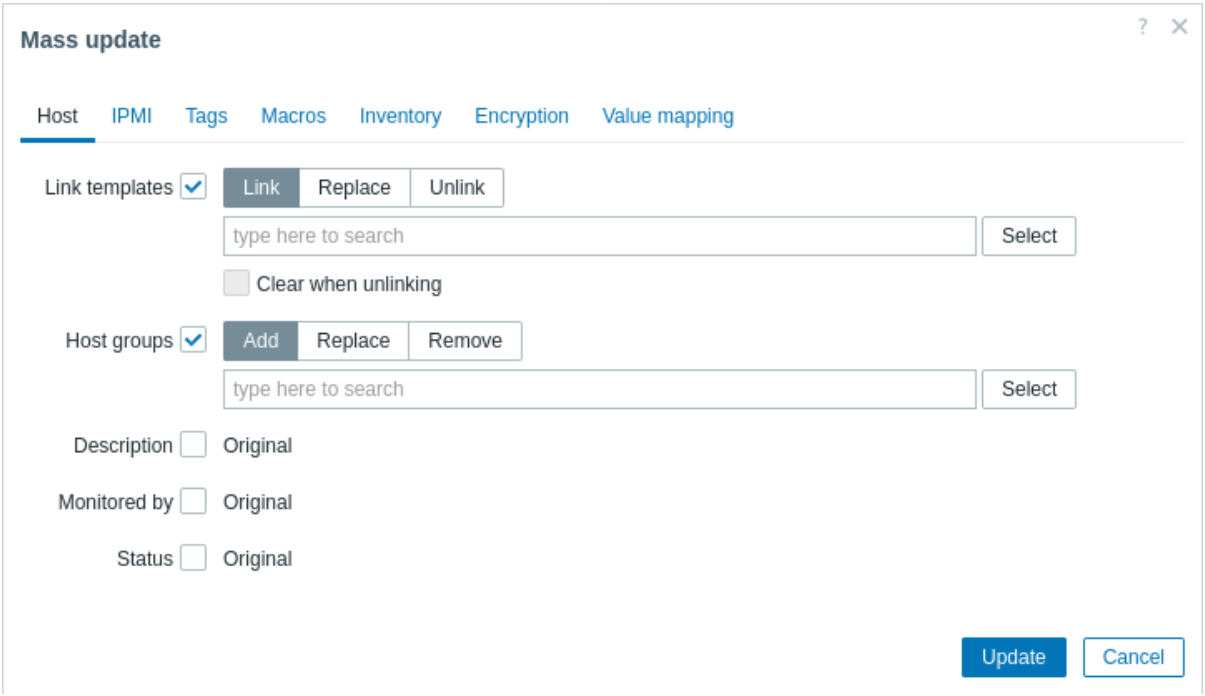
Pārskats

Dažkārt var būt nepieciešams vienlaikus mainīt kādu atribūtu vairākiem hostiem. Tā vietā, lai atvērtu katru atsevišķo hostu rediģēšanai, varat izmantot masveida atjaunināšanas funkciju.

Masveida atjaunināšana

Lai masveidā atjauninātu dažus hostus, rīkojieties, kā norādīts tālāk:

- Atzīmējiet izvēles rūtiņas pirms hostiem, kurus vēlaties atjaunināt, hostu sarakstā
- Zem saraksta noklikšķiniet uz *Masveida atjaunināšana*
- Atveriet cilni ar nepieciešamajiem atribūtiem (*Hosts*, *IPMI*, *Tagi*, *Makrosi*, *Inventārs*, *Šifrēšana* vai *Vērtību kartēšana*)
- Atzīmējiet jebkura atjaunināmā atribūta izvēles rūtiņu un ievadiet tam jaunu vērtību



Izvēloties attiecīgo pogu **veidnes** saišu atjaunināšanai, ir pieejamas šādas opcijas:

- *Saistīt* - norādiet, kuras papildu veidnes saistīt
- *Aizstāt* - norādiet, kuras veidnes saistīt, vienlaikus atsaistot visas veidnes, kas iepriekš bija saistītas ar hostiem
- *Atsaistīt* - norādiet, kuras veidnes atsaistīt

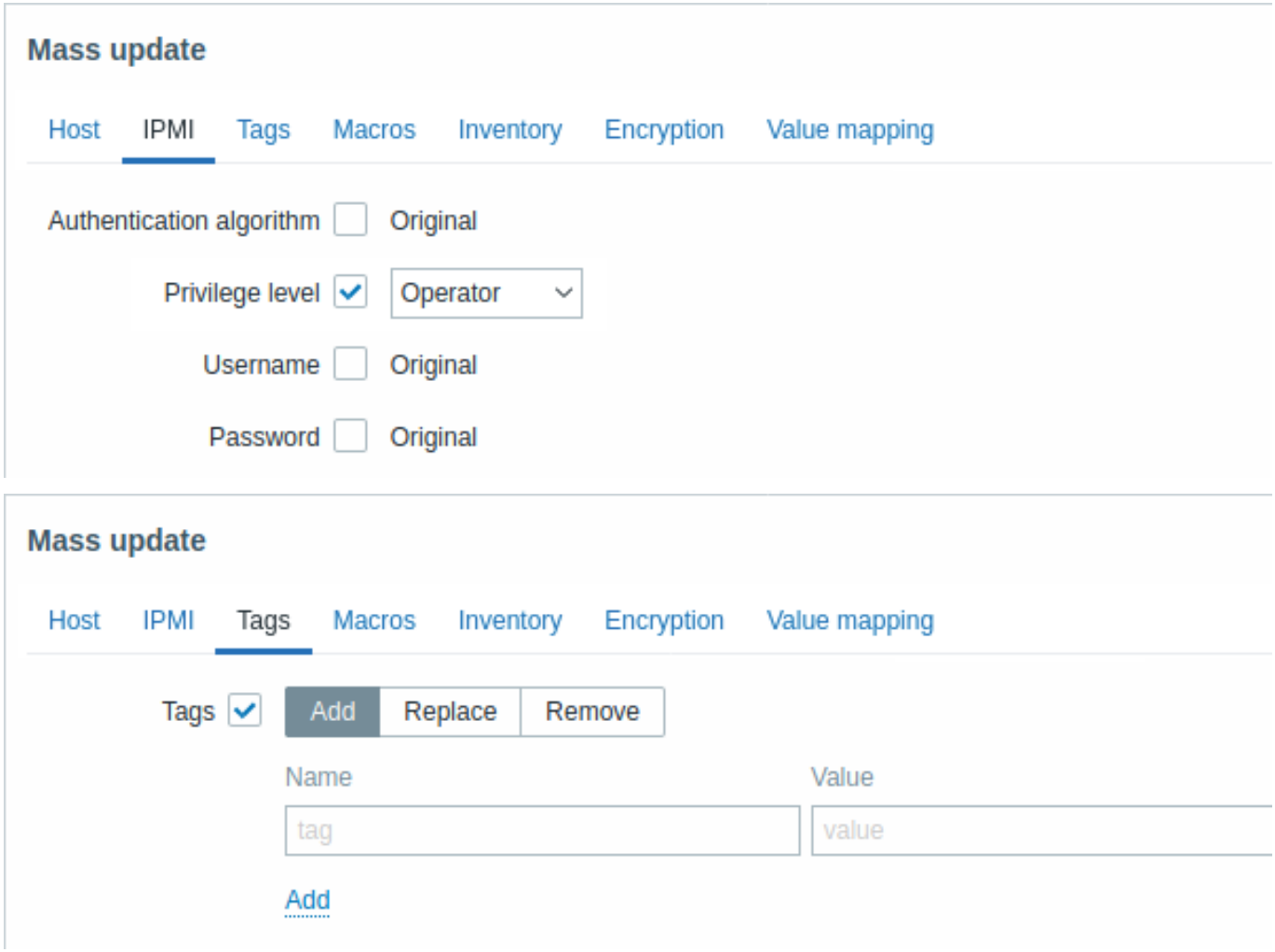
Lai norādītu saistāmās vai atsaistāmās veidnes, automātiskās pabeigšanas laukā sāciet rakstīt veidnes nosaukumu, līdz tiek parādīts nolaižamais saraksts ar atbilstošajām veidnēm. Ritiniet sarakstu uz leju, lai atlasītu nepieciešamo veidni.

Opcija *Notīrīt, atsaistot* ļauj ne tikai atsaistīt iepriekš saistītās veidnes, bet arī noņemt visas no tām mantotās entītijas (vienumus, trigerus, grafikus u. c.).

Izvēloties attiecīgo pogu **hostu grupu** atjaunināšanai, ir pieejamas šādas opcijas:

- *Pievienot* - norādiet papildu hostu grupas no esošajām vai ievadiet hostiem pilnīgi jaunas hostu grupas;
- *Aizstāt* - noņemiet hostu no visām esošajām hostu grupām un aizstājiet tās ar šajā laukā norādīto grupu vai grupām (esošām vai jaunām hostu grupām);
- *Noņemt* - noņemiet konkrētas hostu grupas no hostiem.

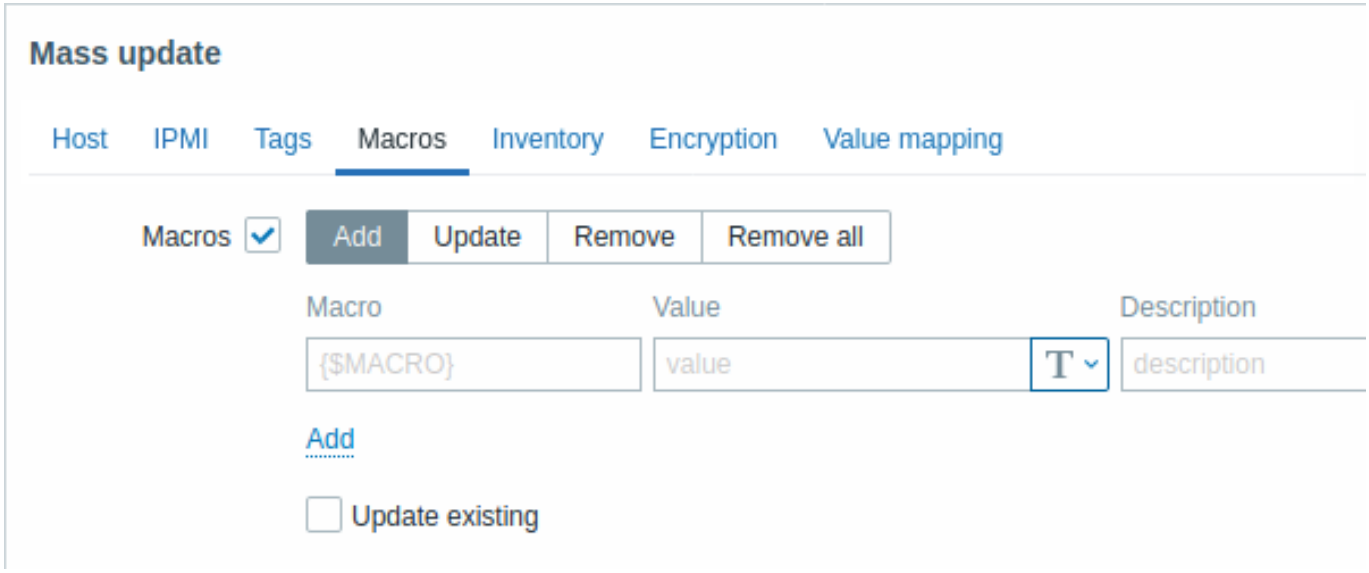
Šie lauki atbalsta automātisko pabeigšanu - sākot tajos rakstīt, tiek piedāvāts nolaižamais saraksts ar atbilstošajām hostu grupām. Ja hostu grupa ir jauna, tā arī tiek parādīta nolaižamajā sarakstā, un aiz virknes tiek norādīts *(jauna)*. Ritiniet sarakstu uz leju, lai atlasītu grupu.



Izvēloties attiecīgo pogu tagu atjaunināšanai, ir pieejamas šādas opcijas:

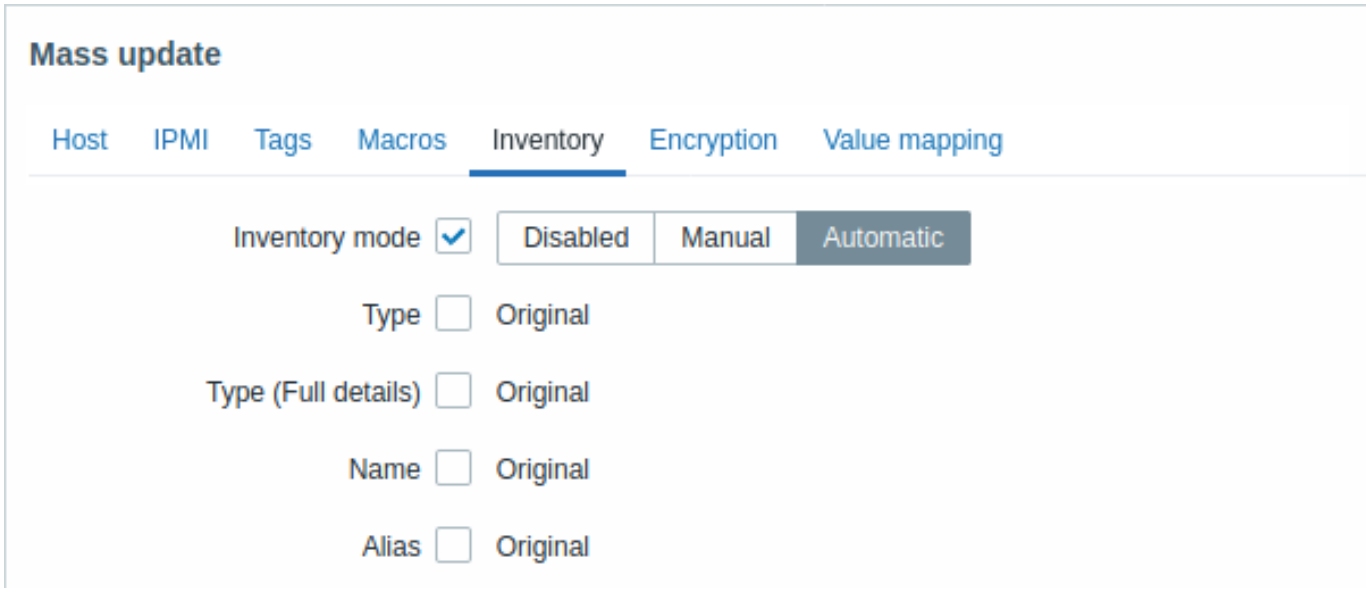
- *Pievienot* - pievienojiet hostiem jaunus tagus;
- *Aizstāt* - noņemiet no hosta visus esošos tagus un aizstājiet tos ar tālāk norādīto tagu vai tagiem;
- *Noņemt* - noņemiet no hostiem norādītos tagus.

Tagos tiek atbalstīti lietotāja makrosi, `{INVENTORY.*}` makrosi, `{HOST.HOST}`, `{HOST.NAME}`, `{HOST.CONN}`, `{HOST.DNS}`, `{HOST.IP}`, `{HOST.PORT}` un `{HOST.ID}` makrosi. Ņemiet vērā, ka tagi ar vienādu nosaukumu, bet atšķirīgām vērtībām netiek uzskatīti par "dublikātiem", un tos var pievienot vienam un tam pašam hostam.

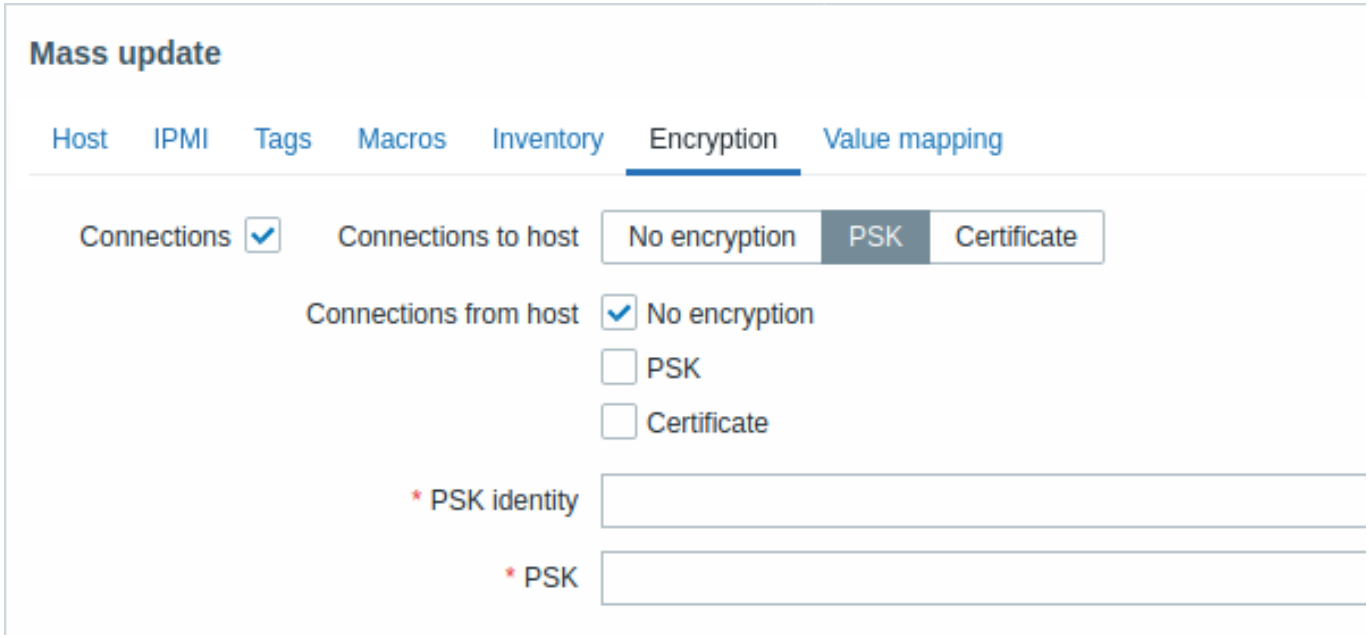


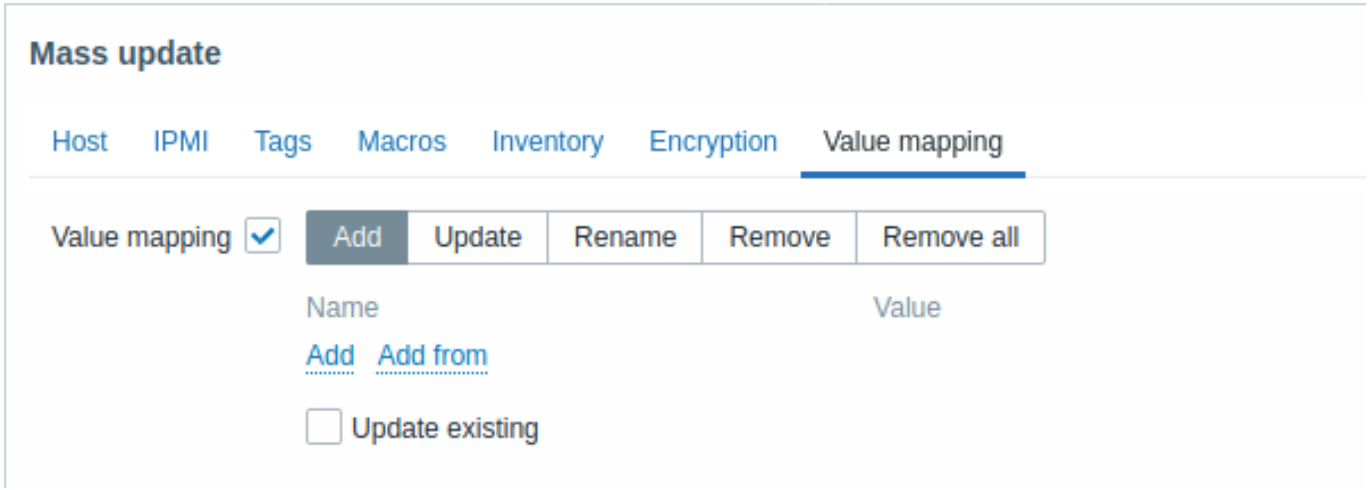
Izvēloties attiecīgo pogu makrosu atjaunināšanai, ir pieejamas šādas opcijas:

- *Pievienot* - norādiet hostiem papildu lietotāja makrosus. Ja ir atzīmēta izvēles rūtiņa *Atjaunināt esošos*, norādītā makrosa nosaukuma vērtība, tips un apraksts tiks atjaunināti. Ja tā nav atzīmēta un hostā vai hostos jau pastāv makross ar šādu nosaukumu, tas netiks atjaunināts.
- *Atjaunināt* - aizstājiet šajā sarakstā norādīto makrosu vērtības, tipus un aprakstus. Ja ir atzīmēta izvēles rūtiņa *Pievienot trūkstošos*, makross, kas iepriekš hostā nepastāvēja, tiks pievienots kā jauns makross. Ja tā nav atzīmēta, tiks atjaunināti tikai makrosi, kas hostā jau pastāv.
- *Noņemt* - noņemiet no hostiem norādītos makrosus. 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* - noņemiet no hostiem visus lietotāja makrosus. Ja nav atzīmēta izvēles rūtiņa *Es apstiprinu visu makrosu noņemšanu*, tiks atvērts jauns uznirstošais logs ar lūgumu apstiprināt visu makrosu noņemšanu.



Lai varētu masveidā atjaunināt inventāra laukus, *Inventāra režīmam* jābūt iestatītam uz ”Manuāls” vai ”Automātisks”.





Vērtību kartējuma atjaunināšanai ir pieejamas pogas ar šādām opcijām:

- *Pievienot* - pievienojiet hostiem vērtību kartējumus. Ja atzīmējat *Atjaunināt esošos*, tiks atjaunināti visi vērtību kartējuma ar šo nosaukumu rekvizīti. Pretējā gadījumā, ja vērtību kartējums ar šādu nosaukumu jau pastāv, tas netiks atjaunināts.
- *Atjaunināt* - atjauniniet esošos vērtību kartējumus. Ja atzīmējat *Pievienot trūkstošos*, vērtību kartējums, kas iepriekš hostā nepastāvēja, tiks pievienots kā jauns vērtību kartējums. Pretējā gadījumā tiks atjaunināti tikai vērtību kartējumi, kas hostā jau pastāv.
- *Pārdēvēt* - piešķiriet esošam vērtību kartējumam jaunu nosaukumu
- *Noņemt* - noņemiet no hostiem norādītos vērtību kartējumus. Ja atzīmējat *Izņemot atlasītos*, visi vērtību kartējumi tiks noņemti, **izņemot** norādītos.
- *Noņemt visus* - noņemiet no hostiem visus vērtību kartējumus. Ja nav atzīmēta izvēles rūtiņa *Es apstiprinu visu vērtību kartējumu noņemšanu*, tiks atvērts jauns uznirstošais logs ar lūgumu apstiprināt noņemšanu.

Kad visas nepieciešamās izmaiņas ir veiktas, noklikšķiniet uz *Atjaunināt*. Atribūti tiks atbilstoši atjaunināti visiem atlasītajiem hostiem.

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.

Vienuma izveide

Pārskats

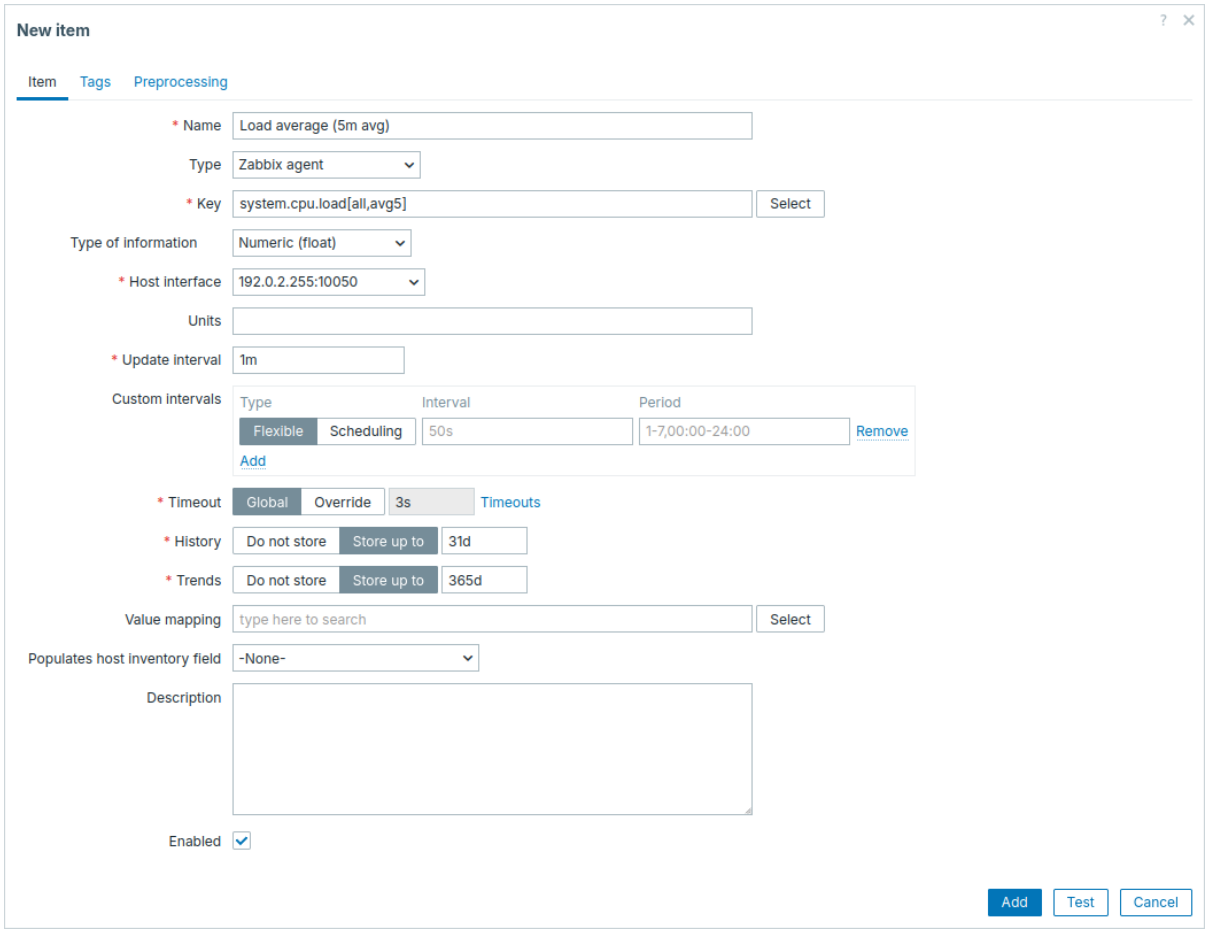
Lai izveidotu vienumu Zabbix lietotāja saskarnē, rīkojieties, kā norādīts tālāk:

1. Atveriet *Data collection* > *Hosts*.
2. Hosts rindā noklikšķiniet uz *Items*.
3. Ekrāna augšējā labajā stūrī noklikšķiniet uz *Create item*.
4. Veidlapā ievadiet vienuma parametrus.

Varat arī izveidot vienumu, atverot esošu vienumu, noklikšķinot uz pogas *Clone* un saglabājot to ar citu nosaukumu.

Konfigurācija

Vienums cilnē ir ietverti vispārīgie vienuma atribūti.



Visi obligātie ievades lauki ir atzīmēti ar sarkanu zvaigznīti.

Parametrs	Apraksts
Name	Vienuma nosaukums.<br[Tiek atbalstīti lietotāja makro.
Type	Vienuma tips. Skatiet atsevišķās vienumu tipu sadaļas.

Parametrs	Apraksts
Key	Vienuma atslēga (līdz 2048 rakstzīmēm). Tiek atbalstītas vienumu atslēgas, kas norādītas atsevišķajās vienumu tipu sadaļās. Atslēgai ir jābūt unikālai viena hosta ietvaros. Ja atslēgas tips ir “Zabbix agent”, “Zabbix agent (active)” vai “Simple check”, atslēgas vērtībai ir jābūt atbalstītai Zabbix aģentā vai Zabbix serverī. Skatiet arī: pareizais atslēgas formāts.
Type of information	Datu tips, kas tiek izmantots vienuma vērtības validēšanai un glabāšanai datubāzē, tostarp pēc jebkādas konvertēšanas: **Numeric (unsigned)** - 64 bitu vesels skaitlis bez zīmes; **Numeric (float)** - 64 bitu peldošā komata skaitlis; **Character** - īss teksts; **Log** - garš teksts ar žurnāla īpašībām vai bez tām (laikspiedols, avots, nopietnība, žurnāla notikuma ID); **Text** - garš teksts; **Binary** - binārs skaitlis (tiek atbalstīts tikai atkarīgajiem vienumiem); **JSON** - strukturēti JSON dati, kas tiek glabāti datubāzē sākotnējā formātā (netiek atbalstīts aprēķinātajiem vienumiem). Vienumi, kas atgriež binära vai JSON datu tipa vērtības, netiek atbalstīti aprēķināto vienumu formulās vai trigeru izteiksmēs. Glabāšanas ierobežojumus katram datu tipam un datubāzes aizmugursistēmai skatiet sadaļā Vienumu datu ierobežojumi. Vienumiem, kas atgriež datus vienā noteiktā formātā, atbilstošais datu tips tiek atlasīts automātiski, piemēram, `system.cpu.load` un `system.cpu.util` abi atgriež 64 bitu peldošā komata skaitli, tāpēc *Numeric (float)* tiek atlasīts automātiski.
Host interface	Atlasiet hosta saskarni. Šis lauks ir pieejams, rediģējot vienumu hosta līmenī.
Units	Ja ir iestatīts mērvienības simbols, Zabbix veic saņemtās vienuma vērtības pēcapstrādi un parāda to ar norādīto mērvienības sufiksu. Atbalstītie mērvienību simboli ar īpašu formatēšanu (un saņemtās vienuma vērtības → parādītās vērtības piemēri): **B** - baiti (1024 → 1 KB) **Bps** - baiti sekundē (1024 → 1 KBps) **s** - sekundes, attēlotas, izmantojot līdz trim lielākajām vienībām, kas nav nulle (881764 → 10d 4h 56m) **uptime** - pagājušais laiks formātā hh:mm:ss vai N days, hh:mm:ss (881764 → 10 days, 04:56:04) **unixtime** - Unix laikspiedols, formatēts kā yyyy.mm.dd hh:mm:ss (881764 → 1970-01-11 04:56:04 AM); pareizai formatēšanai saņemtajai vienuma vērtībai ir jābūt *Numeric (unsigned)*. Citām mērvienībām (piemēram, Hz, W u. c.), ja saņemtā vērtība pārsniedz 1000, tā tiek dalīta ar 1000 un parādīta ar atbilstošu prefiksu (5000 → 5 KHz, 881764 → 881.76 KW). Ja mērvienības simbola priekšā ir `!`, mērvienību konvertēšana un prefiksu pievienošana tiek atspējota (1024 !B → 1024 B, 61 !s → 61 s). Vairāk piemēru un detalizētu informāciju par mērvienību simboliem un mērvienību konvertēšanu skatiet sadaļā Vienuma vērtību sufiksi.
Update interval	Iegūt jaunu šī vienuma vērtību ik pēc N sekundēm. Maksimālais atļautais atjaunināšanas intervāls ir 86400 sekundes (1 diena). Tiek atbalstīti laika sufiksi, piemēram, 30s, 1m, 2h, 1d. Tiek atbalstīti lietotāja makro. Vienam makro ir jāaizpilda viss lauks. Vairāki makro vienā laukā vai makro 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 ieplānots) ar vērtību, kas nav nulle, vienums tiks aptaujāts pielāgotā intervāla laikā. *Ņemiet vērā*, ka pirmā vienuma aptauja pēc vienuma aktivizēšanas vai pēc atjaunināšanas intervāla maiņas var notikt agrāk par konfigurēto vērtību. Jauni vienumi tiks pārbaudīti 60 sekunžu laikā pēc to izveides, izņemot gadījumus, kad tiem ir Scheduling vai Flexible atjaunināšanas intervāls un *Update interval* ir iestatīts uz 0. Esošu pasīvu vienumu var nekavējoties aptaujāt, nospiežot *Execute now* pogu.

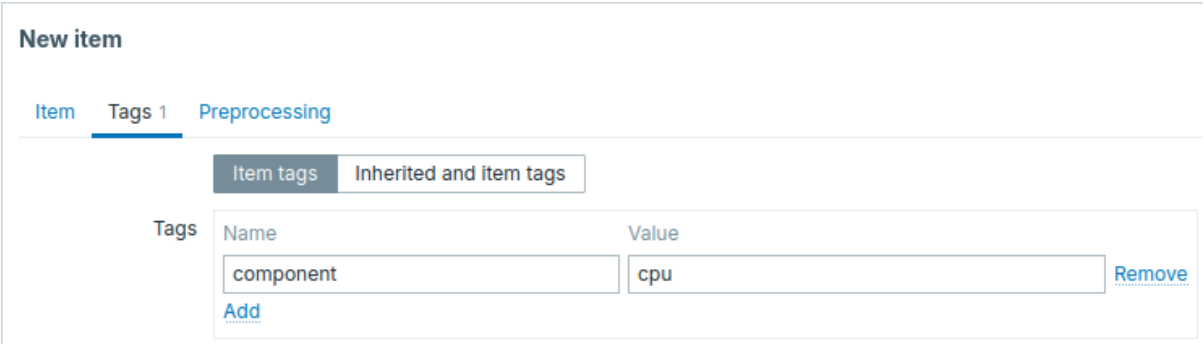
Parametrs	Apraksts
Custom intervals	Varat izveidot pielāgotus noteikumus vienuma pārbaudei: **Flexible** - izveidot izņēmumu *Update interval* (intervālu ar citu biežumu). **Scheduling** - izveidot pielāgotu aptaujas grafiku. Detalizētu informāciju skatiet sadaļā Pielāgotie intervāli. *Interval* laukā tiek atbalstīti laika sufiksi, piemēram, 30s, 1m, 2h, 1d. Tiek atbalstīti lietotāja makro. Vienam makro ir jāaizpilda viss lauks. Vairāki makro vienā laukā vai makro kopā ar tekstu netiek atbalstīti.
Timeout	Iestatiet vienuma pārbaudes taimautu (pieejams atbalstītajiem vienumu tipiem). Atlasiet taimauta opciju: **Global** - tiek izmantots starpniekservera vai globālais taimauts (parādīts pelēkotajā *Timeout* laukā). **Override** - tiek izmantots pielāgots taimauts (iestatīts *Timeout* laukā; atļautais diapazons: 1–600 s). Tiek atbalstīti laika sufiksi, piemēram, 30s, 1m, un lietotāja makro. Noklikšķinot uz saites *Timeouts*, varat konfigurēt starpniekservera taimautus vai globālos taimautus (ja starpniekserveris netiek izmantots). Ņemiet vērā, ka saite *Timeouts* ir redzama tikai lietotājiem ar *Super admin* tipu un atļaujām piekļūt *Administration* > *General* vai *Administration* > *Proxies* lietotāja saskarnes sadaļām.
History	Atlasiet vienu no opcijām: **Do not store** - vienuma vēsture netiek glabāta. Noder galvenajiem vienumiem, ja vēsture ir jāglabā tikai atkarīgajiem vienumiem. Šo iestatījumu nevar ignorēt globālie housekeeper iestatījumi. **Store up to** - norādiet detalizētas vēstures glabāšanas ilgumu datubāzē (no 1 stundas līdz 25 gadiem). Vecākus datus noņems housekeeper. Tiek glabāts sekundēs. Tiek atbalstīti laika sufiksi, piemēram, 2h, 1d. Tiek atbalstīti lietotāja makro. Vērtību *Store up to* var globāli ignorēt sadaļā *Administration* > *Housekeeping*. Ja pastāv globāls ignorējošs iestatījums, tiek parādīta oranža informācijas ikona. Novietojot uz tās peles rādītāju, tiek parādīts brīdinājuma ziņojums, piemēram, *Overridden by global housekeeper settings (1d)*. Ieteicams ierakstītās vērtības glabāt pēc iespējas īsāku laiku, lai samazinātu vērtību vēstures izmēru datubāzē. Tā vietā, lai glabātu ilgu vērtību vēsturi, varat ilgāk glabāt tendenču datus. Skatiet arī Vēsture un tendences.
Trends	Atlasiet vienu no opcijām: **Do not store** - tendences netiek glabātas. Šo iestatījumu nevar ignorēt globālie housekeeper iestatījumi. **Store up to** - norādiet apkopotās vēstures (stundas minimums, maksimums, vidējā vērtība, skaits) glabāšanas ilgumu datubāzē (no 1 dienas līdz 25 gadiem). Vecākus datus noņems housekeeper. Tiek glabāts sekundēs. Tiek atbalstīti laika sufiksi, piemēram, 24h, 1d. Tiek atbalstīti lietotāja makro. Vērtību *Store up to* var globāli ignorēt sadaļā *Administration* > *Housekeeping*. Ja pastāv globāls ignorējošs iestatījums, tiek parādīta oranža informācijas ikona. Novietojot uz tās peles rādītāju, tiek parādīts brīdinājuma ziņojums, piemēram, *Overridden by global housekeeper settings (7d)*. *Piezīme:* tendenču glabāšana nav pieejama neciparu datiem - rakstzīmju, žurnāla un teksta datiem. Skatiet arī Vēsture un tendences.
Value mapping	Piemērojiet šim vienumam vērtību kartēšanu. Vērtību kartēšana nemaina saņemtās vērtības, tā paredzēta tikai datu attēlošanai. Tā darbojas ar *Numeric(unsigned)*, *Numeric(float)* un *Character* vienumiem. Piemēram, “Windows service states”.

Parametrs	Apraksts
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 ir tukšs, laikspiedols Unix laikā tiks iestatīts uz 0, kas apzīmē 1970. gada 1. janvāri. Piemēram, aplūkosim šo rindu no Zabbix aģenta žurnālfaila: “ 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 “p” un “:” rakstzīmes ir vietturi un var būt jebkuras rakstzīmes, izņemot “yMdhms”.
Populates host inventory field	Varat atlasīt hosta inventāra lauku, kurā tiks ievadīta vienuma vērtība. Tas darbosies, ja hostam ir iespējota automātiska inventāra aizpildīšana un tiek izmantoti *Numeric(unsigned)*, *Numeric(float)*, *Character* vai *Text* vienumi.
Description	Ievadiet vienuma aprakstu. Tiek atbalstīti lietotāja makro.
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:
Vienumu tipiem specifiskie lauki ir aprakstīti atbilstošajās lapās.

Note:
Rediģējot esošu veidnes līmeņa vienumu hosta līmenī, vairāki lauki ir tikai lasāmi. Varat izmantot saiti veidlapas galvenē, lai pārietu uz veidnes līmeni un rediģētu laukus tur, ņemot vērā, ka izmaiņas veidnes līmenī mainīs vienumu visiem hostiem, ar kuriem veidne ir saistīta.

Tags cilne ļauj definēt vienuma līmeņa tagus.



Vienuma vērtības priekšapstrāde

Cilne **Priekšapstrāde** ļauj definēt transformācijas noteikumus saņemtajām vērtībām.

Vienuma testēšana

Attention:
Lai veiktu vienuma testēšanu, pārliecinieties, ka servera un starpniekservera sistēmas laiks ir sinhronizēts. Ja servera laiks ir aizkavēts, vienuma testēšana var atgriezt kļūdas ziņojumu ”The task has been expired.” Tomēr atšķirīgi laika joslu iestatījumi serverī un starpniekserverī neietekmēs testēšanas rezultātu.

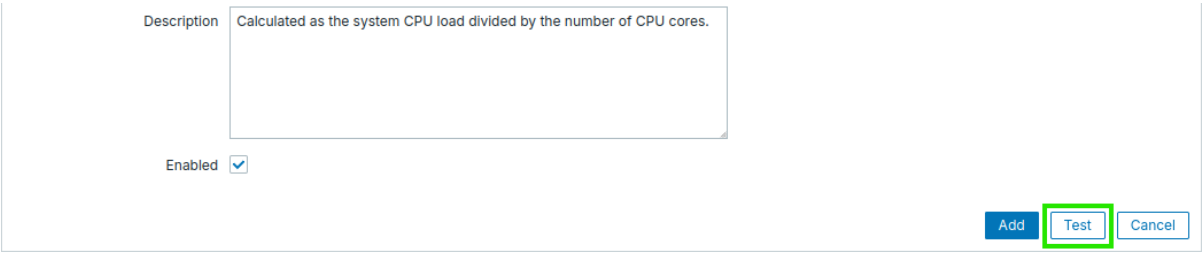
Ir iespējams testēt vienumu un, ja tas ir pareizi konfigurēts, saņemt atpakaļ reālu vērtību. Testēšanu var veikt pat pirms vienums ir saglabāts.

Testēšana ir pieejama hosts un veidnes vienumiem, vienumu prototipiem un zema līmeņa atklāšanas noteikumiem. Testēšana nav pieejama aktīvajiem vienumiem.

Vienuma testēšana ir pieejama šādiem pasīvajiem vienumu tipiem:

- Zabbix aģents
- SNMP aģents (v1, v2, v3)
- IPMI aģents
- SSH pārbaudes
- Telnet pārbaudes
- JMX aģents
- Vienkāršās pārbaudes (izņemot `icmpping*`, `vmware.*` vienumus)
- Zabbix iekšējais
- Aprēķinātie vienumi
- Ārējās pārbaudes
- Datubāzes monitors
- HTTP aģents
- Skripts
- Pārlūks

Lai testētu vienumu, vienuma konfigurācijas formas apakšā noklikšķiniet uz pogas *Test*. Ņemiet vērā, ka poga *Test* būs atspējota vienumiem, kurus nevar testēt (piemēram, aktīvajām pārbaudēm, izslēgtajām vienkāršajām pārbaudēm).

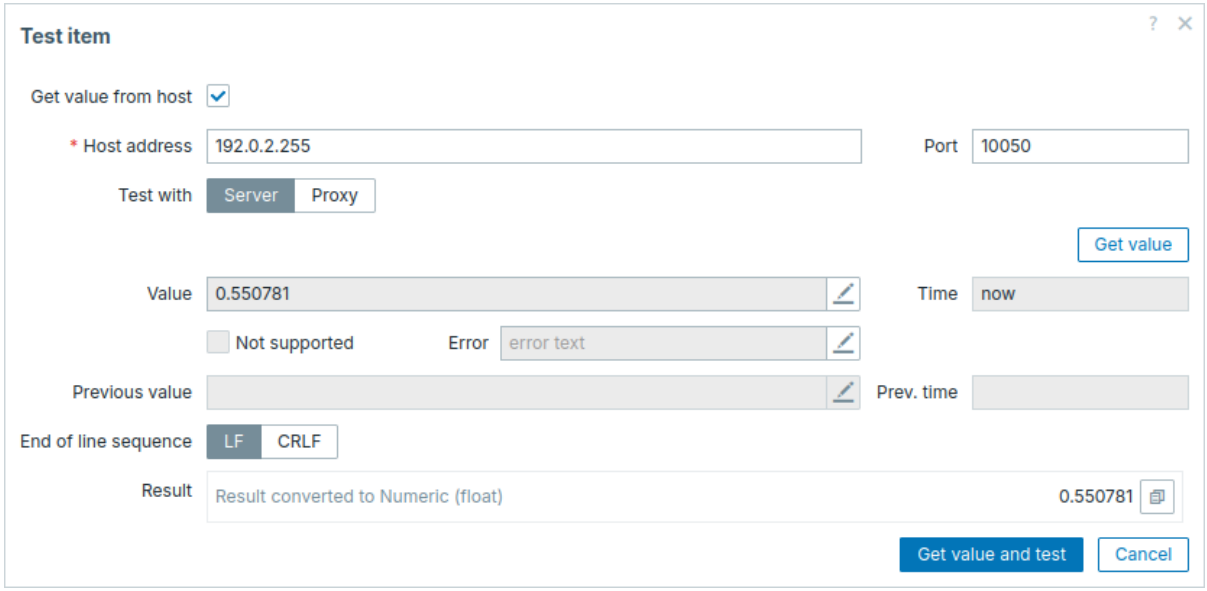


Vienuma testēšanas formai ir lauki nepieciešamajiem hosts parametriem (hosta adrese, ports, testēt ar serveri/starpniekserveri (starpniekservera nosaukums)) un vienumam specifiskām detaļām (piemēram, SNMPv2 kopienai vai SNMPv3 drošības akreditācijas datiem). Šie lauki ir konteksta ziņā atbilstoši:

- Vērtības, ja iespējams, tiek aizpildītas iepriekš, t. i., vienumiem, kuriem nepieciešams aģents, izmantojot informāciju no izvēlētās hosta aģenta saskarnes.
- Veidnes vienumiem vērtības jāaizpilda manuāli.
- Vienkārša teksta makro vērtības tiek atrisinātas.
- Lauki, kuros vērtība (vai daļa no vērtības) ir noslēpums vai Vault makro, ir tukši un jāievada manuāli. Ja kāds vienuma parametrs satur noslēpuma 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)

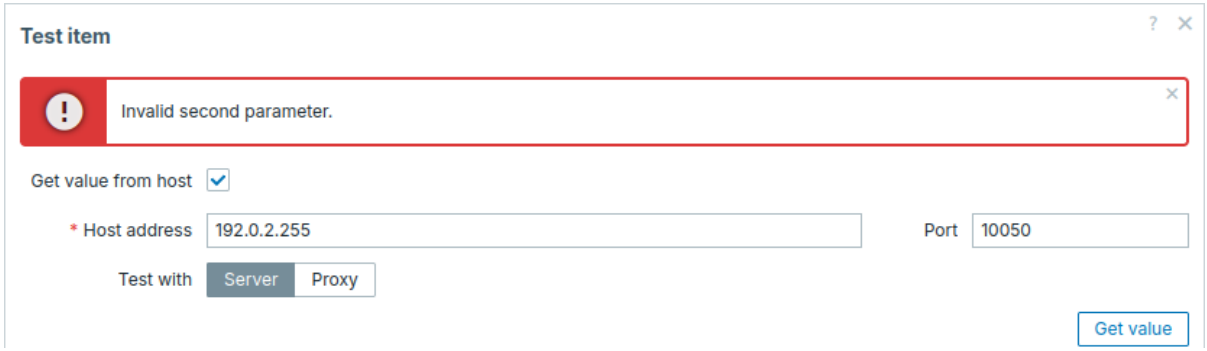
Lai testētu vienumu, noklikšķiniet *Get value*. Ja vērtība tiek veiksmīgi izgūta, tā tiks ievietota laukā *Value*, pārvietojot pašreizējo vērtību (ja tāda ir) uz lauku *Previous value*, vienlaikus aprēķinot lauku *Prev. time*, t. i., laika starpību starp abām vērtībām (klikšķiem), un mēģinot noteikt EOL secību un pārslēgties uz CRLF, ja izgūtajā vērtībā tiek noteikts "\n\r".

Noklikšķiniet *Get value and test*, lai testētu priekšapstrādi.



No hosta izgūtās vērtības un testēšanas rezultāti, nosūtot uz lietotāja saskarni, tiek saīsināti līdz maksimālajam 512KB lielumam. Ja testēšanas 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), var neizdoties datu tipa validācija; šādos gadījumos tiek parādīts kļūdas ziņojums. Ņemiet vērā, ka dati, kas ir lielāki par 512KB, Zabbix serveris joprojām apstrādā pilnībā.

Ja konfigurācija ir nepareiza, tiek parādīts kļūdas ziņojums, kurā aprakstīts iespējamais iemesls.



Veiksmīgi izgūtu vērtību no hosta 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, lai iegūtu jaunu vienuma vērtību. Atbalstītas tikai **pasīvās** pārbaudes (skatiet papildinformāciju). *Piezīme*: nekavējoties pārbaudot vērtību, konfigurācijas kešatmiņa netiek atjaunināta, tādēļ vērtība neatspoguļos pavisam nesen veiktās 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 limiti

Vienumu datu limiti 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 vesela daļa (piem., 1.23 → 1).

Skaitliskās (float) vērtības tiek saglabātas tādas, kādas tās ir, un atbalsta aptuveni 15 līdz 17 ciparu precizitāti ar diapazonu aptuveni no -1.79E+308 līdz 1.79E+308. Tiek atbalstītas arī vērtības zinātniskajā pierakstā (piem., 1.23E+7, 1e308, 1.1E-4).

Teksta vērtības pirms saglabāšanas tiek saīsinātas, lai atbilstu datubāzes vērtības tipa ierobežojumam:

Database	Type of information		
	Character	**Log**	**Text**
MySQL	255 rakstzīmes	65536 baiti	65536 baiti
PostgreSQL	255 rakstzīmes	65536 rakstzīmes	65536 rakstzīmes
SQLite (only Zabbix starpniekserveris)	255 rakstzīmes	65536 rakstzīmes	65536 rakstzīmes

Binārās vērtības ir ierobežotas līdz 16 MiB (16777216 baitiem). Vērtības, kas pārsniedz šo limitu, tiek atmestas, un lietotāja saskarnē tiek parādīta atbilstoša kļūda.

Vērtības ar JSON datu tipu ir ierobežotas līdz 128 MiB (134217728 baitiem). Vērtības, kas pārsniedz šo limitu, tiek atmestas, un lietotāja saskarnē tiek parādīta atbilstoša kļūda. MySQL/MariaDB gadījumā apsveriet iespēju iestatīt dienas nodalījumus tabulai `history_json` (kas tiek izmantota JSON vērtību glabāšanai), jo tā var ātri kļūt liela un uz ilgu laiku bloķēt housekeeper. Ja jūsu lietošanas scenārijā ir lielas JSON vērtības (1 MiB vai vairāk), skatiet, kā konfigurēt sistēmu lielu JSON vērtību atbalstam.

Kad Zabbix raksta jebkādus vienuma datus datubāzē, tas izmanto INSERT vaicājumus un tos reģistrē (ja `DebugLevel` ir iestatīts uz 4 vai 5). Lieli žurnāla ieraksti tiek saīsināti līdz 64 KB.

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 pārbaude tiek pārtraukta kā neveiksmīga.

Kad noildze tiek sasniegta, pārbaude tiek pārtraukta pat tad, ja datu izgūšana vēl nav pabeigta. Ja dati tiek saņemti daļēji, vienums kļūst neatbalstīts, un tiek reģistrēts kļūdas ziņojums (piemēram, ja SNMP pārbaudē veiksmīgi tiek savākti dati tikai vienam no vairākiem OID).

Daudziem vienumu tipiem var iestatīt **elastīgas** vienumu noildzes:

- vienam atsevišķam vienumam
- vienumu tipam (starpniekservera līmenī)
- vienumu tipam (globālā līmenī)

Pielāgota noildze *vienam atsevišķam vienumam* ir noderīga, ja vēlaties norādīt garāku noildzi konkrētam vienumam, vienlaikus saglabājot īsas noildzes citiem vienumiem.

Noildzes prioritāte

1. Viena atsevišķa vienuma noildze pārraksta jebkuras citas noildzes.
2. Starpniekservera līmeņa noildzes pārraksta globālās noildzes.

Zabbix serveris vai starpniekserveris konfigurācijā norādītajai noildzei **nav** nozīmes pārbaudēs, kurām ir iestatītas elastīgas noildzes.

Elastīga taimauta atbalsts

Elastīgi vienuma taimauti tiek atbalstīti šādiem vienumu tipiem:

- Zabbix aģents (gan pasīvajām, gan aktīvajām pārbaudēm)
- SNMP aģents (izņemot mantotās SNMP pārbaudes[1])
- Vienkārša 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 aptauja) tiek piemēroti taimauta iestatījumi no serveris vai starpniekserveris.
 [2] `icmpping*` vienumiem taimauta vērtība tiek norādīta tieši vienuma atslēgā; turklāt tiek piemērots arī fiksēts maksimālais iespējamais taimauts 600 sekundes. VMware uzraudzības vienumi izmanto savu VMwareTimeout parametru no serveris vai starpniekserveris konfigurācijas.

Neatbalstīti vienumi

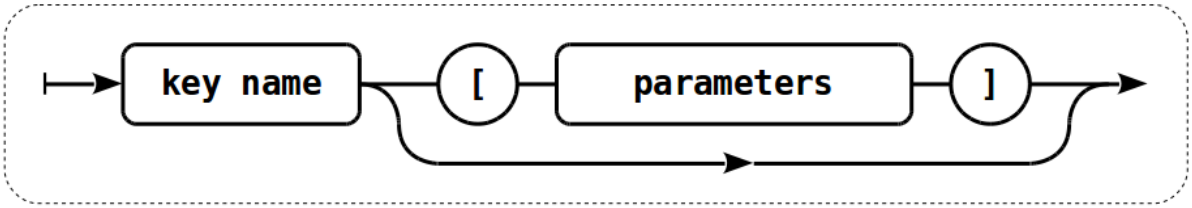
Par neatbalstītiem vienumiem tiek ziņots ar statusu *Not supported*, un tie joprojām tiek atkārtoti pārbaudīti to standarta *Update interval* laikā.

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).

Vienuma atslēgas formāts

Vienuma atslēgas formātam, tostarp atslēgas parametriem, ir jāatbilst sintakses noteikumiem. Tālāk redzamie attēli parāda atbalstīto sintaksi. Atļautos elementus un rakstzīmes katrā punktā var noteikt, sekojot bultiņām - ja kādu bloku var sasniegt pa līniju, tas ir atļauts, ja nevar - tas nav atļauts.



Lai izveidotu derīgu vienuma atslēgu, vispirms norāda atslēgas nosaukumu, pēc tam ir izvēle - vai nu iekļaut parametrus, vai neiekļaut tos, kā to parāda divas līnijas, pa kurām var sekot.

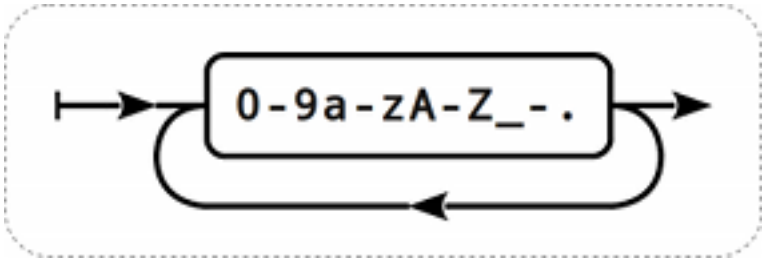
Atslēgas nosaukums

Pašam atslēgas nosaukumam ir ierobežots atļauto rakstzīmju diapazons, kuras vienkārši seko cita citai. Atļautās rakstzīmes ir:

`0-9a-zA-Z_-.`

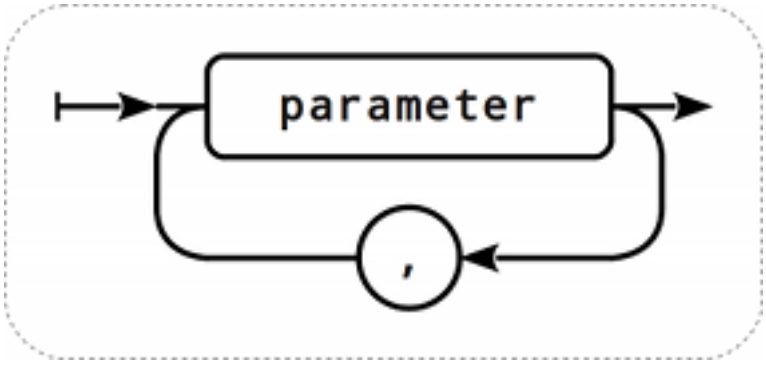
Tas nozīmē:

- visi cipari;
- visi mazie burti;
- visi lielie burti;
- pasvītrojuma zīme;
- defise;
- punkts.

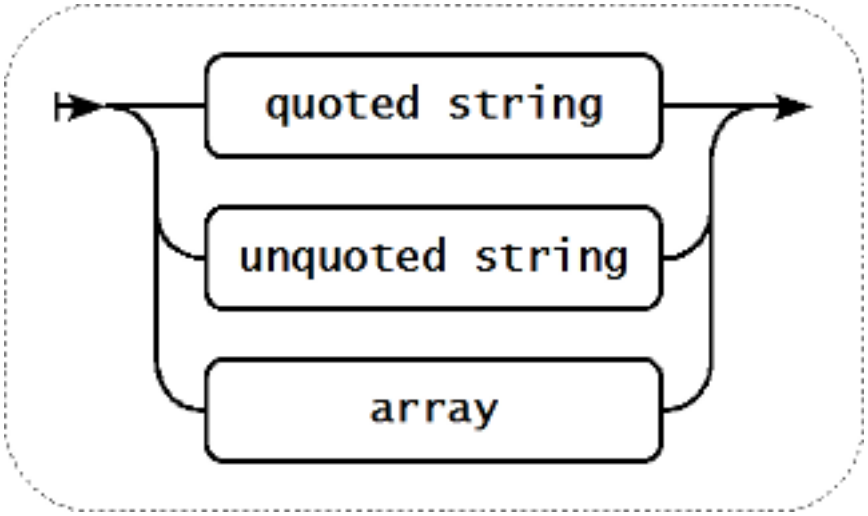


Atslēgas parametri

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.

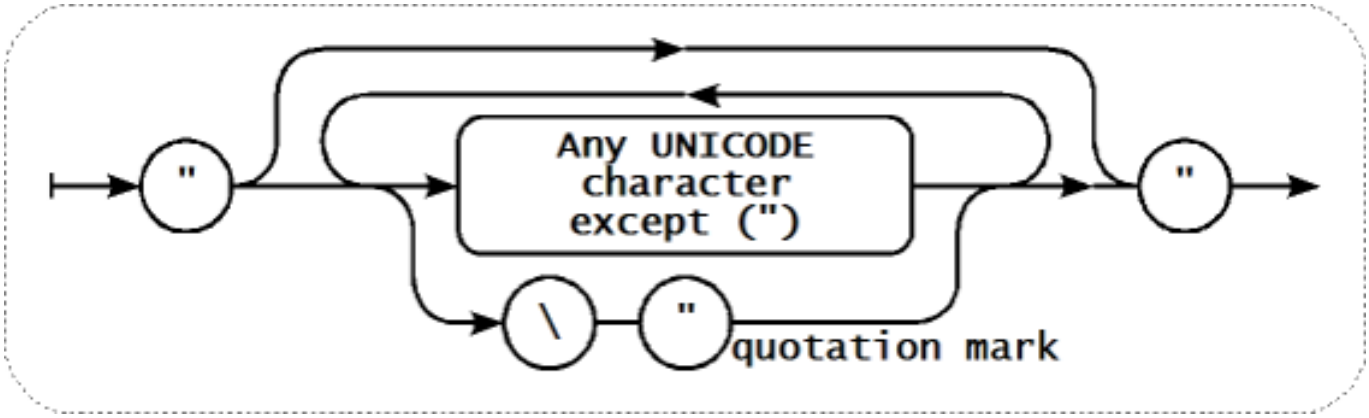


Parametru var arī atstāt tukšu, tādējādi izmantojot noklusējuma vērtību. Šādā gadījumā ir jāpievieno atbilstošs komatu skaits, ja ir norādīti vēl citi parametri. Piemēram, vienuma atslēga **icmpping[„200„500]** norāda, ka intervāls starp atsevišķiem ping pieprasījumiem ir 200 milisekundes, noildze — 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.

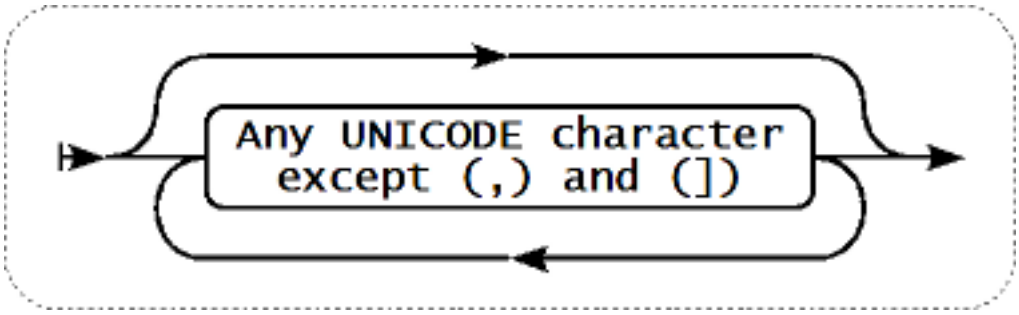


Warning:
Lai liktu vienums atslēgas parametrus pēdiņās, izmantojiet tikai dubultpēdiņas. Vienkāršās pēdiņas netiek atbalstītas.

Attention:
Daudzlīmeņu parametru masīvi, piemēram, `[a,[b,[c,d]],e]`, nav atļauti.

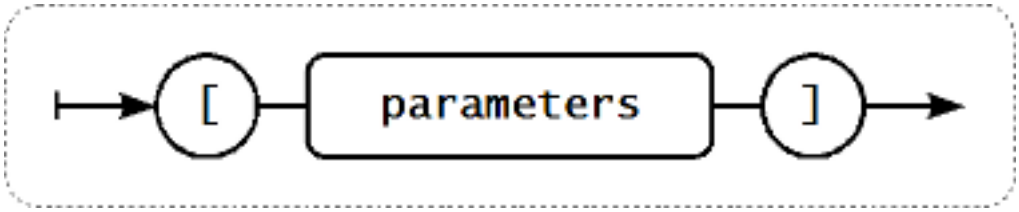
Parametrs — neievietota virkne pēdiņās

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.



Pielāgoti intervāli

Pārskats

Ir iespējams izveidot pielāgotus noteikumus attiecībā uz laiku, kad tiek pārbaudīts vienums. Tam ir pieejamas divas metodes: *elastīgie intervāli*, kas ļauj no jauna definēt noklusējuma atjaunināšanas intervālu, un *plānošana*, ar kuru vienuma pārbaudi var izpildīt noteiktā laikā vai noteiktā laiku secībā.

Elastīgie intervāli

Elastīgie intervāli ļauj atkārtoti definēt noklusējuma atjaunināšanas intervālu konkrētiem laika periodiem. Elastīgo intervālu definē ar *Interval* un *Period*, kur:

- *Interval* - atjaunināšanas intervāls norādītajam laika periodam. Taimera sufiksi tiek atbalstīti, piemēram, 30s, 1m, 2h, 1d.
- *Period* - laika periods, kurā elastīgais intervāls ir aktīvs (detalizētu *Period* formāta aprakstu skatiet sadaļā laika periodi)

Ja vairāki elastīgie intervāli pārklājas, pārklājošajam periodam tiek izmantota mazākā *Interval* vērtība. Ņemiet vērā: ja pārklājošos elastīgo intervālu mazākā vērtība ir ’0’, aptauja netiks veikta. Ārpus elastīgajiem intervāliem tiek izmantots noklusējuma atjaunināšanas intervāls.

Ņemiet vērā: ja elastīgais intervāls ir vienāds ar perioda ilgumu, vienums tiks pārbaudīts tieši vienu reizi. Ja elastīgais intervāls ir lielāks par periodu, vienums var tikt pārbaudīts vienu reizi vai arī netikt pārbaudīts vispār (tāpēc šāda konfigurācija nav ieteicama). Ja elastīgais intervāls ir mazāks par periodu, vienums tiks pārbaudīts vismaz vienu reizi.

Ja elastīgais intervāls ir iestatīts uz ’0’, elastīgā intervāla periodā vienuma aptauja netiek veikta, un pēc perioda beigām aptauja tiek atsākta atbilstoši noklusējuma *Update interval*. Piemēri:

Interval	Period	Description
10	1-5,09:00-18:00	Vienums darba laikā tiks pārbaudīts ik pēc 10 sekundēm.
0	1-7,00:00-7:00	Vienums naktī netiks pārbaudīts.
0	7-7,00:00-24:00	Vienums svētdienās netiks pārbaudīts.
60	1-7,12:00-12:01	Vienums tiks pārbaudīts katru dienu plkst. 12:00. Ņemiet vērā, ka tas tika izmantots kā risinājums ieplānotajām pārbaudēm; 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 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 ar diapazonu, vai ar *<step>* vērtību.

Tukšs filtrs atbilst vai nu '0', ja nav definēts neviens zemāka līmeņa filtrs, vai visām iespējamajām vērtībām pretējā gadījumā. Piemēram, ja stundu filtrs ir izlaists, tad atbildīs tikai stunda '0', ja arī minūšu un sekunžu filtri ir izlaisti; pretējā gadījumā tukšs stundu filtrs atbildīs visām stundu vērtībām.

Derīgās `<from>` un `<to>` vērtības attiecīgajiem filtra prefiksiem 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 vairākus plānošanas intervālus ievada 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"
```

Vienuma vērtības pirmapstrāde

Pārskats

Priekšapstrāde ļauj piemērot transformācijas saņemtajām vienuma vērtībām pirms to saglabāšanas datubāzē. Šīs transformācijas/priekšapstrādes darbības veic Zabbix serveris vai starpniekserveris (ja vienumus uzrauga starpniekserveris).

Šī funkcija 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 izgūtu vai modificētu vērtības;
- pielāgotu skriptu izpildi vērtībām;
- nemainīgu vērtību atmešanu, lai optimizētu datubāzes glabātuves izmantošanu.

Vienumam var konfigurēt vienu vai vairākas priekšapstrādes darbības. Šīs darbības tiek izpildītas to konfigurēšanas secībā.

Attention:
Ja priekšapstrādes darbība neizdodas, vienums kļūst neatbalstīts. No tā var izvairīties, izmantojot kļūdu apstrādi *Custom on fail* (pieejama lielākajai daļai transformāciju), kas ļauj atmest vērtības vai iestatīt pielāgotas vērtības.

 Žurnāla vienumiem žurnāla metadati (bez vērtības) vienmēr atiestatīs vienuma neatbalstīto stāvokli, padarot to atkal atbalstītu. Tas notiek pat tad, ja sākotnējā kļūda radās pēc žurnāla vērtības saņemšanas no aģenta.

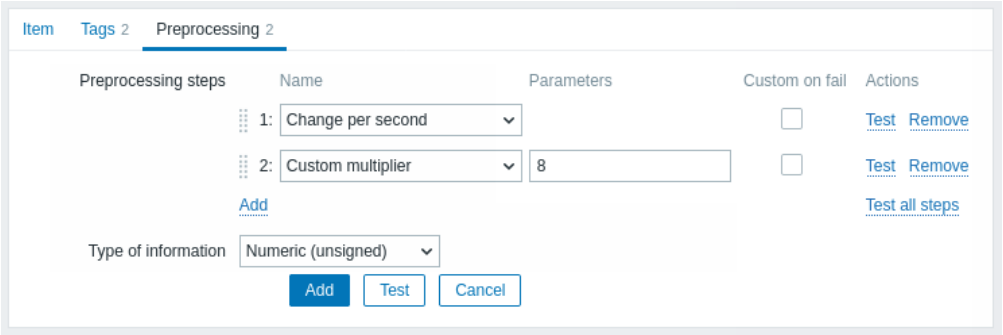
Visas vērtības, kas tiek nodotas priekšapstrādei, sākotnēji tiek apstrādātas kā virknes. Pārveidošana uz vēlamo vērtības tipu (kā definēts vienuma konfigurācijā) tiek veikta priekšapstrādes procesa beigās. Tomēr noteiktas priekšapstrādes darbības vajadzības gadījumā var izraisīt agrāku pārveidošanu. Detalizētai tehniskajai informācijai skatiet Priekšapstrādes detalizēta informācija.

Lai pārliecinātos, ka jūsu priekšapstrādes konfigurācija darbojas, kā paredzēts, varat to pārbaudīt.

Skatiet arī: Priekšapstrādes piemēri

Konfigurācija

Priekšapstrādes darbības tiek definētas vienuma konfigurācijas veidlapas cilnē **Priekšapstrāde**.



Noklikšķiniet uz *Pievienot*, lai atlasītu atbalstītu pārveidošanu.

Lauks *Informācijas tips* tiek parādīts cilnes apakšdaļā, kad ir definēta vismaz viena priekšapstrādes darbība. Ja nepieciešams, informācijas tipu var mainīt, neatverot citu cilni. Detalizētu parametru aprakstu skatiet sadaļā Vienuma izveide.

Atbalstītās transformācijas

Visas atbalstītās transformācijas ir norādītas tālāk. 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
Aizstāšana	Atrast meklējamo virkni un aizstāt to ar citu virkni (vai neko).	
Apgriešana	Noņemt norādītās rakstzīmes no vērtības sākuma un beigām.	
Apgriešana labajā pusē	Noņemt norādītās rakstzīmes no vērtības beigām.	
Apgriešana kreisajā pusē	Noņemt norādītās rakstzīmes no vērtības sākuma.	
XML XPath	Izvilkt vērtību vai fragmentu no XML datiem, izmantojot XPath funkcionalitāti.	Strukturēti dati
JSON ceļš	Izvilkt vērtību vai fragmentu no JSON datiem, izmantojot JSONPath funkcionalitāti.	
CSV uz JSON	Konvertēt CSV faila datus JSON formātā.	
XML uz JSON	Konvertēt XML formāta datus JSON formātā.	
SNMP walk vērtība	Izvilkt vērtību pēc norādītā OID/MIB nosaukuma un lietot formatēšanas opcijas.	SNMP
SNMP walk uz JSON	Konvertēt SNMP vērtības JSON formātā.	
SNMP get vērtība	Lietot formatēšanas opcijas SNMP get vērtībai.	
Pielāgots reizinātājs	Reizināt vērtību ar norādīto veselo skaitli vai peldošā komata vērtību.	Aritmētika
Vienkāršas izmaiņas	Aprēķināt starpību starp pašreizējo un iepriekšējo vērtību.	Izmaiņas
Izmaiņas sekundē	Aprēķināt vērtības izmaiņas (starpības starp pašreizējo un iepriekšējo vērtību) ātrumu sekundē.	
Būla vērtība uz decimālo	Konvertēt vērtību no būla formāta decimālformātā.	Skaitļu sistēmas
Oktālā uz decimālo	Konvertēt vērtību no oktālā formāta decimālformātā.	
Heksadecimālā uz decimālo	Konvertēt vērtību no heksadecimālā formāta decimālformātā.	
JavaScript	Ievadīt JavaScript kodu.	Pielāgoti skripti
Diapazonā	Definēt diapazonu, kurā jāatrodas vērtībai.	Validācija
Atbilst regulārajai izteiksmei	Norādīt regulāro izteiksmi, kurai vērtībai jāatbilst.	
Neatbilst regulārajai izteiksmei	Norādīt regulāro izteiksmi, kurai vērtība nedrīkst atbilst.	
Pārbaudīt kļūdu JSON datos	Pārbaudīt lietojumprogrammas līmeņa kļūdas ziņojumu, kas atrodas JSONPath norādītajā vietā.	
Pārbaudīt kļūdu XML datos	Pārbaudīt lietojumprogrammas līmeņa kļūdas ziņojumu, kas atrodas XPath norādītajā vietā.	
Pārbaudīt kļūdu, izmantojot regulāro izteiksmi	Pārbaudīt lietojumprogrammas līmeņa kļūdas ziņojumu, izmantojot regulāro izteiksmi.	
Pārbaudīt, vai vērtība nav neatbalstīta	Pārbaudīt, vai nevarēja izgūt nevienu vienuma vērtību.	
Atmest nemainīto	Atmest vērtību, ja tā nav mainījusies.	Datu plūsmas ier-obežošana

Nosaukums	Apraksts	Tips
Atmest nemainīto ar sirdspukstu	Atmest vērtību, ja tā nav mainījusies noteiktajā laika periodā.	
Prometheus modelis	Izmantot tālāk norādīto vaicājumu, lai no Prometheus metrikām izgūtu nepieciešamos datus.	Prometheus
Prometheus uz JSON	Konvertēt nepieciešamās Prometheus metrikas JSON formātā.	

Ja atlasītajai transformācijai ir iespējota izvēles rūtiņa *Pielāgota darbība kļūmes gadījumā*, varat to atzīmēt un norādīt pielāgotas kļūdu apstrādes opcijas: *Atmest vērtību* (atmest vērtību), *Iestatīt vērtību uz* (iestatīt norādīto vērtību) vai *Iestatīt kļūdu uz* (iestatīt norādīto kļūdas ziņojumu). Ja priekšapstrādes darbība neizdodas, vienums netiks atzīmēts kā neatbalstīts, ja ir atlasīta opcija *Atmest vērtību* vai *Iestatīt vērtību uz*.

Ņemiet vērā, ka *Izmaiņu* un *Datu plūsmas ierobežošanas* priekšapstrādes darbībām Zabbix ir jāatceras pēdējā vērtība, lai pēc nepieciešamības aprēķinātu vai salīdzinātu jauno vērtību. Šīs iepriekšējās vērtības apstrādā priekšapstrādes pārvaldnieks. Ja Zabbix serveris vai starpniekserveris tiek restartēts vai tiek veiktas izmaiņas priekšapstrādes darbībās, attiecīgā vienuma pēdējā vērtība tiek atiestatīta, kā rezultātā:

- *Vienkāršas izmaiņas*, *Izmaiņas sekundē* darbībām nākamā vērtība tiks ignorēta, jo nav iepriekšējās vērtības, no kuras aprēķināt izmaiņas;
- *Atmest nemainīto*, *Atmest nemainīto ar sirdspukstu* darbībām nākamā vērtība nekad netiks atmesta, pat ja saskaņā ar atmešanas noteikumiem tā būtu jāatmet.

Regulārā izteiksme

Salīdziniet vērtību ar regulāro izteiksmi un aizstājiet to ar nepieciešamo izvadi.

Parametri:

- **pattern** - regulārā izteiksme;
- **output** - izvades formatēšanas veidne. Escape secība \N (kur N=1...9) tiek aizstāta ar N-to atrasto grupu. Escape secība \0 tiek aizstāta ar atrasto tekstu.

Komentāri:

- Ja ievades vērtībai netiek atrasta atbilstība, vienums kļūs neatbalstīts;
- Regulārā izteiksme atbalsta ne vairāk kā 10 uztverto grupu izgūšanu, izmantojot secību \N;
- Dažus esošos piemērus skatiet sadaļā regulārās izteiksmes.

Aizstāt

Atrod meklējamo virkni un aizstāj to ar citu virkni (vai neko).

Parametri:

- **meklējamā virkne** - virkne, kas jāatrod un jāaizstāj; reģistrs tiek ņemts vērā (obligāts);
- **aizstājējvirkne** - virkne, ar ko aizstāt meklējamo virkni. Aizstājējvirkne var būt arī tukša, tādējādi ļaujot dzēst atrasto meklējamo virkni.

Komentāri:

- Tiks aizstātas visas meklējamās virknes parādīšanās reizes;
- Lai meklētu vai aizstātu rindu pārtraukumus, rakstatgriezes, tabulācijas un atstarpes, var izmantot atsoļa sekvences ”\n \r \t \s”; atpakaļvērstā slīpsvītra var tikt ekranēta kā ”\\” un atsoļa sekvences var tikt ekranētas kā ”\\n”;
- Rindu pārtraukumu, rakstatgriežu un tabulāciju ekranēšana zema līmeņa atklāšanas laikā tiek veikta automātiski.

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

Izvelciet vērtību vai fragmentu no XML datiem, izmantojot XPath funkcionalitāti.

Komentāri:

- Lai šī opcija darbotos, Zabbix serverim (vai Zabbix starpniekserverim) jābūt kompilētam ar libxml atbalstu;

- Nosaukumvietas netiek atbalstītas.

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></docu
```

JSON ceļš

Izgūstiet vērtību vai fragmentu no JSON datiem, izmantojot JSONPath funkcionalitāti.

CSV uz JSON

Pārveidojiet CSV faila datus JSON formātā.

Papildinformāciju skatiet: CSV uz JSON priekšapstrāde.

XML uz JSON

Konvertēt XML formāta datus uz JSON.

Papildinformāciju skatiet sadaļā Serializācijas noteikumi.

SNMP walk vērtība

Izgūt vērtību, izmantojot norādīto OID/MIB nosaukumu, un lietot formatēšanas opcijas:

- **Nemainīta** - atgriezt heksadecimālo virkni kā neekranētu heksadecimālo virkni (*ņemiet vērā*, ka attēlošanas norādes joprojām tiek lietotas);
- **UTF-8 no hex-STRING** - pārveidot heksadecimālo virkni par UTF-8 virkni;
- **MAC no hex-STRING** - validēt heksadecimālo virkni kā MAC adresi un atgriezt atbilstošu MAC adreses virkni (kur ' ' tiek aizstātas ar ':');
- **Vesels skaitlis no BITS** - pārveidot bitu virknes pirmos 8 baitus, kas izteikti kā heksadecimālo rakstzīmju secība (piemēram, "1A 2B 3C 4D"), par 64 bitu neparakstītu veselu skaitli. Bitu virknēs, kas garākas par 8 baitiem, nākamie baiti tiks ignorēti.

SNMP walk uz JSON

Konvertējiet SNMP vērtības uz JSON.

Norādiet lauka nosaukumu JSON un atbilstošo SNMP OID ceļu.
Lauku vērtības tiks aizpildītas ar vērtībām no norādītā SNMP OID ceļa.

Komentāri:

- Ir pieejamas līdzīgas vērtību formatēšanas opcijas kā darbībā *SNMP walk value*;
- Šo priekšapstrādes darbību var izmantot SNMP OID atklāšanai.

SNMP get vērtība

Lietot formatēšanas opcijas SNMP get vērtībai:

- **UTF-8 no Hex-STRING** - konvertēt heksadecimālo virkni par UTF-8 virkni;
- **MAC no Hex-STRING** - validēt heksadecimālo virkni kā MAC adresi un atgriezt atbilstošu MAC adreses virkni (kur ' ' tiek aizstātas ar ':');
- **Integer no BITS** - konvertēt bitu virknes pirmos 8 baitus, kas izteikti kā heksadecimālo rakstzīmju secība (piemēram, "1A 2B 3C 4D"), par 64 bitu unsigned integer. Bitu virknēs, kas garākas par 8 baitiem, nākamie baiti tiks ignorēti.

Pielāgots reizinātājs

Reiziniet vērtību ar norādīto veselo skaitli vai peldošā komata vērtību.

Komentāri:

- Izmantojiet šo opciju, lai konvertētu vērtības, kas saņemtas KB, MBps u. c., uz B, Bps. Pretējā gadījumā Zabbix nevar pareizi iestatīt prefiksus (K, M, G u. c.).
- Ņemiet vērā: ja vienuma informācijas tips ir *Numeric (unsigned)*, ienākošās vērtības ar daļskaitli tiks noapaļotas, noņemot daļskaitli (t. i., "0.9" kļūs par "0"), pirms tiek lietots pielāgotais reizinātājs;
- Ja izmantojat pielāgotu reizinātāju vai vienumiem, kuru informācijas tips ir *Numeric (unsigned)*, saglabājat vērtību kā *Change per second* un iegūtā aprēķinātā vērtība faktiski ir peldošā komata skaitlis, aprēķinātā vērtība joprojām tiek pieņemta kā pareiza, 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 jāatrisinās par veselu skaitli vai peldošā komata skaitli.

Vienkāršas izmaiņas

Aprēķina starpību starp pašreizējo un iepriekšējo vērtību.

Komentāri:

- Šis solis var būt noderīgs, lai mērītu nepārtraukti augošu vērtību;
- Tiek aprēķināts kā **value**-**prev_value**, kur *value* ir pašreizējā vērtība; *prev_value* ir iepriekš saņemtā vērtība;
- Katram vienumam ir atļauta tikai viena izmaiņu darbība (”Vienkāršas izmaiņas” vai ”Izmaiņas sekundē”);
- Ja pašreizējā vērtība ir mazāka par iepriekšējo vērtību, Zabbix šo starpību noraida (neko nesaglabā) un gaida nākamo vērtību.

Izmaiņa sekundē

Aprēķina vērtības izmaiņas (starpības starp pašreizējo un iepriekšējo vērtību) ātrumu sekundē.

Komentāri:

- Šis solis ir noderīgs, lai aprēķinātu nepārtraukti augošas vērtības ātrumu sekundē;
- Tā kā šis aprēķins var radīt peldošā komata skaitļus, ieteicams iestatīt “Informācijas tips” uz *Numeric (float)*, pat ja saņemtās neapstrādātās vērtības ir veseli skaitļi. Tas ir īpaši svarīgi maziem skaitļiem, kuriem nozīme ir decimāldaļai. Ja peldošā komata vērtības ir lielas un var pārsniegt lauka “float” garumu, kā rezultātā visa vērtība var tikt zaudēta, ieteicams izmantot *Numeric (unsigned)* un tādējādi noņemt tikai decimāldaļu;
- Tiek aprēķināts 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 zīmoga vērtība;
- Vienam vienumam ir atļauta tikai viena izmaiņas darbība (“Simple change” vai “Change per second”);
- Ja pašreizējā vērtība ir mazāka par iepriekšējo vērtību, Zabbix šo starpību atmet (neko nesaglabā) un gaida nākamo vērtību. Tas palīdz pareizi apstrādāt, piemēram, 32 bitu SNMP skaitītāju pāreju (pārpildi).

Būla vērtība uz decimālu

Konvertē vērtību no Būla formāta uz decimālo formātu.

Komentāri:

- Tekstuālais attēlojums tiek pārveidots par 0 vai 1. Tādējādi “TRUE” tiek saglabāts kā 1, bet “FALSE” tiek saglabāts kā 0. Visas vērtības tiek salīdzinātas, neņemot vērā reģistru. Pašlaik atpazītās vērtības ir šādas: *TRUE* gadījumā — true, t, yes, y, on, up, running, enabled, available, ok, master; *FALSE* gadījumā — false, f, no, n, off, down, unused, disabled, unavailable, err, slave. Turklāt jebkura vērtība, kas nav nulle, tiek uzskatīta par TRUE, bet nulle — par FALSE.

No oktālās sistēmas uz decimālo sistēmu

Konvertēt vērtību no oktālās sistēmas uz decimālo sistēmu.

No heksadecimālās uz decimālo

Konvertē vērtību no heksadecimālā formāta uz decimālo.

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 jāiekļaujas vērtībai, norādot minimālo/maksimālo vērtību (ieskaitot robežvērtības).

Komentāri:

- Tiek pieņemtas skaitliskas vērtības (tostarp ar jebkādu ciparu skaitu, neobligātu decimāldaļu un neobligātu eksponentes daļu, kā arī negatīvas vērtības);
- Minimālajai vērtībai jābūt mazākai par maksimālo;
- Jābūt norādītai vismaz vienai vērtībai;
- Var izmantot lietotāja makro un zema līmeņa atklāšanas makro.

Atbilst regulārajai izteiksmei

Norādiet regulāro izteiksmi, kurai vērtībai ir jāatbilst.

Neatbilst regulārajai izteiksmei

Norādiet regulāro izteiksmi, kurai vērtība nedrīkst atbilst.

Pārbaudīt, vai JSON ir kļūda

Pārbaudīt lietojumprogrammas līmeņa kļūdas ziņojumu, kas atrodas JSONPath. Ja pārbaude ir veiksmīga un ziņojums nav tukšs, pārtraukt apstrādi; pretējā gadījumā turpināt apstrādi ar vērtību, kas bija pirms šīs priekšapstrādes darbības.

Komentāri:

- Šīs ārējā servisa kļūdas lietotājam tiek parādītas tādas, kādas tās ir, nepievienojot informāciju par priekšapstrādes darbību;
- Ja neizdodas parsēt nederīgu JSON, kļūda netiks ziņota.

Pārbaudīt, vai XML nav kļūdu

Pārbaudīt lietojumprogrammas līmeņa kļūdas ziņojumu, kas atrodas XPath ceļā.
Pārtraukt apstrādi, ja pārbaude ir veiksmīga un ziņojums nav tukšs; pretējā gadījumā turpināt apstrādi ar vērtību, kas bija pirms šīs priekšapstrādes darbības.

Komentāri:

- Šīs ārējā servisa kļūdas lietotājam tiek parādītas tādas, kādas tās ir, nepievienojot informāciju par priekšapstrādes darbību;
- Ja neizdodas parsēt nederīgu XML, kļūda netiek ziņota.

Pārbaudīt kļūdu, izmantojot regulāro izteiksmi

Pārbaudīt lietojumprogrammas līmeņa kļūdas ziņojumu, izmantojot regulāro izteiksmi.
Pārtrauciet apstrādi, ja pārbaude ir veiksmīga un ziņojums nav tukšs; pretējā gadījumā turpiniet apstrādi ar vērtību, kas bija pirms šīs priekšapstrādes darbības.

Parametri:

- **pattern** - regulārā izteiksme;
- **output** - izvades formatēšanas veidne.
  Escape secība \N (kur N=1...9) tiek aizstāta ar N-to atrasto grupu.
  Escape secība \0 tiek aizstāta ar atrasto tekstu.

Komentāri:

- Šīs ārējā servisa kļūdas lietotājam tiek parādītas tādas, kādas tās ir, nepievienojot informāciju par priekšapstrādes darbību.

Pārbaudīt, vai vērtība nav atbalstīta

Pārbaudiet, vai nav izdevies izgūt vienuma vērtību. Norādiet, kā jāapstrādā kļūme, pamatojoties uz atgrieztā kļūdas ziņojuma pārbaudi.

Parametri:

- **scope** - atlasiet kļūdas apstrādes tvērumu:
any error - jebkura kļūda;
error matches - tikai kļūda, kas atbilst laukā *pattern* norādītajai regulārajai izteiksmei;
error does not match - tikai kļūda, kas neatbilst laukā *pattern* norādītajai regulārajai izteiksmei
- **pattern** - regulārā izteiksme, kurai jāsakrīt ar kļūdu. Ja tvēruma parametrā ir atlasīts *any error*, šis lauks netiek parādīts. Ja lauks tiek parādīts, tas ir obligāts.

Komentāri:

- Parasti nespēja izgūt vērtību vai tās neesamība izraisītu vienuma statusa maiņu uz neatbalstītu. Šis priekšapstrādes solis ļauj mainīt šo darbību.
- Šis priekšapstrādes solis pārbauda tikai to, vai nav izdevies izgūt vienuma vērtību. Tas nepārbauda, piemēram, vai izgūtās vērtības tips (piemēram, virkne) atbilst vienuma informācijas tipam (piemēram, skaitliskam); detalizētu informāciju skatiet sadaļā Priekšapstrādes piemēri. Ja tipi nesakrīt, pēc visu priekšapstrādes darbību izpildes vienuma statuss joprojām var tikt mainīts uz neatbalstītu. Lai pārbaudītu tipu neatbilstību, varat izmantot, piemēram, priekšapstrādes soli *Custom multiplier*; skatiet Priekšapstrādes piemērus.
- Regulārās izteiksmes grupu tveršana tiek atbalstīta laukos *Set value to* vai *Set error to*. Izmantojiet \N (kur N=1...9), lai izgūtu N-to atbilstošo grupu; izmantojiet \0, lai izgūtu atbilstošo tekstu;
- Šie soļi vienmēr tiek izpildīti kā pirmie priekšapstrādes soļi un pēc izmaiņu saglabāšanas vienumā tiek novietoti virs visiem pārējiem soļiem;
- Tiek atbalstīti vairāki *Check for not supported value* soļi norādītajā secībā. Solis *any error* tiks automātiski novietots kā pēdējais solis 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 nemainīto ar sirdspukstu

Atmest vērtību, ja tā noteiktajā laika periodā (sekundēs) nav mainījusies.

Komentāri:

- Sekunžu norādīšanai tiek atbalstītas pozitīvas veselas vērtības (minimums - 1 sekunde);
- Var izmantot laika sufiksus (piemēram, 30s, 1m, 2h, 1d);
- Var izmantot lietotāja makro un zema līmeņa atklāšanas makro;
- Ja vērtība tiek atmesta, tā netiek saglabāta datubāzē, un Zabbix serveris nezina, ka šī vērtība ir saņemta. Trigeru izteiksmes netiks novērtētas, tāpēc saistītajiem trigeriem netiks izveidotas vai atrisinātas problēmas. Funkcijas darbosies tikai, pamatojoties uz datiem, kas faktiski saglabāti datubāzē. Tā kā tendenču dati tiek veidoti, pamatojoties uz datiem datubāzē, ja stundu nav saglabāta neviena vērtība, arī par šo stundu tendenču datu nebūs.
- Katram vienumam var norādīt tikai vienu ierobežošanas opciju.

Prometheus pattern

Izmantojiet šo vaicājumu, lai iegūtu nepieciešamos datus no Prometheus metrikām.

Skatiet Prometheus pārbaudes, lai iegūtu vairāk informācijas.

Prometheus uz JSON

Pārveido nepieciešamos Prometheus metriku datus JSON formātā.

Skatiet Prometheus pārbaudes, lai iegūtu vairāk informācijas.

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.

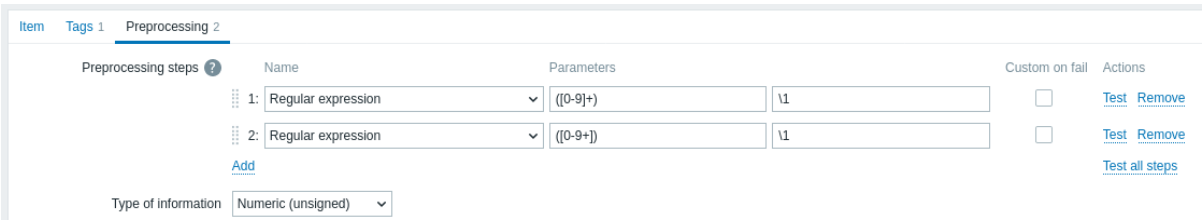
Testēšana

Skatiet priekšapstrādes testēšanu.

Priekšapstrādes testēšana

Testēšana

Pārbaudes priekšapstrādes soļiem ir noderīgas, lai pārliecinātos, ka sarežģītas priekšapstrādes ķēdes sniedz no tām sagaidītos rezultātus, negaidot, līdz vienuma vērtība tiks saņemta un priekšapstrādāta.



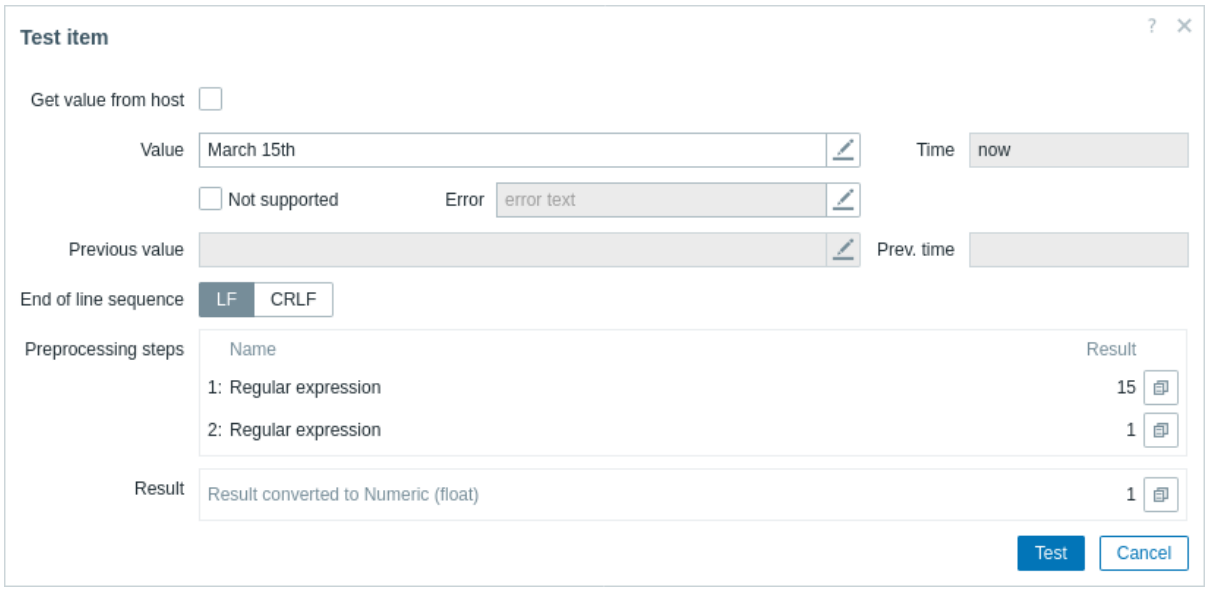
Ir iespējams testēt:

- pret hipotētisku vērtību

- pret reālu vērtību no hosta

Katru priekšapstrādes soli var testēt atsevišķi, kā arī visus soļus var testēt kopā. Kad sadaļā Actions noklikšķināt uz pogas *Test* vai *Test all steps*, tiek atvērts testēšanas logs.

Hipotētiskas vērtības testēšana



Parameter	Description
Get value from host	Ja vēlaties testēt hipotētisku vērtību, neatzīmējiet šo izvēles rūtiņu. Skatiet arī: Reālas vērtības testēšana.
Value	Ievadiet testēšanai paredzēto ievades vērtību. Noklikšķinot parametra laukā vai uz skatīšanas/rediģēšanas pogas, tiks atvērts teksta apgabala logs vērtības vai koda bloka ievadīšanai.
Not supported	Atzīmējiet šo izvēles rūtiņu, lai testētu neatbalstītu vērtību. Šī opcija ir noderīga, lai testētu priekšapstrādes soli *Check for not supported value*.
Error	Ievadiet kļūdas tekstu. Šis lauks ir iespējots, ja *Get value from host* nav atzīmēts, bet *Not supported* ir atzīmēts. Ja *Get value from host* ir atzīmēts, šis lauks tiek aizpildīts ar faktisko kļūdas ziņojumu (tikai lasāms) no hosta.
Time	Tiek parādīts ievades vērtības laiks: `now` (tikai lasāms).
Previous value	Ievadiet iepriekšējo ievades vērtību salīdzināšanai. Tikai priekšapstrādes soļiem *Change* un *Throttling*.
Previous time	Ievadiet iepriekšējās ievades vērtības laiku salīdzināšanai. Tikai priekšapstrādes soļiem *Change* un *Throttling*. Noklusējuma vērtība ir balstīta uz vienuma lauka 'Update interval' vērtību (ja tā ir '1m', tad šis lauks tiek aizpildīts ar `now-1m`). Ja nekas nav norādīts vai lietotājam nav piekļuves hostam, noklusējuma vērtība ir `now-30s`.
Macros	Ja tiek izmantoti kādi makrosi, tie tiek uzskaitīti kopā ar to vērtībām. Vērtības testēšanas nolūkos ir rediģējamas, taču izmaiņas tiks saglabātas tikai testēšanas kontekstā.
End of line sequence	Izvēlieties rindas beigu secību daudzrindu ievades vērtībām: **LF** - LF (line feed) secība **CRLF** - CRLF (carriage-return line-feed) secība.
Preprocessing steps	Tiek uzskaitīti priekšapstrādes soļi; testēšanas rezultāts tiek parādīts katram solim pēc tam, kad ir noklikšķināts uz pogas *Test*. Testēšanas rezultāti, nosūtot uz lietotāja saskarne, tiek saīsināti līdz maksimālajam 512 KB lielumam. Testēšanas rezultātus var kopēt (ne vairāk kā saīsinātos 512 KB). Ja rezultāts ir saīsināts, tiek parādīta brīdinājuma ikona. Brīdinājuma apraksts tiek parādīts, uzvedot peli virs tā. Ņemiet vērā, ka datus, kas ir lielāki par 512 KB, Zabbix serveris joprojām apstrādā pilnībā. Ja solis testēšanā neizdevās, tiek parādīta kļūdas ikona. Kļūdas apraksts tiek parādīts, uzvedot peli virs tā. Ja solim ir norādīts "Custom on fail" un šī darbība tiek izpildīta, uzreiz aiz priekšapstrādes testa soļa rindas parādās jauna rinda, kurā redzams, kāda darbība tika veikta un kādu rezultātu tā radīja (kļūdu vai vērtību).

Parameter	Description
Result	Galīgais priekšapstrādes soļu testēšanas rezultāts tiek parādīts visos gadījumos, kad visi soļi tiek testēti kopā (kad noklikšķināt uz pogas *Test all steps*). Tiek parādīts arī konvertēšanas veids uz vienuma vērtības tipu, piemēram, `Result converted to Numeric (unsigned)`. Testēšanas rezultāti, nosūtot uz lietotāja saskarne, tiek saīsināti līdz maksimālajam 512 KB lielumam. Galīgo rezultātu var kopēt (ne vairāk kā saīsinātos 512 KB). Ja rezultāts ir saīsināts, tiek parādīta brīdinājuma ikona. Brīdinājuma apraksts tiek parādīts, uzvedot peli virs tā. Ņemiet vērā, ka datus, kas ir lielāki par 512 KB, Zabbix serveris joprojām apstrādā pilnībā.

Noklikšķiniet uz *Test*, lai redzētu rezultātu pēc katra priekšapstrādes soļa.

Testēšanas 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, atkārtoti neievadot informāciju. Tomēr pēc lapas atsvaidzināšanas vērtības tiek zaudētas.

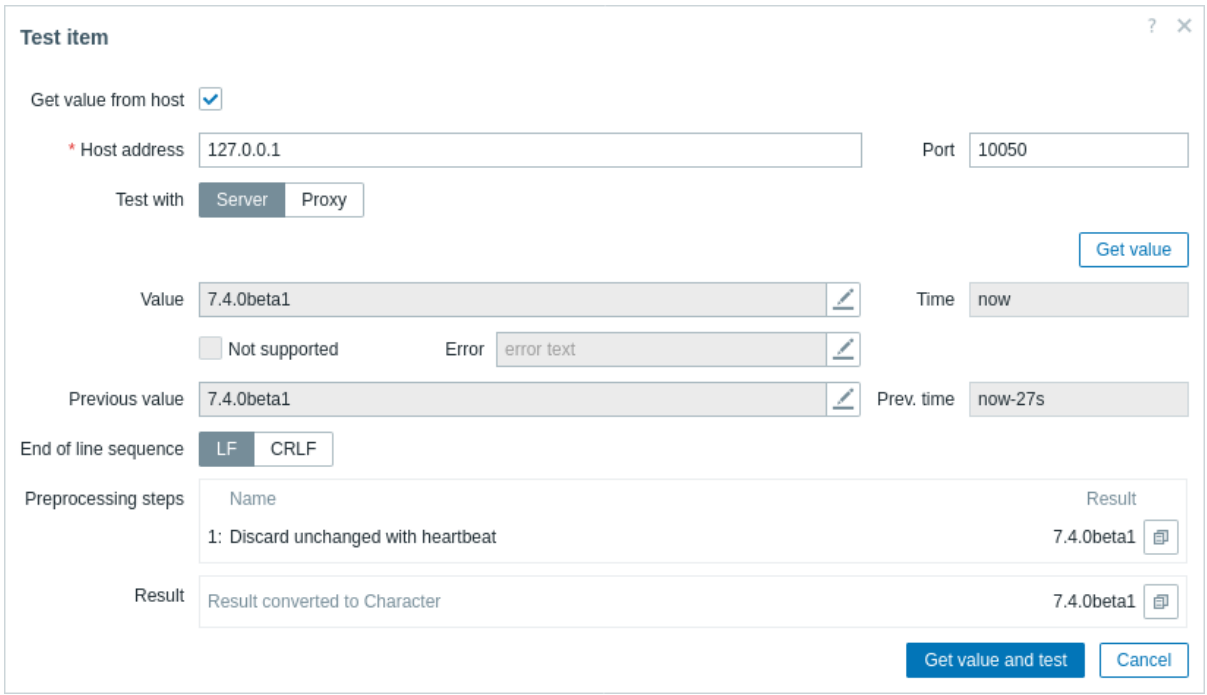
Testēšanu veic Zabbix serveris. Lietotāja saskarne nosūta atbilstošu pieprasījumu serverim un gaida rezultātu. Pieprasījums satur ievades vērtību un priekšapstrādes soļus (ar izvērstiem lietotāja makro). Soļiem *Change* un *Throttling* var norādīt izvēles iepriekšējo vērtību un laiku. Serveris atbild ar rezultātiem katram priekšapstrādes solim.

Visas tehniskās kļūdas vai ievades validācijas kļūdas tiek parādītas kļūdu lodziņā testēšanas loga augšdaļā.

Reālas vērtības testēšana

Lai pārbaudītu priekšapstrādi, izmantojot reālu vērtību:

- Atzīmējiet izvēles rūtiņu *Iegūt vērtību no hosts*
- Ievadiet vai pārbaudiet hosts parametrus (hosts adresi, portu, starpniekservera nosaukumu vai norādi, ka starpniekserveris netiek izmantots) un vienumam specifisko informāciju (piemēram, SNMPv2 kopienu vai SNMPv3 drošības akreditācijas datus). Šie lauki ir kontekstatkarīgi:
  - Ja iespējams, vērtības tiek aizpildītas automātiski, piemēram, vienumiem, kuriem nepieciešams aģents, izmantojot informāciju no izvēlētās hosts aģenta saskarnes
  - Veidnes vienumiem vērtības jāaizpilda manuāli
  - Vienkārša teksta makro vērtības tiek atrisinātas
  - Ja lauka vērtība (vai tās daļa) ir slepena vērtība vai Vault makro, lauks būs tukšs un tas jāaizpilda manuāli. Ja kāds vienuma parametrs satur slepenas vērtības makro, 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, hosts adreses un starpniekservera lauki ir atspējoti aprēķinātiem vienumiem)
- Noklikšķiniet uz *Iegūt vērtību un testēt*, lai pārbaudītu priekšapstrādi



Ja vienuma konfigurācijas veidlapā esat norādījis vērtību kartējumu (lauks *Rādīt vērtību*), vienuma testa dialoglodziņā pēc gala rezultāta tiks parādīta vēl viena rinda ar nosaukumu *Rezultāts ar lietotu vērtību kartējumu*.

Parametri, kas attiecas uz reālas vērtības iegūšanu no hosts:

Parametrs	Apraksts
Iegūt vērtību no hosts	Atzīmējiet šo izvēles rūtiņu, lai iegūtu reālu vērtību no hosts.
Hosts adrese	Ievadiet hosts adresi. Šis lauks tiek automātiski aizpildīts ar vienuma hosts saskarnes adresi.
Ports	Ievadiet hosts portu. Šis lauks tiek automātiski aizpildīts ar vienuma hosts saskarnes portu.
Papildu lauki SNMP saskarnēm (SNMP versija, SNMP kopiena, konteksta nosaukums u. c.)	Papildinformāciju par SNMP saskarnes (v1, v2 un v3) konfigurēšanu skatiet sadaļā SNMP uzraudzības konfigurēšana. Šie lauki tiek automātiski aizpildīti no vienuma hosts saskarnes.
Starpniekserveris	Norādiet starpniekserveri, ja hosts tiek uzraudzīts, izmantojot starpniekserveri. Šis lauks tiek automātiski aizpildīts ar hosts starpniekserveri (ja tāds ir).
Vērtība	No hosts iegūtā vērtība. Noklikšķinot parametra laukā vai uz skatīšanas/rediģēšanas pogas, tiks atvērts vērtības vai koda bloka teksta apgabala logs. Vērtības tiek saīsinātas līdz maksimālajam 512 KB izmēram tikai lietotāja saskarnē. Ja rezultāts ir saīsināts, tiek parādīta brīdinājuma ikona. Brīdinājuma apraksts tiek parādīts, novietojot peles rādītāju virs ikonas. Ņemiet vērā, ka datus, kas ir lielāki par 512 KB, Zabbix serveris joprojām apstrādā pilnībā.

Pārējos parametrus skatiet iepriekš sadaļā Hipotētiskas vērtības testēšana.

Priekšapstrādes detaļas

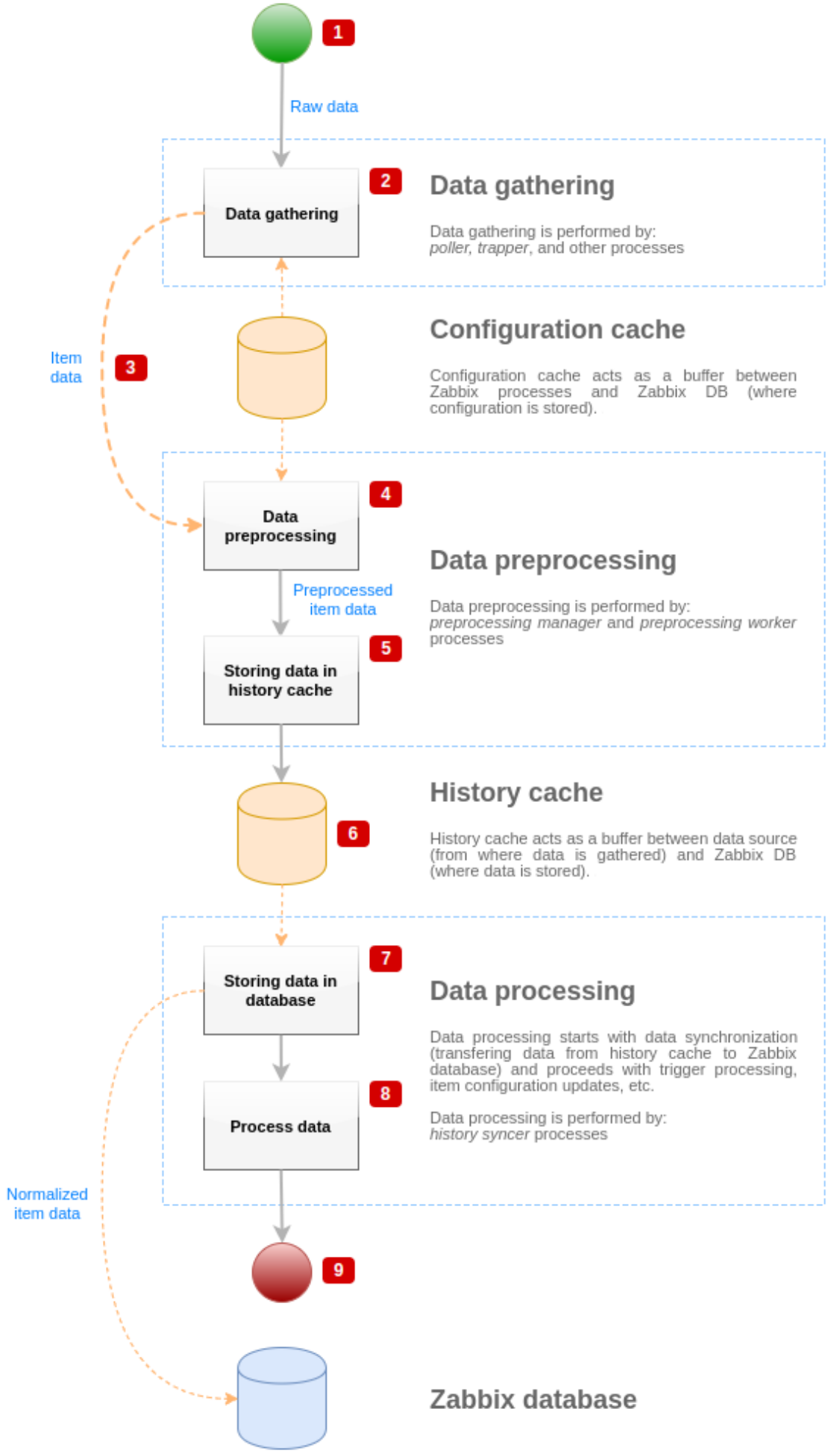
Pārskats

Šajā sadaļā ir sniegta informācija par vienumu vērtību priekšapstrādi. Vienumu vērtību priekšapstrāde ļauj definēt un izpildīt transformācijas noteikumus saņemtajām vienumu vērtībām.

Priekšapstrādi pārvalda priekšapstrādes pārvaldnieka process kopā ar priekšapstrādes darba procesiem, kas izpilda priekšapstrādes darbības. Visas vērtības, kurām tiek veikta priekšapstrāde un kuras saņemtas no dažādiem datu savācējiem, pirms pievienošanas vēstures kešatmiņai iziet cauri priekšapstrādes pārvaldniekam. Starp datu savācējiem (aptaujātājiem, uztvērējiem u. c.) un priekšapstrādes procesu tiek izmantota uz ligzdām balstīta IPC saziņa. Priekšapstrādes darbības 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
6
History cache
History cache acts as a buffer between data source (from where data is gathered) and Zabbix DB (where data is stored).
Storing data in database
7
Data processing
Data processing starts with data synchronization (transfering data from history cache to Zabbix database) and proceeds with trigger processing, item configuration updates, etc.
Data processing is performed by:
history syncer processes
Process data
8
Normalized item data
9
Zabbix database

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.

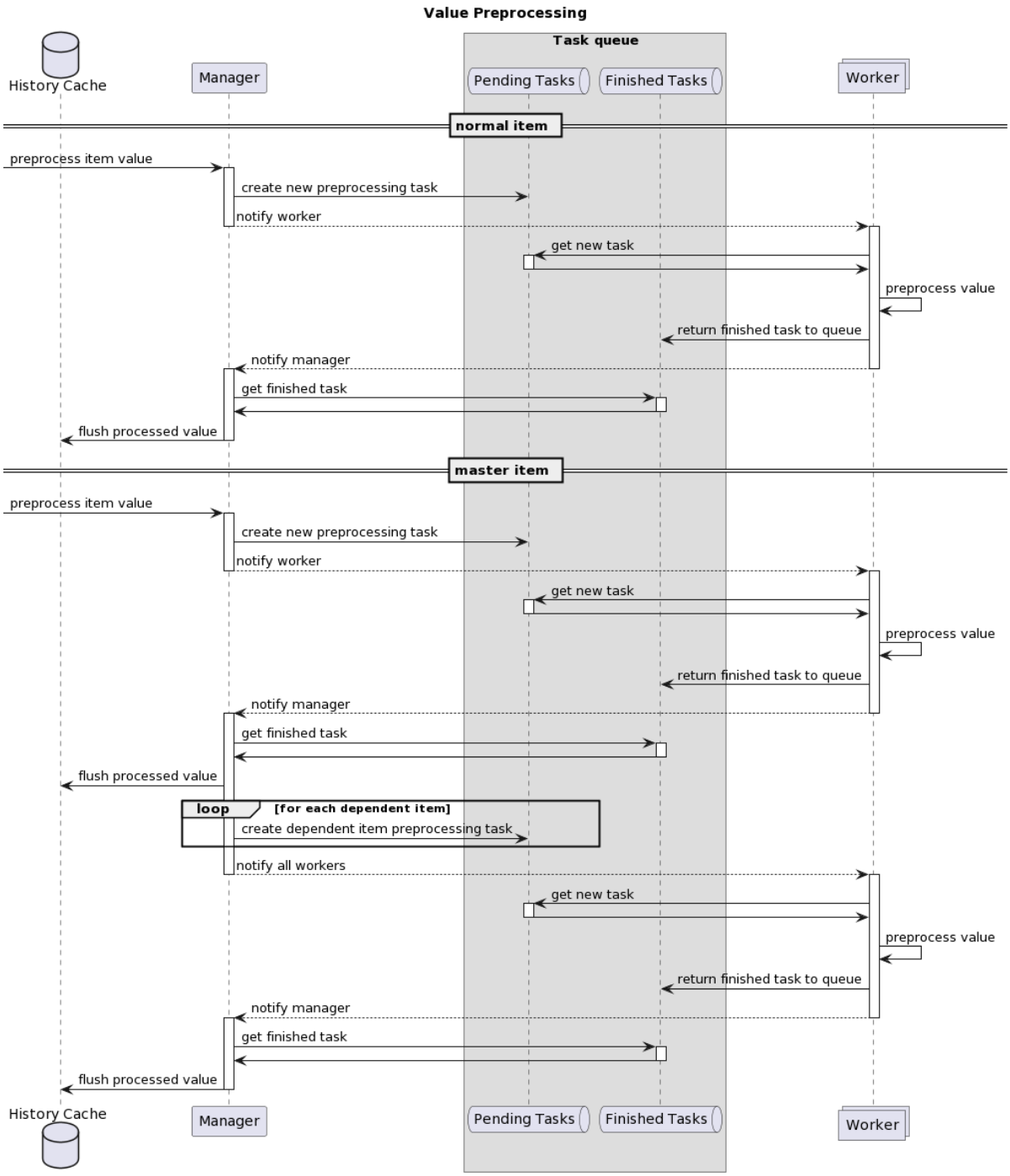
Note:
Vienums var mainīt savu stāvokli uz NOT SUPPORTED, ja datu normalizēšana neizdodas (piemēram, ja teksta vērtību nevar pārveidot par skaitli).

- 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 vajadzētu vismaz atbilst pieejamo CPU kodolu skaitam.

Ja priekšapstrādes uzdevumi nav CPU intensīvi 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 vākšanas procesu skaits, vidējais soļu skaits vienuma priekšapstrādei utt. Nepietiekams darbinieku skaits var izraisīt augstu atmiņas patēriņu. Lai novērstu pārmērīgu atmiņas patēriņu jūsu Zabbix instalācijā, skatiet Pārmērīga atmiņas patēriņa profilēšana ar tcmalloc.

Tomēr, pieņemot, ka nav smagu priekšapstrādes operāciju, piemēram, lielu XML/JSON fragmentu parsēšanas, priekšapstrādes darbinieku skaits var atbilst kopējam datu vācēju skaitam. Tādā veidā pārsvarā (izņemot gadījumus, kad dati no vācēja tiek saņemti lielā apjomā) kolektētajiem datiem būs vismaz viens brīvs priekšapstrādes darbinieks.

Warning:
Pārāk daudz datu vākšanas procesu (polleri, nepieejamības polleri, ODBC polleri, HTTP polleri, Java polleri, pingeri, traperi, starpniekservera polleri) kopā ar IPMI pārvaldnieku, SNMP trapperi un priekšapstrādes darbiniekiem var izsmelt priekšapstrādes pārvaldniekam noteikto failu deskriptoru limitu uz procesu.

 Failu deskriptoru limita izsmelšana uz procesu izraisīs Zabbix servera apstāšanos, parasti neilgi pēc palaišanas, taču dažkārt tas var aizņemt ilgāku laiku. 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.

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.

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, kas atbalsta zinātnisko pierakstu. Piemērs: 123
`<jsonpath starting with $>`	Vērtība, uz kuru norāda JSONPath no ievades dokumenta saknes mezgla; tiek atbalstīti tikai noteikti ceļi. Piemērs: `$.object.name`
`<jsonpath starting with @>`	Vērtība, uz kuru norāda JSONPath no pašreizējā objekta/elementa; tiek atbalstīti tikai noteikti 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 vai vienāds ar|Būla vērtība (1/0)| |>|Binārs|Lielāks par|Būla vērtība (1/0)| |>=|Binārs|Lielāks vai vienāds ar|Būla vērtība (1/0)| |=~|Binārs|Atbilst regulārajai izteiksmei|Būla vērtība (1/0)| |!|Vienāds|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": "reference",
      "author": "Nigel Rees",
      "title": "Gadsimta teicieni",
      "price": 8.95,
      "id": 1
    },
    {
      "category": "fiction",
      "author": "Evelyn Waugh",
      "title": "Goda zobens",
      "price": 12.99,
      "id": 2
    },
    {
      "category": "fiction",
      "author": "Herman Melville",
      "title": "Mobijs Diks",
      "isbn": "0-553-21311-3",
      "price": 8.99,
      "id": 3
    },
    {
      "category": "fiction",
      "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 iesiešana A5 formātā",
      "active": true,
      "price": 154.99
    },
    "restoration": {
      "servicegroup": 1002,
      "description": "Dažādas restaurācijas metodes",
```

```
      "active": false,
      "methods": [
        {
          "description": "Ķīmiskā tīrīšana",
          "price": 46
        },
        {
          "description": "Ar mitrumu bojātu lapu presēšana",
          "price": 24.5
        },
        {
          "description": "Saplēstas grāmatas pārsiešana",
          "price": 99.49
        }
      ]
    }
  },
  "filters": {
    "price": 10,
    "category": "fiction",
    "no filters": "no \"filters\""
  },
  "closed message": "Veikals ir slēgts",
  "tags": [
    "a",
    "b",
    "c",
    "d",
    "e"
  ]
}
```

JSONPath	Type	Result
`$.filters.price`	definite	10
`$.filters.category`	definite	fiction
`$.filters['no filters']`	definite	no ”filters”
`$.filters`	definite	{ ”price”: 10, ”category”: ”fiction”, ”no filters”: ”no \”filters\”” }
`$.books[1].title`	definite	Goda zobens
`$.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
`$.books[0, 2].title`	indefinite	[”Mobijs Diks”, ”Gadsimta teicieni”]
`$.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]
`$.books[?(@.id == 4 - 0.4 * 5)].title`	indefinite	[”Goda zobens”] Piezīme: Šis vaicājums parāda, ka vaicājumos var izmantot aritmētiskas darbības; to var vienkāršot līdz `$.books[?(@.id == 2)].title`

JSONPath	Type	Result
`$.books[?(@.id == 2 || @.id == 4)].title`	indefinite	[”Goda zobens”, ”Gredzenu pavēlnieks”]
`$.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”: ”reference”,
		”author”: ”Nigel Rees”,
		”title”: ”Gadsimta teicieni”
		},
		{
		”price”: 12.99,
		”id”: 2,
		”category”: ”fiction”,
		”author”: ”Evelyn Waugh”,
		”title”: ”Goda zobens”
		},
		{
		”price”: 8.99,
		”id”: 3,
		”category”: ”fiction”,
		”author”: ”Herman Melville”,
		”title”: ”Mobijs Diks”,
		”isbn”: ”0-553-21311-3”
		},
		{
		”price”: 22.99,
		”id”: 4,
		”category”: ”fiction”,
		”author”: ”J. R. R. Tolkien”,
		”title”: ”Gredzenu pavēlnieks”,
		”isbn”: ”0-395-19395-8”
		}
		]

`$.services..[?(@.price > 50)].description`	indefinite	[”Grāmatas drukāšana un iesiešana A5 formātā”, ”Saplēstas grāmatas pārsiešana”]
`$..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.

JSONPath	Type	Result
`$.books[*].price.min()`	definite	8.95
`$..price.max()`	definite	154.99
`$.books[?(@.category == "fiction")].price.avg()`	definite	14.99
`$.books[?(@.category == $.filters.xyz)].title`	indefinite	Piezīme: Vaicājums bez atbilstības atgriež NULL gan noteiktiem, gan nenoteiktiem ceļiem.
`$.services[?(@.active=="true")].servicegroup`	indefinite	[1001,1000] Piezīme: Teksta konstantes ir jāizmanto, salīdzinot loģiskās vērtības.
`$.services[?(@.active=="false")].servicegroup`	indefinite	[1002] Piezīme: Teksta konstantes ir jāizmanto, salīdzinot loģiskās vērtības.
`$.services[?(@.servicegroup==1002)].first()`	definite	{"servicegroup":1002,"active":"false"}

Īpašo rakstzīmju atsoļošana no LLD makro vērtībām JSONPath

Ja zemlīmeņa atklāšanas makro tiek izmantoti JSONPath pirmapstrādē un to vērtības tiek atrisinātas, tiek piemēroti šādi īpašo rakstzīmju atsoļošanas noteikumi:

- tikai atpakaļslīpsvītra (\) un dubultpēdiņas (") rakstzīmes tiek ņemtas vērā atsoļošanai;
- ja atrisinātajā makro vērtībā ir šīs rakstzīmes, katra no tām tiek atsoļota ar atpakaļslīpsvītru;
- ja tās jau ir atsoļotas ar atpakaļslīpsvītru, tas netiek uzskatīts par atsoļošanu, un gan atpakaļslīpsvītra, gan tai sekojošās īpašās rakstzīmes tiek atsoļotas 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, tam var būt īpašas rakstzīmes, un tas jāievieto 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, tam var būt īpašas rakstzīmes, un tas jāievieto 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**

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 ķermeni. Priekšapstrādes soļa rezultāts ir šīs funkcijas atgrieztā vērtība, piemēram, lai veiktu Fārenheita pārvēršanu Celsijā, ievadiet:

```
return (value - 32)  * 5 / 9
```

JavaScript priekšapstrādes parametros, kurus serveris ietvers JavaScript funkcijā:

```
function (value)
{
   return (value - 32) * 5 / 9
}
```

Ievades parametrs 'value' vienmēr tiek nodots kā virkne. Atgrieztā vērtība automātiski tiek pārveidota par virkni, izmantojot toString() metodi (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 'Discard value' priekšapstrāde darbībā 'Custom on fail'.

Kļūdas var atgriezt, izmetot vērtības/objektus (parasti vai nu virknes, vai Error objektus).

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 tiek aktivizēta); tās pārsniegšanas gadījumā tiks atgriezta kļūda. Tiek piemērots 512 megabaitu kaudzes ierobežojums.

JavaScript priekšapstrādes soļa baitkods tiek kešots un atkārtoti izmantots, kad solis tiek lietots nākamreiz. Jebkuras izmaiņas vienuma priekšapstrādes soļos izraisīs kešotā skripta atiestatīšanu un vēlāk tā atkārtotu kompilēšanu.

Secīgas izpildlaika kļūmes (3 pēc kārtas) izraisīs dzinēja inicializēšanu no jauna, lai mazinātu iespēju, ka viens skripts sabojā izpildes vidi nākamajiem skriptiem (šī darbība tiek reģistrēta ar DebugLevel 4 un augstāku).

JavaScript priekšapstrāde ir ieviesta, izmantojot Duktape JavaScript dzinēju.

> **Note:**
> Iebūvēto JavaScript objektu metožu definīcijas nevar pārrakstīt vienuma priekšapstrādes skriptos.

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 Zabbix aģenta vienuma web.page.get (piemēram, web.page.get[http://127.0.0.1:80/server-status?auto]), par JSON objektu.

```
// Convert Apache status to JSON

// Split the value into substrings and put these substrings into an array
var lines = value.split('\n');

// Create an empty object "output"
var output = {};

// Create an object "workers" with predefined properties
var workers = {
    '_': 0, 'S': 0, 'R': 0, 'W': 0,
    'K': 0, 'D': 0, 'C': 0, 'L': 0,
    'G': 0, 'I': 0, '.': 0
};

// Add the substrings from the "lines" array to the "output" object as properties (key-value pairs)
for (var i = 0; i < lines.length; i++) {
    var line = lines[i].match(/([A-z0-9 ]+): (.*)/);

    if (line !== null) {
        output[line[1]] = isNaN(line[2]) ? line[2] : Number(line[2]);
    }
}

// Multiversion metrics
output.ServerUptimeSeconds = output.ServerUptimeSeconds || output.Uptime;
output.ServerVersion = output.ServerVersion || output.Server;

// Parse "Scoreboard" property to get the worker count
if (typeof output.Scoreboard === 'string') {
    for (var i = 0; i < output.Scoreboard.length; i++) {
        var char = output.Scoreboard[i];

        workers[char]++;
    }
}

// Add worker data to the "output" object
output.Workers = {
    waiting: workers['_'], starting: workers['S'], reading: workers['R'],
    sending: workers['W'], keepalive: workers['K'], dnslookup: workers['D'],
    closing: workers['C'], logging: workers['L'], finishing: workers['G'],
    cleanup: workers['I'], slot: workers['.']
};

// Return JSON string
return JSON.stringify(output);
```

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
    }
}
```

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 iekapsulē cURL apdarinātāju, ļaujot veikt vienkāršus HTTP pieprasījumus. Kļūdas tiek izmestas kā izņēmumi.

Attention:
Vairāku `HttpRequest` objektu inicializēšana vienas skripta izpildes laikā ir ierobežota līdz 10.

Method	Description
`addHeader(value)`	Pievieno HTTP galvenes lauku. Šis lauks tiek izmantots visiem turpmākajiem pieprasījumiem, līdz tas tiek notīrīts ar metodi `clearHeader()`. Galvenes lauku kopējais garums, ko var pievienot vienam `HttpRequest` objektam, ir ierobežots līdz 128 kilobaitiem (ieskaitot speciālās rakstzīmes un galveņu nosaukumus).
`clearHeader()`	Notīra HTTP galveni. Ja nav iestatīts neviens galvenes lauks, `HttpRequest` iestatīs Content-Type vērtību 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)`	Pirmajā parametrā ļauj norādīt jebkuru HTTP metodi. Nosūta metodes pieprasījumu uz URL ar neobligātu *data* datu saturu un atgriež atbildi.
`delete(url, data)`	Nosūta HTTP DELETE pieprasījumu uz URL ar neobligātu *data* datu saturu un atgriež atbildi.
`getHeaders(<asArray>)`	Atgriež saņemto HTTP galvenes lauku objektu. Parametram `asArray` var iestatīt vērtību "true" (piemēram, `getHeaders(true)`), "false, vai arī to var nenorādīt. Ja iestatīta vērtība "true", saņemto HTTP galvenes lauku vērtības tiek atgrieztas kā masīvi; tas jāizmanto, lai izgūtu vairāku galveņu ar vienādu nosaukumu lauku vērtības. Ja parametrs nav norādīts vai tam iestatīta vērtība "false", saņemto HTTP galvenes lauku vērtības tiek atgrieztas kā virknes.| |get(url, data)|`Nosūta HTTP GET pieprasījumu uz URL ar neobligātu *data* datu saturu 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* datu saturu un atgriež atbildi.`| |put(url, data)|`Nosūta HTTP PUT pieprasījumu uz URL ar neobligātu *data* datu saturu un atgriež atbildi.`| |post(url, data)|`Nosūta HTTP POST pieprasījumu uz URL ar neobligātu *data* datu saturu un atgriež atbildi.`| |getStatus()|`Atgriež pēdējā HTTP pieprasījuma statusa kodu.`| |setProxy(proxy)|`Iestata HTTP starpniekserveri uz vērtību "proxy". Ja šis parametrs ir tukšs, starpniekserveris netiek izmantots.`| |setHttpAuth(bitmask, username, password)|Parametrā 'bitmask' iestata iespējotās HTTP autentifikācijas metodes (HTTPAUTH_BASIC, HTTPAUTH_DIGEST, HTTPAUTH_NEGOTIATE, HTTPAUTH_NTLM, HTTPAUTH_NONE). Karodziņš HTTPAUTH_NONE ļauj atspējot HTTP autentifikāciju. Piemēri: "request.setHttpAuth(HTTPAUTH_NTLM | HTTPAUTH_BASIC, username, password)" request.setHttpAuth(HTTPAUTH_NONE)| |trace(url, data)'

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’).

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 vērtību kā virkni (vai atgriež `null`, ja norādītais atribūts netika atrasts). Parameters: `name` - (string) Atribūta nosaukums.
`getProperty(name)`	Atgriež elementa īpašības vērtību kā virkni (vai atgriež `null`, ja norādītā īpašība netika atrasta). Parameters: `name` - (string) Īpašības nosaukums.
`getText()`	Atgriež elementa teksta vērtību kā virkni.
`click()`	Noklikšķina uz elementa.
`clear()`	Notīra rediģējama elementa saturu.
`sendKeys(keys)`	Nosūta taustiņus. Parameters: `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.

CSV uz JSON priekšapstrāde

Pārskats

Šajā priekšapstrādes darbībā CSV faila datus var pārveidot JSON formātā. Tā tiek atbalstīta:

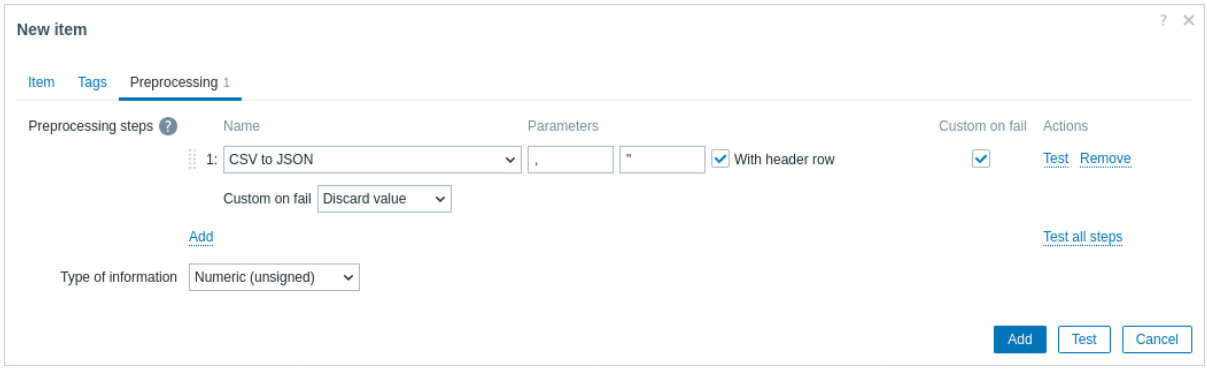
- vienumos (vienumu prototipos)
- zema līmeņa atklāšanas noteikumos

Konfigurēšana

Lai konfigurētu CSV uz JSON priekšapstrādes darbību:

1. Atveriet cilni *Priekšapstrāde* vienuma/atklāšanas kārtulas konfigurācijā.

2. Noklikšķiniet uz *Pievienot*.
3. Atlasiet opciju *CSV uz JSON*.



Pirmais parametrs ļauj iestatīt pielāgotu atdalītāju. Ņemiet vērā: ja CSV ievades pirmā rinda sākas ar "Sep=" un tai seko viena UTF-8 rakstzīme, šī rakstzīme tiks izmantota kā atdalītājs, ja pirmais parametrs nav iestatīts. Ja pirmais parametrs nav iestatīts un atdalītājs netiek iegūts no rindas "Sep=", kā atdalītājs tiek izmantots komats.

Otrais neobligātais parametrs ļauj iestatīt pēdiņu simbolu.

Ja ir atzīmēta izvēles rūtiņa *Ar galvenes rindu*, galvenes rindas vērtības tiks interpretētas kā kolonnu nosaukumi (plašāku informāciju skatiet sadaļā Galvenes apstrāde).

Ja ir atzīmēta izvēles rūtiņa *Pielāgota darbība kļūmes gadījumā*, vienums nekļūs par neatbalstītu, ja priekšapstrādes darbība neizdosies. Turklāt var iestatīt pielāgotas kļūdu apstrādes opcijas: atmest vērtību, iestatīt norādītu vērtību vai iestatīt norādītu kļūdas ziņojumu.

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"
    }
]
```

Vienumu tipi

Pārskats

Vienumu tipi aptver dažādas datu iegūšanas metodes no jūsu sistēmas. Katram vienuma tipam ir savs atbalstīto vienumu atslēgu un nepieciešamo parametru kopums.

Zabbix pašlaik piedāvā šādus vienumu tipus:

- Zabbix aģents
  - Žurnālfailu uzraudzība
- Vienkāršā pārbaude
  - VMware uzraudzība
- SNMP aģents
- SNMP traps
- Zabbix iekšējais
- Zabbix trapper
- Ārējā pārbaude
- Datu bāzes uzraudzība
- HTTP aģents
  - Prometheus pārbaude
- IPMI aģents
- SSH aģents
- Telnet aģents
- JMX aģents
- Aprēķinātais vienums
- Atkarīgais vienums
- Skripta vienums
- Pārlūkprogrammas vienums
- Telemetrijas vaicājums

Detalizēta informācija par visiem vienumu tipiem ir pieejama šīs sadaļas apakšlapās. Lai gan vienumu tipi piedāvā daudz datu iegūšanas iespēju, papildu iespējas nodrošina lietotāja parametri vai ielādējamie moduļi.

Dažas pārbaudes veic tikai Zabbix serveris (kā uzraudzību bez aģenta), savukārt citām ir nepieciešams Zabbix aģents vai pat Zabbix Java gateway (JMX uzraudzībai).

Attention:
Ja konkrētam vienuma tipam ir nepieciešama konkrēta saskarne (piemēram, IPMI pārbaudei hostā ir nepieciešama IPMI saskarne), šai saskarnei ir jābūt norādītai hosta definīcijā.

Hosta definīcijā var iestatīt vairākas saskarnes: Zabbix aģenta, SNMP aģenta, JMX un IPMI saskarni. Ja vienums var izmantot vairāk nekā vienu saskarni, tas meklēs pieejamajās hosta saskarnēs (šādā secībā: Agent→SNMP→JMX→IPMI) pirmo piemēroto saskarni, ar kuru to saistīt.

Visi vienumi, kas atgriež tekstu (rakstzīmju, žurnāla vai teksta informācijas tipi), var atgriezt arī tikai atstarpes (ja piemērojams), iestatot atgriežamo vērtību uz tukšu virkni (atbalstīts kopš versijas 2.0).

Zabbix aģents

Pārskats

Šī sadaļa sniedz informāciju par vienumu atslēgām, kas datu iegūšanai izmanto saziņu ar Zabbix aģentu.

Ir pasīvas un aktīvas aģenta pārbaudes. Konfigurējot vienumu, varat atlasīt nepieciešamo tipu:

- *Zabbix aģents* - pasīvām pārbaudēm
- *Zabbix aģents (aktīvs)* - aktīvām pārbaudēm

Ņemiet vērā, ka visas vienumu atslēgas, ko atbalsta Zabbix aģents, atbalsta arī jaunās paaudzes Zabbix aģents 2. Skatiet papildu vienumu atslēgas, ko varat izmantot tikai ar aģentu 2.

Atbalstītās vienumu atslēgas

Tālāk ir norādītas vienumu atslēgas, kuras varat izmantot kopā ar Zabbix aģentu.

Vienumu atslēgas ir norādītas bez parametriem un papildu informācijas.
Noklikšķiniet uz vienuma atslēgas, lai skatītu pilnu informāciju.

Vienuma atslēga	Apraksts	Vienumu grupa
kernel.maxfiles	Maksimālais OS atbalstītais atvērto failu skaits.	Kodols
kernel.maxproc	Maksimālais OS atbalstītais procesu skaits.	
kernel.openfiles	Pašlaik atvērto failu deskriptoru skaits.	
log	Žurnālfaila uzraudzība.	Žurnālu uzraudzība
log.count	Atbilstošo rindu skaits uzraudzītajā žurnālfailā.	
logrt	Rotēta žurnālfaila uzraudzība.	
logrt.count	Atbilstošo rindu skaits rotētajā uzraudzītajā žurnālfailā.	
modbus.get	Nolasa Modbus datus.	Modbus
net.dns	Pārbauda DNS pakalpojuma statusu.	Tīkls
net.dns.perf	Pārbauda DNS pakalpojuma veiktspēju.	
net.dns.record	Veic DNS vaicājumu.	
net.if.collisions	Sadursmju skaits ārpus loga.	
net.if.discovery	Tīkla saskarņu saraksts.	
net.if.get	Tīkla saskarņu saraksts ar detalizētu informāciju.	
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 stāvoklī LISTEN.	
net.tcp.port	Pārbauda, vai ir iespējams izveidot TCP savienojumu ar norādīto portu.	
net.tcp.service	Pārbauda, vai pakalpojums darbojas un pieņem TCP savienojumus.	
net.tcp.service.perf	Pārbauda TCP pakalpojuma veiktspēju.	
net.tcp.socket.count	Atgriež parametriem atbilstošo TCP ligzdu skaitu.	
net.udp.listen	Pārbauda, vai šis UDP ports ir stāvoklī LISTEN.	
net.udp.service	Pārbauda, vai pakalpojums darbojas un atbild uz UDP pieprasījumiem.	
net.udp.service.perf	Pārbauda UDP pakalpojuma veiktspēju.	

Vienuma atslēga	Apraksts	Vienumu grupa
net.udp.socket.count	Atgriež parametriem atbilstošo UDP ligzdu skaitu.	
proc.cpu.util	Procesa CPU izmantojuma procentuālā daļa.	Procesi
proc.get	OS 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/procesora kodolu saraksts.	
system.cpu.intr	Ierīču pārtraukumi.	
system.cpu.load	CPU noslodze.	
system.cpu.num	CPU skaits.	
system.cpu.switches	Konteksta pārslēgšanu skaits.	
system.cpu.util	CPU izmantojuma procentuālā daļa.	
system.hostname	Sistēmas hosts nosaukums.	
system.hw.chassis	Informācija par šasiju.	
system.hw.cpu	Informācija par CPU.	
system.hw.devices	PCI vai USB ierīču saraksts.	
system.hw.macaddr	MAC adrešu saraksts.	
system.localtime	Sistēmas laiks.	
system.run	Palaiž norādīto komandu hostā.	
system.stat	Sistēmas statistika.	
system.sw.arch	Informācija par programmatūras arhitektūru.	
system.sw.os	Informācija par operētājsistēmu.	
system.sw.os.get	Detalizēta informācija par operētājsistēmu (versija, tips, distribūcijas nosaukums, mazā un lielā versija u. c.).	
system.sw.packages	Instalēto pakotņu saraksts.	
system.sw.packages.get	Detalizēts instalēto pakotņu saraksts.	
system.swap.in	Swap-in (no ierīces atmiņā) statistika.	
system.swap.out	Swap-out (no atmiņas uz ierīci) statistika.	
system.swap.size	Swap 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.get	Blokierīču saraksts ar detalizētu informāciju.	
vfs.dev.read	Diska lasīšanas statistika.	
vfs.dev.write	Diska rakstīšanas statistika.	
vfs.dir.count	Direktorijas ierakstu skaits.	
vfs.dir.get	Direktorijas ierakstu saraksts.	
vfs.dir.size	Direktorijas lielums.	
vfs.file.cksum	Faila kontrolsumma, kas aprēķināta, izmantojot UNIX cksum algoritmu.	
vfs.file.contents	Faila satura izgūšana.	
vfs.file.exists	Pārbauda, vai fails pastāv.	
vfs.file.get	Atgriež informāciju par failu.	
vfs.file.md5sum	Faila MD5 kontrolsumma.	
vfs.file.owner	Iegūst faila īpašnieku.	
vfs.file.permissions	Atgriež četrciparu virkni, kas satur oktālo skaitli ar UNIX atļaujām.	
vfs.file.regexp	Iegūst virkni no faila.	
vfs.file.regmatch	Atrod virkni failā.	
vfs.file.size	Faila lielums.	
vfs.file.time	Informācija par faila laiku.	
vfs.fs.discovery	Piemontēto failu sistēmu saraksts ar to tipu un montēšanas opcijām.	
vfs.fs.get	Piemontēto failu sistēmu saraksts ar to tipu un montēšanas opcijām, kā arī *full* režīmā pieejamās diska vietas un inode statistika.	
vfs.fs.inode	Inode skaits vai procentuālā daļa.	
vfs.fs.size	Diska vieta baitos vai procentos no kopējā apjoma.	
vm.memory.size	Atmiņas lielums baitos vai procentos no kopējā apjoma.	Virtuālā atmiņa

Vienuma atslēga	Apraksts	Vienumu grupa
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 hosts metadati.	Zabbix
agent.hostname	Aģenta hosts nosaukums.	
agent.ping	Aģenta pieejamības pārbaude.	
agent.variant	Zabbix aģenta variants (Zabbix aģents vai Zabbix aģents 2).	
agent.version	Zabbix aģenta versija.	
zabbix.stats	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.	
zabbix.stats (queue)	Atgriež uzraudzīto vienumu skaitu rindā, kuru apstrāde Zabbix serverī vai starpniekserverī 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>]

 Log faila uzraudzība.
 Atgrieztā vērtība: *Log*.
 Skatiet atbalstītās platformas.

Parametri:

- **file** - pilns ceļš un log faila nosaukums;
- **regexp** - regulāra izteiksme, kas apraksta nepieciešamo rakstu;
- **encoding** - kodu lapas identifikators;
- **maxlines** - maksimālais jauno rindu skaits sekundē, ko aģents nosūtīs uz serveris vai starpniekserveris. Šis parametrs pārraksta `MaxLinesPerSecond` vērtību `zabbix_agentd.conf`;
- **mode** - iespējamās vērtības:
  - *all* (noklusējums);
  - *skip* - izlaist vecāku datu apstrādi (ietekmē tikai jaunizveidotos vienumus);
- **output** - neobligāta izvades formatēšanas veidne. **\0** bēgšanas secība tiek aizstāta ar atbilstošo teksta daļu (no pirmās rakstzīmes, kur sākas atbilstība, līdz rakstzīmei, kur atbilstība beidzas), savukārt **\N** (kur N=1...9) bēgšanas secība tiek aizstāta ar N-to atbilstošo grupu (vai ar tukšu virkni, ja N pārsniedz uztverto grupu skaitu);

- **maxdelay** - maksimālā aizture sekundēs. Tips: float. Vērtības:
  - 0 - (noklusējums) nekad neignorēt log faila rindas;
  - > 0.0 - ignorēt vecākas rindas, lai pēdējās analizētās rindas tiktu iegūtas ”maxdelay” sekunžu laikā. Pirms lietošanas izlasiet piezīmes par maxdelay!
- **options** - papildu opcijas:
mtime-noreread - neunikāli ieraksti, pārlasīt tikai tad, ja faila izmērs mainās (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 agent 2) - direktorijas absolūtais ceļš, kur glabāt pastāvīgos failus. Skatiet arī papildu piezīmes par pastāvīgajiem failiem.

Komentāri:

- Vienums ir jākonfigurē kā aktīvā pārbaude.
- Ja fails nav pieejams vai atļaujas liedz piekļuvi, vienums kļūst neatbalstīts.
- Ja `output` ir atstāts tukšs - tiek atgriezta visa rinda, kurā ir atbilstošais teksts. Ņemiet vērā, ka visi globālo regulāro izteiksmju tipi, izņemot *Matches regular expression*, vienmēr atgriež visu atbilstošo rindu, un parametrs `output` tiek ignorēts.
- Satura izvilkšana, izmantojot `output` parametru, notiek aģentā.

Piemēri:

```
log[/var/log/syslog]
log[/var/log/syslog,error]
log[/home/zabbix/logs/logfile,,,100]
```

Piemērs `output` parametra izmantošanai skaitļa izvilkšanai no log ieraksta:

```
log[/app1/app.log,"task run [0-9.]+ sec, processed ([0-9]+) records, [0-9]+ errors",,,,\1] #this item will
```

Piemērs `output` parametra izmantošanai log ieraksta pārrakstīšanai 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>]

 Sakritoto rindu skaits uzraudzītajā žurnāla failā.
 Atgrieztā vērtība: *Integer*.
 Skatiet atbalstītās platformas.

Parametri:

- **file** - pilns žurnāla faila ceļš un nosaukums;
- **regexp** - regulāra izteiksme, kas apraksta nepieciešamo rakstu;
- **encoding** - kodu lapas 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);
  - *skip* - izlaist vecāku datu apstrādi (ietekmē tikai jaunizveidotos vienumus).
- **maxdelay** - maksimālā aizkave sekundēs. Tips: float. Vērtības:
  - 0 - (noklusējums) nekad neignorēt žurnāla faila rindas;
  - > 0.0 - ignorēt vecākas rindas, lai jaunākās rindas tiktu 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; nav atbalstīts Zabbix aģents 2) - direktorijas absolūtais ceļš, kur glabāt pastāvīgos failus. Skatiet arī papildu piezīmes par pastāvīgajiem failiem.

Komentāri:

- Vienums ir jākonfigurē kā aktīvā pārbaude.
- Sakritīgās rindas tiek skaitītas jaunajās rindās kopš pēdējās žurnāla pārbaudes, ko veicis aģents, un tādējādi ir atkarīgas no vienuma atjaunināšanas intervāla.
- Ja fails nav atrodams vai piekļuve tam nav atļauta, vienums kļūst neatbalstīts.

logrt[file regexp,<regexp>,<encoding>,<maxlines>,<mode>,<output>,<maxdelay>,<options>,<persistent dir>]

 Pārraudzība rotētam žurnālfailam.
 Atgrieztā vērtība: *Log*.
 Skatiet atbalstītās platformas.

Parametri:

- **file regexp** - absolūtais ceļš uz failu, kur faila nosaukums ir norādīts, izmantojot regulāro izteiksmi. Ņemiet vērā, ka regulārā izteiksme attiecas tikai uz faila nosaukumu un tai nav jāsakrīt ar visu nosaukumu (piem., `/path/to/agent` sakritīs ar `zabbix_agentd.log`).
- **regexp** - regulārā izteiksme, kas apraksta nepieciešamo satura paraugu.

- **encoding** - kodu lapas identifikators.
- **maxlines** - maksimālais jauno rindu skaits sekundē, ko aģents nosūtīs uz Zabbix serveris vai starpniekserveris. Šis parametrs pārraksta `MaxLinesPerSecond` vērtību `zabbix_agentd.conf`.
- **mode** - iespējamās vērtības:
  - *all* (noklusējums);
  - *skip* - izlaiž vecāku datu apstrādi (ietekmē tikai jaunizveidotos vienumus).
- **output** - neobligāta izvades formatēšanas veidne. **\0** bēgšanas secība tiek aizstāta ar atrasto teksta daļu (no pirmās rakstzīmes, kur sākas sakritība, līdz rakstzīmei, kur sakritība beidzas), savukārt **\N** (kur N=1...9) bēgšanas secība tiek aizstāta ar N-to atrasto grupu (vai tukšu virkni, ja N pārsniedz uztverto grupu skaitu).
- **maxdelay** - maksimālā aizture sekundēs. Tips: float. Vērtības:
  - 0 - (noklusējums) nekad neignorēt žurnālfaila rindas;
  - > 0.0 - ignorēt vecākas rindas, lai pēdējās analizētās rindas tiktu iegūtas "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* - nedublējoši ieraksti, pārlasa, ja mainās modificēšanas laiks vai izmērs (pēc noklusējuma tiek izmantots, ja nav skaidri iestatīta neviena `mtime-*` opcija).
  - *mtime-noreread* - nedublējoši ieraksti, pārlasa tikai tad, ja mainās izmērs (ignorē modificēšanas laika izmaiņas).
- **persistent dir** (tikai `zabbix_agentd` Unix sistēmās; Zabbix aģents 2 netiek atbalstīts) - direktorijas absolūtais ceļš, kur glabāt pastāvīgos failus. Skatiet arī papildu piezīmes par pastāvīgajiem failiem.

Piezīmes:

- Vienums ir jākonfigurē kā aktīvā pārbaude.
- Žurnālfaila rotācija balstās uz failu pēdējo modificēšanas laiku.
- Ņemiet vērā, ka `logrt` ir paredzēts darbam ar vienu pašlaik aktīvu žurnālfailu, kam ir vairāki citi atbilstoši neaktīvi rotēti faili. Ja, piemēram, direktorijā ir daudz aktīvu žurnālfailu, katram no tiem jāizveido atsevišķs `logrt` vienums. Pretējā gadījumā, ja viens `logrt` vienums apstrādā pārāk daudz failu, tas var izraisīt atmiņas izsīkumu un pārraudzības avāriju.
- Ja `output` tiek atstāts tukšs, tiek atgriezta visa rinda, kurā ir atrastais teksts. Ņemiet vērā, ka visi globālo regulāro izteiksmju tipi, izņemot *Matches regular expression*, vienmēr atgriež visu atrasto rindu, un `output` parametrs tiek ignorēts.
- Satura izvilkšana, izmantojot `output` parametru, notiek aģentā.
- Parametrā `file regexp` žurnāla direktorijas ceļam un žurnālfaila regulārajai izteiksmei jābūt atdalītiem ar pareizo direktorijas atdalītāju:
  - Operētājsistēmā Windows atdalītājam jābūt slīpsvītrai atpakaļ (\); slīpsvītras uz priekšu var tikt pieņemtas citās vietās, izņemot to, kas atdala žurnāla direktorijas ceļu un žurnālfaila regulāro izteiksmi (skatiet piemērus zemāk).
  - Citās sistēmās tam jābūt slīpsvītrai uz priekšu (/).

Windows piemēri:

```
logrt["c:/dir1/dir2/dir3\filename.*\.log","pattern_to_match"] #this item will collect data from log files
logrt["//example.com/share/dir1/dir2/dir3\filename.*\.log","pattern_to_match"] #this item will collect dat
```

Piemēri citām sistēmām:

```
logrt["/home/zabbix/logs/^logfile[0-9]{1,3}$",,,100] #this item will match a file like "logfile1" (will no
logrt["/home/user/^logfile_.*_[0-9]{1,3}$","pattern_to_match","UTF-8",100] #this item will collect data fr
```

Piemērs `output` parametra izmantošanai skaitļa izvilkšanai no žurnāla ieraksta:

```
logrt[/app1/^test.*log$,"task run [0-9.]+ sec, processed ([0-9]+) records, [0-9]+ errors",,,,\1] #this ite
```

Piemērs `output` parametra izmantošanai žurnāla ieraksta pārrakstīšanai 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>]

 Pārraudzītā rotējošā žurnālfailā atrasto atbilstošo rindu skaits.
 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 (piem., `/path/to/agent` sakritīs ar `zabbix_agentd.log`).
- **regexp** - regulāra izteiksme, kas apraksta nepieciešamo rakstu.
- **encoding** - kodu lapas 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);
  - *skip* - izlaiž vecāku datu apstrādi (ietekmē tikai jaunizveidotos vienumus).
- **maxdelay** - maksimālā aizture sekundēs. Tips: float. Vērtības:
  - 0 - (noklusējums) nekad neignorēt žurnālfaila rindas;
  - > 0.0 - ignorēt vecākas rindas, lai jaunākās rindas tiktu 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 tas nav jānorāda; skatiet piezīmes par copytruncate.
  - *mtime-reread* - nedubultoti ieraksti, pārlasīt, ja mainās modificēšanas laiks vai izmērs (noklusējums).
  - *mtime-noreread* - nedubultoti ieraksti, pārlasīt tikai tad, ja mainās izmērs (ignorēt modificēšanas laika izmaiņas).
- **persistent dir** (tikai zabbix_agentd Unix sistēmās; nav atbalstīts Zabbix aģents 2) - direktorijas absolūtais ceļš, kur glabāt pastāvīgos failus. Skatiet arī papildu piezīmes par pastāvīgajiem failiem.

Komentāri:

- Vienums ir jākonfigurē kā aktīvā pārbaude.
- Atbilstošās rindas tiek skaitītas jaunajās rindās kopš pēdējās žurnāla pārbaudes, ko veic aģents, un tādējādi ir atkarīgas no vienuma atjaunināšanas intervāla.
- Žurnālfaila rotācija balstās uz failu pēdējo modificēšanas laiku.
- Parametrā file regexp žurnāla direktorijas ceļš un žurnālfaila regulārā izteiksme ir jāatdala ar pareizo direktorijas atdalītāju:
  - Operētājsistēmā Windows atdalītājam jābūt slīpsvītrai pretējā virzienā (\); slīpsvītras uz priekšu var tikt pieņemtas citās pozīcijās, izņemot to, kas atdala žurnāla direktorijas ceļu un žurnālfaila regulāro izteiksmi (skatiet piemērus zemāk).
  - Citās sistēmās tam jābūt slīpsvītrai uz priekšu (/).

Piemēri operētājsistēmai Windows:

```
logrt.count["c:/dir1/dir2/dir3\filename.*\.log","pattern_to_match"] #šis vienums skaitīs atbilstību skaitu
logrt.count["//example.com/share/dir1/dir2/dir3\filename.*\.log","pattern_to_match"] #šis vienums skaitīs
```

Piemēri citām sistēmām:

```
logrt.count["/home/zabbix/logs/.*","^logfile[0-9]{1,3}$",,100] #šis vienums skaitīs atbilstību skaitu raks
logrt.count["/home/user/^logfile_.*_[0-9]{1,3}$","pattern_to_match","UTF-8",100] #šis vienums skaitīs raks
```

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** - slave ID;
- **function** - Modbus funkcija;
- **address** - pirmā reģistra, coil vai ievades adrese;
- **count** - nolasāmo ierakstu skaits;
- **type** - datu tips;
- **endianness** - baitu secības konfigurācija;
- **offset** - reģistru skaits, sākot no address, kuru rezultāti tiks ignorēti.

Operētājsistēmā Windows šis vienums seko UNC ceļiem endpoint parametrā, kas var būt drošības risks.

Skatiet detalizētu parametru 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 operētājsistēmā Windows, ja vien netiek izmantots Zabbix aģents 2) - DNS servera IP adrese (atstājiet tukšu, lai izmantotu noklusējuma DNS serveri);
- **name** - DNS nosaukums, ko vaicāt;
- **type** - ieraksta tips, ko vaicāt (noklusējums ir SOA);

- **timeout** (tiek ignorēts operētājsistēmā Windows, ja vien netiek izmantots Zabbix aģents 2) - pieprasījuma noildze sekundēs (noklusējums ir 1 sekunde);
- **count** (tiek ignorēts operētājsistēmā Windows, ja vien netiek izmantots Zabbix aģents 2) - pieprasījuma mēģinājumu skaits (noklusējums ir 2);
- **protocol** - protokols, ko izmanto DNS vaicājumu veikšanai: *udp* (noklusējums) vai *tcp*.

Piezīmes:

- Iespējamās `type` vērtības ir: *ANY*, *A*, *NS*, *CNAME*, *MB*, *MG*, *MR*, *PTR*, *MD*, *MF*, *MX*, *SOA*, *NULL*, *WKS* (netiek atbalstīts Zabbix aģentam operētājsistēmā Windows, Zabbix aģentam 2 visās OS), *HINFO*, *MINFO*, *TXT*, *SRV*.
- Apgrieztajiem DNS uzmeklējumiem (ja `type` ir iestatīts uz *PTR*) DNS nosaukumu var norādīt gan apgrieztā, gan neapgrieztā formātā (skatiet piemērus tālāk). Ņemiet vērā, ka, pieprasot PTR ierakstu, DNS nosaukums faktiski ir IP adrese.
- Internacionalizētie domēna vārdi netiek atbalstīti, lūdzu, izmantojiet 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.0.0.2.0.0.0.0.0.0.0.0.0.0.0.0.8.0.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 pakalpojuma veiktspēju.
 Atgrieztā vērtība: *Float* (0 - pakalpojums nedarbojas; sekundes - sekunžu skaits, kas pavadīts, gaidot atbildi no pakalpojuma).
 Skatiet atbalstītās platformas.

Parametri:

- **ip** (Windows sistēmā tiek ignorēts, ja vien netiek izmantots Zabbix aģents 2) - DNS servera IP adrese (atstājiet tukšu, lai izmantotu noklusējuma DNS serveri);
- **name** - DNS nosaukums, kuru vaicāt;
- **type** - ieraksta tips, kuru vaicāt (noklusējums ir `SOA`);
- **timeout** (Windows sistēmā tiek ignorēts, ja vien netiek izmantots Zabbix aģents 2) - pieprasījuma noildze sekundēs (noklusējums ir 1 sekunde);
- **count** (Windows sistēmā tiek ignorēts, 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* (nav atbalstīts Zabbix aģentam operētājsistēmā Windows, Zabbix aģentam 2 visās OS), *HINFO*, *MINFO*, *TXT*, *SRV*.
- Veicot reversos DNS vaicājumus (kad `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.
- Internacionalizētie domēna vārdi netiek atbalstīti, lūdzu, izmantojiet IDNA kodētus nosaukumus.
- Vienums atgriež atbildes laiku, nevis 0, 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.0.0.2.0.0.0.0.0.0.0.0.0.0.0.0.8.0.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 arī nodrošina net.dns.get, kas piedāvā papildu iespējas, piemēram, vairāk ierakstu tipu un lielāku kontroli pār DNS uzraudzību. Atgrieztā vērtība: rakstzīmju virkne ar nepieciešamo informācijas tipu.
 Skatiet atbalstītās platformas.

Parametri:

- **ip** (Windows sistēmā tiek ignorēts, ja vien netiek izmantots Zabbix aģents 2) - DNS servera IP adrese (atstājiet tukšu, lai izmantotu noklusējuma DNS serveri);

- **name** - DNS nosaukums, kuru vaicāt;
- **type** - vaicājamais ieraksta tips (noklusējuma vērtība ir SOA);
- **timeout** (Windows sistēmā tiek ignorēts, ja vien netiek izmantots Zabbix aģents 2) - pieprasījuma noildze sekundēs (noklusējuma vērtība ir 1 sekunde);
- **count** (Windows sistēmā tiek ignorēts, ja vien netiek izmantots Zabbix aģents 2) - pieprasījuma mēģinājumu skaits (noklusējuma vērtība ir 2);
- **protocol** - protokols, ko izmanto DNS vaicājumu veikšanai: *udp* (noklusējuma) vai *tcp*.

Piezīmes:

- Iespējamās `type` vērtības ir:
ANY, *A*, *NS*, *CNAME*, *MB*, *MG*, *MR*, *PTR*, *MD*, *MF*, *MX*, *SOA*, *NULL*, *WKS* (netiek atbalstīts Zabbix aģentam operētājsistēmā Windows, Zabbix aģentam 2 visās OS), *HINFO*, *MINFO*, *TXT*, *SRV*.
- Apgrieztajiem DNS uzmeklējumiem (kad `type` ir iestatīts uz *PTR*) varat norādīt DNS nosaukumu gan apgrieztā, gan neapgrieztā formātā (skatiet piemērus zemāk). Ņemiet vērā, ka, pieprasot PTR ierakstu, DNS nosaukums faktiski ir IP adrese.
- Internacionalizētie domēna vārdi netiek atbalstīti, lūdzu, 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. Izmanto zema līmeņa atklāšanai.
 Atgrieztā vērtība: *JSON string*.
 Atbalstītās platformas: Linux, FreeBSD, Solaris, HP-UX, AIX, OpenBSD, NetBSD, Windows.

net.if.get[<regexp>]

 Tīkla saskarņu saraksts ar detalizētu informāciju.
 Tiek izmantots zemā līmeņa atklāšanai.
 Šis vienums ir paplašināta Zabbix aģenta vienuma `net.if.discovery` versija.
 Atgrieztā vērtība: *JSON objekts*.
 Atbalstītās platformas: Linux.

Parametri:

- **regexp** - regulāra izteiksme, kas atbilst tīkla saskarņu nosaukumiem.

Komentāri:

- Atgriež JSON objektu ar tīkla saskarnēm, kuru nosaukumi atbilst dotajai regulārajai izteiksmei (skatiet papildu informāciju).

net.if.in[if,<mode>]

 Ienākošās datplūsmas statistika tīkla interfeisā.
 Atgrieztā vērtība: *Integer*.
 Atbalstītās platformas: Linux, FreeBSD, Solaris[5], HP-UX, AIX, MacOS X, OpenBSD, NetBSD, Windows. NetBSD ir nepieciešamas root tiesības.

Parametri:

- **if** - tīkla interfeisa nosaukums (Unix); tīkla interfeisa pilns apraksts vai IPv4 adrese; vai, ja iekļauts iekavās, tīkla interfeisa 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 kadrējuma kļūdu skaits;
  - *compressed* - ierīces draivera saņemto saspiesto pakešu skaits;
  - *multicast* - ierīces draivera saņemto multicast kadru skaits.

Komentāri:

- Varat izmantot šo atslēgu kopā ar priekšapstrādes soli *Change per second*, lai iegūtu baitu skaitu 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 šis vienums nesniedz detalizētu informāciju par loopback interfeisiem (piem., lo0).

Piemēri:

```
net.if.in[eth0]
net.if.in[eth0,errors]
```

net.if.out[if,<mode>]

 Izejošās datplūsmas statistika tīkla interfeisā.
 Atgrieztā vērtība: *Integer*.
 Atbalstītās platformas: Linux, FreeBSD, Solaris[5], HP-UX, AIX, MacOS X, OpenBSD, NetBSD, Windows. NetBSD ir nepieciešamas root tiesības.

Parametri:

- **if** - tīkla interfeisa nosaukums (Unix); tīkla interfeisa pilns apraksts vai IPv4 adrese; vai, ja iekavās, tīkla interfeisa 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)* - sadursmju skaits, kas konstatētas interfeisā;
  - *carrier* - nesēja zudumu skaits, ko konstatējis ierīces draiveris;
  - *compressed* - saspiesto pakešu skaits, ko pārsūtījis ierīces draiveris.

Komentāri:

- Lai iegūtu baitu sekundē statistiku, varat izmantot šo atslēgu kopā ar priekšapstrādes soli *Change per second*.
- Režīms *dropped* tiek atbalstīts tikai Linux, HP-UX.
- Režīmi *overruns*, *collision*, *carrier*, *compressed* tiek atbalstīti tikai Linux.
- HP-UX šis vienums nesniedz detalizētu informāciju par loopback interfeisiem (piem., lo0).

Piemēri:

```
net.if.out[eth0]
net.if.out[eth0,errors]
```

net.if.total[if,<mode>]

 Tīkla interfeisa ienākošās un izejošās datplūsmas statistikas summa.
 Atgrieztā vērtība: *Integer*.
 Atbalstītās platformas: Linux, FreeBSD, Solaris[5], HP-UX, AIX, MacOS X, OpenBSD, NetBSD, Windows. NetBSD ir nepieciešamas root tiesības.

Parametri:

- **if** - tīkla interfeisa nosaukums (Unix); tīkla interfeisa pilns apraksts vai IPv4 adrese; vai, ja iekļauts iekavās, tīkla interfeisa 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)* - interfeisā konstatēto sadursmju skaits;
  - *compressed* - ierīces draivera nosūtīto vai saņemto saspiesto pakešu skaits.

Komentāri:

- Varat izmantot šo atslēgu kopā ar pirmapstrādes soli *Change per second*, lai iegūtu baitu sekundē statistiku.
- Režīms *dropped* ir atbalstīts tikai Linux, HP-UX. Nomestās paketes tiek atbalstītas tikai tad, ja jūsu platformā gan `net.if.in`, gan `net.if.out` darbojas ar nomestajām paketēm.
- Režīmi *overruns*, *collision*, *compressed* ir atbalstīti tikai Linux.
- HP-UX šis vienums nesniedz detalizētu informāciju par loopback interfeisiem (piem., 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 jaunākos informācija par TCP ligzdām, kas klausās, tiek iegūta no kodola NETLINK saskarnes, ja iespējams. Pretējā gadījumā informācija tiek iegūta no `/proc/net/tcp` un `/roc/net/tcp6` failiem.

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 (noklusējums 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).

Piezīmes:

- Šīs pārbaudes var izraisīt papildu ziņojumus sistēmas dēmonu žurnālfailos (parasti tiek reģistrētas SMTP un SSH sesijas).
- Šifrētu protokolu pārbaude (piemēram, IMAP uz porta 993 vai POP uz porta 995) pašlaik netiek atbalstīta. Kā risinājumu, lūdzu, izmantojiet `net.tcp.port []` šādām pārbaudēm.
- LDAP un HTTPS pārbaude Windows vidē tiek atbalstīta tikai ar Zabbix aģents 2.
- Telnet pārbaude meklē pieteikšanās uzvedni (':' rindas beigās).

Piemērs:

```
net.tcp.service[ftp,,45] #this item can be used to test the availability of FTP server on TCP port 45
```

net.tcp.service.perf[service,<ip>,<port>]

 Pārbauda TCP pakalpojuma veiktspēju.
 Atgrieztās vērtības: *Float* (0 - pakalpojums nedarbojas; sekundes - sekunžu skaits, kas pavadīts, gaidot atbildi no pakalpojuma).
 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).

Piezīmes:

- Šobrīd šifrētu protokolu pārbaude (piemēram, IMAP uz porta 993 vai POP uz porta 995) netiek atbalstīta. Kā risinājumu, lūdzu, izmantojiet `net.tcp.service.perf[tcp,<ip>,<port>]` šādām pārbaudēm.
- Telnet pārbaude meklē pieteikšanās uzvedni (':' rindas 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 nedarbojas; 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] #this item can be used to test the availability of NTP service on UDP port 45
```

net.udp.service.perf[service,<ip>,<port>]

 Pārbauda UDP pakalpojuma veiktspēju.
 Atgrieztās vērtības: *Float* (0 - pakalpojums nedarbojas; sekundes - sekunžu skaits, kas pavadīts, gaidot atbildi no pakalpojuma).
 Skatiet atbalstītās platformas.

Parametri:

- **service** - *ntp* (skatiet detalizētu informāciju);
- **ip** - IP adrese vai DNS nosaukums (pēc noklusējuma tiek izmantots 127.0.0.1);
- **port** - porta numurs (pēc noklusējuma tiek izmantots standarta pakalpojuma porta numurs).

Piemērs:

```
net.udp.service.perf[ntp] #this item can be used to test response time from NTP service
```

net.udp.socket.count[<laddr>,<lport>,<raddr>,<rport>,<state>]

 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 procentuālā vērtība.
 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 (tas 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 procentuālo vērtību. Piemēram, procesa CPU izmantošana, kas pilnībā izmanto divus kodolus, ir 200%.
- Procesa CPU izmantošanas dati tiek iegūti ar savācēju, kas atbalsta ne vairāk kā 1024 unikālus (pēc nosaukuma, lietotāja un komandrindas) vaicājumus. Vaicājumi, kuri nav izmantoti pēdējo 24 stundu laikā, tiek noņemti no savācēja.
- Ja `zone` parametrs ir iestatīts uz *current* (vai noklusējumu) gadījumā, kad aģents ir kompilēts uz Solaris bez zonu atbalsta, bet darbojas jaunākā Solaris versijā, kur zonas tiek atbalstītas, aģents atgriezīs NOTSUPPORTED (aģents nevar ierobežot rezultātus tikai uz pašreizējo zonu). Tomēr šajā gadījumā *all* tiek atbalstīts.

Piemēri:

```
proc.cpu.util[,root] #CPU izmantošana visiem procesiem, kas darbojas zem "root" lietotāja
proc.cpu.util[zabbix_server,zabbix] #CPU izmantošana visiem zabbix_server procesiem, kas darbojas zem zabb
```

proc.get[<name>,<user>,<cmdline>,<mode>]

 OS procesu un to parametru saraksts. Var izmantot zemlīmeņa 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 (tas ir regulāra izteiksme). Šis parametrs nav atbalstīts Windows; citās platformās tas nav atbalstīts, ja mode ir iestatīts uz 'summary'.
- **mode** - iespējamās vērtības:
  - *process* (noklusējums);
  - *thread* (nav atbalstīts NetBSD);
  - *summary*. Skatiet katram režīmam un OS atgriezto procesu parametru sarakstu.

Komentāri:

- Ja vērtību nevar izgūt, piemēram, kļūdas dēļ (process jau ir beidzis darbību, trūkst atļauju, sistēmas izsaukuma kļūme), tiks atgriezts −1.
- Skatiet piezīmes par procesu atlasi ar `name` un `cmdline` parametriem (tikai Linux).

Piemēri:

```
proc.get[zabbix_server,zabbix,,process] #visu zabbix_server procesu saraksts, kas darbojas zem zabbix liet
proc.get[java,,,thread] #visu Java procesu saraksts, atgriež vienu ierakstu uz katru pavedienu
proc.get[,zabbix,,summary] #apkopotie dati par katra tipa procesiem, kas darbojas zem zabbix lietotāja, at
```

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 *all processes*);
- **user** - lietotāja vārds (noklusējums ir *all users*);
- **mode** - iespējamās vērtības: *avg*, *max*, *min* vai *sum* (noklusējums);
- **cmdline** - filtrēšana pēc komandrindas (tā ir regulāra izteiksme);
- **memtype** - procesā izmantotās atmiņas tips

Komentāri:

- memtype parametrs tiek atbalstīts tikai Linux, FreeBSD, Solaris[6], AIX.
- Ja vairāki procesi izmanto koplietojamo atmiņu, procesu izmantotās atmiņas summa var radīt lielas, nereālistiskas vērtības.

Skatiet piezīmes par procesu atlasi ar name un cmdline parametriem (tikai Linux).

Ja šis vienums tiek izsaukts no komandrindas un satur komandrindas parametru (piem., izmantojot aģents testa režīmu: zabbix_agentd -t proc.mem[,,,apache2]), tiks ieskaitīts viens papildu process, jo aģents skaitīs arī pats sevi.

Piemēri:

```
proc.mem[,root] #atmiņa, ko izmanto visi procesi, kas darbojas zem "root" lietotāja
proc.mem[zabbix_server,zabbix] #atmiņa, ko izmanto visi zabbix_server procesi, kas darbojas zem zabbix lie
proc.mem[,oracle,max,oracleZABBIX] #atmiņa, ko izmanto process ar vislielāko atmiņas patēriņu, kas darboja
```

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 *all processes*).
- **user** - lietotāja vārds (noklusējums ir *all users*).
- **state** - iespējamās vērtības:
  - *all* (noklusējums);
  - *disk* - nepārtraucams miegs;
  - *run* - darbojas;
  - *sleep* - pārtraucams miegs;
  - *trace* - apturēts;
  - *zomb* - zombijs.
- **cmdline** - filtrē pēc komandrindas (tā ir regulāra izteiksme).
- **zone** - mērķa zona: *current* (noklusējums) vai *all*. Šis parametrs tiek atbalstīts tikai Solaris.

Komentāri:

- *disk* un *trace* stāvokļa parametri tiek atbalstīti tikai Linux, FreeBSD, OpenBSD, NetBSD.
- Windows tiek atbalstīti tikai name un user parametri.
- Ja šis vienums tiek izsaukts no komandrindas un satur komandrindas parametru (piem., izmantojot aģenta testa režīmu: zabbix_agentd -t proc.num[,,,apache2]), tiks ieskaitīts vēl viens process, jo aģents ieskaitīs pats sevi.
- Ja, iestatot zone parametru uz current (vai noklusējumu), gadījumā, kad aģents ir kompilēts Solaris versijā bez zonu atbalsta, bet darbojas jaunākā Solaris versijā, kur zonas tiek atbalstītas, aģents atgriezīs NOTSUPPORTED (aģents nevar ierobežot rezultātus tikai līdz pašreizējai zonai). Tomēr šajā gadījumā tiek atbalstīts *all*.
- Skatiet piezīmes par procesu atlasi ar name un cmdline parametriem (tikai Linux).

Piemēri:

```
proc.num[,mysql] #procesu skaits, kas darbojas zem mysql lietotāja
proc.num[apache2,www-data] #apache2 procesu skaits, kas darbojas zem www-data lietotāja
proc.num[,oracle,sleep,oracleZABBIX] #procesu skaits sleep stāvoklī, kas darbojas zem Oracle un kuru koman
```

sensor[device,sensor,<mode>]

 Aparatūras sensora nolasījums.
 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 sistēmās.
- Nolasa /sys/class/hwmon Linux 2.6+ sistēmās. Skatiet detalizētāku sensor vienuma aprakstu Linux vidē.
- Nolasa hw.sensors MIB OpenBSD sistēmās.
- Linux 2.4 sistēmās, ja norādāt mode, tad device un sensor tiek apstrādāti kā regulāras izteiksmes.
- Linux 2.6+ un OpenBSD sistēmās, vai ja mode nav norādīts, device un sensor tiek apstrādāti burtiski.

Piemēri:

```
sensor[w83781d-i2c-0-2d,temp1]
sensor[cpu0,temp0] #viena CPU temperatūra
sensor["cpu[0-2]$",temp,avg] #pirmo trīs CPU vidējā temperatūra
```

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 Windows);
  - *host* (noklusējums 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);
  - *lower* (pārveido uz mazajiem burtiem).

Vērtība tiek iegūta, no uname() sistēmas API izvades paņemot `nodename`.

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 uzskaitījums.
 Atgrieztā vērtība: *String*.
 Atbalstītās platformas: Linux.

Parametri:

- **interface** - *all* (noklusējums) vai regulāra izteiksme;
- **format** - *full* (noklusējums) vai *short*.

Komentāri:

- Uzskaita to interfeisu MAC adreses, kuru nosaukums atbilst norādītajai `interface` regulārajai izteiksmei (*all* uzskaita visiem interfeisiem).
- Ja `format` ir norādīts kā *short*, interfeisu nosaukumi un identiskas 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 Unix ēras sākuma (00:00:00 UTC, 1970. gada 1. janvāris), mērīts sekundēs;
  - *local* - laiks formātā 'yyyy-mm-dd,hh:mm:ss.nnn,+hh:mm'.

Jāizmanto kā pasīva pārbaude Zabbix aģentam; Zabbix aģents 2 to var konfigurēt kā aktīvu pārbaudi.

Piemērs:

```
system.localtime[local] #izveidojiet vienumu, izmantojot šo atslēgu, un pēc tam izmantojiet to, lai parādī
```

system.run[command,<mode>]

 Izpilda norādīto komandu uz hosta.
 Atgrieztā vērtība: *Text* komandas rezultāts vai 1 - ar `mode` kā *nowait* (neatkarīgi no komandas rezultāta).
 Skatiet atbalstītās platformas.

Parametri:

- **command** - komanda izpildei;
- **mode** - iespējamās vērtības:
  - *wait* - gaidīt izpildes beigas (noklusējums);
  - *nowait* - negaidīt.

Piezīmes:

- Pēc noklusējuma šis vienums ir atspējots. Uzziniet, kā to iespējot.
- Vienuma atgrieztā vērtība ir standarta izvade kopā ar standarta kļūdu, ko rada komanda. Izpildes koda pārbaude netiek veikta.
- Lai to apstrādātu pareizi, komandas atgrieztajai vērtībai jābūt `text` datu tipam. Tukšs rezultāts arī ir atļauts.
- Atgrieztā vērtība ir ierobežota līdz 16 MB (ieskaitot beigu atstarpes, kas tiek saīsinātas); 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, uz kuru šai nodalījumam ir tiesības (float).
- **kthr,<type>** - informācija par kodola pavedienu stāvokļiem:
  - *r* - izpildāmo kodola pavedienu vidējais skaits (float);
  - *b* - kodola pavedienu vidējais skaits, kas ievietoti Virtual Memory Manager gaidīšanas rindā (float).
- **memory,<type>** - informācija par virtuālās un reālās atmiņas izmantošanu:
  - *avm* - aktīvās virtuālās lapas (integer);
  - *fre* - brīvā saraksta lielums (integer).

- **page,<type>** - informācija par lapu kļūdām un lapošanas aktivitāti:
  - *fi* - failu lapu ielādes sekundē (float);
  - *fo* - failu lapu izlādes sekundē (float);
  - *pi* - lapas, kas ielādētas no lapošanas vietas (float);
  - *po* - lapas, kas izvadītas uz lapošanas vietu (float);
  - *fr* - atbrīvotās lapas (lapu aizstāšana) (float);
  - *sr* - lapas, ko skenējis lapu aizstāšanas algoritms (float).
- **faults,<type>** - pārtraukumu un trapu 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, kurā sistēmai bija neizpildīti diska/NFS I/O pieprasījumi (float);
  - *pc* - izmantoto fizisko procesoru skaits (float);
  - *ec* - izmantotās piešķirtās jaudas procentuālā daļa (float);
  - *lbusy* - norāda loģisko procesoru izmantošanas procentuālo daļu, kas notikusi, izpildot lietotāja un sistēmas līmenī (float);
  - *app* - norāda pieejamos fizikos procesorus koplietotajā pūlā (float).
- **disk,<type>** - diska statistika:
  - *bps* - norāda uz ierīci pārsūtīto datu apjomu (lasīto vai rakstīto) baitos sekundē (integer);
  - *tps* - norāda pārsūtījumu skaitu sekundē, kas tika nosūtīti uz fizisko disku/lenti (float).

Komentāri:

- Ņemiet vērā šādus ierobežojumus šajos vienumos:
  - `system.stat[cpu,app]` - atbalstīts tikai AIX LPAR tipa ”Shared”;
  - `system.stat[cpu,ec]` - atbalstīts AIX LPAR tipa ”Shared” un ”Dedicated” (”Dedicated” vienmēr atgriež 100 (procentus));
  - `system.stat[cpu,lbusy]` - atbalstīts tikai AIX LPAR tipa ”Shared”;
  - `system.stat[cpu,pc]` - atbalstīts AIX LPAR tipa ”Shared” un ”Dedicated”;
  - `system.stat[ent]` - atbalstīts AIX LPAR tipa ”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* (noklusējuma), *short* vai *name*

Informācija tiek iegūta no (ņemiet vērā, ka ne visi faili un opcijas ir pieejami visos distributīvos):

- `/proc/version` (*full*) Linux sistēmās;
- `/proc/version_signature` (*short*) Linux sistēmās;
- PRETTY_NAME parametrs no `/etc/os-release` Linux sistēmās, kas to atbalsta, vai `/etc/issue.net` (*name*);
- reģistra atslēgas `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: *Text*.
 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):
  - pkg (`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 šādus atslēgvārdus: name, manager, version, size, architecture, buildtime un installtime (skatiet sīkāku informāciju).

system.swap.in[<device>,<type>]

 Swap-in (no ierīces atmiņā) statistika.
 Atgrieztā vērtība: *Integer*.
 Atbalstītās platformas: Linux, FreeBSD, OpenBSD.

Parametri:

- **device** - norādiet ierīci, kas tiek izmantota swap operācijām (tikai Linux) vai *all* (noklusējums);
- **type** - iespējamās vērtības:
  - *count* (swap-in skaits, noklusējums platformās, kas nav Linux);
  - *sectors* (swap-in sektori);
  - *pages* (swap-in 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 device nav norādīts.
- *sectors* tipa parametrs tiek atbalstīts tikai Linux.

Piemērs:

```
system.swap.in[,pages]
```

system.swap.out[<device>,<type>]

 Maiņas izrakstīšanas (no atmiņas uz ierīci) statistika.
 Atgrieztā vērtība: *Integer*.
 Atbalstītās platformas: Linux, FreeBSD, OpenBSD.

Parametri:

- **device** - norādiet ierīci, kas tiek izmantota maiņai (tikai Linux) vai *all* (noklusējums);
- **type** - iespējamās vērtības:

  - *count* (maiņas izrakstīšanas reižu skaits, noklusējums platformās, kas nav Linux);
  - *sectors* (izrakstītie sektori);
  - *pages* (izrakstītās 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 device nav norādīts.
- Parametrs *sectors* tiek atbalstīts tikai Linux.

Piemērs:

```
system.swap.out[,pages]
```

system.swap.size[<device>,<type>]

 Maiņas vietas lielums baitos vai procentos no kopējā apjoma.
 Atgrieztā vērtība: *Integer* - baitos; *Float* - procentos.
 Atbalstītās platformas: Linux, FreeBSD, Solaris, AIX, OpenBSD, Windows.

Parametri:

- **device** - norādiet ierīci, ko izmanto maiņai (tikai FreeBSD) vai *all* (noklusējums);
- **type** - iespējamās vērtības:
  - *free* (brīva maiņas vieta, noklusējums);
  - *pfree* (brīva maiņas vieta, procentos);
  - *pused* (izmantota maiņas vieta, procentos);
  - *total* (kopējā maiņas vieta);
  - *used* (izmantota maiņas vieta).

Piezīmes:

- Ņemiet vērā, ka *pfree*, *pused* netiek atbalstīti Windows, ja maiņas vietas lielums ir 0.
- Ja device nav norādīts, Zabbix aģents ņems vērā tikai maiņas ierīces (failus), fiziskā atmiņa tiks ignorēta. Piemēram, Solaris sistēmās komanda `swap -s` ietver daļu fiziskās atmiņas un maiņas ierīces (atšķirībā no `swap -l`).

Piemērs:

```
system.swap.size[,pfree] → brīvās maiņas 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.get[<mode>,<regexp>]

 Bloku ierīču saraksts ar detalizētu informāciju. Tiek izmantots zemā līmeņa atklāšanai.
 Atgrieztā vērtība: *JSON virkne*.
 Atbalstītās platformas: Linux, AIX.

Parametri:

- **mode** - iespējamās vērtības:
  - *disks* (noklusējums) - atgriež disku sarakstu;
  - *disk_stats* - atgriež statistiku par katru disku;
  - *devices* (tikai Linux) - atgriež bloku ierīču sarakstu (diski un nodalījumi);
  - *device_stats* (tikai Linux) - atgriež statistiku par katru ierīci (diski un nodalījumi).
- **regexp** - regulāra izteiksme, ko izmanto bloku ierīču filtrēšanai pēc to nosaukuma.

vfs.dev.read[<device>,<type>,<mode>]

 Diska lasīšanas 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* nozīmē attiecīgi: 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.
- *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.
- Parametrs `mode` tiek atbalstīts tikai Linux, FreeBSD.
- Var izmantot gan relatīvus ierīču nosaukumus (piemēram, `sda`), gan arī izvēles `/dev/` prefiksu (piemēram, `/dev/sda`).
- LVM loģiskie sējumi tiek atbalstīti.
- Noklusējuma `type` parametra 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 atsevišķām un viena *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* nozīmē attiecīgi: 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.

Piezīmes:

- Ja tiek izmantots atjaunināšanas intervāls trīs stundas vai vairāk[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.
- Parametrs `mode` tiek atbalstīts tikai Linux, FreeBSD.
- Var izmantot gan relatīvus ierīču nosaukumus (piemēram, `sda`), gan arī izvēles `/dev/` prefiksu (piemēram, `/dev/sda`).
- LVM loģiskie sējumi tiek atbalstīti.
- Noklusējuma `type` parametra 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 atsevišķām 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, direktorijas, simboliskās saites) nosaukuma modeli; ja vērtība ir tukša, tiek iekļautas visas entītijas (noklusējuma vērtība);
- **regex excl** - regulāra izteiksme, kas apraksta izslēdzamās entītijas (faila, direktorijas, simboliskās saites) nosaukuma modeli; ja vērtība ir tukša, nekas netiek izslēgts (noklusējuma vērtība);
- **types incl** - skaitāmo direktorijas ierakstu tipi; iespējamās vērtības:
  - *file* - parasts fails;
  - *dir* - apakšdirektorija;
  - *sym* - simboliskā saite;
  - *sock* - ligzda;
  - *bdev* - bloka ierīce;
  - *cdev* - rakstzīmju ierīce;
  - *fifo* - FIFO;
  - *dev* - sinonīms vērtībām "bdev,cdev";
  - *all* - visi tipi (noklusējuma vērtība), t. i., "file,dir,sym,sock,bdev,cdev,fifo". Vairāki tipi jāatdala ar komatu un jāiekļauj pēdiņās.
- **types excl** - direktorijas ierakstu tipi (skatiet `types incl`), kurus NEIESKAITA. Ja kāds ieraksta tips ir norādīts gan **types incl**, gan **types excl**, šāda tipa direktorijas ieraksti NETIEK ieskaitīti.
- **max depth** - maksimālais apakšdirektoriju dziļums, līdz kuram veikt traversēšanu:
  - **-1** (noklusējuma vērtība) - neierobežots;
  - **0** - neveikt traversēšanu apakšdirektorijās.
- **min size** - minimālais faila izmērs (baitos), lai fails tiktu ieskaitīts. Mazāki faili netiks ieskaitīti. Var izmantot atmiņas sufiksus.
- **max size** - maksimālais faila izmērs (baitos), lai fails tiktu ieskaitīts. Lielāki faili netiks ieskaitīti. Var izmantot atmiņas sufiksus.
- **min age** - minimālais direktorijas ieraksta vecums (sekundēs), lai tas tiktu ieskaitīts. Jaunāki ieraksti netiks ieskaitīti. Var izmantot laika sufiksus.
- **max age** - maksimālais direktorijas ieraksta vecums (sekundēs), lai tas tiktu ieskaitīts. Ieraksti, kas ir tikpat veci vai vecāki, netiks ieskaitīti (pēc modificēšanas laika). Var izmantot laika sufiksus.
- **regex excl dir** - regulāra izteiksme, kas apraksta izslēdzamās direktorijas nosaukuma modeli. Viss direktorijas saturs tiks izslēgts (atšķirībā no `regex_excl`).

Komentāri:

- Operētājsistēmā Windows šis vienums atbalsta UNC ceļus, kas var radīt drošības risku.
- Vides mainīgie, piemēram, `%APP_HOME%`, `$HOME` un `%TEMP%`, netiek atbalstīti.
- Pseidodirektorijas "." un ".." nekad netiek ieskaitītas.
- Simboliskās saites nekad netiek izmantotas direktoriju traversēšanai.
- Gan `regex incl`, gan `regex excl` tiek lietotas failiem un direktorijām, aprēķinot ierakstu skaitu, taču tās tiek ignorētas, izvēloties traversējamās apakšdirektorijas (ja `regex incl` ir `(?i)^.+\.zip$` un `max depth` nav iestatīts, tiks traversētas visas apakšdirektorijas, bet tiks ieskaitīti tikai zip tipa faili).
- Izpildes laiku ierobežo taimauta vērtība aģenta konfigurācijā (3 sek.). Tā kā lielas direktoriju struktūras traversēšana var aizņemt vairāk laika, dati netiks atgriezti un vienums kļūs neatbalstīts. Daļējs skaits netiks atgriezts.
- Filtrējot pēc izmēra, nozīmīgs izmērs ir tikai parastajiem failiem. Linux un BSD sistēmās direktorijām arī ir izmērs, kas nav nulle (parasti daži KB). Ierīču 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 neparedzētiem 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, direktorijas, simboliskās saites) nosaukuma modeli; ja vērtība ir tukša, tiek iekļautas visas entītijas (noklusējuma vērtība);
- **regex excl** - regulāra izteiksme, kas apraksta izslēdzamās entītijas (faila, direktorijas, simboliskās saites) nosaukuma modeli; ja vērtība ir tukša, nekas netiek izslēgts (noklusējuma vērtība);
- **types incl** - uzskaitāmo direktorijas ierakstu tipi; iespējamās vērtības:
  - *file* - parasts fails;
  - *dir* - apakšdirektorija;
  - *sym* - simboliskā saite;
  - *sock* - ligzda;
  - *bdev* - blokierīce;
  - *cdev* - rakstzīmierīce;
  - *fifo* - FIFO;
  - *dev* - sinonīms vērtībai "bdev,cdev";
  - *all* - visi tipi (noklusējuma vērtība), t. i., "file,dir,sym,sock,bdev,cdev,fifo". Vairāki tipi jāatdala ar komatu un jāiekļauj pēdiņās.
- **types excl** - direktorijas ierakstu tipi (skatiet `types incl`), kurus NEUZSKAITĪT. Ja kāds ieraksta tips ir norādīts gan `types incl`, gan `types excl`, šī tipa direktorijas ieraksti NETIEK uzskaitīti.
- **max depth** - maksimālais apakšdirektoriju dziļums, kurā veikt pārlūkošanu:
  - **-1** (noklusējuma vērtība) - neierobežots;
  - **0** - neveikt pārlūkošanu apakšdirektorijās.
- **min size** - minimālais faila izmērs (baitos), lai fails tiktu uzskaitīts. Mazāki faili netiks uzskaitīti. Var izmantot atmiņas sufiksus.
- **max size** - maksimālais faila izmērs (baitos), lai fails tiktu uzskaitīts. Lielāki faili netiks uzskaitīti. Var izmantot atmiņas sufiksus.
- **min age** - minimālais direktorijas ieraksta vecums (sekundēs), lai ieraksts tiktu uzskaitīts. Jaunāki ieraksti netiks uzskaitīti. Var izmantot laika sufiksus.
- **max age** - maksimālais direktorijas ieraksta vecums (sekundēs), lai ieraksts tiktu uzskaitīts. Ieraksti, kas ir tikpat veci vai vecāki, netiks uzskaitīti (pēc modificēšanas laika). Var izmantot laika sufiksus.
- **regex excl dir** - regulāra izteiksme, kas apraksta izslēdzamās direktorijas nosaukuma modeli. Viss direktorijas saturs tiks izslēgts (atšķirībā no `regex excl`).

Piezīmes:

- Operētājsistēmā Windows šis vienums seko UNC ceļiem, kas var radīt drošības risku.
- Vides mainīgie, piemēram, `%APP_HOME%`, `$HOME` un `%TEMP%`, netiek atbalstīti.
- Pseidodirektorijas "." un ".." nekad netiek uzskaitītas.
- Simboliskās saites direktoriju pārlūkošanas laikā nekad netiek izmantotas.
- Gan `regex incl`, gan `regex excl` tiek lietotas failiem un direktorijām, ģenerējot ierakstu sarakstu, taču tās tiek ignorētas, izvēloties pārlūkojamās apakšdirektorijas (ja `regex incl` ir `(?i)^.+\.zip$` un `max depth` nav iestatīts, tiks pārlūkotas visas apakšdirektorijas, bet tiks uzskaitīti tikai zip tipa faili).
- Izpildes laiku ierobežo taimauta vērtība aģenta konfigurācijā. Tā kā lielas direktoriju struktūras pārlūkošana var aizņemt vairāk laika, dati netiks atgriezti un vienums kļūs neatbalstīts. Daļējs saraksts netiks atgriezts.
- Filtrējot pēc izmēra, nozīmīgs izmērs ir tikai parastiem failiem. Linux un BSD sistēmās arī direktorijām ir izmērs, kas nav nulle (parasti daži KB). Ierīču 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 neparedzētiem 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>]

 Direktorijas izmērs (baitos).
 Atgrieztā vērtība: *Integer*.
 Atbalstītās platformas: Linux, Windows. Vienums var darboties arī citās UNIX līdzīgās platformās.

Parametri:

- **dir** - absolūtais ceļš uz direktoriju;
- **regex incl** - regulāra izteiksme, kas apraksta iekļaujamā objekta (fails, direktorija, simboliskā saite) nosaukuma rakstu; ja tukšs, iekļauj visus (noklusējuma vērtība);
- **regex excl** - regulāra izteiksme, kas apraksta izslēdzamā objekta (fails, direktorija, simboliskā saite) nosaukuma rakstu; ja tukšs, neizslēdz nevienu (noklusējuma vērtība);
- **mode** - iespējamās vērtības:
  - *apparent* (noklusējums) - iegūst failu šķietamo izmēru, 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 direktorijas izmēru, ņem vērā arī slēptos failus (darbojas kā `du -sb .[^.]* *` direktorijā).
- **max depth** - maksimālais apakšdirektoriju dziļums, ko traversēt:
  - **-1** (noklusējums) - neierobežots;
  - **0** - neieiet apakšdirektorijās.
- **regex excl dir** - regulāra izteiksme, kas apraksta izslēdzamās direktorijas nosaukuma rakstu. Visa direktorijas satura apstrāde tiks izslēgta (atšķirībā no `regex excl`)

Komentāri:

- Operētājsistēmā Windows šis vienums seko UNC ceļiem, kas var radīt drošības risku.
- Tiek aprēķinātas tikai tās direktorijas, kurām lietotājam *zabbix* ir vismaz lasīšanas atļauja. Direktorijām tikai ar lasīšanas atļauju tiek aprēķināts pašas direktorijas izmērs. Direktorijas ar lasīšanas un izpildes atļaujām tiek aprēķinātas, iekļaujot saturu.
- Lielām direktorijām vai lēniem diskiem šis vienums var pārsniegt noildzi aģents un serveris/starpniekserveris konfigurācijas failos norādītā Timeout iestatījuma dēļ. Palieliniet noildzes vērtības, ja nepieciešams.
- Faila izmēra ierobežojums ir atkarīgs no lielo failu atbalsta.

Piemēri:

```
vfs.dir.size[/tmp,log] #aprēķina visu failu izmēru /tmp, kuru nosaukumos ir 'log'
vfs.dir.size[/tmp,log,^.+\.old$] #aprēķina visu failu izmēru /tmp, kuru nosaukumos ir 'log', izslēdzot fai
```

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** - pilnais ceļš uz failu;
- **mode** - *crc32* (noklusējums), *md5* vai *sha256*.

Komentāri:

- Windows vidē 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 izgūšana[7].
 Atgrieztā vērtība: *Text*.
 Skatiet atbalstītās platformas.

Parametri:

- **file** - pilnais 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 16 MB (ieskaitot beigu atstarpes, kas tiek saīsinātas); attiecas arī datubāzes ierobežojumi.
- Tiek atgriezta tukša virkne, ja fails ir tukšs vai satur tikai LF/CR rakstzīmes.
- 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 - norādītā tipa fails eksistē.
 Skatiet atbalstītās platformas.

Parametri:

- **file** - pilnais ceļš uz failu;
- **types incl** - iekļaujamo failu tipu saraksts, iespējamās vērtības:
  - *file* (parasts fails, noklusējums (ja `types_excl` nav iestatīts));
  - *dir* (direktorija);
  - *sym* (simboliskā saite);
  - *sock* (sokets);
  - *bdev* (bloka 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, skatiet `types_incl` par iespējamām vērtībām (pēc noklusējuma netiek izslēgti nekādi tipi)

Piezīmes:

- Operētājsistēmā Windows šis vienums seko UNC ceļiem, kas var radīt drošības risku.
- Vairāki tipi jāatdala ar komatu, un viss kopums jāievieto pēdiņās "".
- Ja viens un tas pats tips ir gan <types_incl>, gan <types_excl>, šī tipa faili tiek izslēgti.
- Faila izmēra ierobežojums ir atkarīgs no lielo 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:

- Operētājsistēmā Windows šis vienums seko UNC ceļiem, kas var būt drošības risks.
- Atbalstītie failu tipi UNIX līdzīgās sistēmās: parasts fails, direktorijs, simboliskā saite, ligzda, bloka ierīce, rakstzīmju ierīce, FIFO.
- Faila lieluma ierobežojums ir atkarīgs no lielo failu atbalsta.

Piemērs:

```
vfs.file.get[/etc/passwd] #return a JSON with information about the /etc/passwd file (type, user, permissi
```

vfs.file.md5sum[file]

 Faila MD5 kontrolsumma.
 Atgrieztā vērtība: rakstzīmju virkne (faila MD5 jaucējkods).
 Skatiet atbalstītās platformas.

Parametrs:

- **file** - pilnais ceļš uz failu

Komentāri:

- Operētājsistēmā Windows šis vienums seko UNC ceļiem, kas var radīt drošības risku.
- Faila lieluma ierobežojums ir atkarīgs no lielo 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 līdz failam;
- **ownertype** - *user* (noklusējums) vai *group* (tikai Unix);
- **resulttype** - *name* (noklusējums) vai *id*; ja *id* - Unix atgriež uid/gid, Windows - SID.

Komentāri:

- Windows vidē šis vienums seko UNC ceļiem, kas var būt drošības risks.
- Faila lieluma ierobežojums ir atkarīgs no lielo failu atbalsta.

Piemēri:

```
vfs.file.owner[/tmp/zabbix_server.log] #atgriež /tmp/zabbix_server.log faila īpašnieku
vfs.file.owner[/tmp/zabbix_server.log,,id] #atgriež /tmp/zabbix_server.log faila īpašnieka ID
```

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>]

 Iegūst virkni failā[7].
 Atgrieztā vērtība: rinda, kurā ir atrasta atbilstošā virkne, vai, kā norādīts izvēles `output` parametrā.
 Skatiet atbalstītās platformas.

Parametri:

- **file** - pilns faila ceļš;
- **regexp** - regulāra izteiksme, kas apraksta nepieciešamo rakstu;
- **encoding** - kodu lapas identifikators;
- **start line** - pirmās meklējamās rindas numurs (pēc noklusējuma faila pirmā rinda);
- **end line** - pēdējās meklējamās rindas numurs (pēc noklusējuma faila pēdējā rinda);
- **output** - izvēles izvades formatēšanas veidne. Atsoļu secība **\0** tiek aizstāta ar atrasto teksta daļu (no pirmās rakstzīmes, kur sākas atbilstība, līdz rakstzīmei, kur atbilstība beidzas), savukārt **\N** (kur N=1...9) atsoļu secība tiek aizstāta ar N-to atrasto grupu (vai ar 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 lielo failu atbalsta.
- Tiek atgriezta tikai pirmā rinda, kas atbilst.
- Ja neviena rinda neatbilst izteiksmei, tiek atgriezta tukša virkne.
- Baitu secības zīme (BOM) tiek izslēgta no izvades.
- Satura iegūšana, 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] → iegūst lietotāja *zabbix* ID
```

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 rakstu;
- **encoding** - kodu lapas identifikators;
- **start line** - pirmās meklējamās rindas numurs (pēc noklusējuma faila pirmā rinda);
- **end line** - pēdējās meklējamās rindas numurs (pēc noklusējuma faila pēdējā rinda).

Komentāri:

- Operētājsistēmā Windows šis vienums seko UNC ceļiem, kas var radīt drošības risku.
- Faila lieluma ierobežojums ir atkarīgs no lielo failu atbalsta.
- Baitu secības marķieris (BOM) tiek ignorēts.

Piemērs:

```
vfs.file.regmatch[/var/log/app.log,error]
```

vfs.file.size[file,<mode>]

 Faila izmērs (baitos).
 Atgrieztā vērtība: *Integer*.
 Skatiet atbalstītās platformas.

Parametri:

- **file** - pilnais ceļš uz failu;
- **mode** - iespējamās vērtības:
  - *bytes* (noklusējums);
  - *lines* (tiek skaitītas arī tukšās rindas).

Komentāri:

- Operētājsistēmā Windows šis vienums seko UNC ceļiem, kas var radīt drošības risku.
- Failam jābūt ar lasīšanas atļaujām lietotājam *zabbix*.
- Faila izmēra ierobežojums ir atkarīgs no lielo 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** - pilnais ceļš uz failu;
- **mode** - iespējamās vērtības:
  - *modify* (noklusējums) - pēdējais faila satura modificēšanas laiks (noklusējums);
  - *access* - pēdējais faila lasīšanas laiks;
  - *change* - pēdējais faila īpašību maiņas laiks.

Komentāri:

- Operētājsistēmā Windows šis vienums seko UNC ceļiem, kas var radīt drošības risku.
- Faila lieluma ierobežojums ir atkarīgs no lielo 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[<mode>,<mountpoint>]

 Pievienoto failu sistēmu saraksts ar to tipu un pievienošanas opcijām, kā arī *full* režīmā — pieejamās diska vietas un inode statistika. Var izmantot zema līmeņa atklāšanai.
 Atgrieztā vērtība: *JSON virkne*.
 Atbalstītās platformas: Linux, FreeBSD, Solaris, HP-UX, AIX, MacOS X, OpenBSD, NetBSD, Windows.

Parametri:

- **mode** - iespējamās vērtības:
  - *full* (noklusējums) - failu sistēmas rekvizīti ar diska vietas un inode statistiku;
  - *short* - tikai failu sistēmas rekvizīti bez diska vietas un inode statistikas;
- **mountpoint** - pievienošanas punkts, par kuru jāatgriež dati (piemēram, `/var` vai `C:`). Ja nav norādīts, tiek atgrieztas visas pievienotās failu sistēmas.

Komentāri:

- *short* režīmā tiek atgriezts tikai failu sistēmas nosaukums, tips un pievienošanas opcijas (Windows sistēmā: nosaukums, tips, etiķete un diska tips). Failu sistēmām statistika netiek vaicāta; Unix līdzīgās sistēmās tiek nolasīts tikai pievienoto failu sistēmu saraksts (piemēram, `/proc/mounts` Linux sistēmā), tāpēc pārbaude neizraisa automātisku pievienošanu un to neietekmē nereaģējošas tīkla failu sistēmas. Tādēļ *short* režīms ir piemērots zema līmeņa atklāšanai.
- **mountpoint** vērtībai precīzi jāatbilst `fsname` rekvizītam. Windows sistēmā salīdzināšana nav reģistrjutīga; norādiet pievienošanas punktu bez beigu slīpsvītras (piemēram, `C:`).
- Ja **mountpoint** ir norādīts, dati tiek atgriezti tikai par šo pievienošanas punktu. Ja neviena pievienotā failu sistēma tam neatbilst, tiek atgriezts tukšs masīvs (`[]`).
- Izvadē var būt vairāki ieraksti ar vienādu `fsname`, ja vienā direktorijā ir pievienota vairāk nekā viena failu sistēma.
- *full* režīmā pieejamā diska vieta un inode statistika tiek atgriezta tikai tad, ja Zabbix aģenta lietotājam ir caurskates atļauja katrā direktorijā pievienošanas punkta ceļā; pretējā gadījumā failu sistēma tiek izlaista no izvades. Tas neattiecas uz *short* režīmu, kurā tādēļ var tikt uzskaitītas failu sistēmas, kas nav iekļautas *full* režīma izvadē.
- Rekvizīti `fslabel` un `fsdrivetype` tiek atgriezti tikai Windows sistēmā; rekvizīti `inodes` un `options` Windows sistēmā netiek atgriezti.
- Tiek uzrādītas arī failu sistēmas, kuru inode skaits ir vienāds ar nulli, kā tas var būt failu sistēmām ar dinamiskiem inode (piemēram, btrfs).
- Skatiet arī: Pievienoto failu sistēmu atklāšana.

Piemēri:

```
vfs.fs.get #visas pievienotās failu sistēmas ar statistiku; tas pats, kas vfs.fs.get[full]
vfs.fs.get[short] #visas pievienotās failu sistēmas bez statistikas
vfs.fs.get[full,/var] #statistika tikai par /var pievienošanas punktu
vfs.fs.get[short,C:] #tikai C: sējuma rekvizīti (Windows)
```

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>,<redirect_limit>]

 Iegūt tīmekļa lapas saturu.
 Atgrieztā vērtība: tīmekļa lapas avots kā teksts (tostarp galvenes).
 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, tiks izmantots *http*. Ja ir norādīts URL, `path` un `port` ir jāatstāj tukši. Lietotājvārdu/paroli, pieslēdzoties serveriem, kuriem nepieciešama autentifikācija, piemēram, `http://user:password@www.example` var norādīt tikai tad, ja ir pieejams cURL atbalsts [4]. Punycode tiek atbalstīts resursdatoru nosaukumos.
- **path** - ceļš uz HTML dokumentu (noklusējums ir /);
- **port** - porta numurs (noklusējums ir 80 HTTP gadījumā);
- **redirect_limit** - maksimālais sekojamo novirzīšanu skaits [4], no 0 līdz 1000 (noklusējums ir 0 - novirzīšanām netiek sekots).

Komentāri:

- Šis vienums kļūst neatbalstīts, ja `host` norādītais resurss neeksistē vai nav pieejams, vai ja `redirect_limit` ir lielāks par 0 un novirzīšanu skaits pārsniedz šo vērtību.
- `host` var būt resursdatora nosaukums, domēna nosaukums, IPv4 vai IPv6 adrese. Tomēr IPv6 adresēm Zabbix aģentam jābūt kompilētam ar iespējotu IPv6 atbalstu.
- Ja novirzīšanām tiek sekots, atgrieztā vērtība satur katras novirzīšanas atbildes galvenes, kam seko galīgās atbildes galvenes un pamatteksts.

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.get[http://www.example.com,,,5]
```

web.page.perf[host,<path>,<port>,<redirect_limit>]

 Pilnas tīmekļa lapas ielādes laiks (sekundēs).
 Atgrieztā vērtība: *Float*.
 Skatiet atbalstītās platformas.

Parametri:

- **host** - resursdatora nosaukums vai URL (formātā `scheme://host:port/path`, kur tikai *host* ir obligāts). Atļautās URL shēmas: *http*, *https*[4]. Ja shēma nav norādīta, tiks izmantots *http*. Ja ir norādīts URL, `path` un `port` jāatstāj tukši. Lietotājvārdu/paroli, pieslēdzoties serveriem, kuriem nepieciešama autentifikācija, piemēram, `http://user:password@www.example` var norādīt tikai tad, ja ir pieejams cURL atbalsts [4]. Resursdatoru nosaukumos tiek atbalstīts Punycode.
- **path** - ceļš uz HTML dokumentu (noklusējums ir /);
- **port** - porta numurs (noklusējums ir 80 HTTP gadījumā);
- **redirect_limit** - maksimālais sekojamo novirzīšanu skaits [4], no 0 līdz 1000 (noklusējums ir 0 - novirzīšanas netiek izpildītas).

Komentāri:

- Šis vienums kļūst neatbalstīts, ja `host` norādītais resurss neeksistē vai nav pieejams, vai ja `redirect_limit` ir lielāks par 0 un novirzīšanu skaits pārsniedz šo vērtību.
- `host` var būt resursdatora nosaukums, domēna nosaukums, IPv4 vai IPv6 adrese. Tomēr IPv6 adresēm Zabbix aģentam jābūt kompilētam ar iespējotu IPv6 atbalstu.

- Ja novirzīšanas tiek izpildītas, ielādes laiks ietver visu novirzīšanu izpildei patērēto laiku.

Piemēri:

```
web.page.perf[www.example.com,index.php,80]
web.page.perf[https://www.example.com]
web.page.perf[http://www.example.com,,,5]
```

web.page.regexp[host,<path>,<port>,regexp,<length>,<output>,<redirect_limit>]

 Atrod virkni tīmekļa lapā.
 Atgrieztā vērtība: atrastā virkne vai virkne, kas norādīta neobligātajā `output` parametrā.
 Skatiet atbalstītās platformas.

Parametri:

- **host** - resursdatora nosaukums vai URL (formātā `scheme://host:port/path`, kur obligāts ir tikai *host*). Atļautās URL shēmas: *http*, *https*[4]. Ja shēma nav norādīta, tiks izmantots *http*. Ja ir norādīts URL, parametriem `path` un `port` jābūt tukšiem. Lietotājvārdu/paroli, pieslēdzoties serveriem, kuriem nepieciešama autentifikācija, piemēram, `http://user:password@www.example.com`, var norādīt tikai tad, ja ir pieejams cURL atbalsts [4]. Saimniekdatoru 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 rakstu.
- **length** - atgriežamo rakstzīmju maksimālais skaits;
- **output** - neobligāta izvades formatēšanas veidne. **\0** atsoļa secība tiek aizstāta ar atrasto teksta daļu (no pirmās rakstzīmes, kur sākas atbilstība, līdz rakstzīmei, kur atbilstība beidzas), savukārt **\N** (kur N=1...9) atsoļa secība tiek aizstāta ar N-to atrasto grupu (vai tukšu virkni, ja N pārsniedz uztverto grupu skaitu).
- **redirect_limit** - maksimālais sekojamo novirzīšanu skaits [4], no 0 līdz 1000 (noklusējums ir 0 - novirzīšanām netiek sekots).

Komentāri:

- Šis vienums kļūst neatbalstīts, ja `host` norādītais resurss neeksistē vai nav pieejams, vai ja `redirect_limit` ir lielāks par 0 un novirzīšanu skaits pārsniedz šo vērtību.
- `host` var būt saimniekdatora nosaukums, domēna nosaukums, IPv4 vai IPv6 adrese. Tomēr IPv6 adresēm Zabbix aģentam jābūt kompilētam ar iespējotu IPv6 atbalstu.
- Satura izvilkšana, izmantojot parametru `output`, notiek aģentā.
- Ja novirzīšanām tiek sekots, tiek meklētas arī katras novirzīšanas atbildes galvenes.

Piemēri:

```
web.page.regexp[www.example.com,index.php,80,OK,2]
web.page.regexp[https://www.example.com,,,OK,2]
web.page.regexp[http://www.example.com,,,OK,2,,5]
```

aģents.hostmetadata

 Aģenta hosta metadati.
 Atgrieztā vērtība: *String*.
 Skatiet atbalstītās platformas.

Atgriež `HostMetadata` vai `HostMetadataItem` parametru vērtību, vai tukšu virkni, ja neviens no tiem nav definēts.

aģents.hostname

 Aģenta hosts nosaukums.
 Atgrieztā vērtība: *String*.
 Skatiet atbalstītās platformas.

Atgriež:

- Kā pasīvajā pārbaudē - pirmā hosta nosaukumu, kas norādīts aģenta konfigurācijas faila Hostname parametrā;
- Kā aktīvajā pārbaudē - pašreizējā hosta nosaukumu.

agent.ping

 Aģenta pieejamības pārbaude.
 Atgrieztā vērtība: Nekas - nav pieejams; 1 - pieejams.
 Skatiet atbalstītās platformas.

Lai pārbaudītu host nepieejamību, izmantojiet `nodata()` trigera funkciju.

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 metrikas kopu. Ja ir norādīti ip un port, metrikas tiek izgūtas no attālā eksemplāra; pretējā gadījumā - no lokālā eksemplāra.
 Atgrieztā vērtība: *JSON virkne*.
 Skatiet atbalstītās platformas.

Parametri:

- **ip** - serveru/starpniekserveru IP/DNS/tīkla masku saraksts, kam veikt attālo vaicājumu (noklusējums ir 127.0.0.1);
- **port** - servera/starpniekservera ports, kam veikt attālo vaicājumu (noklusējums ir 10051)

Komentāri:

- Ar šo vienumu tiek atgriezts atlasīts iekšējo metrikas kopums. Sīkāku informāciju skatiet sadaļā Zabbix statistikas attālā uzraudzība.
- Ņemiet vērā, ka stats pieprasījums tiks pieņemts tikai no adresēm, kas norādītas mērķa eksemplāra StatsAllowedIP serveris/starpniekserveris parametrā.

zabbix.stats[<ip>,<port>,queue,<from>,<to>]

 Atgriež uzraudzīto vienumu skaitu rindā, kas ir aizkavēti Zabbix serverī vai starpniekserverī. Ja ir norādīti ip un port, metrikas tiek izgūtas no attālās instances; pretējā gadījumā - no lokālās instances.
 Atgrieztā vērtība: *JSON string*.
 Skatiet atbalstītās platformas.

Parametri:

- **ip** - serveru/starpniekserveru IP/DNS/tīkla masku saraksts, kuriem veikt attālinātu vaicājumu (noklusējums ir 127.0.0.1);
- **port** - servera/starpniekservera ports, kuram veikt attālinātu vaicājumu (noklusējums ir 10051)
- **queue** - konstante (jāizmanto tāda, kāda tā ir)
- **from** - aizkavēts vismaz par (noklusējums ir 6 sekundes)
- **to** - aizkavēts ne vairāk kā par (noklusējums ir bezgalība)

Ņemiet vērā, ka stats pieprasījums tiks pieņemts tikai no adresēm, kas norādītas mērķa instances parametra 'StatsAllowedIP' serveris/starpniekserveris.

Zemsvītras piezīmes

[1] Piezīme, kas attiecas tikai uz Linux. Zabbix aģentam ir 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 vairāk nekā 3 stundas netiek piekļūts. 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 tiem izmantots atjaunināšanas intervāls, kas ir 3 stundas vai ilgāks, vienmēr atgriezīs "0".

[3] vfs.dev.read[], vfs.dev.write[]: Ja pirmajam parametram tiek izmantota noklusējuma vērtība all, atslēga atgriezīs kopsavilkuma statistiku, tostarp visu bloku ierīču, piemēram, sda, sdb, un to nodalījumu (sda1, sda2, sdb3...), kā arī vairāku ierīču (MD RAID), kuru pamatā ir šīs bloku ierīces/nodalījumi, un loģisko sējumu (LVM), kuru pamatā ir šīs bloku ierīces/nodalījumi, statistiku. Šādos gadījumos atgrieztās vērtības jāuzskata tikai par relatīvām vērtībām (dinamiskām laikā), nevis par absolūtām vērtībām.

[4] HTTPS un sekošana novirzēm (redirect_limit, kas ir lielāks par 0) tiek atbalstīta tikai tad, ja Zabbix aģents ir kompilēts ar cURL atbalstu; pretējā gadījumā vienums kļūst neatbalstīts. Zabbix aģentam 2 cURL nav nepieciešams nevienā no šiem gadījumiem.

[5] Vērtības bytes un errors netiek atbalstītas cilpas saskarnēm Solaris sistēmās līdz Solaris 10 6/06 versijai (ieskaitot), jo kodols nesaglabā un/vai neziņo baitu, kļūdu un izmantošanas statistiku. Tomēr, ja Solaris sistēma tiek pārraudzīta, izmantojot net-snmp, vērtības var tikt atgrieztas, jo net-snmp ietver mantotu kodu no cmu-snmp, kas datēts pat ar 1997. gadu un, nespējot nolasīt baitu vērtības no saskarnes statistikas, atgriež pakešu skaitītāju (kas cilpas 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 sistēmu cilpas saskarņu MTU ierobežojums ir 8892 baiti. Šīs vērtības nevajadzētu uzskatīt par pareizām vai pat aptuveni precīzām. Tās ir aptuvenas aplēses. Zabbix aģents neveic minējumus, taču net-snmp atgriezīs šo lauku vērtību.

[6] Solaris komandrinda, kas iegūta no /proc/pid/psinfo, ir ierobežota līdz 80 baitiem un satur komandrindu tādā formā, kāda tā bija procesa palaišanas brīdī.

[7] vfs.file.contents[], vfs.file.regexp[], vfs.file.regmatch[] vienumus var izmantot failu satura izgūšanai. Ja vēlaties ierobežot piekļuvi konkrētiem failiem ar sensitīvu informāciju, palaidiet Zabbix aģentu ar lietotāju, 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, parametrā `encoding` varat norādīt pareizo kodējumu pārbaudes (piemēram, `vfs.file.contents`) apstrādei. Atbalstīto kodējumu (kodu lapu identifikatoru) saraksts ir pieejams libiconv (GNU Project) dokumentācijā vai Microsoft Windows SDK dokumentācijas sadaļā "Code Page Identifiers". Ņemiet vērā, ka Microsoft dažkārt dažas koda lapas atzīmē kā "available only to managed applications", kas nozīmē, ka tās ir pieejamas tikai, izmantojot .NET izpildlaiku, un nav pieejamas, izmantojot vietējās Win32 API. Zabbix aģents izmanto savu kodējuma pārveidošanas loģiku, tāpēc šīs koda lapas atbalsta Zabbix aģents arī tad, ja vietējās Windows funkcijas tās nenodrošina.

Ja parametrā `encoding` nav norādīts kodējums, tiek lietotas 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 baitu secības marķieri (BOM) faila sākumā. Ja BOM nav pieejams, tā vietā tiek lietota standarta noteikšana (skatiet iepriekš).

Aģenta vienumu problēmu novēršana

Pasīvo pārbaužu gadījumā, lai novērstu situāciju, kad vienums nesaņem vērtību, jo servera pieprasījumam aģentam 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 nekā `Timeout` vērtībai aģenta konfigurācijas failā.
- Ja aģenta versija ir jaunāka par servera versiju, `Timeout` vērtībai servera konfigurācijas failā var būt jābūt lielākai nekā `Timeout` vērtībai aģenta konfigurācijas failā.

Zabbix aģents 2

Zabbix agent 2 atbalsta visas vienumu atslēgas, ko atbalsta Zabbix aģents Unix un Windows. Šajā lapā ir sniegta informācija par papildu vienumu atslēgām, kuras var izmantot tikai ar Zabbix agent 2, sagrupētām pēc spraudņa, kuram tās pieder.

Vienumu atslēgas ir uzskaitītas bez parametriem un papildu informācijas. Noklikšķiniet uz vienuma atslēgas, lai skatītu pilnu informāciju.

Vienuma atslēga	Apraksts	Spraudnis
ceph.df.details	Klastera datu izmantojums un sadalījums starp pūliem.	Ceph (ielādējams)
ceph.osd.stats	Apkopotā statistika un statistika pa OSD.	
ceph.osd.discovery	Atklāto OSD saraksts.	
ceph.osd.dump	OSD izmantošanas 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 izmantošanas statistika.	
docker.containers	Atgriež konteineru sarakstu.	
docker.containers.discovery	Atgriež konteineru sarakstu. Izmanto zema līmeņa atklāšanai.	
docker.data.usage	Informācija par pašreizējo datu izmantojumu.	
docker.images	Atgriež attēlu sarakstu.	
docker.images.discovery	Atgriež attēlu sarakstu. Izmanto 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 STATS komandas izvadi.	

Vienuma atslēga	Apraksts	Spraudnis
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 izmantošanas statistiku.	
mongodb.connpool.stats	Atgriež informāciju par atvērtajiem izejošajiem savienojumiem no pašreizējās datubāzes instances uz citiem sadalītā klastera vai replikācijas kopas dalībniekiem.	
mongodb.db.stats	Atgriež statistiku, kas atspoguļo norādītās datubāzes sistēmas stāvokli.	
mongodb.db.discovery	Atgriež atklāto datubāzu sarakstu.	
mongodb.jumbo_chunks.count	Atgriež jumbo chunk skaitu.	
mongodb.oplog.stats	Atgriež replikācijas kopas statusu, izmantojot no oplog aptaujātos datus.	
mongodb.ping	Pārbauda, vai savienojums ir aktīvs.	
mongodb.rs.config	Atgriež pašreizējo replikācijas kopas konfigurāciju.	
mongodb.rs.status	Atgriež replikācijas kopas statusu no tā dalībnieka skatpunkta, kurā metode tiek izpildīta.	
mongodb.server.status	Atgriež datubāzes stāvokli.	
mongodb.sh.discovery	Atgriež klasterī esošo atklāto shardu 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 darbu statusu.	
mssql.last.backup.get	Atgriež pēdējā dublējuma 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) un 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 kvora informāciju.	
mssql.quorum.member.get	Atgriež kvora 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.	
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	Veic DNS vaicājumu un atgriež detalizētu DNS ieraksta informāciju.	Network
nvml.device.count	GPU ierīču skaits.	NVIDIA GPU (ielādējams)
nvml.device.decoder.utilization	GPU ierīces dekodētāja izmantojums.	
nvml.device.ecc.mode	GPU ierīces ECC režīms.	
nvml.device.encoder.stats.get	GPU ierīces kodētāja statistika.	
nvml.device.encoder.utilization	GPU ierīces kodētāja izmantojums.	
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 vidējais ātrums.	
nvml.device.get	Atklātās GPU ierīces ar UUID un nosaukumu.	
nvml.device.graphics.frequency	GPU ierīces grafikas takts frekvence.	
nvml.device.memory.bar1.get	GPU ierīces BAR1 atmiņas informācija.	

Vienuma atslēga	Apraksts	Spraudnis
nvml.device.memory.fb.get	GPU ierīces kadru bufera atmiņas informācija.	
nvml.device.memory.frequency	GPU ierīces atmiņas takts frekvence.	
nvml.device.pci.utilization	GPU ierīces PCI izmantojums.	
nvml.device.performance.state	GPU ierīces veiktspējas stāvoklis.	
nvml.device.power.limit	GPU ierīces jaudas limits.	
nvml.device.power.usage	GPU ierīces jaudas patēriņš.	
nvml.device.serial	GPU ierīces sērijas numurs.	
nvml.device.sm.frequency	GPU ierīces straumēšanas multiprocesora takts frekvence.	
nvml.device.temperature	GPU ierīces temperatūra.	
nvml.device.utilization	GPU ierīces izmantošanas statistika.	
nvml.device.video.frequency	GPU ierīces video takts 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 tablespace statistiku.	
oracle.ts.discovery	Atgriež tablespace sarakstu.	
oracle.user.info	Atgriež Oracle lietotāja informāciju.	
oracle.version	Atgriež datubāzes servera versiju.	
pgsql.autovacuum.count	Autovacuum darbu skaits.	PostgreSQL (ielādējams)
pgsql.archive	Informācija par arhivētajiem failiem.	
pgsql.bgwriter	Kopējais kontrolpunktu skaits datubāzes klasterī, sadalīts pēc kontrolpunkta veida.	
pgsql.cache.hit	PostgreSQL bufera kešatmiņas trāpījumu īpatsvars.	
pgsql.connections	Atgriež savienojumus pēc veida.	
pgsql.custom.query	Atgriež pielāgota vaicājuma rezultātu.	
pgsql.db.age	Vecākā datubāzes 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 pa datubāzēm.	
pgsql.dbstat.sum	Apkopotie dati visām klastera datubāzēm.	
pgsql.locks	Informācija par piešķirtajām bloķēm 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	Rezerves serveru skaits.	
pgsql.replication.process	Flush aizture, write aizture un replay aizture katram sūtītāja procesam.	
pgsql.replication.process.discovery	Replikācijas procesa nosaukuma 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.	

Vienuma atslēga	Apraksts	Spraudnis
pgsql.version	Atgriež PostgreSQL versiju.	
pgsql.wal.stat	WAL statistika.	
redis.config	Iegūst Redis instances konfigurācijas parametrus, kas atbilst paraugam.	Redis
redis.info	Iegūst INFO komandas 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. ierīces 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 informācija.	
web.certificate.get	Validē sertifikātus un atgriež sertifikāta 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 noteiktu tēmu vai tēmas (ar aizstājējzīmēm) norādītajā brokerī un gaida publikācijas.
 Atgrieztā vērtība: atkarībā no tēmas satura. Ja tiek izmantotas aizstājējzīmes, tēmas saturs tiek atgriezts kā JSON.

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).

Piezīmes:

- vienums ir jākonfigurē kā aktīvā pārbaude ('Zabbix agent (active)' vienuma tips);
- TLS šifrēšanas sertifikātus var izmantot, saglabājot tos noklusējuma atrašanās vietā (piemēram, `/etc/ssl/certs/` direktorijā Ubuntu sistēmā). TLS gadījumā izmantojiet `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 spraudņa readme.

mssql.replica.get

 Atgriež replikas.
 Atgrieztā vērtība: *JSON objekts*.

Papildinformāciju skatiet MSSQL spraudņa readme.

mssql.version

 Atgriež MSSQL versiju.
 Atgrieztā vērtība: *String*.

Papildinformāciju skatiet MSSQL spraudņa 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 spraudņa 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 ieraksta informāciju.
 Šis vienums ir paplašināta `net.dns.record` Zabbix aģents vienuma versija ar vairāk atbalstītiem ierakstu tipiem un pielāgojamiem karodziņiem.
 Atgrieztās vērtības: *JSON object*

Parametri:

- **ip** - DNS servera IP adrese (atstājiet tukšu, lai izmantotu noklusējuma DNS serveri);
- **name** - DNS nosaukums, par kuru veikt vaicājumu;
- **type** - vaicājamais 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, ko nodot vaicājumam.

Piezīmes:

- 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; mazie burti vai jaukta reģistra vērtības netiek atbalstītas.
- Apgrieztajiem DNS uzmeklējumiem (kad `type` ir iestatīts uz *PTR*) varat norādīt DNS nosaukumu 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` *dnssec* un *nsid* nevar izmantot kopā ar *noedns0*, jo abiem ir nepieciešams *edns0*. Ņemiet vērā, ka vērtībām jābūt tikai ar mazajiem burtiem; lielie burti vai jaukta reģistra vērtības netiek atbalstītas.
- Internacionālizēti domēna vārdi netiek atbalstīti, lūdzu, izmantojiet IDNA kodētus nosaukumus.
- Izvade ir objekts, kas satur DNS ieraksta informāciju, pamatojoties uz norādītajiem parametriem (skatiet sīkāku informāciju).

Piemēri:

```
net.dns.get[192.0.2.1,example.com,DNSKEY,3,3,tcp,"cdflag,rdflag,nsid"]

net.dns.get[,198.51.100.1,PTR,,,,"cdflag,rdflag,nsid"]
net.dns.get[,1.100.51.198.in-addr.arpa,PTR,,,,"cdflag,rdflag,nsid"]

net.dns.get[,2a00:1450:400f:800::200e,PTR,,,,"cdflag,rdflag,nsid"]
net.dns.get[,e.0.0.2.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.0.f.0.0.4.0.5.4.1.0.0.a.2.ip6.arpa,PTR,,,,"cdflag,rdfla
```

nvml.device.count

 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 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 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 nedrīkst saturēt atstarpi beigās).
- **password** - Oracle parole;
- **service** - Oracle servisa nosaukums;
- **destination** - vaicājamā mērķa nosaukums.

oracle.cdb.info[connString,<user>,<password>,<service>,<database>]

 Konteinera datubāzu (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; 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;
- **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 atgriezt JSON tieši 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 kā 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 kā true un nav spēkā 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.db.discovery[connString,<user>,<password>,<service>]

 Atgriež datubāzu 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.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 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.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 beigu atstarpi).
- **password** - Oracle parole;
- **service** - Oracle servisa nosaukums.

oracle.pdb.info[connString,<user>,<password>,<service>,<database>]

 Informācija par pieslēdzamajām datubāzēm (PDB).
 Atgrieztā vērtība: *JSON objekts*.

Parametri:

- **connString** - tas var būt:
  - URI - ja sesija ar norādīto nosaukumu netiek atrasta un `Plugins.Oracle.ResolveTNS` ir iestatīts kā 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 kā 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 beigu atstarpi).
- **password** - Oracle parole;
- **service** - Oracle servisa nosaukums;
- **database** - vaicājamās datubāzes nosaukums.

oracle.pdb.discovery[connString,<user>,<password>,<service>]

 Atgriež PDB sarakstu. Tiek izmantots 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 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; 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.

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 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.

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 nav 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 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 jebkurā gadījumā tas 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 beigu atstarpi).
- **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 object*.

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.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 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;
- **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 metrikas vērtību kopu.
 Atgrieztā vērtība: *JSON objekts*.

Parametri:

- **connString** - tā var būt:
  - URI - ja sesija ar norādīto nosaukumu nav atrasta un `Plugins.Oracle.ResolveTNS` ir iestatīts uz false. Ja tajā ir shēma (piemēram, "tcp://"), ports (piemēram, 1521) vai abi, ResolveTNS opcija netiks ņemta vērā un tas jebkurā gadījumā tiks apstrādāts kā URI;
  - sesijas nosaukums - ja šāds nosaukums ir atrasts sprauduņa konfigurācijas failā;
  - TNS atslēga - ja ResolveTNS opcija ir iestatīta uz true un neviens no iepriekš minētajiem nosacījumiem nav spēkā;
  - TNS vērtība - ja tā sākas ar atverošo iekavu "(" (sākuma atstarpes tiek ignorētas). Šādā gadījumā ResolveTNS opcija 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 tā nedrīkst saturēt beigu atstarpi).
- **password** - Oracle parole;
- **service** - Oracle servisa nosaukums;
- **duration** - sistēmas metrikas vērtību iegūšanas intervāls (sekundēs). Iespējamās vērtības: *60* - ilgs 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 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.ts.stats[connString,<user>,<password>,<service>,<tablespace>,<type>,<conname>]

 Atgriež tabultelpu statistiku.
 Atgrieztā vērtība: *JSON object*.

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;
- **tablespace** - vaicājamās tabultelpas 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 tabultelpas tips. Noklusējums (ja `tablespace` ir iestatīts): "PERMANENT".
- **conname** - konteinera nosaukums, kuram ir nepieciešama informācija.

Ja `tablespace`, `type` vai `conname` nav norādīts, vienums atgriezīs tabultelpas statistiku visiem atbilstošajiem konteineriem (ieskaitot PDB un CDB).

oracle.ts.discovery[connString,<user>,<password>,<service>]

 Atgriež tabulvietu 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 beigu atstarpi).
- **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 tās beigās nedrīkst būt atstarpe).
- **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; 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.

pgsql.autovacuum.count[uri,<username>,<password>,<database name>]

 Autovacuum darbu skaits.
 Atgrieztā vērtība: *Integer*.

Parametri:

- **uri** - URI vai sesijas nosaukums;
- **username, password** - PostgreSQL piekļuves dati;
- **database name** - datubāzes nosaukums (ja tas 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 piekļuves 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 datubāzes klasterim, sadalīts pa kontrolpunktu tipiem.
 Atgrieztā vērtība: *JSON objekts*.

Parametri:

- **uri** - URI vai sesijas nosaukums;
- **username, password** - PostgreSQL piekļuves dati;
- **database name** - datubāzes nosaukums (ja nav norādīts, tiek izmantota vērtība `Plugins.PostgreSQL.Default.Database` no `postgresql.conf`).

pgsql.cache.hit[uri,<username>,<password>,<database name>]

 PostgreSQL bufera kešatmiņas trāpījumu īpatsvars.
 Atgrieztā vērtība: *Float*.

Parametri:

- **uri** - URI vai sesijas nosaukums;
- **username, password** - PostgreSQL piekļuves 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 no `Plugins.PostgreSQL.Default.Database` failā `postgresql.conf`).

pgsql.custom.query[uri,<username>,<password>,<database name>,queryName,<args...>]

 Atgriež pielāgota vaicājuma rezultātu.
 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);
- **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 (darījumu 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.db.bloating_tables[uri,<username>,<password>,<database name>]

 Bloatēto tabulu skaits katrā datubāzē.
 Atgrieztā vērtība: *Integer*.

Parametri:

- **uri** - URI vai sesijas nosaukums;
- **username, password** - PostgreSQL piekļuves 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 zemlīmeņa 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 `Plugins.PostgreSQL.Default.Database` vērtība no `postgresql.conf`).

pgsql.db.size[uri,<username>,<password>,<database name>]

 Datubāzes lielums baitos.
 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.dbstat[uri,<username>,<password>,<database name>]

 Apkopo statistiku katrai datubāzei. Tiek izmantots zemlīmeņa 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 no `Plugins.PostgreSQL.Default.Database` failā `postgresql.conf`).

pgsql.dbstat.sum[uri,<username>,<password>,<database name>]

 Apkopotie dati visām datubāzēm klasterī.
 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 no `Plugins.PostgreSQL.Default.Database` failā `postgresql.conf`).

pgsql.locks[uri,<username>,<password>,<database name>]

 Informācija par piešķirtajām bloķēšanām katrā datubāzē. Tiek izmantots zemlīmeņa 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 `Plugins.PostgreSQL.Default.Database` vērtība no `postgresql.conf`).

pgsql.oldest.xid[uri,<username>,<password>,<database name>]

 Vecākā XID vecums (darījumu 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 no `Plugins.PostgreSQL.Default.Database` failā `postgresql.conf`).

pgsql.ping[uri,<username>,<password>,<database name>]

 Pārbauda, vai savienojums ir aktīvs vai nav.
 Atgrieztā vērtība: *1* - savienojums ir aktīvs; *0* - savienojums ir pārtraukts (ja tiek konstatēta 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 no `Plugins.PostgreSQL.Default.Database` failā `postgresql.conf`).

pgsql.queries[uri,<username>,<password>,<database name>,time period]

 Vaicā 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 `Plugins.PostgreSQL.Default.Database` vērtība no `postgresql.conf`);
- **time period** - izpildes laika ierobežojums lēno vaicājumu skaita noteikšanai (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 no `Plugins.PostgreSQL.Default.Database` failā `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 no `Plugins.PostgreSQL.Default.Database` failā `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 objekts*.

Komentāri:

- Tiek atgrieztas šādas makro 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}` RAID gadījumā tiks papildināts, 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 operētājsistēmā Windows, un, ja tas ir izvēlēts, tiks atgriezta kļūda

Komentāri:

- Tiek atgrieztas šādas makro 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. `{#NAME}` RAID gadījumā saturēs papildu informāciju, 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 `{#PATH}` makro;
- **raid_type** - RAID tips, kā vērtību var izmantot `{#RAID}` makro

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 nepieciešamajiem papildinājumiem RAID atklāšanai, piem.: `{"disk_name": "/dev/sda cciss,2"}`
disk_type - satur diska tipu HDD, SSD vai NVME, piem.: `{"disk_type": "ssd"}`;
- Ja parametri nav norādīti, vienums atgriezīs informāciju par visiem diskiem.

systemd.unit.get[unit name,<interface>]

 Atgriež visas systemd vienuma īpašības.
 Atgrieztā vērtība: *JSON objekts*.

Parametri:

- **unit name** - vienuma nosaukums (varat izmantot `{#UNIT.NAME}` makro vienuma prototipā, lai noteiktu nosaukumu);
- **interface** - vienuma interfeisa 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 Unit interfeisam tiek atgriezti kā teksts un vesels skaitlis: `"ActiveState":{"state":1,"tex`

systemd.unit.info[unit name,<property>,<interface>]

 systemd vienības informācija.
 Atgrieztā vērtība: *String*.

Parametri:

- **unit name** - vienības nosaukums (iespējams, vēlēsieties izmantot makro `{#UNIT.NAME}` vienuma prototipā, lai noteiktu nosaukumu);
- **property** - vienības rekvizīts (piemēram, ActiveState (noklusējums), LoadState, Description);
- **interface** - vienības saskarnes tips (piemēram, Unit (noklusējums), Socket, Service).

Komentāri:

- Šis vienums tiek atbalstīts tikai Linux platformā;
- Šis vienums ļauj izgūt konkrētu rekvizītu no konkrēta saskarnes tipa, kā aprakstīts dbus API.

Piemēri:

```
systemd.unit.info["{#UNIT.NAME}"] #apkopot aktīvā stāvokļa (active, reloading, inactive, failed, activatin
systemd.unit.info["{#UNIT.NAME}",LoadState] #apkopot ielādes stāvokļa informāciju par atklātajām systemd v
systemd.unit.info[mysqld.service,Id] #izgūt servisa tehnisko nosaukumu (mysqld.service)
systemd.unit.info[mysqld.service,Description] #izgūt servisa aprakstu (MySQL Server)
systemd.unit.info[mysqld.service,ActiveEnterTimestamp] #izgūt pēdējo laiku, kad serviss pārgāja aktīvā stā
systemd.unit.info[dbus.socket,NConnections,Socket] #apkopot savienojumu skaitu no šīs ligzdas vienības
```

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 informāciju.
 Atgrieztā vērtība: *JSON objekts*.

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 adreses parametrs (3. parametrs), hostname tiek izmantots tikai SNI un hostname pārbaudei;
- **port** - porta numurs (HTTPS noklusējums 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.

Piezīmes:

- Šis vienums kļūst neatbalstīts, ja hosta konfigurācijā norādītais galamērķis neeksistē, nav pieejams vai ja TLS rokasspiediens neizdodas ar jebkuru kļūdu, izņemot nederīgu sertifikātu;
- Pašlaik netiek atbalstīti AIA (Authority Information Access) X.509 paplašinājums, CRL un OCSP (tostarp OCSP stapling), kā arī Certificate Transparency;
- JSON atbildes lauki:
  - *x509*: satur X.509 sertifikāta 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*: publiskās atslēgas algoritms (piemēram, `RSA`).
    - *alternative_names*: subjekta alternatīvie nosaukumi (ja tādi ir), pretējā gadījumā `null`.
  - result: satur validācijas rezultātu.
    - *value*: validācijas statuss (skatiet iespējamos vērtību variantus 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, t. i., tā subjekts sakrīt ar izdevēju.
  - *invalid* - sertifikāts nav derīgs, piemēram, derīguma termiņa beigšanās, nepareizs hostname 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"
}
```

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

Koplietotie vienumi

Tālāk esošajā tabulā ir uzskaitīti Zabbix aģenta vienumi, kas tiek atbalstīti operētājsistēmā Windows un ir koplietoti ar UNIX Zabbix aģentu:

- Vienuma atslēga ir saite uz pilnīgu UNIX Zabbix aģenta vienuma informāciju
- Ir iekļauti Windows attiecināmie vienuma komentāri

Item key	Description	Item group
log	Žurnālfaila uzraudzība. Šis vienums netiek atbalstīts Windows notikumu žurnālam. Parametrs `persistent_dir` operētājsistēmā Windows netiek atbalstīts.	Žurnālu uzraudzība
log.count	Atbilstošo rindu skaits uzraudzītajā žurnālfailā. Šis vienums netiek atbalstīts Windows notikumu žurnālam. Parametrs `persistent_dir` operētājsistēmā Windows netiek atbalstīts.	
logrt	Rotējoša žurnālfaila uzraudzība. Šis vienums netiek atbalstīts Windows notikumu žurnālam. Parametrs `persistent_dir` operētājsistēmā Windows netiek atbalstīts.	
logrt.count	Atbilstošo rindu skaits uzraudzītajā rotējošā žurnālfailā. Šis vienums netiek atbalstīts Windows notikumu žurnālam. Parametrs `persistent_dir` operētājsistēmā Windows netiek atbalstīts.	
modbus.get	Nolasa Modbus datus.	Modbus
net.dns	Pārbauda, vai DNS pakalpojums darbojas. Parametri `ip`, `timeout` un `count` operētājsistēmā Windows tiek ignorēti, ja vien netiek izmantots Zabbix aģents 2.	Tīkls
net.dns.perf	Pārbauda DNS pakalpojuma veiktspēju. Parametri `ip`, `timeout` un `count` operētājsistēmā Windows tiek ignorēti, ja vien netiek izmantots Zabbix aģents 2.	
net.dns.record	Veic DNS vaicājumu. Parametri `ip`, `timeout` un `count` operētājsistēmā Windows tiek ignorēti, ja vien netiek izmantots Zabbix aģents 2.	
net.if.discovery	Tīkla interfeisu saraksts. Dažām Windows versijām (piemēram, Server 2008) var būt nepieciešami jaunākie atjauninājumi, lai atbalstītu ne-ASCII rakstzīmes interfeisu nosaukumos.	

Item key	Description	Item group
net.if.in	Ienākošās datplūsmas statistika tīkla interfeisā. Operētājsistēmā Windows vienums iegūst vērtības no 64 bitu skaitītājiem, ja tie ir pieejami. 64 bitu interfeisa statistikas skaitītāji tika ieviesti Windows Vista un Windows Server 2008. Ja 64 bitu skaitītāji nav pieejami, aģents izmanto 32 bitu skaitītājus. Operētājsistēmā Windows tiek atbalstīti vairāku baitu interfeisu nosaukumi. Tīkla interfeisu aprakstus operētājsistēmā Windows var iegūt ar net.if.discovery vai net.if.list vienumiem.	
net.if.out	Izejošās datplūsmas statistika tīkla interfeisā. Operētājsistēmā Windows vienums iegūst vērtības no 64 bitu skaitītājiem, ja tie ir pieejami. 64 bitu interfeisa statistikas skaitītāji tika ieviesti Windows Vista un Windows Server 2008. Ja 64 bitu skaitītāji nav pieejami, aģents izmanto 32 bitu skaitītājus. Operētājsistēmā Windows tiek atbalstīti vairāku baitu interfeisu nosaukumi. Tīkla interfeisu aprakstus operētājsistēmā Windows var iegūt ar net.if.discovery vai net.if.list vienumiem.	
net.if.total	Ienākošās un izejošās datplūsmas statistikas summa tīkla interfeisā. Operētājsistēmā Windows vienums iegūst vērtības no 64 bitu skaitītājiem, ja tie ir pieejami. 64 bitu interfeisa statistikas skaitītāji tika ieviesti Windows Vista un Windows Server 2008. Ja 64 bitu skaitītāji nav pieejami, aģents izmanto 32 bitu skaitītājus. Operētājsistēmā Windows tiek atbalstīti vairāku baitu interfeisu nosaukumi. Tīkla interfeisu aprakstus operētājsistēmā Windows 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 operētājsistēmā Windows tiek atbalstīta tikai ar Zabbix aģentu 2.	
net.tcp.service.perf	Pārbauda TCP pakalpojuma veiktspēju. LDAP un HTTPS pārbaude operētājsistēmā Windows tiek atbalstīta tikai ar Zabbix aģentu 2.	
net.tcp.socket.count	Atgriež TCP ligzdu skaitu, kas atbilst parametriem. Šis vienums ir atbalstīts Linux ar Zabbix aģentu, bet operētājsistēmā Windows tas ir atbalstīts tikai ar Zabbix aģentu 2 64 bitu Windows vidē.	
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 ir atbalstīts Linux ar Zabbix aģentu, bet operētājsistēmā Windows tas ir atbalstīts tikai ar Zabbix aģentu 2 64 bitu Windows vidē.	
proc.get	Operētājsistēmas procesu un to parametru saraksts. Parametrs `cmdline` operētājsistēmā Windows netiek atbalstīts.	Procesi
proc.num	Procesu skaits. Operētājsistēmā Windows tiek atbalstīti tikai parametri `name` un `user`.	
system.cpu.discovery	Noteikto CPU/CPU kodolu saraksts.	Sistēma
system.cpu.load	CPU noslodze. Kad Zabbix aģentā tiek palaists savācēja process, tiek inicializēti un vēlāk šim vienumam izmantoti šādi veiktspējas skaitītāji: `\System\Processor Queue Length`	
system.cpu.num	CPU skaits.	
system.cpu.util	CPU izmantošanas procentuālā daļa. Vērtība tiek iegūta, izmantojot veiktspējas skaitītāju *Processor Time*. Ņemiet vērā, ka kopš Windows 8 uzdevumu pārvaldnieks CPU izmantošanu rāda, pamatojoties uz veiktspējas skaitītāju *Processor Utility*, savukārt iepriekšējās versijās tika izmantots *Processor Time* skaitītājs (skatiet sīkāku informāciju). Operētājsistēmā Windows vienīgais atbalstītais `type` parametrs ir *system*.	
system.hostname	Sistēmas hosta nosaukums. Vērtība operētājsistēmā Windows tiek iegūta, izmantojot funkcijas `GetComputerName()` (**netbios**), `GetComputerNameExA()` (**fqdn**) vai `gethostname()` (**host**). Skatiet arī detalizētāku aprakstu.	

Item key	Description	Item group
system.localtime	Sistēmas laiks.	
system.run	Palaist norādīto komandu uz hosta.	
system.sw.arch	Programmatūras arhitektūras informācija.	
system.sw.os	Operētājsistēmas informācija.	
system.sw.os.get	Detalizēta informācija par operētājsistēmu (versija, tips, distribūcijas nosaukums, mazākā un lielākā versija utt.).	
system.swap.size	Maiņas vietas lielums baitos vai procentos no kopējā apjoma. Parametrs `pused` operētājsistēmā Linux ir atbalstīts ar Zabbix aģentu, bet operētājsistēmā Windows tas ir atbalstīts tikai ar Zabbix aģentu 2. Ņemiet vērā, ka šī atslēga virtualizētās (VMware ESXi, VirtualBox) Windows platformās var ziņot par nepareizu maiņas vietas lielumu/procentuālo daļu. Šādā gadījumā varat izmantot atslēgu `perf_counter[\700(_Total)\702]`, lai iegūtu pareizu maiņas vietas procentuālo daļu.	
system.uname	Sistēmas identifikācija. Operētājsistēmā Windows šī vienuma vērtība tiek iegūta no `Win32_OperatingSystem` un `Win32_Processor` WMI klasēm. OS nosaukums (tostarp izdevums) var tikt tulkots lietotāja displeja valodā. Dažās Windows versijās tas satur preču zīmju simbolus un papildu atstarpes.	
system.uptime	Sistēmas darbības laiks sekundēs.	
vfs.dir.count	Kataloga ierakstu skaits. Operētājsistēmā Windows kataloga simboliskās saites tiek izlaistas, un cietās saites tiek skaitītas tikai vienu reizi.	Virtuālās failu sistēmas
vfs.dir.get	Kataloga ierakstu saraksts. Operētājsistēmā Windows kataloga simboliskās saites tiek izlaistas, un cietās saites tiek skaitītas tikai vienu reizi.	
vfs.dir.size	Kataloga lielums. Operētājsistēmā Windows jebkura simboliskā saite tiek izlaista, un cietās saites tiek ņemtas vērā tikai vienu reizi.	
vfs.file.cksum	Faila kontrolsumma, kas aprēķināta, izmantojot UNIX cksum algoritmu.	
vfs.file.contents	Faila satura iegūšana.	
vfs.file.exists	Pārbauda, vai fails pastāv. Operētājsistēmā Windows, izmantojot komandrindas utilītu `zabbix_get.exe` vai `agent2` izsaukšanai, dubultpēdiņām jābūt ar atpakaļslīpsvītru \\ ekrānētām, un visai vienuma atslēgai jābūt ietvertai dubultpēdiņās. Ņemiet vērā, ka vienums var kļūt neatbalstīts operētājsistēmā Windows, ja katalogs tiek meklēts neeksistējošā katalogā, piemēram, `vfs.file.exists[C:\no\dir,dir]` (kur 'no' nepastāv).	
vfs.file.get	Atgriež informāciju par failu. Atbalstītie failu tipi operētājsistēmā Windows: parasts fails, katalogs, 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. Operētājsistēmā Windows XP `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 pievienošanas opcijām. Makro `{#FSLABEL}` operētājsistēmā Windows tiek atbalstīts.	
vfs.fs.get	Pievienoto failu sistēmu saraksts ar to tipu, pieejamo diska vietu, inode statistiku un pievienošanas opcijām. Makro {#FSLABEL} operētājsistēmā Windows tiek atbalstīts.	
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

Item key	Description	Item group
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.	
zabbix.stats	Attālināti atgriež Zabbix servera vai starpniekservera iekšējo metriku kopu.	
zabbix.stats (queue)	Attālināti atgriež uzraudzīto vienumu skaitu rindā, kas ir aizkavēti Zabbix serverī vai starpniekserverī.	

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* Event Viewer GUI);
- **regexp** - regulāra izteiksme, kas apraksta nepieciešamo rakstu (reģistrjutīga);
- **severity** - regulāra izteiksme, kas apraksta nopietnības līmeni (reģistrnejutīga). Šis parametrs pieņem regulāru izteiksmi, pamatojoties uz šādām vērtībām: "Information", "Warning", "Error", "Critical", "Verbose" (darbojoties Windows Vista vai jaunākā versijā).
- **source** - regulāra izteiksme, kas apraksta avota identifikatoru (reģistrnejutīga);
- **eventid** - regulāra izteiksme, kas apraksta notikuma identifikatoru(s) (reģistrjutīga);
- **maxlines** - maksimālais jauno rindu skaits sekundē, ko aģents nosūtīs uz Zabbix serveri vai starpniekserveri. Šis parametrs pārraksta `MaxLinesPerSecond` vērtību `zabbix_agentd.conf`.
- **mode** - iespējamās vērtības:
  - *all* (noklusējums);
  - *skip* - izlaist vecāku datu apstrādi (ietekmē tikai jaunizveidotos vienumus).

Piezīmes:

- Vienumam jābūt konfigurētam kā aktīvai pārbaudei;
- Aģents nevar nosūtīt notikumus no žurnāla "Forwarded events";
- Tiek atbalstīts Windows Eventing 6.0;
- Izvēloties šim vienumam citu informācijas tipu, nevis Log, tiks zaudēts lokālais laikspiedols, kā arī žurnāla nopietnības 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 tiek izmantota pielāgota regulārā izteiksme ar nosaukumu `TWOSHORT` (d
```

eventlog.count[name,<regexp>,<severity>,<source>,<eventid>,<maxproclines>,<mode>]

 Windows notikumu žurnāla rindu skaits.
 Atgrieztā vērtība: *Integer*.

Parametri:

- **name** - notikumu žurnāla kanāla nosaukums (*Log Name* Event Viewer GUI);
- **regexp** - regulāra izteiksme, kas apraksta nepieciešamo rakstu (reģistrjutīga);
- **severity** - regulāra izteiksme, kas apraksta nopietnības līmeni (nereģistrjutīga). Šis parametrs pieņem regulāru izteiksmi, pamatojoties uz šādām vērtībām: "Information", "Warning", "Error", "Critical", "Verbose" (darbojoties Windows Vista vai jaunākā versijā).
- **source** - regulāra izteiksme, kas apraksta avota identifikatoru (nereģistrjutīga);
- **eventid** - regulāra izteiksme, kas apraksta notikuma identifikatoru(s) (reģistrjutīga);
- **maxproclines** - maksimālais jauno rindu skaits sekundē, ko aģents analizēs (nedrīkst pārsniegt 10000). Noklusējuma vērtība ir 10*`MaxLinesPerSecond` `zabbix_agentd.conf`.
- **mode** - iespējamās vērtības:
  - *all* (noklusējums);
  - *skip* - izlaist vecāku datu apstrādi (ietekmē tikai jaunizveidotos vienumus).

Komentāri:

- Vienumam jābūt konfigurētam kā aktīvai pārbaudei;
- Aģents nevar nosūtīt notikumus no žurnāla "Forwarded events";
- Tiek atbalstīts Windows Eventing 6.0;
- Izvēloties šim vienumam ne-Log informācijas tipu, 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** - skaitītāja ceļš;

- **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 būt drošības risks;
- `interval` tiek izmantots skaitītājiem, kuriem nepieciešams vairāk nekā viens paraugs (piemēram, CPU izmantošana), tāpēc pārbaude katru reizi atgriež pēdējo "interval" sekunžu vidējo vērtību;
- Performance Monitor var izmantot, lai iegūtu pieejamo skaitītāju sarakstu.
- Skatīt arī: Windows veiktspējas skaitītāji.

perf_counter_en[counter,<interval>]

 Jebkuras Windows veiktspējas skaitītāja vērtība angļu valodā.
 Atgrieztā vērtība: *Integer*, *float*, *string* vai *text* (atkarībā no pieprasījuma).

Parametri:

- **counter** - skaitītāja ceļš 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 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 pēdējo "interval" sekunžu vidējo vērtību;
- Šis vienums tiek atbalstīts tikai **Windows Server 2008/Vista** un jaunākās versijās;
- Angļu virkņu sarakstu var atrast, skatot 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>]

 Informācija par konkrētu procesu(-iem).
 Atgrieztā vērtība: *Float*.

Parametri:

- **process** - procesa nosaukums;
- **attribute** - pieprasītais procesa atribūts;
- **type** - attēlojuma tips (nozīmīgs, ja pastāv vairāki procesi ar vienādu nosaukumu)

Komentāri:

- Tiek atbalstīti šādi `attributes`:
  - *vmsize* (noklusējums) - procesa virtuālās atmiņas lielums Kbytes;
  - *wkset* - procesa darba kopas lielums (procesa izmantotās fiziskās atmiņas apjoms) Kbytes;
  - *pf* - lappušu kļūdu skaits;
  - *ktime* - procesa kodola laiks milisekundēs;
  - *utime* - procesa lietotāja laiks milisekundēs;
  - *io_read_b* - procesa nolasīto baitu skaits I/O operāciju laikā;
  - *io_read_op* - procesa veikto lasīšanas operāciju skaits;
  - *io_write_b* - procesa ierakstīto baitu skaits I/O operāciju laikā;
  - *io_write_op* - procesa veikto rakstīšanas operāciju skaits;
  - *io_other_b* - procesa pārsūtīto baitu skaits operācijās, kas nav lasīšanas un rakstīšanas operācijas;
  - *io_other_op* - procesa veikto I/O operāciju skaits, kas nav lasīšanas un rakstīšanas operācijas;
  - *gdiobj* - procesa izmantoto GDI objektu skaits;

    - *userobj* - procesa izmantoto USER objektu skaits.
- Derīgās `type`s vērtības ir:
    - *avg* (noklusējums) - visu procesu ar nosaukumu <process> vidējā vērtība;
    - *min* - minimālā vērtība starp visiem procesiem ar nosaukumu <process>;
    - *max* - maksimālā vērtība starp visiem procesiem ar nosaukumu <process>;
    - *sum* - visu procesu ar nosaukumu <process> vērtību summa.
- 64 bitu sistēmā, lai šis vienums darbotos pareizi, ir nepieciešams 64 bitu Zabbix aģents.

Piemēri:

```
proc_info[iexplore.exe,wkset,sum] #iegūst fiziskās atmiņas apjomu, ko izmanto visi Internet Explorer proce
proc_info[iexplore.exe,pf,avg] #iegūst Internet Explorer procesu vidējo lappušu kļūdu skaitu
```

registry.data[key,<value name>]

 Atgriež norādītā vērtības nosaukuma datus Windows reģistra atslēgā.
 Atgrieztā vērtība: *Integer*, *string* vai *text* (atkarībā no vērtības tipa)

Parametri:

- **key** - reģistra atslēga, ieskaitot saknes atslēgu; saknes saīsinājumi (piem., HKLM) ir atļauti;
- **value name** - 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 jānorobežo ar dubultpēdiņām.

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 šajā atslēgā
```

registry.get[key,<mode>,<name regexp>]

 Windows reģistra vērtību vai atslēgu saraksts, kas atrodas norādītajā atslēgā.
 Atgrieztā vērtība: *JSON objekts*.

Parametri:

- **key** - reģistra atslēga, ieskaitot saknes atslēgu; saknes saīsinājumi (piem., HKLM) ir atļauti (skatiet komentārus pie `registry.data[]`, lai redzētu pilnu saīsinājumu sarakstu);
- **mode** - iespējamās vērtības:
values (noklusējums) vai *keys*;
- **name regexp** - atklāj tikai tās vērtības, kuru nosaukumi atbilst regexp (noklusējums - atklāt visas vērtības). Atļauts tikai tad, ja *mode* ir *values*.

Atslēgas ar atstarpēm jāievieto dubultpēdiņās.

Piemēri:

```
registry.get[HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall,values,"^DisplayName|DisplayVersion$
JSON ietvers informāciju par atslēgu, pēdējo apakšatslēgu, vērtības nosaukumu, vērtības tipu un vērtības d
registry.get[HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall,values] #atgriež visu vērtību datus
JSON ietvers 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 ietvers 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 pakalpojumu.
 Atgrieztā vērtība: - *Integer* - ar `param` kā *state*, *startup*;

 Konkrēti *state*: - 0 - darbojas - 1 - pauzēts - 2 - palaišana gaida - 3 - pauzes gaida - 4 - turpināšanas gaida - 5 - apturēšanas gaida - 6 - apturēts - 7 - nezināms - 255 - šāds pakalpojums neeksistē

 Konkrēti *startup*: - 0 - automātisks - 1 - automātisks ar aizkavi - 2 - manuāls - 3 - atspējots - 4 - nezināms - 5 - automātiska trigeris palaišana - 6 - automātiska aizkavēta trigeris palaišana - 7 - manuāla trigeris palaišana - *String* - ar `param` kā *displayname*, *path*, *user*; - *Text* - ar `param` kā *description*.

Parametri:

- **service** - reāls pakalpojuma nosaukums vai tā attēlojamais nosaukums, kā redzams MMC Services snap-in;
- **param** - *state* (noklusējums), *displayname*, *path*, *user*, *startup* vai *description*.

Komentāri:

- Vienumi, piemēram, `service.info[service,state]` un `service.info[service]`, atgriezīs vienu un to pašu informāciju;
- Tikai ar `param` kā *state* šis vienums atgriež vērtību neesošiem pakalpojumiem (255).

Piemēri:

```
service.info[SNMPTRAP] - SNMPTRAP pakalpojuma statuss;
service.info[SNMP Trap] - tā paša pakalpojuma statuss, bet norādīts attēlojamais nosaukums;
service.info[EventLog,startup] - EventLog pakalpojuma 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>]

 Izpilda WMI vaicājumu un atgriež visu atbildi. Var izmantot zema līmeņa atklāšanai.
 Atgrieztā vērtība: *JSON objekts*

Parametri:

- **namespace** - WMI nosaukumvieta;
- **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 pakalpojuma nosaukumu.

Pakalpojuma nosaukumu varat iegūt, atverot MMC Services snap-in un atverot pakalpojuma rekvizītus. Cilnē *General* vajadzētu redzēt lauku ar nosaukumu *Service name*. Tam sekojošā vērtība ir nosaukums, ko izmantosiet, konfigurējot vienumu uzraudzībai. Piemēram, ja vēlaties uzraudzīt pakalpojumu "workstation", jūsu pakalpojums 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.

Žurnālfaila uzraudzība

Pārskats

Zabbix var izmantot centralizētai žurnālfailu uzraudzībai un analīzei ar vai bez žurnālfailu rotācijas atbalsta.

Paziņojumus var izmantot, lai brīdinātu lietotājus, ja žurnālfailā ir noteiktas virknes vai virkņu raksti.

Lai uzraudzītu žurnālfailu, jums ir nepieciešams:

- Zabbix aģents, kas darbojas uz hosta.
- Konfigurēts žurnāla uzraudzības vienums.

Attention:
Uzraudzītā žurnālfaila lieluma ierobežojums ir atkarīgs no lielu failu atbalsta.

Konfigurācija

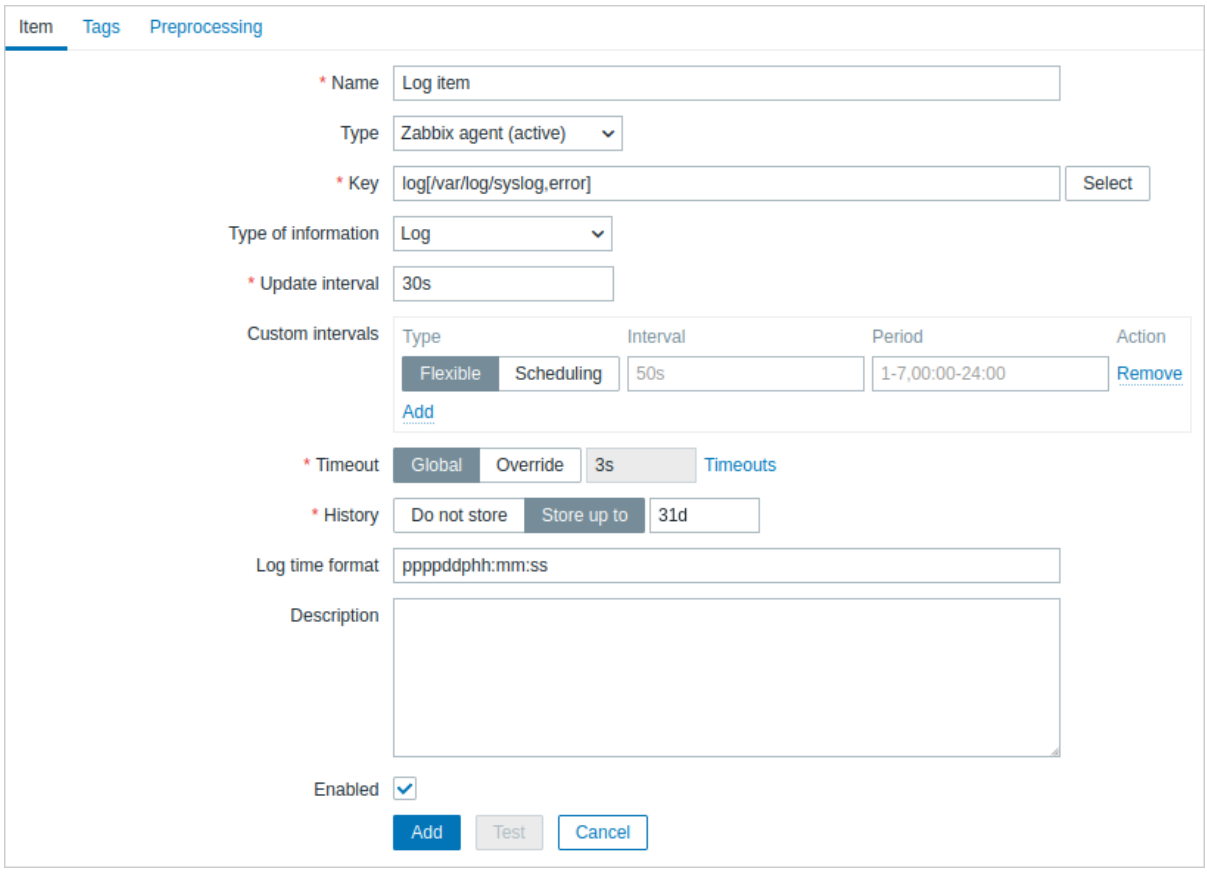
Pārbaudiet aģenta parametrus

Pārliecinieties, ka aģenta konfigurācijas failā:

- `Hostname` parametrs atbilst hosta nosaukumam lietotāja saskarnē.
- `ServerActive` parametrā norādītie serveri ir paredzēti aktīvo pārbaužu apstrādei.

Vienuma konfigurēšana

Konfigurējiet žurnāla pārraudzības vienumu.



Visi obligātie ievades lauki ir atzīmēti ar sarkanu zvaigznīti.

Konkrēti žurnāla pārraudzības vienumiem jāievada:

Type	Šeit atlasiet **Zabbix agent (active)**.
Key	Izmantojiet vienu no šīm vienuma atslēgām: `log[]` vai `logrt[]`: Šīs abas vienuma atslēgas ļauj pārraudzīt žurnālus un filtrēt žurnāla ierakstus pēc satura regulārās izteiksmes, ja tā ir norādīta. Piemēram: `log[/var/log/syslog,error]`. Pārliecinieties, ka 'zabbix' lietotājam ir lasīšanas atļaujas failam; pretējā gadījumā vienuma statuss tiks iestatīts uz 'unsupported'. **log.count[]** vai **logrt.count[]**: Šīs abas vienuma atslēgas ļauj atgriezt tikai atbilstošo rindu skaitu. Detalizētu informāciju par šo vienuma atslēgu un to parametru izmantošanu skatiet atbalstīto Zabbix agent vienuma atslēgu sadaļā.

Type of information	Tiek automātiski aizpildīts: `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 zaudēta lokālā laika zīmoga informācija.
Update interval (in sec)	Šis parametrs nosaka, cik bieži Zabbix aģents pārbaudīs, vai žurnāla failā nav veiktas izmaiņas. Iestatot intervālu uz 1 sekundi, tiks nodrošināts, ka jaunie ieraksti tiek iegūti pēc iespējas ātrāk.
Log time format	Šajā laukā pēc izvēles varat norādīt žurnāla rindas laika zīmoga parsēšanas modeli. 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 ir tukšs, laika zīmogs tiks iestatīts uz 0 Unix laikā, kas apzīmē `January 1, 1970`. Piemēram, apsveriet šo 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 jebkura rakstzīme, izņemot "yMdhms".

Svarīgas piezīmes

- Serveris un aģents uztur uzraudzītā žurnāla faila lieluma un pēdējās modificēšanas laika (for `logrt`) izsekošanu divos skaitītājos. Papildus:
  - Aģents arī iekšēji izmanto inode numurus (UNIX/GNU/Linux), failu indeksus (Microsoft Windows) un MD5 summas pirmajiem 512 žurnāla faila baitiem, lai uzlabotu lēmumus, kad žurnālfaili tiek saīsināti un rotēti.
  - UNIX/GNU/Linux sistēmās tiek pieņemts, ka failu sistēmas, kurās glabājas žurnāla faili, ziņo inode numurus, kurus var izmantot failu izsekošanai.
  - Microsoft Windows sistēmās Zabbix aģents nosaka failu sistēmas tipu, kurā atrodas žurnāla faili, un izmanto:
    * NTFS failu sistēmās 64 bitu failu indeksus.
    * ReFS failu sistēmās (tikai no Microsoft Windows Server 2012) 128 bitu failu ID.
    * Failu sistēmās, kur failu indeksi mainās (piem., FAT32, exFAT), tiek izmantots rezerves algoritms, lai pieņemtu saprātīgu pieeju nenoteiktos apstākļos, kad žurnāla failu rotācija rezultējas ar vairākiem žurnāla failiem ar vienādu pēdējās modificēšanas laiku.
  - Inode numuri, failu indeksi un MD5 summas tiek iekšēji savāktas ar Zabbix aģents. Tās netiek nosūtītas uz Zabbix serveris un tiek zaudētas, kad Zabbix aģents tiek apturēts.
  - Nemainiet žurnāla faila pēdējās modificēšanas laiku (piemēram, ar `touch`) un neaizstājiet uzraudzītu žurnāla failu, kopējot failu atpakaļ tā sākotnējā nosaukumā (tas izveido jaunu inode). Jebkurā no šiem gadījumiem Zabbix var uzskatīt failu par citu failu un nolasīt to no sākuma, kas var radīt dublētus brīdinājumus.
  - Ja `logrt[]` vienumam ir vairāki atbilstoši žurnāla faili un Zabbix aģents seko jaunākajam no tiem, bet šis jaunākais žurnāla fails tiek dzēsts, tiek reģistrēts brīdinājuma ziņojums `"there are no files matching "<regexp mask>" in "<directory>"`. Zabbix aģents ignorē žurnāla failus, kuru modificēšanas laiks ir mazāks par jaunāko modificēšanas laiku, ko aģents ir redzējis attiecīgajam pārbaudāmajam `logrt[]` vienumam.
- Aģents sāk lasīt žurnāla failu no vietas, kur tas apstājās iepriekšējā reizē.
- Jau analizēto baitu skaits (lieluma skaitītājs) un pēdējās modificēšanas laiks (laika skaitītājs) tiek glabāti Zabbix datubāzē un tiek nosūtīti aģentam, lai nodrošinātu, ka aģents sāk lasīt žurnāla failu no šīs vietas gadījumos, kad aģents ir tikko palaists vai ir saņēmis vienumus, kas iepriekš bija atspējoti vai netika atbalstīti. Tomēr, ja aģents no serveris ir saņēmis nenulles lieluma skaitītāju, bet `logrt[]` vai `logrt.count[]` vienums nevar atrast atbilstošus failus, lieluma skaitītājs tiek atiestatīts uz 0, lai analizētu no sākuma, ja faili parādās vēlāk.
- Ikreiz, kad žurnāla fails kļūst mazāks par žurnāla lieluma skaitītāju, kas zināms aģentam, skaitītājs tiek atiestatīts uz nulli un aģents sāk lasīt žurnāla failu no sākuma, ņemot vērā laika skaitītāju.
- Ja direktorijā ir vairāki atbilstoši faili ar vienādu pēdējās modificēšanas laiku, aģents mēģina pareizi analizēt visus žurnāla failus 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 žurnāla failu rotācijas shēmu un arī nenosaka to. Ja tiek atrasti vairāki žurnāla faili ar vienādu pēdējās modificēšanas laiku, aģents tos apstrādās leksikogrāfiski dilstošā secībā. Tādējādi dažām rotācijas shēmām žurnāla faili tiks analizēti un ziņoti to sākotnējā secībā. Citām rotācijas shēmām sākotnējā žurnāla failu secība netiks ievērota, kas var novest pie atbilstošo žurnāla failu ierakstu ziņošanas mainītā secībā (problēma

nerodas, ja žurnāla failiem ir atšķirīgi pēdējās modificēšanas laiki).

- Zabbix aģents apstrādā jaunus žurnāla faila ierakstus reizi ik pēc *Update interval* sekundēm.
- Zabbix aģents nesūta vairāk par `maxlines` žurnāla faila rindām sekundē. Šis ierobežojums novērš tīkla un CPU resursu pārslodzi un pārraksta noklusējuma vērtību, ko nodrošina `MaxLinesPerSecond` parametrs aģenta konfigurācijas failā.
- Lai atrastu nepieciešamo virkni, Zabbix apstrādās 10 reizes vairāk jaunu rindu nekā iestatīts `MaxLinesPerSecond`. Tādējādi, piemēram, ja `log[]` vai `logrt[]` vienumam *Update interval* ir 1 sekunde, pēc noklusējuma aģents analizēs ne vairāk kā 200 žurnāla faila ierakstus un vienā pārbaudē nosūtīs ne vairāk kā 20 atbilstošus ierakstus uz Zabbix serveris. Palielinot `MaxLinesPerSecond` aģenta konfigurācijas failā vai iestatot `maxlines` parametru vienuma atslēgā, ierobežojumu var palielināt līdz 10000 analizētiem žurnāla faila ierakstiem un 1000 atbilstošiem ierakstiem, kas vienā pārbaudē nosūtīti uz Zabbix serveris. Ja *Update interval* ir iestatīts uz 2 sekundēm, viena pārbaudei noteiktie ierobežojumi būs 2 reizes lielāki nekā pie *Update interval* 1 sekunde.
- Papildus `log` un `log.count` vērtības vienmēr ir ierobežotas līdz 50% no aģenta nosūtīšanas bufera lieluma, pat ja tajā nav citu, ne žurnāla vērtību. Tāpēc, lai `maxlines` vērtības tiktu nosūtītas vienā savienojumā (nevis vairākos savienojumos), aģenta `BufferSize` parametram jābūt vismaz `maxlines` x 2. Zabbix aģents var augšupielādēt datus žurnāla apkopošanas laikā un tādējādi atbrīvot buferi, savukārt Zabbix aģents 2 apturēs žurnāla apkopošanu, līdz dati tiks augšupielādēti un buferis atbrīvots, kas tiek veikts asinhroni.
- Ja nav žurnāla vienumu, viss aģenta bufera lielums tiek izmantots ne žurnāla vērtībām. Kad parādās žurnāla vērtības, tās pēc vajadzības aizstāj vecākās ne žurnāla vērtības līdz noteiktajiem 50%.
- Žurnāla faila ierakstiem, kas ir garāki par 256kB, ar regulāro izteiksmi tiek salīdzināti tikai pirmie 256kB, 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 `file` direktorija, jo nav iespējams nepārprotami noteikt, vai `.` ir ekrānēts vai ir faila nosaukuma pirmais simbols.
- Regulārās izteiksmes `logrt` tiek atbalstītas tikai faila nosaukumā, direktorija regulārās izteiksmes atbilstība netiek atbalstīta.
- UNIX platformās `logrt[]` vienums kļūst NOTSUPPORTED, ja direktorijs, kurā paredzēts atrast žurnāla failus, nepastāv.
- Microsoft Windows, ja direktorijs nepastāv, vienums nekļūs NOTSUPPORTED (piemēram, ja direktorija nosaukums vienuma atslēgā ir uzrakstīts nepareizi).
- Žurnāla failu neesamība `logrt[]` vienumam nepadara to par NOTSUPPORTED. Žurnāla failu lasīšanas kļūdas `logrt[]` vienumam tiek reģistrētas kā brīdinājumi Zabbix aģenta žurnāla failā, bet nepadara vienumu par NOTSUPPORTED.
- Zabbix aģenta žurnāla fails var būt noderīgs, lai noskaidrotu, kāpēc `log[]` vai `logrt[]` vienums kļuva NOTSUPPORTED. Zabbix var uzraudzīt sava aģenta žurnāla failu, izņemot gadījumus, kad ir `DebugLevel=4` vai `DebugLevel=5`.
- Meklēšana pēc jautājuma zīmes, izmantojot regulāro izteiksmi, piemēram, `\?`, var radīt kļūdaini pozitīvus rezultātus, ja teksta fails satur NUL simbolus, jo Zabbix tos aizstāj ar ?, lai turpinātu rindas apstrādi līdz jaunās rindas rakstzīmei.

Regulārās izteiksmes atbilstošās daļas izvilkšana

Dažkārt mēs varam vēlēties no mērķa faila izvilkt tikai interesējošo vērtību, nevis atgriezt visu rindu, kad tiek atrasta atbilstība regulārajai izteiksmei.

Žurnāla vienumiem ir iespēja izvilkt nepieciešamās vērtības no atbilstošajām rindām. To nodrošina papildu **output** parametrs `log` un `logrt` vienumos.

Izmantojot parametru `output`, var norādīt atbilstības "tveršanas grupu", kas mūs interesē.

Piemēram:

```
log[/path/to/the/file,"large result buffer allocation.*Entries: ([0-9]+)",,,,\1]
```

tam vajadzētu atgriezt ierakstu skaitu, kas atrasts 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]+)`.

Iespēja izvilkt un atgriezt skaitli ļauj šo vērtību izmantot trigeru definēšanai.

Izmantojot maxdelay parametru

Parametrs `maxdelay` žurnāla vienumos ļauj ignorēt dažas vecākas rindas no žurnālfailiem, lai tiktu analizētas jaunākās rindas `maxdelay` sekunžu ietvaros.

Warning:
Norādot `maxdelay` > 0, var tikt **ignorēti svarīgi žurnālfaila ieraksti un palaisti garām brīdinājumi**. Izmantojiet to uzmanīgi un uz savu risku tikai tad, ja tas ir nepieciešams.

Pēc noklusējuma vienumi žurnālu uzraudzībai seko visām jaunajām rindām, kas parādās žurnālfailos. Tomēr ir lietojumprogrammas, kas noteiktās situācijās sāk rakstīt milzīgu ziņojumu skaitu savos žurnālfailos. Piemēram, ja datubāze vai DNS serveris nav pieejams, šādas lietojumprogrammas pārpludina žurnālfailus ar tūkstošiem gandrīz identisku kļūdu ziņojumu, līdz normāla darbība tiek atjaunota. 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 `maxlines` parametra (tas aizsargā serveri no pārāk daudziem ienākošiem atbilstošiem žurnāla ierakstiem) un 10*`maxlines` limita (tas aizsargā hosts CPU un I/O no pārslodzes, ko vienā pārbaudē izraisa aģents). Tomēr iebūvētajai aizsardzībai ir 2 problēmas. Pirmkārt, serverim tiek ziņots par lielu skaitu potenciāli ne pārāk informatīvu ziņojumu, kas aizņem vietu datubāzē. Otrkārt, ierobežotā sekundē analizējamo rindu skaita dēļ aģents var vairākas stundas atpalikt no jaunākajiem žurnāla ierakstiem. Visdrīzāk jūs dotu priekšroku tam, lai par pašreizējo situāciju žurnālfailos tiktu paziņots ātrāk, nevis vairākas stundas tiktu pārlūkoti veci ieraksti.

Abu problēmu risinājums ir `maxdelay` parametra izmantošana. Ja ir norādīts `maxdelay` > 0, katras pārbaudes laikā tiek mērīts apstrādāto baitu skaits, atlikušo baitu skaits un apstrādes laiks. No šiem datiem aģents aprēķina aptuveno aizkavi - cik sekundes būtu nepieciešamas, lai analizētu visus atlikušos ierakstus žurnālfailā.

Ja aizkave nepārsniedz `maxdelay`, aģents turpina žurnālfaila analīzi kā parasti.

Ja aizkave ir lielāka par `maxdelay`, aģents **ignorē daļu no žurnālfaila, "pārlecot" tai pāri** uz jaunu aptuvenu pozīciju, lai atlikušās rindas varētu tikt analizētas `maxdelay` sekunžu laikā.

Ņemiet vērā, ka aģents pat nelasa ignorētās rindas 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
```

Numurs "to byte" ir aptuvens, jo pēc "pārlekšanas" aģents pielāgo pozīciju failā līdz žurnāla rindas sākumam, kas var atrasties tālāk failā vai agrāk.

Atkarībā no tā, kā pieauguma ātrums salīdzinās ar žurnālfaila analīzes ātrumu, jūs varat neredzēt nevienu "pārlekšanu", redzēt retas vai biežas "pārlekšanas", lielas vai mazas "pārlekšanas", vai pat nelielu "pārlekšanu" katrā pārbaudē. Sistēmas slodzes svārstības un tīkla latentums arī ietekmē aizkaves aprēķinu un līdz ar to "pārlekšanu" uz priekšu, lai neatpaliktu no `maxdelay` parametra.

Nav ieteicams iestatīt `maxdelay` < `update interval` (tas var izraisīt biežas nelielas "pārlekšanas").

Piezīmes par `copytruncate` žurnālfaila rotācijas apstrādi

`logrt` ar `copytruncate` opciju pieņem, ka dažādiem žurnālfailiem ir atšķirīgi ieraksti (vismaz to laikspiedoli ir atšķirīgi), tāpēc sākotnējo bloku MD5 summas (līdz pirmajiem 512 baitiem) būs atšķirīgas. Diviem failiem ar vienādām sākotnējo bloku MD5 summām tas nozīmē, ka viens no tiem ir oriģināls, bet otrs - kopija.

`logrt` ar `copytruncate` opciju cenšas pareizi apstrādāt žurnālfaila kopijas, neziņojot par dublikātiem. Tomēr tādas darbības kā vairāku žurnālfaila kopiju izveide ar vienādu laikspiedolu, žurnālfaila rotācija biežāk nekā `logrt[]` vienuma atjaunināšanas intervāls, kā arī bieža aģenta restartēšana nav ieteicamas. Aģents cenšas visas šīs situācijas apstrādāt saprātīgi, taču labus rezultātus visos apstākļos garantēt nevar.

Piezīmes par pastāvīgajiem failiem log*[] vienumiem

Pastāvīgo failu mērķis

Kad Zabbix aģents tiek palaists, tas saņem aktīvo pārbaužu sarakstu no Zabbix serveris vai starpniekserveris. Metriķiem `log*[]` tas saņem apstrādāto žurnāla izmēru un modificēšanas laiku, lai noteiktu, no kurienes sākt žurnāla faila uzraudzību. Atkarībā no faktiskā žurnāla faila izmēra un failu sistēmas ziņotā modificēšanas laika aģents nolemj vai nu turpināt žurnāla faila uzraudzību no apstrādātā žurnāla izmēra, vai arī atkārtoti analizēt žurnāla failu no sākuma.

Darbojošs aģents uztur lielāku atribūtu kopu, lai starp pārbaudēm sekotu visiem uzraudzītajiem žurnāla failiem. Šis atmiņā glabātais stāvoklis tiek zaudēts, kad aģents tiek apturēts.

Jaunais izvēles parametrs `persistent_dir` norāda direktoriju šī `log[]`, `log.count[]`, `logrt[]` vai `logrt.count[]` vienums stāvokļa glabāšanai failā. Žurnāla vienums stāvoklis tiek atjaunots no pastāvīgā faila pēc Zabbix aģents restartēšanas.

Galvenais lietošanas gadījums ir žurnāla faila uzraudzība, kas atrodas spoguļotā failu sistēmā. Līdz noteiktam brīdim žurnāla fails tiek rakstīts abos spoguļos. Pēc tam spoguļi tiek sadalīti. Aktīvajā kopijā žurnāla fails joprojām pieaug, saņemot jaunus ierakstus. Zabbix aģents to analizē un nosūta serveris apstrādātā žurnāla izmēru un modificēšanas laiku. Pasīvajā kopijā žurnāla fails paliek nemainīgs, ievērojami atpaliekot no aktīvās kopijas. Vēlāk operētājsistēma un Zabbix aģents tiek pārstartēti no pasīvās kopijas. Apstrādātais žurnāla izmērs un modificēšanas laiks, ko Zabbix aģents saņem no serveris, var nebūt derīgi situācijai pasīvajā kopijā.

Lai turpinātu žurnāla faila uzraudzību no vietas, kur aģents apstājās failu sistēmas spoguļa sadalīšanas brīdī, aģents atjauno savu stāvokli no pastāvīgā faila.

Aģenta darbība ar pastāvīgu failu

Palaišanas brīdī Zabbix aģents neko nezina par pastāvīgajiem failiem. Tikai pēc aktīvo pārbaužu saraksta saņemšanas no Zabbix serveris (starpniekserveris) aģents redz, ka daži žurnāla vienumi ir jāsaglabā pastāvīgos failos 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 spoguļošanas sadalīšanās notiek vienlaikus, ir ļoti maza, tāpēc tam nav nepieciešama īpaša apstrāde. Rakstīšanai pastāvīgajā failā neseko piespiedu sinhronizācija ar datu nesēju (`fsync()` netiek izsaukts).

Pārrakstīšana ar jaunākajiem datiem tiek veikta pēc veiksmīgas atbilstošā žurnālfaila ieraksta vai metadatu (apstrādātais žurnāla lielums un modificēšanas laiks) nosūtīšanas uz Zabbix serveris. Tas var notikt tik bieži kā katrā vienuma pārbaudē, ja žurnālfails turpina mainīties.

Aģenta izslēgšanas laikā īpašas darbības netiek veiktas.

Pēc aktīvo pārbaužu saraksta saņemšanas aģents atzīmē novecojušos pastāvīgos failus dzēšanai. Pastāvīgs fails kļūst novecojis, ja:

1. Attiecīgais žurnāla vienums vairs netiek uzraudzīts.
2. Žurnāla vienums tiek pārkonfigurēts ar citu `persistent_dir` atrašanās vietu nekā iepriekš.

Dzēšana tiek veikta ar 24 stundu aizkavi, jo žurnālfaili NOTSUPPORTED stāvoklī nav iekļauti aktīvo pārbaužu sarakstā, taču tie vēlāk var kļūt SUPPORTED, un to pastāvīgie faili būs noderīgi.

Ja aģents tiek apturēts pirms 24 stundu termiņa beigām, 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 veco `persistent_dir` atrašanās vietu, kamēr aģents ir apturēts, un lietotājs nav izdzēsis veco pastāvīgo failu, tas izraisīs aģenta stāvokļa atjaunošanu no vecā pastāvīgā faila, kā rezultātā tiks palaistas garām ziņas vai tiks ģenerēti kļūdaini brīdinājumi.

Pastāvīgo failu nosaukumi un atrašanās vieta

Zabbix aģents aktīvās pārbaudes atšķir pēc to atslēgām. Piemēram, `logrt[/home/zabbix/test.log]` un `logrt[/home/zabbix/test.l` ir dažādi vienumi. Mainot lietotāja saskarnē vienumu `logrt[/home/zabbix/test.log,,,10]` uz `logrt[/home/zabbix/test.log,,,2` vienums `logrt[/home/zabbix/test.log,,,10]` tiek dzēsts no aģenta aktīvo pārbaužu saraksta un tiek izveidots vienums `logrt[/home/zabbix/test.log,,,20]` (daži atribūti tiek pārnesti, veicot izmaiņas lietotāja saskarnē/serverī, bet ne aģentā).

Faila nosaukumu veido vienuma atslēgas MD5 kontrolsumma, kurai pievienots vienuma atslēgas garums, lai samazinātu sadursmju iespējamību. Piemēram, vienuma `logrt[/home/zabbix50/test.log,,,,,,,,/home/zabbix50/agent_private]` stāvoklis tiks saglabāts pastāvīgajā failā c963ade4008054813bbc0a650bb8e09266.

Vairāki žurnāla vienumi var izmantot vienādu `persistent_dir` vērtību.

`persistent_dir` tiek norādīts, ņemot vērā konkrēto failu sistēmu izkārtojumu, montēšanas punktus un opcijas, kā arī krātuves spoguļošanas konfigurāciju - pastāvīgajam failam jāatrodas tajā pašā spoguļotajā failu sistēmā, kurā atrodas uzraudzītais žurnāla fails.

Ja `persistent_dir` direktoriju nevar izveidot vai tā neeksistē, vai arī Zabbix aģentam nav pietiekamu atļauju failu izveidei, rakstīšanai, lasīšanai vai dzēšanai, žurnāla vienums kļūst par `NOTSUPPORTED`.

Ja aģenta darbības laikā tiek noņemtas piekļuves tiesības pastāvīgās krātuves failiem vai rodas citas kļūdas (piemēram, disks ir pilns), kļūdas tiek ierakstītas aģenta žurnālfailā, taču žurnāla vienuma statuss nekļūst par `NOTSUPPORTED`.

I/O slodze

Vienuma pastāvīgais fails tiek atjaunināts pēc katras datu partijas veiksmīgas nosūtīšanas (kas satur vienuma datus) uz serveri. Piemēram, noklusējuma `BufferSize` ir 100. Ja žurnāla vienums ir atradis 70 atbilstošus ierakstus, tad pirmie 50 ieraksti tiks nosūtīti vienā partijā, pastāvīgais fails tiks atjaunināts, pēc tam atlikušie 20 ieraksti tiks nosūtīti (iespējams, ar nelielu aizkavi, kad tiks uzkrāts vairāk datu) otrajā partijā, un pastāvīgais fails tiks atjaunināts vēlreiz.

Darbības, ja saziņa starp aģentu un serveri neizdodas

Katra atbilstošā rinda no `log[]` un `logrt[]` vienuma, kā arī katra `log.count[]` un `logrt.count[]` vienuma pārbaudes rezultāts prasa brīvu slotu aģenta nosūtīšanas bufera norādītajā 50% zonā. Bufera elementi tiek regulāri nosūtīti uz serveri (vai starpniekserveri), un bufera sloti atkal kļūst brīvi.

Kamēr aģenta nosūtīšanas bufera norādītajā logu zonā ir brīvi sloti un saziņa starp aģentu un serveri (vai starpniekserveri) neizdodas, logu uzraudzības rezultāti tiek uzkrāti nosūtīšanas buferī. Tas palīdz mazināt īslaicīgas saziņas kļūmes.

Ilgāku saziņas kļūmju laikā visi logu sloti tiek aizņemti, un tiek veiktas šādas darbības:

- `log[]` un `logrt[]` vienuma pārbaudes tiek apturētas. Kad saziņa tiek atjaunota un buferī ir pieejami brīvi sloti, pārbaudes tiek atsāktas no iepriekšējās pozīcijas. Neviena atbilstoša rinda netiek zaudēta, tā vienkārši tiek paziņota vēlāk.
- `log.count[]` un `logrt.count[]` pārbaudes tiek apturētas, ja `maxdelay = 0` (noklusējums). Uzvedība 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 slotu, pārbaude tiek apturēta. Kad saziņa tiek atjaunota, aģents saskaita tās pašas 100 atbilstošās rindas, kā arī 70 jaunas atbilstošas rindas. Tagad aģents nosūta count = 170 tā, it kā tās būtu atrastas vienā pārbaudē.
- `log.count[]` un `logrt.count[]` pārbaudes ar `maxdelay > 0`: ja pārbaudes laikā nenotika "lēciens", tad uzvedība ir līdzīga iepriekš aprakstītajai. Ja notika "lēciens" pāri žurnāla faila rindām, tad pozīcija pēc "lēciens" tiek saglabāta, bet saskaitītais rezultāts tiek atmests. Tādējādi aģents cenšas sekot līdzi augošam žurnāla failam pat saziņas kļūmes gadījumā.

Regulāro izteiksmju kompilēšanas un izpildlaika kļūdu apstrāde

Ja regulāro izteiksmi, kas izmantota `log[]`, `logrt[]`, `log.count[]` vai `logrt.count[]` vienumā, nevar kompilēt PCRE vai PCRE2 bibliotēka, vienums pāriet stāvoklī NOTSUPPORTED ar kļūdas ziņojumu. Lai turpinātu uzraudzīt žurnāla vienumu, regulārā izteiksme ir jāizlabo.

Ja regulārā izteiksme tiek veiksmīgi kompilēta, bet izpildlaikā neizdodas (dažiem vai visiem žurnāla ierakstiem), 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 ierakstu apstrāde neizdodas regulārās izteiksmes izpildlaika kļūdas dēļ, aģenta žurnālā tiek izveidots viens ieraksts.

Izņēmums: ja `MaxLinesPerSecond=1` un `update interval=1` (katrā pārbaudē ļaujot analizēt tikai vienu ierakstu), regulārās izteiksmes izpildlaika kļūdas netiek reģistrētas.

zabbix_agentd izpildlaika kļūdas gadījumā reģistrē vienuma atslēgu, savukārt zabbix_agent2 reģistrē vienuma ID, lai palīdzētu noteikt, kuram žurnāla vienumam ir izpildlaika kļūdas. Izpildlaika kļūdu gadījumā ieteicams pārveidot regulāro izteiksmi.

Vienkārša pārbaude

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 izvēles parametriem un papildu informācijas. Noklikšķiniet uz vienuma atslēgas, lai skatītu pilnu informāciju.

Skatiet arī VMware monitoringa vienumu atslēgas.

Vienuma atslēga	Apraksts
icmpping	Hosts pieejamība, izmantojot ICMP ping.
icmppingloss	Zaudēto pakešu procentuālā daļa.
icmppingretry	Hosts pieejamība, izmantojot ICMP ping ar atkārtotiem mēģinājumiem.
icmppingsec	ICMP ping atbildes laiks.
net.tcp.service	Pārbauda, vai pakalpojums darbojas un pieņem TCP savienojumus.

Vienuma atslēga	Apraksts
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>]

 Hosts pieejamība, izmantojot ICMP ping.
 Atgrieztā vērtība: *0* - ICMP ping neizdodas; *1* - ICMP ping ir veiksmīgs.

Parametri:

- **target** - hosts IP vai DNS nosaukums;
- **packets** - pakešu skaits;
- **interval** - laiks starp secīgām paketēm milisekundēs;
- **size** - paketes lielums baitos;
- **timeout** - noildze milisekundēs;
- **options** - tiek izmantots pāradresācijas atļaušanai: ja tukšs (noklusējuma vērtība), pāradresētās atbildes tiek uzskatītas par mērķa hostu, kas nav pieejams; ja iestatīts uz *allow_redirect*, pāradresētās atbildes tiek uzskatītas par mērķa hostu, kas ir pieejams.

Skatiet arī noklusējuma vērtību tabulu.

Piezīme:

- Ja tiek saņemta vismaz viena no `packets` parametrā definētajām paketēm, vienums atgriež 1. Ja neviena no paketēm netiek saņemta, vienums atgriež 0.

Piemērs:

```
icmpping[,4] #Ja tiek saņemta vismaz viena no četrām paketēm, vienums atgriež 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 *FpingLocation* parametrā (vai *Fping6Location* parametrā, ja tiek izmantotas IPv6 adreses).

fping jābūt izpildāmam lietotājam, ar kuru 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

ICMP pārbaudes parametru noklusējuma vērtības, ierobežojumi un apraksts:

Parametrs	Mērvienība	Apraksts	Fping karo-dziņš	Noklusējuma vērtības iestata		Atļautie ierobežo-jumi pēc Zabbix	
				fping	**Zabbix**	**min**	**max**
packets	skaits	Mērķim nosūtīto pieprasījuma pakešu skaits	-C		3	1	10000
interval	milisekundes	Gaidīšanas laiks starp secīgām paketēm, kas tiek nosūtītas vienam mērķim	-p	1000		20	neierobežots

Parametrs	Mērvienība	Apraksts	Fping karo-dziņš	Noklusējuma vērtības iestata		Atļautie ierobežo-jumi pēc Zabbix	
size	baiti	Pakešu lielums baitos 56 baiti x86 arhitektūrā, 68 baiti x86_64 arhitektūrā	-b	56 vai 68		24	65507
timeout	milisekundes	**fping v3.x** - gaidīšanas noildze pēc pēdējās nosūtītās paketes, ko ietekmē karodziņš `-C` **fping v4.x** - katras paketes individuālā noildze	-t	**fping v3.x** - 500 **fping v4.x** un jaunākās versijas - mantota no karo-dziņa `-p`, bet ne lielāka par 2000		50	neierobežots
retries	skaits	Mēģinājumu skaits, lai veiktu mērķa ping pārbaudi, neieskaitot pirmo mēģinājumu	-r	3	1	0	neierobežots
backoff factor	skaits	Skaitlis, ar kuru katrā nākamajā pieprasījumā tiek reizināts gaidīšanas laiks	-B	1.5	1.0	1.0	5.0

Noklusējuma vērtības var nedaudz atšķirties atkarībā no platformas un versijas.

Turklāt Zabbix izmanto fping opcijas `-i interval ms` (nejauciet to ar vienuma parametru `interval`, kas minēts iepriekšējā tabulā un atbilst fping opcijai `-p`) un `-S source IP address` (vai `-I` vecākās fping versijās). Šīs opcijas tiek automātiski noteiktas, veicot pārbaudes ar dažādām opciju kombinācijām. Zabbix mēģina noteikt minimālo vērtību milisekundēs, ko fping pieļauj opcijai `-i`, izmēģinot trīs vērtības: 0, 1 un 10. Pirmā vērtība, ar kuru pārbaude ir veiksmīga, tiek izmantota turpmākajām ICMP pārbaudēm. Šo procesu katrs ICMP pinger process veic atsevišķi.

Automātiski noteiktās fping opcijas tiek anulētas reizi stundā un atkārtoti noteiktas nākamajā ICMP pārbaudes izpildes mēģinājumā. Iestatiet DebugLevel>=4, lai servera vai starpniekservera žurnālfailā skatītu detalizētu informāciju par šo procesu.

Zabbix ieraksta IP adreses, kuras 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 vienumi, kuriem ir identiski atslēgas parametri. Visas vienā failā ierakstītās IP adreses fping pārbauda paralēli, tāpēc Zabbix ICMP pinger process patērē nemainīgu laiku neatkarīgi no failā esošo IP adrešu skaita.

VMware monitoringa vienumu atslēgas

VMware uzraudzības vienumu atslēgu saraksts ir pārvietots uz sadaļu VMware uzraudzība.

SNMP aģents

Pārskats

Jūs varat vēlēties izmantot SNMP uzraudzību tādās ierīcēs kā printeri, tīkla komutatori, maršrutētāji vai UPS, kas parasti atbalsta SNMP un kurās būtu nepraktiski mēģināt iestatīt pilnas operētājsistēmas un Zabbix aģentus.

Lai varētu izgūt datus, ko SNMP aģenti sniedz šajās ierīcēs, Zabbix serverim sākotnēji jābūt konfigurētam ar SNMP atbalstu, norādot `--with-net-snmp` karogu. 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 līdzīgas rindas kā tālāk redzamā, 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 atsevišķu SNMP ierīču identificēšanai, kurām jāatspējo apvienotie pieprasījumi.

Zabbix serveris/starpniekserveris atkārtos mēģinājumu līdz 5 reizēm SNMP `walk` un `get` vienumiem. Atkārtošanas mehānisms neattiecas uz DNS atrisināšanas kļūmēm.

Mantotajām SNMP pārbaudēm (viens OID numurs vai virkne) Zabbix serveris/starpniekserveris pēc neveiksmīga vaicājuma mēģinājuma atkārtos vismaz vienu reizi: vai nu izmantojot SNMP bibliotēkas atkārtošanas mehānismu, vai iekšējo apvienotās apstrādes mehānismu.

Warning:
Ja uzraugāt SNMPv3 ierīces, pārliecinieties, ka msgAuthoritativeEngineID (pazīstams 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 savus engineBoots. Dažas ierīces to nedara, kā rezultātā pēc restartēšanas to SNMP ziņojumi tiek atmesti kā novecojuši. Šādā situācijā SNMP kešatmiņa serverī/starpniekserverī ir jānotīra manuāli (izmantojot -R snmp_cache_reload) vai arī serveris/starpniekserveris ir jārestartē.

Note:
Zabbix kešo SNMPv3 EngineID → IP kartējumus un atkārtotām pārbaudēm izmanto kešotos EngineID, nevis katru reizi sūta probe, tādējādi samazinot tīkla trafiku. Ja EngineID nevar atkārtoti izmantot, tiks veikts atkārtots mēģinājums ar probe, lai atklātu jauno EngineID.

SNMP uzraudzības konfigurēšana

Lai sāktu ierīces uzraudzību, izmantojot SNMP, ir jāveic šādas darbības:

1. solis

Noskaidrojiet SNMP virkni (vai OID) vienumam, ko vēlaties uzraudzīt.

Lai iegūtu SNMP virkņu sarakstu, izmantojiet komandu `snmpwalk` (net-snmp programmatūras daļa, kurai vajadzētu būt instalētai kā Zabbix instalācijas sastāvdaļai) vai līdzīgu rīku:

```
snmpwalk -v 2c -c public <host IP> .
```

Tā kā 2c šeit apzīmē SNMP versiju, to var aizstāt arī ar 1, lai norādītu ierīcē SNMP 1. versiju.

Tam vajadzētu atgriezt SNMP virkņu sarakstu un to pēdējo vērtību. Ja tas 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, ko vēlaties uzraudzīt, piemēram, ja vēlaties uzraudzīt ienākošos baitus savā komutatorā 3. portā, izmantosiet virkni `IF-MIB::ifHCInOctets.3` no šīs rindas:

```
IF-MIB::ifHCInOctets.3 = Counter64: 3409739121
```

Tagad varat izmantot komandu `snmpget`, lai noskaidrotu skaitlisko OID priekš `IF-MIB::ifHCInOctets.3`:

```
snmpget -v 2c -c public -On <host IP> IF-MIB::ifHCInOctets.3
```

Ņemiet vērā, ka pēdējais skaitlis virknē ir porta numurs, ko 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
```

Arī šeit OID pēdējais skaitlis ir porta numurs.

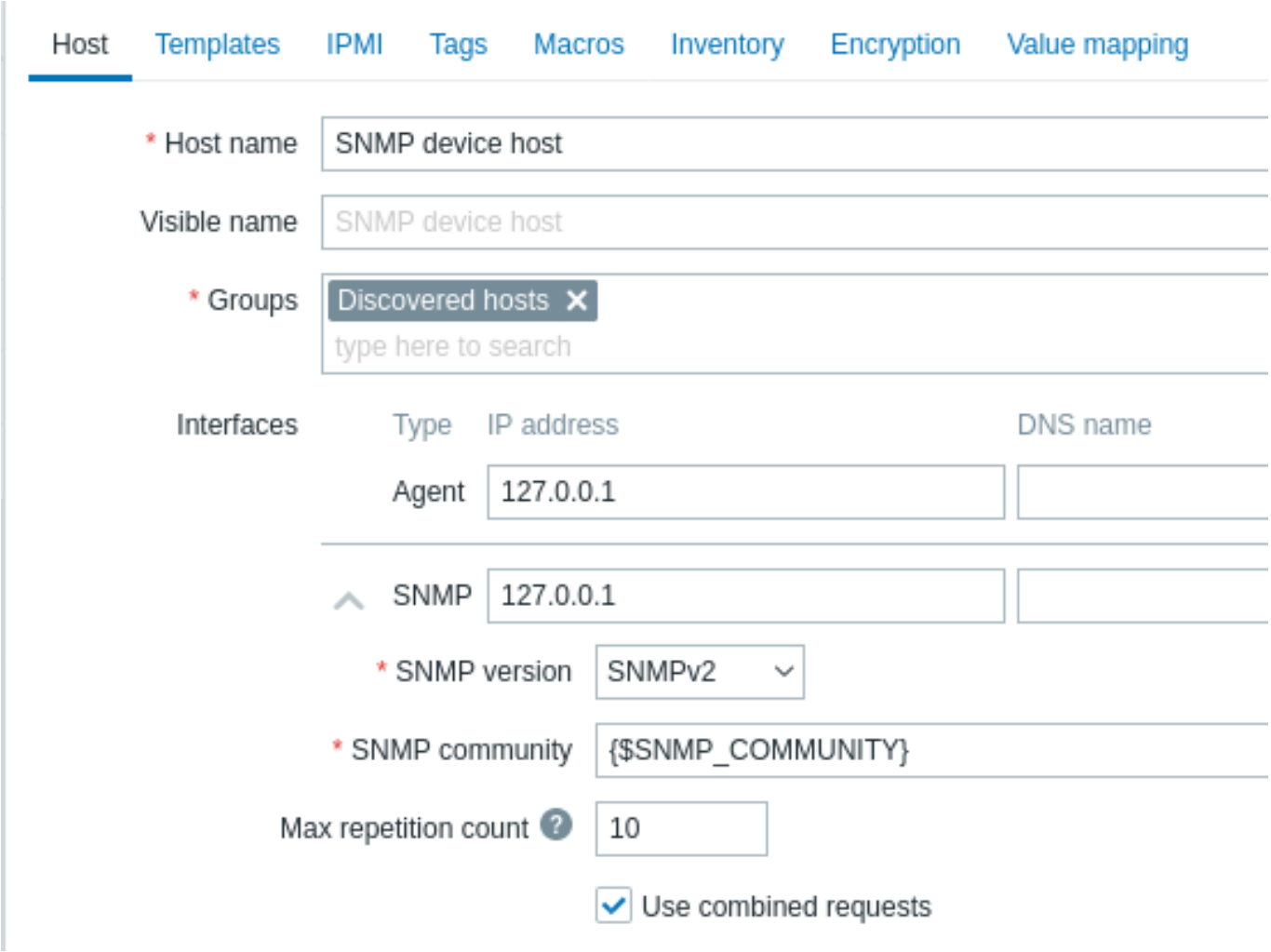
Note:
Daži no visbiežāk izmantotajiem SNMP OID tiek automātiski pārtulkoti skaitliskā attēlojumā ar Zabbix.

Iepriekšējā piemērā vērtības tips ir "Counter64", kas iekšēji atbilst ASN_COUNTER64 tipam. 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** izvades formātā, taču atkarībā no attēlošanas norādes klātbūtnes tie var tikt rādīti arī kā "STRING", "Hex-STRING", "OID" un citi.

2. solis

Izveidojiet hostu, kas atbilst ierīcei.



Pievienojiet hostam SNMP saskarni:

1. Ievadiet IP adresi/DNS nosaukumu un porta numuru.
2. Nolaižamajā sarakstā atlasiet *SNMP versiju*.
3. Pievienojiet saskarnes akreditācijas datus atkarībā no atlasītās SNMP versijas:
    - SNMPv1, v2 nepieciešams tikai community (parasti 'public').
    - SNMPv3 nepieciešamas specifiskākas opcijas; skatiet SNMPv3.
4. Norādiet maksimālo atkārtojumu vērtību (noklusējums: 10) native SNMP bulk requests (GetBulkRequest-PDU); tikai `discovery[]` un `walk[]` vienumiem SNMPv2 un v3. Ņemiet vērā, ka pārāk augstas šīs vērtības iestatīšana var izraisīt SNMP aģenta pārbaudes noildzi.
5. Atzīmējiet izvēles rūtiņu *Use combined requests*, lai atļautu SNMP pieprasījumu combined processing (nav saistīts ar native SNMP bulk requests "walk" un "get").

Varat izmantot kādu no piedāvātajām SNMP veidnēm, kas automātiski pievienos vienumu kopu. Pirms veidnes izmantošanas pārliecinieties, ka tā ir saderīga ar hostu.

Noklikšķiniet uz *Add*, lai saglabātu hostu.

SNMPv3

SNMPv3 ir nepieciešami šādi parametri:

- *Context name* - ievadiet konteksta nosaukumu, lai identificētu vienums SNMP apakštīklā.
Šajā laukā tiek aizstāti lietotāja makrosi.
- *Security name* - ievadiet drošības nosaukumu.
Šajā laukā tiek aizstāti lietotāja makrosi.
- *Security level* - izvēlieties 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* - izvēlieties autentifikācijas protokolu: *MD5*, *SHA1*; ar net-snmp 5.8 un jaunāku versiju - *SHA224*, *SHA256*, *SHA384* vai *SHA512*.
- *Authentication passphrase* - ievadiet autentifikācijas paroli.
Šajā laukā tiek aizstāti lietotāja makrosi.
- *Privacy protocol* - izvēlieties privātuma protokolu: *DES*, *AES128*, *AES192*, *AES256*, *AES192C* (Cisco) vai *AES256C* (Cisco).
Skatiet piezīmes par privātuma protokola atbalstu.
- *Privacy passphrase* - ievadiet privātuma paroli.
Šajā laukā tiek aizstāti lietotāja makrosi.

Ja SNMPv3 akreditācijas dati ir nepareizi (drošības nosaukums, autentifikācijas protokols/parole, privātuma protokols):

- Zabbix no net-snmp saņem ERROR, izņemot gadījumu, kad ir nepareiza *Privacy passphrase*; šādā gadījumā Zabbix no net-snmp saņem TIMEOUT kļūdu.
- SNMP saskarnes pieejamība tiks pārslēgta uz sarkanu (nav pieejama).

Note:
Izmaiņas *Authentication protocol*, *Authentication passphrase*, *Privacy protocol* vai *Privacy passphrase*, kas veiktas, nemainot *Security name*, parasti tiek automātiski piemērotas, kad attiecīgā SNMPv3 saskarne tiek atjaunināta Zabbix. Gadījumos, kad tiek mainīts arī *Security name*, visi parametri tiks atjaunināti nekavējoties.

Privātuma protokolu atbalsts

Atkarībā no jūsu operētājsistēmas un net-snmp konfigurācijas daži privātuma protokoli var nebūt pieejami:

- Dažās jaunākās operētājsistēmās (piemēram, RHEL9) DES atbalsts pakotnei `net-snmp` ir noņemts.
- Šifrēšanas protokoli AES192 un spēcīgāki pēc noklusējuma netiek atbalstīti vecākās operētājsistēmās nekā 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+ ir 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 instalēts.

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.

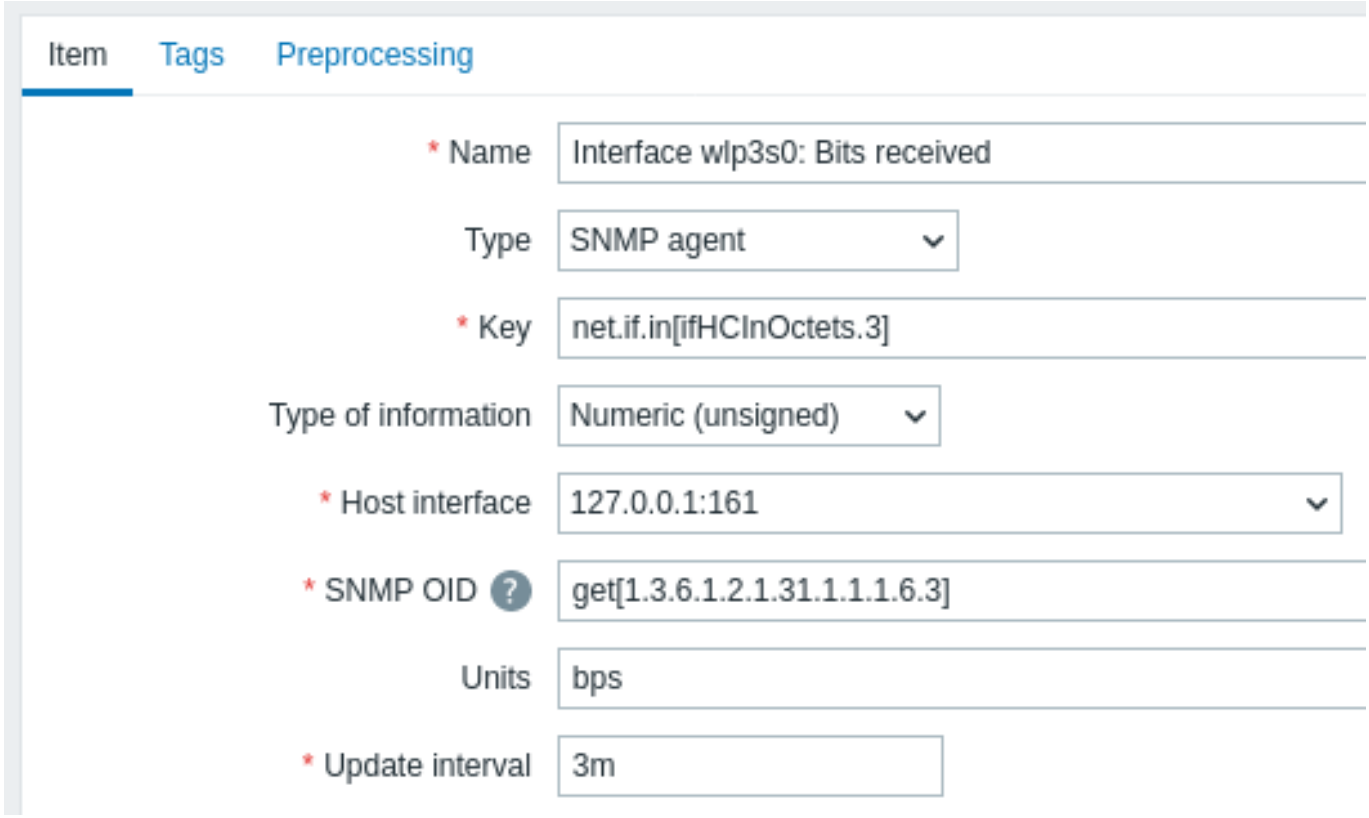
Ja jūsu net-snmp bibliotēka neatbalsta AES192 un augstākus protokolus, pārkompilējiet net-snmp ar `--enable-blumenthal-aes` opciju, pēc tam pārkompilējiet Zabbix serveri, norādot opciju `--with-net-snmp=/home/user/yourcustomnetsnmp/bin/net-snmp-cor`

3. darbība

Izveidojiet vienumu uzraudzībai.

Tagad atgriezieties Zabbix un noklikšķiniet uz *Items* iepriekš izveidotajam SNMP hostam. Atkarībā no tā, vai, veidojot hostu, izmantojāt veidni, jums būs vai nu ar hostu saistīto SNMP vienumu saraksts, vai arī tukšs saraksts. Pieņemsim, ka vienumu izveidosiet pats, izmantojot informāciju, ko tikko ieguvāt ar snmpwalk un snmpget, tāpēc noklikšķiniet uz *Create item*.

Jaunā vienuma veidlapā aizpildiet obligātos parametrus:



Parameter	Description
Name	Ievadiet vienuma nosaukumu.
Type	Šeit atlasiet **SNMP agent**.
Key	Ievadiet jēgpilnu atslēgu.
Host interface	Noteikti atlasiet SNMP saskarni, piemēram, savai komutatora vai maršrutētāja ierīcei.

Parameter	Description
SNMP OID	OID vērtības ievadīšanai izmantojiet kādu no atbalstītajiem formātiem: **walk[OID1,OID2,...]** - izgūst 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 lielapjoma pieprasījumus (GetBulkRequest-PDU). Šī vienuma taimauta iestatījumus var konfigurēt vienuma konfigurācijas veidlapā. Apsveriet iespēju iestatīt nelielu taimauta vērtību, lai izvairītos no ilgām aizturēm, ja ierīce nav sasniedzama, jo tiks veikti līdz 5 atkārtoti mēģinājumi, ja iepriekšējie mēģinājumi beidzas ar taimautu vai neizdodas (piemēram, 3 sekunžu taimauts var izraisīt 15 sekunžu gaidīšanu). Varat izmantot šo vienumu kā galveno vienumu un izveidot atkarīgos vienumus, kas, izmantojot priekšapstrādi, izgūst datus no galvenā vienuma. Vienā snmp walk var norādīt vairākus OID, piemēram, `walk[OID1,OID2,...]`, lai asinhroni apstrādātu pa vienam OID. Ja lielapjoma pieprasījums neatgriež rezultātus, tiek mēģināts izgūt vienu ierakstu bez lielapjoma pieprasījuma. MIB nosaukumus var izmantot kā parametrus; 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]`, izvade būs apvienots saraksts. GetBulk pieprasījumi tiek izmantoti ar SNMPv2 un v3 saskarnēm, bet GetNext - ar SNMPv1 saskarnēm; maksimālais atkārtojumu skaits lielapjoma pieprasījumiem tiek konfigurēts saskarnes līmenī. Maksimālais atkārtojumu skaita parametrs ietekmē lielapjoma pieprasījumus, nosakot maksimālo vienā lielapjoma atbildē atgriežamo OID skaitu. Lielāka vērtība rada lielākas lielapjoma atbildes, samazinot nepieciešamo pārraižu skaitu. Tomēr ne visas ierīces var atbalstīt ļoti lielas vērtības, kas var radīt problēmas. Šis vienums atgriež snmpwalk utilītas izvadi ar parametriem -Oe -Ot -On. Šo vienumu varat izmantot kā galveno vienumu SNMP atklāšanā. **get[OID]** - asinhroni izgūst *vienu* vērtību. Piemēram: `get[1.3.6.1.2.1.31.1.1.1.6.3]` Šī vienuma taimauta iestatījumus var konfigurēt vienuma konfigurācijas veidlapā. Apsveriet iespēju iestatīt nelielu taimauta vērtību, lai izvairītos no ilgām aizturēm, ja ierīce nav sasniedzama, jo tiks veikti līdz 5 atkārtoti mēģinājumi, ja iepriekšējie mēģinājumi beidzas ar taimautu vai neizdodas (piemēram, 3 sekunžu taimauts var izraisīt 15 sekunžu gaidīšanu). **OID** - (mantots) ievadiet vienu teksta vai skaitlisku OID, lai sinhroni izgūtu vienu vērtību, pēc izvēles apvienojot to ar citām vērtībām. Piemēram: `1.3.6.1.2.1.31.1.1.1.6.3`. Šai opcijai vienuma pārbaudes taimauts būs vienāds ar vērtību, kas iestatīta servera konfigurācijas failā. **Ieteicams** izmantot `walk[OID]` un `get[OID]` vienumus labākai veiktspējai. Visi `walk[OID]` un `get[OID]` vienumi tiek izpildīti asinhroni - nav nepieciešams saņemt atbildi uz vienu pieprasījumu, pirms tiek sākta cita pārbaude. Arī DNS atrisināšana notiek asinhroni. Asinhrono pārbaužu maksimālā vienlaicība ir 1000 (to nosaka MaxConcurrentChecksPerPoller). Asinhrono SNMP aptauju skaitu nosaka parametrs StartSNMPPollers. Ņemiet vērā, ka tīkla datplūsmas statistikai, ko atgriež jebkura no metodēm, cilnē *Preprocessing* jāpievieno darbība *Change per second*; pretējā gadījumā no SNMP ierīces saņemsiet kumulatīvo vērtību, 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

Vienums **walk[OID1,OID2,...]** ļauj izmantot vietējo SNMP funkcionalitāti lielapjoma pieprasījumiem (GetBulkRequest-PDU), kas ir pieejami SNMP 2. un 3. versijā.

GetBulk pieprasījums SNMP izpilda vairākus GetNext pieprasījumus un atgriež rezultātu vienā atbildē.
To var izmantot gan parastiem SNMP vienumiem, gan SNMP atklāšanai, lai samazinātu tīkla turp-atpakaļ pieprasījumu skaitu.

SNMP vienumu **walk[OID1,OID2,...]** var izmantot kā galveno vienumu, kas vienā pieprasījumā apkopo datus, savukārt atkarīgie vienumi pēc vajadzības analizē atbildi, izmantojot priekšapstrādi.

Ņemiet vērā, ka vietējo SNMP lielapjoma pieprasījumu izmantošana nav saistīta ar SNMP pieprasījumu apvienošanas opciju, kas ir Zabbix 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` taimauts (iestatīts vienuma konfigurācijas veidlapā) tiek piemērots visai sesijai. Taimauts tiek piemērots neatkarīgi no tā, vai dati ir izgūti pilnībā; ja dati tiek saņemti daļēji (piemēram, dati ir veiksmīgi apkopoti tikai vienam no vairākiem OID), vienums kļūst neatbalstīts ar ziņojumu ”Only partial data received”.
Ja tiek sasniegts taimauts, tiks veikts atkārtots mēģinājums, taimauts tiks atiestatīts un pēdējais pieprasījums tiks nosūtīts atkārtoti, ļaujot sesijai turpināties no pēdējā pieprasījuma, ja viena pakete tiek pazaudēta vai pienāk pārāk vēlu.
Apsveriet iespēju iestatīt zemu taimauta vērtību, lai izvairītos no ilgstošas aizkaves, ja ierīce nav sasniedzama, jo tiks veikti līdz 5 atkārtotiem mēģinājumiem, ja iepriekšējie mēģinājumi beidzas ar taimautu vai neizdodas (piemēram, 3 sekunžu taimauts var izraisīt 15 sekunžu gaidīšanas laiku).

Kombinētās apstrādes iekšējā darbība

Zabbix serveris un starpniekserveris vienā pieprasījumā var vaicāt SNMP ierīcēm vairākas vērtības. Tas ietekmē vairākus SNMP vienumu veidus:

- parastie SNMP vienumi
- SNMP vienumi ar dinamiskiem indeksiem
- SNMP zema līmeņa atklāšanas noteikumi

Visi SNMP vienumi vienā saskarnē ar identiskiem parametriem tiek ieplānoti vaicāšanai vienlaikus. Pirmo divu veidu vienumus polleri paņem partijās, kurās ir ne vairāk kā 128 vienumi, savukārt zema 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 apstaigāšana.

”Iegūšanai” tiek izmantots GetRequest-PDU ar ne vairāk kā 128 mainīgo piesaistēm. ”Apstaigāšanai” SNMPv1 tiek izmantots GetNextRequest-PDU, bet SNMPv2 un SNMPv3 tiek izmantots GetBulkRequest ar lauku ”max-repetitions”, kura vērtība nepārsniedz 128.

Tādējādi kombinētās apstrādes priekšrocības katram SNMP vienumu veidam ir šādas:

- parastie SNMP vienumi izmanto ”iegūšanas” uzlabojumus;

- SNMP vienumi ar dinamiskiem indeksiem izmanto gan ”iegūšanas”, gan ”apstaigāšanas” uzlabojumus: ”iegūšana” tiek izmantota indeksa pārbaudei, bet ”apstaigāšana” - kešatmiņas izveidei;
- SNMP zema līmeņa atklāšanas noteikumi izmanto ”apstaigāšanas” uzlabojumus.

Tomēr pastāv tehniska problēma: ne visas ierīces spēj vienā pieprasījumā atgriezt 128 vērtības. Dažas ierīces vienmēr atgriež pareizu atbildi, bet citas vai nu atbild ar kļūdu ”tooBig(1)”, vai arī vispār neatbild, tiklīdz iespējamās atbildes apjoms pārsniedz noteiktu ierobežojumu.

Lai atrastu konkrētai ierīcei optimālo vaicājamo objektu skaitu, Zabbix izmanto šādu stratēģiju. Sākumā tas piesardzīgi pieprasa 1 vērtību vienā pieprasījumā. Ja tas izdodas, tiek pieprasītas 2 vērtības vienā pieprasījumā. Ja arī tas izdodas, tiek pieprasītas 3 vērtības vienā pieprasījumā, un pēc tam process turpinās līdzīgi, reizinot vaicājamo objektu skaitu ar 1,5. Tādējādi tiek iegūta šāda pieprasījumu izmēru secība: 1, 2, 3, 4, 6, 9, 13, 19, 28, 42, 63, 94, 128.

Tomēr, tiklīdz ierīce atsakās sniegt pareizu atbildi, piemēram, 42 mainīgajiem, Zabbix veic divas darbības.

Pirmkārt, pašreizējai vienumu partijai tas samazina objektu skaitu vienā pieprasījumā uz pusi un pieprasa 21 mainīgo. Ja ierīce darbojas, vaicājumam vajadzētu izdoties 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 vaicājums neizdodas, ierīce noteikti neatbild un pieprasījuma izmērs nav problēmas cēlonis.

Otrkārt, nākamajām vienumu partijām Zabbix sāk ar pēdējo veiksmīgo mainīgo skaitu (mūsu piemērā - 28) un turpina palielināt pieprasījumu izmēru par 1, līdz tiek sasniegts ierobežojums. Piemēram, pieņemot, ka lielākais atbildes izmērs ir 32 mainīgie, nākamo pieprasījumu izmēri būs 29, 30, 31, 32 un 33. Pēdējais pieprasījums neizdosies, un Zabbix vairs nekad neizsniegs pieprasījumu ar 33 mainīgajiem. No šī brīža Zabbix šai ierīcei vaicās ne vairāk kā 32 mainīgos.

Ja lieli vaicājumi ar šādu mainīgo skaitu neizdodas, tam var būt viens no diviem iemesliem. Precīzus kritērijus, ko ierīce izmanto atbildes izmēra ierobežošanai, nav iespējams zināt, taču mēs mēģinām tos aptuveni noteikt, izmantojot mainīgo skaitu. Pirmā iespēja ir tāda, ka šis mainīgo skaits vispārīgā gadījumā ir aptuveni vienāds ar ierīces faktisko atbildes izmēra ierobežojumu: dažreiz atbilde ir mazāka par ierobežojumu, bet dažreiz - lielāka. Otrā iespēja ir tāda, ka UDP pakete vienā no virzieniem vienkārši tika pazaudēta. Šo iemeslu dēļ, ja Zabbix saņem neveiksmīgu atbildi uz vaicājumu, tas samazina maksimālo mēģināmo mainīgo skaitu, lai nonāktu dziļāk ierīces komforta diapazonā, taču ne vairāk kā divas reizes.

Iepriekš minētajā piemērā, ja vaicājums ar 32 mainīgajiem nejauši neizdodas, Zabbix samazina skaitu līdz 31. Ja arī tas neizdodas, Zabbix samazina skaitu līdz 30. Tomēr Zabbix nesamazinās skaitu zem 30, jo pieņems, ka turpmākās kļūmes izraisa pazaudētas UDP paketes, nevis ierīces ierobežojums.

Ja ierīce citu iemeslu dēļ nespēj pareizi apstrādāt kombinētos pieprasījumus un iepriekš aprakstītā heiristika nedarbojas, katrai saskarnei ir pieejams iestatījums ”Use combined requests”, kas ļauj atspējot kombinētos pieprasījumus šai ierīcei.

Ja kombinētie pieprasījumi izraisa daļējas vai nepareizi formatētas atbildes, kuru rezultātā tiek veikti nepareizi aprēķini sekundē (delta), piemēram, šķietami lieli saskarnes skaitītāju lēcieni, atspējojiet *Use combined requests* ietekmētajai saskarnei, lai piespiestu veikt atsevišķus vaicājumus katram vienumam; tas bieži novērš kļūdainus lēcienus. Varat arī apsvērt asinhrono `get[]` vai `walk[]` vienumu izmantošanu. Tie tiek izpildīti asinhroni un uz tiem neattiecas katras saskarnes *Use combined requests* pakešu apstrāde - tos var izmantot mantoto sinhrono OID pārbaužu vietā, lai izvairītos no ar kombinētajiem pieprasījumiem saistītām problēmām. Lai identificētu ietekmētās ierīces, skatiet servera/starpniekservera žurnāla ierakstus, kas līdzīgi sadaļā Pārskats parādītajam ierakstam.

Turklāt, ja saskarne bieži kļūst nepieejama, var būt nepieciešams palielināt parametra `UnavailableDelay` vērtību Zabbix servera vai Zabbix starpniekservera konfigurācijas failos, lai samazinātu pieprasījumu biežumu. Vienumi var kļūt neatbalstīti, ja atklāšanas vai OID koku apstaigāšanas laikā tiek saņemti daļēji dati.

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>[”indekss”,”<indeksa bāzes OID>”,”<meklējamā virkne>”]

Parametrs	Apraksts
Datu OID	Galvenais OID, ko izmantot datu izgūšanai vienumā.
indekss	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.

Īpašie OID

Daži no visbiežāk izmantotajiem SNMP OID tiek automātiski pārvērsti skaitliskā attēlojumā, izmantojot Zabbix. Piemēram, **ifIndex** tiek pārvērsts par **1.3.6.1.2.1.2.2.1.1**, bet **ifIndex.0** tiek pārvērsts par **1.3.6.1.2.1.2.2.1.1.0**.

Tabulā ir uzskaitīti īpašie OID.

Special OID	Identifier	Description
ifIndex	1.3.6.1.2.1.2.2.1.1	Unikāla vērtība katram interfeisam.
ifDescr	1.3.6.1.2.1.2.2.1.2	Teksta virkne ar informāciju par interfeisu. Šajā virknē jāietver ražotāja nosaukums, produkta nosaukums un aparatūras interfeisa versija.
ifType	1.3.6.1.2.1.2.2.1.3	Interfeisa 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	Lielākā datagramma, ko var nosūtīt/saņemt interfeisā, norādīta oktetos.
ifSpeed	1.3.6.1.2.1.2.2.1.5	Interfeisa pašreizējā joslas platuma novērtējums bitos sekundē.
ifPhysAddress	1.3.6.1.2.1.2.2.1.6	Interfeisa adrese protokola slānī, kas protokolu stekā atrodas tieši ‘zem’ tīkla slāņa.
ifAdminStatus	1.3.6.1.2.1.2.2.1.7	Interfeisa pašreizējais administratīvais stāvoklis.
ifOperStatus	1.3.6.1.2.1.2.2.1.8	Interfeisa pašreizējais darbības stāvoklis.
ifInOctets	1.3.6.1.2.1.2.2.1.10	Kopējais interfeisā saņemto oktetu skaits, ieskaitot ietvara rakstzīmes.
ifInUcastPkts	1.3.6.1.2.1.2.2.1.11	Apakštīkla vienraides pakotņu skaits, kas piegādātas augstāka līmeņa protokolam.
ifInNUcastPkts	1.3.6.1.2.1.2.2.1.12	Vienraides neesošo pakotņu (t. i., apakštīkla apraides vai apakštīkla multiraides) skaits, kas piegādātas augstāka līmeņa protokolam.

Special OID	Identifier	Description
ifInDiscards	1.3.6.1.2.1.2.2.1.13	Ienākošo pakotņu skaits, kas tika izvēlētas atmešanai, lai gan netika konstatētas kļūdas, lai novērstu to piegādi augstāka līmeņa protokolam. Viens no iespējamiem šādas pakotnes atmešanas iemesliem var būt bufera vietas atbrīvošana.
ifInErrors	1.3.6.1.2.1.2.2.1.14	Ienākošo pakotņu skaits, kurās bija kļūdas, kas neļāva tās piegādāt augstāka līmeņa protokolam.
ifInUnknownProtos	1.3.6.1.2.1.2.2.1.15	Pakotņu skaits, kas saņemtas, izmantojot interfeisu, un tika atmestas nezināma vai neatbalstīta protokola dēļ.
ifOutOctets	1.3.6.1.2.1.2.2.1.16	Kopējais no interfeisa nosūtīto oktetu skaits, ieskaitot ietvara rakstzīmes.
ifOutUcastPkts	1.3.6.1.2.1.2.2.1.17	Kopējais pakotņu skaits, ko augstāka līmeņa protokoli pieprasīja nosūtīt un kas 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 pakotņu skaits, ko augstāka līmeņa protokoli pieprasīja nosūtīt un kas 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 pakotņu skaits, kas tika izvēlētas atmešanai, lai gan netika konstatētas kļūdas, lai novērstu to nosūtīšanu. Viens no iespējamiem šādas pakotnes atmešanas iemesliem var būt bufera vietas atbrīvošana.
ifOutErrors	1.3.6.1.2.1.2.2.1.20	Izejošo pakotņu skaits, kuras nevarēja nosūtīt kļūdu dēļ.
ifOutQLen	1.3.6.1.2.1.2.2.1.21	Izejošo pakotņu rindas garums (pakotnēs).

MIB faili

Ievads

MIB ir saīsinājums no Management Information Base (pārvaldības informācijas bāze). MIB faili ļauj izmantot OID (objektu identifikatoru) teksta attēlojumu. Uzraugot SNMP ierīces ar Zabbix, var izmantot neapstrādātus OID, taču, ja dodat priekšroku teksta attēlojumam, 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.

SNMP trap

Pārskats

SNMP trapu saņemšana ir pretējs process SNMP iespējotu ierīču vaicāšanai.

Šajā gadījumā informācija tiek nosūtīta no SNMP iespējotas ierīces uz `snmptrapd` un tiek savākta jeb "notverta" no faila, ko nolasa Zabbix serveris vai Zabbix starpniekserveris.

Parasti trapi tiek nosūtīti, kad mainās kāds nosacījums, un aģents savienojas ar serveri 162. portā (pretstatā 161. portam aģenta pusē, kas tiek izmantots vaicājumiem). Trapu izmantošana var palīdzēt noteikt īslaicīgas problēmas, kas rodas starp vaicājumu intervāliem un var palikt nepamanītas vaicājumu datos.

SNMP trapu saņemšana Zabbix ir izstrādāta darbam ar `snmptrapd` un vienu no mehānismiem trapu nodošanai Zabbix - vai nu Bash, vai Perl skriptu, vai SNMPTT.

Note:
Vienkāršākais veids, kā pēc Zabbix konfigurēšanas iestatīt trapu uzraudzību, ir izmantot Bash skripta risinājumu, jo Perl un SNMPTT mūsdienu distribūcijās bieži nav pieejami un tiem nepieciešama sarežģītāka konfigurācija. Tomēr šis risinājums izmanto skriptu, kas konfigurēts kā `traphandle`. Labākai veiktspējai ražošanas sistēmās izmantojiet iebūvēto Perl risinājumu (vai nu skriptu ar do `perl` opciju, vai SNMPTT).

Trapa saņemšanas darbplūsma:

1. `snmptrapd` saņem trapu.
2. `snmptrapd` nodod trapu saņēmēja skriptam (Bash, Perl) vai SNMPTT.
3. Saņēmējs parsē, formatē un ieraksta trapu failā.
4. Zabbix SNMP trapper nolasa un parsē trapu failu.
5. Katram trapam Zabbix atrod visus SNMP trapper vienumus ar host interfeisiem, kas atbilst saņemtā trapa adresei. Ņemiet vērā, ka atbilstības noteikšanai tiek izmantots tikai izvēlētais *IP* vai *DNS* host interfeisā.
6. Katram atrastajam vienumam trapis tiek salīdzināts ar regexp `snmptrap[regexp]`. Trapis tiek iestatīts kā vērtība **visiem** atbilstošajiem vienumiem. Ja netiek atrasts neviens atbilstošs vienums un ir `snmptrap.fallback` vienums, trapis tiek iestatīts kā tā vērtība.
7. Ja trapis nav iestatīts kā neviena vienuma vērtība, Zabbix pēc noklusējuma reģistrē neatbilstošo trapu 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 trapu konfigurēšana

Šim vienuma tipam ir nepieciešama šāda lietotāja saskarnes konfigurācija:

1. darbība: izveidojiet SNMP interfeisu hostam

1. Sadaļā *Data collection* > *Hosts* izveidojiet hostu vai rediģējiet esošo.
2. Laukā *Interfaces* atlasiet *SNMP* interfeisa tipu.
3. Ievadiet IP adresi/DNS nosaukumu un porta numuru.
4. Nolaižamajā sarakstā atlasiet *SNMP version*: (*SNMPv1*, *SNMPv2* vai *SNMPv3*) un pievienojiet interfeisa akreditācijas datus atkarībā no atlasītās SNMP versijas.

Adrese no katras saņemtās trapas tiks salīdzināta ar visu SNMP interfeisu IP un DNS adresēm, lai atrastu atbilstošos hostus.

2. darbība: konfigurējiet vienumu

1. Hostam izveidojiet vienumu vai rediģējiet esošo.
2. Laukā *Key* izmantojiet vienu no SNMP trap atslēgām:
   - `snmptrap[regexp]` - uztver visas SNMP trapas, kas atbilst `regexp` parametrā norādītajai regulārajai izteiksmei; ja `regexp` nav norādīts, uztver jebkuru trapu.
 Parametrā tiek atbalstīti lietotāja makrosi un globālās regulārās izteiksmes.
 Atgrieztā vērtība: SNMP trapa.
 Šo vienumu var iestatīt tikai SNMP interfeisiem.
   - `snmptrap.fallback` - uztver visas SNMP trapas, kuras nav uztvēris neviens no šī interfeisa `snmptrap[]` vienumiem.
 Atgrieztā vērtība: SNMP trapa.
 Šo vienumu var iestatīt tikai SNMP interfeisiem.

> **Note:**
> Pašlaik vairāku rindu regulāro izteiksmju atbilstība netiek atbalstīta.

3. Iestatiet *Type of information* uz *Log*, lai tiktu parsēti laikspiedoli. Arī citi formāti, piemēram, *Numeric*, ir pieņemami, taču tiem var būt nepieciešams pielāgots trapu apstrādātājs.

SNMP trap monitoringa iestatīšana

Zabbix servera/starpniekservera konfigurēšana

Lai nolasītu trapus, Zabbix serveris vai starpniekserveris ir jākonfigurē tā, lai tas palaistu SNMP trapper procesu un norādītu uz trap failu, ko raksta SNMPTT vai Bash/Perl trapu saņēmējs. 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`, šis fails, visticamāk, nedarbosies `/tmp`.

Bash trapu saņēmēja konfigurēšana

Prasības: tikai snmptrapd.

Bash trapu saņēmēja skriptu var izmantot, lai nodotu trapus Zabbix serverim no snmptrapd, izmantojot trapper failu. Lai to konfigurētu, pievienojiet opciju `traphandle` snmptrapd konfigurācijas failam (`snmptrapd.conf`), skatiet piemēru.

> **Note:**
> Var būt nepieciešams restartēt snmptrapd, lai tas ielādētu konfigurācijas izmaiņas.

Perl trapu uztvērēja konfigurēšana

Prasības: Perl, Net-SNMP, kas kompilēts ar `--enable-embedded-perl` (pēc noklusējuma tas ir izdarīts kopš Net-SNMP 5.4)

Perl trapu uztvērēju (meklējiet `misc/snmptrap/zabbix_trap_receiver.pl`) var izmantot, lai nodotu trapus uz Zabbix serveris 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 stātos spēkā konfigurācijas izmaiņas, snmptrapd, iespējams, būs jārestartē.

> **Note:**
> Ja skripta nosaukums nav ievietots pēdiņās, snmptrapd atteiksies startēties un parādīs ziņojumus, kas 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 ir jākonfigurē, lai tas izmantotu SNMPTT.

> **Note:**
> Lai nodrošinātu vislabāko veiktspēju, SNMPTT jākonfigurē kā dēmons, izmantojot `snmptthandler-embedded`, lai tam nodotu trapes. Skatiet norādījumus par SNMPTT konfigurēšanu.

Kad SNMPTT ir konfigurēts trapju saņemšanai, konfigurējiet `snmptt.ini`:

1. Iespējojiet Perl moduļa izmantošanu no `NET-SNMP` pakotnes:

```
net_snmp_perl_enable = 1
```

2. Reģistrējiet trapes trapju 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` ir noņemta RHEL 8.0-8.2; tā tika atkārtoti pievienota RHEL 8.3. Papildinformāciju skatiet zināmo problēmu sadaļā.

Tagad noformējiet trapes tā, lai Zabbix tās atpazītu (rediģējiet `snmptt.conf`):

1. Katram `FORMAT` paziņojumam jāsākas ar `ZBXTRAP [address]`, kur `[address]` tiks salīdzināta ar SNMP saskarņu IP un DNS adresēm Zabbix. Piem.:

```
EVENT coldStart .1.3.6.1.6.3.1.1.5.1 "Status Events" Normal
FORMAT ZBXTRAP $aA Device reinitialized (coldStart)
```

2. Skatiet tālāk vairāk par SNMP trapju formātu.

Attention:
Neizmantojiet nezināmas trapes - Zabbix tās nespēs atpazīt. Nezināmas trapes var apstrādāt, definējot vispārīgu notikumu `snmptt.conf` failā:


```
EVENT general .* "General event" Normal
```

SNMP trapa formāts

Visiem pielāgotajiem Perl trapa uztvērējiem un SNMPTT trapa konfigurācijai trapa formāts ir jāveido šādi:

```
[timestamp] [the trap, part 1] ZBXTRAP [address] [the trap, part 2]
```

kur

- `[timestamp]` - laikspiedols `%Y-%m-%dT%H:%M:%S%z` formātā.
- `ZBXTRAP` - galvene, kas norāda, ka šajā rindā sākas jauns trapa ieraksts.
- `[address]` - IP adrese, ko izmanto, lai atrastu šī trapa hostu.

Ņemiet vērā, ka `ZBXTRAP` un `[address]` apstrādes laikā tiks izgriezti no ziņojuma. Ja trapa formāts ir citāds, Zabbix var parsēt trapus negaidītā veidā.

Trapa 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
```

Tas SNMP saskarnei ar IP=`192.168.1.1` rezultēsies šādā trapā:

```
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 nodrošina lielu failu atbalstu 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 žurnālfailu rotācijas sistēmu - tā ir jānodrošina lietotājam. Žurnālfailu rotācijai vispirms ir jāpārdēvē vecais fails un tikai pēc tam tas jādzēš, lai netiktu zaudēti nekādi slazdi:

1. Zabbix atver trap failu pēdējā zināmajā atrašanās vietā un pāriet uz 3. soli.
2. Zabbix pārbauda, vai pašlaik atvērtais fails ir ticis rotēts, salīdzinot inode numuru ar definētā trap 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 tiek atgriezts 2. solī.
5. Ja jaunu datu nav, Zabbix guļ 1 sekundi un atgriežas 2. solī.

Failu sistēma

Trap faila implementācijas dēļ Zabbix failu sistēmai ir jāatbalsta 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 nosūtītu traperus uz Zabbix serveri.

Iestatīšana:

1. Konfigurējiet Zabbix, lai palaistu SNMP traperi, un iestatiet traperu failu. Pievienojiet failam `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/temp
```

Ja nepieciešams, pielāgojiet skripta mainīgo ZABBIX_TRAPS_FILE. Lai izmantotu noklusējuma vērtību, vispirms izveidojiet vecākelementa direktoriju:

```
mkdir -p /var/lib/zabbix/snmptraps
```

3. Pievienojiet failam snmtrapd.conf tālāk norādīto (skatiet darba piemēru):

```
traphandle default /bin/bash /usr/sbin/zabbix_trap_handler.sh
```

Note:
Lai snmptrapd pārņemtu konfigurācijas izmaiņas, tas, iespējams, ir jārestartē.

4. Izveidojiet SNMP vienumu TEST (ņemiet vērā sākotnējās konfigurācijas prasības):

   Type: *SNMP trap*
 Type of information: *Log*
 Host interface: *SNMP 127.0.0.1*
 Key: snmptrap["linkUp"]
 Log time format: yyyyMMdd.hhmmss

Ņemiet vērā, ka tiek izmantots ISO 8601 datuma un laika formāts.

5. Pēc tam konfigurēsim snmptrapd izvēlētajai SNMP protokola versijai un nosūtīsim testa traperus, izmantojot utilītu snmptrap.

SNMPv1, SNMPv2

SNMPv1 un SNMPv2 protokoli autentifikācijai izmanto "kopienas virkni". Tālāk redzamajā piemērā kā kopienas virkni izmantosim "secret". SNMP traperu sūtītājiem ir jābūt iestatītai tai pašai vērtībai.

Ņemiet vērā, ka, lai gan SNMPv2 joprojām tiek plaši izmantots ražošanas vidēs, tas nenodrošina šifrēšanu un reālu sūtītāja autentifikāciju. Dati tiek nosūtīti vienkāršā tekstā, tāpēc šīs protokola versijas drīkst izmantot tikai drošās vidēs, piemēram, privātā tīklā, un tās nekādā gadījumā nedrīkst izmantot publiskā vai trešās puses tīklā.

SNMP 1. versija mūsdienās praktiski netiek izmantota, jo tā neatbalsta 64 bitu skaitītājus un tiek uzskatīta par mantotu protokolu.

Lai iespējotu SNMPv1 vai SNMPv2 traperu pieņemšanu, pievienojiet failam snmptrapd.conf tālāk norādīto rindu. Aizstājiet secret ar SNMP kopienas virkni, kas konfigurēta SNMP traperu sūtītājiem:

```
authCommunity log,execute,net secret
```

Pēc tam varam nosūtīt testa traperi, izmantojot snmptrap. Šajā piemērā izmantosim izplatīto "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, kā arī DES vai vairākus AES šifrus.

Konfigurējiet snmptrapd, lai tas pieņemtu SNMPv3 traperus, pievienojot failam snmptrapd.conf tālāk norādītās rindas:

```
##### Create an SNMPv3 USM user with authentication and privacy
createUser -e <engineID> <securityName> SHA <authKey> AES <privKey>
##### Allow the user to log traps and execute trap handler scripts
authUser log,execute <securityName>
```

Piemēram:

```
createUser -e 0x8000000001020304 traptest SHA myAuthPassword AES myPrivPassword
authUser log,execute traptest
```

Attention:
Ņemiet vērā atslēgvārdu "execute", kas ļauj šim lietotāja drošības modelim izpildīt skriptus.

Lai pārbaudītu SNMPv3 traperu konfigurāciju, nosūtiet testa traperi, izmantojot tālāk norādīto komandu:

```
snmptrap -v 3 -n "" -a SHA -A myAuthPassword -x AES -X myPrivPassword -l authPriv -u traptest -e 0x8000000
```

Warning:
Ja vēlaties izmantot tādas spēcīgas šifrēšanas metodes kā AES192 vai AES256, izmantojiet net-snmp, sākot ar 5.8. versiju. Iespējams, tas būs jākompilē atkārtoti ar configure opciju: --enable-blumenthal-aes. Vecākas net-snmp versijas neatbalsta AES192/AES256. Skatiet arī: Strong Authentication or Encryption.

Pārbaude

Abos piemēros failā `/var/lib/zabbix/snmptraps/snmptraps.log` būs redzamas 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)

Zabbix iekšējais

Pārskats

Iekšējās pārbaudes ļauj pārraudzīt Zabbix iekšējos procesus.
Citiem vārdiem, varat pārraudzīt, kas notiek ar Zabbix serveri vai Zabbix starpniekserveri.

Iekšējās pārbaudes tiek aprēķinātas:

- Zabbix serverī — ja hosts tiek pārraudzīts ar serveri
- Zabbix starpniekserverī — ja hosts tiek pārraudzīts ar starpniekserveri

Serveris vai starpniekserveris apstrādā iekšējās pārbaudes neatkarīgi no hosta uzturēšanas statusa.

Lai izmantotu šo vienumu, izvēlieties vienuma tipu **Zabbix internal**.

> **Note:**
> Iekšējās pārbaudes apstrādā Zabbix aptauju veicēji.

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[apm,]	Telemetrijas datu, ko saņem APM kolektors, statistika.
zabbix[boottime]	Zabbix servera vai Zabbix starpniekservera procesa palaišanas laiks sekundēs.
zabbix[cluster,discovery,nodes]	Atklāj augstas pieejamības klastera mezglus.
zabbix[connector_queue]	Savienotāja rindā ievietoto vērtību skaits.
zabbix[discovery_queue]	Atklāšanas rindā ievietoto tīkla pārbaužu skaits.
zabbix[host,items]	Hosts iespējoto vienumu (atbalstīto un neatbalstīto) skaits.
zabbix[host,items_unsupported]	Hosts iespējoto neatbalstīto vienumu skaits.
zabbix[host,maintenance]	Pašreizējais hosts uzturēšanas statuss.
zabbix[host,active_agent,available]	Aktīvā aģenta pārbaužu pieejamība hosts.
zabbix[host,discovery,interfaces]	Detalizēta informācija par visām konfigurētajām hosts saskarnēm Zabbix lietotāja saskarnē.
zabbix[host,available]	Konkrēta tipa pārbaužu galvenās saskarnes pieejamība hosts.
zabbix[hosts]	Uzraudzīto hostu skaits.
zabbix[items]	Iespējoto vienumu (atbalstīto un neatbalstīto) skaits.
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]	Priekšapstrādes pārvaldnieka saņemto vērtību statistika.
zabbix[preprocessing_queue]	Priekšapstrādes rindā ievietoto vērtību skaits.
zabbix[process,,]	Laika procentuālā daļa, ko konkrēts Zabbix process vai procesu grupa (identificēta ar <type> un <mode>) pavadīja stāvoklī <state>.
zabbix[proxy,]	Informācija par Zabbix starpniekserveri.
zabbix[proxy,discovery]	Zabbix starpniekserveru saraksts.
zabbix[proxy group,available]	Tiešsaistes starpniekserveru skaits starpniekserveru grupā.
zabbix[proxy group,pavailable]	Tiešsaistes 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 un reāllaika datiem.
zabbix[proxy_buffer,buffer,]	Atgriež starpniekservera atmiņas bufera lietojuma statistiku.
zabbix[proxy_buffer,state,changes]	Atgriež stāvokļa izmaiņu skaitu starp diska un atmiņas bufera režīmiem kopš palaišanas.
zabbix[proxy_buffer,state,current]	Atgriež pašreizējo darbības stāvokli, kurā tiek glabāti jaunie dati.
zabbix[proxy_history]	Starpniekservera vēstures tabulā esošo vērtību skaits, kas gaida nosūtīšanu serverim.
zabbix[queue,]	Uzraudzīto vienumu skaits rindā, kuri aizkavējas vismaz par <from> sekundēm, bet mazāk nekā par <to> sekundēm.
zabbix[rcache,]	Zabbix konfigurācijas kešatmiņas pieejamības statistika.
zabbix[requiredperformance]	Nepieciešamā Zabbix servera vai Zabbix starpniekservera veiktspēja, izteikta kā paredzamais jauno vērtību skaits sekundē.
zabbix[stats,]	Atgriež Zabbix servera vai starpniekservera iekšējos rādītājus. Ja ir norādīti <ip> un <port>, rādītāji tiek iegūti no attālās instances; pretējā gadījumā — no lokālās instances.
zabbix[stats,,queue,]	Atgriež Zabbix servera vai starpniekservera iekšējos rindas rādītājus. Ja ir norādīti <ip> un <port>, rādītāji tiek iegūti no attālās instances; pretējā gadījumā — no lokālās instances.
zabbix[tcache,]	Zabbix tendenču funkcijas kešatmiņas efektivitātes statistika.
zabbix[triggers]	Zabbix datubāzē iespējoto trigeru skaits, ja visi vienumi iespējotajos hostos ir iespējoti.
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.

Vienuma atslēga	Apraksts
zabbix[vps,written]	Datubāzē ierakstīto vēstures vērtību kopējais skaits.
zabbix[wcache,,]	Zabbix ierakstīšanas 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>,avail`
- 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[apm,<type>]

 Telemetrijas datu statistika, ko saņem APM kolektors:

- *requests* - APM kolektora saņemto pieprasījumu skaits (skaitītājs):
  - *accepted* - apstrādei pieņemto pieprasījumu skaits;
  - *dropped* - atcelto pieprasījumu skaits;
- *rows* - datu avotā ierakstīto telemetrijas datu rindu skaits katram datu tipam (skaitītājs).

Atgrieztā vērtība: *JSON objekts*.

Parametri:

- **type** - *traces*, *metrics* vai *logs*. Ja nav norādīts, tiek atgriezta visu tipu statistika.

Komentāri:

- Šis vienums netiek atbalstīts Zabbix serverī.

Atgriezto vērtību piemērs `zabbix[apm]`:

```
{
  "requests": {
    "accepted": 15230,
    "dropped": 12
  },
  "rows": {
    "traces": 48710,
    "metrics_gauge": 31250,
    "metrics_sum": 18400,
    "metrics_histogram": 5120,
    "metrics_exponential_histogram": 640,
    "metrics_summary": 210,
    "logs": 22870
  }
}
```

zabbix[boottime]

 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` - sekunžu skaits kopš pēdējās mezgla sirdspuksta ziņas (`db_timestamp - lastaccess`)
  - `status` - mezgla stāvoklis:
    - 0 - gaidstāve
    - 1 - manuāli apturēts
    - 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 pārbaudes tipa galvenās saskarnes pieejamība hostam.
 Atgrieztās vērtības: *0* - nav pieejams; *1* - pieejams; *2* - nezināms.

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()` trigeris funkciju.
- Otrajam parametram jābūt tukšam, un tas ir rezervēts turpmākai izmantošanai.

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 ar <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*, *apm manager*, *apm worker*, *availability manager*, *browser poller*, *configuration syncer*, *data sender*, *discovery manager*, *discovery worker*, *history syncer*, *housekeeper*, *http agent poller*, *http poller*, *icmp pinger*, *internal poller ipmi manager*, *ipmi poller*, *java poller*, *odbc poller*, *poller*, *preprocessing manager*, *preprocessing worker*, *self-monitoring*, *snmp poller*, *snmp trapper*, *task manager*, *trapper*, *unreachable poller*, *vmware collector*;
- **mode**:
  - *avg* - visu norādītā tipa procesu vidējā vērtība (noklusējums);
  - *count* - atgriež norādītā procesa tipa atzarojumu skaitu; <state> nedrīkst būt norādīts;
  - *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 aizņemts, piemēram, apstrādā pieprasījumu (noklusējums);
  - *idle* - process ir dīkstāvē 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.
- Minimums un maksimums attiecas uz viena procesa lietojuma procentuālo daļu. Tātad, ja 3 poller procesu lietojuma procentuālās daļas ir 2, 18 un 66, min atgriezīs 2, bet max - 66.
- Procesi koplietojamajā atmiņā ziņo par savu darbību, un pašuzraudzības process šos datus apkopo katru sekundi. Stāvokļa izmaiņas (busy/idle) tiek reģistrētas izmaiņu brīdī - tādējādi process, kas kļūst aizņemts, tiek reģistrēts kā tāds un nemaina vai neatjaunina stāvokli, līdz kļūst dīkstāvē. Tas nodrošina, ka pat pilnībā iestrēguši procesi tiek pareizi reģistrēti kā 100% aizņemti.
- Pašlaik "busy" nozīmē "negulēt", taču nākotnē var tikt ieviesti papildu stāvokļi - gaidīšana uz slēdzenēm, datubāzes vaicājumu izpilde u. c. Ņemiet vērā, ka asinhronie poller procesi tiek uzskatīti par aizņemtiem, ja tie ir sasnieguši ierobežojumu, ko nosaka MaxConcurrentChecksPerPoller servera/starpniekservera konfigurācijas parametrs.
- 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 pēdējās minūtes laikā kaut ko darot pavadīji
zabbix[process,"history syncer",2,busy] #laiks, ko pēdējās minūtes laikā kaut ko darījis history syncer pr
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* - laika zīmogs pēdējam no starpniekservera saņemtajam sirdspukstu ziņojumam;
  - *delay* - cik ilgi apkopotās vērtības nav nosūtītas; tiek aprēķināts kā "starpniekservera aizkave" + ("pašreizējais servera laiks" - "starpniekservera lastaccess"), kur "starpniekservera aizkave" ir starpība starp pašreizējo starpniekservera laiku un vecākās nenosūtītās vērtības laika zīmogu starpniekserverī.

Komentāri:

- Šis vienums netiek atbalstīts hostos, kurus uzrauga starpniekserveris vai starpniekserveru grupa.
- Funkciju `fuzzytime()` var izmantot, lai pārbaudītu starpniekservera pieejamību.

Piemērs:

```
zabbix[proxy,"Germany",lastaccess] #laika zīmogs pēdējam no "Germany" starpniekservera saņemtajam sirdspuk
```

zabbix[proxy,discovery]

 Zabbix starpniekserveru saraksts ar nosaukumu, režīmu, šifrēšanu, saspiešanu, versiju, pēdējās saziņas laiku, hostu skaitu, vienumu skaitu, nepieciešamo vērtību skaitu sekundē (vps), versijas statusu (current/outdated/unsupported), taimautiem pēc vienumu tipa, starpniekserveru grupas nosaukumu (ja starpniekserveris pieder grupai), stāvokli (unknown/offline/online).
 Atgrieztā vērtība: *JSON objekts*.

Komentārs:

- Šis vienums netiek atbalstīts hostos, kurus uzrauga starpniekserveris vai starpniekserveru grupa.

Atgriezto vērtību piemērs:

```
[
  {
    "name":"Zabbix proxy",
    "passive":false,
    "unencrypted":true,
    "psk":false,
    "cert":false,
    "items":22,
    "compression":true,
    "version":"8.0.0beta2",
    "compatibility":1,
    "last_seen":1,
    "hosts":1,
    "requiredperformance":0.333333,
    "timeouts": {
        "zabbix_agent":3,
        "simple_check":3,
        "snmp_check":3,
        "external_check":3,
        "db_monitor":3,
        "http_agent":3,
        "ssh_agent":3,
        "telnet_agent":3,
        "script":3,
        "browser":60
    },
    "state":"online"
  }
]
```

zabbix[proxy group,<name>,available]

 Tiešsaistē esošo starpniekserveru skaits starpniekserveru grupā.
 Atgrieztā vērtība: *Integer*.

Parametri:

- **name** - starpniekserveru grupas nosaukums.

Komentārs:

- Šis vienums netiek atbalstīts hostos, kurus uzrauga starpniekserveris vai starpniekserveru grupa.

zabbix[proxy group,<name>,pavailable]

 Tiešsaistē esošo starpniekserveru procentuālā daļa starpniekserveru grupā.
 Atgrieztā vērtība: *Float*.

Parametri:

- **name** - starpniekserveru grupas nosaukums.

Komentārs:

- Šis vienums netiek atbalstīts hostos, kurus uzrauga starpniekserveris vai starpniekserveru grupa.

zabbix[proxy group,<name>,proxies]

 Zabbix starpniekserveru saraksts starpniekserveru grupā, kurā norādīts nosaukums, režīms, šifrēšana, saspiešana, versija, pēdējā saziņa, hostu skaits, vienumu skaits, nepieciešamās vērtības sekundē (vps), versijas statuss (pašreizējā/novecojusi/neatbalstīta), taimauti, starpniekserveru grupas nosaukums un stāvoklis (nezināms/bezsaistē/tiešsaistē).
 Atgrieztā vērtība: *JSON*.

Parametri:

- **name** - starpniekserveru grupas nosaukums.

Komentārs:

- Šis vienums netiek atbalstīts hostos, kurus uzrauga starpniekserveris vai starpniekserveru grupa.

zabbix[proxy group,<name>,state]

 Starpniekserveru grupas stāvoklis.
 Atgrieztā vērtība: *0* - nezināms; *1* - bezsaistē; *2* - atkopjas; *3* - tiešsaistē; *4* - stāvoklis pasliktinās.

Parametri:

- **name** - starpniekserveru grupas nosaukums.

Komentārs:

- Šis vienums netiek atbalstīts hostos, kurus uzrauga starpniekserveris vai starpniekserveru grupa.

zabbix[proxy group,discovery]

 Atgriež starpniekserveru grupu sarakstu ar konfigurācijas un reāllaika datiem. Konfigurācijas dati ietver starpniekserveru grupas nosaukumu, pārslēgšanās aizkavi un nepieciešamo tiešsaistē esošo starpniekserveru minimālo skaitu. Reāllaika dati ietver starpniekserveru grupas stāvokli (detalizētu informāciju skatiet komentāros), tiešsaistē esošo starpniekserveru skaitu un tiešsaistē esošo starpniekserveru procentuālo daļu.
 Atgrieztā vērtība: *JSON*.

Komentāri:

- Šis vienums netiek atbalstīts hostos, kurus uzrauga starpniekserveris vai starpniekserveru grupa.
- Šis vienums neatgriež starpniekserverus, kas nav nevienā grupā.
- Ja `failover_delay` vai `min_online` vērtība nav derīga, tiek ziņota īpaša vērtība −1, kas to norāda. Nederīgas vērtības var rasties, ja konfigurācijā tiek izmantotas makrokomandas un tās nevar izvērst par derīgu vērtību.
- Starpniekserveru grupas stāvoklis tiek ziņots kā vesels skaitlis: *0* - nezināms; *1* - bezsaistē; *2* - atkopjas; *3* - tiešsaistē; *4* - pasliktinās.

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 lietojuma 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:

- Ja atmiņas buferis ir atspējots, tiek atgriezta kļūda “Proxy memory buffer is disabled”;
- 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 darbības stāvokli, kurā tiek glabāti jaunie dati.
 Atgrieztās vērtības: *0* - disks; *1* - atmiņa.

Komentārs:

- 0 tiek atgriezts arī tad, ja atmiņas buferis ir atspējots.

zabbix[proxy_history]

 Vērtību skaits starpniekservera vēstures tabulā, kas gaida nosūtīšanu serverim.
 Atgrieztās vērtības: *Integer*.

Komentārs:

- 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** - aizkavēti vismaz par (noklusējums ir 6 sekundes);
- **to** - aizkavēti ne vairāk kā par (noklusējums ir bezgalība).

Komentāri:

- Laika sufiksi (`s,m,h,d,w`) ir atbalstīti parametros.

Attention:
Zabbix starpniekserveris nezina par apkopes periodiem; skatiet Rindu aprēķins apkopes laikā, lai iegūtu detalizētu informāciju.

zabbix[rcache,<cache>,<mode>]

 Zabbix konfigurācijas kešatmiņas pieejamības statistika.
 Atgrieztās vērtības: *Integer* (izmēram); *Float* (procentuālajai vērtībai).

Parametri:

- **cache** - *buffer*;
- **mode**:
  - *total* - bufera kopējais izmērs;
  - *free* - brīvā bufera izmērs;
  - *pfree* - brīvā bufera procentuālā vērtība;
  - *used* - izmantotā bufera izmērs;
  - *pused* - izmantotā bufera procentuālā vērtība.

zabbix[requiredperformance]

 Nepieciešamā Zabbix servera vai Zabbix starpniekservera veiktspēja, sagaidāmais jauno vērtību skaits sekundē.
 Atgrieztā vērtība: *Float*.

Komentāri:

- Aptuveni atbilst ”Required server performance, new values per second” sadaļā *Reports* > *System information*.

zabbix[stats,<ip>,<port>]

 Atgriež Zabbix servera vai starpniekservera iekšējās metrikas. Ja ir norādīti `<ip>` un `<port>`, metrikas tiek izgūtas no attālā eksemplāra; pretējā gadījumā - no lokālā eksemplāra.
 Atgrieztās vērtības: *JSON objekts*.

Parametri:

- **ip** - serveru/starpniekserveru IP/DNS/tīkla masku saraksts, kuriem veikt attālo vaicājumu (noklusējums ir 127.0.0.1);
- **port** - servera/starpniekservera ports, kuram veikt attālo vaicājumu (noklusējums ir 10051).

Piezīmes:

- stats pieprasījums tiks pieņemts tikai no adresēm, kas norādītas `StatsAllowedIP` serveris/starpniekserveris parametrā mērķa eksemplārā.
- Ar šo vienumu tiek atgriezts atlasīts iekšējo metriku kopums. Sīkāku informāciju skatiet sadaļā Attālā Zabbix stats uzraudzība.

zabbix[stats,<ip>,<port>,queue,<from>,<to>]

 Atgriež Zabbix servera vai starpniekservera iekšējās rindas metriku (skatiet `zabbix[queue,<from>,<to>]`). Ja ir norādīti `<ip>` un `<port>`, metrika tiek izgūta no attālā eksemplāra; pretējā gadījumā - no lokālā eksemplāra.
 Atgrieztās vērtības: *JSON objekts*.

Parametri:

- **ip** - serveru/starpniekserveru IP/DNS/tīkla masku saraksts, kuriem veic attālo vaicājumu (noklusējums ir `127.0.0.1`);
- **port** - servera/starpniekservera ports, kuram veic attālo vaicājumu (noklusējums ir 10051);
- **from** - aizkave vismaz par (noklusējums ir 6 sekundes);
- **to** - aizkave ne vairāk kā par (noklusējums ir bezgalība).

Piezīmes:

- stats pieprasījums tiks pieņemts tikai no adresēm, kas norādītas `StatsAllowedIP` serveris/starpniekserveris parametrā mērķa eksemplārā.
- Ar šo vienums tiek atgriezts atlasīts iekšējo metriku kopums. Sīkāku informāciju skatiet sadaļā Attālā Zabbix statistikas uzraudzība.

> **Attention:**
> Zabbix starpniekserveris nav informēts par apkopes periodiem; sīkāku informāciju skatiet sadaļā Rindu aprēķināšana apkopes laikā.

zabbix[tcache,cache,<parameter>]

 Zabbix tendenču funkcijas kešatmiņas efektivitātes statistika.
 Atgrieztās vērtības: *Integer* (izmēram); *Float* (procentuālai vērtībai).

Parametri:

- **parameter**:
  - *all* - kopējie kešatmiņas pieprasījumi (noklusējums);
  - *hits* - kešatmiņas trāpījumi;
  - *phits* - kešatmiņas trāpījumu procentuālā daļa;
  - *misses* - kešatmiņas kļūmes;
  - *pmisses* - kešatmiņas kļūmju procentuālā daļa;
  - *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.
  - *paccessed* - pēdējo 24 stundu laikā piekļūto ierakstu procentuālā daļa.

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* - bufera kopējais izmērs;
  - *free* - brīvā bufera izmērs;
  - *pfree* - brīvā bufera procentuālā daļa;
  - *used* - izmantotā bufera izmērs;
  - *pused* - izmantotā bufera procentuālā daļa.

Komentāri:

- Šis vienums nav 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* - normāls režīms; *1* - zema atmiņas režīms.

Parametri:

- **parameter**:
  - *requests* - kopējais pieprasījumu skaits;
  - *hits* - kešatmiņas trāpījumu skaits (no kešatmiņas ņemtās vēstures vērtības);
  - *misses* - kešatmiņas netrāpījumu skaits (no datubāzes ņemtās vēstures vērtības);
  - *mode* - vērtību kešatmiņas darbības režīms.

Komentāri:

- Kad zema atmiņas režīms ir ieslēgts, vērtību kešatmiņa paliks šajā stāvoklī 24 stundas, pat ja problēma, kas izraisīja šī režīma aktivizēšanu, tiks atrisināta ātrāk.
- Varat izmantot šo atslēgu kopā ar *Change per second* pirmapstrādes soli, lai iegūtu vērtības sekundē statistiku.
- Šis vienums netiek atbalstīts Zabbix starpniekserverī.

zabbix[version]

 Zabbix servera vai starpniekservera versija.
 Atgrieztā vērtība: *String*. Piemēram: `8.0.0`.

zabbix[vmware,buffer,<mode>]

 Zabbix vmware kešatmiņas pieejamības statistika.
 Atgrieztās vērtības: *Integer* (izmēram); *Float* (procentuālajai vērtībai).

Parametri:

- **mode**:
  - *total* - bufera kopējais 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* (procentiem).

Parametri:

- **cache** - *values*, *history*, *index* vai *trend*;
- **mode**:
  - ar *values* kešatmiņas parametru:
    - *all* - kopējais Zabbix servera/starpniekservera apstrādāto vērtību skaits, izņemot neatbalstītos vienumus (skaitītājs) (noklusējums);
    - *float* - apstrādāto peldošā komata vērtību skaits (skaitītājs);
    - *uint* - apstrādāto bezzīmes veselo skaitļu vērtību skaits (skaitītājs);
    - *str* - apstrādāto rakstzīmju/teksta 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 apstrāde rezultējās ar to, ka vienums kļuva neatbalstīts vai saglabāja šo stāvokli (skaitītājs);
  - ar *history*, *index*, *trend* kešatmiņas parametriem:
    - *pfree* - brīvā bufera procentuālā daļa (noklusējums);
    - *total* - bufera kopējais izmērs;
    - *free* - brīvā bufera izmērs;
    - *used* - izmantotā bufera izmērs;
    - *pused* - izmantotā bufera procentuālā daļa.

Komentāri:

- `trend` kešatmiņas parametrs nav atbalstīts ar Zabbix starpniekserveri.
- History kešatmiņa tiek izmantota vienumu vērtību glabāšanai. Zems skaitlis norāda uz veiktspējas problēmām datubāzes pusē.
- History index kešatmiņa tiek izmantota, lai indeksētu history kešatmiņā glabātās vērtības.
- Pēc tam, kad history kešatmiņa ir pilna un pēc tam notīrīta, history index kešatmiņa joprojām saglabās daļu datu. Šāda uzvedība ir paredzēta un palīdz sistēmai darboties efektīvāk, izvairoties no papildu apstrādes, kas būtu nepieciešama, lai pastāvīgi mainītu atmiņas izmēru.
- Trend kešatmiņa glabā tekošās stundas apkopojumu visiem vienumiem, kas saņem datus.
- Lai iegūtu vērtību sekundē statistiku, varat izmantot `zabbix[wcache,values]` atslēgu kopā ar priekšapstrādes soli *Change per second*.

Zabbix traperis

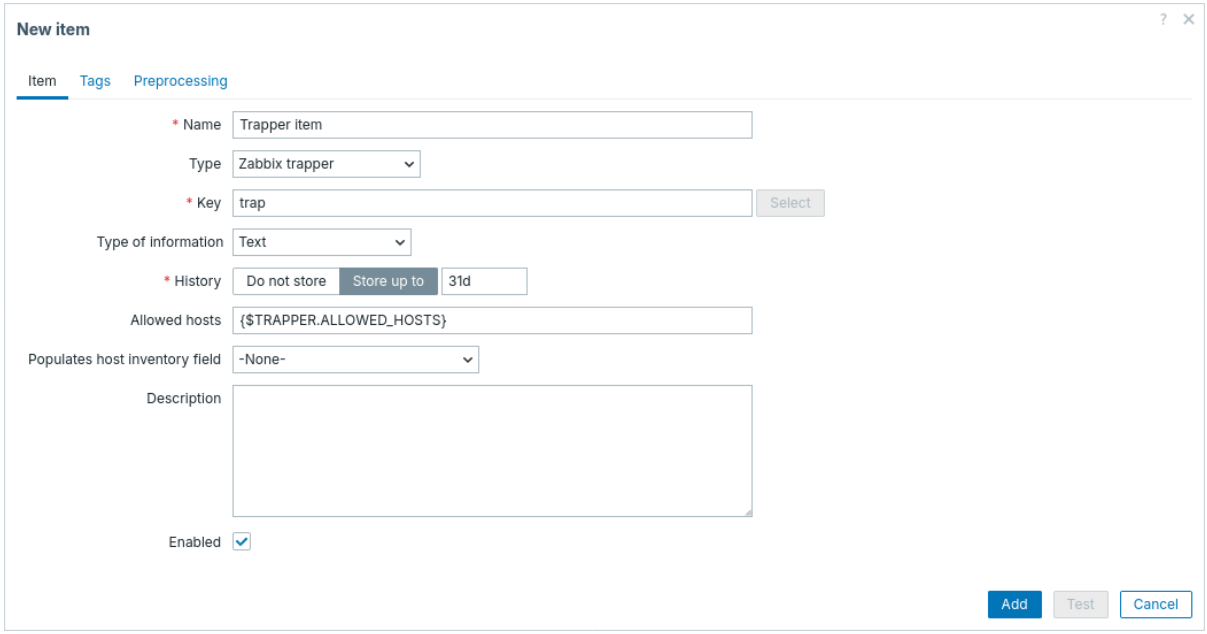
Pārskats

Trapper vienumi pieņem ienākošos datus, nevis vaicā tos. Tas ir noderīgi jebkuriem datiem, kurus vēlaties nosūtīt uz Zabbix.

Konfigurācija

Lai konfigurētu trapper vienumu:

1. Dodieties uz *Datu vākšana* > *Hosts*.
2. Noklikšķiniet uz *Vienumi* attiecīgā hosta rindā.
3. Noklikšķiniet uz *Izveidot vienumu*.
4. Konfigurācijas formā ievadiet vienuma parametrus.



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 vienuma atpazīšanai, sūtot datus uz Zabbix serveris.
Informācijas tips	Atlasiet informācijas tipu, kas atbildīs sūtāmo datu formātam.

Atļautie hosts	Ar komatiem atdalītu IP adrešu saraksts (pēc izvēles CIDR notācijā) vai DNS nosaukumi. Noklusējuma vērtība ir lietotāja makro `{$TRAPPER.ALLOWED_HOSTS}`. Pēc noklusējuma `{$TRAPPER.ALLOWED_HOSTS}` ir `127.0.0.1,::1`. Jūs varat vai nu ievadīt adreses tieši laukā, vai mainīt makro vērtību. Ja lauks tiek atstāts tukšs, ienākošie savienojumi no visiem hosts tiks noraidīti. Ja tas ir norādīts, ienākošie savienojumi tiks pieņemti tikai no šeit uzskaitītajiem hosts. Ja ir iespējots IPv6 atbalsts, tad '127.0.0.1', '::127.0.0.1', '::ffff:127.0.0.1' tiek uzskatīti par ekvivalentiem, un '::/0' ļaus jebkuru IPv4 vai IPv6 adresi. '0.0.0.0/0' var izmantot, lai atļautu jebkuru IPv4 adresi. Ņemiet vērā, ka "IPv4-compatible IPv6 addresses" (0000::/96 prefikss) 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} Tiek atbalstītas atstarpes, lietotāja makro un host makro {HOST.HOST}, {HOST.NAME}, {HOST.IP}, {HOST.DNS}, {HOST.CONN}.

Note:
Pirms vērtību nosūtīšanas pēc vienuma saglabāšanas var būt jāgaida līdz 60 sekundēm, līdz Zabbix serveris no konfigurācijas kešatmiņas atjauninājuma paņems izmaiņas.

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, izmantojot Zabbix sender utilītu, nosūtītu datus uz Zabbix serveri vai starpniekserveri, varat palaist 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 servera IP adresi
- `-p` — lai norādītu Zabbix servera porta numuru (pēc noklusējuma 10051)
- `-s` — lai norādītu hostu (pārliecinieties, ka izmantojat tehnisko, nevis redzamo host name)
- `-k` — lai norādītu configured vienuma atslēgu trapper vienumā
- `-o` — lai norādītu nosūtāmo vērtību

Attention:
Zabbix trapper process neizvērš makro, kas izmantoti vienuma atslēgā, lai pārbaudītu atbilstošās vienuma atslēgas esamību mērķa hostam.

Plašāku informāciju par saziņu starp Zabbix sender un Zabbix serveri vai starpniekserveri 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:

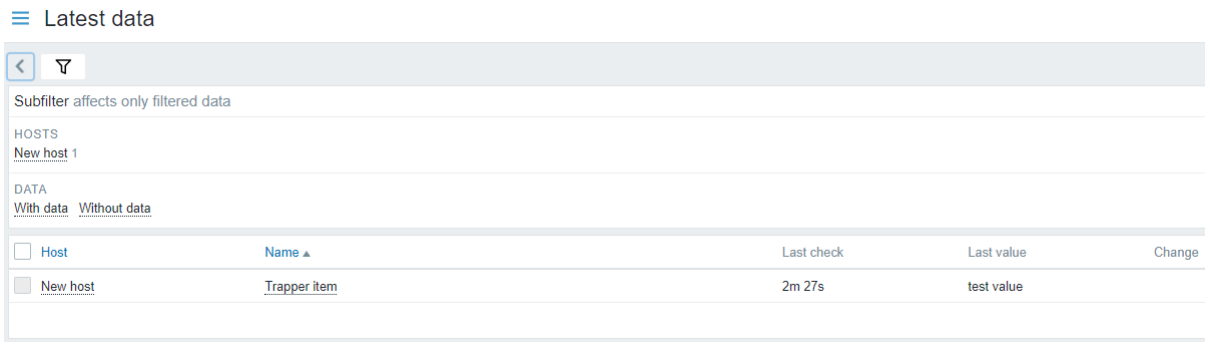
- lietotājam, kurš nosūta datus, nav *lasīšanas* atļaujas uz vienuma hostu;
- hosts ir atspējots vai atrodas uzturēšanas režīmā bez datu vākšanas;
- vienums neeksistē vai vēl nav iekļauts servera konfigurācijas kešatmiņā;
- vienums ir atspējots vai tā tips nav Zabbix trapper vai HTTP aģents (ar iespējotu 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 pāriet uz *Monitoring* > *Latest data*, lai redzētu rezultātu:



Note:
Ja tiek nosūtīta viena skaitliska vērtība, datu grafiks rādīs horizontālu līniju pa kreisi un pa labi no vērtības laika punkta.

Ārējā pārbaude

Pārskats

Ārējā pārbaude ir pārbaude, ko izpilda Zabbix serveris, palaižot čaulas skriptu vai bināro failu. Tomēr, ja hosts tiek uzraudzīti ar Zabbix starpniekserveri, ārējās pārbaudes izpilda starpniekserveris.

Ārējām pārbaudēm nav nepieciešams, lai uzraugāmajā hostā darbotos aģents.

Vienuma atslēgas sintakse ir:

`script[<parameter1>,<parameter2>,...]`

Kur:

ARGUMENTS	DEFINĪCIJA
script	Čaulas skripta vai binārā faila nosaukums.
parameter(s)	Neobligāti komandrindas parametri.

Ja nevēlaties nodot skriptam nekādus parametrus, varat izmantot:

```
script[] or
script
```

Zabbix serveris vai starpniekserveris meklēs norādīto ārējo skriptu direktoriju un izpildīs komandu (skatiet `ExternalScripts` parametru Zabbix servera/starpniekservera konfigurācijas failā). Komanda tiks izpildīta ar to pašu lietotāju, ar kuru darbojas Zabbix serveris/starpniekserveris, tāpēc jebkādas piekļuves atļaujas vai vides mainīgie, ja nepieciešams, jāapstrādā ietīšanas skriptā. Komandas atļaujām arī jāļauj šim lietotājam to izpildīt. Izpildei ir pieejamas tikai komandas norādītajā direktorijā.

Warning:
Nepārmērīgi izmantojiet ārējās pārbaudes, jo katram skriptam Zabbix serverim/starpniekserverim ir jāpalaiž 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.

Datu bāzes monitors

Pārskats

Database monitor vienuma tips Zabbix lietotāja saskarnē tiek izmantots ODBC uzraudzībai (ODBC pārbaudēm).

ODBC ir C programmēšanas valodas starpprogrammatūras API datubāzu pārvaldības sistēmu (DBMS) piekļuvei. ODBC koncepciju izstrādāja Microsoft, un vēlāk tā tika portēta uz citām platformām.

Zabbix var vaicāt jebkurai datubāzei, ko atbalsta ODBC. Lai to izdarītu, Zabbix nepieslēdzas datubāzēm tieši, bet izmanto ODBC saskarni un ODBC iestatītos draiverus. Tas ļauj efektīvāk uzraudzīt dažādas datubāzes vairākiem mērķiem (piemēram, konkrētu datubāzu rindu rindām, lietojuma statistikai utt.).

Zabbix atbalsta unixODBC, kas ir viena no visbiežāk izmantotajām atvērtā pirmkoda ODBC API implementācijām.

Attention:
Skatiet arī: zināmās problēmas ODBC pārbaudēm.

unixODBC instalēšana

Ieteicamais unixODBC instalēšanas veids ir izmantot Linux operētājsistēmas noklusējuma pakotņu repozitorijus. Populārākajos Linux distributīvos unixODBC pēc noklusējuma ir iekļauts pakotņu repozitorijā. Ja pakotnes nav pieejamas, avota failus var iegūt unixODBC sākumlapā: http://www.unixodbc.org/download.html.

Lai instalētu unixODBC, izmantojiet savas 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:
Pakotne `unixodbc-dev` vai `unixODBC-devel` 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.

Parametrs	Apraksts
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.

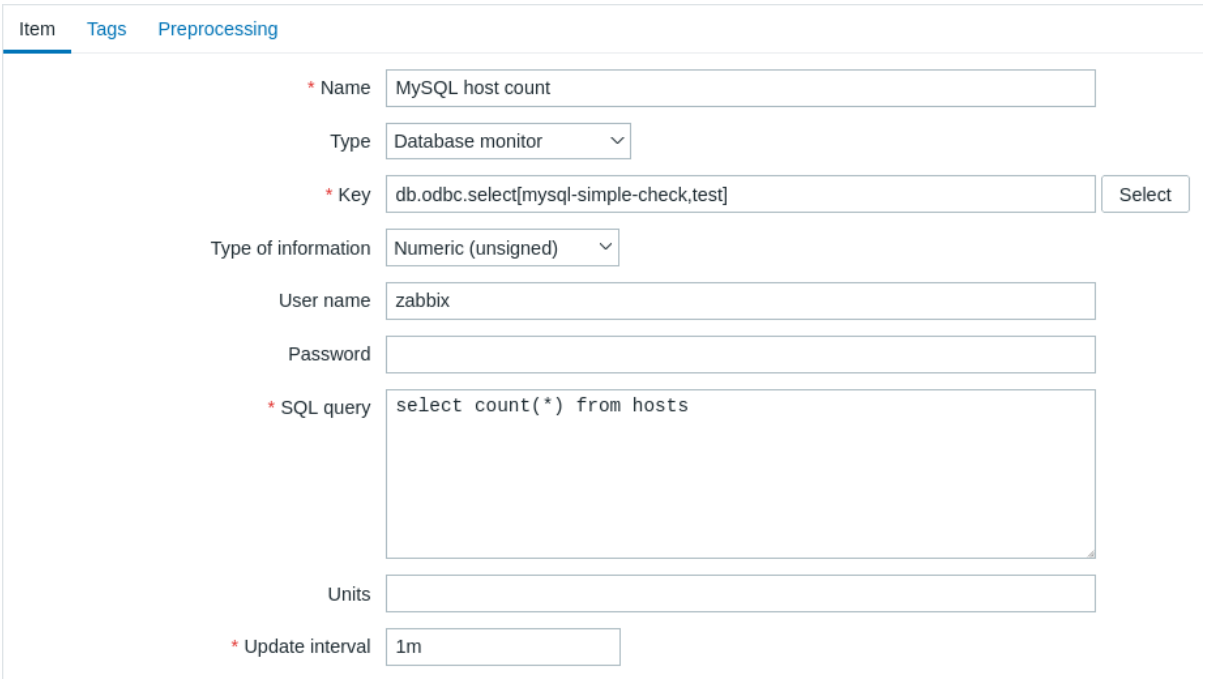
ODBC savienojuma testēšana

Lai pārbaudītu, vai ODBC savienojums darbojas veiksmīgi, varat izmantot `isql` utilītu (iekļauta `unixODBC` pakotnē):

```
isql test
+---------------------------------------+
| Connected!                            |
|                                       |
| sql-statement                         |
| help [tablename]                      |
| quit                                  |
|                                       |
+---------------------------------------+
```

Vienuma konfigurēšana Zabbix lietotāja saskarnē

Konfigurējiet **Datubāzes monitoringa** vienumu.



Visi obligātie ievades lauki ir atzīmēti ar sarkanu zvaigznīti.

Datubāzes monitoringa vienumiem ir jānorāda:

Type	Šeit atlasiet "Database monitor".
Key	Ievadiet vienu no atbalstītajiem vienuma atslēgas nosaukumiem: **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 zemā 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 tas tiek pievienots savienojuma virknei kā `PWD=<password>`. Šajā laukā tiek atbalstītas speciālās rakstzīmes. Visām datubāzēm, izņemot Oracle, paroles, kas satur %, ir jāievieto figūriekavās {}. Parole tiks pievienota savienojuma virknei aiz lietotājvārda, piemēram, `UID=<username>;PWD=P?;)*word`. Lai pārbaudītu iegūto virkni, varat palaist 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 izvēlēts nepareizi, vienums kļūs neatbalstīts.

Svarīgi norādījumi

- Datu bāzes uzraudzības vienumi kļūs neatbalstīti, ja servera vai starpniekservera konfigurācijā netiks palaisti neviens *odbc poller* process. Lai aktivizētu ODBC pollerus, Zabbix serveris konfigurācijas failā iestatiet parametru `StartODBCPollers` vai, ja pārbaudes tiek veiktas ar starpniekserveri, Zabbix starpniekserveris konfigurācijas failā.
- Vienuma konfigurācijas formā parametra *Timeout* vērtība 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 ir jāatgriež rezultātu kopa, tāpat kā jebkuram vaicājumam, kas izmanto `select` priekšrakstu. Vaicājuma sintakse būs atkarīga no RDBMS, kas to apstrādās. Pieprasījuma uzglabāšanas procedūrai sintaksei 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[<unique short description>,<dsn>,<connection string>]

 Pārveido SQL vaicājuma rezultātu JSON masīvā.
 Atgrieztā vērtība: *JSON objekts*.

Parametri:

- **unique short description** - unikāls īss apraksts, lai identificētu vienumu (izmantošanai trigeros u.c.);
- **dsn** - datu avota nosaukums (kā norādīts `odbc.ini`);
- **connection string** - savienojuma virkne (var saturēt draiverim specifiskus argumentus).

Piezīmes:

- Lai gan `dsn` un `connection string` ir neobligāti parametri, ir nepieciešams vismaz viens no tiem; ja abi ir definēti, `dsn` tiks ignorēts.
- JSON formātā var tikt atgrieztas vairākas rindas/kolonnas. Šo vienumu var izmantot kā galveno vienumu, kas vienā sistēmas izsaukumā apkopo visus datus, savukārt atkarīgajos vienumos individuālu vērtību iegūšanai var izmantot JSONPath priekšapstrādi. Plašāku informāciju skatiet piemērā par atgriezto formātu, kas izmantots zemā 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.

HTTP aģents

Pārskats

Šis vienuma tips ļauj veikt datu aptauju, izmantojot HTTP/HTTPS protokolu. Iespējama arī datu saņemšana, izmantojot Zabbix sender utilītu vai Zabbix sender protokolu (datu nosūtīšanai uz Zabbix serveri vai starpniekserveri), vai izmantojot `history.push` API metodi (datu nosūtīšanai uz Zabbix serveri).

HTTP vienuma pārbaudes izpilda Zabbix serveris. Tomēr, ja hosts tiek uzraudzīti ar Zabbix starpniekserveri, HTTP vienuma pārbaudes izpilda starpniekserveris.

HTTP vienuma pārbaudēm nav nepieciešams, lai uz uzraudzītā 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 fiksēts 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 viena pieprasījuma atbildi, pirms tiek sāktas citas pārbaudes. DNS atrisināšana arī ir asinhrona.

Asinhrono pārbaužu maksimālais paralēlais skaits ir 1000 (definēts ar MaxConcurrentChecksPerPoller).

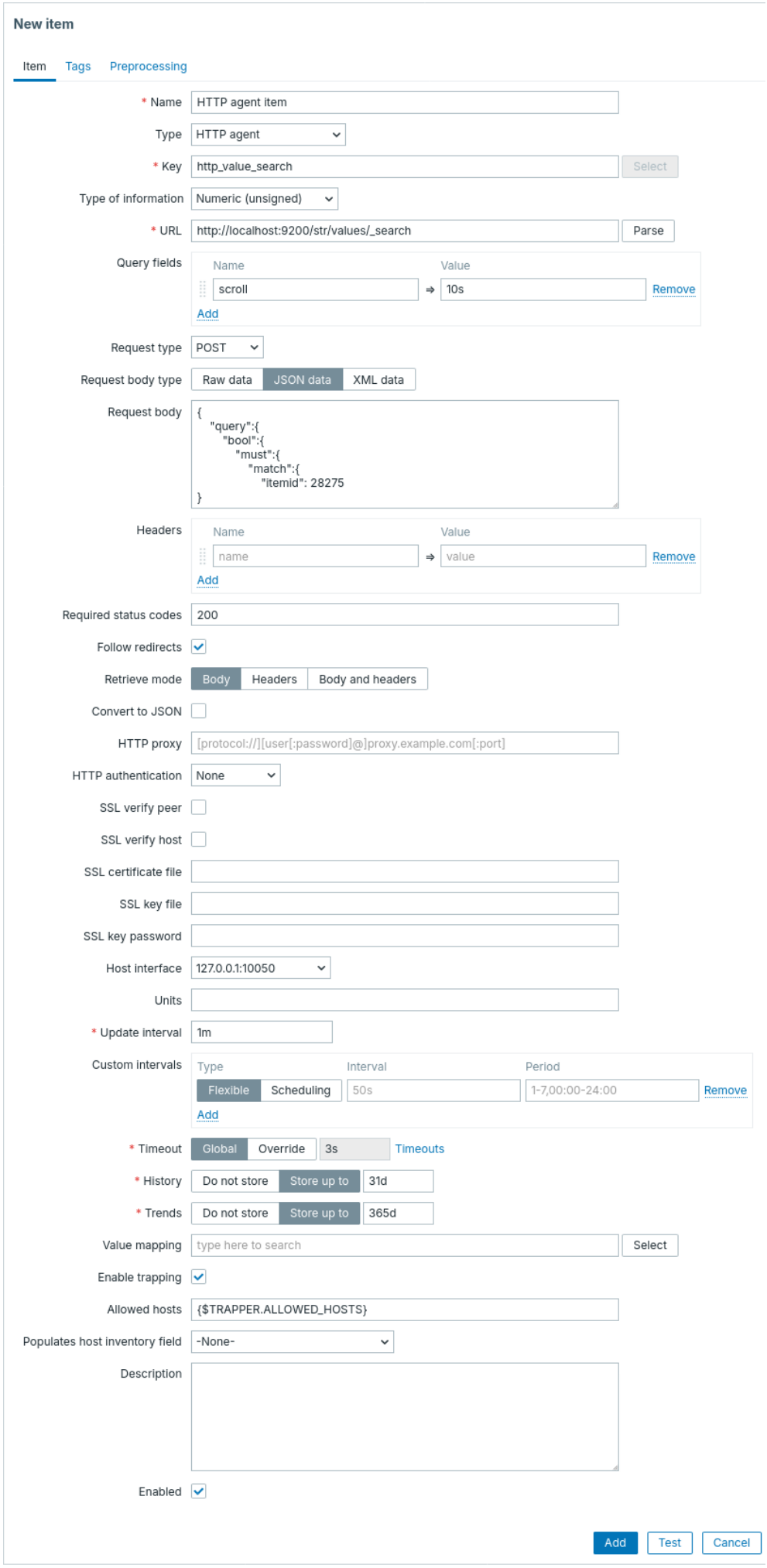
Asinhrono HTTP aģenta aptaujātāju skaits ir definēts ar parametru StartHTTPAgentPollers.

Pastāvīgo savienojumu cURL funkcionalitāte ir pievienota HTTP aģenta pārbaudēm kopš Zabbix 7.0.

Konfigurēšana

Lai konfigurētu HTTP vienumu:

1. Atveriet *Data collection* > *Hosts*.
2. Hosts rindā noklikšķiniet uz *Items*.
3. Noklikšķiniet uz *Create item*.
4. Veidlapā ievadiet vienuma parametrus.


New item
Item
Tags
Preprocessing
* Name
HTTP agent item
Type
HTTP agent
* Key
http_value_search
Select
Type of information
Numeric (unsigned)
* URL
http://localhost:9200/str/values/_search
Parse
Query fields
Name
Value
scroll
10s
Remove
Add
Request type
POST
Request body type
Raw data
JSON data
XML data
Request body
{
"query":{
"bool":{
"must":{
"match":{
"itemid": 28275
}
Headers
Name
Value
name
value
Remove
Add
Required status codes
200
Follow redirects
Retrieve mode
Body
Headers
Body and headers
Convert to JSON
HTTP proxy
[protocol://][user[:password]@]proxy.example.com[:port]
HTTP authentication
None
SSL verify peer
SSL verify host
SSL certificate file
SSL key file
SSL key password
Host interface
127.0.0.1:10050
Units
* Update interval
1m
Custom intervals
Type
Interval
Period
Flexible
Scheduling
50s
1-7,00:00-24:00
Remove
Add
* Timeout
Global
Override
3s
Timeouts
* History
Do not store
Store up to
31d
* Trends
Do not store
Store up to
365d
Value mapping
type here to search
Select
Enable trapping
Allowed hosts
{$TRAPPER.ALLOWED_HOSTS}
Populates host inventory field
-None-
Description
Enabled
Add
Test
Cancel

Visi obligātie ievades lauki ir atzīmēti ar sarkanu zvaigznīti.

Lauki, kuriem HTTP vienumiem nepieciešama specifiska informācija:

Parametrs	Apraksts
Type	Šeit atlasiet **HTTP agent**.
Key	Ievadiet unikālu vienuma atslēgu.

URL	URL, ar kuru izveidot savienojumu un izgūt datus. Piemēram:
	`https://www.example.com`
	`http://www.example.com/download`
	Domēna nosaukumus var norādīt Unicode rakstzīmēs. Izpildot HTTP pārbaudi, tie tiek automātiski pārveidoti par ASCII punycode formātā.
	Pogu *Parse* var izmantot, lai no URL atdalītu neobligātos vaicājuma laukus (piemēram, `?key=value&limit=10`), pārvietojot atribūtus un vērtības uz *Query fields*, lai veiktu automātisku URL kodēšanu.
	Drošības apsvērumu dēļ URL neiekļaujiet faktiskos lietotājvārdus vai paroles. Tā vietā izmantojiet lietotāja makro, lai nepieļautu akreditācijas datu atklāšanu.
	Maksimālais garums — 2048 rakstzīmes.
	Atbalstītie makro: {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {HOST.PORT}, {ITEM.ID}, {ITEM.KEY}, {ITEM.KEY.ORIG}, lietotāja makro, zema līmeņa atklāšanas makro.
	Tiek iestatīta cURL opcija CURLOPT_URL.

Query fields	URL mainīgie (skatiet iepriekš). Norādīti kā atribūtu un vērtību pāri. Vērtības tiek automātiski kodētas URL formātā. Makro vērtības tiek aizstātas un pēc tam automātiski kodētas URL formātā. Atbalstītie makro: {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {HOST.PORT}, {ITEM.ID}, {ITEM.KEY}, {ITEM.KEY.ORIG}, lietotāja makro, zema līmeņa atklāšanas makro. Tiek iestatīta cURL opcija 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; makro tiek aizstāti, bet kodēšana netiek veikta. **JSON data** — HTTP pieprasījuma pamatteksts JSON formātā. Makro var izmantot kā virkni, skaitli, true un false; makro, kas tiek izmantoti kā virknes, jāiekļauj dubultpēdiņās. Makro vērtības tiek aizstātas un pēc tam automātiski ekranētas. Ja galvenēs nav norādīts "Content-Type", pēc noklusējuma tiks izmantots "Content-Type: application/json". **XML data** — HTTP pieprasījuma pamatteksts XML formātā. Makro var izmantot kā teksta mezglu, atribūtu vai CDATA sadaļu. Makro vērtības teksta mezglā un atribūtā tiek aizstātas un pēc tam automātiski ekranētas. Ja galvenēs nav norādīts "Content-Type", pēc noklusējuma tiks izmantots "Content-Type: application/xml".
Request body	Ievadiet pieprasījuma pamattekstu. Atbalstītie makro: {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {HOST.PORT}, {ITEM.ID}, {ITEM.KEY}, {ITEM.KEY.ORIG}, lietotāja makro, zema līmeņa atklāšanas makro.
Headers	Pielāgotas HTTP galvenes, kas tiks nosūtītas, izpildot pieprasījumu. Norādītas kā atribūtu un vērtību pāri. Atbalstītie makro: {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {HOST.PORT}, {ITEM.ID}, {ITEM.KEY}, {ITEM.KEY.ORIG}, lietotāja makro, zema līmeņa atklāšanas makro. Tiek iestatīta cURL opcija CURLOPT_HTTPHEADER.
Required status codes	Sagaidāmo HTTP statusa kodu saraksts. Ja Zabbix saņem kodu, kas nav iekļauts sarakstā, vienums kļūs neatbalstīts. Ja saraksts ir tukšs, pārbaude netiek veikta. Piemēram: 200,201,210-299. Sarakstā atbalstītie makro: lietotāja makro, zema līmeņa atklāšanas makro. Tiek izmantota cURL opcija CURLINFO_RESPONSE_CODE.
Follow redirects	Atzīmējiet izvēles rūtiņu, lai sekotu HTTP novirzēm. Tiek iestatīta cURL opcija CURLOPT_FOLLOWLOCATION.
Retrieve mode	Atlasiet izgūstamo atbildes daļu: **Body** — tikai pamatteksts **Headers** — tikai galvenes **Body and headers** — pamatteksts un galvenes

Parametrs	Apraksts
Convert to JSON	Ja *Retrieve mode* nav iestatīts uz *Body*, galvenes tiek saglabātas kā atribūtu un vērtību pāri zem atslēgas "header". Ja tiek atrasts 'Content-Type: application/json', pamatteksts tiek saglabāts kā objekts; pretējā gadījumā tas tiek saglabāts kā virkne, piemēram: `{ "header": { "<key>": "<value>", "<key2>": "<value>" }, "body": <body> }`
HTTP proxy	Varat norādīt izmantojamo HTTP starpniekserveri, izmantojot formātu `[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). Ja protokols nav norādīts, starpniekserveris tiks uzskatīts par HTTP starpniekserveri. Ja norādīts nepareizs protokols, savienojums neizdosies un vienums kļūs neatbalstīts. Pēc noklusējuma tiks izmantots ports 1080. Ja starpniekserveris ir norādīts, tas aizstās ar starpniekserveri saistītos vides mainīgos, piemēram, `http_proxy`, `HTTPS_PROXY`. Ja tas nav norādīts, starpniekserveris ar starpniekserveri saistītos vides mainīgos neaizstās. Ievadītā vērtība tiek nodota "kā ir", un tās pareizība netiek pārbaudīta. Ņemiet vērā, ka HTTP starpniekserverim tiek atbalstīta tikai vienkāršā autentifikācija. Atbalstītie makro: `{HOST.IP}`, `{HOST.CONN}`, `{HOST.DNS}`, `{HOST.HOST}`, `{HOST.NAME}`, `{HOST.PORT}`, `{ITEM.ID}`, `{ITEM.KEY}`, `{ITEM.KEY.ORIG}`, lietotāja makro, zema līmeņa atklāšanas makro. Tiek iestatīta cURL opcija `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. Tiek iestatīta cURL opcija `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 makro un zema līmeņa atklāšanas makro.
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 makro un zema līmeņa atklāšanas makro.
SSL verify peer	Atzīmējiet izvēles rūtiņu, lai pārbaudītu tīmekļa servera SSL sertifikātu. Servera sertifikāts tiek pārbaudīts, izmantojot uzticamas sertificēšanas iestādes (CA) sertifikātus no visas sistēmas CA atrašanās vietas. CA failu atrašanās vietu var mainīt, izmantojot Zabbix servera vai starpniekservera konfigurācijas parametru SSLCALocation. Tiek iestatīta cURL opcija `CURLOPT_SSL_VERIFYPEER`.
SSL verify host	Atzīmējiet izvēles rūtiņu, lai pārbaudītu, vai tīmekļa servera sertifikāta lauks *Common Name* vai *Subject Alternate Name* atbilst. Tiek iestatīta cURL opcija `CURLOPT_SSL_VERIFYHOST`.
SSL certificate file	Klienta autentifikācijai izmantotā SSL sertifikāta faila nosaukums. Sertifikāta failam jābūt PEM[1] formātā. Ja sertifikāta fails satur arī privāto atslēgu, atstājiet lauku *SSL key file* tukšu. Ja atslēga ir šifrēta, laukā *SSL key password* norādiet paroli. Direktoriju, kurā atrodas šis fails, norāda Zabbix servera vai starpniekservera konfigurācijas parametrs `SSLCertLocation`. Atbalstītie makro: `{HOST.IP}`, `{HOST.CONN}`, `{HOST.DNS}`, `{HOST.HOST}`, `{HOST.NAME}`, `{HOST.PORT}`, `{ITEM.ID}`, `{ITEM.KEY}`, `{ITEM.KEY.ORIG}`, lietotāja makro, zema līmeņa atklāšanas makro. Tiek iestatīta cURL opcija `CURLOPT_SSLCERT`.

Parametrs	Apraksts
SSL key file	Klienta autentifikācijai izmantotā SSL privātās atslēgas faila nosaukums. Privātās atslēgas failam jābūt PEM[1] formātā. Direktoriju, kurā atrodas šis fails, norāda Zabbix servera vai starpniekservera konfigurācijas parametrs `SSLKeyLocation`. Atbalstītie makro: `{HOST.IP}`, `{HOST.CONN}`, `{HOST.DNS}`, `{HOST.HOST}`, `{HOST.NAME}`, `{HOST.PORT}`, `{ITEM.ID}`, `{ITEM.KEY}`, `{ITEM.KEY.ORIG}`, lietotāja makro, zema līmeņa atklāšanas makro. Tiek iestatīta cURL opcija `CURLOPT_SSLKEY`.
SSL key password	SSL privātās atslēgas faila parole. Atbalstītie makro: lietotāja makro, zema līmeņa atklāšanas makro. Tiek iestatīta cURL opcija `CURLOPT_KEYPASSWD`.
Timeout	Zabbix URL apstrādei netērēs vairāk par iestatīto laiku (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. Tiek iestatīta cURL opcija `CURLOPT_TIMEOUT`. Papildinformāciju par parametru *Timeout* skatiet vispārīgajos vienuma atribūtos.
Enable trapping	Ja šī izvēles rūtiņa ir atzīmēta, vienums darbosies arī kā trapper vienums un pieņems datus, kas nosūtīti Zabbix serverim vai starpniekserverim, izmantojot utilītu Zabbix sender vai Zabbix sender protokolu, kā arī pieņems datus, kas nosūtīti Zabbix serverim, izmantojot `history.push` API metodi. Papildinformāciju par datu nosūtīšanu skatiet: Trapper vienumi.
Allowed hosts	Redzams tikai tad, ja ir atzīmēta izvēles rūtiņa *Enable trapping*. Ar komatiem atdalītu IP adrešu (pēc izvēles CIDR apzīmējumā) vai DNS nosaukumu saraksts. Noklusējuma vērtība ir lietotāja makro `{$TRAPPER.ALLOWED_HOSTS}`. Pēc noklusējuma `{$TRAPPER.ALLOWED_HOSTS}` ir `127.0.0.1,::1`. Varat ievadīt adreses tieši laukā vai mainīt makro vērtību. Ja lauks ir tukšs, ienākošie savienojumi no visiem hosts tiks noraidīti. Ja tas ir norādīts, ienākošie savienojumi tiks pieņemti tikai no šeit uzskaitītajiem hosts. Ja IPv6 atbalsts ir iespējots, `127.0.0.1`, `::127.0.0.1`, `::ffff:127.0.0.1` tiek uzskatītas par līdzvērtīgām, 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" (`0000::/96` prefikss) tiek atbalstītas, taču RFC4291 tās ir novecojušas. 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 makro. Šajā laukā ir atļauti host makro: `{HOST.HOST}`, `{HOST.NAME}`, `{HOST.IP}`, `{HOST.DNS}`, `{HOST.CONN}`.

Note:
Ja lauks *HTTP proxy* ir tukšs, vēl viens veids, kā izmantot HTTP starpniekserveri, ir iestatīt ar starpniekserveri saistītos vides mainīgos.
HTTP gadījumā iestatiet `http_proxy` vides mainīgo Zabbix servera lietotājam. Piemēram:
 `http_proxy=http://proxy_ip:proxy_port`.
HTTPS gadījumā iestatiet `HTTPS_PROXY` vides mainīgo. Piemēram:
 `HTTPS_PROXY=http://proxy_ip:proxy_port`.
Papildinformācija ir pieejama, izpildot čaulas komandu: *# man curl*.

Attention:
[1] Zabbix atbalsta tikai sertifikātu un privāto atslēgu failus PEM formātā. Ja sertifikāta un privātās atslēgas dati atrodas PKCS #12 formāta failā (parasti ar paplašinājumu `*.p12` vai `*.pfx`), PEM failu no tā varat ģenerēt, 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

Nosūtiet vienkāršus GET pieprasījumus, lai izgūtu 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 izvelciet versijas numuru, izmantojot JSONPath pirmapstrādes soli: `$.version.number`

Piemērs 2

Nosūtiet vienkāršus POST pieprasījumus, lai izgūtu datus no tādiem pakalpojumiem kā Elasticsearch:

- Izveidojiet POST vienumu ar URL: `http://localhost:9200/str/values/_search?scroll=10s`
- Konfigurējiet šādu POST ķermeni, lai iegūtu procesora noslodzi (1 minūtes vidējais rādītājs 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": {
```

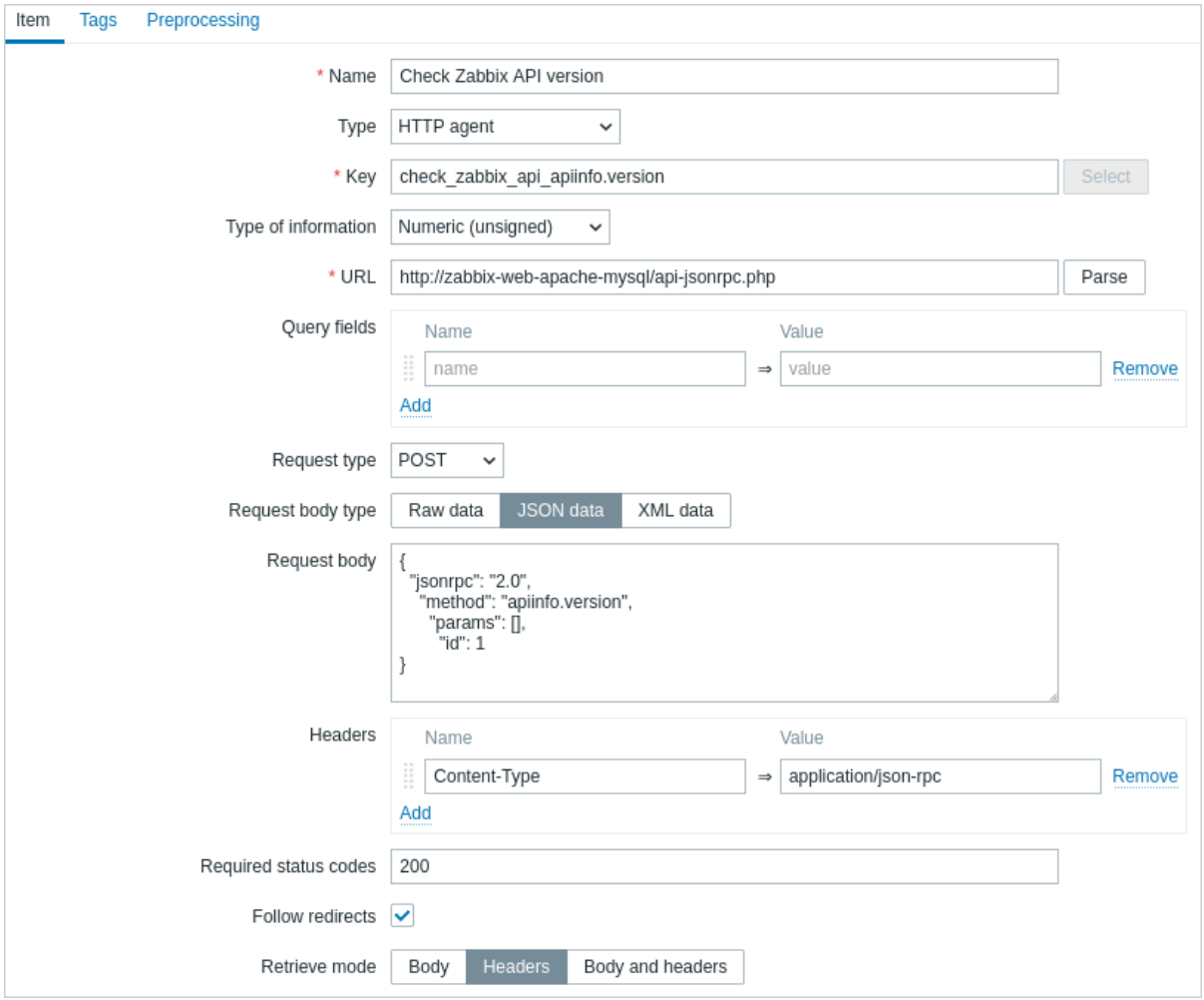
```
                "itemid": 28275,
                "value": "0.138750",
                "clock": 1517566136,
                "ns": 25388713,
                "ttl": 604800
            }
        }]
    }
}
```

- Tagad izmantojiet JSONPath priekšapstrādes soli, lai iegūtu vienuma vērtību: `$.hits.hits[0]._source.value`

Piemērs 3

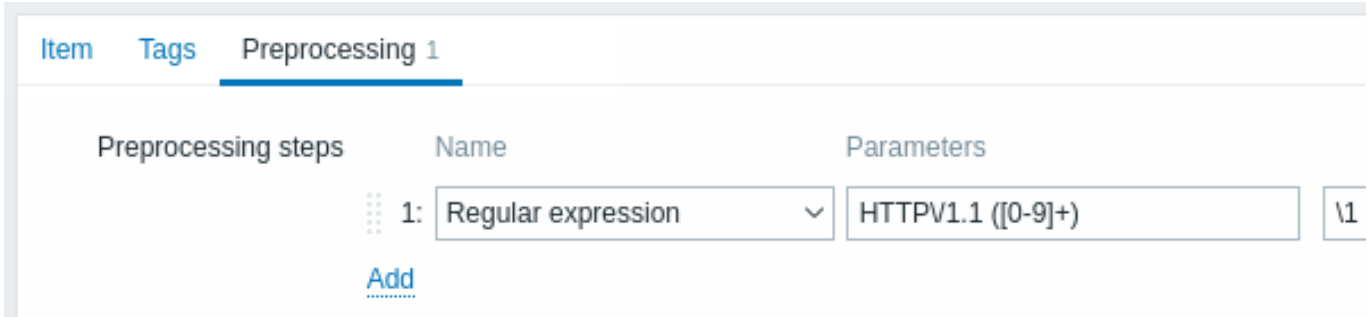
Pārbaude, vai Zabbix API ir aktīva, izmantojot apiinfo.version.

- Vienuma konfigurācija:

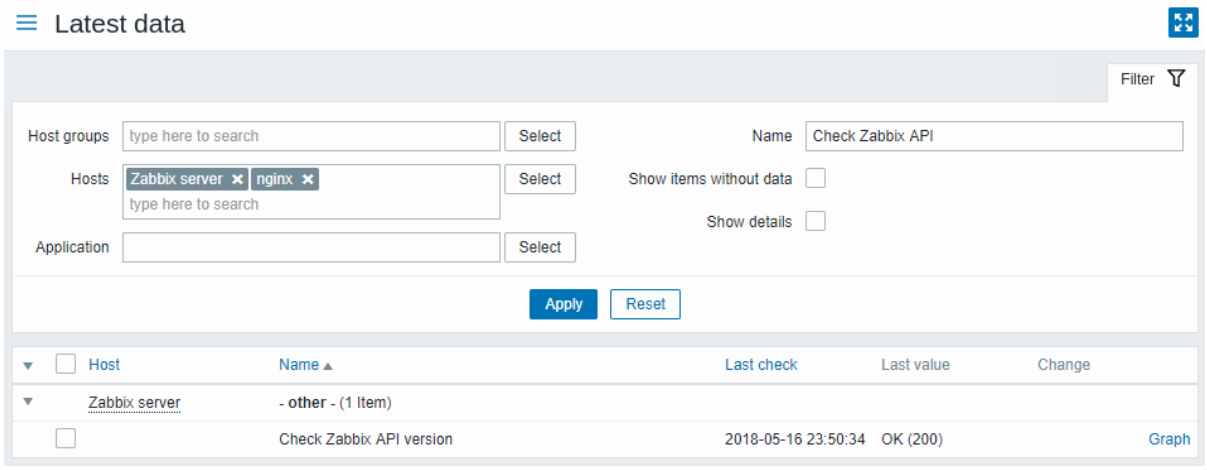


Ņemiet vērā POST metodes izmantošanu ar JSON datiem, pieprasījuma galvenes iestatīšanu un pieprasījumu atgriezt tikai galvenes:

- Vienuma vērtības pirmapstrāde ar regulāro izteiksmi, lai iegūtu HTTP kodu:



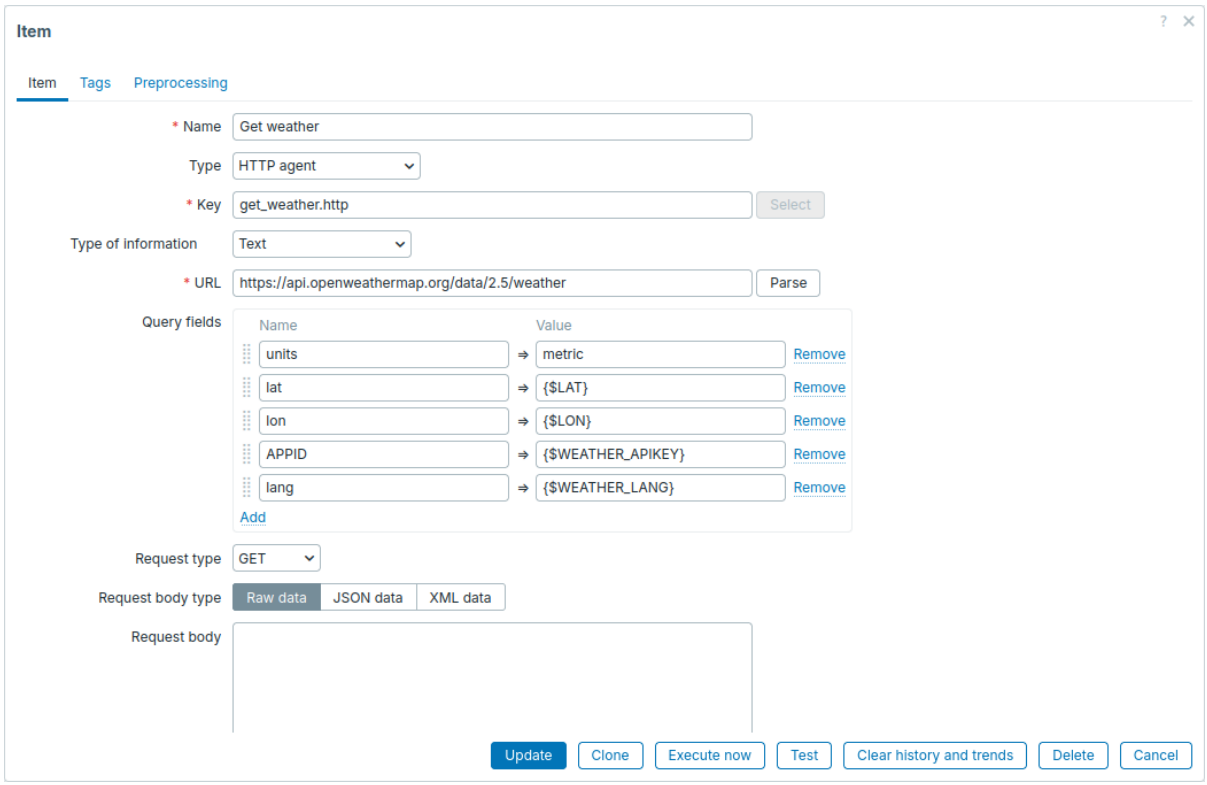
- Rezultāta pārbaude sadaļā *Latest data*:



Piemērs 4

Laika apstākļu informācijas iegūšana, pieslēdzoties Openweathermap publiskajam servisam.

- Konfigurējiet galveno vienumu, lai veiktu masveida datu vākšanu vienā JSON:



Ņemiet vērā makro izmantošanu vaicājuma laukos. Skatiet Openweathermap API, lai uzzinātu, kā tos aizpildīt.

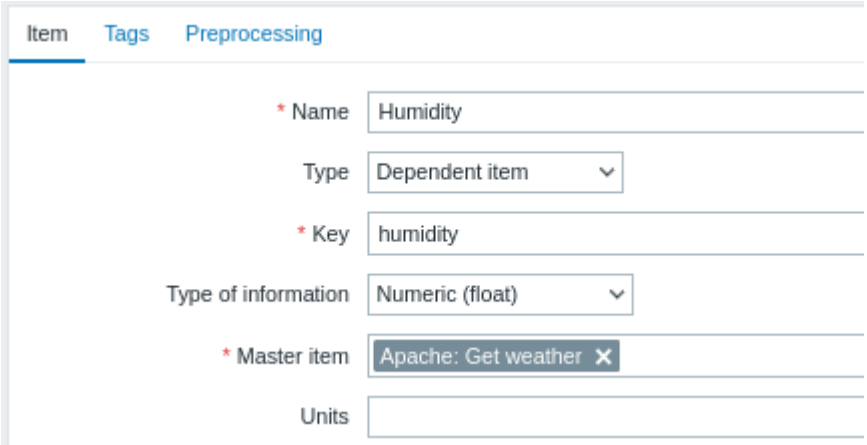
Parauga JSON, kas tiek atgriezts HTTP aģenta atbildē:

```
{
    "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
    }
}
```

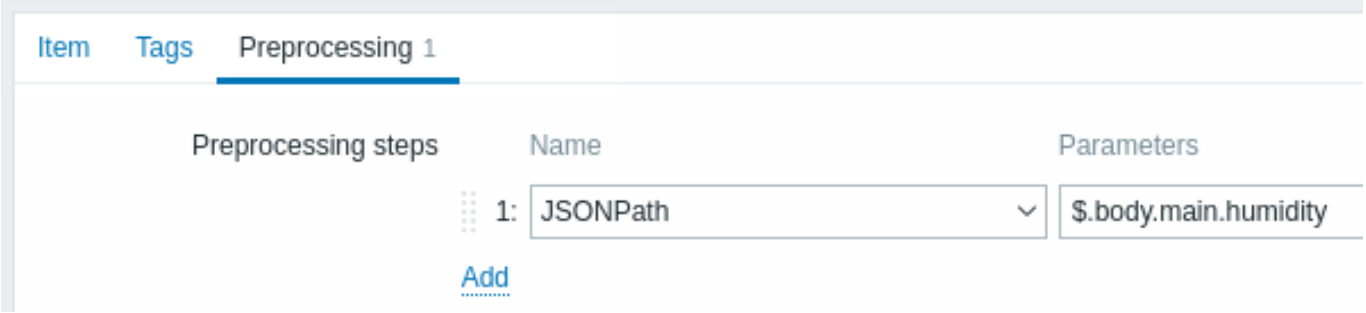

Nākamais uzdevums ir konfigurēt atkarīgos vienumus, kas no JSON izgūst datus.

- Konfigurējiet parauga atkarīgo vienumu mitrumam:

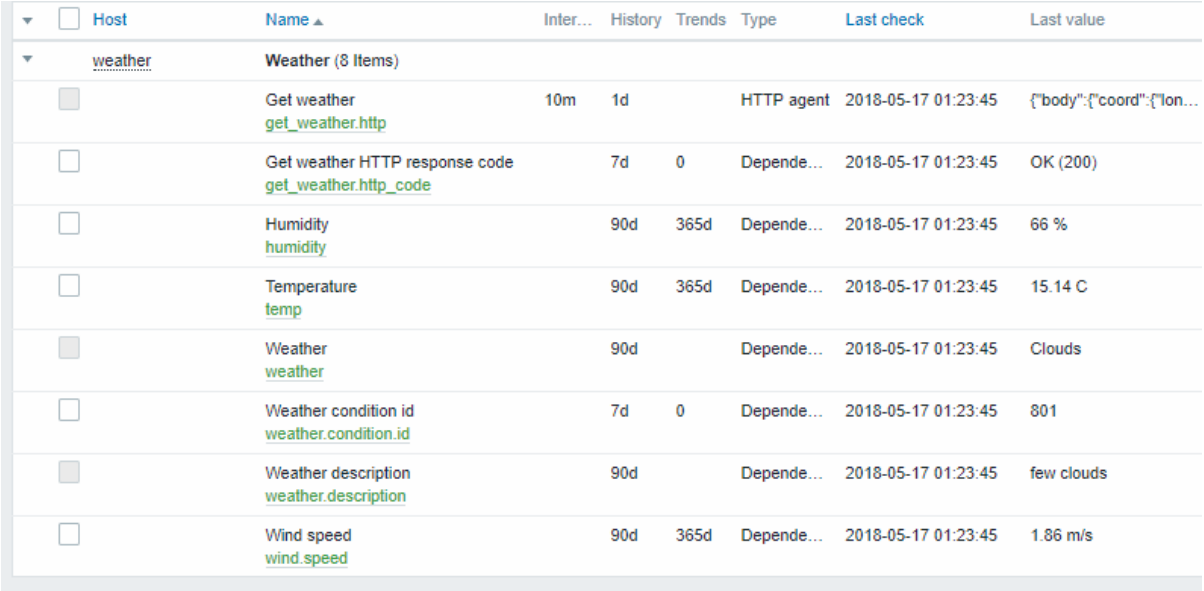


Citi laikapstākļu rādītāji, piemēram, ”Temperature”, tiek pievienoti tādā pašā veidā.

- Parauga atkarīgā vienuma vērtības pirmapstrāde ar JSONPath:



- Pārbaudiet laikapstākļu datu rezultātu sadaļā *Latest data*:

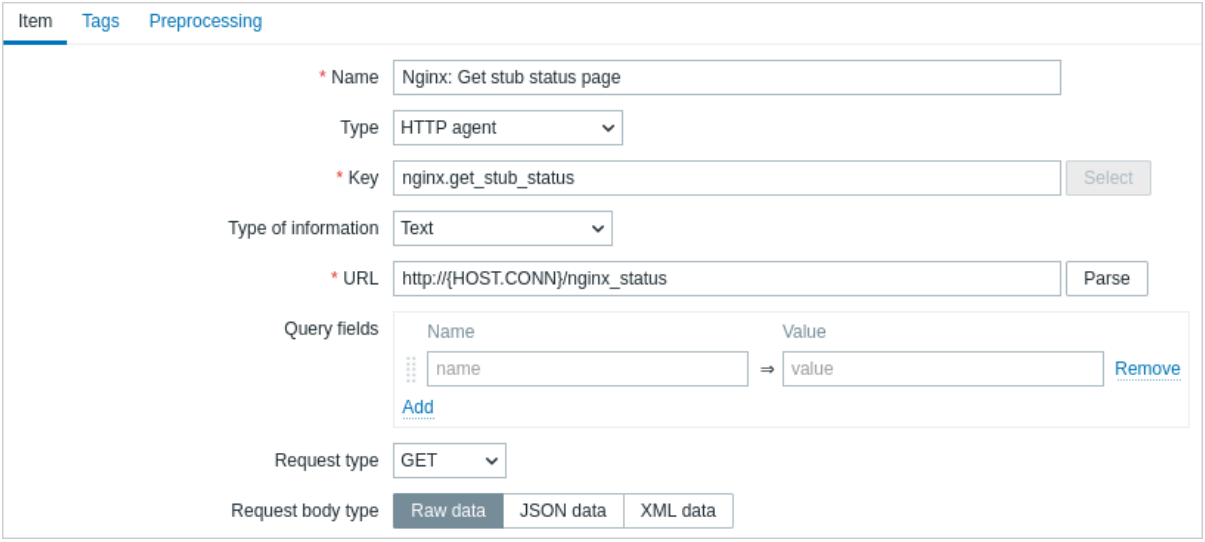


Host	Name ▲	Inter...	History	Trends	Type	Last check	Last value
weather	**Weather** (8 Items)						
	Get weather get_weather.http	10m	1d		HTTP agent	2018-05-17 01:23:45	{"body":{"coord":{"lon...
	Get weather HTTP response code get_weather.http_code		7d	0	Depende...	2018-05-17 01:23:45	OK (200)
	Humidity humidity		90d	365d	Depende...	2018-05-17 01:23:45	66 %
	Temperature temp		90d	365d	Depende...	2018-05-17 01:23:45	15.14 C
	Weather weather		90d		Depende...	2018-05-17 01:23:45	Clouds
	Weather condition id weather.condition.id		7d	0	Depende...	2018-05-17 01:23:45	801
	Weather description weather.description		90d		Depende...	2018-05-17 01:23:45	few clouds
	Wind speed wind.speed		90d	365d	Depende...	2018-05-17 01:23:45	1.86 m/s

5. piemērs

Savienojuma izveide ar NGINX statusa lapu un tās metrikas iegūšana lielapjomā.

- Konfigurējiet NGINX, izmantojot oficiālo rokasgrāmatu.
- Konfigurējiet galveno vienumu lielapjoma datu apkopošanai:

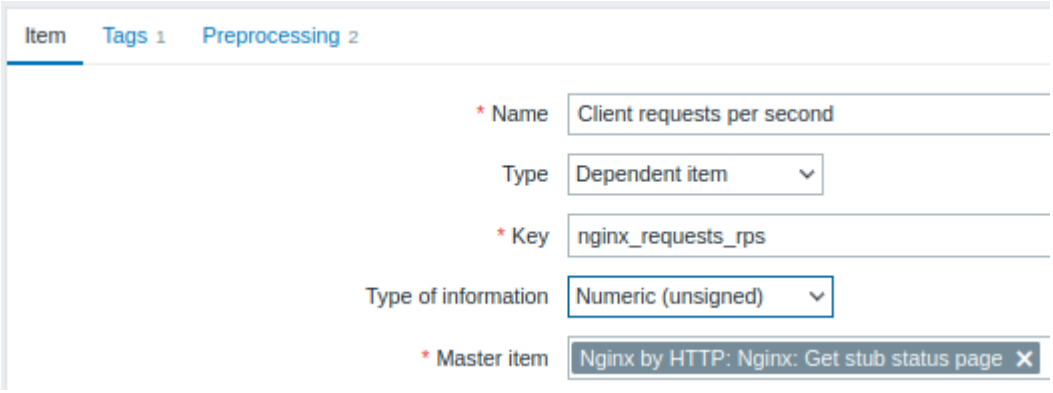


NGINX stub statusa izvades piemērs:

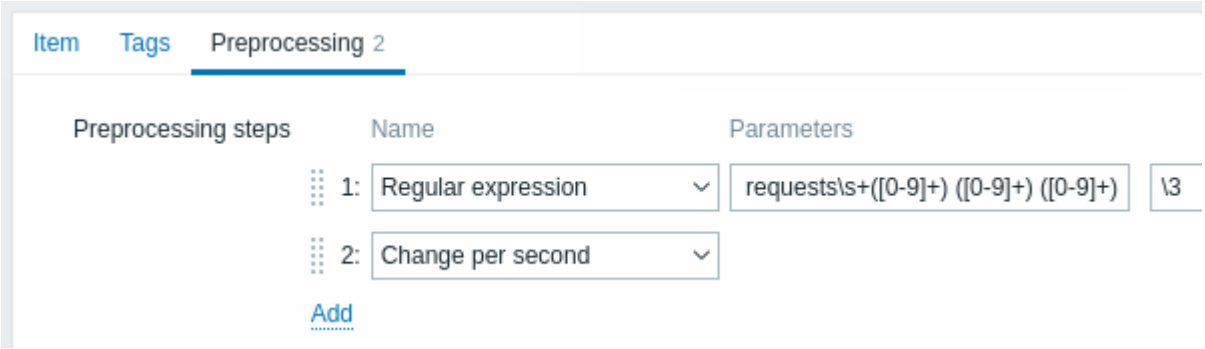
```
Active connections: 1 Active connections:
server accepts handled requests
 52 52 52
Reading: 0 Writing: 1 Waiting: 0
```

Nākamais uzdevums ir konfigurēt atkarīgos vienumus, kas iegūst datus.

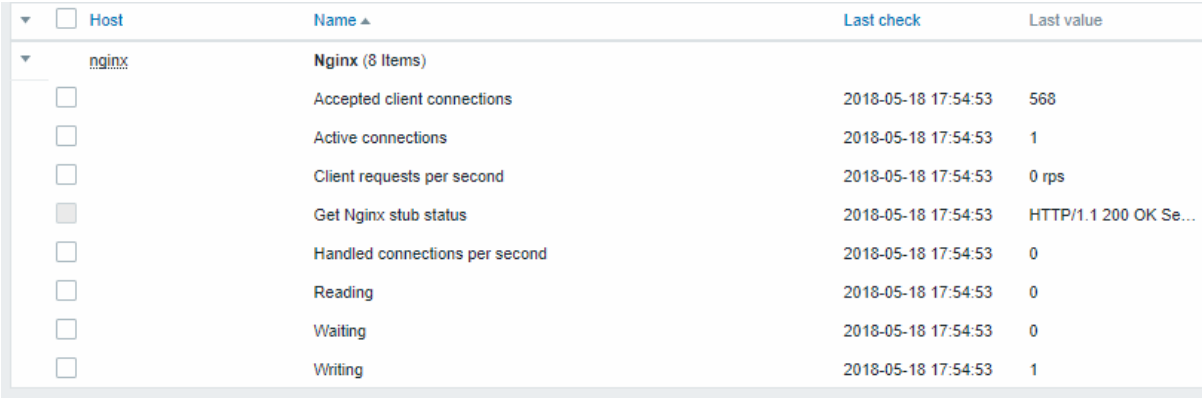
- Konfigurējiet atkarīga vienuma piemēru pieprasījumiem sekundē:



- Atkarīga vienuma vērtības priekšapstrādes piemērs ar regulāro izteiksmi `server accepts handled requests\s+([0-9]+) ([0-9]+) ([0-9]+)`:



- Pārbaudiet pilno stub moduļa rezultātu sadaļā *Jaunākie dati*:



Prometheus pārbaude

Pārskats

Zabbix var vaicāt metriku, kas tiek eksponētas Prometheus līniju 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, izmantojot Prometheus priekšapstrādes opciju, lai vaicātu nepieciešamos datus no metriku, ko savācis galvenais vienums

Pastāv divas Prometheus datu priekšapstrādes opcijas:

- *Prometheus pattern* - tiek izmantota parastajos vienumos, lai vaicātu Prometheus datus
- *Prometheus to JSON* - tiek izmantota parastajos vienumos un zemā līmeņa atklāšanai. Šajā gadījumā vaicātie Prometheus dati tiek atgriezti JSON formātā.

Masveida apstrāde

Masveida apstrāde tiek atbalstīta atkarīgajiem vienumiem. Lai iespējotu kešošanu un indeksēšanu, *Prometheus pattern* pirmapstrādei ir jābūt **pirmajam** pirmapstrādes solim. Ja *Prometheus pattern* ir pirmais pirmapstrādes solis, tad parsētie Prometheus dati

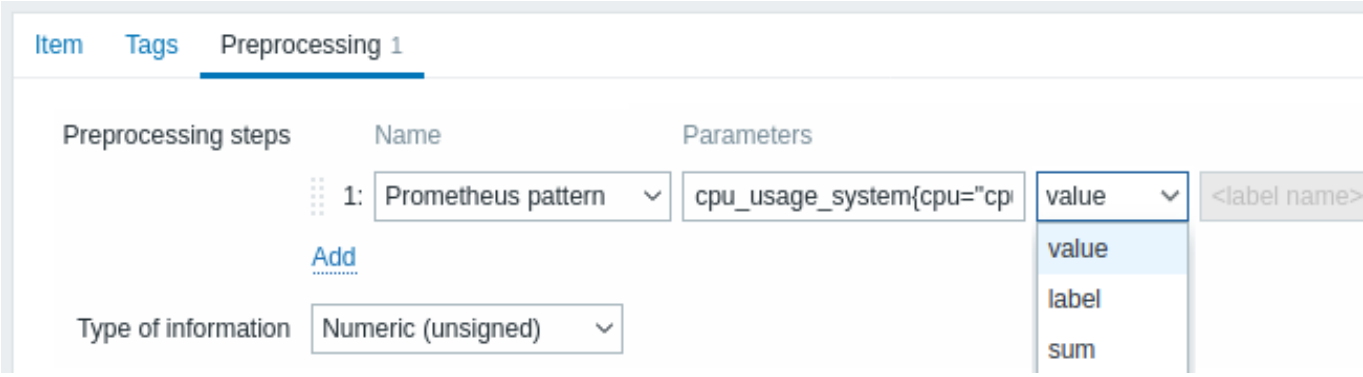
tiek kešoti un indeksēti pēc pirmā `<label>==<value>` nosacījuma *Prometheus pattern* pirmapstrādes solī. Šis kešs tiek izmantots atkārtoti, apstrādājot citus atkarīgos vienumus šajā paketē. Optimālai veiktspējai pirmajai etiķetei vajadzētu būt tai, kurai ir visvairāk atšķirīgu vērtību.

Ja pirms pirmā soļa ir jāveic cita pirmapstrāde, tā ir jāpārvieto vai nu uz galveno vienumu, vai uz jaunu atkarīgo vienumu, kas tiks izmantots kā galvenais vienums atkarīgajiem vienumiem.

Konfigurācija

Ja HTTP galvenais vienums ir konfigurēts, jāizveido atkarīgais vienums, kas izmanto Prometheus priekšapstrādes darbību:

1. Konfigurācijas formā ievadiet vispārīgos atkarīgā vienuma parametrus.
2. Atveriet cilni *Priekšapstrāde*.
3. Atlasiet Prometheus priekšapstrādes opciju (*Prometheus pattern* vai *Prometheus to JSON*).



Tālāk norādītie parametri attiecas tikai uz priekšapstrādes opciju *Prometheus pattern*:

Parametrs	Apraksts	Piemēri
Pattern	Lai definētu nepieciešamo datu modeli, varat izmantot vaicājumu valodu, kas ir līdzīga Prometheus vaicājumu valodai (skatiet salīdzināšanas tabulu), piemēram: <metric name> - atlase pēc metrikas nosaukuma {__name__="<metric name>"} - atlase pēc metrikas nosaukuma {__name__=~"<regex>"} - atlase pēc metrikas nosaukuma, kas atbilst regulārajai izteiksmei {<label name>="<label value>",...} - atlase pēc etiķetes nosaukuma {<label name>=~"<regex>",...} - atlase pēc etiķetes nosaukuma, kas atbilst regulārajai izteiksmei {__name__=~".*"}==<value> - atlase pēc metrikas vērtības Vai iepriekš minēto nosacījumu kombinācija: <metric name>{<label1 name>="<label1 value>",<label2 name>=~"<regex>",...}==<value> Etiķetes vērtība var būt jebkura UTF-8 rakstzīmju virkne, taču slīpsvītras, dubultpēdiņas un rindu pārneses rakstzīmes ir jāekranē attiecīgi kā \\, \" un \n; pārējās 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 jāatgriež vērtība, etiķete vai jālieto atbilstošā funkcija (ja modelim atbilst vairākas rindas un rezultāts ir jāapkopo): **value** - atgriezt metrikas vērtību (kļūda, ja atbilst vairākas rindas) **label** - atgriezt *Label* laukā norādītās etiķetes vērtību (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 izmantošanas piemērus.
Output	Norādiet etiķetes nosaukumu (neobligāti). Šādā gadījumā tiek atgriezta etiķetes nosaukumam atbilstošā vērtība. Šis lauks ir pieejams tikai opcijai *Prometheus pattern*, ja laukā *Result processing* ir atlasīta opcija *Label*.	

Parametru izmantošanas piemēri

1. Visbiežāk vērtība tiek atgriezta kā **value**. Lai no tālāk norādītajiem datiem atgrieztu `/var/db` vērtību:

   `node_disk_usage_bytes{path="/var/cache"} 2.1766144e+09`
`node_disk_usage_bytes{path="/var/db"} 20480`
`node_disk_usage_bytes{path="/var/dpkg"} 8192`
`node_disk_usage_bytes{path="/var/empty"} 4096`

izmantojiet šādus parametrus:

- *Pattern* - `node_disk_usage_bytes{path="/var/db"}`
- *Result processing* - atlasiet 'value'

2. Var būt nepieciešama 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, bieži tiek izmantots risinājums, kas ļauj atgriezt arī atbilstošo teksta aprakstu. To var panākt, izmantojot filtru un norādot etiķeti. Tātad, lai no tālāk norādītajiem datiem atgrieztu etiķetes 'color' vērtību:

   `elasticsearch_cluster_health_status{cluster="elasticsearch",color="green"} 1`
`elasticsearch_cluster 0`
`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 (kas balstīts uz skaitlisko vērtību '1') atlasīs atbilstošo rindu, savukārt etiķete atgriezīs veselības stāvokļa aprakstu (pašlaik 'green', bet potenciāli arī 'red' vai 'yellow').

Prometheus uz JSON

Datus no Prometheus var izmantot zemā līmeņa atklāšanai. Šajā gadījumā ir nepieciešami dati JSON formātā, un priekšapstrādes opcija *Prometheus to JSON* atgriezīs tieši to.

Plašāku informāciju skatiet sadaļā Atklāšana, izmantojot Prometheus datus.

Vaicājumu valodas salīdzinājums

Tālāk esošajā tabulā ir uzskaitītas atšķirības un līdzības starp PromQL un Zabbix Prometheus priekšapstrādes vaicājumu valodu.

PromQL instant vector selector	Zabbix Prometheus preprocessing
Atšķirības	

PromQL instant vector selector		Zabbix Prometheus preprocessing
Vaicājuma mērķis	Prometheus serveris	Vienkāršs teksts Prometheus izklāsta formātā
Atgriež	Tūlītējo vektoru	Metrikas vai etiķetes vērtību (Prometheus shēma) Vienas vērtības metrikas masīvu JSON formātā (Prometheus uz JSON)
Etiķešu saskaņošanas operatori	**=**, **!=**, **=~**, **!~**	**=**, **!=**, **=~**, **!~**
Regulārā izteiksme, ko izmanto etiķešu vai metrikas nosaukuma saskaņošanā	RE2	PCRE
Salīdzināšanas operatori	Skatiet sarakstu	Vērtību filtrēšanai tiek atbalstīts tikai **==** (vienāds ar)
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>

IPMI aģents

Pārskats

Zabbix ļauj pārraudzīt Intelligent Platform Management Interface (IPMI) ierīču stāvokli un pieejamību. Lai veiktu IPMI pārbaudes, Zabbix serveris sākotnēji ir jākonfigurē ar IPMI atbalstu.

IPMI ir standartizēta saskarne datorsistēmu attālinātai "lights-out" jeb "ārpusjoslas" pārvaldībai. Tā ļauj pārraudzīt aparatūras statusu tieši no tā dēvētajām "ārpusjoslas" pārvaldības kartēm neatkarīgi no operētājsistēmas vai no tā, vai dators vispār ir ieslēgts.

Zabbix IPMI pārraudzība darbojas tikai ierīcēm, kurām ir IPMI atbalsts (HP iLO, DELL DRAC, IBM RSA, Sun SSP u. c.).

IPMI pārvaldnieka process ieplāno IPMI pārbaudes, izmantojot IPMI aptaujātājus. Vienlaikus hosts vienmēr aptaujā tikai viens IPMI aptaujātājs, tādējādi samazinot atvērto savienojumu skaitu ar BMC kontrolleriem. Tāpēc IPMI aptaujātāju skaitu var droši palielināt, neuztraucoties par BMC kontrollera pārslogošanu. IPMI pārvaldnieka process tiek automātiski palaists, kad tiek palaists vismaz viens IPMI aptaujātājs.

Skatiet arī zināmās problēmas, kas attiecas 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, lai startētu kādus IPMI aptaujātājus, tāpēc jebkuri pievienotie IPMI vienumi nedarbosies. Lai to mainītu, atveriet Zabbix servera konfigurācijas failu (`zabbix_server.conf`) kā root lietotājs un atrodiet šādu rindu:

```
# StartIPMIPollers=0
```

Noņemiet komentāru un iestatiet aptaujātāju skaitu, piemēram, uz 3, lai tas izskatītos šādi:

```
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 jaunināts uz 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.

SSH aģents

Pārskats

SSH pārbaudes tiek veiktas kā monitorings 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 ieteicams izmantot libssh, nevis libssh2.

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.

Atslēgu faila autentifikācija

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, kur atradīsies publiskās un privātās atslēgas:

```
SSHKeyLocation=/home/zabbix/.ssh
```

Saglabājiet failu un pēc tam restartējiet Zabbix serveri.

Ceļš `/home/zabbix` šeit ir *zabbix* lietotāja konta mājas direktorijs, bet `.ssh` ir direktorijs, kur pēc noklusējuma publiskās un privātās atslēgas tiks ģenerētas ar ssh-keygen komandu mājas direktorijā.

Parasti dažādu OS distribūciju Zabbix servera instalācijas pakotnes izveido *zabbix* lietotāja kontu ar mājas direktoriju citur, piemēram, `/var/lib/zabbix` (kā sistēmas kontiem).

Pirms atslēgu ģenerēšanas jūs varētu pārvietot mājas direktoriju uz `/home/zabbix`, lai tas atbilstu iepriekš minētajam Zabbix servera `SSHKeyLocation` konfigurācijas parametram.

Note:
Tālāk norādītās darbības var izlaist, ja *zabbix* konts ir pievienots manuāli saskaņā ar instalācijas 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 un mēģinātu to pārvietot (ja tas eksistē), jāizpilda šāda komanda:

```
usermod -m -d /home/zabbix zabbix
```

Iespējams arī, ka vecajā atrašanās vietā mājas direktorijs neeksistēja, tāpēc tas jāizveido jaunajā vietā. Drošs mēģinājums 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 publisko un privāto atslēgu ģenerēšanas darbības var veikt ar šādām komandām (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):
<Leave empty>
##### Enter same passphrase again:
<Leave empty>
##### Your identification has been saved in /home/zabbix/.ssh/id_rsa.
##### Your public key has been saved in /home/zabbix/.ssh/id_rsa.pub.
##### The key fingerprint is:
##### 90:af:e4:c7:e3:f0:2e:5a:8d:ab:48:a2:0c:92:30:b9 zabbix@it0
##### The key's randomart image is:
##### +--[ RSA 2048]----+
##### |                 |
##### |        .        |
##### |      o          |
##### | .     o         |
##### |+     . S        |
##### |.+   o =         |
##### |E .   * =        |
##### |=o . ..* .       |
##### |... oo.o+        |
##### +-----------------+
```

Note:
Publiskās un privātās atslēgas (`id_rsa.pub` un `id_rsa`) pēc noklusējuma ir ģenerētas `/home/zabbix/.ssh` direktorijā, kas atbilst Zabbix servera `SSHKeyLocation` konfigurācijas parametram.

Attention:
Atslēgu tipi, kas nav "rsa", var būt atbalstīti ar ssh-keygen rīku un SSH serveriem, taču tos var neatbalstīt libssh2, ko izmanto Zabbix.

Shell konfigurācijas forma

Šī darbība jāveic tikai vienu reizi katram hostam, kas tiks uzraudzīts ar SSH pārbaudēm.

Izmantojot šādas komandas, **publiskās** atslēgas fails var tikt instalēts attālajā hostā *10.10.10.10*, lai SSH pārbaudes varētu tikt veiktas ar *root* kontu (lasāmības labad komandu uzvednes ir komentētas):

```
sudo -u zabbix ssh-copy-id root@10.10.10.10
##### The authenticity of host '10.10.10.10 (10.10.10.10)' can't be established.
##### RSA key fingerprint is 38:ba:f2:a4:b5:d9:8f:52:00:09:f7:1f:75:cc:0b:46.
##### Are you sure you want to continue connecting (yes/no)?
yes
##### Warning: Permanently added '10.10.10.10' (RSA) to the list of known hosts.
##### root@10.10.10.10's password:
<Enter root password>
##### Now try logging into the machine, with "ssh 'root@10.10.10.10'",
##### and check to make sure that only the key(s) you wanted were added.
```

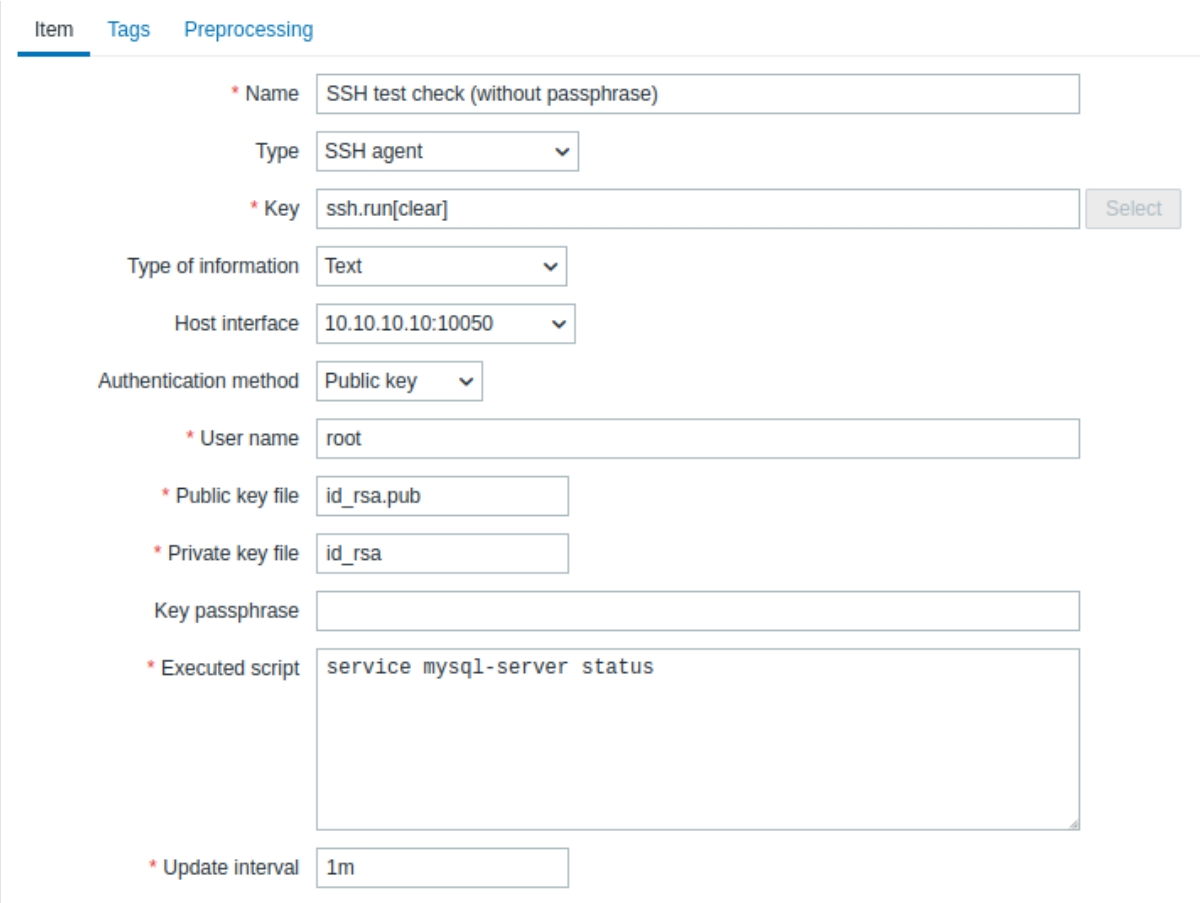
Tagad ir iespējams pārbaudīt SSH pieteikšanos, izmantojot noklusējuma privāto atslēgu (`/home/zabbix/.ssh/id_rsa`) lietotāja kontam *zabbix*:

```
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

Faktiskās izpildāmās komandas jāievieto laukā *Executed script* vienuma konfigurācijā. Vairākas komandas var izpildīt vienu pēc otras, ievietojot tās jaunā rindā. Šādā 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:

Parametrs	Apraksts	Komentāri
Type	Šeit atlasiet **SSH agent**.	

Parametrs	Apraksts	Komentāri

Key	Unikāla (katram hostam) vienuma atslēga formātā `ssh.run[unique short description,<ip>,<port>,<encoding>,<ssh options>,<subsystem>]`	**unique short description** ir obligāts, un tam jābūt unikālam katram SSH vienumam konkrētajā hostā.
		
		Noklusējuma ports ir 22, nevis ports, kas norādīts saskarnē, kurai piešķirts šis vienums.
		
		ssh options ļauj nodot papildu SSH opcijas formātā `key1=value1;key2=value2,value3`. Vienai atslēgai var nodot vairākas ar komatu atdalītas vērtības (šādā gadījumā parametrs jāievieto pēdiņās); vairākas opciju atslēgas var atdalīt 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 "!" zīme 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 ar konkrētām apakšsistēmas atļautām darbībām (piemēram, failu pārsūtīšanu, izmantojot SFTP, vai tīkla ierīču pārvaldību, izmantojot NETCONF). Ņemiet vērā, ka apakšsistēmas izmantošanai var būt nepieciešama arī īpaša skripta sintakse laukā *Executed script*.
		
		Piemēri:
		=>
		`ssh.run[SFTPBackup,192.0.2.18,,,,sftp]`
		=>
		`ssh.run[Cisco1234,192.0.2.18,,,,netconf]`

Authentication method	Viena no opcijām: "Password" vai "Public key".	
User name	Lietotājvārds (līdz 255 rakstzīmēm), ko izmanto 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.

Parametrs	Apraksts	Komentāri
Password vai *Key passphrase*	Parole (līdz 255 rakstzīmēm) autentifikācijai vai ieejas frāze **ja** tā tika izmantota privātajai atslēgai.	Atstājiet lauku *Key passphrase* tukšu, ja ieejas frāze netika izmantota. Skatiet arī informāciju par zināmajām problēmām, kas saistītas ar ieejas frāzes izmantošanu.

Executed script	Izmantojot SSH attālo sesiju, izpildāmā čaulas komanda vai komandas.	Izpildītās čaulas komandas atgrieztā vērtība ir ierobežota līdz 16 MB (ieskaitot beigu atstarpes, kas tiek nogrieztas); tiek piemēroti arī datu bāzes ierobežojumi.
		
		Ņemiet vērā, ka libssh2 bibliotēka var saīsināt izpildāmos skriptus līdz aptuveni 32 kB.
		
		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>`
		`]]>]]>`

Telnet aģents

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 aprak-sts>,<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.

JMX aģents

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.

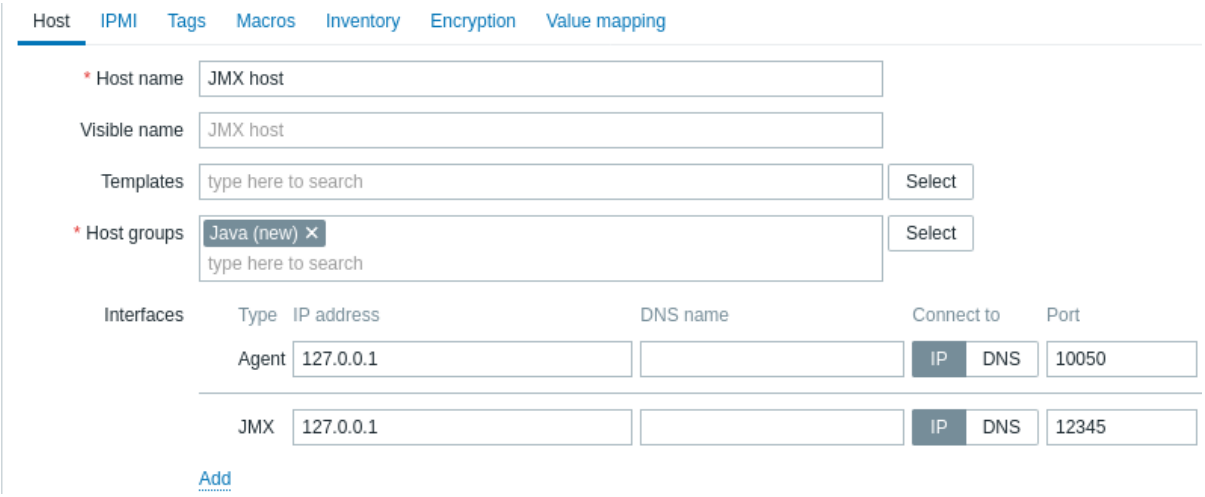
Detalizētu aprakstu skatiet Monitoring and Management Using JMX.

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.

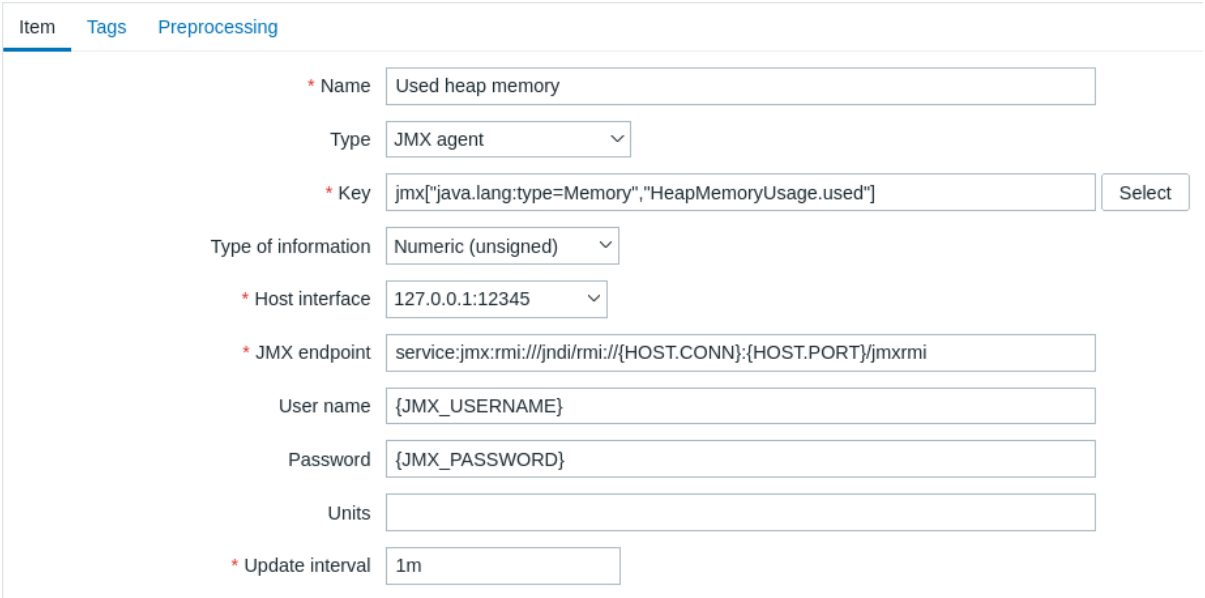


Visi obligātie ievades lauki ir atzīmēti ar sarkanu zvaigznīti.

JMX aģenta vienuma pievienošana

Katram JMX skaitītājam, kas jūs interesē, pievienojiet **JMX aģents** vienumu, kas piesaistīts šai saskarnei.

Tālāk redzamajā ekrānuzņēmumā atslēga 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:

Type	Šeit iestatiet **JMX agent**.
Key	`jmx[]` vienuma atslēga satur trīs parametrus: **object name** - MBean objekta nosaukums **attribute name** - MBean atribūta nosaukums ar izvēles saliktās datu lauku nosaukumiem, kas atdalīti ar punktiem **unique short description** - unikāls apraksts, kas ļauj vienam hostam izmantot vairākus JMX vienumus ar vienādu objekta nosaukumu un atribūta nosaukumu (neobligāti) Skatiet tālāk, lai iegūtu vairāk informācijas par JMX vienumu atslēgām. Jūs varat atklāt MBean un MBean atribūtus, izmantojot `jmx.discovery[]` zemā līmeņa atklāšanas vienumu.
JMX endpoint	Jūs varat norādīt pielāgotu JMX galapunktu. Pārliecinieties, ka JMX galapunkta savienojuma parametri atbilst JMX saskarnei. To var panākt, izmantojot `{HOST.*}` makro, kā tas ir izdarīts noklusējuma JMX galapunktā. Tiek atbalstīti `{HOST.*}` makro un lietotāja makro.

User name	Norādiet lietotājvārdu (līdz 255 rakstzīmēm), ja jūsu Java lietojumprogrammā ir konfigurēta autentifikācija. Tiek atbalstīti lietotāja makro.
Password	Norādiet paroli (līdz 255 rakstzīmēm), ja jūsu Java lietojumprogrammā ir konfigurēta autentifikācija. Tiek atbalstīti lietotāja makro.

Ja vēlaties uzraudzīt Boolean skaitītāju, kas ir vai nu ”true”, vai ”false”, tad norādiet informācijas tipu kā ”Numeric (unsigned)” un cilnē Preprocessing atlasiet priekšapstrādes soli ”Boolean to decimal”. Serveris saglabās Boolean 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 arī ir viss. Veiksmīgu JMX uzraudzību!

Neprimitīvie datu tipi

Ir iespējams strādāt ar pielāgotiem MBeans, kas atgriež neprimitīvos datu tipus, kuri pārraksta **toString()** metodi.

Pielāgota endpoint izmantošana ar JBoss EAP 6.4

Pielāgoti endpointi ļauj strādāt ar dažādiem transporta protokoliem, ne tikai ar noklusējuma RMI.

Lai parādītu šo iespēju, mēģināsim kā piemēru konfigurēt JBoss EAP 6.4 uzraudzību. Vispirms pieņemsim dažus nosacījumus:

- Jums jau ir instalēts Zabbix Java gateway. Ja nē, to var izdarīt saskaņā ar dokumentāciju.
- Zabbix serveris un Java gateway ir instalēti ar prefiksu `/usr/local/`
- JBoss jau ir instalēts `/opt/jboss-eap-6.4/` un darbojas standalone režīmā
- Pieņemsim, ka visi šie komponenti darbojas vienā hostā
- Ugunsmūris un SELinux ir atspējoti (vai attiecīgi konfigurēti)

Veiksim dažus vienkāršus iestatījumus failā `zabbix_server.conf`:

```
JavaGateway=127.0.0.1
StartJavaPollers=5
```

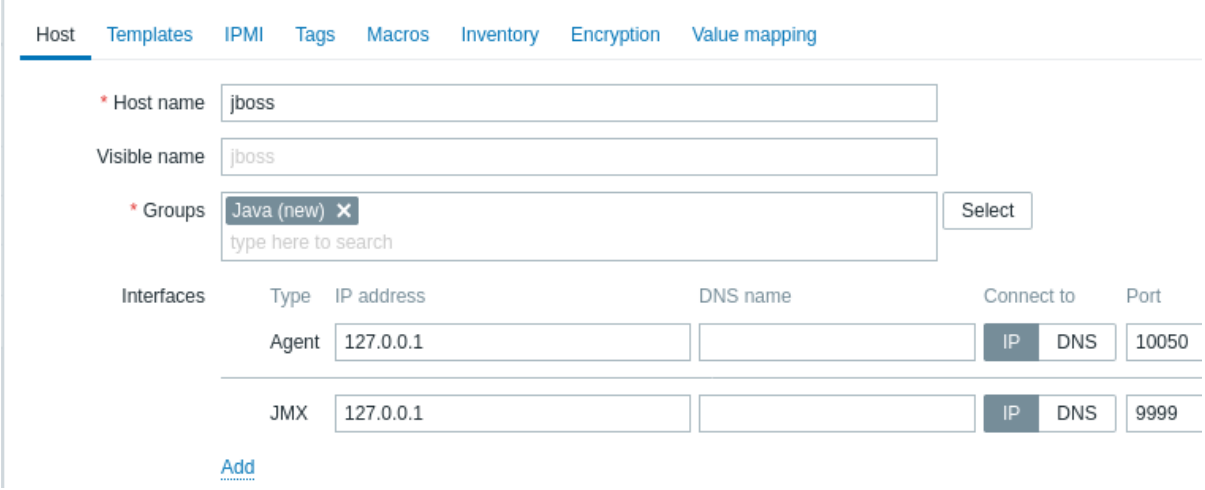
Un konfigurācijas failā `zabbix_java/settings.sh` (vai `zabbix_java_gateway.conf`):

```
START_POLLERS=5
```

Pārbaudiet, vai JBoss klausās savā standarta pārvaldības portā:

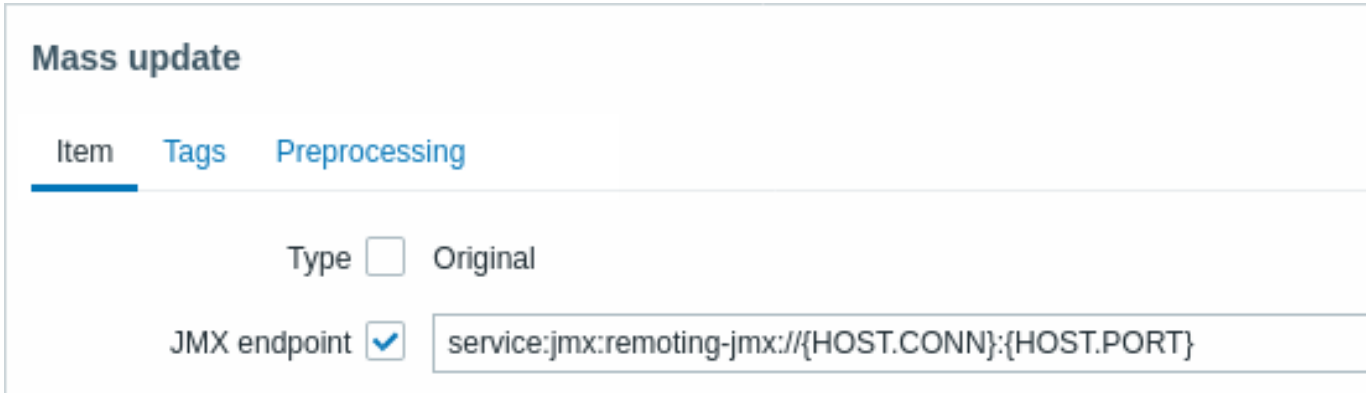
```
$ netstat -natp | grep 9999
tcp        0      0 127.0.0.1:9999          0.0.0.0:*               LISTEN      10148/java
```

Tagad izveidosim hostu ar JMX interfeisu 127.0.0.1:9999 Zabbix.



Tā kā zināms, ka šī JBoss versija izmanto JBoss Remoting protokolu RMI vietā, mēs varam attiecīgi masveidā atjaunināt JMX endpoint 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 tikt parādīta 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 `~/needed_modules.txt` failu 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 uzgaidīt un, ja viss ir izdarīts pareizi, redzēt, ka JMX uzraudzības dati sāk nonākt Zabbix (skatiet arī: Latest data).

Aprēķinātais vienums

Pārskats

Šis vienuma tips ļauj veikt aprēķinus, izmantojot jau esošus metrikas datus. Aprēķinātais vienums pats par sevi nekādus datus neapkopo.

Piemēram, varat:

- Saskaitīt divas vērtības.
- Aprēķināt vairāku vienumu vērtību vidējo.
- Aprēķināt vienumu grupas kopējo vērtību.

Iegūtā vērtība tiek saglabāta Zabbix datubāzē tāpat kā jebkuram citam vienumam; tiek saglabātas gan vēstures, gan tendenču vērtības, un var izveidot grafikus.

Aprēķins tiek veikts, pamatojoties uz formulu. Formulas sintakse ir tāda pati kā trigeru izteiksmēm.

Faktiski trigerus var uzturēt vienkāršus, sarežģīto datu pārveides loģiku ievietojot aprēķinātajā vienumā un pēc tam trigerī atsaucoties uz aprēķināto vienumu, izmantojot vienuma atslēgu.

Datu avota norādīšanai aprēķiniem ir divas iespējas:

1. Norādīt vienumus formulā.
2. Veikt datu apkopošanu, pamatojoties uz filtru, kas atlasa vienumus.

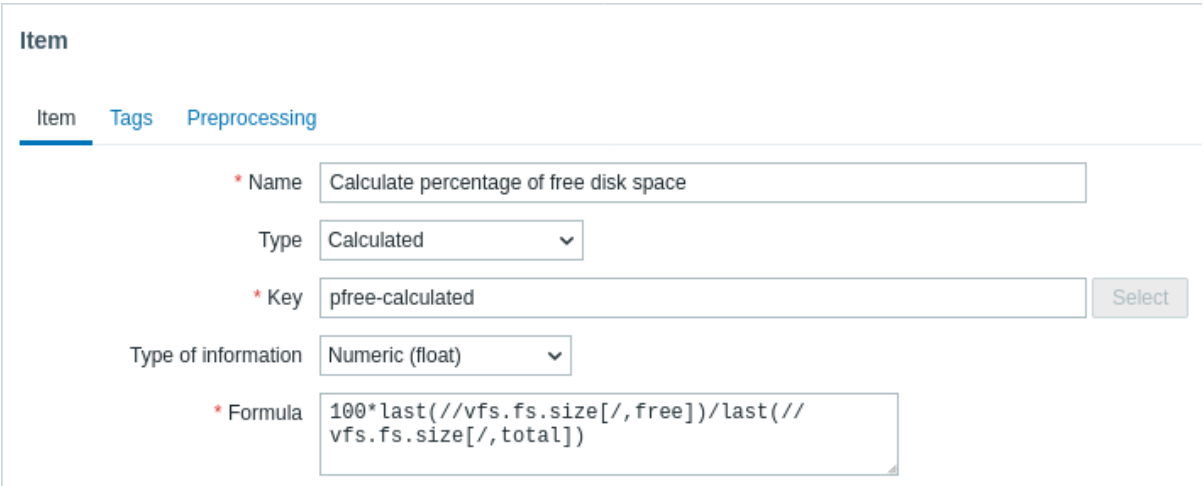
Otrā iespēja ļauj aprēķināt grupu vidējās vērtības, kopsummas u. c.

Visus aprēķinus veic Zabbix serveris, un tos apstrādā `history poller` procesi; starpniekserveris vai aģenti šajā procesā vispār netiek iesaistīti. Ņemiet vērā, ka plaša aprēķināto vienumu izmantošana ar gariem laika periodiem var ietekmēt servera veiktspēju.

Ņemiet vērā, ka aprēķinātie vienumi var piekļūt citu vienumu vēsturei, tostarp tādu vienumu vēsturei, kuriem lietotājam, kas izveido aprēķināto vienumu, nav piekļuves. Ja šāda darbība nav vēlama, varat atspējot aprēķinātos vienumus, iestatot `StartHistoryPollers` konfigurācijas parametru uz 0.

Konfigurējamie lauki

Lai izmantotu aprēķinātos vienumus, izvēlieties vienuma tipu **Calculated**.



key ir unikāls vienuma identifikators (katram hostam). Varat izveidot jebkādu atslēgas nosaukumu, izmantojot atbalstītos simbolus. Atslēga ļauj atsaukties uz aprēķināto vienumu trigeros.

Aprēķina definīcija jāievada laukā **Formula**.

Starp formulu un atslēgu nav nekādas saistības. Atslēgas parametri formulā nekādā veidā netiek izmantoti.

Atšķirībā no trigeru izteiksmēm, Zabbix apstrādā aprēķinātos vienumus atbilstoši vienuma atjaunināšanas intervālam, nevis pēc jaunas vērtības saņemšanas.

Ņemiet vērā: ja kā informācijas tipu izvēlaties *Numeric (unsigned)* un aprēķina rezultāts ir peldošā komata vērtība, tā tiek noapaļota līdz veselam skaitlim.

Vienkāršas aprēķina formulas sintakse

Vienkāršas formulas sintakse ir:

`function(/host/key,<parameter1>,<parameter2>,...)`

kur:

Element	Description
`function`	Viena no atbalstītajām funkcijām: last, min, max, avg, count utt.
`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. Ja maināt atsauktā vienuma atslēgu, atslēga formulā ir jāatjaunina manuāli. Visiem vienumiem, uz kuriem atsaucas vēstures funkcijas, ir jāeksistē un jāvāc dati.
`parameter(s)`	Funkcijas parametri, ja nepieciešams. Tiek atbalstīti laika sufiksi un atmiņas lieluma sufiksi.

Sarežģītāka formula var izmantot funkciju, operatoru un iekavu kombināciju.

Skatiet aprēķināto vienumu formulu piemērus:

- (1) Kopējā datplūsma saskarnē
- (2) Brīvās diska vietas procentuālā daļa
- (3) Ienākošās datplūsmas procentuālā daļa
- (4) Apstrādāto vērtību vidējais skaits

Dažus mainīgos formulai var nodot ar lietotāja makrosiem.
Ņemiet vērā, ka lietotāja makrosi:

- tiks atrisināti — ja tie tiek izmantoti, lai norādītu funkcijas parametru, vienuma filtra parametru vai konstanti.
- **netiks** atrisināti — ja tie norāda uz funkciju, hosta nosaukumu, vienuma atslēgu, vienuma atslēgas parametru vai operatoru.

Ir atļauta salīdzināšana ar virknēm.

Ja kešatmiņā nav nesenu datu un funkcijā nav definēts vaicājuma periods, Zabbix pēc noklusējuma meklēs datubāzē vēsturiskās vērtības līdz pat vienai nedēļai pagātnē.

Aprēķināts vienums var kļūt neatbalstīts vairākos gadījumos:

- atsauktais vienums
  - nav atrasts
  - ir atspējots vai pieder atspējotam hostam
  - nav atbalstīts (izņemot ar nodata() funkciju un operatoriem ar nezināmām vērtībām)
- nav datu funkcijas aprēķināšanai
- dalīšana ar nulli
- izmantota nepareiza sintakse

Datu agregācijas

Datu agregācijas ir balstītas uz filtru, kas atlasa vairākus vienumus. Datu agregācijas arī ir aprēķināto vienumu tips, taču formulā tiek izmantota nedaudz atšķirīga sintakse.

Lai iegūtu agregētās vērtības, 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 norādīt vairākus vienumus kā agregācijas parametrus:

```
aggregate_function(function(/host/key,parameter),function(/host2/key2,parameter),...)
```

Ņemiet vērā, ka `function` šeit jābūt vēstures/tendenču funkcijai.

Skatiet agregēto aprēķinu piemērus:

- (5) Kopējā ienākošā datplūsma hostam
- (6) Kopējā diska vieta grupai
- (7) Vidējā procesora slodze grupai

- (8) Vidējā procesora slodze grupai un birkai
- (9) Vidējais apstrādāto vērtību skaits
- (10) Vidējais vaicājumu skaits grupai
- (11) Kopējais veiksmīgo DNS pārbaužu skaits
- (12) Kopējais neatbalstīto vienumu skaits grupai
- (13) Saskarnes statistika grupai

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

Agregētais aprēķins var kļūt neatbalstīts, ja:

- nav atrasts neviens no atsauktajiem vienumiem (tas var notikt, ja vienuma atslēga ir nepareiza, neviens no vienumiem neeksistē vai visas iekļautās grupas ir nepareizas)
- funkcijas aprēķinam nav datu

Aprēķināto vienumu piemēri

(1) Kopējā datplūsma saskarnē

```
last(//net.if.in[eth0,bytes])+last(//net.if.out[eth0,bytes])
```

Aprēķināt kopējo joslas platumu uz eth0.

(2) Brīvās diska vietas procentuālā daļa

```
100*last(//vfs.fs.size[/,free])/last(//vfs.fs.size[/,total])
```

Zabbix ņems jaunākās brīvās un kopējās diska vietas vērtības (uz /) un aprēķinās procentuālo daļu saskaņā ar doto formulu.

(3) Ienākošās datplūsmas procentuālā daļa

```
100*last(//net.if.in[eth0,bytes])/(last(//net.if.in[eth0,bytes])+last(//net.if.out[eth0,bytes]))
```

Aprēķina ienākošās datplūsmas procentuālo daļu no kopējās datplūsmas.

(4) Apstrādāto vērtību vidējais skaits

```
avg(/Zabbix server/zabbix[wcache,values],10m)
```

Aprēķina 10 minūšu vidējo Zabbix apstrādāto vērtību skaitu.

(5) Kopējā ienākošā datplūsma hostam

```
sum(last_foreach(/host/net.if.in[*]))
```

Aprēķina visu hosta vienumu, kas atbilst `net.if.in[*]`, kopējo summu.

(6) Kopējā diska vieta grupai

```
sum(last_foreach(/*/vfs.fs.size[/,total]?[group="MySQL Servers"]))
```

Aprēķināt kopējo diska vietu visiem `MySQL Servers`.

(7) Vidējā procesora slodze grupai

```
avg(last_foreach(/*/system.cpu.load[,avg1]?[group="MySQL Servers"]))
```

Aprēķināt vidējo procesora slodzi visiem `MySQL Servers`.

(8) Vidējā procesora slodze grupai un birkai

```
avg(last_foreach(/*/system.cpu.load?[(group="Servers A" or group="Servers B" or group="Servers C") and (ta
```

Aprēķināt vidējo CPU slodzi visos hostos vairākās hostu grupās, kuriem ir noteiktas birkas.

(9) Vidējais vaicājumu skaits grupai

```
avg(avg_foreach(/*/mysql.qps?[group="MySQL Servers"],5m))
```

Aprēķināt 5 minūšu vidējo vaicājumu skaitu sekundē grupai `MySQL Servers`.

(10) Kopējais veiksmīgo DNS pārbaužu skaits

```
sum(last_foreach(/*/net.dns[*,*,*]))
```

Aprēķina kopējo veiksmīgo DNS pārbaužu skaitu visos hosts.

Parādītā vienuma forma ir `net.dns[192.0.2.1,example.com,A]` kā viena iespējamā atslēgas piemērs.

Ņemiet vērā, ka aizstājējzīmēm ir jāsakrīt ar parametru skaitu atslēgā (šeit `net.dns` ir trīs parametri: ip, name, type).

(11) Kopējais neatbalstīto vienumu skaits grupai

```
sum(last_foreach(/*/zabbix[host,,items_unsupported]?[group="Zabbix servers"]))
```

Aprēķina kopējo neatbalstīto vienumu skaitu priekš `Zabbix servers`.

(12) Saskarnes statistika grupai

```
sum(last_foreach(/*/net.if.out[eth0,bytes]?[group="video"])) / sum(last_foreach(/*/nginx_stat.sh[active]?[
```

Aprēķināt saskarnes statistiku grupai `video`.

Pareizas/nepareizas sintakses piemēri

Izteiksmes (tostarp funkciju izsaukumus) nevar izmantot kā vēstures, tendenču vai foreach funkcijas parametrus. Tomēr šīs funkcijas pašas var izmantot citos (ar vēsturi nesaistītos) funkciju parametros.

Izteiksme	Piemērs
Derīga	`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)))`
Nederīga	`sum(/host/key,10+2)`
	`sum(/host/key, avg(10,2))`
	`sum(/host/key,last(/host/key2))`

Ņemiet vērā, ka tādā izteiksmē kā:

```
sum(sum_foreach(//resptime[*],5m))/sum(count_foreach(//resptime[*],5m))
```

(kur `resptime[*]` ir vienuma atslēgas paraugs, kas atbilst atslēgām `resptime[1]`, `resptime[2]` utt. un tiek izmantots, lai iegūtu atbildes laiku no vairākiem aktīviem klastera mezgliem), nevar garantēt, ka abām vienādojuma daļām vienmēr būs vienāda vērtību kopa. Kamēr tiek aprēķināta viena izteiksmes daļa, var pienākt jauna vērtība pieprasītajam periodam, un tad otrai izteiksmes daļai būs cita vērtību kopa.

Atkarīgais vienums

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.

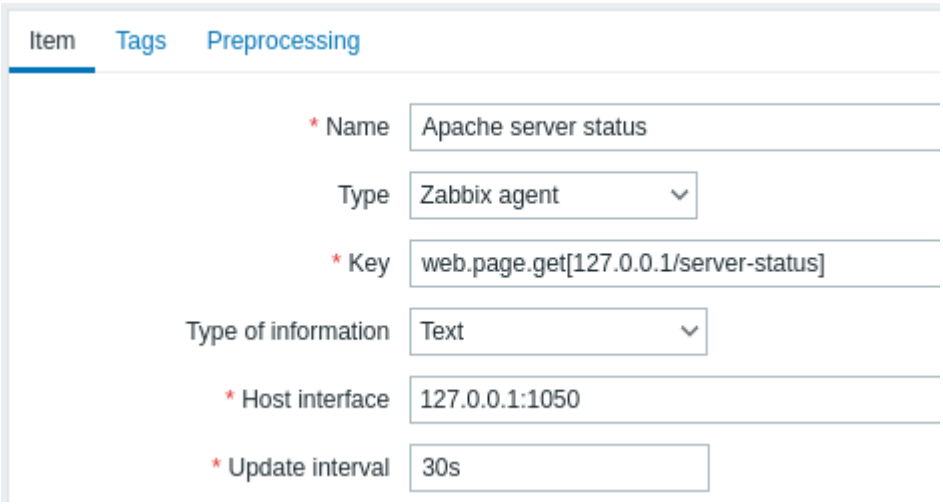
Ierobežojumi

- 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 **galvenais vienums** vispirms ir jākonfigurē (vai tam jau jāeksistē):

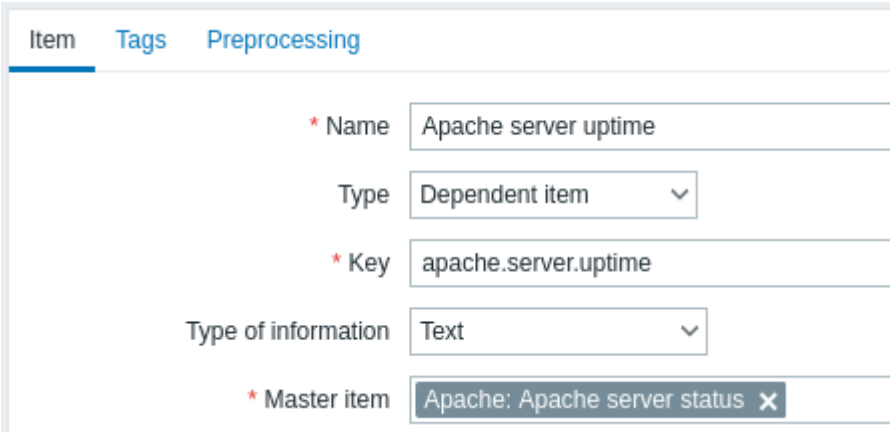
1. Dodieties uz *Datu vākšana* > *Hosts*.
2. Hosta rindā noklikšķiniet uz *Vienumi*.
3. Noklikšķiniet uz *Izveidot vienumu*.
4. Formā ievadiet vienuma parametrus.



Visi obligātie ievades lauki ir atzīmēti ar sarkanu zvaigznīti.

Noklikšķiniet uz *Pievienot*, lai saglabātu galveno vienumu.

Pēc tam varat konfigurēt **atkarīgo vienumu**.

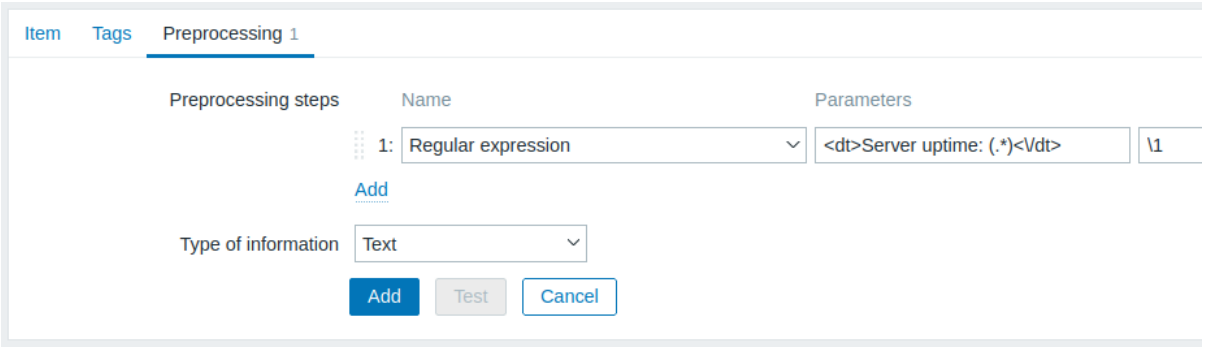


Visi obligātie ievades lauki ir atzīmēti ar sarkanu zvaigznīti.

Lauki, kuriem atkarīgajiem vienumiem nepieciešama specifiska informācija, ir:

Type	Šeit atlasiet **Dependent item**.
Key	Ievadiet atslēgu, kas tiks izmantota vienuma identificēšanai.
Master item	Atlasiet galveno vienumu. Galvenā vienuma vērtība tiks izmantota atkarīgā vienuma vērtības aizpildīšanai.
Type of information	Atlasiet informācijas tipu, kas atbildīs saglabājamo datu formātam.

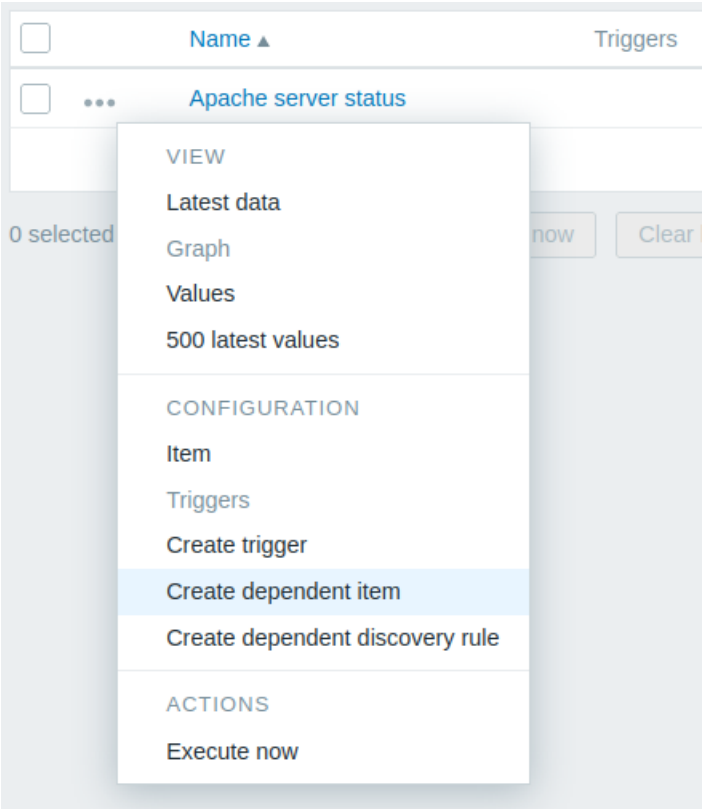
Varat izmantot vienuma vērtības priekšapstrādi, lai izvilktu nepieciešamo galvenā vienuma vērtības daļu.



Bez priekšapstrādes atkarīgā vienuma vērtība būs tieši tāda pati kā galvenā vienuma vērtība.

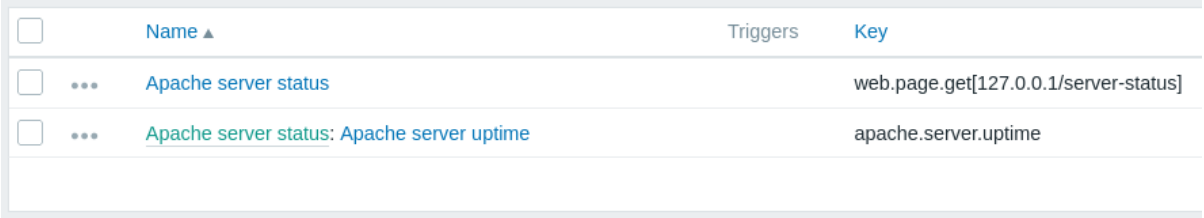
Noklikšķiniet uz *Pievienot*, lai saglabātu atkarīgo vienumu.

Ātrāku saīsni atkarīgā vienuma izveidei var atvērt, vienumu sarakstā noklikšķinot uz ••• pogas un atlasot *Izveidot atkarīgo vienumu*.



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.

Skripta vienums

Pārskats

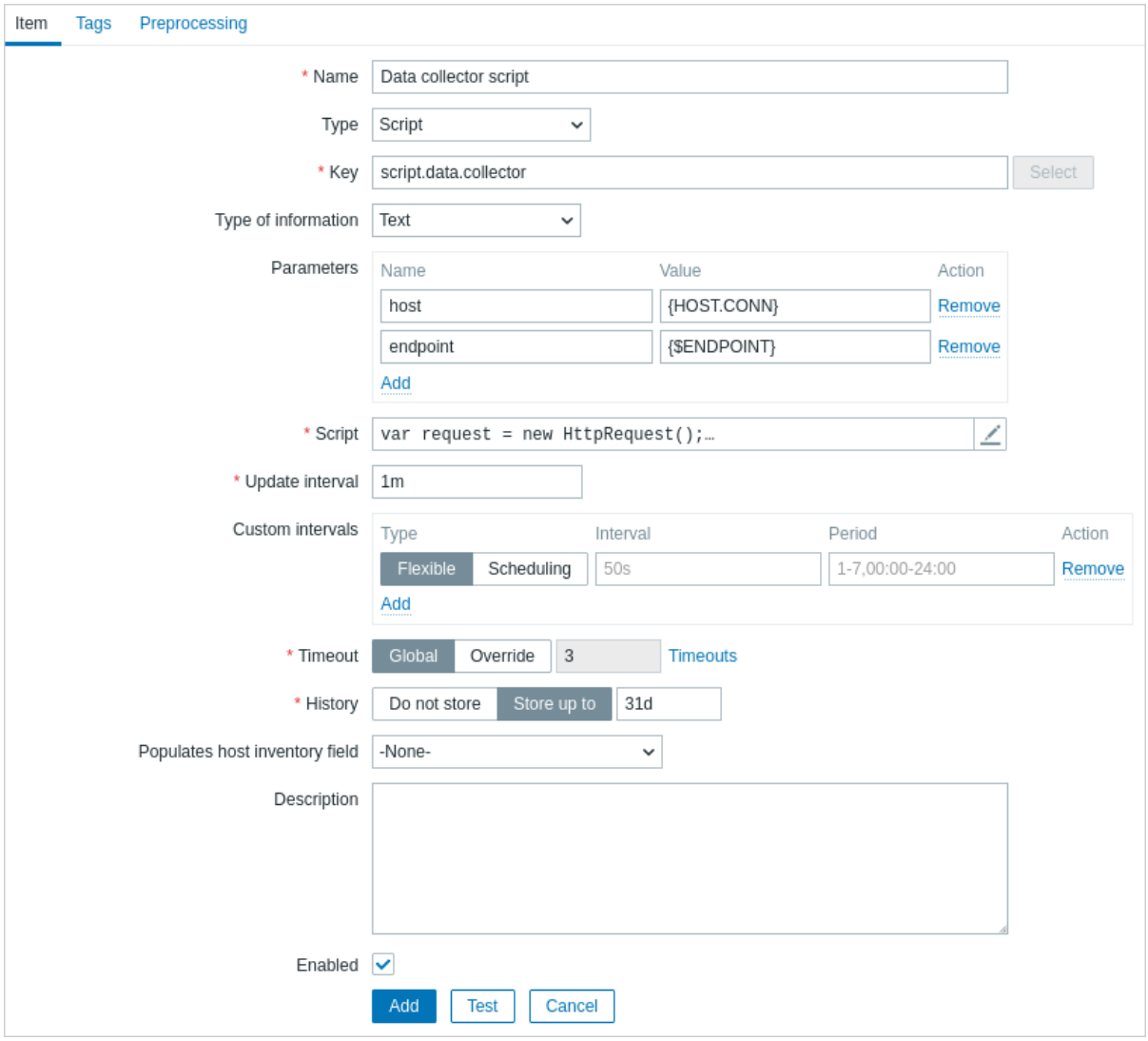
Skripta vienumus var izmantot datu vākšanai, izpildot lietotāja definētu JavaScript kodu ar iespēju iegūt datus, izmantojot HTTP/HTTPS. Papildus skriptam var norādīt neobligātu parametru sarakstu (nosaukuma un vērtības pārus) un noildzi.

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:

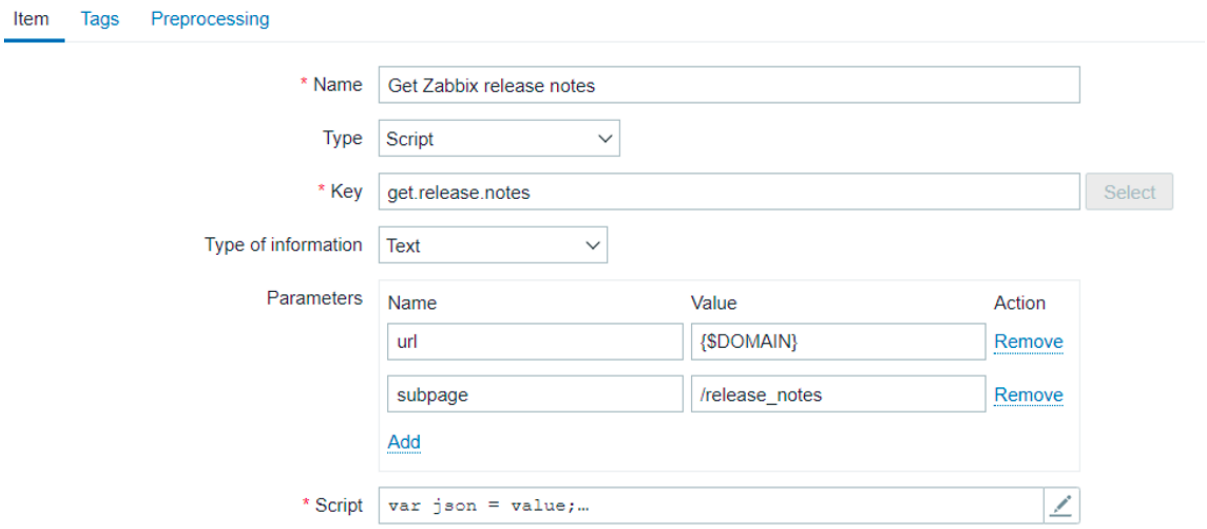
- Izveidojiet vienumu ar tipu "Script".
- Laukā *Script* ievadiet:

```
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;
```

Pārlūka vienums

Pārskats

Browser vienumi ļauj uzraudzīt sarežģītas tīmekļa vietnes un tīmekļa lietotnes, izmantojot pārlūkprogrammu.

Attention:
Browser vienumu atbalsts pašlaik ir eksperimentāls.

Browser vienumi apkopo datus, izpildot lietotāja definētu JavaScript kodu un izgūstot datus, izmantojot HTTP/HTTPS. Šis vienums var simulēt tādas ar pārlūkprogrammu saistītas darbības kā klikšķināšana, teksta ievade, navigācija tīmekļa lapās un citas lietotāja mijiedarbības ar tīmekļa vietnēm vai tīmekļa lietotnēm.

Papildus skriptam var norādīt arī izvēles parametru sarakstu (nosaukuma un vērtības pārus) un taimautu.

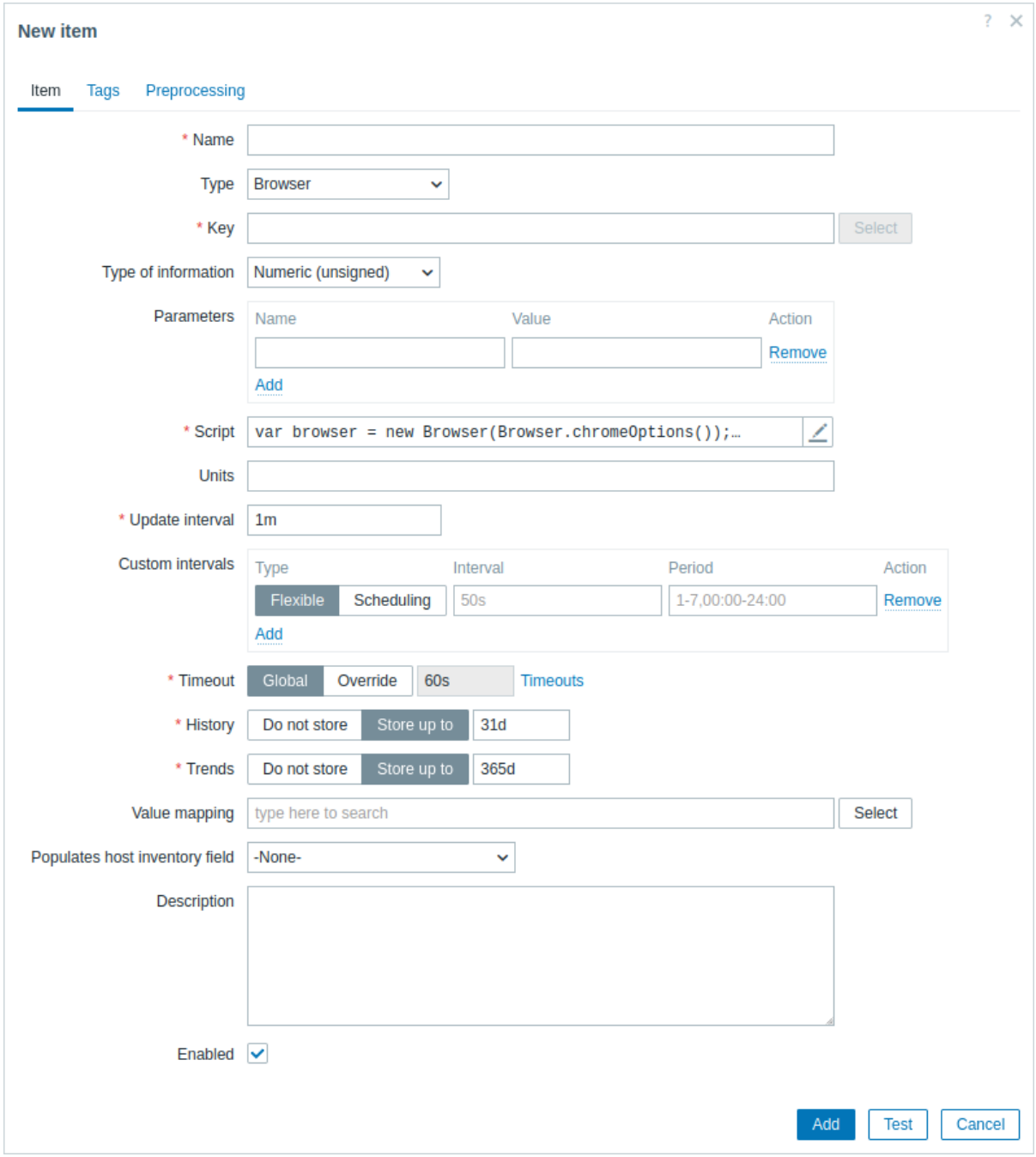
Attention:
Šis vienums daļēji ievieš W3C WebDriver standartu ar Selenium Server vai parastu WebDriver (piemēram, ChromeDriver) kā tīmekļa testēšanas galapunktu. Lai vienums darbotos, Zabbix serveris/starpniekserveris konfigurācijas failā iestatiet `WebDriverURL` parametru (ja izmantojat ChromeDriver, skatiet Drošības apsvērumus). Lai uzlabotu veiktspēju, apsveriet iespēju izmantot atsevišķu serveri tīmekļa testēšanas videi.

Browser vienumu pārbaudes izpilda un apstrādā Zabbix serveris vai starpniekservera browser pollers. Ja nepieciešams, Zabbix serveris/starpniekserveris konfigurācijas failā varat pielāgot browser pollers iepriekš izveidoto instanču skaitu, izmantojot `StartBrowserPollers` parametru.

Sarežģītu tīmekļa vietņu un tīmekļa lietotņu uzraudzībai ir pieejama Website by Browser veidne kā gatava lietošanai 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 uzzinātu, kā iestatīt Zabbix vietņu uzraudzībai, izmantojot *Website by Browser* veidni, skatiet Monitor websites with Browser items.

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);
}
```

Telemetrijas vaicājums

Šis vienuma tips ļauj izgūt vai apkopot telemetrijas datus. Telemetrijas dati tiek izmantoti lietojumprogrammu veiktspējas uzraudzībai (APM), kas palīdz noskaidrot, kā pieprasījumi plūst caur jūsu lietojumprogrammām un pakalpojumiem, izmērīt to veiktspēju un atrast lēnu vai neveiksmīgu darbību cēloni.

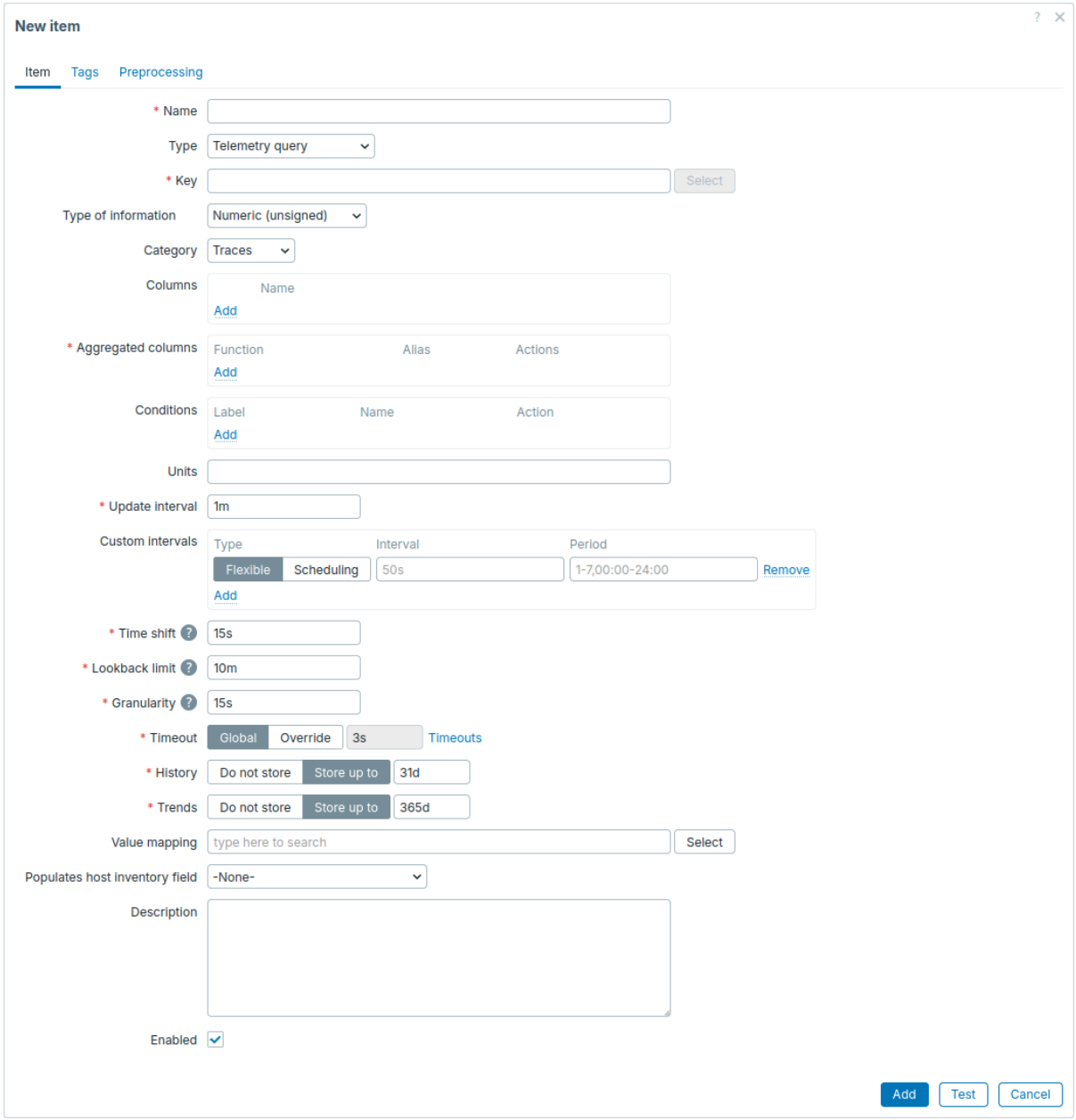
Telemetrijas dati ir trīs veidu: trasējumi (ceļš, ko pieprasījums veic caur jūsu lietojumprogrammām un pakalpojumiem), metrikas (mērījumi, piemēram, atbildes laiks) un žurnāli (notikumu teksta ieraksti). Izmantojot telemetrijas vaicājuma vienumu, varat, piemēram, atrast visus trasējumus, kuros pakalpojums atgrieza kļūdu, iegūt vidējo atbildes laiku no metrikām vai meklēt žurnālos konkrētu kļūdas ziņojumu.

Konfigurējot telemetrijas vaicājuma vienumu, jūs definējat instrukciju kopu (vaicājumu), kas norāda Zabbix, kādus telemetrijas datus pieprasīt, kā tos filtrēt un kā apkopot. Kad tiek izpildīta vienuma pārbaude, Zabbix nosūta šo vaicājumu telemetrijas datu avotam un izgūst rezultātu, pamatojoties uz vaicājumu.

Lai izmantotu šo vienumu, norādiet telemetrijas datu avotu un iestatiet parametru `StartTelemetryQueryPollers` Zabbix servera vai Zabbix starpniekservera konfigurācijā atkarībā no tā, kurš no tiem uzrauga vienuma hostu.

Konfigurācija

Veidojot vienumu, iestatiet tā tipu uz *Telemetry query*:



Parametrs	Apraksts
Name	Skatiet parametrus, kas ir kopīgi visiem vienumiem.
Type	Atlasiet *Telemetry query*.
Key	Skatiet parametrus, kas ir kopīgi visiem vienumiem.
Type of information	
Category	Atlasiet signāla tipu, par kuru vēlaties veikt vaicājumu: *Traces*, *Metrics* vai *Logs*.
Metric points	Atlasiet mērījuma punkta tipu, par kuru vēlaties veikt vaicājumu: *Sum*, *Gauge*, *Histogram* vai *Exponential histogram*. Šis parametrs ir pieejams, ja *Category* ir iestatīts uz *Metrics*.
Columns	Pievienojiet telemetrijas datu kolonnas, par kurām vēlaties veikt vaicājumu. Kolonnām, kurās tiek glabāti atslēgu un vērtību pāri, ievadiet arī atslēgas nosaukumu. Šo kolonnu ieraksti tiks iekļauti vienuma vērtībā bez izmaiņām.
Aggregated columns	Pievienojiet telemetrijas datu kolonnas, par kurām vēlaties veikt vaicājumu, un norādiet, kā aprēķināt vērtību no to ierakstiem: **Function** - atlasiet funkciju, kas jāpiemēro kolonnas ierakstiem: *min*, *max*, *avg*, *count*, *sum* vai *percentile*. **Percentage** - ievadiet procentili, kas jāaprēķina no kolonnas ierakstiem (ja *Function* ir iestatīts uz *percentile*). **Column** - atlasiet kolonnu, no kuras jāveic aprēķins (ja *Function* nav iestatīts uz *count*). **Alias** - ievadiet unikālu aprēķinātās vērtības nosaukumu, kas tiks izmantots tās identificēšanai vienuma vērtībā.

Parametrs	Apraksts
Type of calculation	Atlasiet, kā novērtēt nosacījumus, kas iestatīti tālāk norādītajā parametrā *Conditions*: **And/Or** - jāizpilda visi nosacījumi; nosacījumi ar vienādu *Type* un *Attribute key* tiek apvienoti ar *Or*. **And** - jāizpilda visi nosacījumi. **Or** - jāizpilda vismaz viens nosacījums. **Custom expression** - nosacījumi jāizpilda atbilstoši formulā norādītajam, piemēram, `A and (B or C)`.
Conditions	Šis parametrs ir pieejams, ja esat iestatījis divus vai vairākus nosacījumus. Pievienojiet nosacījumus telemetrijas datu kolonnu ierakstu filtrēšanai: **Type** - atlasiet telemetrijas datu kolonnu, kurai tiks piemērots nosacījums. **Attribute key** - ievadiet atslēgu, kurai tiks piemērots nosacījums (ja *Type* ir iestatīts uz kolonnu ar atslēgu un vērtību pāriem). **Operator** - atlasiet izmantojamo salīdzināšanas operatoru (skatiet tālāk) starp ieraksta vērtību un *Attribute value*. **Attribute value** - ievadiet vērtību, ar kuru jāsalīdzina ieraksta vērtība (ja *Operator* nav iestatīts uz *Exists*). Ja *Type* ir iestatīts uz kolonnu ar vienu vērtību, nosacījums tiks piemērots kolonnas ierakstam; ja kolonnā tiek glabāts vērtību masīvs, nosacījums atbildīs ierakstam, ja vismaz viena masīva vērtība atbildīs nosacījumam. Ja *Type* ir iestatīts uz kolonnu, kurā tiek glabāti atslēgu un vērtību pāri, nosacījums tiks piemērots atslēgas vērtībai, kas norādīta laukā *Attribute key*; ja kolonnā tiek glabāts atslēgu un vērtību pāru masīvs, nosacījums atbildīs ierakstam, ja vismaz viens pāris ar norādīto *Attribute key* atbildīs nosacījumam. Kolonnām ar vienu vērtību varat izmantot šādus operatorus: **Equals** - atrod ierakstus, kas ir vienādi ar *Attribute value* (reģistrjutīgi). **Does not equal** - atrod ierakstus, kas nav vienādi ar *Attribute value* (reģistrjutīgi). **Contains** - atrod ierakstus, kuros ir ietverta *Attribute value* (reģistrjutīgi). **Does not contain** - atrod ierakstus, kuros nav ietverta *Attribute value* (reģistrjutīgi). Kolonnām, kurās tiek glabāti atslēgu un vērtību pāri, varat izmantot šādus operatorus: **Equals** - atrod ierakstus, kuros *Attribute key* pastāv un tā vērtība ir vienāda ar *Attribute value* (reģistrjutīgi). **Does not equal** - atrod ierakstus, kuros *Attribute key* pastāv un tā vērtība nav vienāda ar *Attribute value* (reģistrjutīgi). **Exists** - atrod ierakstus, kuros *Attribute key* pastāv (reģistrjutīgi).
Units *Update interval* *Custom intervals*	Skatiet parametrus, kas ir kopīgi visiem vienumiem.
Time shift	Ievadiet, cik tālu pagātnē (pirms pašreizējā laika) jāpabeidz laika diapazons, par kuru vienums veic vaicājumu. Tiek atbalstīts sekunžu skaits (piemēram, 1, 60, 300), vērtība ar laika sufiksu (piemēram, `30s`, `1m`, `2h`, `1d`) vai lietotāja makro. Katrs vaicājums pieprasa datus par noteiktu laika diapazonu. Parasti diapazons sākas vietā, kur beidzās iepriekšējā vaicājuma laika diapazons, un beidzas punktā, ko nosaka *Time shift*. Ja iepriekšējā vaicājuma laika diapazons beidzās vēlāk nekā pašreizējais laiks mīnus *Granularity*, pašreizējā vaicājuma laika diapazons sākas pašreizējā laika mīnus *Granularity* punktā, tādējādi pārklājoties ar iepriekšējā vaicājuma laika diapazonu. Ja nav iepriekšēja vaicājuma, no kura turpināt, diapazons sākas punktā, ko nosaka *Lookback limit*. Ja šī vienuma hosts tiek uzraudzīts ar citas entītijas palīdzību (piemēram, cita starpniekservera no starpniekserveru grupas), vienums var vairs nezināt, kad beidzās iepriekšējais vaicājums. Tas var izraisīt dublētas vērtības.

Parametrs	Apraksts
Lookback limit	Ievadiet, cik tālu pagātnē (pirms pašreizējā laika) vienums pieprasa datus, ja nav iepriekšēja vaicājuma, no kura turpināt (piemēram, vienuma pirmās palaišanas reizē), vai ja iepriekšējais vaicājums ir vecāks par *Update interval* (piemēram, tāpēc, ka Zabbix serveris nedarbojās). Tiek atbalstīts sekunžu skaits (piemēram, 1, 60, 300), vērtība ar laika sufiksu (piemēram, 30s, 1m, 2h, 1d) vai lietotāja makro. Ja telemetrijas datu avotā tiek glabāts liels datu apjoms (piemēram, vairāk nekā miljons ierakstu stundā), liels *Lookback limit* var ievērojami samazināt veiktspēju, kad vienums to izmanto.
Granularity	Ievadiet tā segmenta garumu, kurā tiek sadalīts viss vaicājuma laika diapazons. Tiek atbalstīts sekunžu skaits (piemēram, 1, 60, 300), vērtība ar laika sufiksu (piemēram, 30s, 1m, 2h, 1d) vai lietotāja makro.
Timeout *History* *Trends* *Value mapping* *Populates host inventory field* *Description* *Enabled*	Skatiet parametrus, kas ir kopīgi visiem vienumiem.

Atbalstītās telemetrijas datu kolonnas

Šajā sadaļā ir uzskaitītas kolonnas, kas tiek atbalstītas šādos vienuma parametros: *Columns*, *Aggregated columns* un *Conditions*. Tās ir kolonnas, ko ClickHouse izmanto telemetrijas datu glabāšanai saskaņā ar OpenTelemetry ClickHouse eksportētāja shēmu.

Atlasiet *Category*:

- Traces
- Metrics: Sum
- Metrics: Gauge
- Metrics: Histogram
- Metrics: Exponential histogram
- Logs

Izsekojumi

Kolonna	*Kolonnas*	*Apkopotās kolonnas*	*Nosacījumi*
Ilgums	✓	✓	–
Events.Attributes	–	–	✓
Events.Name	–	–	✓
ParentSpanId	✓	–	✓
ResourceAttributes	✓	–	✓
ScopeName	✓	–	✓
ScopeVersion	✓	–	✓
ServiceName	✓	–	✓
SpanAttributes	✓	–	✓
SpanId	✓	–	✓
SpanKind	✓	–	✓
SpanName	✓	–	✓
StatusCode	✓	–	✓
StatusMessage	✓	–	✓
Laikspiedogs	✓	✓	–
TraceId	✓	–	✓
TraceState	✓	–	✓

Metrikas: summa

Kolonna	*Kolonnas*	*Apkopotās kolonnas*	*Nosacījumi*
AggregationTemporality	✓	–	–
Attributes	✓	–	✓

Kolonna	*Kolonnas*	*Apkopotās kolonnas*	*Nosacījumi*
Exemplars.FilteredAttributes	–	–	✓
Flags	✓	–	–
IsMonotonic	✓	–	–
MetricDescription	✓	–	✓
MetricName	✓	–	✓
MetricUnit	✓	–	✓
ResourceAttributes	✓	–	✓
ResourceSchemaUrl	✓	–	✓
ScopeAttributes	✓	–	✓
ScopeDroppedAttrCount	✓	✓	–
ScopeName	✓	–	✓
ScopeSchemaUrl	✓	–	✓
ScopeVersion	✓	–	✓
ServiceName	✓	–	✓
StartTimeUnix	✓	✓	–
TimeUnix	✓	✓	–
Value	✓	✓	–

Metrikas: mērinstruments

Kolonna	*Kolonnas*	*Apkopotās kolonnas*	*Nosacījumi*
Atribūti	✓	–	✓
Exemplars.FilteredAttributes	–	–	✓
Karodziņi	✓	–	–
MetricDescription	✓	–	✓
MetricName	✓	–	✓
MetricUnit	✓	–	✓
ResourceAttributes	✓	–	✓
ResourceSchemaUrl	✓	–	✓
ScopeAttributes	✓	–	✓
ScopeDroppedAttrCount	✓	✓	–
ScopeName	✓	–	✓
ScopeSchemaUrl	✓	–	✓
ScopeVersion	✓	–	✓
ServiceName	✓	–	✓
StartTimeUnix	✓	✓	–
TimeUnix	✓	✓	–
Value	✓	✓	–

Metrikas: histogramma

Kolonna	*Kolonnas*	*Apkopotās kolonnas*	*Nosacījumi*
AggregationTemporality	✓	–	–
Attributes	✓	–	✓
Count	✓	✓	–
Exemplars.FilteredAttributes	–	–	✓
Flags	✓	–	–
Max	✓	✓	–
MetricDescription	✓	–	✓
MetricName	✓	–	✓
MetricUnit	✓	–	✓
Min	✓	✓	–
ResourceAttributes	✓	–	✓
ResourceSchemaUrl	✓	–	✓
ScopeAttributes	✓	–	✓
ScopeDroppedAttrCount	✓	✓	–
ScopeName	✓	–	✓
ScopeSchemaUrl	✓	–	✓

Kolonna	*Kolonnas*	*Apkopotās kolonnas*	*Nosacījumi*
ScopeVersion	✓	–	✓
ServiceName	✓	–	✓
StartTimeUnix	✓	✓	–
Sum	✓	✓	–
TimeUnix	✓	✓	–

Metrikas: eksponenciālā histogramma

Kolonna	*Kolonnas*	*Apkopotās kolonnas*	*Nosacījumi*
AggregationTemporality	✓	–	–
Attributes	✓	–	✓
Count	✓	✓	–
Exemplars.FilteredAttributes	–	–	✓
Flags	✓	–	–
Max	✓	✓	–
MetricDescription	✓	–	✓
MetricName	✓	–	✓
MetricUnit	✓	–	✓
Min	✓	✓	–
NegativeOffset	✓	✓	–
PositiveOffset	✓	✓	–
ResourceAttributes	✓	–	✓
ResourceSchemaUrl	✓	–	✓
Scale	✓	✓	–
ScopeAttributes	✓	–	✓
ScopeDroppedAttrCount	✓	✓	–
ScopeName	✓	–	✓
ScopeSchemaUrl	✓	–	✓
ScopeVersion	✓	–	✓
ServiceName	✓	–	✓
StartTimeUnix	✓	✓	–
Sum	✓	✓	–
TimeUnix	✓	✓	–
ZeroCount	✓	✓	–

Žurnāli

Kolonna	*Kolonnas*	*Apkopotās kolonnas*	*Nosacījumi*
Body	✓	–	✓
EventName	✓	–	✓
LogAttributes	✓	–	✓
ResourceAttributes	✓	–	✓
ResourceSchemaUrl	✓	–	✓
ScopeAttributes	✓	–	✓
ScopeName	✓	–	✓
ScopeSchemaUrl	✓	–	✓
ScopeVersion	✓	–	✓
SpanId	✓	–	✓
ServiceName	✓	–	✓
SeverityNumber	✓	✓	–
SeverityText	✓	–	✓
Timestamp	✓	✓	–
TraceFlags	✓	–	–
TraceId	✓	–	✓

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 glabāšana

Varat iestatīt, cik dienas tiks glabāta 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 nepārslogotu datubāzi ar lielu vēsturisko vērtību apjomu.

Tā vietā, lai glabātu garu vēsturi, varat ilgāk glabāt tendenču datus. Piemēram, varat glabāt vēsturi 14 dienas un tendences 5 gadus.

Labu priekšstatu par to, cik daudz vietas nepieciešams vēstures datiem salīdzinājumā ar tendenču datiem, var iegūt, skatot datubāzes izmēra lapu.

Pat glabājot 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 iespēju izmantot ielādējamo moduļu vēstures eksportēšanas funkcionalitāti.

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.

Lietotāja parametri

Pārskats

Dažkārt var būt nepieciešams palaist aģenta pārbaudi, kas nav iepriekš definēta Zabbix. Šeit noder lietotāja parametri.

Jūs varat uzrakstīt komandu, kas izgūst nepieciešamos datus, un iekļaut to lietotāja parametrā aģenta konfigurācijas failā ('UserParameter' konfigurācijas parametrā).

Lietotāja parametram ir šāda sintakse:

```
UserParameter=<key>,<command>
```

Kā redzams, lietotāja parametrs satur arī atslēgu. Atslēga būs nepieciešama, konfigurējot vienumu. Norādiet sev ērtu atslēgu, kuru būs viegli izmantot atsaucēm (tai jābūt unikālai viena hosta ietvaros).

Pārstartējiet aģentu vai izmantojiet aģenta izpildlaika vadības opciju, lai ielādētu jauno parametru, piemēram:

```
zabbix_agentd -R userparameter_reload
```

Pēc tam, konfigurējot vienumu, ievadiet atslēgu, lai norādītu uz komandu no lietotāja parametra, kuru vēlaties izpildīt.

Lietotāja parametri ir komandas, ko izpilda Zabbix aģents. Ņemiet vērā, ka pirms vienuma vērtības pirmapstrādes soļiem var tikt atgriezti dati līdz 16 MB apjomā.

UNIX operētājsistēmās kā komandrindas interpretētājs tiek izmantots **/bin/sh**. Lietotāja parametri ievēro aģenta pārbaudes noildzi; ja noildze tiek sasniegta, 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'.

Elastīgie lietotāja parametri

Elastīgie lietotāja parametri pieņem parametrus ar atslēgu. Tādējādi elastīgs lietotāja parametrs var kalpot par pamatu vairāku vienumu izveidei.

Elastīgajiem lietotāja parametriem ir šāda sintakse:

```
UserParameter=key[*],command
```

Parameter	Description
Key	Unikāla vienuma atslēga. [*] nosaka, ka šī atslēga pieņem parametrus iekavās. Parametri tiek norādīti, konfigurējot vienumu.

Parameter	Description
Command	Komanda, kas jāizpilda, lai noteiktu atslēgas vērtību. *Tikai elastīgajiem lietotāja parametriem*: Komandā var izmantot pozicionālās atsauces $1...$9, lai norādītu uz attiecīgo parametru vienuma atslēgā. Zabbix parsē vienuma atslēgas [ ] ietvertos parametrus un attiecīgi aizstāj komandā $1,...,$9. $0 tiks aizstāts ar sākotnējo komandu (pirms $0,...,$9 paplašināšanas), kas jāizpilda. Pozicionālās atsauces tiek interpretētas neatkarīgi no tā, vai tās ir ietvertas dubultpēdiņās (") vai vienpēdiņās ('). Lai izmantotu pozicionālās atsauces nemainītā veidā, norādiet dubultu dolāra zīmi - piemēram, awk '{print $$2}'. Šajā gadījumā $$2 izpildes laikā faktiski kļūs par $2.

Attention:
Pozicionālās atsauces ar $ zīmi meklē un aizstāj tikai Zabbix aģents elastīgajiem lietotāja parametriem. Vienkāršajiem lietotāja parametriem šāda atsauču apstrāde tiek izlaista, un tāpēc $ zīmes citēšana nav nepieciešama.

Attention:
Pēc noklusējuma lietotāja parametros nav atļautas noteiktas rakstzīmes. 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.

Zabbix aģentu paplašināšana

Šī pamācība sniedz soli pa solim instrukcijas, kā paplašināt Zabbix aģenta funkcionalitāti, izmantojot lietotāja parametru.

1. solis

Uzrakstiet skriptu vai komandrindu, lai iegūtu nepieciešamo parametru.

Piemēram, varam uzrakstīt šādu komandu, lai iegūtu kopējo MySQL serveris izpildīto vaicājumu skaitu:

```
mysqladmin -uroot status | cut -f4 -d":" | cut -f1 -d"S"
```

Izpildot šo komandu, tā atgriež kopējo SQL vaicājumu skaitu.

2. darbība

Pievienojiet komandu Zabbix aģenta konfigurācijas 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 aģentu ar karodziņu `-t` (ja tas tiek palaists ar root lietotāja tiesībām, ņemiet vērā, ka aģentam, kas palaists kā dēmons, var būt atšķirīgas atļaujas):

```
zabbix_agentd -t mysql.questions
```

3. solis

Pārlādējiet lietotāja parametrus no konfigurācijas faila, izpildot:

```
zabbix_agentd -R userparameter_reload
```

Varat arī aģentu restartēt runtime vadības komandas vietā.

Pārbaudiet parametru, izmantojot `zabbix_get` utilītu.

4. solis

Pievienojiet jaunam vienumam `Key=mysql.questions` uzraudzītajam hostam. Vienuma tipam jābūt vai nu *Zabbix Agent*, vai *Zabbix Agent (active)*.

Ņemiet vērā, ka atgriezto vērtību tipam jābūt pareizi iestatītam Zabbix serverī. Pretējā gadījumā Zabbix tās nepieņems.

Windows veiktspējas skaitītāji

Pārskats

Windows veiktspējas skaitītājus var efektīvi pārraudzīt, izmantojot atslēgu `perf_counter[]`.

Piemēram:

```
perf_counter["\Processor(0)\Interrupts/sec"]
```

vai

```
perf_counter["\Processor(0)\Interrupts/sec", 10]
```

Papildinformāciju par šīs atslēgas vai tās tikai angļu valodā pieejamā ekvivalenta `perf_counter_en` izmantošanu skatiet sadaļā Windows specifiskās vienumu atslēgas.

Lai iegūtu pilnu pārraudzībai 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 vairākiem instanču objektiem.

Skaitliskais attēlojums

Windows uztur objektu un veiktspējas skaitītāju nosaukumu skaitliskos attēlojumus (indeksus). Zabbix atbalsta šos skaitliskos attēlojumus kā parametrus `perf_counter`, `perf_counter_en` vienumu atslēgām un `PerfCounter`, `PerfCounterEn` konfigurācijas parametriem.

Tomēr nav ieteicams tos izmantot, ja vien nevarat garantēt, ka jūsu skaitliskie indeksi konkrētajos hosts atbilst pareizajām virknēm. Ja jāizveido pārnesami vienumi, kas darbojas dažādos hosts 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 skaitliskās atbilstības, palaidiet regedit un pēc tam atrodiet *Counter* sadaļā HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft 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 katras veiktspējas skaitītāja virknes daļas atbilstošos numurus, piemēram, \System\% Processor Time:

```
System → 2
% Processor Time → 6
```

Pēc tam varat izmantot šos numurus, lai ceļu attēlotu skaitliski:

```
\2\6
```

Veiktspējas skaitītāja parametri

Varat izmantot PerfCounter parametrus Windows veiktspējas skaitītāju uzraudzībai.

Piemēram, varat tos pievienot Zabbix aģenta konfigurācijas failam:

```
PerfCounter=UserPerfCounter1,"\Memory\Page Reads/sec",30
or
PerfCounter=UserPerfCounter2,"\4\24",30
```

Pēc šādu parametru pievienošanas varat vienkārši izmantot UserPerfCounter1 vai UserPerfCounter2 kā atslēgas attiecīgo vienumu izveidei.

Pēc konfigurācijas faila izmaiņu veikšanas neaizmirstiet restartēt Zabbix aģentu.

Masveida atjaunināšana

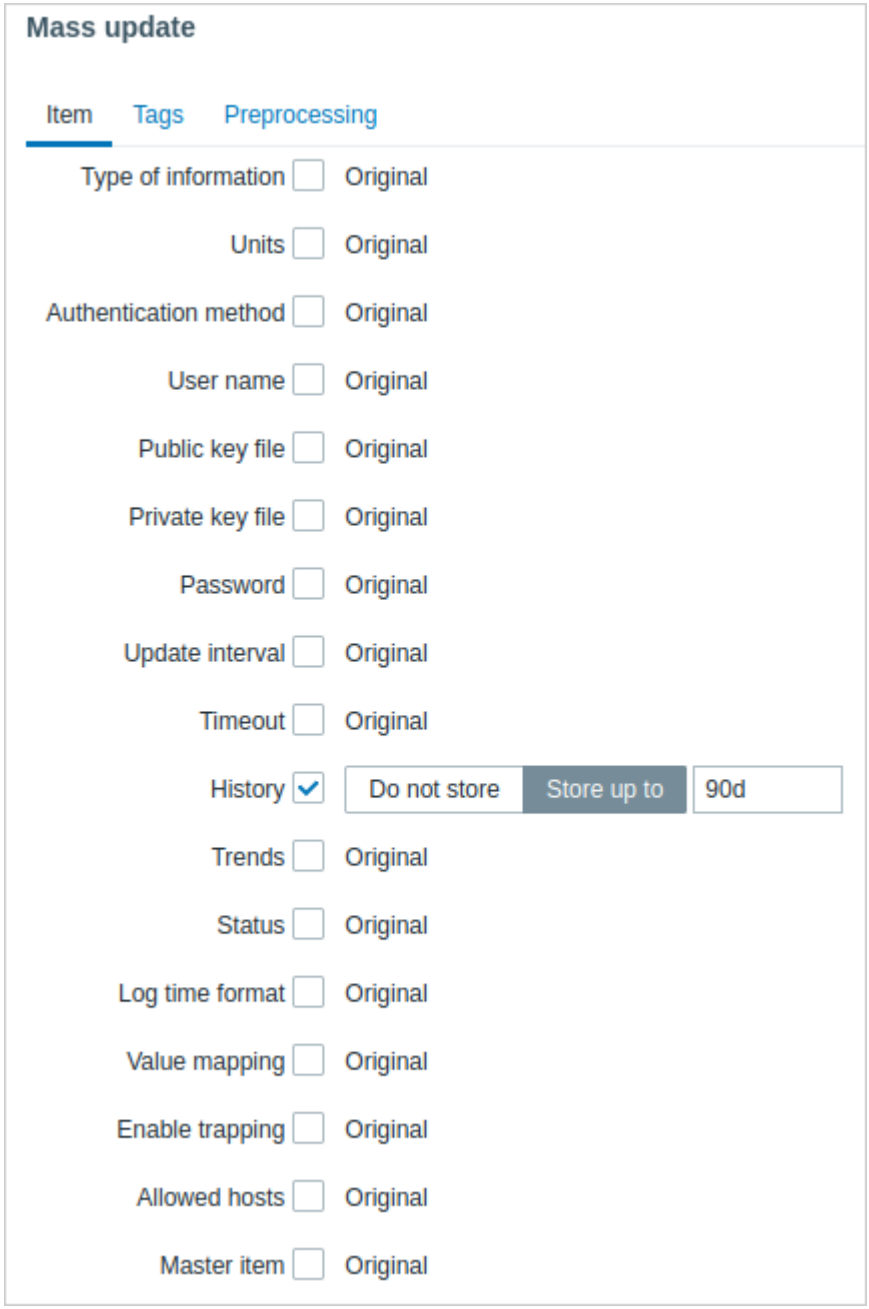
Pārskats

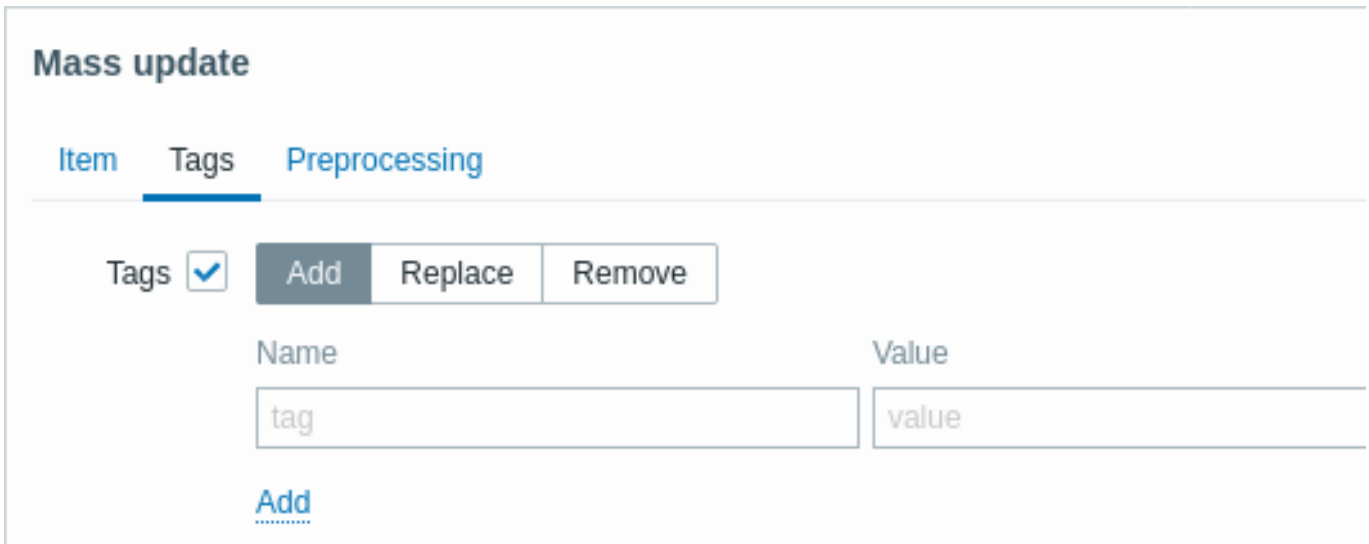
Dažkārt var būt nepieciešams vienlaikus mainīt kādu atribūtu vairākiem vienumiem. Tā vietā, lai atvērtu katru atsevišķo vienumu rediģēšanai, šim nolūkam varat izmantot masveida atjaunināšanas funkciju.

Masveida atjaunināšana

Lai masveidā atjauninātu dažus vienumus, rīkojieties, kā norādīts tālāk:

- Sarakstā atzīmējiet atjaunināmo vienumu izvēles rūtiņas
- Zem saraksta noklikšķiniet uz *Mass update*
- Atveriet cilni ar nepieciešamajiem atribūtiem (*Item*, *Tags* vai *Preprocessing*)
- Atzīmējiet atjaunināmo atribūtu izvēles rūtiņas
- Ievadiet jaunas atribūtu vērtības



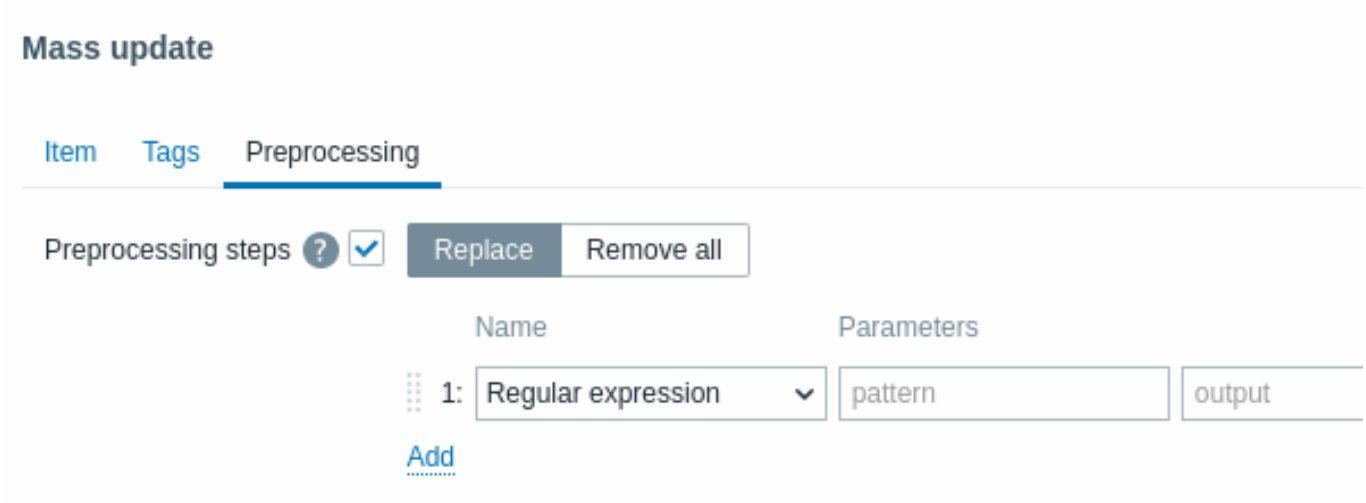


Opcija *Tags* ļauj:

- *Add* - pievienot vienumiem norādītās birkas (birkas ar tādu pašu nosaukumu, bet atšķirīgām vērtībām netiek uzskatītas par dublikātiem un tās var pievienot vienam un tam pašam vienumam).
- *Replace* - noņemt norādītās birkas un pievienot birkas ar jaunām vērtībām

- *Remove* - noņemt norādītās birkas no vienumiem

Birkās tiek atbalstīti lietotāja makro, `{INVENTORY.*}` makro, `{HOST.HOST}`, `{HOST.NAME}`, `{HOST.CONN}`, `{HOST.DNS}`, `{HOST.IP}`, `{HOST.PORT}` un `{HOST.ID}` makro.



Atzīmējiet izvēles rūtiņu blakus *Preprocessing steps*, lai aktivizētu priekšapstrādes darbību masveida atjaunināšanu.

Preprocessing masveida atjaunināšana ļauj:

- *Replace* - aizstāt esošās vienumu priekšapstrādes darbības ar tālāk norādītajām priekšapstrādes darbībām
- *Remove all* - noņemt visas vienumu priekšapstrādes darbības

Kad esat pabeidzis, noklikšķiniet uz *Update*.

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*.

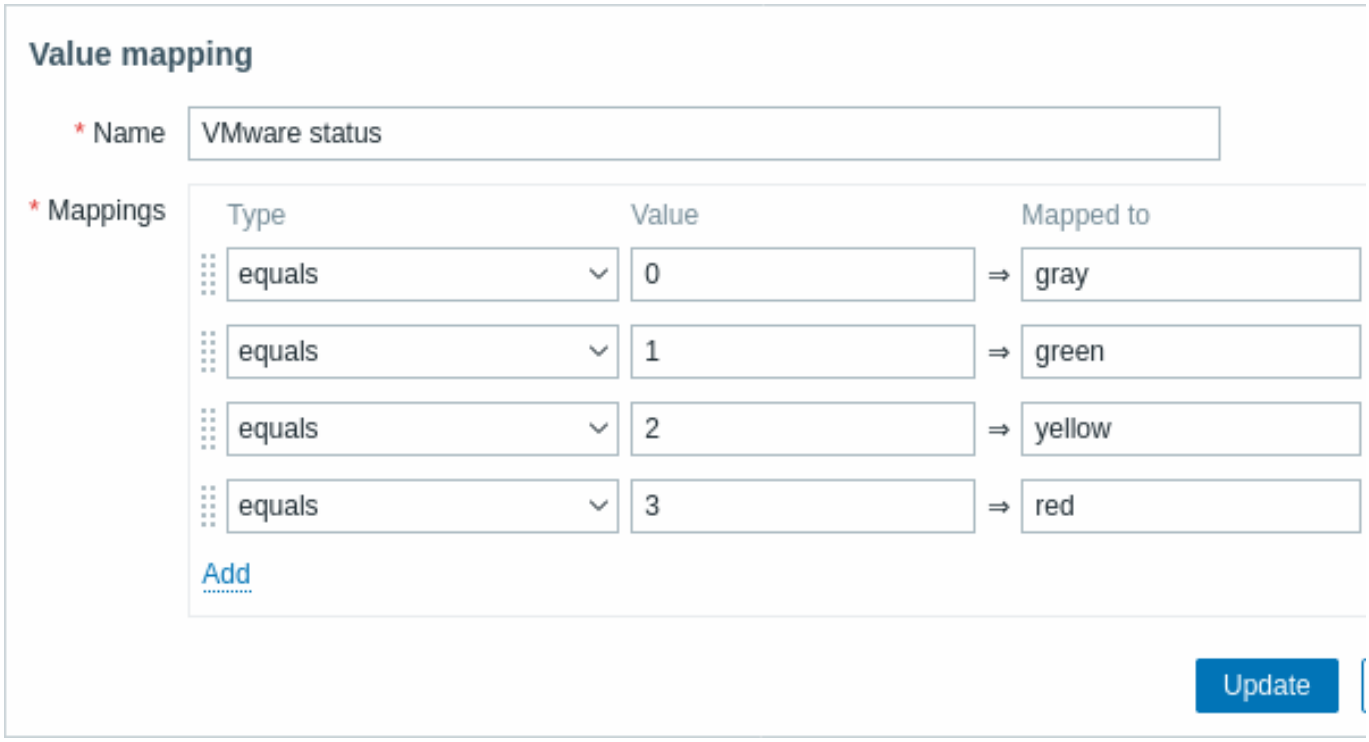
Vērtību kartēšana tiek eksportēta/importēta kopā ar attiecīgo veidni vai hostu. To var arī masveidā atjaunināt, izmantojot hosta un veidnes masveida atjaunināšanas formas.

Konfigurēšana

Lai konfigurētu vērtību kartēšanu, veiciet tālāk norādītās darbības:

1. Atveriet hosta vai veidnes konfigurācijas veidlapu.

2. Cilnē *Value mapping* noklikšķiniet uz *Add*, lai pievienotu jaunu vērtību kartējumu, vai noklikšķiniet uz esoša kartējuma nosaukuma, lai to rediģētu.



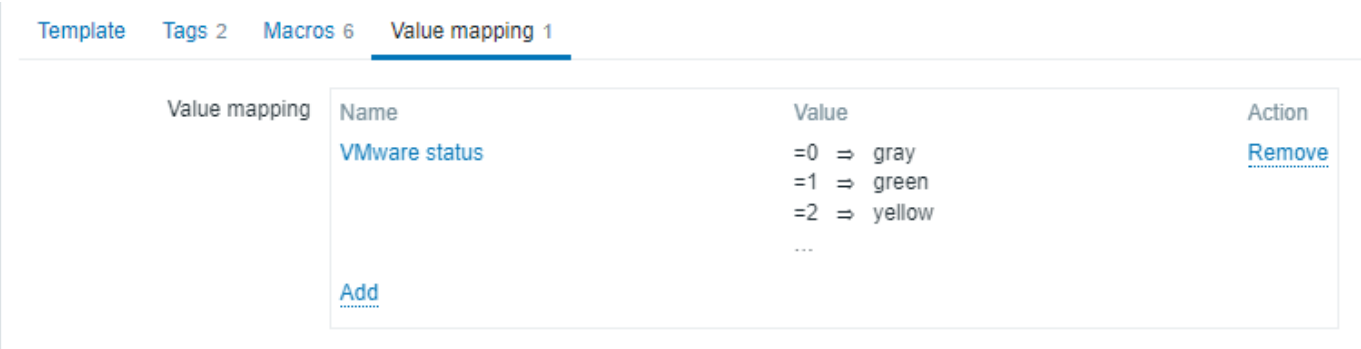
Vērtību kartējuma parametri:

Parameter	Description
Name	Unikāls vērtību kartējumu kopas nosaukums.
Mappings	Atsevišķi noteikumi skaitlisko vai virkņu vērtību kartēšanai virkņu attēlojumos. Kartēšana tiek lietota noteikumu secībā; šo secību var mainīt, velkot noteikumus.
Type	Kartēšanas tips: **equals** - tiks kartētas vienādas vērtības; **is greater than or equals** - tiks kartētas vienādas vai lielākas vērtības; **is less than or equals** - tiks kartētas vienādas vai mazākas vērtības; **in range** - tiks kartētas diapazonā ietilpstošās vērtības; 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); **default** - tiks kartētas visas atlikušās vērtības, izņemot tās, kurām ir konkrēti kartējumi. Diapazonu kartēšanai tiek atbalstīti tikai skaitliskie vērtību tipi (*is greater than or equals*, *is less than or equals*, *in range*).
Value	Ienākošā vērtība (atkarībā no kartēšanas tipa var saturēt diapazonu vai regulāro izteiksmi).

Parameter	Description
Mapped to	Ienākošās vērtības virknes attēlojums (līdz 64 rakstzīmēm).

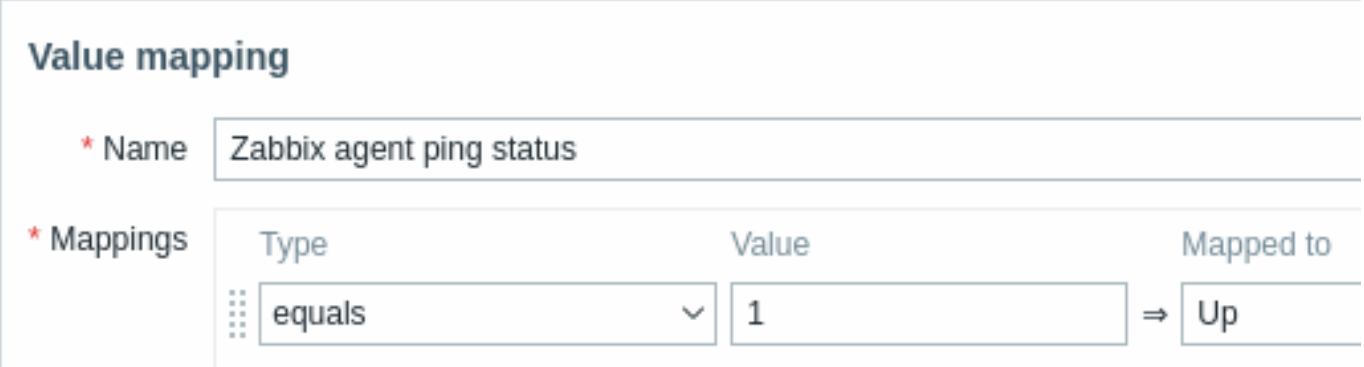
Visi obligātie ievades lauki ir atzīmēti ar sarkanu zvaigznīti.

Apskatot vērtību kartējumu sarakstā, ir redzami tikai pirmie trīs kartējumi; trīs punkti norāda, ka ir vēl citi kartējumi.



Vērtību kartēšanas piemērs

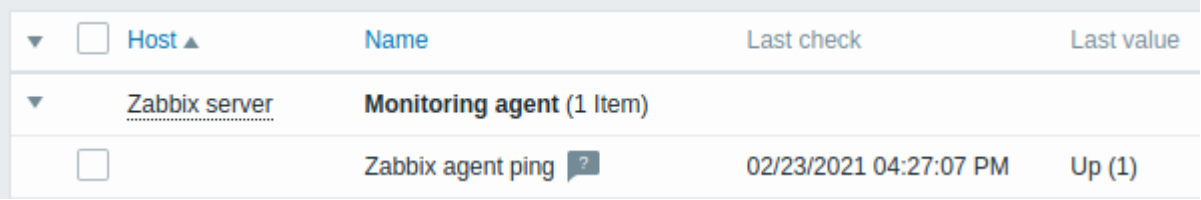
Viens no iepriekš definētajiem aģenta vienumiem *Zabbix agent ping* izmanto veidnes līmeņa vērtību kartējumu "Zabbix agent ping status", lai attēlotu savas vērtības.



Vienuma konfigurācijas veidlapā atsauci uz šo vērtību kartējumu var atrast laukā *Value mapping*:

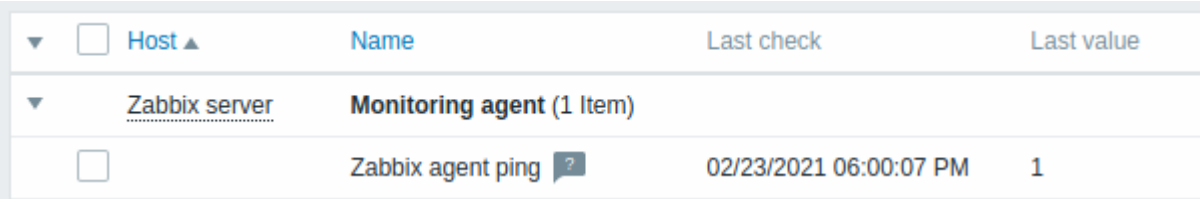


Šis kartējums tiek izmantots sadaļā *Monitoring > Latest data*, lai attēlotu "Up" (iekavās norādot neapstrādāto vērtību).



Note:
Sadaļā *Latest data* attēlotās vērtības tiek saīsinātas līdz 20 simboliem. Ja tiek izmantots vērtību kartējums, saīsināšana netiek piemērota kartētajai vērtībai, bet tikai neapstrādātajai vērtībai (kas tiek attēlota iekavās).

Ja nebūtu iepriekš definēta vērtību kartējuma, jūs redzētu tikai "1", ko varētu būt grūti saprast.



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 lasītu rindu, dodieties uz *Administration* > *Queue* > *Queue overview*.

Queue overview

Items	5 seconds	10 seconds	30 seconds	1 minute	5 minutes	More than 10 minutes
Zabbix agent	1	11	1	0	0	0
Zabbix agent (active)	0	0	0	0	0	0
Simple check	0	0	0	0	0	0
SNMPv1 agent	0	0	0	0	0	0
SNMPv2 agent	0	0	0	0	0	0
SNMPv3 agent	0	0	0	0	0	0
Zabbix internal	0	0	0	0	0	0
Zabbix aggregate	0	0	0	0	0	0
External check	0	0	0	0	0	0
Database monitor	0	0	0	0	0	0
HTTP agent	0	0	0	0	0	0

Attēls kopumā izskatās "ok", tāpēc varam pieņemt, ka serveris darbojas labi.

Rindā redzami daži vienumi, kas gaida līdz 30 sekundēm. Būtu noderīgi zināt, kuri tie ir.

Lai to noskaidrotu, virsraksta nolaižamajā sarakstā atlasiet *Queue details*. Tagad varat skatīt šo aizkavēto vienumu sarakstu.

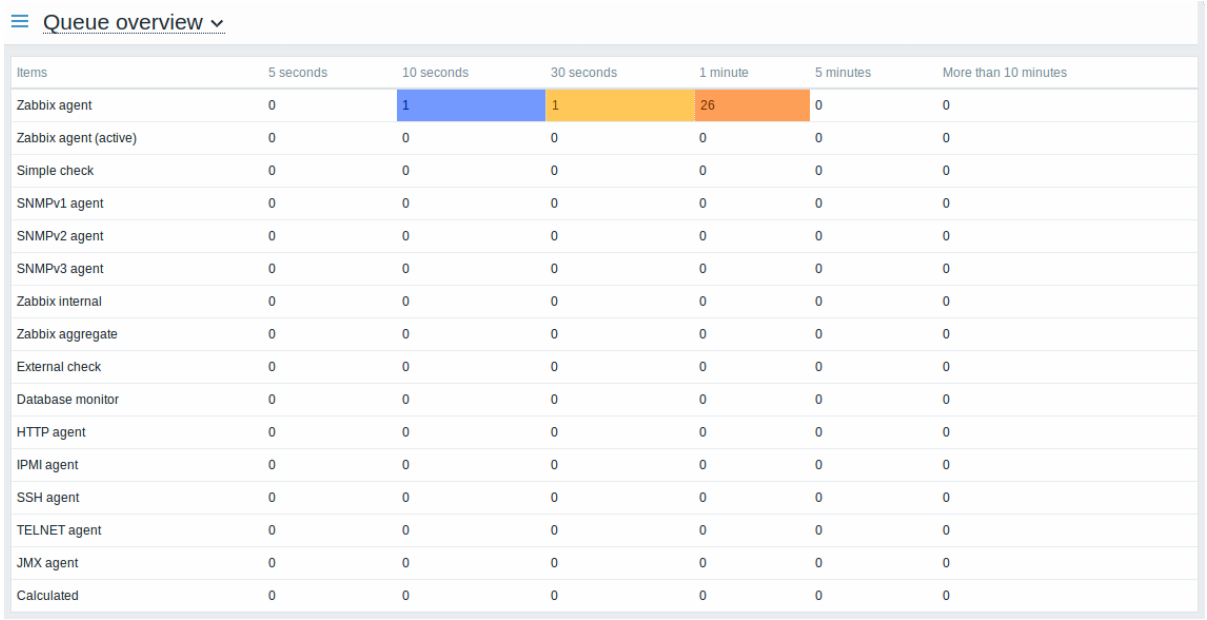
Queue details

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

Izmantojot šo detalizēto informāciju, iespējams noskaidrot, kāpēc šie vienumi varētu būt aizkavēti.

Ja aizkavēts ir viens vai divi vienumi, tas, iespējams, nav iemesls satraukumam. Tie var tikt atjaunināti jau pēc sekundes. Tomēr, ja redzat, ka daudzi vienumi tiek aizkavēti pārāk ilgi, iespējams, pastāv nopietnāka problēma.

Skatiet arī: Laika joslu saskaņošana, izmantojot plānošanas intervālus.



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

Lai uzraudzītu rindas stāvokli Zabbix, var izmantot īpašu iekšējo vienumu `zabbix[queue,<from>,<to>]`. Tas atgriezīs vienumu skaitu, kuru apstrāde ir aizkavējusies par iestatīto laika periodu. Plašāku informāciju skatiet sadaļā Iekšējie vienumi.

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 konfigurācijā var norādīt neobligātu parametru `ValueCacheSize`.

Vērtību kešatmiņas uzraudzībai tiek atbalstīti divi iekšējie vienumi: `zabbix[vcache,buffer,<mode>]` un `zabbix[vcache,cache,<para`
Plašāku informāciju skatiet sadaļā par iekšējiem vienumiem.

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

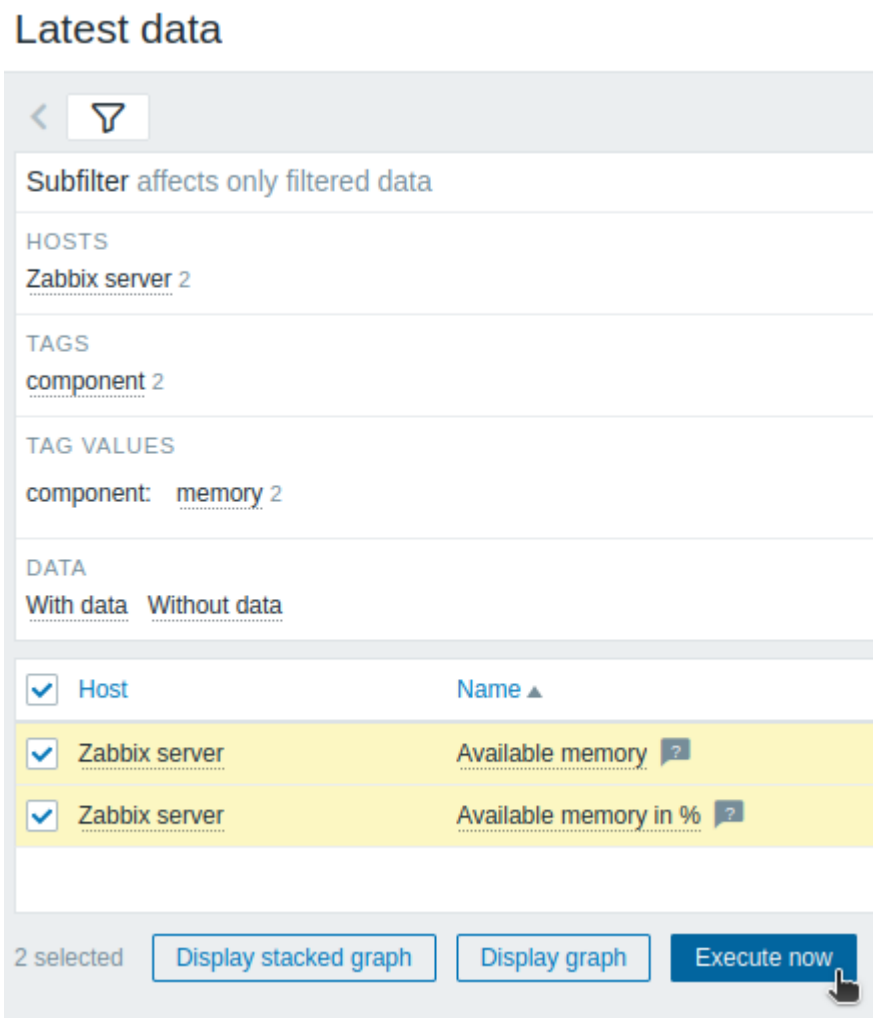
Pasīvos pārbaudes var izpildīt no:

- *Uzraudzība* > *Jaunākie dati*
- *Datu vākšana* > *Hosts* > *Vienumi* vai *Atklāšana*
- Vienumu izvēlnē
- Rediģējot vienumus vai zemā līmeņa atklāšanas noteikumus (to konfigurācijas formās)

Attention:
Pārbaudei ir jāatrodas konfigurācijas kešatmiņā, lai to varētu izpildīt; skatiet arī CacheUpdateFrequency. Kešatmiņa pirms pārbaudes izpildes **netiek atjaunināta**, tāpēc nesen veiktās izmaiņas vienuma vai zemā līmeņa atklāšanas noteikuma konfigurācijā netiks ņemtas vērā. Lai testētu vienumus vai LLD noteikumus, kas tiek veidoti vai tikko ir izveidoti, tā vietā varat izmantot opciju Test.

Piemēram, lai izpildītu pasīvu pārbaudi sadaļā *Uzraudzība* > *Jaunākie dati*:

1. Atlasiet vienumus; pārbaudi var izpildīt vairākiem vienumiem vienlaikus.
2. Noklikšķiniet uz *Izpildīt tagad*.



Note:
Sadaļā *Jaunākie dati* lietotāji var *Izpildīt tagad* vienumus uz hosts, kuriem viņiem ir *lasīšanas-rakstīšanas* atļauja, vai arī *lasīšanas* atļauja un viņu lomai ir iespējota darbība *Izsaukt "Izpildīt tagad" tikai lasāmiem hosts* darbība.

Aģenta pārbaužu ierobežošana

Pārskats

Jūs varat kontrolēt, kurus vienumu atslēgu nosaukumus Zabbix aģents vai aģents 2 drīkst vai nedrīkst izmantot, izpildot vienumu pārbaudes, attālinātās komandas vai skriptus.

Lai to izdarītu, izmantojiet šos aģenta konfigurācijas parametrus, lai definētu atļaušanas/aizliegšanas noteikumus:

- `AllowKey=<pattern>`
- `DenyKey=<pattern>`
- `AllowKeyRegexp=<pattern>`
- `DenyKeyRegexp=<pattern>`

`<pattern>` jāietver viena vienuma atslēga.
 Parametriem `AllowKey` un `DenyKey` `<pattern>` atbalsta aizstājējzīmes (`*`). Aizstājējzīme atbilst jebkuram simbolu skaitam savā pozīcijā, un to var izmantot, lai atbilstinātu vienumu atslēgas vai parametrus (piem., `vfs.file.*[*]`).
 Parametriem `AllowKeyRegexp` un `DenyKeyRegexp` `<pattern>` atbalsta regulārās izteiksmes. Regulārās izteiksmes paraugu var izmantot, lai atbilstinātu vienumu atslēgas un to parametru virknes. Skatiet arī pareizas ekrānēšanas piemērus.
 Lūdzu, ņemiet vērā, ka Zabbix aģents un Zabbix aģents 2 var atbalstīt atšķirīgu regulāro izteiksmju sintaksi.

Attention:
Lai uzlabotu drošību, ieteicams `AllowKey` un `DenyKey` vietā izmantot precīzas vienumu atslēgas, nevis aizstājējzīmes. Tāpat ieteicams `AllowKeyRegexp` un `DenyKeyRegexp` izmantot pēc iespējas specifiskākus regulāro izteiksmju paraugus. Sīkāku informāciju skatiet sadaļā Atļaušanas/aizliegšanas noteikumu aizsardzība.

Atšķirībā no citiem aģenta konfigurācijas parametriem, jūs varat norādīt neierobežotu skaitu AllowKey, DenyKey, AllowKeyRegexp vai DenyKeyRegexp parametru.

Svarīgas piezīmes

- Visi system.run vienumi pēc noklusējuma ir atspējoti (pat ja DenyKey un DenyKeyRegexp ir tukši), it kā būtu iestatīts DenyKey=system.run[*] vai DenyKeyRegexp=^system\.run\[.*\]$ kā pēdējais noteikums. Tāpēc varat atļaut konkrētus system.run vienumus, nepārprotami nenoraidot citus system.run vienumus.
- Ja iespējams, izmantojiet AllowKey/AllowKeyRegexp, lai atļautu tikai nepieciešamos vienumus un noraidītu visu pārējo. Dažus atslēgu nosaukumus var izmantot ļaunprātīgi, lai, izmantojot ceļa pārlūkošanu, nolasītu neparedzētus failus (piem., vfs.file.contents["../../../../etc/passwd"]), un jaunās Zabbix aģents versijas var ieviest atslēgas, kuras jūsu DenyKey/DenyKeyRegexp noteikumi nesedz.
- AllowKey, DenyKey, AllowKeyRegexp un DenyKeyRegexp konfigurācija neietekmē HostnameItem, HostMetadataItem vai HostInterfaceItem aģents parametrus.
- Noraidītie vienumi kļūst neatbalstīti bez jebkādām norādēm vai kļūdas ziņojumiem; piemēram:
  - Zabbix aģents komandrindas parametrs --print (-p) nerādīs noraidīto vienumu atslēgas.
  - Zabbix aģents komandrindas parametrs --test (-t) noraidītajām vienumu atslēgām atgriezīs "Unsupported item key.".
  - Kad reģistrēšana ir aktivizēta (LogRemoteCommands=1), Zabbix aģents žurnālfailā netiks reģistrētas noraidītās attālās komandas.

Atļaušanas/noraidīšanas noteikumu secība

Jūs varat norādīt neierobežotu skaitu AllowKey, DenyKey, AllowKeyRegexp vai DenyKeyRegexp noteikumu, lai gan to secībai ir nozīme.

- Noteikumi tiek izvērtēti pa vienam, no augšas uz leju.
- Kad vienuma atslēga atbilst noteikumam, tā tiek vai nu atļauta, vai noraidīta, 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 paraugs neatbilst, aģents pāriet uz nākamo
AllowKey=vfs.file.contents[/etc/passwd]     # Vienuma atslēgas paraugs atbilst; aģents atļauj vienuma pārb
DenyKey=vfs.file.*[*]                       # Aģents ignorē noteikumu, jo izvērtēšana jau ir pārtraukta.
```

Šāda noteikumu secība noraidīs vienuma pārbaudi:

```
DenyKey=vfs.file.*[*]                       # Vienuma atslēgas paraugs atbilst; aģents noraida vienuma pār
AllowKey=vfs.file.contents[/etc/passwd]     # Aģents ignorē noteikumu, jo izvērtēšana jau ir pārtraukta.
AllowKey=vfs.file.contents[/tmp/app.log]    # Aģents ignorē noteikumu, jo izvērtēšana jau ir pārtraukta.
```

Piemēri

Tālāk redzamie piemēri parāda biežāk sastopamos konfigurācijas modeļus AllowKey, DenyKey, AllowKeyRegexp un DenyKeyRegexp.

Konkrētu pārbažu un komandu atļaušana

Atļaujiet tikai divas vfs.file vienums pārbaudes un divas system.run komandas:

izmantojot AllowKey un DenyKey:

```
AllowKey=vfs.file.contents[/tmp/app.log]
AllowKey=vfs.file.size[/tmp/app.log]
AllowKey=system.run[/usr/bin/uptime]
AllowKey=system.run[/usr/bin/df -h /]
DenyKey=vfs.file.*[*]
```

izmantojot AllowKeyRegexp un DenyKeyRegexp:

```
AllowKeyRegexp=^vfs\.file\.(contents|size)\[/tmp/app\.log\]$
AllowKeyRegexp=^system\.run\[/usr/bin/(uptime|df -h /)\]$
DenyKeyRegexp=^vfs\.file\..*\[.*\]$
```

Note:
Iestatījums DenyKey=system.run[*] vai DenyKeyRegexp=^system\.run\[.*\]$ nav nepieciešams, jo visas pārējās system.run komandas pēc noklusējuma tiek liegtas.

Skriptu atļaušana

Atļaujiet Zabbix aģentam izpildīt skriptus uz hosts, izmantojot visas pieejamās metodes:

- Globālie skripti, kurus var izpildīt lietotāja saskarnē vai caur 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,*]
```

vai

```
AllowKeyRegexp=^system\.run\[myscript\.sh,.*\]$
```

Lai kontrolētu `wait/nowait` parametru, jums jāiestata atšķirīgs noteikums. Piemēram, varat atļaut tikai `system.run[myscript.sh,wait]` vienumus, tādējādi izslēdzot citas metodes:

```
AllowKey=system.run[myscript.sh,wait]
```

vai

```
AllowKeyRegexp=^system\.run\[myscript\.sh,wait\]$
```

Drošības nodrošināšana allow/deny noteikumiem

Šie piemēri parāda, kā nodrošināt pārlieku atļaujošus `AllowKey/AllowKeyRegexp` vai `DenyKey/DenyKeyRegexp` noteikumus.

Apsveriet šādus noteikumus:

izmantojot `AllowKey` un `DenyKey`:

```
AllowKey=system.run["C:\Program^ Files\Zabbix^ Agent^ 2\scripts\test.bat*"]
DenyKey=vfs.file.*
DenyKey=system.cpu.load[*]
```

izmantojot `AllowKeyRegexp` un `DenyKeyRegexp`:

```
AllowKeyRegexp=^system\.run\["C:\\Program\sFiles\\Zabbix\sAgent\s2\\scripts\\test\.bat.*"\]$
DenyKeyRegexp=^vfs\.file\..*$
DenyKeyRegexp=^system\.cpu\.load\[.*\]$
```

Note:
Operētājsistēmā Windows ceļā atstarpes ir jāekranē, izmantojot caret (^) `AllowKey` vai `DenyKey` gadījumā un `\s` `AllowKeyRegexp` vai `DenyKeyRegexp` gadījumā.

Šie noteikumi satur aizstājējzīmi (*) (vai `.*` regulāro izteiksmju bāzētiem paraugiem), ko var izmantot neatbilstoši:

- `test.bat` skriptu var izpildīt ar jebkādiem argumentiem, tostarp ar neparedzētiem.
- `vfs.file.*` (`^vfs\.file\..*$`) paraugs atbilst vienumu atslēgām gan ar parametriem, gan bez tiem; tomēr visiem `vfs.file` vienumiem ir nepieciešami parametri.
- `system.cpu.load[*]` (`^system\.cpu\.load\[.*\]$`) paraugs atbilst tikai vienumu atslēgām ar parametriem; tomēr `system.cpu.load` vienumiem parametri nav nepieciešami.

Lai šos noteikumus padarītu drošus, skaidri atļaujiet `test.bat` izpildi tikai ar konkrētiem argumentiem un aizliedziet pareizus vienumu atslēgu paraugus; piemēram:

izmantojot `AllowKey` un `DenyKey`:

```
AllowKey=system.run["C:\Program^ Files\Zabbix^ Agent^ 2\scripts\test.bat status"]
AllowKey=system.run["C:\Program^ Files\Zabbix^ Agent^ 2\scripts\test.bat version"]
DenyKey=vfs.file.*[*]
DenyKey=system.cpu.load
DenyKey=system.cpu.load[*]
```

izmantojot `AllowKeyRegexp` un `DenyKeyRegexp`:

```
AllowKeyRegexp=^system\.run\["C:\\Program\sFiles\\Zabbix\sAgent\s2\\scripts\\test\.bat (status|version)"\]
DenyKeyRegexp=^vfs\.file\..+\[.*\]$
DenyKeyRegexp=^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]
```

Rakstu paraugi

Tālāk redzamajās tabulās ir parādīts, kā tiek salīdzināti vienuma atslēgu paraugi:

- Atslēga atbilst paraugam tikai tad, ja tā atbilst **visiem** nosacījumiem kolonnā *Matches*.
- Parametriem pilnībā jāatrodas kvadrātiekavās (piem., `vfs.file.contents[*` un `vfs.file.contents*utf8]` ir nederīgi paraugi).
- Piemēri ilustrē tikai paraugu salīdzināšanu; faktiskajiem vienumiem ir nepieciešami parametri.

Attiecībā uz `AllowKey` un `DenyKey`:

Pattern	Matches	Examples
*	Jebkura atslēga ar parametriem vai bez tiem	
vfs.file.*	Atslēga sākas ar vfs.file. Ar parametriem vai bez tiem	Matches: vfs.file.size vfs.file.contents vfs.file.contents[] vfs.file.size[/var/log/app.log]
vfs.*.contents	Atslēga sākas ar vfs. Atslēga beidzas ar .contents Bez parametriem	Matches: vfs..contents vfs.mount.point.file.contents Does not match: vfs.contents vfs.file.contents[]
vfs.file.*[*]	Atslēga sākas ar vfs.file. Jebkuri vai tukši parametri	Matches vfs.file.get.custom[] vfs.file.size[/var/log/app.log, utf8] Does not match: vfs.file.get.custom
vfs.file.contents	Atslēga ir vfs.file.contents Bez parametriem	Matches: vfs.file.contents Does not match: vfs.file.contents[/etc/passwd]
vfs.file.contents[]	Atslēga ir vfs.file.contents[] Tukši parametri	Matches: vfs.file.contents[] Does not match: vfs.file.contents
vfs.file.contents[*]	Atslēga ir vfs.file.contents Jebkuri vai tukši parametri	Matches: vfs.file.contents[/path/to/file] Does not match: vfs.file.contents

Pattern	Matches	Examples
vfs.file.contents[/etc/passwd,]	Atslēga ir vfs.file.contents Pirmais parametrs ir /etc/passwd Jebkurš vai tukšs otrais parametrs	Matches: vfs.file.contents[/etc/passwd,] vfs.file.contents[/etc/passwd,utf8] Does not match: vfs.file.contents[] vfs.file.contents[/etc/passwd]
vfs.file.contents[*passwd*]	Atslēga ir vfs.file.contents Jebkuri parametri, vismaz viens satur passwd	Matches: vfs.file.contents[/etc/passwd] 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	Matches: vfs.file.contents[/etc/passwd,] vfs.file.contents[/etc/passwd,utf8] Does not match: vfs.file.contents[/etc/passwd] vfs.file.contents[/tmp/test]
vfs.file.contents[/etc/passwd,utf8]	Atslēga ir vfs.file.contents Pirmais parametrs ir /etc/passwd Otrais parametrs ir utf8	Matches: vfs.file.contents[/etc/passwd,utf8] Does not match: vfs.file.contents[/etc/passwd,] vfs.file.contents[/etc/passwd,utf16]

Attiecībā uz AllowKeyRegexp un DenyKeyRegexp:

Pattern	Matches	Examples
^.*$	Jebkura atslēga ar parametriem vai bez tiem	
^vfs\.file\..*$	Atslēga sākas ar vfs.file. Ar parametriem vai bez tiem	Matches: vfs.file.size vfs.file.contents vfs.file.contents[] vfs.file.size[/var/log/app.log]
^vfs\..*\.contents$	Atslēga sākas ar vfs. Atslēga beidzas ar .contents Bez parametriem	Matches: vfs..contents vfs.mount.point.file.contents Does not match: vfs.contents vfs.file.contents[]
^vfs\.file\..*\[.*\]$	Atslēga sākas ar vfs.file. Jebkuri vai tukši parametri	Matches vfs.file.get.custom[] vfs.file.size[/var/log/app.log,utf8] Does not match: vfs.file.get.custom
^vfs\.file\.contents$	Atslēga ir vfs.file.contents Bez parametriem	Matches: vfs.file.contents Does not match: vfs.file.contents[/etc/passwd]
^vfs\.file\.contents\[\]$	Atslēga ir vfs.file.contents[] Tukši parametri	Matches: vfs.file.contents[] Does not match: vfs.file.contents

Pattern	Matches	Examples

`^vfs\.file\.contents\[.*\]$`	Atslēga ir `vfs.file.contents`
	Jebkuri vai tukši parametri
		Matches:
		`vfs.file.contents[/path/to/file]`
		
		Does not match:
		`vfs.file.contents`

`^vfs\.file\.contents\[/etc/passwd,.*\]$`	Atslēga ir `vfs.file.contents`
	Pirmais parametrs ir `/etc/passwd`
	Jebkurš vai tukšs otrais parametrs
		Matches:
		`vfs.file.contents[/etc/passwd,]`
		`vfs.file.contents[/etc/passwd,utf8]`
		
		Does not match:
		`vfs.file.contents[]`
		`vfs.file.contents[/etc/passwd]`

`^vfs\.file\.contents\[.*passwd.*\]$`	Atslēga ir `vfs.file.contents`
	Jebkuri parametri, vismaz viens satur `passwd`
		Matches:
		`vfs.file.contents[/etc/passwd]`
		`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
		Matches:
		`vfs.file.contents[/etc/passwd,]`
		`vfs.file.contents[/etc/passwd,utf8]`
		
		Does not match:
		`vfs.file.contents[/etc/passwd]`
		`vfs.file.contents[/tmp/test]`

`^vfs\.file\.contents\[/etc/passwd,(utf8|windows-1252)\]$`	Atslēga ir `vfs.file.contents`
	Pirmais parametrs ir `/etc/passwd`
	Otrais parametrs ir `utf8` vai `windows-1252`
		Matches:
		`vfs.file.contents[/etc/passwd,utf8]`
		`vfs.file.contents[/etc/passwd,windows-1252]`
		
		Does not match:
		`vfs.file.contents[/etc/passwd,]`
		`vfs.file.contents[/etc/passwd,utf16]`

Problēmu noteikšana ar trigeriem

Problēmu noteikšana Zabbix 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 apkopo datus.
2. Definējiet trigeri šim vienumam. Bez trigera Zabbix tikai apkopo datus, bet nekad uz tiem nereaģē.
3. Pārraugiet konstatētās problēmas:
   - *Monitoring* > *Problems*
   - *Problems* logrīkā
4. Saņemiet paziņojumus, izpildiet attālinātās komandas (ja tās ir definētas).

Ņemiet vērā, ka oficiālajās Zabbix veidnēs ir iekļauti iepriekš definēti trigeri.
Skatiet rokasgrāmatu par veidnes lietošanu, lai monitorētu 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.

Trigeru piemēri

Šī lapa ir trigeru piemēru kolekcija.

Piemēri ir sakārtoti pēc lietošanas gadījuma:

- Nepieejams hosts
- Nepieejams starpniekserveris
- Nepieejams SMTP klasteris
- Nepieejams hosts
- Negaidīta pārstartēšana
- Izmaiņas svarīgos failos
- Izmaiņas DNS vaicājumā
- Nesakrītoša programmatūra dažādos hosts
- Nesinhronizēti pulksteņi
- Novecojis aģents
- Svārstīgs interfeiss
- Liels ienākošais trafiks
- Maz vietas diskā
- Maz vietas diskā (dinamisks slieksnis)
- Liela CPU slodze
- CPU slodzes salīdzinājums
- Ilgtermiņa CPU slodzes 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

Noteikt izmaiņas DNS vaicājumā

```
last(/Zabbix server/net.dns.record[192.0.2.1,{$WEBSITE_NAME},{$DNS_RESOURCE_RECORD_TYPE},2,1])<>"{$WEBSITE
```

Ņemiet vērā pēdiņas ap otro operandu.

Šis trigeris tiek aktivizēts, 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[192.0.2.1,{$WEBSITE_NAME},{$DNS_RESOURCE_RECORD_TYPE},2,1], ar definētām makrokomandām:

```
{$WEBSITE_NAME} = example.com
{$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

Nesinhronizētu pulksteņu noteikšana

```
fuzzytime(/host/system.localtime,10s)=0
```

Trigeris nostrādā, ja klienta vietējais laiks un Zabbix servera laiks atšķiras par vairāk nekā 10 sekundēm.
 Funkcija: fuzzytime
 Datu vienums: system.localtime

Ņemiet vērā, ka system.localtime Zabbix aģentam ir jākonfigurē kā pasīva pārbaude; Zabbix agent 2 to var konfigurēt kā aktīvu 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`

Trigeris konfigurēšana

Šajā lapā ir aprakstīts, kā konfigurēt trigeri Zabbix lietotāja saskarnē.

Ņemiet vērā, ka oficiālās Zabbix veidnes satur iepriekš definētus trigerus.
Skatiet rokasgrāmatu par veidnes lietošanu Linux uzraudzībai.
Ir arī iespējams veikt masveida atjaunināšanu konfigurētajiem trigeriem.

Citi resursi:

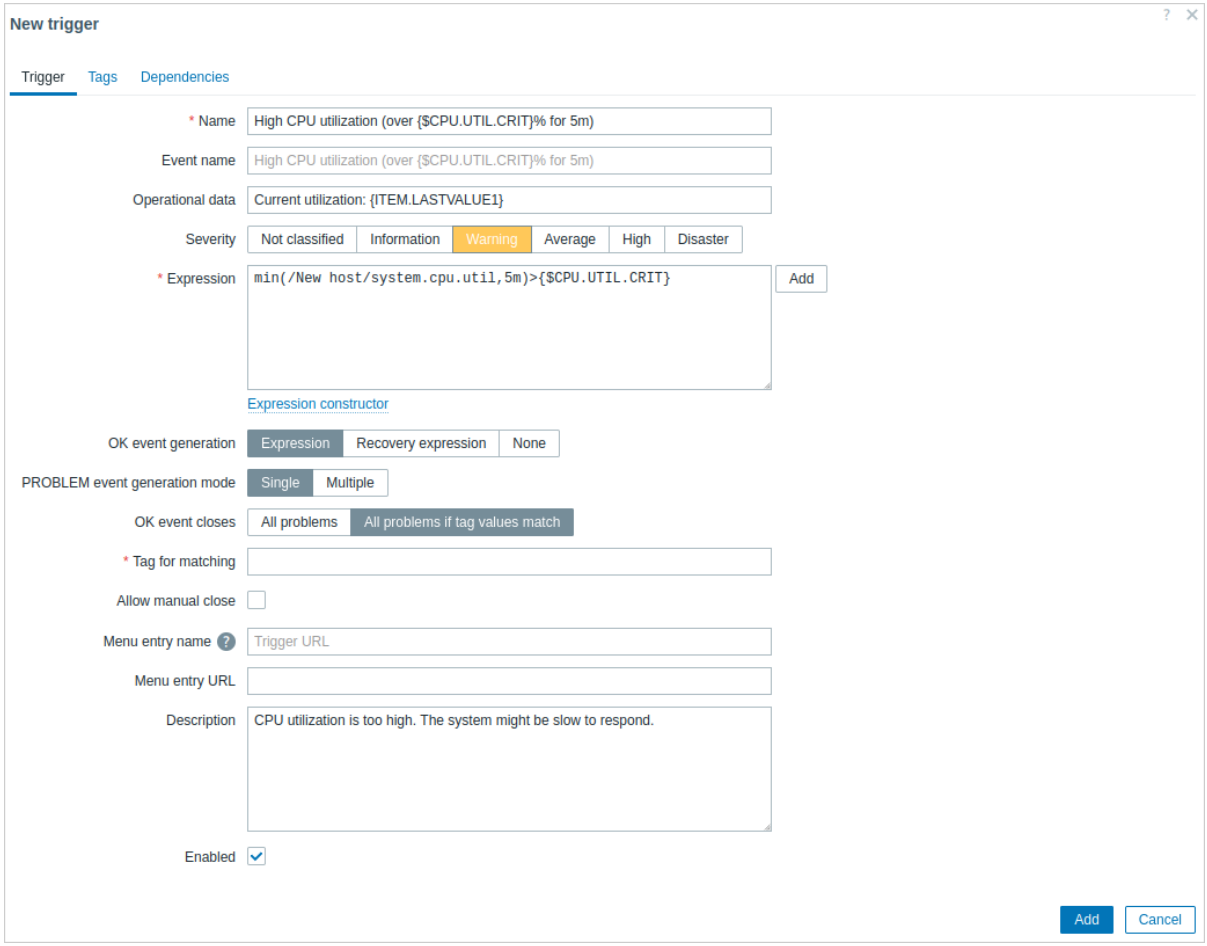
- Problēmu noteikšana ar trigeriem (vispārīgs ievads)
- Trigeru piemēri (pēc lietošanas gadījuma)
- Trigeru izteiksme un funkcijas (sintakses detaļas)
- Labākā prakse

Konfigurācija

Lai konfigurētu trigeri, rīkojieties šādi:

1. Dodieties uz *Datu vākšana* > *Hosts*.
2. Hosta rindā noklikšķiniet uz *Trigeri*.
3. Labajā pusē noklikšķiniet uz *Izveidot trigeri* (vai uz trigera nosaukuma, lai rediģētu esošu trigeri).
4. Formā ievadiet trigera parametrus.

Cilnē **Trigger** ir visi būtiskākie trigera atribūti.

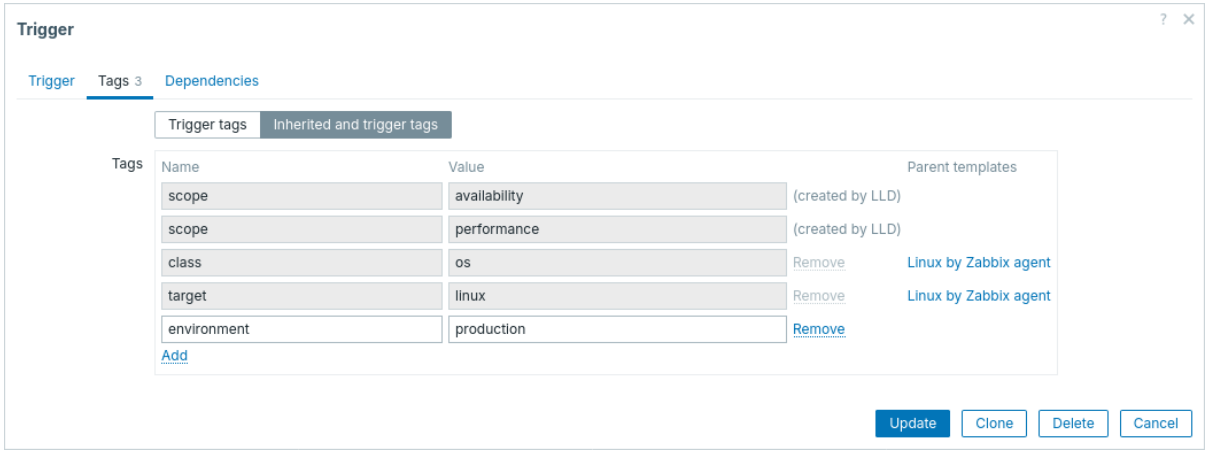


Visi obligātie ievades lauki ir atzīmēti ar sarkanu zvaigznīti.

Parameter	Description
Name	Trigera nosaukums. Atbalstītie makro 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 {$MACRO} lietotāja makro. Makro **$1, $2...$9** var izmantot, lai atsauktos uz izteiksmes pirmo, otro...devīto konstanti. *Piezīme*: $1-$9 makro tiks pareizi atrisināti, ja tie attiecas uz konstantēm salīdzinoši vienkāršās, tiešās izteiksmēs. Piemēram, nosaukums ”Processor load above $1 on {HOST.NAME}” automātiski mainīsies uz ”Processor load above 5 on New host”, ja izteiksme ir last(/New host/system.cpu.load[percpu,avg1])>5.
Event name	Ja tas ir definēts, šis nosaukums tiks izmantots problēmas notikuma nosaukuma izveidei. Pēc noklusējuma notikuma nosaukums ir tāds pats kā trigera nosaukums. Notikuma nosaukumu var izmantot, lai izveidotu jēgpilnus brīdinājumus, kas satur problēmas datus (skatiet piemēru). Tiek atbalstīts tas pats makro kopums kā trigera nosaukumā, kā arī {TIME}, {TIMESTAMP} un {?EXPRESSION} izteiksmes makro.
Operational data	Ievadiet virkni ar makro, lai *Monitoring* > *Problems* attēlotu dinamiskus reāllaika datus. Varat arī atstāt šo lauku tukšu, lai attēlotu visu vienumu jaunākās vērtības no trigera izteiksmes. Tiek atbalstīts tas pats makro kopums kā trigera nosaukumā, ar iespēju tos atrisināt dinamiski. Piemēram: {ITEM.VALUE<1-9>} tiek atrisināts uz vienuma vērtībām brīdī, kad trigera stāvoklis tiek mainīts (problēma izveidota, atrisināta, aizvērta manuāli vai aizvērta korelācijas dēļ). {ITEM.LASTVALUE<1-9>} tiek atrisināts uz jaunākajām vienuma vērtībām. Ņemiet vērā, ka problēmas manuāla aizvēršana nerada jaunu vērtību, tāpēc abi makro joprojām rādīs problēmas laika vērtību. Tāpat ņemiet vērā, ka abi makro tiek atrisināti kā *UNKNOWN*, ja jaunākā vērtība ir vecāka par *Max history display period* (skatiet *Administration > General*).
Severity	Iestatiet nepieciešamo trigera smaguma pakāpi, noklikšķinot uz pogām.

Parameter	Description
Expression	Loģiska izteiksme, ko izmanto, lai definētu problēmas nosacījumus. Tiek atbalstīti laika sufiksi un atmiņas apjoma sufiksi. Problēma tiek izveidota, kad ir izpildīti visi izteiksmē iekļautie nosacījumi, t. i., izteiksmes rezultāts ir TRUE. Problēma tiks atrisināta, tiklīdz izteiksmes rezultāts kļūs FALSE, ja vien papildu atjaunošanas nosacījumi nav norādīti sadaļā *Recovery expression*.
OK event generation	OK notikumu ģenerēšanas opcijas: **Expression** - OK notikumi tiek ģenerēti, pamatojoties uz to pašu izteiksmi kā problēmas notikumi; **Recovery expression** - OK notikumi tiek ģenerēti, ja problēmas izteiksmes rezultāts ir FALSE un atjaunošanas izteiksmes rezultāts ir TRUE; **None** - šajā gadījumā trigera stāvoklis nekad pats neatgriezīsies uz OK.
Recovery expression	Loģiska izteiksme (neobligāta), kas definē papildu nosacījumus, kuriem jābūt izpildītiem, pirms problēma tiek atrisināta, pēc tam, kad sākotnējā problēmas izteiksme jau ir novērtēta kā FALSE. Atjaunošanas izteiksme ir noderīga trigera histerēzei. Nav **iespējams** atrisināt problēmu tikai ar atjaunošanas izteiksmi, ja problēmas izteiksme joprojām ir TRUE. Šis lauks ir pieejams tikai tad, ja *OK event generation* ir atlasīts 'Recovery expression'.
PROBLEM event generation mode	Problēmas notikumu ģenerēšanas režīms: **Single** - viens notikums tiek ģenerēts, kad trigeris pirmo reizi pāriet stāvoklī 'Problem'; **Multiple** - notikums tiek ģenerēts pie *katras* trigera 'Problem' novērtēšanas.
OK event closes	Atlasiet, vai OK notikums aizver: **All problems** - visas šī trigera problēmas; **All problems if tag values match** - tikai tās trigera problēmas, kuru notikumu tagu vērtības sakrīt.
Tag for matching	Ievadiet notikuma taga nosaukumu, ko izmantot notikumu korelācijai. Šis lauks tiek parādīts, ja *OK event closes* īpašumam ir atlasīts 'All problems if tag values match', un šajā gadījumā tas ir obligāts.
Allow manual close	Atzīmējiet, lai atļautu šī trigera ģenerēto problēmas notikumu manuālu aizvēršanu. Manuāla aizvēršana ir iespējama, apstiprinot problēmas notikumus.
Menu entry name	Ja nav tukšs, šeit ievadītais nosaukums (līdz 64 rakstzīmēm) tiek izmantots vairākās lietotāja saskarnes vietās kā etiķete trigera URL, kas norādīts parametrā *Menu entry URL*. Ja lauks ir tukšs, tiek izmantots noklusējuma nosaukums *Trigger URL*. Tiek atbalstīts tas pats makro kopums kā trigera nosaukumā, kā arī {EVENT.ID}, {HOST.ID} un {TRIGGER.ID}.
Menu entry URL	Ja nav tukšs, šeit ievadītais URL (līdz 2048 rakstzīmēm) ir pieejams kā saite notikumu izvēlnē vairākās lietotāja saskarnes vietās, piemēram, noklikšķinot uz problēmas nosaukuma *Monitoring* > *Problems* vai *Problems* paneļa logrīkā. Tiek atbalstīts tas pats makro kopums kā trigera nosaukumā, kā arī {EVENT.ID}, {HOST.ID} un {TRIGGER.ID}. Piezīme: lietotāja makro ar slepenām vērtībām URL netiks atrisināti.
Description	Teksta lauks, ko izmanto, lai sniegtu papildu informāciju par šo trigeri. Var saturēt norādījumus konkrētas problēmas novēršanai, atbildīgo darbinieku kontaktinformāciju utt. Tiek atbalstīts tas pats makro kopums kā trigera nosaukumā.
Enabled	Noņemot atzīmi šajā izvēles rūtiņā, trigeris tiks atspējots, ja nepieciešams. Atspējota trigera problēmas vairs netiek rādītas lietotāja saskarnē, taču netiek dzēstas.

Cilne **Tags** ļauj definēt trigera līmeņa tagus. Visas šī trigera problēmas tiks marķētas ar šeit ievadītajām vērtībām.



Turklāt opcija *Inherited and trigger tags* ļauj skatīt veidnes līmenī definētos tagus, ja trigeris nāk no šīs veidnes. Ja ir vairākas veidnes

ar vienu un to pašu tagu, šie tagi tiek parādīti vienreiz, un veidņu nosaukumi tiek atdalīti ar komatiem. Trigeris ”nepārmanto” un nerāda hosta līmeņa tagus.

Parameter	Description
Name/Value	Iestatiet pielāgotus tagus, lai marķētu trigera notikumus. Tagi ir taga nosaukuma un vērtības pāris. Varat izmantot tikai nosaukumu vai to apvienot ar vērtību. Trigerim var būt vairāki tagi ar vienu un to pašu nosaukumu, bet atšķirīgām vērtībām. Lietotāja makro, lietotāja makro ar kontekstu, zemā līmeņa atklāšanas makro un makro funkcijas ar `{{ITEM.VALUE}}`, `{{ITEM.LASTVALUE}}` tiek atbalstīti notikumu tagos. Zemā līmeņa atklāšanas makro var izmantot makro kontekstā. {TRIGGER.ID} makro tiek atbalstīts trigera tagu vērtībās. Tas var būt noderīgs, lai identificētu trigerus, kas izveidoti no trigera prototipiem, un, piemēram, lai uzturēšanas laikā slāpētu šo trigera problēmas. Ja izvērstās vērtības kopējais garums pārsniedz 255, tas tiks saīsināts līdz 255 rakstzīmēm. Skatiet visus makro, kas atbalstīti notikumu tagiem. Notikumu tagus var izmantot notikumu korelācijai, darbību nosacījumos, un tie būs redzami arī *Monitoring* > *Problems* vai *Problems* logrīkā.

Cilne **Dependencies** satur visas trigera atkarības.

Noklikšķiniet uz *Add*, lai pievienotu jaunu atkarību.

Note:
Varat arī konfigurēt trigeri, atverot esošu trigeri, noklikšķinot uz pogas *Clone* un pēc tam saglabājot to ar citu nosaukumu.

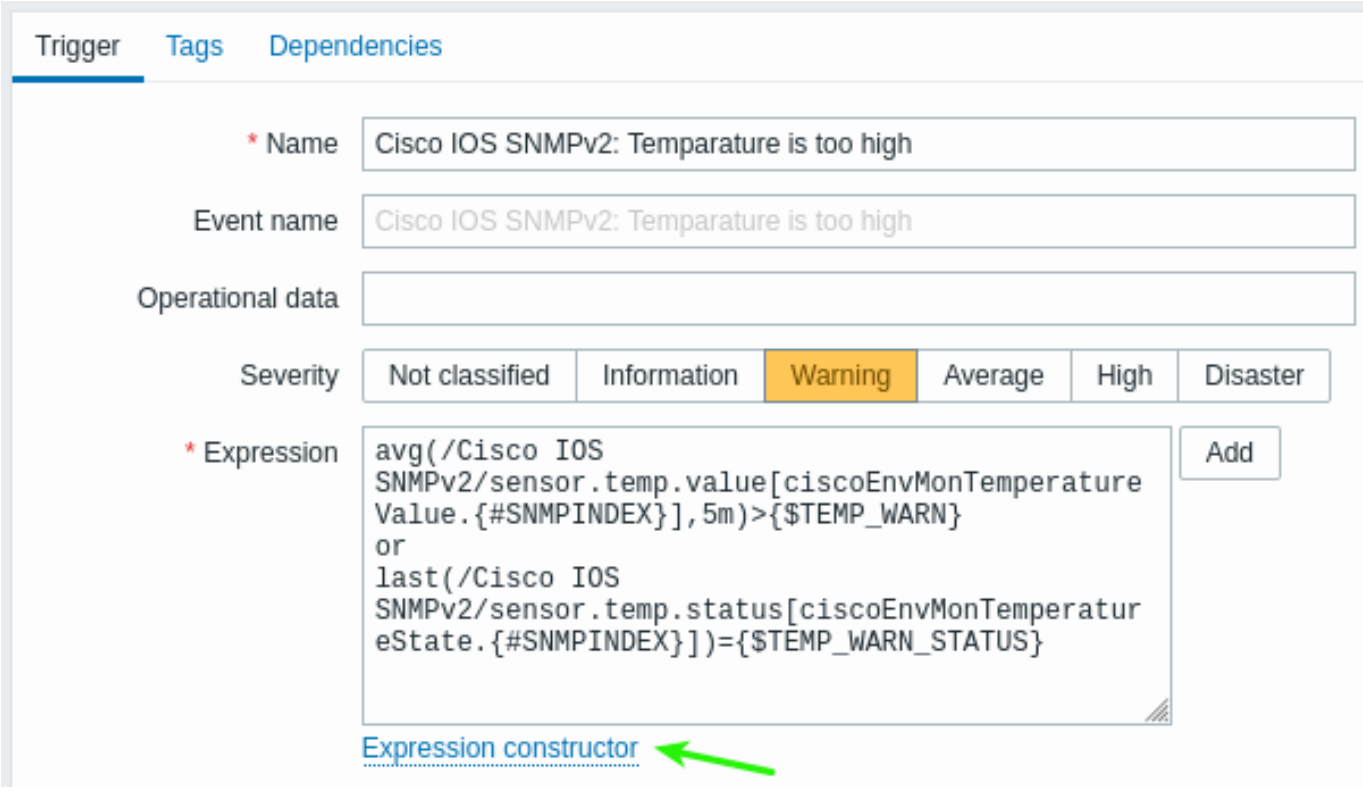
Trigeru izteiksmes testēšana

Ir iespējams pārbaudīt, kāds būtu izteiksmes rezultāts atkarībā no saņemtās vērtības.

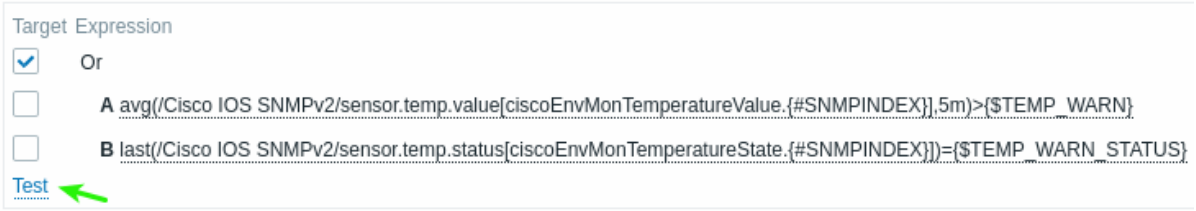
Kā piemērs tiek izmantota šāda izteiksme no oficiālas veidnes:

```
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}
```

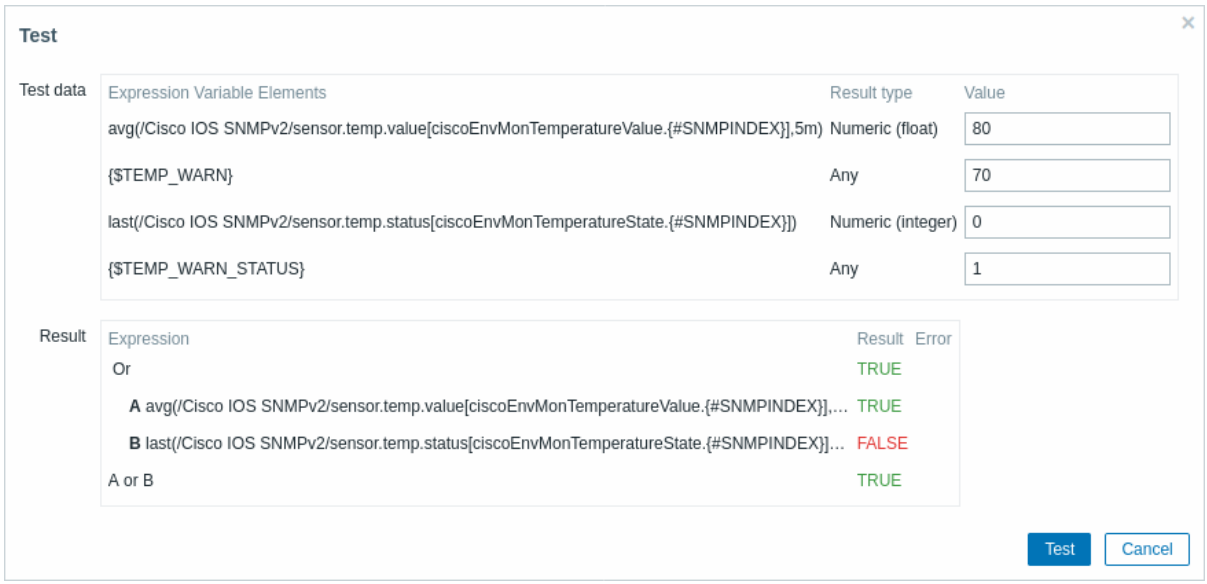
Lai pārbaudītu izteiksmi, izteiksmes laukā noklikšķiniet uz *Expression constructor*.



Expression constructor sadaļā tiek uzskaitītas visas atsevišķās izteiksmes. Lai atvērtu testēšanas logu, zem izteiksmju saraksta noklikšķiniet uz *Test*.



Testēšanas logā varat ievadīt parauga vērtības (’80’, ’70’, ’0’, ’1’ šajā piemērā) un pēc tam, noklikšķinot uz pogas *Test*, skatīt 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 ”TRUE” rezultāts.

”FALSE” nozīmē, ka norādītā izteiksme ir nepareiza. Šajā konkrētajā B gadījumā `{$TEMP_WARN_STATUS}` ”1” ir jābūt vienādam ar norādīto vērtību, kas šajā piemērā ir ”0”. Kā paredzēts, tiek parādīts ”FALSE” rezultāts.

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.

Trigeru izteiksme un funkcijas

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 apkopotās vērtības, piemēram, aprēķināt vidējo vērtību vai atrast noteiktu virkni.

Noklikšķiniet uz attiecīgās funkciju grupas, lai skatītu papildinformāciju.

Funkciju grupa	Funkcijas
Apkopošanas 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 operāciju 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, jsonpath, last, lastclock, logeventid, logseverity, logsource, logtimestamp, monodec, monoinc, nodata, percentile, rate, xmlxpath
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
Prognozēšanas funkcijas	forecast, timeleft
Virkņu funkcijas	ascii, bitlength, bytelength, char, concat, contains, insert, jsonpath, left, length, ltrim, mid, repeat, replace, right, rtrim, substring, trim, xmlxpath

Ja vien nav norādīts citādi, šīs funkcijas tiek atbalstītas:

- Trigeru izteiksmēs
- Aprēķinātajos vienumos
- Izteiksmju makros

Foreach funkcijas tiek atbalstītas tikai apkopošanas aprēķiniem.

Parasti funkcijas atgriež skaitliskas vērtības salīdzināšanai. Ja tiek atgrieztas virknes, salīdzināšana ir iespējama ar operatoriem **=** un **<>** (skatiet piemēru Neatbilstošas programmatūras noteikšana dažādos hostos).

Funkciju parametri

Funkciju parametri ļauj norādīt:

- vienuma atslēgu (kā `/host/key`) funkcijām, kas atsaucas uz hosts 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

Funkcijām specifiskie parametri tiek norādīti aiz vienuma atslēgas un no tās atdalīti 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 - piecas jaunākās vērtības).

Laika perioda norādīšanai varat izmantot sekundes vai laika sufiksus.
Ja parametra priekšā ir jaucējzīme, tam ir cita nozīme:

Izteiksme	Apraksts
sum(/host/key,**10m**)	Pēdējo 10 minūšu vērtību summa.
sum(/host/key,**#10**)	Pēdējo desmit vērtību summa.

Parametriem ar jaucējzīmi ir cita nozīme funkcijā **last** - tie apzīmē N-to iepriekšējo vērtību. Piemēram, ja vērtības ir 30, 70, 20, 60, 50 (no jaunākās līdz vecākajai):

- `last(/host/key,#2)` atgrieztu '70'
- `last(/host/key,#5)` atgrieztu '50'

Laika periods tiek mērīts līdz "tagad" - brīdim, kad trigeris tika pārrēķināts pēdējo reizi (skatiet sadaļu Aprēķināšanas biežums); "tagad" nav servera pašreizējais laiks.

Laika periods norāda vienu no šīm iespējām:

- Visas vērtības starp "tagad - laika periods" un "tagad" (vai, ja tiek lietota laika nobīde, starp "tagad - laika nobīde - laika periods" un "tagad - laika nobīde").
- Norādīto pagātnes vērtību skaitu līdz "tagad".

Ja norādītajā novērtēšanas periodā nav datu, 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 aprēķināts līdz vērtībai, kas tika saņemta pirms stundas.
- Jaunam trigerim aprēķins tiek veikts, tiklīdz tiek saņemta pirmā vērtība (vēstures funkcijas); datuma un laika un nodata() funkcijām aprēķins tiek veikts 30 sekunžu laikā. Tādējādi trigeris tiks aprēķināts, pat ja kopš trigera izveides vēl nav pagājis iestatītais laika periods (piemēram, viena stunda). Trigeris tiks aprēķināts arī pēc pirmās vērtības saņemšanas, pat ja laika diapazons ir iestatīts, piemēram, uz desmit jaunākajām vērtībām.

Laika nobīde

Ar laiku vai vērtību skaitu kā funkcijas parametru var izmantot neobligātu laika nobīdi. Šis parametrs ļauj atsaukties uz datiem no kāda pagātnes laika perioda.

Laika nobīde sākas ar `now`, kas norāda 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 vienai stundai pirms vienas dienas.

Attention:
Laika nobīde, kas norādīta mēnešos (M) un gados (y), tiek atbalstīta tikai tendenču funkcijās. 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 vienai dienai, no pirmdienas līdz svētdienai vienai nedēļai, no mēneša pirmās līdz pēdējai dienai vienam mēnesim.

Laika nobīde absolūtiem laika periodiem sākas ar `now`, kas norāda pašreizējo laiku, un tai seko jebkāds skaits laika darbību: `/<time unit>` nosaka laika vienības sākumu un beigas, piemēram, no pusnakts līdz pusnaktij vienai dienai, `+N<time unit>` vai `-N<time unit>`, lai pieskaitītu vai atņemtu N laika vienības.

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	Vakardiena
1d:now/d+1d	Šodiena
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 nezināms operands parādās trigeris izteiksmē, ja:

- tiek izmantots neatbalstīts vienums
- funkcijas novērtēšana atbalstītam vienumam beidzas ar kļūdu

Šādā gadījumā trigeris izteiksme parasti tiek novērtēta kā `Unknown` (jo to nevar novērtēt)

Ir iespējams saņemt paziņojumu par nezināmiem trigeriem.

Izņēmumi

Neraugoties uz nezināmu operandu, trigeris izteiksmes dažos gadījumos var tikt novērtētas līdz zināmam rezultātam (Problem/OK):

- `nodata()` funkcija tiek novērtēta neatkarīgi no tā, vai atsauces vienums ir atbalstīts vai nē.
- Izteiksmes ar AND/OR var tikt novērtētas līdz zināmam rezultātam divos gadījumos:
  - **1. gadījums**: "`1 or some_function(unsupported_item1) or some_function(unsupported_item2) or ...`" tiek novērtēta līdz zināmam rezultātam ('1' vai "Problem"),
  - **2. gadījums**: "`0 and some_function(unsupported_item1) and some_function(unsupported_item2) and ...`" tiek novērtēta līdz zināmam rezultātam ('0' vai "OK").
- Ja funkcijas novērtēšana atbalstītam vienumam beidzas ar kļūdu, funkcijas vērtība kļūst par `Unknown`, un tā turpmākajā izteiksmes novērtēšanā piedalās kā nezināms operands.

Nezināmi operandi var "pazust" tikai loģiskās izteiksmēs, kā aprakstīts iepriekš. Aritmētiskās izteiksmēs nezināmi operandi vienmēr noved pie `Unknown` (izņemot dalīšanu ar 0).

Attention:
Nezināms izteiksmes stāvoklis nemaina trigeris stāvokli (Problem/OK). Tātad, ja trigeris stāvoklis bija "Problem" (skat. 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 trigeris stāvokli.

Ja trigeris izteiksme ar vairākiem neatbalstītiem vienumiem tiek novērtēta kā `Unknown`, kļūdas ziņojums lietotāja saskarne 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.

Apkopošanas funkcijas

Ja nav norādīts citādi, visas šeit uzskaitītās funkcijas tiek atbalstītas:

- trigeru izteiksmēs
- aprēķināto vienumu formulās
- izteiksmju makrosos

Apkopošanas funkcijas var darboties ar:

- vienumu vēsturi, piemēram, `min(/host/key,1h)`

- foreach funkcijām kā vienīgo parametru, piemēram, `min(last_foreach(/*/key))` (tikai aprēķināto vienumu formulās; lai izmantotu trigeros, atsaucieties uz aprēķināto vienumu pēc tā atslēgas)

Funkcijas ir uzskaitītas bez papildu informācijas. Noklikšķiniet uz funkcijas, lai skatītu pilnu informāciju.

Funkcija	Apraksts
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-kvantiļu no histogrammas spaiņiem.
item_count	Esošo konfigurācijas vienumu skaits, kas atbilst filtra kritērijiem.
kurtosis	Varbūtības sadalījuma ”astes smagums” savāktajās vērtībās noteiktajā novērtēšanas periodā.
mad	Mediānas absolūtā novirze savāktajās vērtībās noteiktajā novērtēšanas periodā.
max	Augstākā vienuma vērtība noteiktajā novērtēšanas periodā.
min	Zemākā vienuma 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	Populācijas dispersija savāktajām vērtībām noteiktajā novērtēšanas periodā.
varsamp	Izlases dispersija savāktajām vērtībām noteiktajā novērtēšanas periodā.

Kopīgie parametri

- `/host/key` ir kopīgs obligāts pirmais parametrs funkcijām, kas izmanto hosts vienuma vēsturi.
- `(sec|#num)<:time shift>` ir kopīgs otrais parametrs funkcijām, kas izmanto hosts vienuma vēsturi, kur:
  - **sec** - maksimālais novērtēšanas periods sekundēs (var izmantot laika sufiksus).
  - **#num** - maksimālais novērtēšanas diapazons jaunākajās apkopotajās vērtībās (ja tā priekšā ir jaucējzīme).
  - **time shift** (neobligāts) ļauj pārvietot novērtēšanas punktu atpakaļ laikā. Plašāku informāciju par laika nobīdes norādīšanu skatiet šeit.

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) #aprēķināt vidējo vērtību pēdējai stundai līdz šim brīdim
avg(/host/key,#5) #aprēķināt piecu jaunāko vērtību vidējo vērtību
avg(/host/key,1h:now-1d) #aprēķināt vidējo vērtību vienai stundai no perioda pirms 25 stundām līdz perioda
avg(last_foreach(/*/system.cpu.load[,avg1]?[group="MySQL Servers"])) #aprēķināt vidējo procesora slodzi vi
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>)

Ar 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 jāskaita 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 `operators`:
eq - vienāds
ne - nav vienāds
gt - lielāks
ge - lielāks vai vienāds
lt - mazāks
le - mazāks vai vienāds
like - atbilst, ja satur paraugu (reģistrjutīgs)
bitand - bitu līmeņa AND
regexp - reģistrjutīga atbilstība regulārajai izteiksmei, kas norādīta `pattern`
iregexp - reģistrnejutīga atbilstība regulārajai izteiksmei, kas norādīta `pattern`
- **pattern** - nepieciešamais paraugs (virknes argumentiem jābūt dubultpēdiņās); atbalstīts, ja trešajā parametrā ir norādīts *operator*.

Piezīmes:

- Izmantojot **count()** kopā ar ar vēsturi saistītu foreach funkciju (max_foreach, avg_foreach u. c.), var rasties veiktspējas ietekme, savukārt, izmantojot **exists_foreach()**, kas darbojas tikai ar konfigurācijas datiem, šādas ietekmes nebūs.
- Neobligātos parametrus *operator* vai *pattern* pēc komata nevar atstāt tukšus; tie ir vai nu jānorāda pilnībā, vai arī pilnībā jāizlaiž.
- Ja trešais parametrs ir *bitand*, ceturto parametru `pattern` var norādīt kā divus skaitļus, kas atdalīti ar '/': **number_to_compare_with/mask**. count() aprēķina "bitwise AND" starp vērtību 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 trešais parametrs ir *regexp* vai *iregexp*, ceturtais parametrs `pattern` var būt parasta vai globāla (sākas ar '@') regulārā izteiksme. Globālo regulāro izteiksmju gadījumā reģistrjutība tiek mantota no globālo regulāro izteiksmju iestatījumiem. Regexp atbilstības nolūkos peldošā komata vērtības vienmēr tiks attēlotas ar 4 decimālzīmēm aiz '.'. Ņemiet vērā arī to, ka lieliem skaitļiem atšķirība decimālajā (glabātajā datubāzē) un binārajā (ko izmanto Zabbix serveris) attēlojumā var ietekmēt 4. decimālzīmi.

Piemēri:

```
count(max_foreach(/*/net.if.in[*],1h)) #aprēķina to net.if.in vienumu skaitu, kas saņēmuši datus pēdējās s
count(last_foreach(/*/vfs.fs.size[*,pused]),"gt",95) #aprēķina to failu sistēmu skaitu, kurās izmantots va
```

histogram_quantile(quantile,bucket1,value1,bucket2,value2,...)

Aprēķina φ-kvantili no histogrammas spaiņiem.
 Atbalstītā foreach funkcija: *bucket_rate_foreach*.

Parametri:

- **quantile** — $0 \leq \varphi \leq 1$;
- **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 definētajā 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:

```
sum(last_foreach(/*/net.if.in[*]?[group="Linux servers"])) #aprēķina kopējo ienākošās tīkla plūsmas apjomu
sum(last_foreach(/*/vfs.fs.size[/,total]?[group="MySQL Servers"])) #aprēķina kopējo diska vietas apjomu vi
sum(last_foreach(/*/net.dns[*,*,*])) #aprēķina kopējo veiksmīgo DNS pārbaužu skaitu
```

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 agregētajos aprēķinos, lai atgrieztu vienu agregēto vērtību katram vienumam, kas ir atlasīts, izmantojot **vienumu filtru**. Tiek atgriezts vērtību masīvs.

Piemēram, funkcija `avg_foreach` atgriezīs vērtību masīvu, kur katra vērtība ir atlasītā vienuma vēstures vērtību *vidējā* vērtība norādītajā laika intervālā.

> **Attention:**
> Foreach funkcijas tiek atbalstītas tikai aprēķinātā vienuma formulās. Tās nevar izsaukt parastās vienumu atslēgās vai izmantot tieši trigeru izteiksmju veidošanai. Trigeru izteiksmēs aprēķinātu vienumu var izmantot, atsaucoties uz tā atslēgu.

item filter ir daļa no sintakses, ko izmanto foreach funkcijas. Vienumu filtrā tiek atbalstīti aizstājējzīmes, tādējādi nepieciešamos vienumus var atlasīt ļoti elastīgi.

Atbalstītās funkcijas

Function	Description
avg_foreach	Atgriež vidējo vērtību katram vienumam.
bucket_rate_foreach	Atgriež pārus (bucket upper bound, rate value), kas ir piemēroti lietošanai funkcijā `histogram_quantile()`, kur "bucket upper bound" ir vienuma atslēgas parametra vērtība, kas definēta ar <parameter number> parametru.
count_foreach	Atgriež vērtību skaitu katram vienumam.
exists_foreach	Atgriež '1' katram iespējotajam vienumam.
last_foreach	Atgriež pēdējo vērtību katram vienumam.
max_foreach	Atgriež maksimālo vērtību katram vienumam.
min_foreach	Atgriež minimālo vērtību katram vienumam.
sum_foreach	Atgriež vērtību summu katram vienumam.

Funkcijas sintakse

Foreach funkcijas atbalsta divus kopīgus parametrus: `vienuma filtrs` (skatiet detalizētu informāciju tālāk) un `laika periods`:

```
foreach_function(item filter,time period)
```

Piemēram:

```
avg_foreach(/*/mysql.qps?[group="MySQL Servers"],5m)
```

atgriezīs katra vienuma 'mysql.qps' 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 apkopošanai.
Laika periodu var izteikt tikai kā laiku; vērtību daudzums (ar prefiksu **#**) netiek atbalstīts.

Šajā parametrā ērtības labad var izmantot atbalstītos mērvienību simbolus, piemēram, `5m` (piecas minūtes), nevis 300s (300 sekundes), vai `1d` (viena diena), nevis 86400 (86400 sekundes).

Funkcijai `last_foreach` laika periods ir neobligāts parametrs (atbalstīts kopš Zabbix 7.0), un to var izlaist:

```
last_foreach(/*/key?[group="host group"])
```

Laika periods netiek atbalstīts ar funkciju `exists_foreach`.

Papildu parametri

bucket_rate_foreach

Funkcija `bucket_rate_foreach` atbalsta trešo izvēles parametru:

```
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 `<parameter number>` ir 1.

count_foreach

Funkcija `count_foreach` atbalsta trešo un ceturto izvēles parametru:

```
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 paraugu (reģistrjutīgs)
bitand - bitu līmeņa AND
regexp - reģistrjutīga atbilstība regulārajai izteiksmei, kas norādīta `pattern`
iregexp - reģistrnejutīga atbilstība regulārajai izteiksmei, kas norādīta `pattern`.

- **pattern** ir nepieciešamais paraugs (virknes argumentiem jābūt dubultpēdiņās); atbalstīts, ja trešajā parametrā ir norādīts *operator*.

Piezīmes:

- Izvēles parametri *operator* vai *pattern* nevar tikt atstāti tukši aiz komata; tie var tikt tikai pilnībā izlaisti.
- Ja 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 skaitīta.
Ja *number_to_compare_with* un *mask* ir vienādi, jānorāda tikai *mask* (bez '/').
- Ja 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 globālo regulāro izteiksmju iestatījumiem. Regulārās izteiksmes salīdzināšanas nolūkos peldošā komata vērtības vienmēr tiks attēlotas ar 4 decimālzīmēm aiz '.'. Ņemiet vērā arī to, ka lieliem skaitļiem atšķirība decimālajā (glabātajā datubāzē) un binārajā (lietotajā Zabbix serveris) attēlojumā var ietekmēt 4. decimālzīmi.

Skatiet aggregate calculations, lai iegūtu vairāk informācijas un piemērus par foreach funkciju izmantošanu.

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.

Bitu funkcijas

Visas šeit uzskaitītās funkcijas tiek atbalstītas:

- Trigeru izteiksmēs
- Aprēķināto vienumu formulās
- Izteiksmju makrosos

Funkcijas ir uzskaitītas bez papildu informācijas. Noklikšķiniet uz funkcijas, lai skatītu pilnu informāciju.

Function	Description
bitand	"bitwise AND" vērtība vienuma vērtībai un maskai.
bitlshift	Vienuma vērtības bitu nobīde pa kreisi.
bitnot	"bitwise NOT" vērtība vienuma vērtībai.
bitor	"bitwise OR" vērtība vienuma vērtībai un maskai.
bitrshift	Vienuma vērtības bitu nobīde pa labi.
bitxor	"bitwise exclusive OR" 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.

Datuma un laika funkcijas

Visas šeit uzskaitītās funkcijas tiek atbalstītas:

- trigeru izteiksmēs
- aprēķināto vienumu formulās
- izteiksmju makrosos

Attention:
Datuma un laika funkcijas nevar izmantot izteiksmē pašas par sevi; izteiksmē jāiekļauj vismaz viena funkcija 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 formātā YYYYMMDD.
dayofmonth	Mēneša diena diapazonā no 1 līdz 31.
dayofweek	Nedēļas diena diapazonā no 1 līdz 7.
now	Sekunžu skaits kopš Unix laikmeta (1970. gada 1. janvāris, 00:00:00 UTC).
time	Pašreizējais laiks formātā HHMMSS.

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.

Vēstures funkcijas

Visas šeit uzskaitītās funkcijas tiek atbalstītas:

- Trigeru izteiksmēs
- Aprēķināto vienumu formulās
- Izteiksmju makros

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	Vērtības atbilstības meklēšana noteiktajā novērtēšanas periodā.
first	Pirmā (vecākā) vērtība noteiktajā novērtēšanas periodā.
firstclock	Pirmās (vecākās) vērtības laikspiedols noteiktajā novērtēšanas periodā.
fuzzytime	Pārbaudīt, cik ļoti pasīvā aģenta laiks atšķiras no Zabbix servera/starpniekservera laika.
jsonpath	Atgriezt JSONPath vaicājuma rezultātu masīvu.
last	Jaunākā vērtība.
lastclock	N-tās jaunākās vērtības laikspiedols noteiktajā novērtēšanas periodā.
logeventid	Pārbaudīt, vai pēdējā žurnāla ieraksta notikuma ID atbilst regulārai izteiksmei.
logseverity	Pēdējā žurnāla ieraksta žurnāla nopietnības līmenis.
logsource	Pārbaudīt, vai pēdējā žurnāla ieraksta avots atbilst regulārai izteiksmei.
logtimestamp	N-tās jaunākās žurnāla vienuma vērtības žurnāla ziņojuma laikspiedols.
monodec	Pārbaudīt, vai vērtības ir monotoniski samazinājušās.
monoinc	Pārbaudīt, vai vērtības ir monotoniski palielinājušās.
nodata	Pārbaudīt, vai nav saņemti dati.
percentile	Perioda P-tā procentile, kur P (procentuālā vērtība) ir norādīta ar trešo parametru.
rate	Monotoniski pieaugoša skaitītāja pieauguma vidējais ātrums sekundē noteiktajā laika periodā.
xmlxpath	Atgriezt XML XPath vaicājuma rezultātu masīvu.

Kopīgie parametri

- `/host/key` ir kopīgs obligāts pirmais parametrs funkcijām, kas atsaucas uz hosts vienuma vēsturi.
- `(sec|#num)<:time shift>` ir kopīgs otrais parametrs funkcijām, kas atsaucas uz hosts vienuma vēsturi, kur:
  - **sec** - maksimālais novērtēšanas periods sekundēs (var izmantot laika sufiksus), vai
  - **#num** - maksimālais novērtēšanas diapazons jaunākajās apkopotajās vērtībās (ja tā priekšā ir jaucējzīme).
  - **time shift** (neobligāts) ļauj pārvietot novērtēšanas punktu atpakaļ laikā. Skatiet papildinformāciju par laika nobīdes 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>)

Noteikto novērtēšanas periodu ietvaros esošo vērtību skaits.
 Atbalstītie vērtību tipi: *Float*, *Integer*, *String*, *Text*, *Log*.

Parametri:

- Skatiet kopējos parametrus
- **operator** (jābūt ietvertam dubultpēdiņās) Atbalstītie `operatori`:
eq - vienāds (noklusējums integer, float)
ne - nav vienāds
gt - lielāks par
ge - lielāks vai vienāds ar
lt - mazāks par
le - mazāks vai vienāds ar
like (noklusējums string, text, log) - atbilst, ja satur paraugu (reģistrjutīgs)
bitand - bitu līmeņa AND
regexp - atbilst dotajai regulārajai izteiksmei `pattern` (reģistrjutīgs)
iregexp - atbilst dotajai regulārajai izteiksmei `pattern` (reģistrnejutīgs)
- **pattern** - nepieciešamais paraugs (string argumentiem jābūt ietvertiem dubultpēdiņās)

Piezīmes:

- Float vienumi atbilst 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 *eq*, *ne*, *like*, *regexp* un *iregexp* operatori.
- Ja kā operators tiek izmantots *bitand*, ceturto `pattern` parametru var norādīt kā divus skaitļus, kas atdalīti ar '/': **skaitlis_salīdzināšanai/ar_masu**. count() aprēķina "bitwise AND" no vērtības un *maskas* un salīdzina rezultātu ar *skaitlis_salīdzināšanai*. Ja "bitwise AND" rezultāts ir vienāds ar *skaitlis_salīdzināšanai*, vērtība tiek skaitīta.
Ja *skaitlis_salīdzināšanai* un *mask* ir vienādi, jānorāda tikai *mask* (bez '/').
- Ja kā operators tiek izmantots *regexp* vai *iregexp*, ceturtais `pattern` parametrs var būt parasta vai globāla regulārā izteiksme (sākas ar '@'). Globālo regulāro izteiksmju gadījumā reģistrjutība tiek mantota no globālo regulāro izteiksmju iestatījumiem. Regulārās izteiksmes atbilstības nolūkos float vērtības vienmēr tiks attēlotas ar 4 decimāldaļām aiz '.'. Ņemiet vērā arī to, ka lieliem skaitļiem atšķirība starp decimālo (datubāzē saglabāto) un bināro (Zabbix serveris izmantoto) attēlojumu var ietekmēt 4. decimāldaļu.

Piemēri:

```
count(/host/icmpping,30m,,"0")>5 #ping ir neveiksmīgs vairāk nekā 5 reizes 30 minūšu laikā
count(/host/key,10m,"like","error") #vērtību skaits pēdējo 10 minūšu laikā līdz šim brīdim, kas satur 'err
count(/host/key,10m,,12) #vērtību skaits pēdējo 10 minūšu laikā līdz šim brīdim, kas ir vienādas ar '12'
count(/host/key,10m,"gt",12) #vērtību skaits pēdējo 10 minūšu laikā līdz šim brīdim, kas ir lielākas par '
count(/host/key,#10,"gt",12) #vērtību skaits pēdējo 10 vērtību ietvaros līdz šim brīdim, kas ir lielākas p
count(/host/key,10m:now-1d,"gt",12) #vērtību skaits starp 24 stundām un 10 minūtēm un 24 stundām pirms šī
count(/host/key,10m,"bitand","6/7") #vērtību skaits pēdējo 10 minūšu laikā līdz šim brīdim, kurām 3 vismaz
count(/host/key,10m:now-1d) #vērtību skaits starp 24 stundām un 10 minūtēm un 24 stundām pirms šī brīža
```

countunique(/host/key,(sec|#num)<:time shift>,<operator>,<pattern>)

Noteikto novērtēšanas periodu laikā unikālo vērtību skaits.
 Atbalstītie vērtību tipi: *Float*, *Integer*, *String*, *Text*, *Log*.

Parametri:

- Skatiet kopīgos parametrus
- **operator** (jābūt ietvertam dubultpēdiņās). Atbalstītie `operatori`:
eq - vienāds (noklusējums integer, float)
ne - nav vienāds
gt - lielāks par
ge - lielāks vai vienāds ar
lt - mazāks par
le - mazāks vai vienāds ar
like (noklusējums string, text, log) - atbilst, ja satur paraugu (reģistrjutīgs)
bitand - bitu līmeņa AND
regexp - atbilst dotajai regulārajai izteiksmei, reģistrjutīgi, kas norādīta `pattern`
iregexp - atbilst dotajai regulārajai izteiksmei, nereģistrjutīgi, kas norādīta `pattern`
- **pattern** - nepieciešamais paraugs (string argumentiem jābūt ietvertiem dubultpēdiņās)

Piezīmes:

- Float vienumi atbilst 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 *eq*, *ne*, *like*, *regexp* un *iregexp* operatori.
- Ja kā operators ir *bitand*, ceturto `pattern` parametru var norādīt kā divus skaitļus, kas atdalīti ar '/': **number_to_compare_with/mask**. countunique() 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 skaitīta.
Ja *number_to_compare_with* un *mask* ir vienādi, jānorāda tikai *mask* (bez '/').
- Ja kā operators ir *regexp* vai *iregexp*, ceturtais `pattern` parametrs var būt parasta vai globāla (sākas ar '@') regulārā izteiksme. Globālo regulāro izteiksmju gadījumā reģistrjutība tiek mantota no globālo regulāro izteiksmju iestatījumiem. Regulārās izteiksmes atbilstības nolūkos float vērtības vienmēr tiks attēlotas ar 4 decimāldaļām aiz '.'. Ņemiet vērā arī to, ka lieliem skaitļiem atšķirība decimālajā (glabātajā datubāzē) un binārajā (lietotajā Zabbix serveris) attēlojumā var ietekmēt 4. decimāldaļu.

Piemēri:

```
countunique(/host/key,10m) #unikālo vērtību skaits pēdējo 10 minūšu laikā līdz šim brīdim
countunique(/host/key,10m,"like","error") #unikālo vērtību skaits pēdējo 10 minūšu laikā līdz šim brīdim,
countunique(/host/key,10m,,12) #unikālo vērtību skaits pēdējo 10 minūšu laikā līdz šim brīdim, kas ir vien
countunique(/host/key,10m,"gt",12) #unikālo vērtību skaits pēdējo 10 minūšu laikā līdz šim brīdim, kas ir
countunique(/host/key,#10,"gt",12) #unikālo vērtību skaits pēdējo 10 vērtību laikā līdz šim brīdim, kas ir
countunique(/host/key,10m:now-1d,"gt",12) #unikālo vērtību skaits starp 24 stundām un 10 minūtēm un 24 stu
countunique(/host/key,10m,"bitand","6/7") #unikālo vērtību skaits pēdējo 10 minūšu laikā līdz šim brīdim,
countunique(/host/key,10m:now-1d) #unikālo vērtību skaits starp 24 stundām un 10 minūtēm un 24 stundas atp
```

find(/host/key,(sec|#num)<:time shift>,<operator>,<pattern>)

Atrod vērtības atbilstību noteiktajā novērtēšanas periodā.
 Atbalstītie vērtību tipi: *Float*, *Integer*, *String*, *Text*, *Log*.
 Atgriež: 1 - atrasts; 0 - citādi.

Parametri:

- Skatiet kopīgos parametrus
- **sec** vai **#num** (neobligāti) - ja nav norādīts, pēc noklusējuma tiek izmantota jaunākā vērtība
- **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 AND
regexp - reģistrjutīga atbilstība regulārajai izteiksmei, kas norādīta `pattern`
iregexp - reģistrnejutīga atbilstība regulārajai izteiksmei, kas norādīta `pattern`
- **pattern** - nepieciešamais paraugs (virknes argumentiem jābūt dubultpēdiņās); Perl Compatible Regular Expression (PCRE) regulārā izteiksme, ja `operator` ir *regexp*, *iregexp*

Piezīmes:

- 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 *eq*, *ne*, *like*, *regexp* un *iregexp* operatori.
- Ja kā operators ir *regexp* vai *iregexp*, ceturtais `pattern` parametrs var būt parasta vai globāla (sākas ar '@') regulārā izteiksme. Globālo regulāro izteiksmju gadījumā reģistrjutība tiek mantota no globālo regulāro izteiksmju iestatījumiem.

Piemēri:

```
find(/host/key,10m,"like","error") #atrod vērtību, kas pēdējo 10 minūšu laikā līdz šim brīdim satur 'error
find(/host/agent.version,,"like","beta")=1 #Zabbix aģents ir beta versijā, tas ir jājaunina
find(/host/log[/var/log/nginx/access.log],,"regexp"," 500 ")=1 #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 ļoti pasīvā aģenta laiks atšķiras no Zabbix servera/starpniekservera laika.
 Atbalstītie vērtību tipi: *Float*, *Integer*.
 Atgriež: 1 - starpība starp pasīvā vienuma vērtību (kā laikspiedolu) un Zabbix servera/starpniekservera laikspiedolu (vērtības savākšanas pulksteni) ir mazāka vai vienāda ar *sec* sekundēm; 0 - citādi.

Parametri:

- Skatiet kopīgos parametrus.

Komentāri:

- Parasti tiek izmantota kopā ar 'system.localtime' vienumu, lai pārbaudītu, vai vietējais laiks ir sinhronizēts ar Zabbix servera vietējo laiku. *Piezīme*, ka 'system.localtime' Zabbix aģentam ir jākonfigurē kā pasīvā pārbaude; Zabbix agent 2 to var konfigurēt kā aktīvu pārbaudi.
- To var izmantot arī ar `vfs.file.time[/path/file,modify]` atslēgu, lai pārbaudītu, vai fails ilgu laiku nav saņēmis atjauninājumus.
- Šo funkciju nav ieteicams izmantot sarežģītās trigeru izteiksmēs (ar vairākiem vienumiem), jo tā var radīt negaidītus rezultātus (laika starpība tiks mērīta ar visjaunāko metriku), piemēram, `fuzzytime(/Host/system.localtime,60s)=0 or last(/Host/trap)<>0`.

Piemēri:

```
fuzzytime(/host/system.localtime,5m)=0 #klienta vietējais laiks atšķiras no Zabbix servera/starpniekserver
fuzzytime(/host/system.localtime,5m)=0 and nodata(/host/system.localtime,10m)=0 #klienta vietējais laiks a
```

jsonpath(/host/key,(sec|#num)<:time shift>,pattern)

Atgriež JSONPath vaicājumu rezultātu masīvu.
 Atbalstītie vērtību tipi: *String*, *Text*, *Log*.

Parametri:

- Skatiet kopīgos parametrus
- **pattern** - nepieciešamais raksts (virknes argumenti jāiekļauj dubultpēdiņās)

Komentāri:

- Šo funkciju izmanto kopā ar apkopošanas funkcijām.
- JSONPath vaicājums tiek izpildīts katrai atlasītajai vēstures vērtībai; katrs veiksmīgs vaicājuma rezultāts, kas nav tukšs, tiek pievienots masīvam. Ja vienas vēstures vērtības vaicājuma rezultāts satur vairākas atbilstības, katra atbilstība tiek pievienota masīvam atsevišķi.
- Funkcija neizdodas ar kļūdu, ja:
  - `pattern` formāts nav derīgs.
  - Kāda no vēstures vērtībām satur nederīgu JSON vai JSONPath vaicājums šai vērtībai neizdodas.

- Funkcija atgriež tukšu masīvu, ja:
  - Netika atlasīta neviena vēstures vērtība.
  - Visas atlasītās vēstures vērtības satur nederīgu JSON.

Piemērs:

```
max(jsonpath(/host/http.response,5m,"$.response_time")) > 500
```

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-tākais jaunākais vērtības ieraksts
- **pattern** (neobligāts) - regulārā izteiksme, kas apraksta nepieciešamo rakstu, Perl Compatible Regular Expression (PCRE) stilā (virknes argumentiem jābūt ietvertiem dubultpēdiņās)

Piemēri:

```
logeventid(/host/eventlog[Security],,"^4625$")=1 #atrasts žurnāla ieraksts ar ID, kas atbilst "4625" (neiz
logeventid(/host/eventlog[System],,"^(6008|41)$")=1 #atrasts žurnāla ieraksts ar ID, kas atbilst "6008" va
```

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-tākais jaunākais vērtības ieraksts
- **pattern** (neobligāts) - regulārā izteiksme, kas apraksta nepieciešamo rakstu, Perl Compatible Regular Expression (PCRE) stilā (virknes argumentiem jābūt ievietotiem dubultpēdiņās)

Parasti tiek izmantots Windows notikumu žurnāliem.

Piemēri:

```
logsource(/host/eventlog[Application],,"MSSQLSERVER")=1 #atrasts žurnāla ieraksts, kura avots atbilst "MSS
logsource(/host/eventlog[System],,"Service Control Manager")=1 #atrasts žurnāla ieraksts, kura avots atbil
logsource(/host/eventlog[System],,"Service Control Manager")=1 and logeventid(/host/eventlog[System],,"^70
```

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 nav saņemti dati; 0 - pretējā gadījumā.

Parametri:

- Skatiet kopīgos parametrus
- **sec** - periods nedrīkst būt īsāks 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 neņems vērā starpniekservera pieejamību (detalizētu informāciju skatiet komentāros)

Komentāri:

- pēc noklusējuma starpniekservera uzraudzītie trigeri 'nodata' ņem vērā starpniekservera pieejamību - ja starpniekserveris kļūst nepieejams, trigeri 'nodata' neaktivizēsies uzreiz pēc savienojuma atjaunošanas, bet izlaidīs datus par aizkavēto periodu. Ņemiet vērā, ka pasīviem starpniekserveriem apspiešana tiek aktivizēta, ja savienojums tiek atjaunots pēc vairāk nekā 15 sekundēm, bet ne vēlāk kā pēc 2 sekundēm. Aktīviem starpniekserveriem apspiešana tiek aktivizēta, ja savienojums tiek atjaunots pēc vairāk nekā 15 sekundēm. Lai izslēgtu starpniekservera pieejamības ņemšanu vērā, izmantojiet trešo parametru, piemēram: `nodata(/host/key,5m,"strict")`; šādā 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 uzturēšana tika pabeigta
- nav datu un vienums tika pievienots vai atkārtoti iespējots.
- Kļūdas tiek parādītas trigeru saraksta kolonnā *Info*.
- Šī funkcija var nedarboties pareizi, ja starp Zabbix serveri, starpniekserveri un aģentu pastāv laika atšķirības. Skatiet arī: Laika sinhronizācijas prasība.

Piemērs:

```
nodata(/host/agent.ping,5m)=1 #trigger if no data has been received from Zabbix agent for 5 minutes
```

percentile(/host/key,(sec|#num)<:time shift>,percentage)

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
```

xmlxpath(/host/key,(sec|#num)<:time shift>,pattern)

Atgriež XML XPath vaicājuma rezultātu masīvu.
 Atbalstītie vērtību tipi: *String*, *Text*, *Log*.

Parametri:

- Skatiet kopīgos parametrus
- **pattern** - nepieciešamais šablons (virknes argumenti jāiekļauj dubultpēdiņās)

Komentāri:

- Šo funkciju izmanto kopā ar apkopošanas funkcijām.
- XML XPath vaicājums tiek izpildīts katrai atlasītajai vēstures vērtībai; katrs veiksmīgs vaicājuma rezultāts, kas nav tukšs, tiek pievienots masīvam. Ja vienas vēstures vērtības vaicājuma rezultāts satur vairākas atbilstības, katra atbilstība tiek pievienota masīvam atsevišķi.
- Funkcija neizdodas ar kļūdu, ja:
  - `pattern` formāts nav derīgs.
  - Zabbix tika kompilēts bez XML atbalsta.
  - Kāda no vēstures vērtībām satur nederīgu XML vai XPath vaicājums šai vērtībai neizdodas.
- Funkcija atgriež tukšu masīvu, ja:
  - Netika atlasīta neviena vēstures vērtība.
  - Visas atlasītās vēstures vērtības satur nederīgu XML.

Piemērs:

```
max(xmlxpath(/host/http.response,5m,"/response/response_time")) > 500
```

Skatiet visas atbalstītās funkcijas.

Trend funkcijas

Trend funkcijas, atšķirībā no vēstures funkcijām, aprēķinos izmanto trend datus.

Trend dati glabā stundas agregētās vērtības. Trend funkcijas izmanto šos stundas vidējos rādītājus, tāpēc tās ir noderīgas ilgtermiņa analīzei.

Trend funkciju rezultāti tiek kešoti, tāpēc vairāki vienas un tās pašas funkcijas izsaukumi ar vienādiem parametriem informāciju no datubāzes iegūst tikai vienu reizi. Trend funkciju kešatmiņu kontrolē servera parametrs TrendFunctionCacheSize.

Izmantojiet `zabbix[tcache,cache,paccessed]`, lai novērtētu, cik liela daļa no kešatmiņas faktiski tiek izmantota. Zems `paccessed`, kopā ar zemu `pitems` un zemu `pmisses`, parasti nozīmē, ka kešatmiņas lielumu var samazināt.

Trigeri, kas atsaucas tikai uz trend funkcijām, tiek izvērtēti vienu reizi katrā mazākajā laika periodā izteiksmē. Piemēram, trigeris kā

```
trendavg(/host/key,1d:now/d) > 1 or trendavg(/host/key2,1w:now/w) > 2
```

tiks izvērtēts vienu reizi dienā. Ja trigeris satur gan trend, 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 (izmantojot stddevpop algoritmu) starp pēdējo datu periodu un tiem pašiem datu periodiem iepriekšējās sezonās.
baselinewma	Aprēķina bāzes līniju, vidējo vērtību iegūstot no tā paša laika perioda vairākos vienādos laika periodos (”sezonās”), izmantojot svērto mainīgo vidējo algoritmu.
trendavg	Trend vērtību vidējā vērtība noteiktajā laika periodā.
trendcount	Veiksmīgi izgūto vēstures vērtību skaits, kas izmantots trend vērtības aprēķināšanai noteiktajā laika periodā.

Function	Description
trendmax	Trend vērtību maksimums noteiktajā laika periodā.
trendmin	Trend vērtību minimums noteiktajā laika periodā.
trendstl	Atgriež anomāliju biežumu noteikšanas periodā - decimālvērtību no 0 līdz 1, kas ir `((anomālo vērtību skaits)/(kopējais vērtību skaits))`.
trendsum	Trend 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.

Matemātiskās funkcijas

Visas šeit uzskaitītās funkcijas tiek atbalstītas:

- Trigeru izteiksmēs
- Aprēķināto vienumu formulās
- Izteiksmju makrosos

Matemātiskās funkcijas tiek atbalstītas ar 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 arkkosinuss kā leņķis, izteikts radiānos.
asin	Vērtības arksinuss kā leņķis, izteikts radiānos.
atan	Vērtības arktangenss kā leņķis, izteikts radiānos.
atan2	Norādīto ordinātas (vērtības) un abscisas koordinātu arktangenss kā leņķis, izteikts radiānos.
avg	Atsauces vienumu vērtību vidējā vērtība.
cbrt	Vērtības kubsakne.
ceil	Noapaļo vērtību uz augšu līdz tuvākajam lielākajam vai vienādam 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	Pārvērš vērtību no radiāniem grādos.
e	Eilera skaitlis (2.718281828459045).
exp	Eilera skaitlis, kas pacelts vērtības pakāpē.
expm1	Eilera skaitlis, kas pacelts vērtības pakāpē, mīnus 1.
floor	Noapaļo vērtību uz leju līdz tuvākajam mazākajam vai vienādam veselajam skaitlim.
log	Naturālais logaritms.
log10	Decimālais logaritms.
max	Atsauces vienumu vērtību lielākā vērtība.
min	Atsauces vienumu vērtību mazākā vērtība.
mod	Dalīšanas atlikums.
pi	Pi konstante (3.14159265358979).
power	Vērtības pakāpe.
radians	Pārvērš vērtību no grādiem radiānos.
rand	Atgriež nejaušu veselu skaitli.
round	Noapaļo vērtību līdz decimāldaļām.
signum	Atgriež '-1', ja vērtība ir negatīva, '0', ja vērtība ir nulle, '1', ja vērtība ir pozitīva.

Function	Description
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	Atsauces vienumu vērtību summa.
tan	Vērtības tangenss.
truncate	Saīsina vērtību līdz decimāldaļām.

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.

Operatoru funkcijas

Visas šeit uzskaitītās funkcijas tiek atbalstītas:

- Trigeru izteiksmēs
- Aprēķināto vienumu formulās
- Izteiksmju makros

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.

Prognozēšanas funkcijas

Visas šeit uzskaitītās funkcijas tiek atbalstītas:

- Trigeru izteiksmēs
- Aprēķināto vienumu formulās
- 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 jaunākajās apkopotajās vērtībās (ja tā priekšā ir jaucējzīme).
  - **time shift** (neobligāts) - ļauj pārvietot novērtēšanas punktu atpakaļ laikā. Skatiet papildinformāciju par laika nobīdes 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, maksimālā, minimālā, delta vai vidējā vērtība.
 Atbalstītie vērtību tipi: *Float*, *Integer*.

Parametri:

- Skatiet kopīgos parametrus;
- **time** - prognozēšanas periods sekundēs (piemēram, 1, 60) (var izmantot laika sufiksus (piemēram, 30s, 1m)); tiek atbalstītas negatīvas vērtības;
- **fit** (neobligāts; jānorāda dubultpēdiņās) - funkcija, ko izmanto vēsturisko datu pielāgošanai. Atbalstītās funkcijas:
linear - lineāra funkcija (noklusējums)
polynomialN - N-tās pakāpes polinoms (1 <= N <= 6)
exponential - eksponenciāla funkcija
logarithmic - logaritmiska funkcija
power - pakāpes funkcija
Ņemiet vērā, ka *polynomial1* ir ekvivalenta *linear*;

- **mode** (neobligāts; jānorāda dubultpēdiņās) - pieprasītais rezultāts. Atbalstītie režīmi:
value - vērtība (noklusējums)
max - maksimālā vērtība
min - minimālā vērtība
delta - *max-min*
avg - vidējā vērtība
Ņemiet vērā, ka *value* aprēķina vienuma vērtību brīdī `now + time`; *max*, *min*, *delta* un *avg* analizē vienuma aprēķināto vērtību intervālā no `now` līdz `now + time`.

Komentāri:

- Ja atgriežamā vērtība ir lielāka par 1.7976931348623158E+308 vai mazāka par -1.7976931348623158E+308, atgrieztā vērtība tiek ierobežota attiecīgi līdz 1.7976931348623158E+308 vai -1.7976931348623158E+308;
- Kļūst neatbalstīta tikai tad, ja izteiksmē tiek izmantota nepareizi (nepareizs vienuma tips, nederīgi parametri); pretējā gadījumā kļūdu gadījumā atgriež -1;
- Papildinformāciju skatiet sadaļā par prognozējošiem 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) #aprēķina vienuma vērtību pirms 20 minūtēm, pamatojoties uz pēdējām divām vērt
```

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 mērvienību sufiksus);
- **fit** (neobligāts; jāiekļauj 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 izteiksmē tiek izmantots nepareizi (nepareizs vienuma tips, nederīgi parametri); pretējā gadījumā kļūdu gadījumā atgriež -1;
- Skatiet arī papildu informāciju par prognozējošām 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 dati
timeleft(/host/key,1h:now-1d,100) #laiks, līdz vienuma vērtība sasniedz 100, pamatojoties uz vienas stunda
timeleft(/host/key,1h,200,"polynomial2") #laiks, līdz vienuma vērtība sasniedz 200, pamatojoties uz pēdējā
```

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	$\max_{\text{now <= t <= now + time}}$ f(t)
min	$\min_{\text{now <= t <= now + time}}$ 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], or {TRIGGER.VALUE}=1)`

Virknes funkcijas

Visas šeit norādītās funkcijas tiek atbalstītas:

- Trigeru izteiksmēs
- Aprēķināto vienumu formulās
- Izteiksmju makros

Funkcijas ir norādītas bez papildu informācijas. Noklikšķiniet uz funkcijas, lai skatītu pilnu informāciju.

Function	Description
ascii	Vērtības kreisākās rakstzīmes ASCII kods.
bitlength	Vērtības garums bitos.
bytelength	Vērtības garums baitos.
char	Atgriež rakstzīmi, interpretējot vērtību kā ASCII kodu.
concat	Virkne, kas iegūta, savienojot atsauces vienumu vērtības vai konstantās vērtības.
contains	Atgriež 1, ja vērtība satur apakšvirkni, kas norādīta ar `search value`, pretējā gadījumā atgriež 0.
insert	Ievieto norādītās rakstzīmes vai atstarpes rakstzīmju virknē, sākot no norādītās pozīcijas virknē.
jsonpath	Atgriež JSONPath rezultātu.
left	Atgriež vērtības kreisākās rakstzīmes.

Function	Description
length	Vērtības garums rakstzīmēs.
ltrim	Noņem norādītās rakstzīmes virknes sākumā.
mid	Atgriež N rakstzīmju apakšvirkni, sākot no rakstzīmes pozīcijas, kas norādīta ar `start`. Šī funkcija ir funkcijas `substring()` aizstājvārds.
repeat	Atkārto virkni.
replace	Atrod vērtībā rakstu un aizstāj to ar aizstājējtekstu.
right	Atgriež vērtības labākās rakstzīmes.
rtrim	Noņem norādītās rakstzīmes virknes beigās.
substring	Atgriež N rakstzīmju apakšvirkni, sākot no rakstzīmes pozīcijas, kas norādīta ar `start`.
trim	Noņem norādītās rakstzīmes virknes sākumā un beigās.
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])
```

contains(value, search value)

Atgriež 1, ja vērtība satur apakšvirkni, kas norādīta parametrā `search value`, pretējā gadījumā atgriež 0.| Atbalstītie vērtību tipi: *String*, *Text*, *Log*.

Parametri:

- **value** - pārbaudāmā vērtība;
- **search value** - meklējamā vērtība.

Komentāri:

- Meklēšana ir reģistrjutīga.
- Ja vērtība ir derīga un `search value` ir tukšs, funkcija atgriež 1.

Piemēram, funkcija atgriež 1, ja vērtība satur "Zabbix".

Piemērs:

```
contains(last(/host/key),"Zabbix")
```

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 N rakstzīmju apakšvirkni, sākot no rakstzīmes pozīcijas, kas norādīta ar parametru "start".
Šī funkcija ir funkcijas `substring()` aizstājvārds.
 Atbalstītie vērtību tipi: *String*, *Text*, *Log*.

Parametri:

- **value** - pārbaudāmā vērtība;
- **start** - apakšvirknes sākuma pozīcija;
- **length** - apakšvirknē atgriežamo pozīciju skaits.

Piemēram, no tādas vērtības kā "Zabbix" var atgriezt "abbi", 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
```

substring(value,start,length)

Atgriež N rakstzīmju apakšvirkni, sākot no rakstzīmes pozīcijas, kas norādīta ar parametru 'start'.
 Atbalstītie vērtību tipi: *String*, *Text*, *Log*.

Parametri:

- **value** - pārbaudāmā vērtība;
- **start** - apakšvirknes sākuma pozīcija;
- **length** - apakšvirknē atgriežamo pozīciju skaits.

Piemēram, no tādas vērtības kā 'Zabbix' var atgriezt 'abbi', ja sākuma pozīcija ir 2 un atgriežamo pozīciju skaits ir 4.

Piemērs:

```
substring(last(/host/key),2,4)="abbi"
```

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.

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 padomus:

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 **percentile** funkciju (iestatiet to uz 95% vai 5%) trigeros, ja vēlaties izvairīties no brīdinājumiem par nejaušiem pīķiem un kritumiem.
3. Izmantojiet histerēzi, lai izvairītos no trigeru svārstīšanās — biežām trigeru stāvokļa izmaiņām (OK ↔ Problem). Problēmas stāvokļa nepārtrauktību var definēt, pievienojot atjaunošanas izteiksmi (atsevišķu nosacījumu problēmas atrisināšanai).
4. Izmantojiet trigeru atkarības, lai izvairītos no brīdinājumiem, kas nav saistīti ar pamatcēloni.
5. Izmantojiet trigeru smaguma pakāpi, 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).

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 hosta. Serveris, kas atrodas aiz maršrutētāja, kļūs nepieejams, ja maršrutētājs pārstās darboties. Ja abiem ir konfigurēti trigeri, jūs varat saņemt paziņojumus par diviem nepieejamiem hostiem, lai gan problēmas cēlonis bija tikai maršrutētājs.

Šādos gadījumos var būt noderīga noteikta atkarība starp hostiem. Ja atkarība ir iestatīta, atkarīgā hosta paziņojumi var tikt aizturēti un nosūtīts tikai paziņojums par pamatproblēmu.

Lai gan Zabbix tieši neatbalsta atkarības starp hostiem, tās var definēt, izmantojot citu, elastīgāku metodi - trigeru atkarības. Trigeris var būt atkarīgs no viena vai vairākiem citiem trigeriem.

Tāpēc mūsu vienkāršajā piemērā mēs atveram servera trigera konfigurācijas formu un iestatām, ka tas ir atkarīgs no attiecīgā maršrutētāja trigera. Ja šāda atkarība ir iestatīta, servera trigeris nemainīs savu stāvokli, kamēr trigeris, no kura tas ir atkarīgs, atrodas stāvoklī 'PROBLEM' - tādējādi netiks veiktas nekādas atkarīgās darbības un netiks nosūtīti paziņojumi.

Ja gan serveris, gan maršrutētājs nedarbojas un ir iestatīta atkarība starp trigeri "serveris nedarbojas" un trigeri "maršrutētājs nedarbojas", Zabbix neveiks darbības atkarīgajam trigerim.

Kamēr vecākais trigeris atrodas stāvoklī PROBLEM, tā atkarīgie trigeri var ziņot par vērtībām, kurām nevar uzticēties. Tāpēc atkarīgie trigeri netiks atkārtoti novērtēti, līdz vecākais trigeris (iepriekš minētajā piemērā - maršrutētājs):

- atgriežas no stāvokļa 'PROBLEM' stāvoklī 'OK';
- maina savu stāvokli no 'PROBLEM' uz 'UNKNOWN';
- tiek manuāli aizvērts, izmantojot korelāciju vai ar datuma un laika un/vai nodata() funkciju palīdzību;
- 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 atkārtoti novērtēts tikai tad, kad par to tiks saņemta jauna metrika. Tas nozīmē, ka atkarīgais trigeris var netikt atjaunināts nekavējoties.

Turklāt:

- Trigeru atkarību var pievienot no jebkura hosta trigera jebkuram cita hosta trigerim, ja vien tā nerada ciklisku atkarību.
- Trigeru atkarību var pievienot no vienas veidnes citai. 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, savukārt veidni B var piesaistīt hostam (vai citai veidnei) atsevišķi.
- Trigeru atkarību var pievienot no veidnes trigera hosta trigerim. Šādā gadījumā, piesaistot šādu veidni hostam, tiks izveidots hosta trigeris, kas būs atkarīgs no tās pašas trigera veidnes, no kuras bija atkarīgs trigeris. Tas ļauj, piemēram, izveidot veidni, kurā daži trigeri ir atkarīgi no maršrutētāja (hosta) trigeriem. Visi ar šo veidni saistītie hosti būs atkarīgi no konkrētā maršrutētāja.
- Trigeru atkarību nevar pievienot no hosta trigera veidnes trigerim.
- Trigeru atkarību var pievienot no trigera prototipa citam trigera prototipam (tās pašas zema līmeņa atklāšanas kārtulas ietvaros) vai reālam trigerim. Trigera prototips nevar būt atkarīgs no trigera prototipa citā ZLL kārtulā vai no trigera, kas izveidots no trigera prototipa. H ost a trigera prototips nevar būt atkarīgs no veidnes trigera.

Konfigurēšana

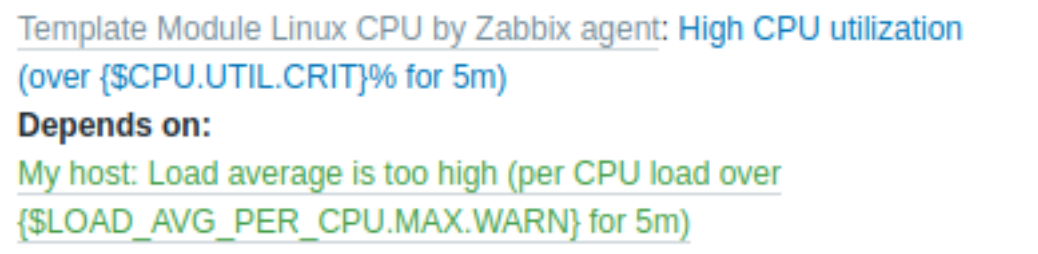
Lai definētu atkarību:

1. Atveriet cilni *Dependencies* trigera konfigurācijas veidlapā.
2. Sadaļā *Dependencies* noklikšķiniet uz *Add* un atlasiet vienu vai vairākus trigerus, no kuriem trigeris būs atkarīgs.



3. Noklikšķiniet uz *Update*.

Tagad trigeru sarakstā ir redzama tā atkarība.



Vairāku atkarību piemērs

Piemēram, hosts atrodas aiz Router2, bet Router2 atrodas aiz Router1.

```
Zabbix - Router1 - Router2 - Host
```

Ja Router1 nedarbojas, tad, protams, arī hosts un Router2 nav sasniedzami, taču trīs paziņojumu saņemšana par to, ka hosts, Router1 un Router2 visi nedarbojas, ir pārmērīga.

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 trigera atkarības. Ja tādas tiek atrastas un viens no šiem trigeriem ir **Problem** stāvoklī, tad trigera statuss netiks mainīts, darbības netiks izpildītas un paziņojumi netiks nosūtīti.

Zabbix veic šo pārbaudi rekursīvi. Ja Router1 vai Router2 nav sasniedzams, hosts trigera statuss netiks atjaunināts.

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.

Vizualizācija

Pārskats

Vizualizācija nodrošina dažādus veidus, kā attēlot Zabbix savāktos uzraudzības datus. Grafiki parāda tendences un metriku, tīkla kartes attēlo infrastruktūras topoloģiju, bet paneļi apvieno pielāgojamus logrīkus, lai nodrošinātu visaptverošus uzraudzības skatus.

Vizualizācijas rīki

- Grafiki - skatiet vienumu datus kā vienkāršus automātiski ģenerētus grafikus, izveidojiet pielāgotus grafikus vai salīdziniet vairākus vienumus ad hoc grafikos.
- Tīkla kartes - izveidojiet tīkla infrastruktūras vizuālus attēlojumus ar pielāgojamiem elementiem un indikatoriem.
- Paneļi - pielāgojiet izkārtojumus, apvienojot dažādus paneļu logrīkus, kas vizualizē datus dažādos formātos.

Grafiki

Pārskats

Kad Zabbix ieplūst liels datu apjoms, lietotājiem ir daudz ērtāk aplūkot vizuālu notiekošā attēlojumu, nevis tikai skaitļus.

Šeit noder grafiki. Grafiki ļauj vienā mirklī uztvert datu plūsmu, sasaistīt problēmas, noskaidrot, kad kaut kas sākās, vai ilustrēt, kad kaut kas varētu kļūt par problēmu.

Zabbix lietotājiem nodrošina:

- Iebūvētus vienkāršus grafikus par viena vienuma datiem.
- Iespēju izveidot sarežģītākus pielāgotus grafikus.
- Ātru piekļuvi vairāku vienumu salīdzinājumam ad hoc grafikos.
- Mūsdienīgus pielāgojamus vektorgrafikus grafikus un sektoru diagrammas.

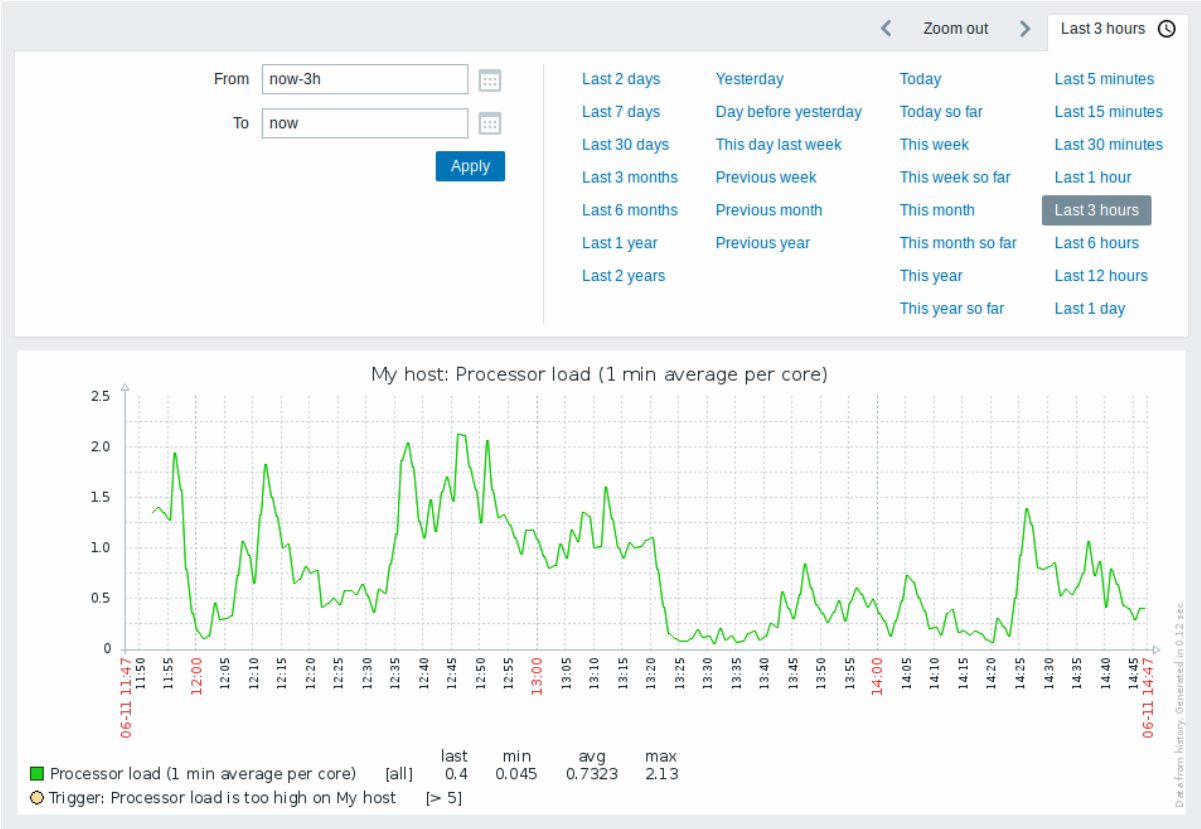
Vienkāršie grafiki

Pārskats

Vienkāršie grafiki ir paredzēti vienumu apkopoto datu vizualizācijai.

Lai skatītu vienkāršos grafikus, lietotājam nav jāveic nekāda konfigurēšana. Zabbix tos nodrošina bez maksas.

Vienkārši dodieties uz *Monitoring* > *Latest data* un noklikšķiniet uz attiecīgā vienuma saites *Graph*, lai tiktu parādīts grafiks.



Note:
Vienkāršie grafiki tiek nodrošināti visiem skaitliskajiem vienumiem. Teksta vienumiem sadaļā *Monitoring* > *Latest data* ir pieejama saite uz *History*.

Laika perioda atlasītājs

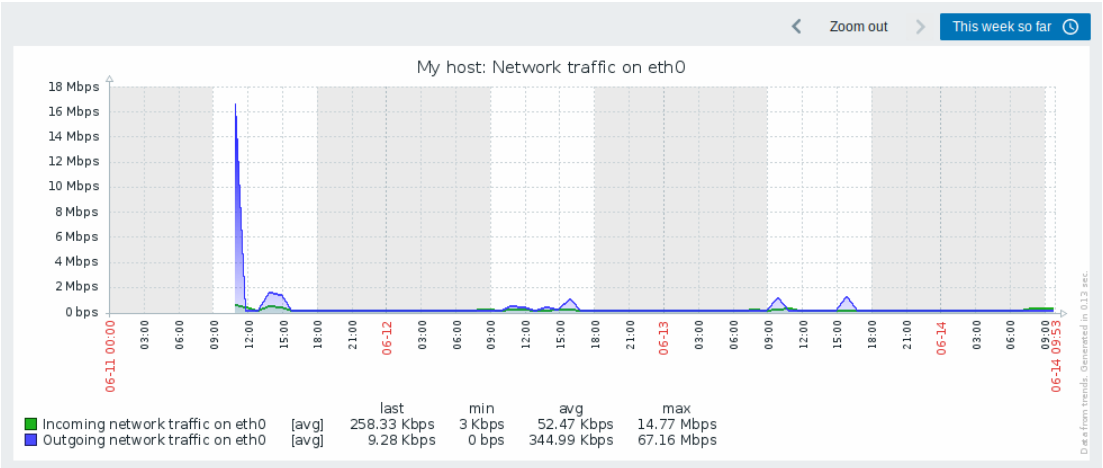
Laika perioda atlasītājs virs grafika ļauj ar vienu peles klikšķi atlasīt bieži nepieciešamus periodus. Plašāku informāciju skatiet sadaļā Laika perioda un hosts atlasītāji.

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 vismaz vienai vērtībai ir pieejams viens horizontāls pikselis.

Datiem, kas attēlo garāku periodu, tiek zīmētas **trīs līnijas** — tumši zaļa līnija rāda vidējo vērtību, bet 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 (izmantojot noklusējuma lietotāja saskarnes tēmu *Original blue*).

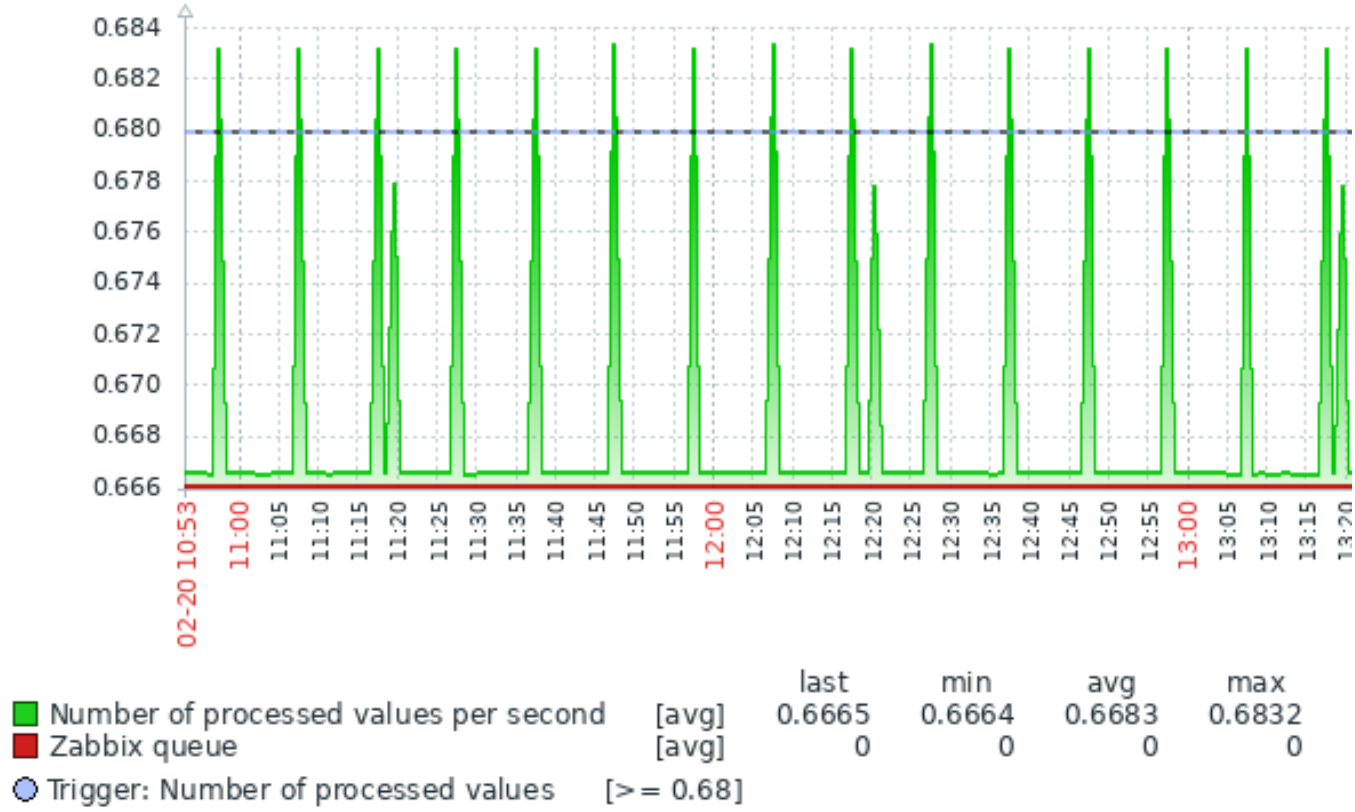


Darba laiks vienmēr tiek attēlots vienkāršajos grafikos, savukārt tā attēlošana pielāgotajos grafikos ir lietotāja izvēle.

Darba laiks netiek attēlots, ja grafikā redzams periods, kas pārsniedz 3 mēnešus.

Trigeru līnijas

Vienkāršie trigeri tiek attēloti kā līnijas ar melniem pārtraukumiem virs trigera svarīguma krāsas — pievērsiet uzmanību zilajai līnijai grafikā un trigera informācijai, kas parādīta leģendā. Grafikā var attēlot līdz 3 trigeru līnijām; ja trigeru ir vairāk, prioritāte tiek piešķirta trigeriem ar zemāku svarīgumu. Trigeri vienmēr tiek attēloti vienkāršajos grafikos, savukārt to attēlošana pielāgotajos grafikos ir lietotāja izvēle.



Ģenerēšana no vēstures/tendencēm

Grafikus var zīmēt, pamatojoties 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 tiek iegūti 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 iegūti no tendencēm.
- Datu blīvums grafikā. Ja horizontālā grafika pikselī attēlojamo sekunžu skaits pārsniedz 3600/16, tiek parādīti tendenču dati (pat ja vienuma vēsture par to pašu periodu joprojām ir pieejama).
- Ja tendences ir atspējotas, grafika izveidei tiek izmantota vienuma vēsture, ja tā ir pieejama par attiecīgo periodu.

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

Augšējā labajā stūrī esošais nolaižamais saraksts ļauj pārslēgties no vienkāršās diagrammas uz *Vērtību/500 jaunāko vērtību* sarakstiem. Tas var būt noderīgi, lai skatītu skaitliskās vērtības, kas veido diagrammu.

Šeit attēlotās vērtības ir neapstrādātas, t. i., netiek izmantotas ne mērvienības, ne vērtību pēcapstrāde. Tomēr tiek lietota vērtību kartēšana.

Zināmās problēmas

Skatiet zināmās problēmas attiecībā uz grafikiem.

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.

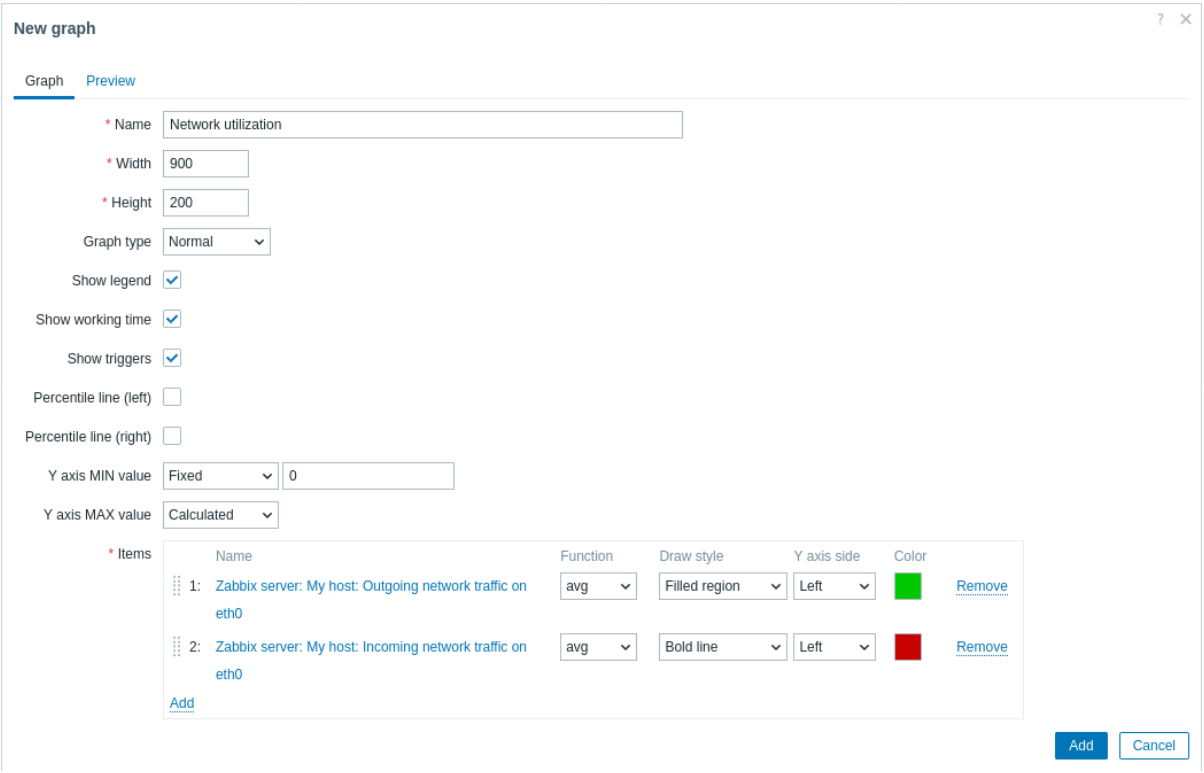
Pielāgotie grafiki tiek konfigurēti manuāli.

Tos var izveidot vienam hostam, vairākiem hostiem vai vienai veidnei.

Pielāgotu grafiku konfigurēšana

Lai izveidotu pielāgotu grafiku, rīkojieties šādi:

1. Dodieties uz *Datu vākšana* > *Hosts* (vai *Veidnes*).
2. Rindā blakus vajadzīgajam hostam vai veidnei noklikšķiniet uz *Grafiki*.
3. Grafiku ekrānā noklikšķiniet uz *Izveidot grafiku*.
4. Rediģējiet grafika atribūtus.



Visi obligātie ievades lauki ir atzīmēti ar sarkanu zvaigznīti.

Grafika atribūti:

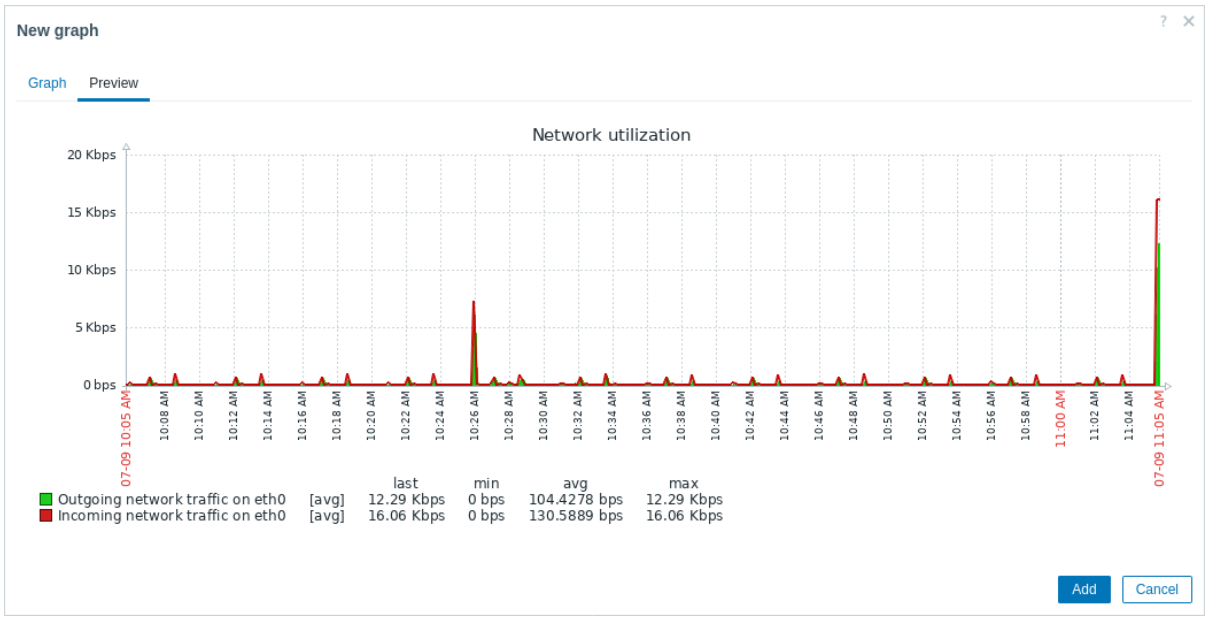
Parameter	Description
Name	Unikāls grafika nosaukums. Šajā laukā tiek atbalstītas izteiksmes macros, taču tikai ar funkcijām `avg`, `last`, `min` un `max`, izmantojot laiku kā parametru (piemēram, `{?avg(/host/key,1h)}`). {HOST.HOST<1-9>} macros tiek atbalstīti šī macro ietvaros, atsaucoties uz pirmo, otro, trešo utt. hostu grafikā, piemēram, `{?avg(/{HOST.HOST2}/key,1h)}`. Ņemiet vērā, ka atsaukšanās uz pirmo hostu ar šo macro ir lieka, jo uz pirmo hostu var atsaukties netieši, piemēram, `{?avg(//key,1h)}`.
Width	Grafika platums pikseļos (tikai priekšskatījumam un sektoru/izvērsto sektoru grafikiem).
Height	Grafika augstums pikseļos.

Parameter	Description
Graph type	Grafika tips: **Normal** - parasts grafiks, vērtības tiek attēlotas kā līnijas **Stacked** - sakrauts grafiks, tiek attēlotas aizpildītas zonas **Pie** - sektoru grafiks **Exploded** - ”izvērsts” sektoru grafiks, daļas tiek attēlotas kā ”izgrieztas” no sektora
Show legend	Atzīmējot šo izvēles rūtiņu, tiks parādīta grafika leģenda.
Show working time	Ja šī opcija ir atlasīta, nestrādājošās stundas tiks parādītas ar pelēku fonu. Šis parametrs nav pieejams sektoru un izvērsto sektoru grafikiem.
Show triggers	Ja šī opcija ir atlasīta, vienkāršie trigeri tiks parādīti kā līnijas ar melnām svītrām virs trigeru nopietnības krāsas. Šis parametrs nav pieejams sektoru un izvērsto sektoru grafikiem.
Percentile line (left)	Parādīt procentili kreisajai Y asij. Ja, piemēram, ir iestatīta 95% procentile, tad procentiles līnija atradīsies līmenī, zem kura ir 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 ir 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 izvērsto 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 izvērsto sektoru grafikiem.
3D view	Iespējot 3D stilu. Tikai sektoru un izvērsto 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 attēlošana, krāsa).
Sort order (0→100)	Zīmēšanas secība. Vispirms tiks apstrādāts 0. Var izmantot, lai zīmētu līnijas vai apgabalus aiz cita (vai priekšā citam). Varat vilkt un nomest vienumus, izmantojot ikonu rindas sākumā, lai iestatītu kārtošanas secību vai noteiktu, kurš vienums tiek attēlots priekšā citam.
Name	Atlasītā vienuma nosaukums tiek parādīts kā saite. Noklikšķinot uz saites, tiek atvērts citu pieejamo vienumu saraksts.
Type	Tips (pieejams tikai sektoru un izvērsto sektoru grafikiem): **Simple** - vienuma vērtība tiek attēlota proporcionāli sektorā **Graph sum** - vienuma vērtība attēlo visu sektoru Ņemiet vērā, ka ”graph sum” vienuma krāsojums būs redzams tikai tiktāl, cik to neaizņem ”proportional” vienumi.
Function	Atlasiet, kuras vērtības tiks attēlotas, ja vienumam uz vienu vertikālo grafika pikseli ir vairāk nekā viena vērtība: **all** - attēlot visas iespējamās vērtības (minimumu, maksimumu, vidējo) grafikā. Ņemiet vērā, ka īsākiem periodiem šim iestatījumam nav ietekmes; tikai garākiem periodiem, kad datu blīvums vertikālajā grafika pikselī palielinās, ’all’ sāk attēlot minimumu, maksimumu un vidējo vērtību. 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

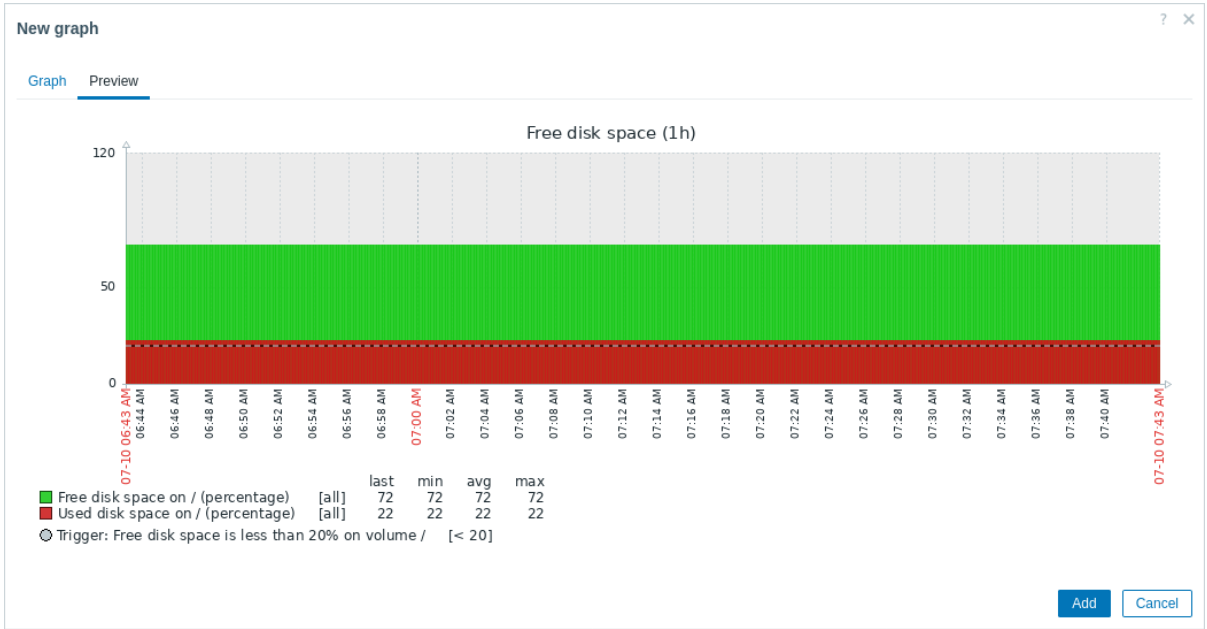
Parameter	Description
Draw style	Atlasiet zīmēšanas stilu (pieejams tikai parastajiem grafikiem; sakrautiem grafikiem vienmēr tiek izmantots aizpildīts apgabals), ko piemērot vienuma datiem - *Line*, *Bold line*, *Filled region*, *Dot*, *Dashed line*, *Gradient line*.
Y axis side	Atlasiet Y ass pusi, kurā rādīt vienuma datus - *Left*, *Right*.
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.

Ad hoc grafiki

Pārskats

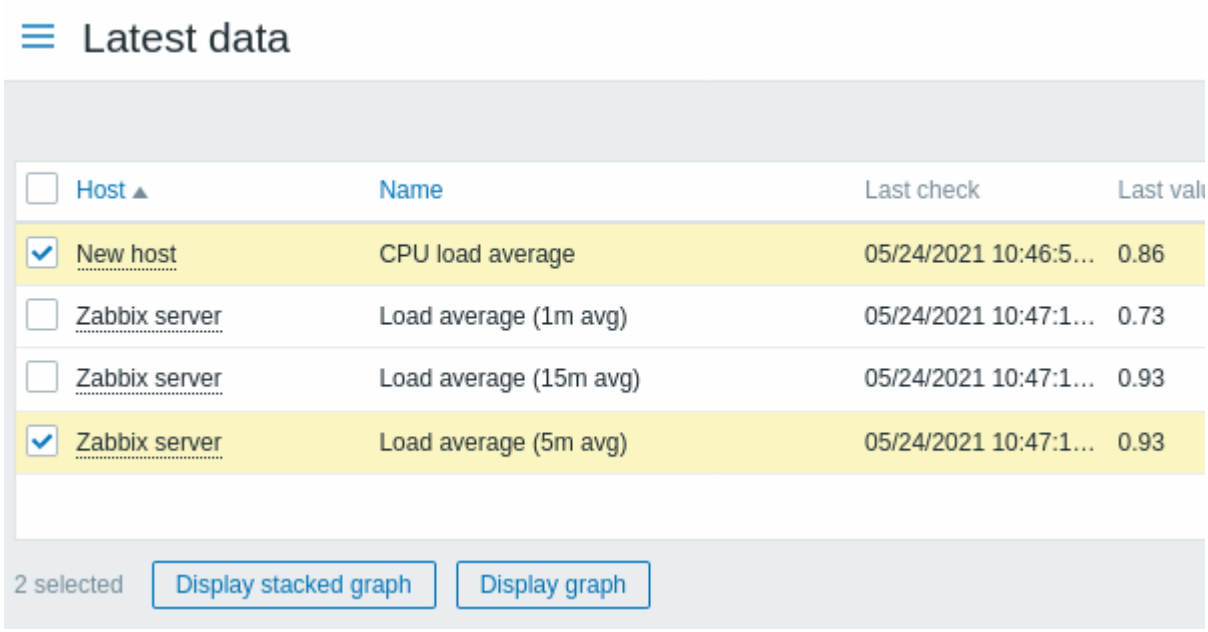
Lai gan vienkārša diagramma ir noderīga datu skatīšanai no viena vienuma un pielāgotas diagrammas piedāvā pielāgošanas iespējas, neviena no tām neļauj ātri izveidot salīdzināšanas diagrammu vairākiem vienumiem ar minimālu piepūli un uzturēšanu.

Lai risinātu šo problēmu, ir iespējams ļoti ātri izveidot īslaicīgas diagrammas vairākiem vienumiem.

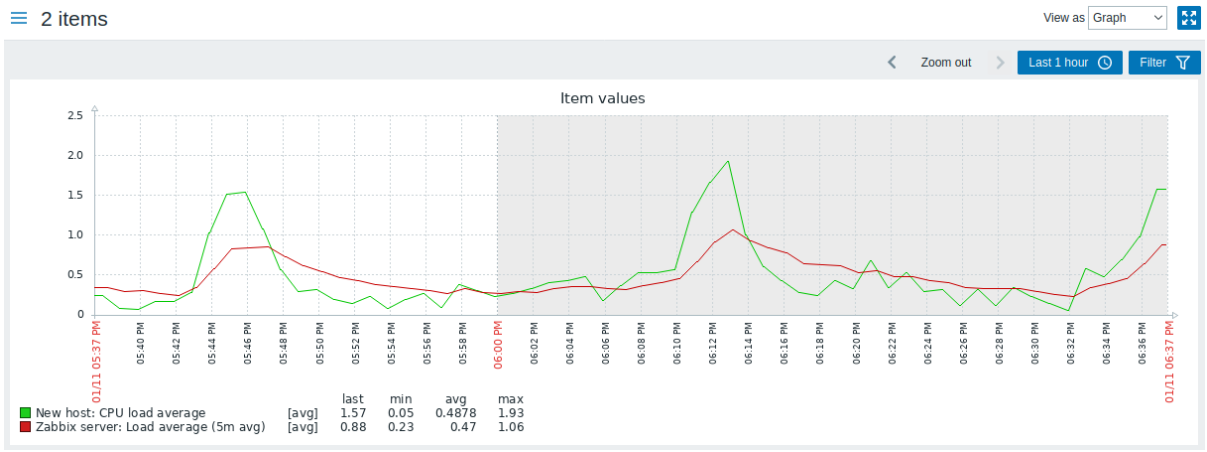
Konfigurācija

Lai izveidotu ad hoc grafiku, rīkojieties šādi:

1. Dodieties uz *Monitoring* > *Latest data*.
2. Izmantojiet filtru, lai parādītu vienumus, kurus vēlaties.
3. Atzīmējiet izvēles rūtiņas tiem vienumiem, kurus vēlaties attēlot grafikā.
4. Noklikšķiniet uz pogām *Display stacked graph* vai *Display graph*.

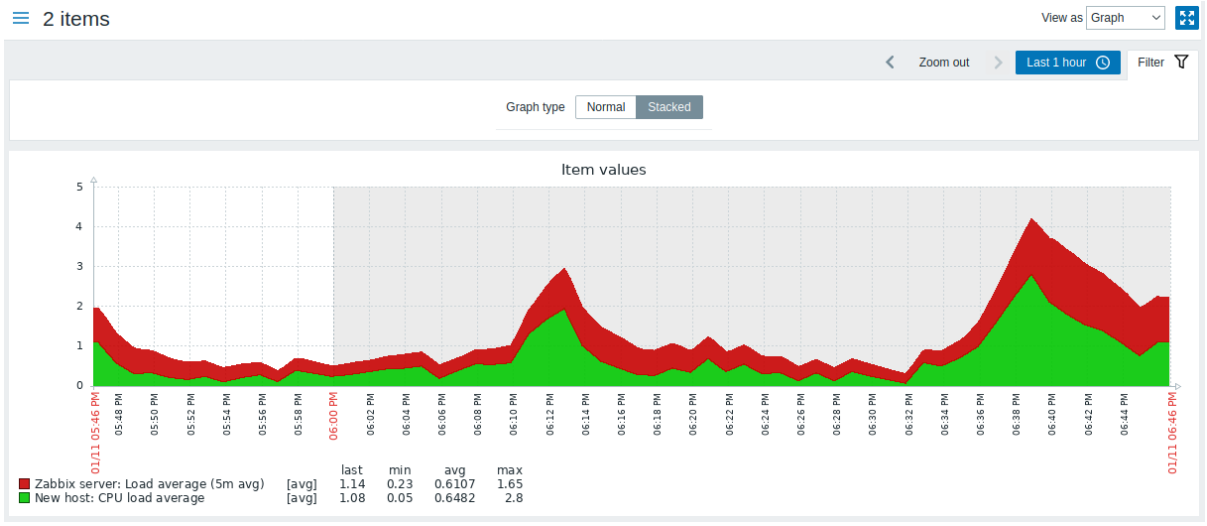


Jūsu grafiks tiek izveidots nekavējoties:



Ņemiet vērā, ka, lai grafikā netiktu rādīts pārāk daudz līniju, tiek attēlota tikai katra vienuma vidējā vērtība (min/max vērtību līnijas netiek rādītas). Trigeri un trigeru informācija grafikā netiek rādīta.

Izveidotā grafika logā ir pieejams laika perioda atlasītājs, kā arī iespēja pārslēgties no ”parastā” līniju grafika uz sakrauto grafiku (un atpakaļ).



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.

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.

Kartes elementu attēlošanai tiek izmantotas ikonas. Var definēt informāciju, kas tiks parādīta kopā ar ikonām, un iestatīt, ka nesenās problēmas tiek attēlotas īpašā veidā. Var savienot ikonas un definēt informāciju, kas tiks parādīta uz saitēm.

Var pievienot pielāgotus URL, kas būs pieejami, noklikšķinot uz ikonām. Tādējādi var saistīt hosta ikonu ar hosta īpašībām vai kartes ikonu ar citu karti.

Problēmu skaits kartēs tiek rādīts tikai cēloņproblēmām.

Kartes tiek pārvaldītas sadaļā *Monitoring* > *Maps*, kur tās var konfigurēt, pārvaldīt un skatīt. Monitoringa skatā var 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.

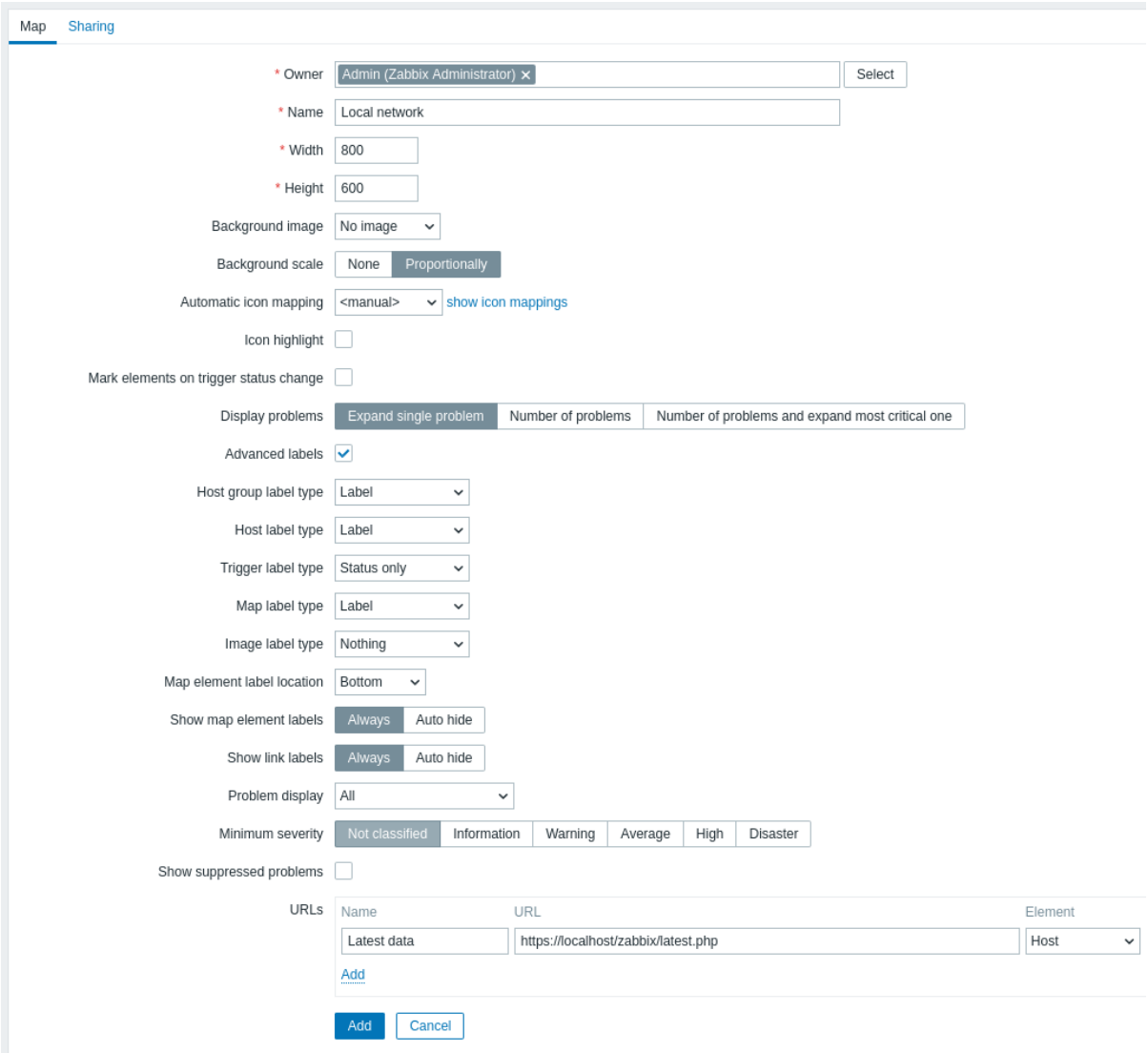
Kartes izveide

Lai izveidotu karti, rīkojieties, kā norādīts tālāk:

1. Atveriet *Monitoring* > *Maps*.
2. Atveriet skatu ar visām kartēm.
3. Noklikšķiniet uz *Create map*.
4. Konfigurējiet kartes informāciju.

Varat arī izmantot pogu *Clone* esošas kartes konfigurācijas veidlapā, lai izveidotu jaunu karti. Šai kartei būs visi esošās kartes rekvizīti, tostarp vispārīgie izkārtojuma atribūti, kā arī esošās kartes elementi.

Cilnē **Map** ir ietverti vispārīgie kartes atribūti:



Visi obligātie ievades lauki ir atzīmēti ar sarkanu zvaigznīti.

Vispārīgie kartes atribūti:

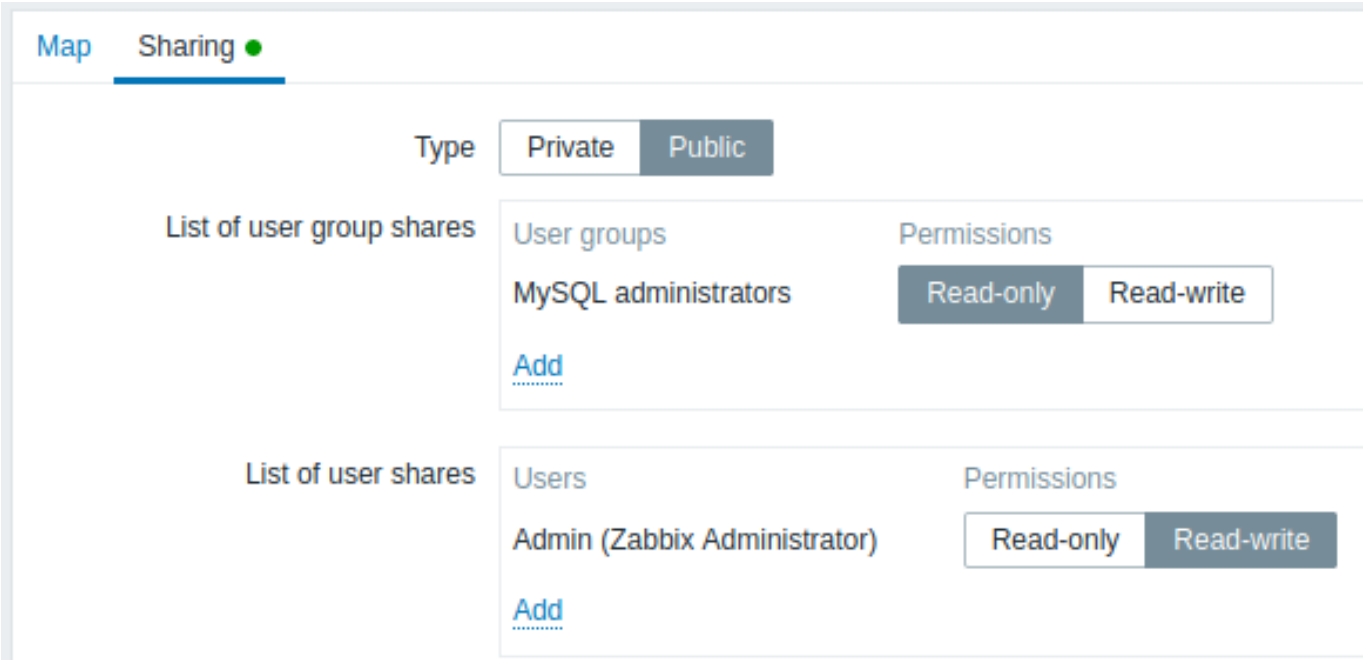
Parametrs	Apraksts
Owner	Kartes īpašnieka vārds.
Name	Unikāls kartes nosaukums.
Width	Kartes platums pikseļos.
Height	Kartes augstums pikseļos.
Background image	Izmantojiet fona attēlu: **No image** - fona attēls netiek izmantots (balts fons) **Image** - atlasītais attēls tiek izmantots kā fona attēls. Varat izmantot ģeogrāfisku karti vai jebkuru citu attēlu, lai uzlabotu karti.

Parametrs	Apraksts
Background scale	Izmantojiet fona mērogošanu: **None** - mērogošana netiek izmantota; **Proportionally** - aizpilda visu kartes fonu, saglabājot attēla proporcijas (noklusējums).
Automatic icon mapping	Varat iestatīt automātisku ikonu kartējumu, kas konfigurēts sadaļā *Administration* > *General* > *Icon mapping*. Ikonu kartējums ļauj noteiktas ikonas sasaistīt ar noteiktiem hosts inventarizācijas laukiem.
Icon highlight	Ja atzīmēsiet šo izvēles rūtiņu, kartes elementi tiks izcelti. Elementi ar aktīvu trigeri saņems apaļu fonu tādā pašā krāsā kā trigeris ar augstāko nopietnības līmeni. Turklāt ap apli tiks parādīta bieza zaļa līnija, ja visas problēmas ir apstiprinātas. Elementi ar statusu "disabled" vai "in maintenance" saņems attiecīgi pelēku vai oranžu kvadrātveida fonu. Skatiet arī: Kartes skatīšana
Mark elements on trigger status change	Nesenās trigeru stāvokļa izmaiņas (nesena problēma vai atrisinājums) tiks izceltas ar marķieriem (uz iekšu vērstiem sarkaniem trijstūriem) tajās trijās elementa ikonas malās, kurās nav etiķetes. Marķieri tiek rādīti 30 minūtes.
Display problems	Atlasiet, kā problēmas tiek rādītas kopā ar kartes elementu: **Expand single problem** - ja ir tikai viena problēma, tiek rādīts problēmas nosaukums. Pretējā gadījumā tiek rādīts kopējais problēmu skaits. **Number of problems** - tiek rādīts kopējais problēmu skaits **Number of problems and expand most critical one** - tiek rādīts viskritiskākās problēmas nosaukums un kopējais problēmu skaits. "Viskritiskākā" problēma tiek noteikta, pamatojoties uz problēmas nopietnību un, ja tā ir vienāda, uz problēmas notikuma ID (vispirms tiek rādīts notikums ar lielāku ID jeb jaunākā problēma). *Trigeru kartes elementam* tā tiek noteikta, pamatojoties uz problēmas nopietnību un, ja tā ir vienāda, uz trigera pozīciju trigeru sarakstā. Ja ir vairākas viena trigera problēmas, tiks rādīta jaunākā.
Advanced labels	Ja atzīmēsiet šo izvēles rūtiņu, varēsiet definēt atsevišķus etiķešu tipus dažādiem elementu tipiem.
Host group label type	Kartes elementam izmantotais etiķetes tips: **Label** - kartes elementa etiķete **IP address** - IP adrese **Element name** - elementa nosaukums (piemēram, hosts 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 atrašanās vieta attiecībā pret kartes elementu: **Bottom** - zem kartes elementa **Left** - pa kreisi **Right** - pa labi **Top** - virs kartes elementa
Show map element labels	Atlasiet, kā rādīt kartes elementu etiķetes: **Always** - vienmēr rādīt (noklusējums); **Auto hide** - paslēpt etiķeti, ja kursors nav novietots virs tās vai tā nav atlasīta.
Show link labels	Atlasiet, kā rādīt saišu etiķetes: **Always** - vienmēr rādīt (noklusējums); **Auto hide** - paslēpt etiķeti, ja kursors nav novietots virs tās vai tā nav atlasīta.
Problem display	Rādīt problēmu skaitu kā: **All** - tiek rādīts pilns problēmu skaits **Separated** - neapstiprināto problēmu skaits tiek rādīts atsevišķi kā daļa no kopējā problēmu skaita **Unacknowledged only** - tiek rādīts tikai neapstiprināto problēmu skaits
Minimum severity	Problēmas, kuru nopietnība ir zemāka par atlasīto minimālo līmeni, kartē netiks rādītas. Piemēram, ja ir atlasīts *Warning*, izmaiņas ar trigeriem līmenī *Information* un *Not classified* kartē netiks atspoguļotas.
Show suppressed problems	Atzīmējiet izvēles rūtiņu, lai rādītu problēmas, kuras citādi tiktu apslēptas (netiktu rādītas) hosts uzturēšanas dēļ.

Parametrs	Apraksts
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 rādīti kā saites, kad lietotājs kartes skatīšanas režīmā noklikšķinās uz elementa. Kartes URL nosaukumos un vērtībās var izmantot makro. Pilnu sarakstu skatiet sadaļā atbalstītie makro un meklējiet "map URL names and values".

Koplietošana

Cilne **Koplietošana** satur kartes tipu, kā arī koplietošanas opcijas (lietotāju grupas, lietotāji) privātajām kartēm:



Parameter	Description
Type	Atlasiet kartes tipu: **Private** - karte ir redzama tikai atlasītajām lietotāju grupām un lietotājiem **Public** - karte ir redzama visiem
List of user group shares	Atlasiet lietotāju grupas, kurām karte ir pieejama. Varat atļaut tikai lasīšanas vai lasīšanas un rakstīšanas piekļuvi.
List of user shares	Atlasiet lietotājus, kuriem karte ir pieejama. Varat atļaut tikai lasīšanas vai lasīšanas un rakstīšanas piekļuvi.

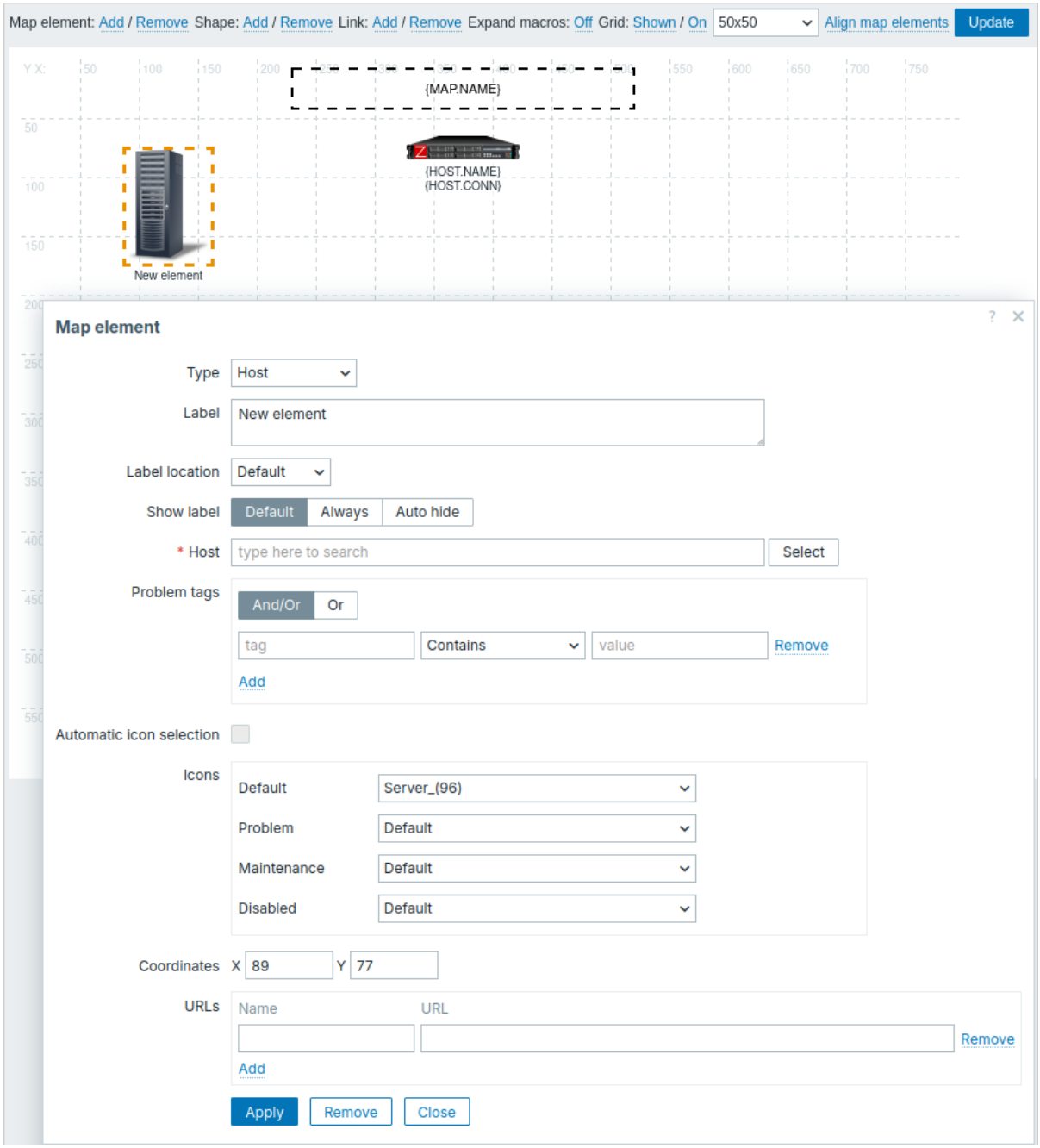
Kad noklikšķināt *Add*, lai saglabātu šo karti, jūs esat izveidojis tukšu karti ar nosaukumu, izmēriem un noteiktām preferencēm. Tagad jums jāpievieno daži elementi. Lai to izdarītu, kartes sarakstā noklikšķiniet *Edit*, lai atvērtu rediģējamo apgabalu.

Elementu pievienošana

Lai pievienotu elementu, noklikšķiniet uz *Add* blakus `Map element`. Jaunais elements parādīsies kartes augšējā kreisajā stūrī. Velciet un nometiet to jebkur, kur vēlaties.

Ņemiet vērā, ka, ja režģa opcija ir "On", elementi vienmēr tiks izlīdzināti pēc režģa (nolaižamajā sarakstā varat izvēlēties dažādus režģa izmērus, kā arī paslēpt/parādīt režģi). Ja vēlaties izvietot elementus jebkur bez izlīdzināšanas, pārslēdziet opciju uz "Off". (Vēlāk nejauši izvietotus elementus varat izlīdzināt pēc režģa, noklikšķinot uz *Align map elements*.)

Tagad, kad jums jau ir daži elementi, iespējams, vēlēsieties tos sākt atšķirt, piešķirot tiem nosaukumus u. tml. Noklikšķinot uz elementa, tiek parādīta forma, kurā varat iestatīt elementa tipu, piešķirt nosaukumu, izvēlēties citu ikonu u. c.



Kartes elementa atribūti:

Parameter	Description
Type	Elementa tips: **Host** - ikona, kas attēlo visu atlasītā 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 to hostu trigeru statusu, kuri pieder atlasītajai grupai **Image** - ikona, kas nav saistīta ar nevienu resursu
Label	Ikonas etiķete, jebkura virkne. Var izmantot makro un vairākrindu virknes. Šajā laukā tiek atbalstīti izteiksmes makro, taču tikai ar funkcijām `avg`, `last`, `min` un `max`, izmantojot laiku kā parametru (piemēram, `{?avg(/host/key,1h)}`). Pilnu atbalstīto makro 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 rādīt (noklusējums); **Auto hide** - paslēpt etiķeti, ja uz tās nenovieto kursoru vai tā nav atlasīta.
Host	Atlasiet hostu; alternatīvi ievadiet hosta nosaukumu. Šis lauks atbalsta automātisko pabeigšanu, tāpēc, sākot rakstīt hosta nosaukumu, tiks parādīts atbilstošo hostu nolaižamais saraksts. Ritiniet uz leju, lai atlasītu vienu. Noklikšķiniet uz noņemšanas (x) ikonas, lai noņemtu atlasi. Šis lauks ir pieejams tikai elementiem ar tipu *Host*.
Map	Atlasiet karti; alternatīvi ievadiet kartes nosaukumu. Šis lauks atbalsta automātisko pabeigšanu, tāpēc, sākot rakstīt kartes nosaukumu, tiks parādīts atbilstošo karšu nolaižamais saraksts. Ritiniet uz leju, lai atlasītu vienu. Noklikšķiniet uz noņemšanas (x) ikonas, lai noņemtu atlasi. Šis lauks ir pieejams tikai elementiem ar tipu *Map*.
Triggers	Atlasiet vienu vai vairākus trigerus laukā *New triggers* zemāk un noklikšķiniet uz *Add*. Alternatīvi sāciet rakstīt trigera nosaukumu un atlasiet vienu no nolaižamajā sarakstā redzamajiem atbilstošajiem trigeriem. Noklikšķiniet uz noņemšanas (x) ikonas, lai noņemtu atlasi. Šis lauks ir pieejams tikai elementiem ar tipu *Trigger*. Atlasīto trigeru secību var mainīt, taču tikai vienas un tās pašas trigera nopietnības ietvaros. Vairāku trigeru atlase ietekmē arī {HOST.*} makro atrisināšanu gan rediģēšanas, gan skatīšanas režīmā. Rediģēšanas režīmā pirmie parādītie {HOST.*} makro tiks atrisināti atkarībā no pirmā sarakstā esošā trigera (pamatojoties uz trigera nopietnību). Skatīšanas režīms ir atkarīgs no parametra Display problems sadaļā General map attributes: - Ja ir izvēlēts režīms *Expand single problem*, pirmie parādītie {HOST.*} makro tiks atrisināti atkarībā no pēdējā konstatētā problēmas trigera (neatkarīgi no nopietnības) vai pirmā sarakstā esošā trigera (ja problēma nav konstatēta). - Ja ir izvēlēts režīms *Number of problems and expand most critical one*, pirmie parādītie {HOST.*} makro tiks atrisināti atkarībā no trigera nopietnības.
Host group	Atlasiet hostu grupu; alternatīvi ievadiet hostu grupu. Šis lauks atbalsta automātisko pabeigšanu, tāpēc, sākot rakstīt hostu grupas nosaukumu, tiks parādīts atbilstošo hostu grupu nolaižamais saraksts. Ritiniet uz leju, lai atlasītu vienu. Noklikšķiniet uz noņemšanas (x) ikonas, lai noņemtu atlasi. Šis lauks ir pieejams tikai elementiem ar tipu *Host group*.
Problem tags	Norādiet tagus, lai ierobežotu widgetā parādīto problēmu skaitu. Ir iespējams iekļaut, kā arī izslēgt konkrētus tagus un tagu vērtības. Var iestatīt vairākus nosacījumus. Tagu nosaukumu atbilstība vienmēr ir reģistrjutīga. Katrai 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, nav reģistrjutīga) **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, nav reģistrjutīga) Nosacījumiem ir pieejami divi aprēķina veidi: **And/Or** - jāizpildās visiem nosacījumiem, nosacījumi ar vienādu taga nosaukumu tiks grupēti ar Or nosacījumu **Or** - pietiek, ja izpildās viens nosacījums Šis lauks ir pieejams tikai elementiem ar tipu *Host* un *Host group*.
Automatic icon selection	Ja izvēles rūtiņa ir atzīmēta, ikonas kartējums tiks izmantots, lai noteiktu, kuru ikonu attēlot.
Icons	Varat izvēlēties, lai elementam tiktu attēlotas dažādas ikonas 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 konkrētam elementam paredzētus URL, kur: **Name** - URL etiķete; **URL** - URL (līdz 2048 rakstzīmēm). Skatīšanas režīmā, noklikšķinot uz elementa kartē, tie tiks parādīti kā saites. 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 makro. Pilnu sarakstu skatiet sadaļā supported macros un meklējiet ”map URL names and values”.

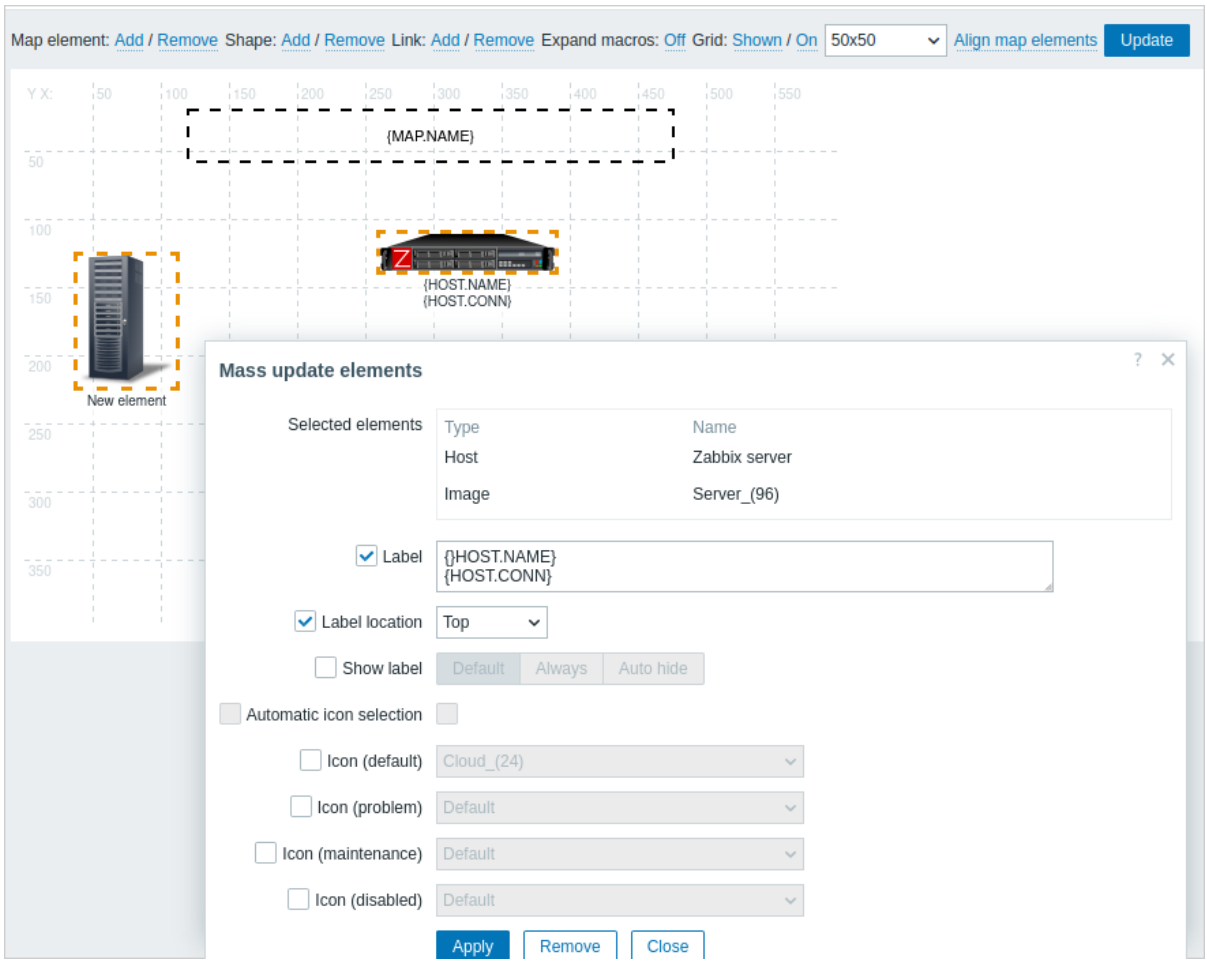
Attention:
Pievienotie elementi netiek saglabāti automātiski. Ja atstāsiet lapu, visas izmaiņas var tikt zaudētas.
Tāpēc ir ieteicams noklikšķināt uz pogas **Update** augšējā labajā stūrī. Kad tā ir nospiesta, izmaiņas tiek saglabātas neatkarīgi no tā, ko jūs izvēlēsieties nākamajā uznirstošajā logā.
Atlasītās režģa opcijas tiek saglabātas arī 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.

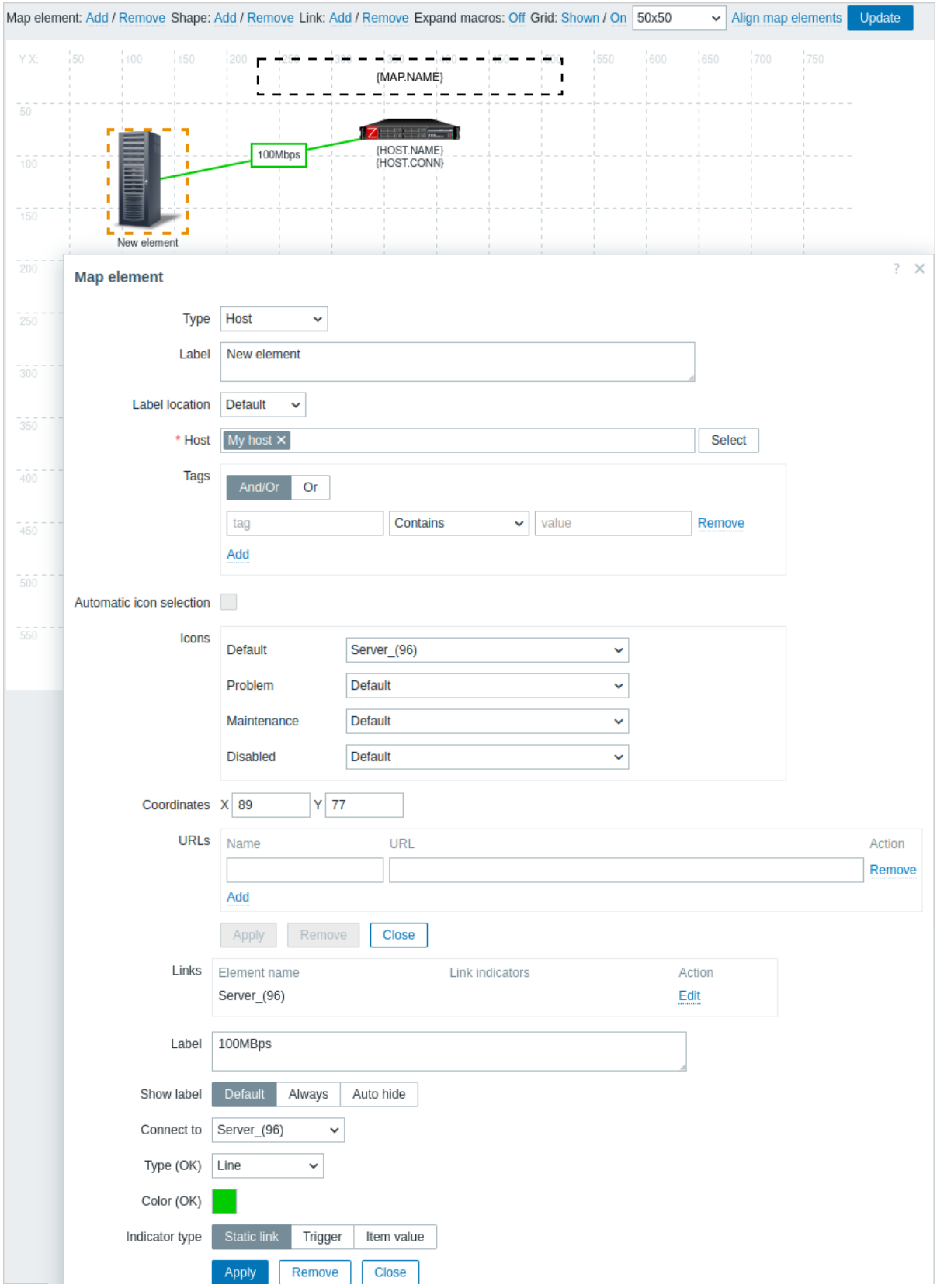
Kad atlasāt vairāk nekā vienu elementu, elementa īpašību forma pārslēdzas uz masveida atjaunināšanas režīmu, lai jūs varētu vienlaikus mainīt atlasīto elementu atribūtus. Lai to izdarītu, atzīmējiet atribūtu, izmantojot izvēles rūtiņu, un ievadiet tam jaunu vērtību. Šeit varat izmantot makrosus (piemēram, {HOST.NAME} elementa etiķetei).



Elementu sasaistīšana

Kad kartē esat izvietojis dažus elementus, ir pienācis laiks tos sākt sasaistīt. Lai sasaistītu divus elementus, vispirms tie ir jāatlasa. Kad elementi ir atlasīti, noklikšķiniet uz *Add* blakus Link.

Kad saite ir izveidota, atsevišķā elementa forma tagad satur papildu *Links* sadaļu. Noklikšķiniet uz *Edit*, lai rediģētu saites atribūtus.



Saites atribūti:

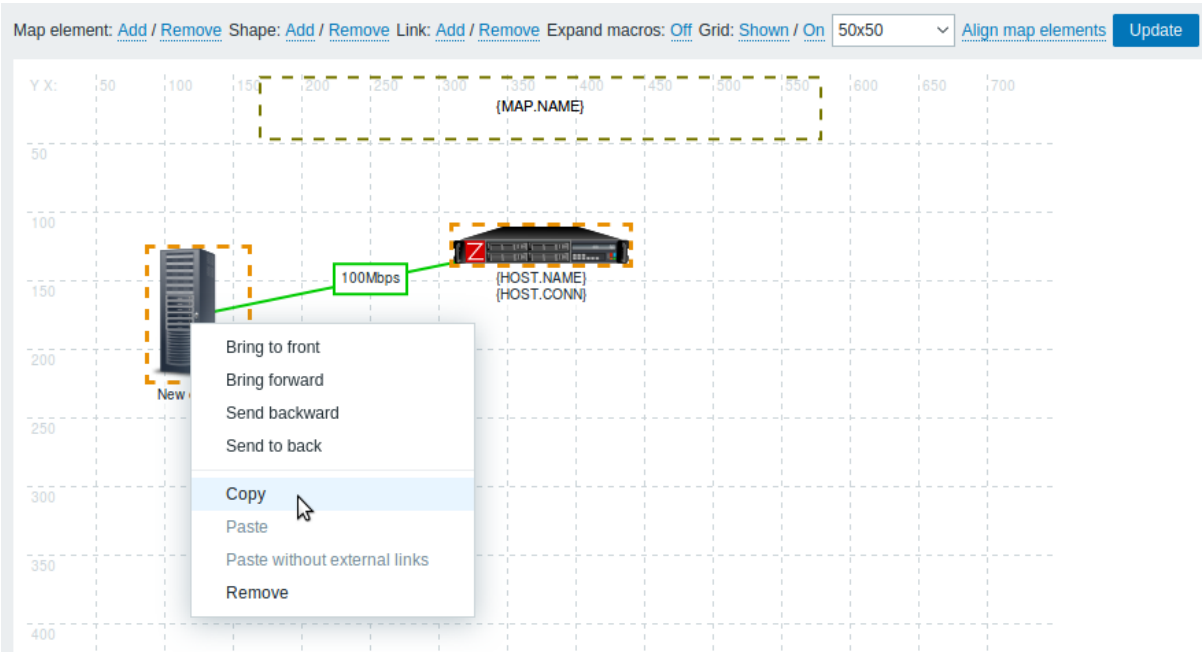
Parameter	Description
Label	Etiķete, kas tiks attēlota virs saites. Šajā laukā tiek atbalstītas izteiksmes macros, taču tikai ar funkcijām `avg`, `last`, `min` un `max`, izmantojot laiku kā parametru (piemēram, `{?avg(/host/key,1h)}`).

Parameter	Description
Show label	Izvēlieties, kā attēlot saites etiķeti: **Default** - izmantot *Show map element labels* iestatījumus no kartes konfigurācijas (noklusējums); **Always** - vienmēr rādīt (noklusējums); **Auto hide** - paslēpt etiķeti, ja uz tās netiek norādīts ar peli 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 krāsu parauga, lai izvēlētos citu krāsu.
Indicator type	Izvēlieties saites indikatora tipu: **Static link** - bez indikatoriem; **Trigger** - ļauj izmantot uz trigeriem balstītus saites indikatorus; **Item value** - ļauj izmantot uz vienuma vērtībām balstītus saites indikatorus.
Item	Izvēlieties vienumu. Šī vienuma vērtība ietekmēs saites stilu. Šis lauks ir pieejams, ja *Indicator type* ir izvēlēts *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, saitei tiks piemērots tā stils (atlasītā krāsa un līnijas tips). Ja ir iestatīts *Item value* un laukā *Item* ir atlasīts vienums, vienumu vērtības var piešķirt kā saites indikatorus. Kad vienuma vērtība sasniedz norādīto slieksni (skaitliskam vienumam) vai atbilst paraugam (teksta vienumam), saitei tiek piemērots tās stils (atlasītā krāsa un līnijas tips).

Elementu pārvietošana un kopēšana

Vairākus atlasītos elementus var **pārvietot** uz citu vietu kartē, noklikšķinot uz viena no atlasītajiem elementiem, turot nospiestu peles pogu un pārvietojot kursoru uz vēlamo atrašanās vietu.

Vienu vai vairākus elementus var **kopēt**, atlasot elementus, pēc tam ar peles labo pogu noklikšķinot uz atlasītā elementa un izvēlnē izvēloties *Copy*.

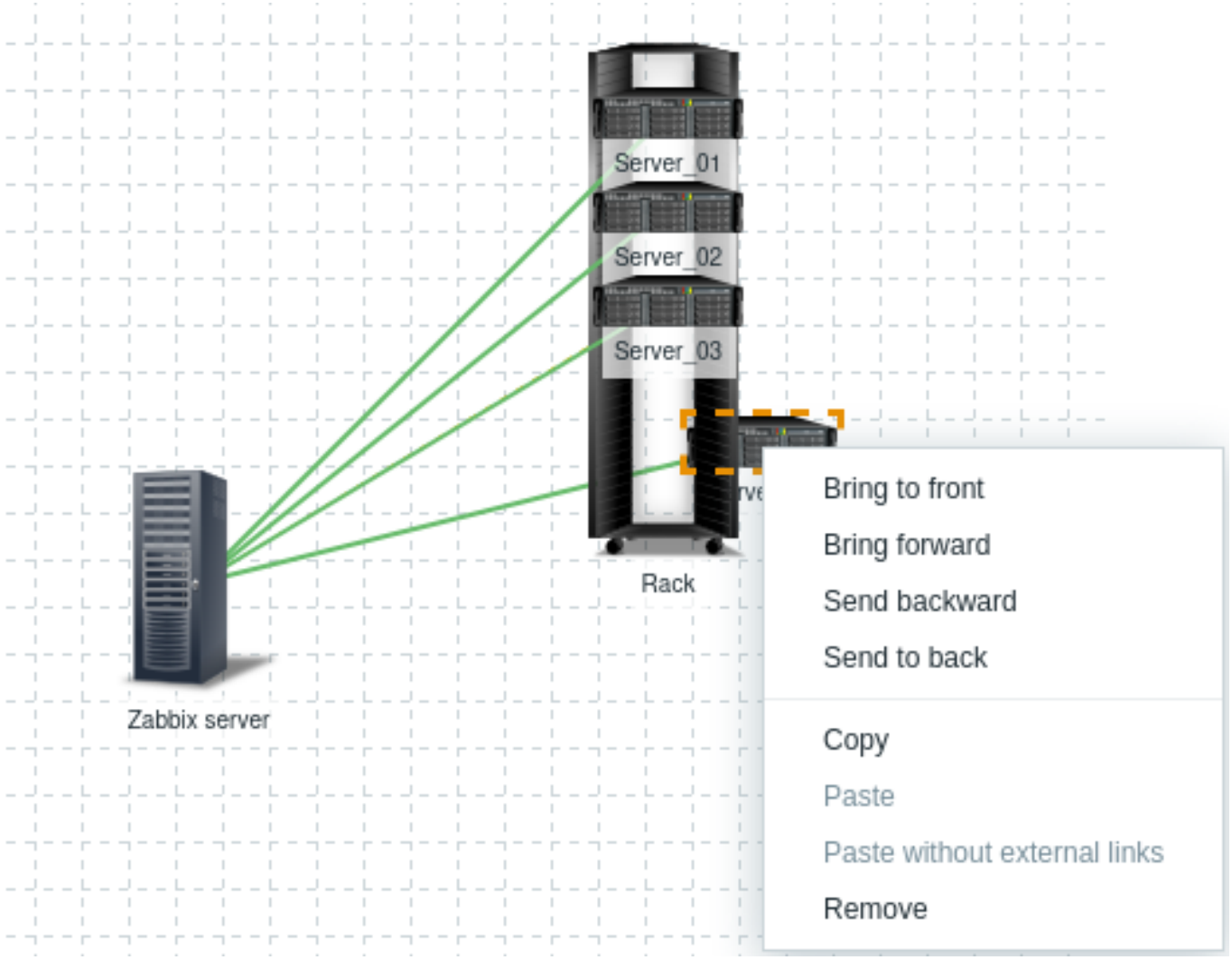


Lai ielīmētu elementus, ar peles labo pogu noklikšķiniet uz kartes apgabala un izvēlnē izvēlieties *Paste*. Opcija *Paste without external links* ielīmēs elementus, saglabājot tikai saites, kas ir starp atlasītajiem elementiem.

Kopēšana un ielīmēšana darbojas vienā un tajā pašā pārlūkprogrammas logā. Tastatūras īsinājumtaustiņi netiek atbalstīti.

Elementu kārtošana

Lai vienu elementu pārvietotu otra priekšā (vai otrādi), noklikšķiniet uz elementa ar peles labo pogu un atlasiet *Bring forward*/*Bring to front* vai *Send backward*/*Send to back*.

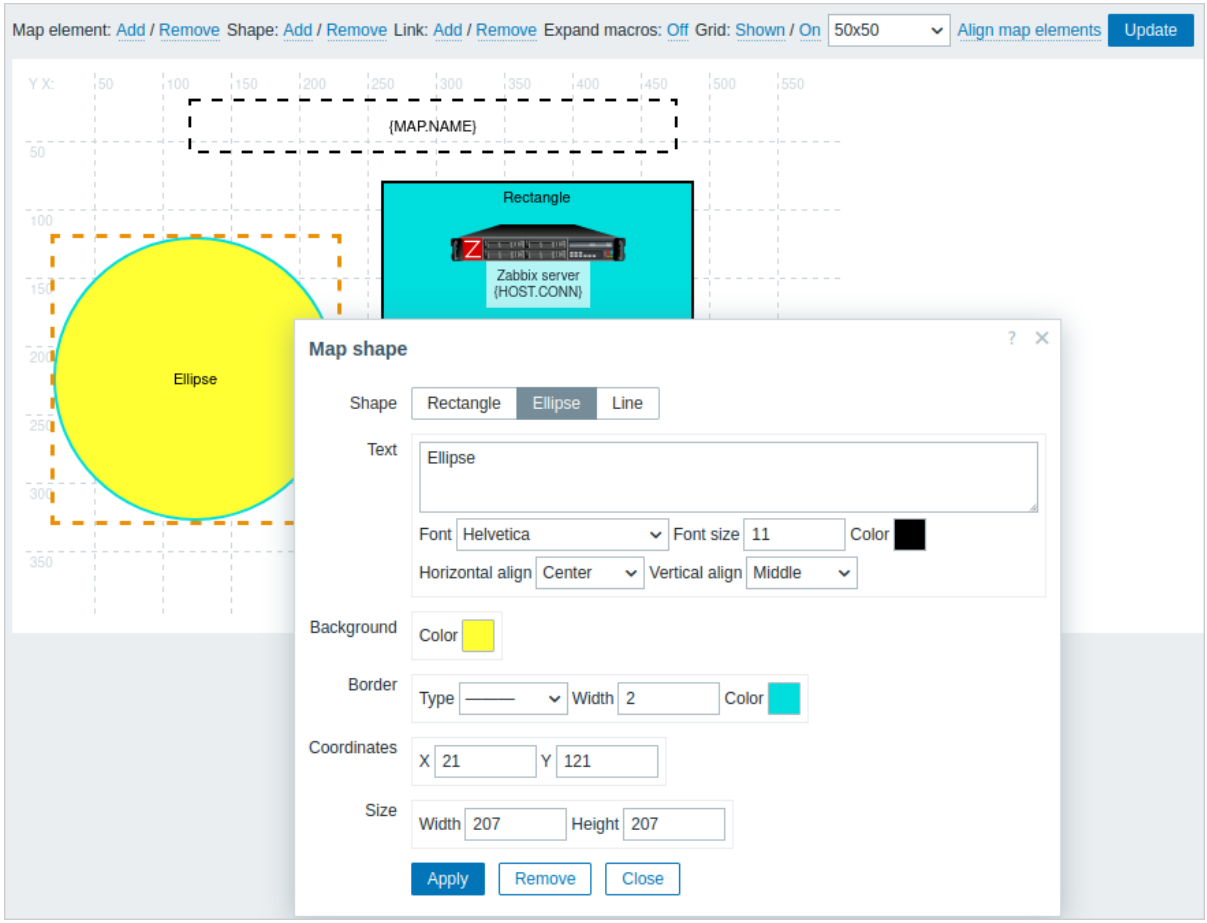


Formu pievienošana

Papildus kartes elementiem ir iespējams pievienot arī dažādas formas. Formas nav kartes elementi; tās ir tikai vizuāls attēlojums. Piemēram, taisnstūra forma var tikt izmantota kā fons, lai grupētu dažus hosts. Var pievienot taisnstūra un elipses formas.

Lai pievienotu formu, noklikšķiniet uz *Add* blakus Shape. Jaunā forma parādīsies kartes augšējā kreisajā stūrī. Velciet un nometiet to jebkurā vietā.

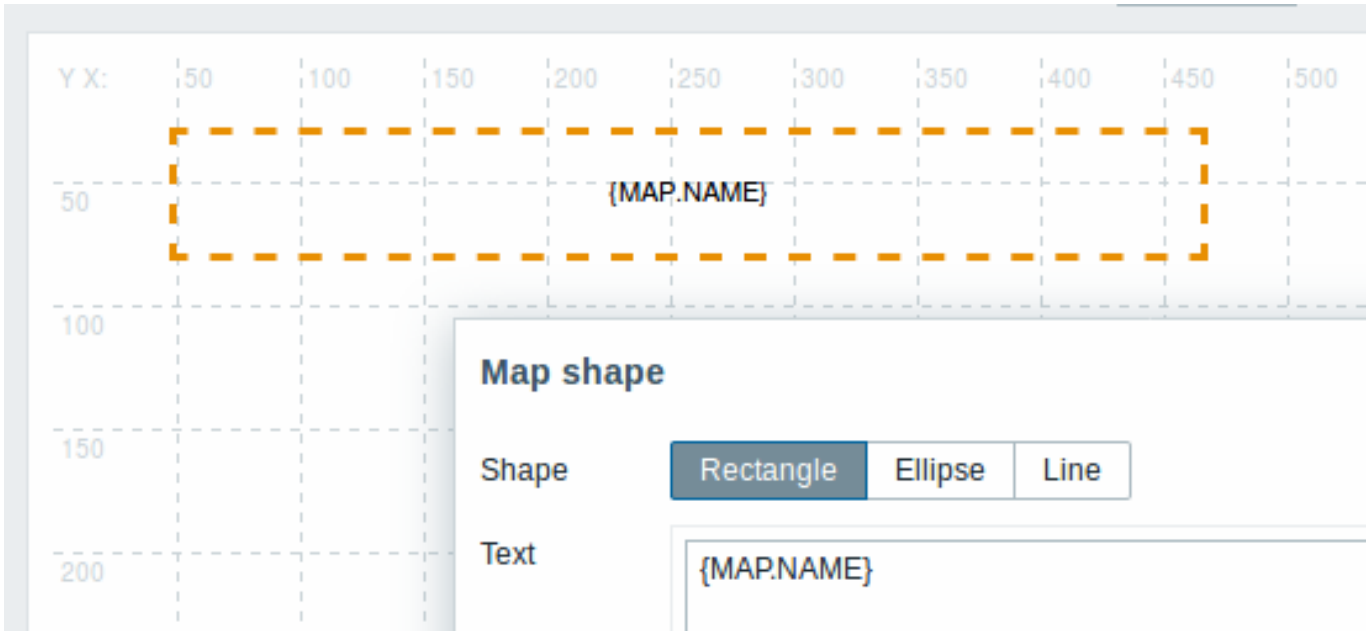
Jauna forma tiek pievienota ar noklusējuma krāsām. Noklikšķinot uz formas, tiek parādīta forma, kurā varat pielāgot formas izskatu, pievienot tekstu utt.



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 makro, taču tikai ar funkcijām `avg`, `last`, `min` un `max`, izmantojot laiku kā parametru (piemēram, `{?avg(/host/key,1h)}`).

Lai parādītu tikai tekstu, formu var padarīt neredzamu, noņemot formas apmali (laukam *Border* izvēlieties 'None'). Piemēram, pievērsiet uzmanību tam, ka {MAP.NAME} makro, kas redzams iepriekš redzamajā ekrānuzņēmumā, patiesībā ir taisnstūra forma ar tekstu, ko var redzēt, noklikšķinot uz makro:



{MAP.NAME} kartes skatīšanas laikā tiek aizstāts ar konfigurēto kartes nosaukumu.

Ja tekstā tiek izmantotas hipersaites, kartes skatīšanas laikā tās kļūst klikšķināmas.

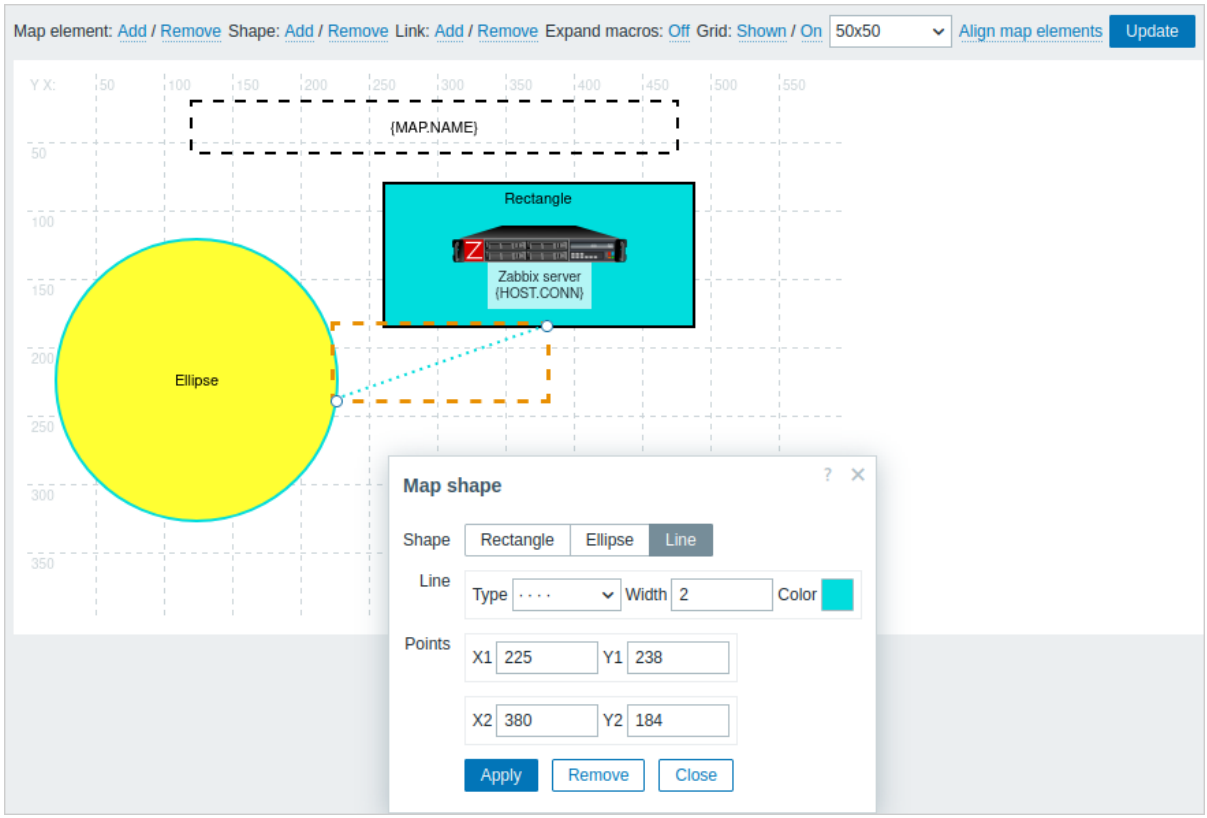
Teksta pārnešana jaunā rindā formās vienmēr ir ieslēgta. Tomēr elipsē rindas tiek pārnestas tā, it kā elipse būtu taisnstūris. Vārdu aplaušana nav ieviesta, tāpēc gari vārdi (vārdi, kas neietilpst formā) netiek aplauzti, bet tiek maskēti (kartes rediģēšanas lapā) vai

apgriezti (citās lapās ar kartēm).

Līniju pievienošana

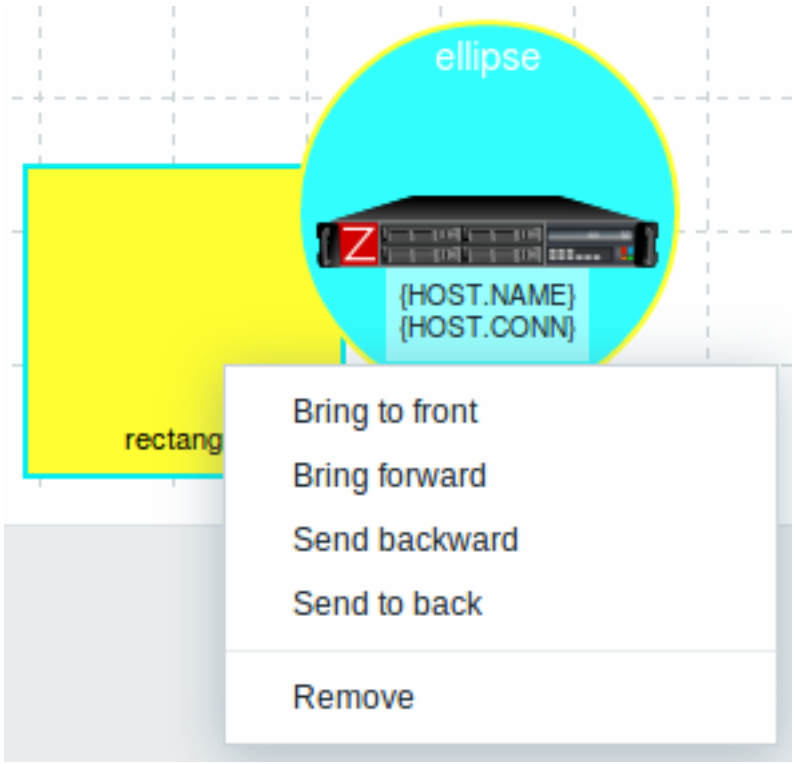
Papildus formām ir iespējams pievienot arī līnijas. Līnijas var izmantot, lai kartē savienotu elementus vai formas.

Lai pievienotu līniju, noklikšķiniet uz *Add* blakus Shape. Kartes augšējā kreisajā stūrī parādīsies jauna forma. Atlasiet to un rediģēšanas formā noklikšķiniet uz *Line*, lai pārveidotu formu par līniju. Pēc tam pielāgojiet līnijas īpašības, piemēram, līnijas tipu, platumu, krāsu utt.



Formu un līniju kārtošana

Lai vienu formu novietotu otras priekšā (vai otrādi), noklikšķiniet uz formas ar peles labo pogu, lai atvērtu kartes formas izvēlni.

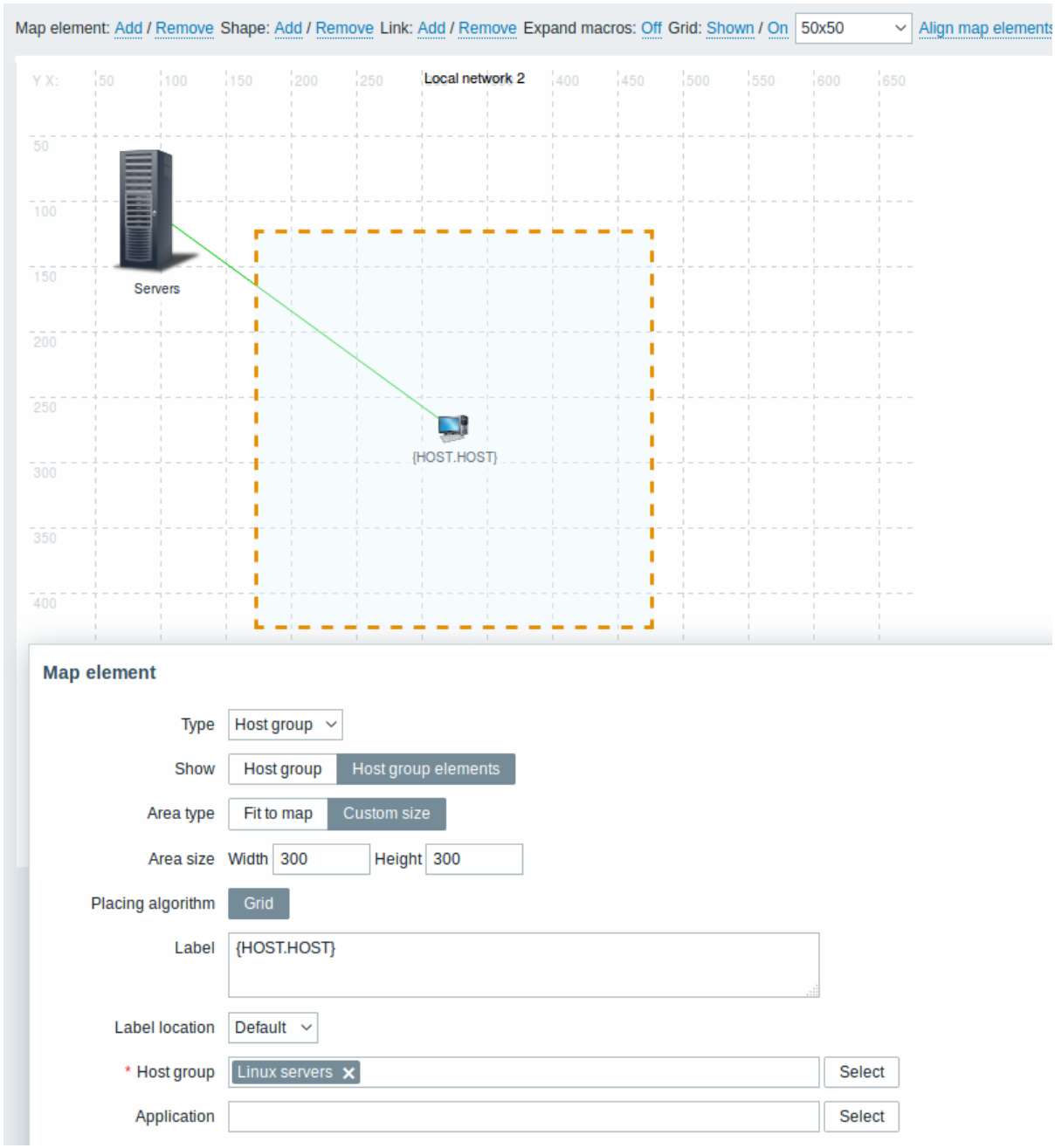


Hostu grupas elementi

Pārskats

Šajā sadaļā ir paskaidrots, kā pievienot *Host group* tipa elementu, konfigurējot tīkla karti.

Konfigurācija



Visi obligātie ievades lauki ir atzīmēti ar sarkanu zvaigznīti.

Šajā tabulā ir norādīti parametri, kas raksturīgi elementa tipam *Host group*:

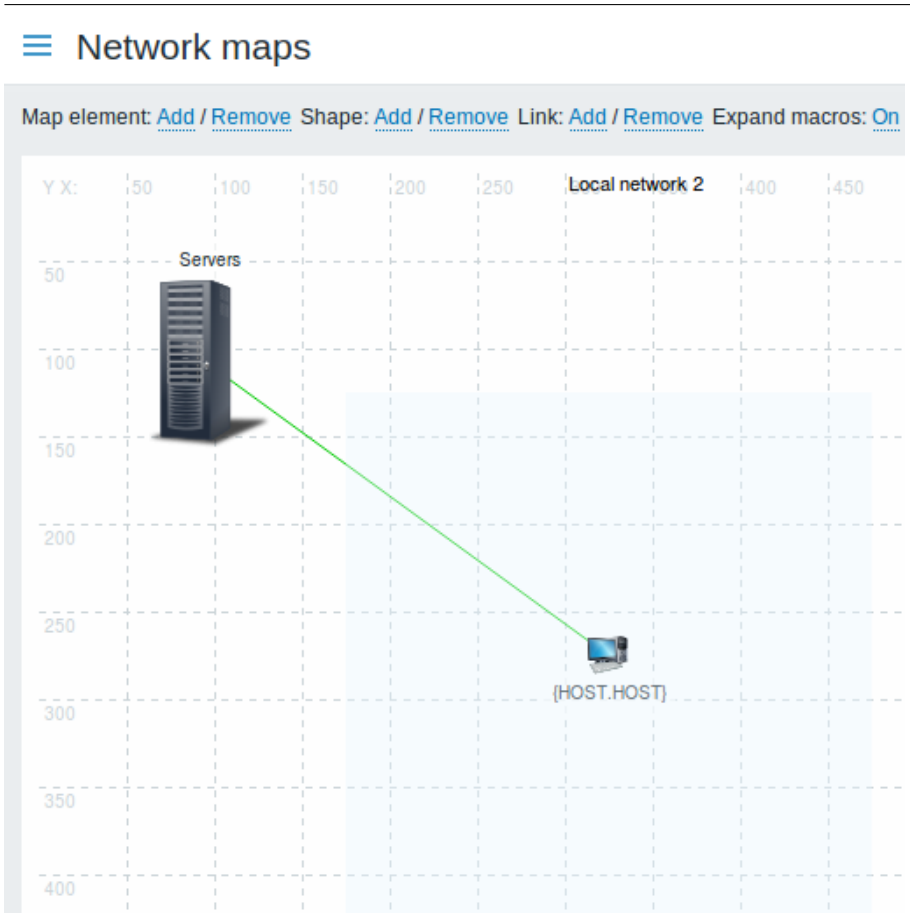
Parameter	Description
Type	Atlasiet elementa tipu: **Host group** - ikona, kas attēlo visu atlasītajai grupai piederošo hostu visu trigeru statusu.
Show	Rādīšanas opcijas: **Host group** - atlasot šo opciju, tiks parādīta viena ikona ar attiecīgo informāciju par konkrēto hostu grupu. **Host group elements** - atlasot šo opciju, tiks parādītas vairākas ikonas ar attiecīgo informāciju par katru konkrētās hostu grupas elementu (hostu).

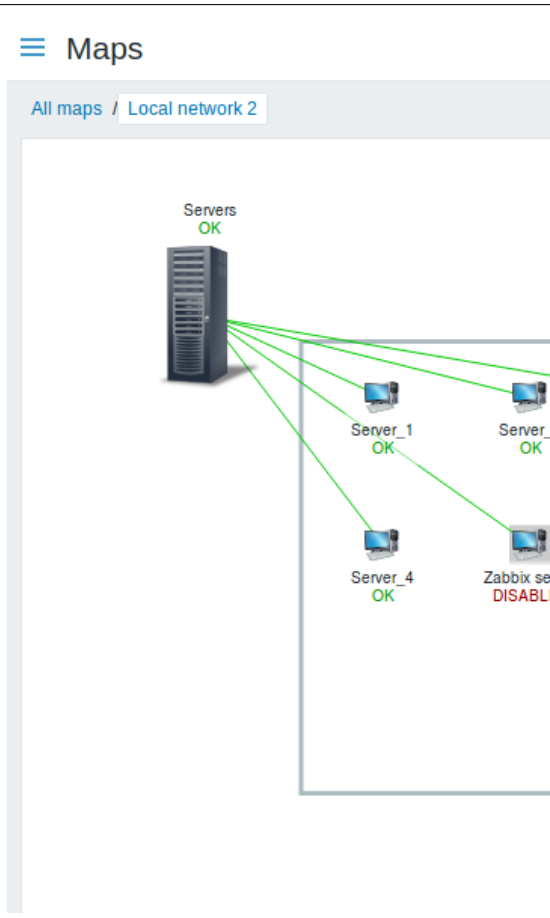
Parameter	Description
Area type	Šis iestatījums ir pieejams, ja *Show* ir iestatīts uz *Host group elements*: **Fit to map** - visi hostu grupas elementi tiek vienmērīgi izvietoti kartē. **Custom size** - manuāla kartes apgabala iestatīšana visiem attēlojamajiem hostu grupas elementiem.
Area size	Šis iestatījums ir pieejams, ja *Area type* ir iestatīts uz *Custom size*: **Width** - vērtība pikseļos, kas norāda kartes apgabala platumu. **Height** - vērtība pikseļos, kas norāda kartes apgabala augstumu.
Placing algorithm	**Grid** - vienīgā pieejamā opcija visu hostu grupas elementu attēlošanai.
Label	Ikonas etiķete, jebkura virkne. Etiķetēs var izmantot makro un vairākrindu virknes. Ja kartes elementa tips ir *Host group*, noteiktu makro norādīšana ietekmē kartes skatu, attēlojot attiecīgo informāciju par katru hostu. Piemēram, ja tiek izmantots makro `{HOST.IP}`, kartes rediģēšanas skatā tiks parādīts tikai pats makro `{HOST.IP}`, savukārt kartes skatā tiks iekļauta un parādīta katra hosta unikālā IP adrese.

Hostu grupas elementu skatīšana

Šī opcija ir pieejama, ja ir izvēlēta rādīšanas opcija *Hostu grupas elementi*.
Izvēloties *Hostu grupas elementi* kā rādīšanas opciju, sākumā redzēsiet tikai vienu hostu grupas ikonu.
Tomēr, saglabājot karti un pēc tam atverot kartes skatu, redzēsiet, ka kartē ir iekļauti visi noteiktās hostu grupas elementi (hosti):

Kartes rediģēšanas skats	Kartes skats





Ņemiet vērā, kā tiek izmantots makross `{HOST.HOST}`.
Kartes rediģēšanas skatā makrosa nosaukums nav atrisināts, savukārt kartes skatā tiek parādīti visi hostu unikālie nosaukumi.

Saites indikatori

Pārskats

Tīkla kartē varat piešķirt indikatorus saitei starp elementiem.

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 regulārai izteiksmei (teksta vienumiem).

Konfigurējot saiti, jūs iestatāt noklusējuma saites tipu un krāsu. Piešķirot saitei indikatorus, kļūst iespējams padarīt saites stilu un krāsu atkarīgu no trigera 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).

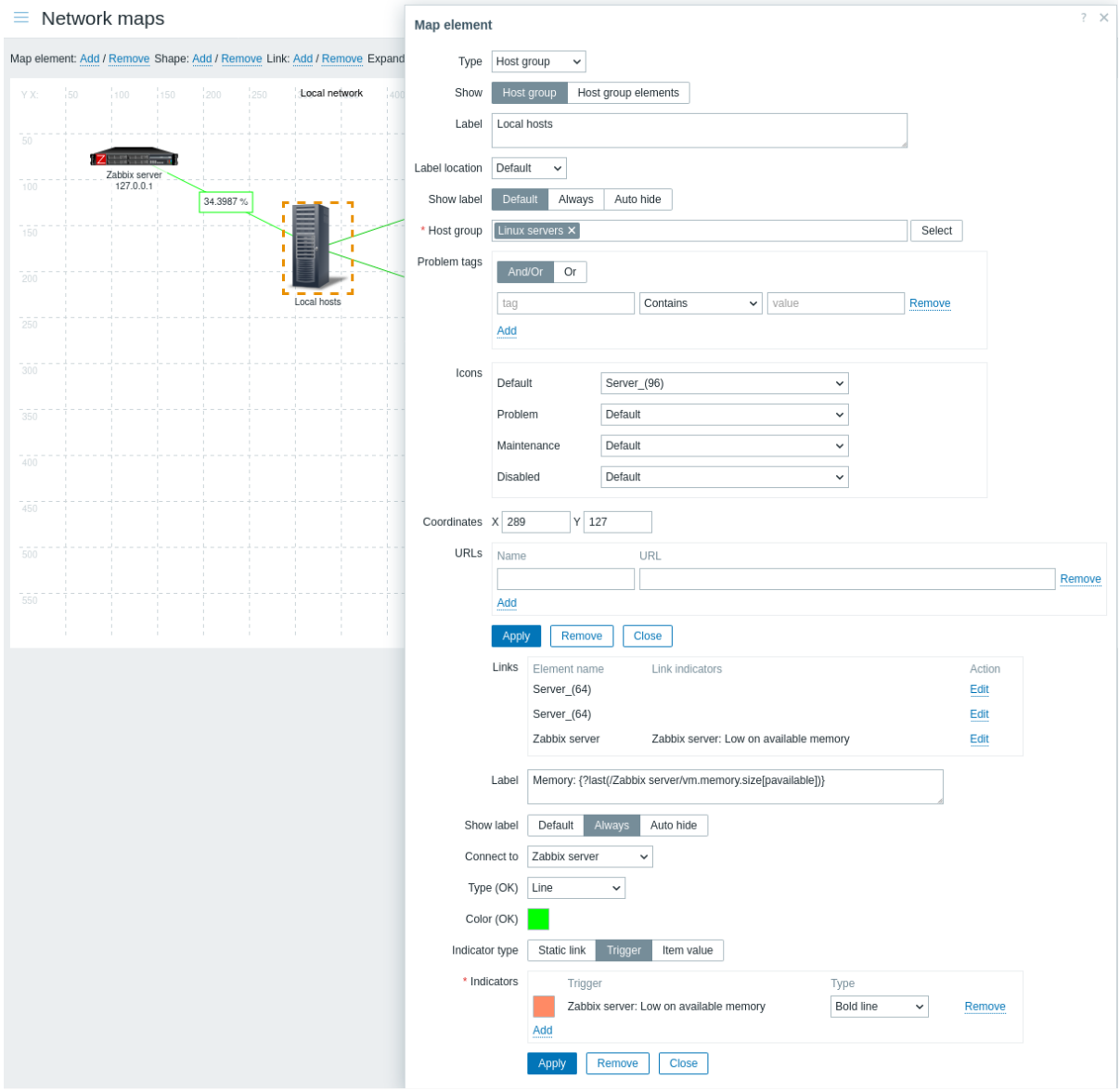
Līdzīgi, ja vienuma vērtība sasniedz norādīto slieksni vai atbilst norādītajai regulārajai izteiksmei, saites stils to var atspoguļot.

Konfigurācija

Trigeri

Lai piešķirtu trigerus kā saites indikatorus, rīkojieties šādi:

- atlasiet kartes elementu
- noklikšķiniet uz *Edit* sadaļā *Links* attiecīgajai saitei
- atlasiet *Trigger* kā indikatora tipu
- noklikšķiniet uz *Add* blokā *Indicators* un atlasiet vienu vai vairākus trigerus

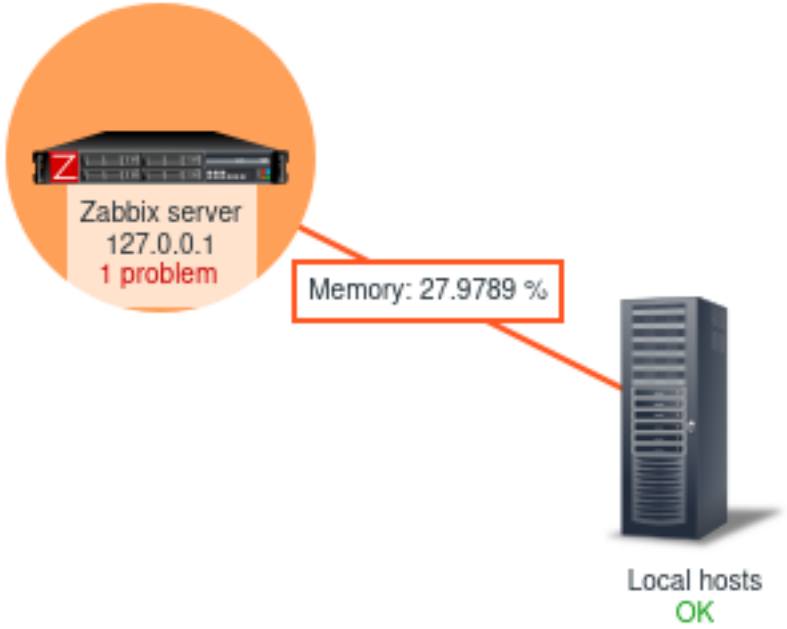


Visi obligātie ievades lauki ir atzīmēti ar sarkanu zvaigznīti.

Pievienotos trigerus var redzēt sarakstā *Indicators*.

Katram indikatoram varat tieši no saraksta iestatīt saites tipu un krāsu. 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 **triggeris** pāries problēmas stāvoklī.



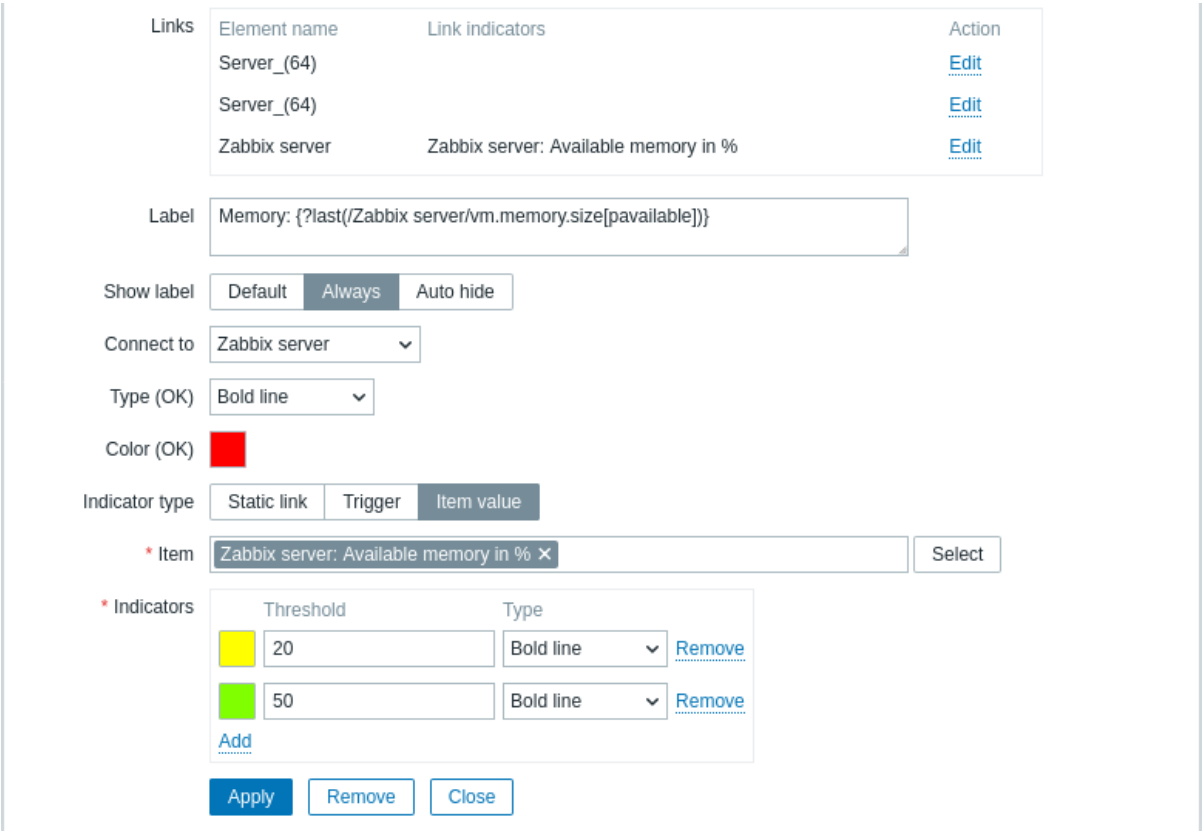
Note:
Ja vairāki trigeri pāriet problēmas stāvoklī, saites stilu un krāsu noteiks problēma ar visaugstāko nopietnību. Ja vienai un tai pašai kartes saitei ir piešķirti vairāki trigeri ar vienādu nopietnību, priekšroka tiek dota tam, kura ID ir mazākais. Ņemiet vērā arī to, ka:

1. Kartes konfigurācijas iestatījumi *Minimum trigger severity* un *Show suppressed problem* ietekmē to, kuri problēmu gadījumi tiek ņemti vērā.
2. Trigeru gadījumā ar vairākām problēmām (multiple problem generation) katrai problēmai var būt atšķirīga nopietnība nekā trigerim (nemainīta manuāli), var būt atšķirīgi tagi (makro dēļ), un tā var būt apspiesta.

Vienumu vērtības

Lai piešķirtu vienumu vērtības kā saites indikatorus, rīkojieties šādi:

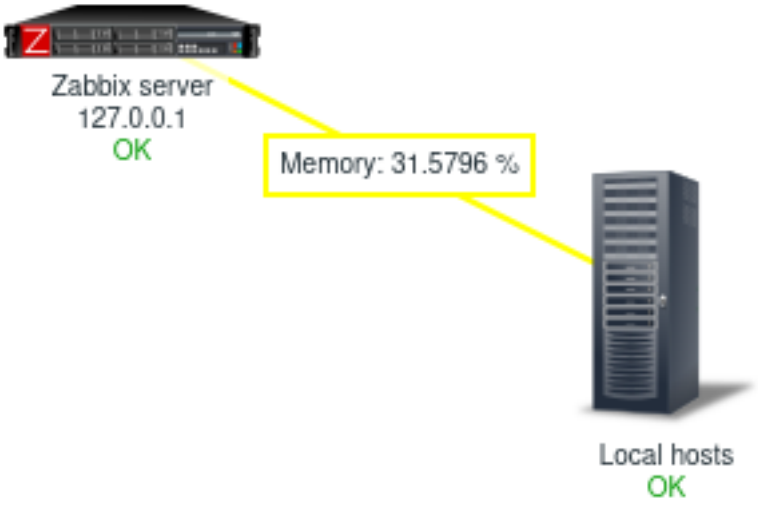
- atlasiet kartes elementu
- noklikšķiniet uz *Edit* sadaļā *Links* attiecīgajai saitei
- atlasiet *Item value* kā indikatora tipu
- atlasiet vienumu
- blokā *Indicators* pievienojiet vienu vai vairākus vienumu vērtību sliekšņus vai rakstus



Pievienotos vienumu sliekšņus/rakstus var redzēt sarakstā *Indicators*.

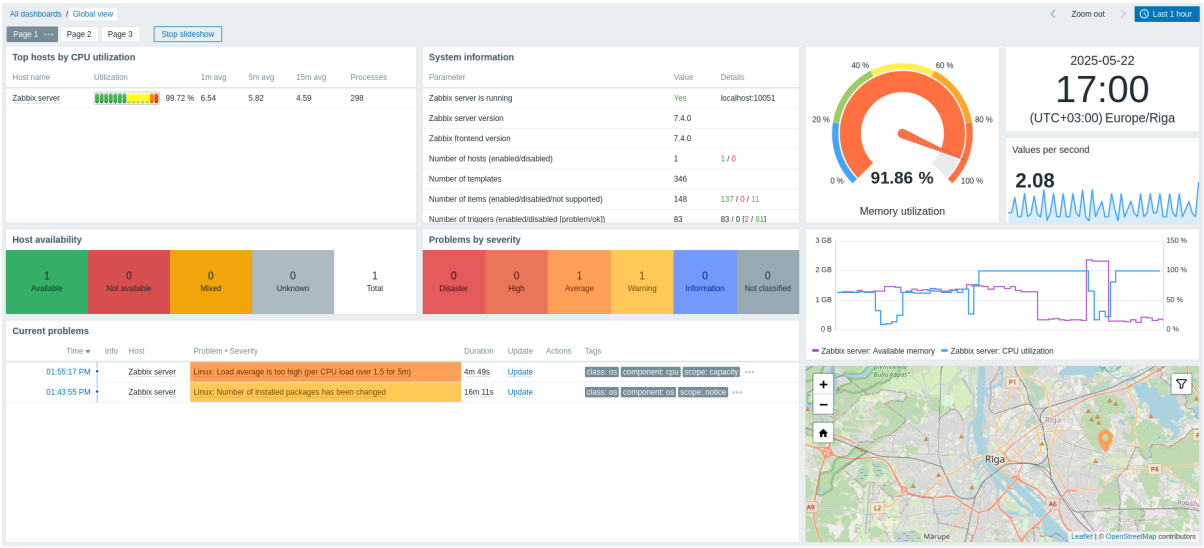
Katram indikatoram varat tieši no saraksta iestatīt saites tipu un krāsu. Kad tas ir izdarīts, noklikšķiniet uz *Apply*, aizveriet formu un noklikšķiniet uz *Update*, lai saglabātu kartes izmaiņas.

Sadaļā *Monitoring* > *Maps* attiecīgais saites tips un krāsa tiks parādīti, ja **vienuma vērtība** sasniedz iestatīto slieksni (skaitlisku datu tipu gadījumā) vai atbilst regulārās izteiksmes rakstam (teksta datu tipu gadījumā).



Informācijas paneļi

Informācijas paneļi - gan globālie informācijas paneļi, gan hostu informācijas paneļi - nodrošina jaudīgu vizualizācijas platformu ar tādiem logrīkiem un rīkiem kā moderni grafiki, kartes, slaidrādes un citi.



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.

Veidnes pēc noklusējuma

Pārskats

Zabbix nodrošina arvien plašāku iepriekš konfigurētu veidņu kopu, lai vienkāršotu un paātrinātu pārraudzības mērķu iestatīšanu.

Visas gatavās veidnes ir pieejamas sadaļā *Datu apkopošana* > *Veidnes*.

Atjauninot Zabbix, veidnes netiek atjauninātas automātiski. Tas novērš jūsu veikto pielāgoto izmaiņu pārrakstīšanu. Lai manuāli atjauninātu vai pievienotu veidnes, skatiet tālāk esošo sadaļu *Veidņu atjaunināšana*.

Izmantojiet sānjoslu, lai piekļūtu informācijai par konkrētiem veidņu tipiem un darbības prasībām.

Globālos informācijas paneļus, kas paredzēti darbam ar esošajām Zabbix veidnēm, var lejupielādēt no informācijas paneļu repozitorija un importēt savā Zabbix vidē. Detalizēts katra informācijas paneļa apraksts, tostarp priekšnosacījumi, ir pieejams informācijas paneļa failā `README`.

Skatiet arī:

- Veidnes saistīšana
- Zināmās problēmas saistībā ar veidnēm

Veidnes jaunināšana

Lai jauninātu vai pievienotu **vienu veidni**:

1. Atveriet `templates` direktoriju Zabbix Git repozitorijā.
2. Izvēlieties zaru (zem etiķetes *Source*), kas atbilst jūsu Zabbix versijai (piemēram, *master* izstrādes versijai).
3. Atrodiet veidni, kuru vēlaties pievienot vai jaunināt (piemēram, MySQL by Zabbix agent).
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.

> **Note:**
> Webhook datu nesēju tipus var jaunināt līdzīgā veidā: iegūstiet tos no `media` direktorijas un importējiet, dodoties uz *Alerts* > *Media types*.

Lai jauninātu vai pievienotu **vairākas veidnes**:

1. Lejupielādējiet veidņu komplektu arhīvu atkarībā no jūsu tīmekļa servera:
   - Nginx - `templates_bundles-8.0-1M-nginx.zip`
   - Apache - `templates_bundles-8.0-2M-apache.zip`
2. Izpakojiet arhīvu. Tajā ir YAML faili, no kuriem katrā ir vairākas veidnes, kā arī indeksa fails, kurā norādīts, kuras veidnes atrodas katrā YAML failā.
3. Zabbix lietotāja saskarnē dodieties uz *Data collection* > *Templates* un importējiet YAML failus pa vienam.

> **Note:**
> Webhook datu nesēju tipus var jaunināt līdzīgā veidā: paņemiet failu `media_bundle-8.0-000.yaml` no veidņu komplektu arhīva un importējiet, dodoties uz *Alerts* > *Media types*.

Lai jauninātu vai pievienotu **visas veidnes** vienlaikus, varat izmantot veidņu komplektu arhīvā iekļauto skriptu `import_templates.sh`. Skriptam ir nepieciešamas utilītprogrammas `curl` un `jq`, un tas izmanto Zabbix API metodi `configuration.import`. Palaižot skriptu, norādiet direktoriju, kurā atrodas YAML faili, kā arī Zabbix API URL un autentifikācijas pilnvaru.

> **Warning:**
> Skripts `import_templates.sh` pārraksta visas esošās veidnes, nenodrošinot iespēju pārskatīt vai atsaukt veidņu izmaiņas.

Piemērs:

```
/tmp/templates_bundles-8.0-2M-apache/import_templates.sh \
  /tmp/templates_bundles-8.0-2M-apache \
  https://example.com/zabbix/api_jsonrpc.php \
  d8d6b5c78ee2a8333db11bfcd3e448c49c239d8ce00702759e2b56f42e24dbe4
```

Veidnes saderība ar Host Wizard

Host Wizard ir saderīgs ar visām gatavajām veidnēm.

Varat arī padarīt savas pielāgotās veidnes saderīgas ar Host Wizard:

1. Zabbix lietotāja saskarnē dodieties uz *Datu apkopošana* > *Veidnes* un eksportējiet pielāgoto veidni.
2. Rediģējiet eksportēto failu:
   - Pievienojiet `wizard_ready` veidnes elementu, lai norādītu saderību ar Host Wizard.
   - Ja nepieciešams, pievienojiet `readme` vai `config` veidnes elementu.
3. Zabbix lietotāja saskarnē dodieties uz *Datu apkopošana* > *Veidnes* un importējiet atjaunināto veidnes failu.

Zabbix aģenta veidnes darbība

Darbības, lai nodrošinātu to veidņu pareizu darbību, kuras apkopo metriku, izmantojot Zabbix aģentu:

1. Pārliecinieties, ka hosts ir instalēts Zabbix aģents. Aktīvajām pārbaudēm arī pārliecinieties, ka Zabbix servera/starpniekservera adrese ir pievienota aģenta konfigurācijas faila parametram 'ServerActive'.
2. Piesaistiet veidni mērķa hostam (ja veidne nav pieejama jūsu Zabbix instalācijā, vispirms, iespējams, tā ir jāimportē).
3. Ja nepieciešams, pielāgojiet veidnes makro vērtības.
4. Konfigurējiet pārraugāmo instanci, lai tā atļautu datu kopīgošanu ar Zabbix.

Detalizēts veidnes apraksts, tostarp pilns makro, vienumu un trigeru saraksts, ir pieejams veidnes README failā (tam var piekļūt, noklikšķinot uz veidnes nosaukuma).

Ir pieejamas šādas veidnes:

- Apache by Zabbix agent
- Apache by Zabbix agent active
- HAProxy by Zabbix agent
- IIS by Zabbix agent
- IIS by Zabbix agent active
- MariaDB by Zabbix agent
- MariaDB by Zabbix agent active
- Microsoft Exchange Server 2016 by Zabbix agent
- Microsoft Exchange Server 2016 by Zabbix agent active
- MySQL by Zabbix agent
- MySQL by Zabbix agent active
- Nginx by Zabbix agent
- Nginx by Zabbix agent active
- Percona by Zabbix agent
- Percona by Zabbix agent active
- PHP-FPM by Zabbix agent
- PHP-FPM by Zabbix agent active
- Podman by Zabbix agent
- Podman by Zabbix agent active
- PostgreSQL by Zabbix agent
- PostgreSQL by Zabbix agent active
- RabbitMQ cluster by Zabbix agent

Zabbix aģents 2 veidnes darbība

Darbības, kas jāveic, lai nodrošinātu pareizu to veidņu darbību, kuras apkopo metriku ar Zabbix aģents 2:

1. Pārliecinieties, ka aģents 2 ir instalēts uz hosta un ka instalētajā versijā ir nepieciešamais spraudnis. Dažos gadījumos vispirms var būt nepieciešams jaunināt aģentu 2.
2. Piesaistiet veidni mērķa hostam (ja veidne nav pieejama jūsu Zabbix instalācijā, vispirms var būt nepieciešams to importēt).
3. Ja nepieciešams, pielāgojiet veidnes makro vērtības. Ņemiet vērā, ka lietotāja makro var izmantot konfigurācijas parametru pārrakstīšanai.
4. Konfigurējiet uzraugāmo instanci, lai tā atļautu datu koplietošanu ar Zabbix.

Attention:
Zabbix aģents 2 veidnes darbojas kopā ar spraudņiem. Lai gan pamata konfigurāciju var veikt, vienkārši pielāgojot lietotāja makro, dziļāku pielāgošanu var panākt, pašam konfigurējot spraudni. Piemēram, ja spraudnis atbalsta nosauktas sesijas, ir iespējams uzraudzīt vairākas viena veida entītijas (piem., MySQL1 un MySQL2), konfigurācijas failā katrai entītijai norādot nosauktu sesiju ar savu URI, lietotājvārdu un paroli.

Detalizēts veidnes apraksts, tostarp pilns makro, vienumu un trigeru saraksts, ir pieejams veidnes README failā (tam var piekļūt, noklikšķinot uz veidnes nosaukuma).

Pieejamas šādas veidnes:

- Ceph by Zabbix agent 2
- Docker
- Memcached
- MongoDB cluster by Zabbix agent 2
- MongoDB node by Zabbix agent 2

- MySQL by Zabbix agent 2
- MySQL by Zabbix agent 2 active
- Oracle by Zabbix agent 2
- PostgreSQL by Zabbix agent 2
- PostgreSQL by Zabbix agent 2 active
- Redis
- SMART by Zabbix agent 2
- SMART by Zabbix agent 2 active
- Systemd by Zabbix agent 2
- Website certificate by Zabbix agent 2

HTTP veidnes darbības

Darbības, lai nodrošinātu to veidņu pareizu darbību, kuras apkopo metriku, izmantojot HTTP aģentu:

1. Izveidojiet Zabbix hosts, kas atbilst uzraudzības mērķim. Kā galveno hosta saskarni norādiet uzraudzības mērķa IP adresi vai DNS nosaukumu. Tas ir nepieciešams, lai veidnes vienumos makro `{HOST.CONN}` tiktu pareizi aizstāts.
2. Piesaistiet veidni 1. darbībā izveidotajam hostam (ja veidne nav pieejama jūsu Zabbix instalācijā, vispirms, iespējams, tā ir jāimportē).
3. Ja nepieciešams, pielāgojiet veidnes līmeņa makro vērtības.
4. Konfigurējiet uzraugāmo instanci tā, lai tā atļautu datu kopīgošanu ar Zabbix.

Detalizēts veidnes apraksts, tostarp pilns makro, 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
- Claude API by HTTP
- ClickHouse by HTTP
- Cloudflare by HTTP
- CockroachDB by HTTP
- Control-M enterprise manager by HTTP
- Control-M server by HTTP
- Cradlepoint NCM v2 by HTTP
- DELL PowerEdge R720 by HTTP
- DELL PowerEdge R740 by HTTP
- DELL PowerEdge R820 by HTTP
- DELL PowerEdge R840 by HTTP
- Domain RDAP by HTTP
- Elasticsearch Cluster by HTTP
- Envoy Proxy by HTTP
- Etcd by HTTP
- FortiGate by HTTP
- GitHub repository by HTTP
- GitLab by HTTP
- GLPI by HTTP
- Google Cloud Platform (GCP) by HTTP

- Hadoop by HTTP
- HAProxy by HTTP
- HashiCorp Consul Cluster by HTTP
- HashiCorp Consul Node by HTTP
- HashiCorp Nomad by HTTP
- HashiCorp Vault by HTTP
- Hikvision camera by HTTP
- HPE iLO by HTTP
- HPE MSA 2040 Storage by HTTP
- HPE MSA 2060 Storage by HTTP
- HPE Primera by HTTP
- HPE Synergy by HTTP
- InfluxDB by HTTP
- Jenkins by HTTP
- Kubernetes API server by HTTP, Kubernetes cluster state by HTTP, Kubernetes Controller manager by HTTP, Kubernetes kubelet by HTTP, Kubernetes nodes by HTTP, Kubernetes Scheduler by HTTP ir novecojuši, un tos ir aizstājusi veidne *Kubernetes Cluster by HTTP*.
- Kubernetes Cluster by HTTP
- MantisBT by HTTP
- Microsoft 365 reports by HTTP
- Microsoft SharePoint by HTTP
- NetApp AFF A700 by HTTP
- Nextcloud by HTTP
- NGINX by HTTP
- NGINX Plus by HTTP
- Nutanix Prism Element by HTTP
- OpenAI Platform by HTTP
- OpenStack by HTTP
- OpenWeatherMap by HTTP
- Oracle Cloud by HTTP
- Palo Alto PA-440 by HTTP
- PHP-FPM by HTTP
- Podman by HTTP
- Proxmox VE by HTTP
- Pure Storage FlashArray v1 and v2 by HTTP
- RabbitMQ cluster by HTTP
- Ribbon SBC Edge by HTTP
- Ribbon SBC SWe CE by HTTP
- Ribbon SBC SWe Core by HTTP
- TiDB by HTTP
- TiDB PD by HTTP
- TiDB TiKV by HTTP
- Travis CI by HTTP
- Veeam Backup Enterprise Manager by HTTP
- Veeam Backup and Replication by HTTP
- VeloCloud SD-WAN by HTTP
- VeloCloud SD-WAN Edge by HTTP
- YugabyteDB by HTTP
- ZooKeeper by HTTP

IPMI veidnes darbība

IPMI veidnēm nav nepieciešama īpaša konfigurēšana. Lai sāktu uzraudzību, saistiet veidni ar mērķa hostu (ja veidne nav pieejama jūsu Zabbix instalācijā, vispirms, iespējams, būs jāveic tās importēšana).

Detalizēts veidnes apraksts, tostarp pilns makro, vienumu un trigeru saraksts, ir pieejams veidnes README failā (tam var piekļūt, noklikšķinot uz veidnes nosaukuma).

Pieejamā veidne:

- Chassis by IPMI

JMX veidnes darbība

Darbības soļi, lai nodrošinātu pareizu to veidņu darbību, kas apkopo metriku, izmantojot JMX:

1. Pārliecinieties, ka Zabbix Java gateway ir instalēts un pareizi konfigurēts.
2. Saistiet veidni ar mērķa hostu. Hostam jābūt konfigurētai JMX saskarnei. Ja veidne jūsu Zabbix instalācijā nav pieejama, iespējams, vispirms tā ir jāimportē.
3. Ja nepieciešams, pielāgojiet veidnes makro vērtības.
4. Konfigurējiet uzraugāmo instanci tā, lai tā atļautu datu koplietošanu ar Zabbix.

Detalizēts veidnes apraksts, tostarp pilns makro, vienumu un trigeru saraksts, ir pieejams veidnes README failā (tam var piekļūt, noklikšķinot uz veidnes nosaukuma).

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

ODBC veidnes darbība

Darbības, lai nodrošinātu pareizu veidņu darbību, kas apkopo metriku, izmantojot ODBC monitoring:

1. Pārliecinieties, ka nepieciešamais ODBC draiveris ir instalēts Zabbix serverī vai starpniekserverī.
2. Piesaistiet veidni mērķa hostam (ja veidne nav pieejama jūsu Zabbix instalācijā, vispirms, iespējams, būs jāveic tās importēšana).
3. Ja nepieciešams, pielāgojiet veidnes makro vērtības.
4. Konfigurējiet uzraudzīto instanci tā, lai tā atļautu datu koplietošanu ar Zabbix.

Detalizēts veidnes apraksts, tostarp pilns makro, vienumu un trigeru saraksts, ir pieejams veidnes README failā (tam var piekļūt, noklikšķinot uz veidnes nosaukuma).

Pieejamas šādas veidnes:

- MariaDB by ODBC
- MSSQL by ODBC
- MySQL by ODBC
- Oracle by ODBC
- Percona by ODBC
- PostgreSQL by ODBC

Standartizētas veidnes tīkla ierīcēm

Pārskats

Lai nodrošinātu tīkla ierīču, piemēram, komutatoru un maršrutētāju, uzraudzību, mēs esam izveidojuši divus tā sauktos modeļus: vienu pašai tīkla ierīcei (pamatā tās šasijai) un otru tīkla saskarnei.

Ir pieejamas veidnes daudzām tīkla ierīču saimēm. Visas veidnes aptver (ja vien no ierīces ir iespējams iegūt šos vienumus):

- Šasijas kļūmju uzraudzība (barošanas bloki, ventilatori un temperatūra, kopējais statuss)
- Šasijas veiktspējas uzraudzība (CPU un atmiņas vienumi)
- Šasijas inventāra apkopošana (sērijas numuri, modeļa nosaukums, programmaparatūras versija)
- Tīkla saskarņu uzraudzība ar IF-MIB un EtherLike-MIB (saskarnes statuss, saskarnes datplūsmas slodze, dupleksa statuss Ethernet)

Ja veidne nav pieejama jūsu Zabbix instalācijā, vispirms, iespējams, būs jāveic veidnes importēšana.

Ja importējat jaunās noklusējuma veidnes, iespējams, vēlēsieties arī atjaunināt globālo regulāro izteiksmi `@Network interfaces for discovery` uz:

```
Does not match regular expression: ^Software Loopback Interface
Does not match regular expression: ^(In)?[lL]oop[bB]ack[0-9._]*$
Does not match regular expression: ^NULL[0-9.]*$
Does not match regular expression: ^[lL]o[0-9.]*$
```

```
Does not match regular expression: ^[sS]ystem$
Does not match regular expression: ^Nu[0-9.]*$
```

lai vairumā sistēmu izfiltrētu cilpas saskarnes un nulles 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
Brocade FC SNMP	Brocade	Brocade FC komu-tatori	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 ugun-smūris	-	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

Veidnes nosaukums	Piegādātājs	Ierīču saime	Zināmie modeļi	OS	Izmantotie MIB	**Birkas**
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
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

Veidnes nosaukums	Piegādātājs	Ierīču saime	Zināmie modeļi	OS	Izmantotie MIB	**Birkas**
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 3000/5000/6000/8000/18000/5300/5500/5600/6800/18500/18800 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
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

Veidnes nosaukums	Piegādātājs	Ierīču saime	Zināmie modeļi	OS	Izmantotie MIB	**Birkas**
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:

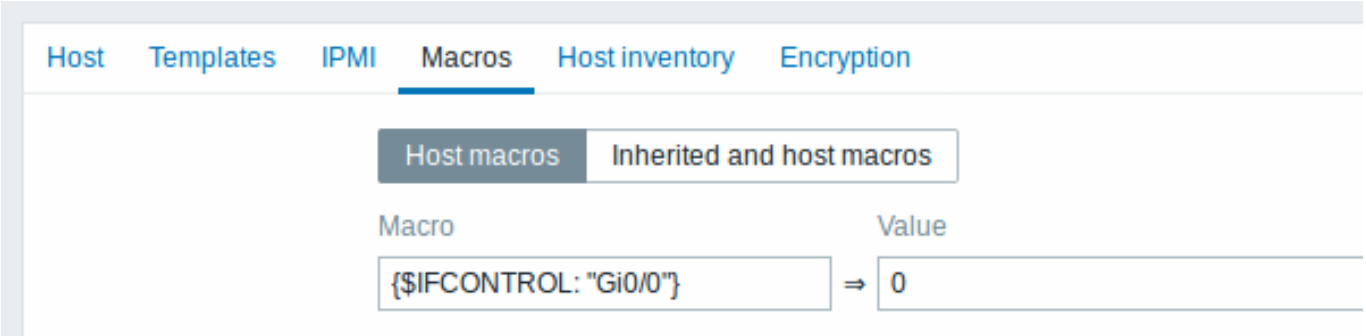
- Lietotāja makrosi tiek izmantoti pēc iespējas vairāk, lai lietotājs varētu pielāgot trigerus;
- Zema līmeņa atklāšana tiek izmantota pēc iespējas vairāk, lai samazinātu neatbalstīto vienumu skaitu;

- 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:

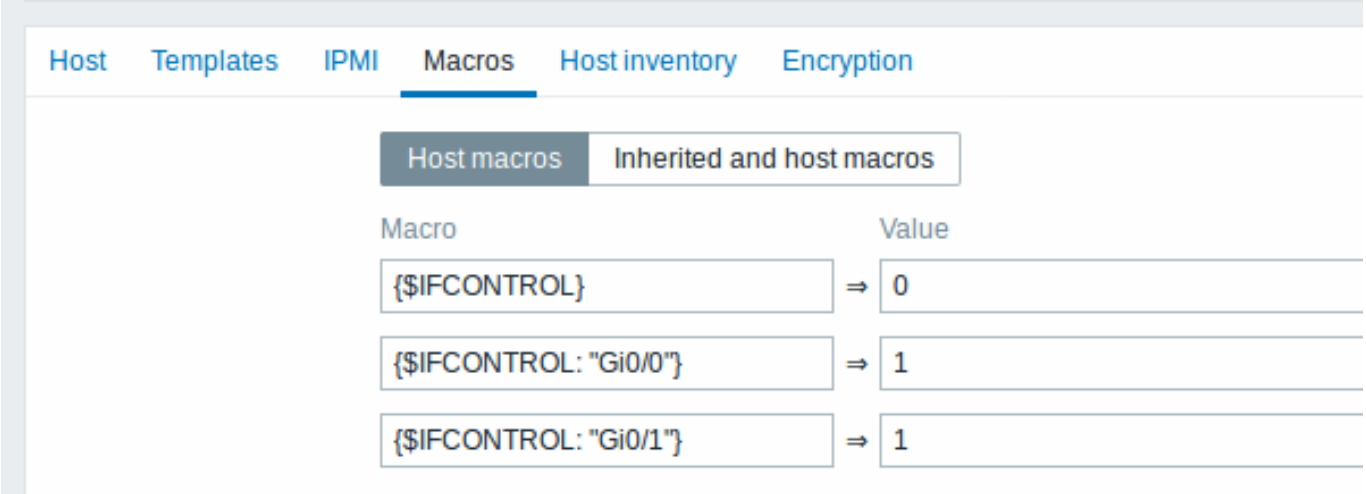
```
{$IFCONTROL:"{#IFNAME}"}=1 and last(/Alcatel Timetra TiMOS SNMP/net.if.status[ifOperStatus.{#SNMPINDEX}])
```

- 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 iegūt vismaz ierīces sērijas numuru un modeļa nosaukumu;
- Sertificēts - ir aptvertas visas trīs iepriekš minētās galvenās kategorijas.

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

OTLP veidnes darbība

Zabbix nodrošina virkni gatavu veidņu OpenTelemetry izsekojumu, metrikas un žurnālu apkopošanai un uzraudzībai, izmantojot *Telemetry query* vienumus.

Darbības, lai nodrošinātu pareizu veidņu darbību:

1. Pārbaudiet veidnes `README` failu, lai noskaidrotu veidnei specifiskās konfigurācijas prasības.
2. Izveidojiet Zabbix hosts, kas atbilst uzraudzības mērķim.
3. Pārliecinieties, ka parametrs `TelemetryProvider` ir konfigurēts Zabbix servera vai Zabbix starpniekservera konfigurācijas failā.
4. Piesaistiet veidni izveidotajam hosts (ja veidne nav pieejama jūsu Zabbix instalācijā, vispirms, iespējams, tā ir jāimportē).
5. Ja nepieciešams, pielāgojiet veidnes līmeņa makros.

Detalizēts veidņu apraksts, tostarp pilns makrosu, vienumu un trigeru saraksts, ir pieejams veidnes `README` failā, kuram var piekļūt, noklikšķinot uz veidnes nosaukuma.

Ir pieejamas šādas veidnes:

- Envoy Proxy by OTLP
- Generic OpenTelemetry by OTLP
- OpenRouter traces by OTLP
- Proxmox VE by OTLP

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.

Multivides tipi

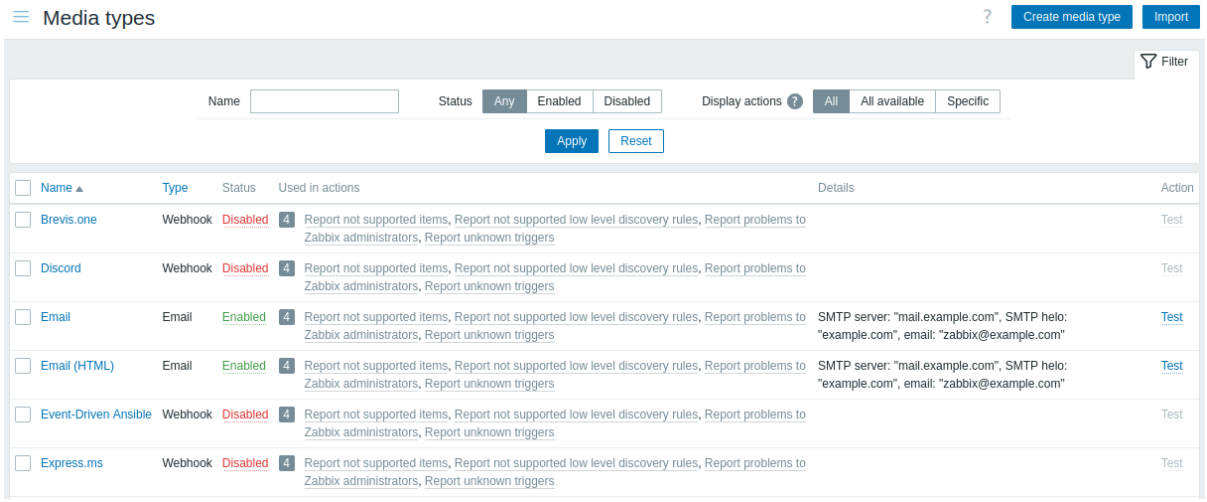
Pārskats

Multivides veidi ir piegādes kanāli, ko izmanto paziņojumu un brīdinājumu nosūtīšanai no Zabbix.

Multivides veidi atbalsta šādas piegādes metodes:

- E-pasts
- Pašpiegādes paziņojums
- SMS
- Pielāgots skripts
- Webhook

Multivides veidi tiek pārvaldīti sadaļā *Brīdinājumi* > *Multivides veidi*. Daži multivides veidi ir iepriekš definēti noklusējuma datu kopā. Lai tos sāktu izmantot, jums tikai jāpielāgo to parametri.



Lai saprastu, kā multivides veidi iekļaujas brīdinājumu nosūtīšanas procesā, apskatīsim trīs prasības paziņojumu piegādei no Zabbix galalietotājiem:

1. Jābūt definētai darbības operācijai, kas nosūta paziņojumus.
2. Jābūt definētam funkcionējošam multivides veidam, piemēram, *E-pastam*, kas nosūta brīdinājumus, izmantojot SMTP.
3. Lietotāja līmeņa piegādes informācija, piemēram, e-pasta adreses, tālruņa numuri u. c., jādefinē sadaļā lietotāja multivide.

Multivides veida testēšana

Lai pārbaudītu, vai konfigurētais multivides veids darbojas, multivides veidu sarakstā noklikšķiniet uz *Testēt*.

Testēšanas pieprasījums tiks nosūtīts Zabbix serverim. Zabbix serveris mēģinās nosūtīt brīdinājumu, izmantojot norādīto multivides veidu, un atgriezīs rezultātu lietotāja saskarnei. Lietotāja saskarne gaidīs, līdz serveris atgriezīs rezultātus. Multivides veida testēšanai pēc noklusējuma ir iestatīts 65 sekunžu noildzes laiks (to var konfigurēt sadaļā *Administrēšana* > *Vispārīgi* > *Noildzes*).

Skatiet arī detalizētu informāciju par šādu multivides veidu testēšanu:

- E-pasts
- Pašpiegādes paziņojums
- Webhook
- Skripts

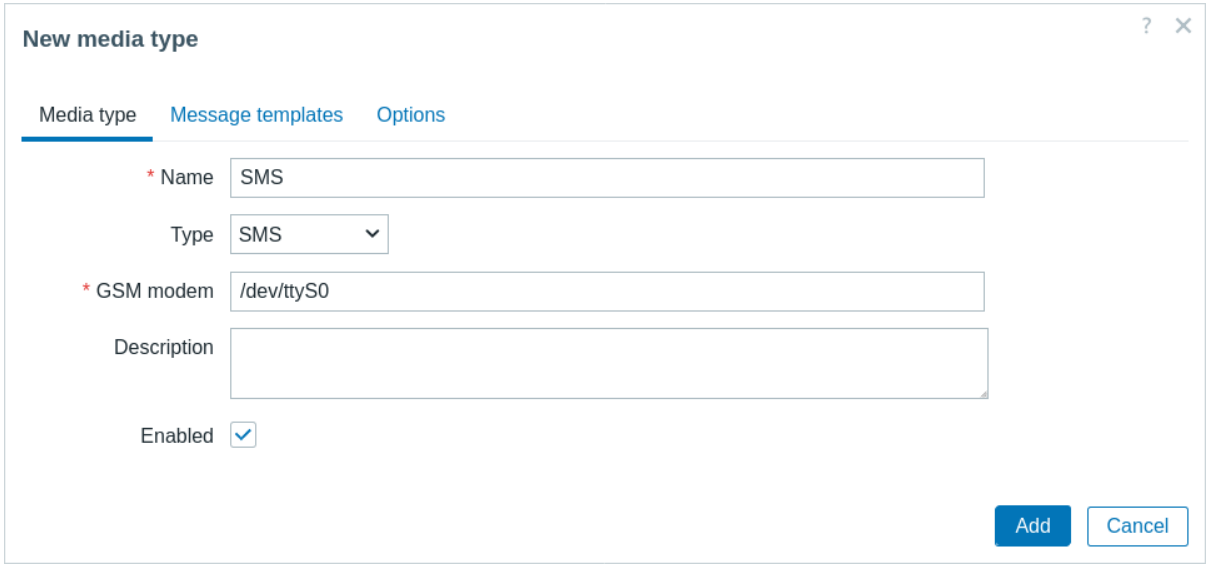
Konfigurēšana

Lai izveidotu multivides tipu Zabbix lietotāja saskarnē:

1. Atveriet *Alerts* > *Media types*.
2. Noklikšķiniet uz *Create media type*.
3. Veidlapā ievadiet multivides tipa parametrus.

Daži parametri ir **kopīgi** visām piegādes metodēm.

Note:
Ja viens vai vairāki piegādes metožu veidi ir norādīti `$ZBX_FEATURE_FLAGS['media_type_denylist']` funkcionalitātes karodziņā lietotāja saskarnes konfigurācijas failā (`zabbix.conf.php`), šīs piegādes metodes sarakstā *Type* nav pieejamas.

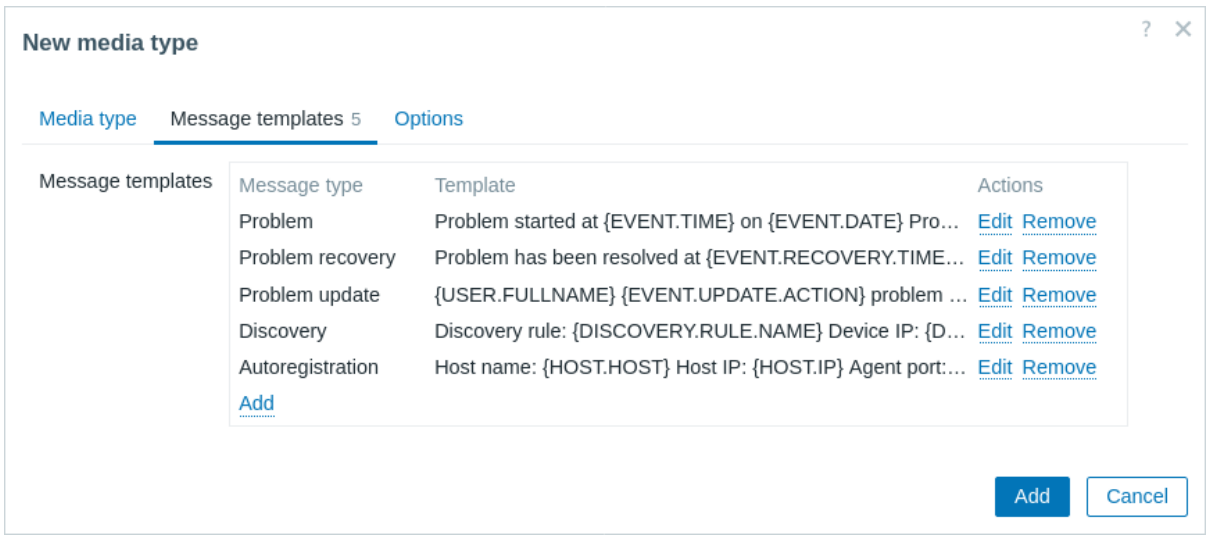


Parameter	Description
Name	Multivides tipa nosaukums.
Type	Atlasiet multivides tipa piegādes metodi.
Description	Ievadiet multivides tipa aprakstu.
Enabled	Atzīmējiet izvēles rūtiņu, lai iespējotu multivides tipu.

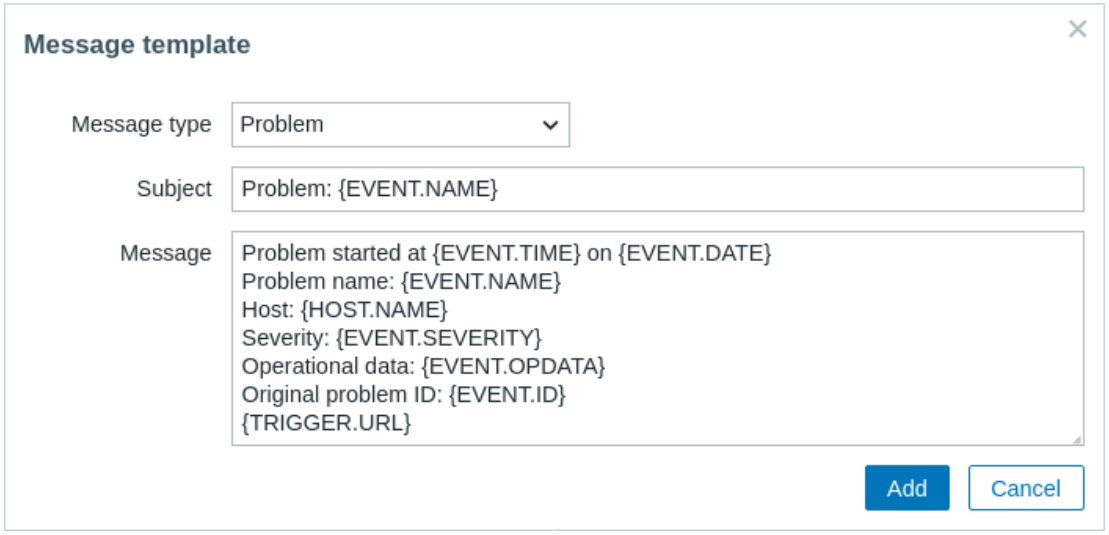
Informāciju par konkrētām metodēm paredzētiem parametriem skatiet e-pasta, push paziņojumu, SMS, pielāgota alertscript vai webhook lapā.

Ziņojumu veidnes

Cilne **Ziņojumu veidnes** satur noklusējuma ziņojumus katram notikuma tipam (problēma, problēmas atkopšana, atklāšana u. c.).



Noklikšķiniet uz **Pievienot**, lai definētu noklusējuma ziņojumu (vai uz **Rediģēt**, lai atjauninātu esošu ziņojumu):



Parameter	Description
Message type	Notikuma tips, kuram jāizmanto noklusējuma ziņojums. Katrai notikuma tipa vērtībai var definēt tikai vienu noklusējuma ziņojumu.
Subject	Noklusējuma ziņojuma temats. Temats var saturēt makro. Tā garums ir ierobežots līdz 255 rakstzīmēm. Temats nav pieejams SMS multivides tipam.
Message	Noklusējuma ziņojums. Tā rakstzīmju skaits ir ierobežots atkarībā no datubāzes tipa (plašāku informāciju skatiet sadaļā Ziņojumu sūtīšana). Ziņojums var saturēt atbalstītos makro. Problēmu un problēmu atjauninājumu ziņojumos tiek atbalstīti izteiksmes makro (piemēram, `{?avg(/host/key,1h)}`).

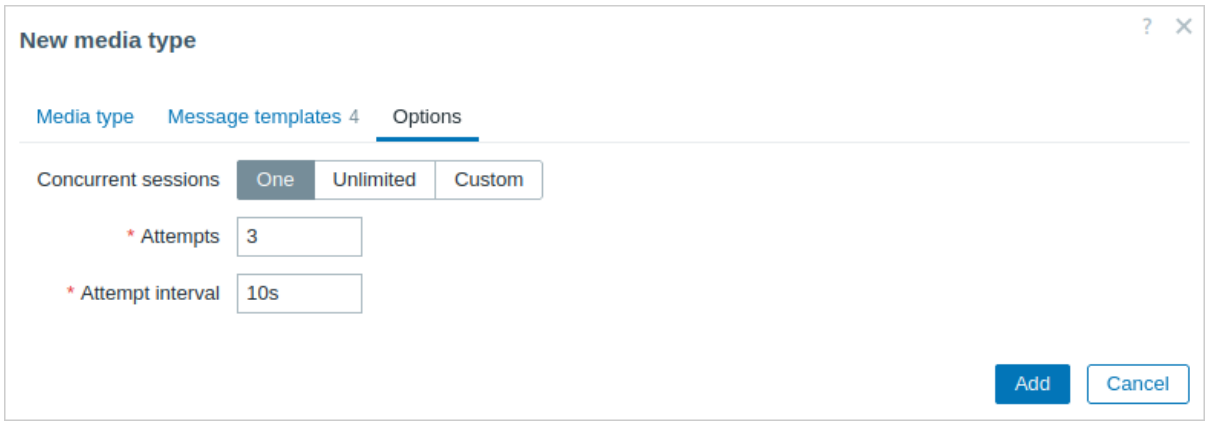
Ņemiet vērā, ka noklusējuma ziņojumus pārraksta pielāgotie ziņojumi, ja tie ir definēti darbību operācijās.

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" nespēs nosūtīt problēmu paziņojumus, ja Pushover webhook problēmas ziņojums nav definēts.

Opcijas

Cilnē **Opcijas** ir ietverti brīdinājumu apstrādes iestatījumi.
Katram multivides veidam var konfigurēt tādu pašu opciju kopu.

Visi multivides veidi tiek apstrādāti paralēli.
Lai gan maksimālo vienlaicīgo sesiju skaitu var konfigurēt katram multivides veidam,
kopējo brīdinājumu apstrādes procesu skaitu serverī var ierobežot tikai ar StartAlerters parametru.
Viena trigera ģenerētie brīdinājumi tiek apstrādāti secīgi.
Tāpēc vairākus paziņojumus var apstrādāt vienlaikus tikai tad, ja tos ģenerē vairāki trigeri.



Parametrs	Apraksts
Vienlaicīgās sesijas	Atlasiet multivides veidam paredzēto paralēlo brīdinājumu apstrādes sesiju skaitu: **Viena** - viena sesija **Neierobežots** - neierobežots sesiju skaits **Pielāgots** - atlasiet pielāgotu sesiju skaitu Neierobežots vai liels skaits nozīmē vairāk paralēlu sesiju un lielāku paziņojumu nosūtīšanas veiktspēju. Neierobežots vai liels skaits jāizmanto lielās vidēs, kur vienlaikus var būt nepieciešams nosūtīt daudz paziņojumu. Ja nosūtāmo paziņojumu ir vairāk nekā vienlaicīgo sesiju, atlikušie paziņojumi tiks ievietoti rindā; tie netiks zaudēti.
Mēģinājumi	Paziņojuma nosūtīšanas mēģinājumu skaits. Var norādīt līdz 100 mēģinājumiem; noklusējuma vērtība ir '3'. Ja norādīts '1', Zabbix nosūtīs paziņojumu tikai vienu reizi un nosūtīšana neizdosies atkārtoti.
Mēģinājumu intervāls	Atkārtotas paziņojuma nosūtīšanas mēģinājumu biežums gadījumā, ja nosūtīšana neizdevās, sekundēs (0-3600). Ja norādīts '0', Zabbix mēģinās nosūtīt paziņojumu atkārtoti nekavējoties. Tiek atbalstīti laika sufiksi (piemēram, 30s, 1m, 1h).

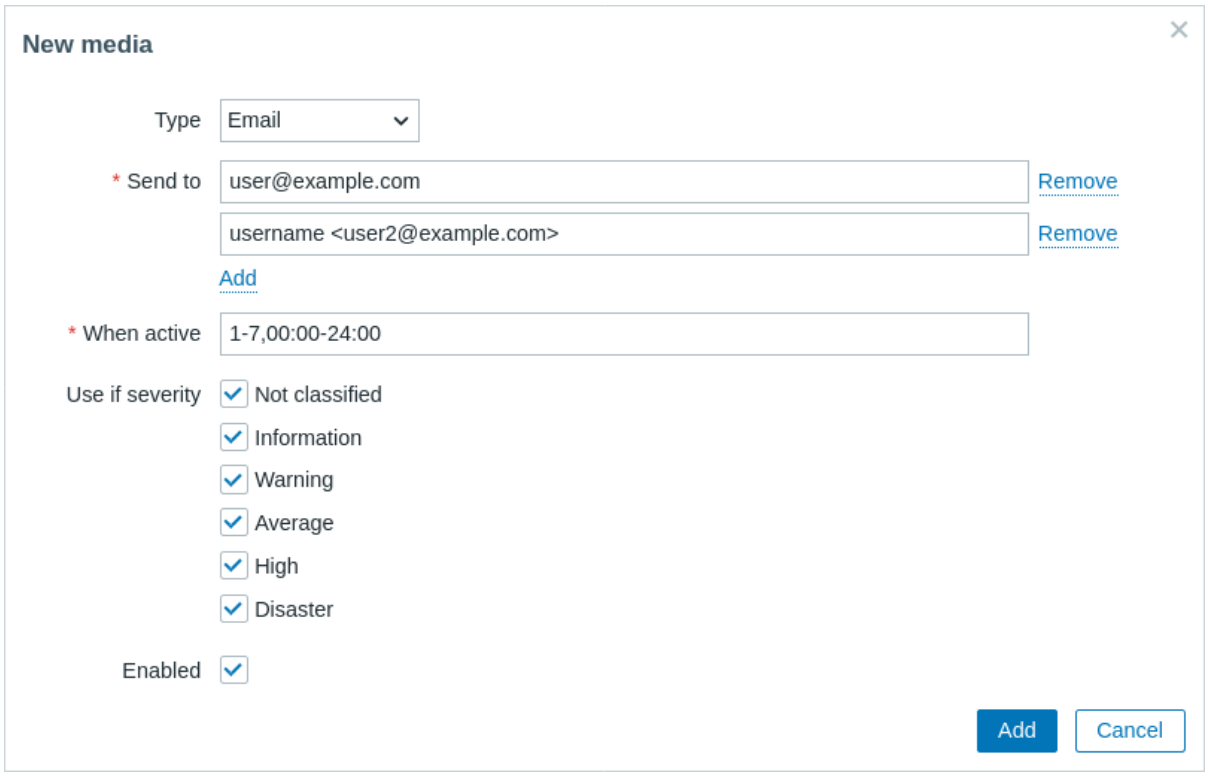
Lietotāja multivide

Kamēr multivides tipi nosaka, **kā** tiks nosūtīts paziņojums, lietotāja multivide nosaka, **kur** paziņojums jānosūta.

Lietotāja multivide (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 tos piegādāt, ja webhook *X* piegādes informācija nebūs definēta lietotāja *Admin* profilā.

Lai definētu lietotāja multividi:

1. Atveriet *Users* > *Users* un atveriet lietotāja rekvizītu veidlapu.
   Varat arī atvērt sava lietotāja profila sadaļu *User settings* > *Notifications*.
2. Cilnē *Media* noklikšķiniet uz *Add*.
3. Veidlapā ievadiet lietotāja multivides parametrus.



Parameter	Description
Type	Nolaižamajā sarakstā ir pieejami iespējoto multivides tipu nosaukumi. Ņemiet vērā: rediģējot atspējota multivides tipa multividi, tips tiks parādīts sarkanā krāsā.

Parameter	Description
Send to	Ievadiet kontaktinformāciju, uz kuru jānosūta ziņojumi. E-pasta multivides tipam var pievienot vairākas adreses, noklikšķinot uz pogas Add zem adreses lauka. Šādā gadījumā paziņojumi tiks nosūtīti uz visām norādītajām adresēm. Adrešu piemērus skatiet multivides tipa email parametra *Email* aprakstā. Push paziņojumu multivides tipam varat izvēlēties lietotāja aktīvās mobilās ierīces vai pievienot vienu vai vairākas konkrētas ierīces.
When active	Varat ierobežot laiku, kad tiek sūtīti ziņojumi, piemēram, iestatīt 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, šis ierobežojums, iespējams, būs attiecīgi jāpielāgo, lai netiktu palaisti garām svarīgi ziņojumi. 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. *Piezīme*: noklusējuma nopietnības līmenim (“Not classified”) **ir jābūt** atzīmētam, ja vēlaties saņemt paziņojumus par netrigeru notikumiem. Pēc saglabāšanas atlasītie trigeru nopietnības līmeņi tiks parādīti atbilstošajās nopietnības krāsās, bet neatlasītie līmeņi būs pelēkoti.
Status	Lietotāja multivides statuss. **Enabled** - tiek izmantota. **Disabled** - netiek izmantota.

E-pasts

Pārskats

Lai konfigurētu e-pastu kā ziņojumu piegādes kanālu, jums jākonfigurē e-pasts kā medija tips un jāpiešķir lietotājiem konkrētas adreses.

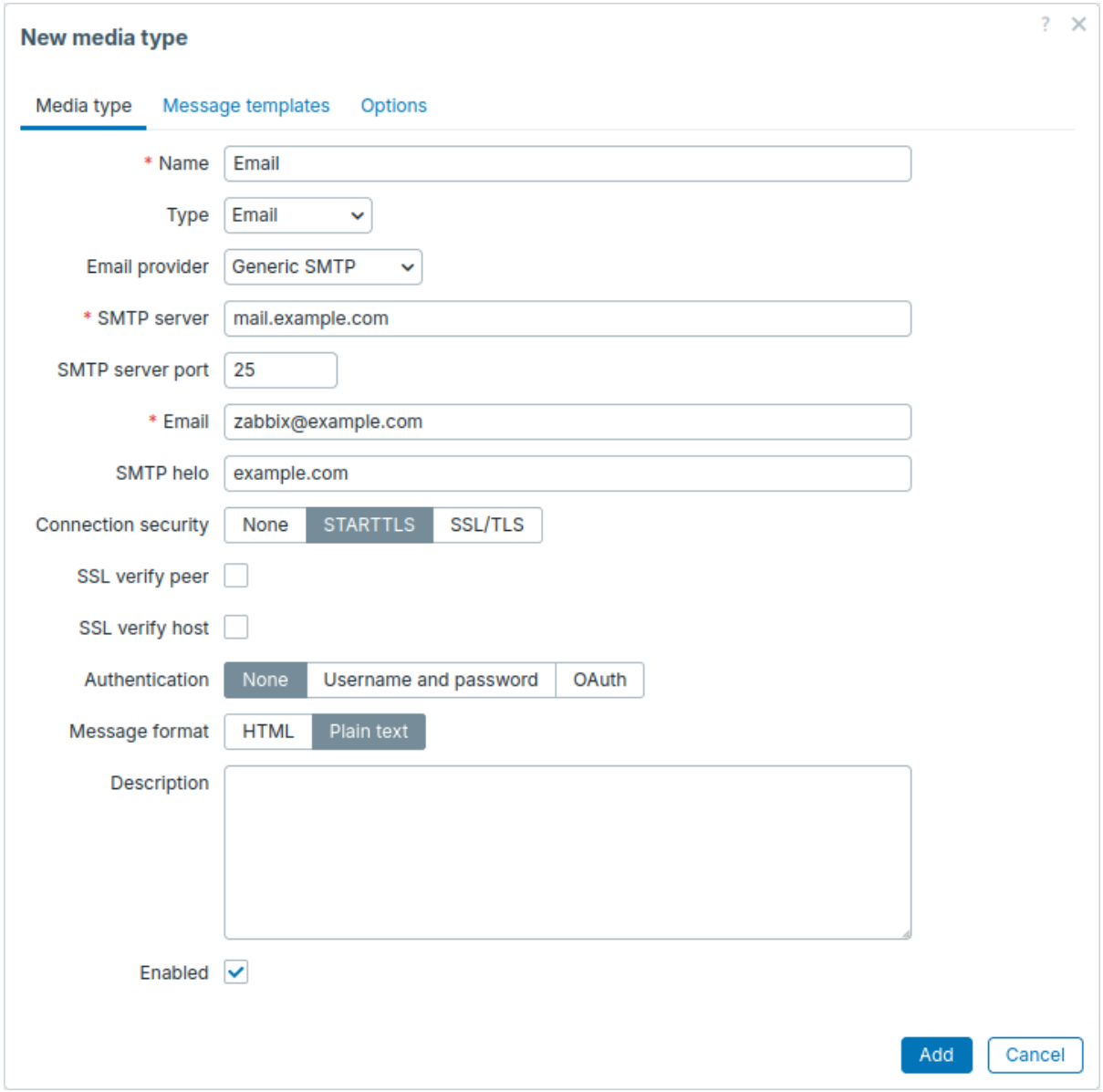
> **Note:**
> Vairāki paziņojumi par vienu notikumu tiks grupēti vienā un tajā pašā e-pasta sarakstē.

Konfigurēšana

Lai konfigurētu e-pastu kā multivides tipu:

1. Dodieties uz *Alerts* > *Media types*.
2. Noklikšķiniet uz *Create media type* (vai sarakstā ar iepriekš definētajiem multivides tipiem noklikšķiniet uz *Email*).
3. Veidlapā ievadiet e-pasta multivides tipa parametrus.

Cilnē **Media type** ir ietverti vispārīgie multivides tipa atribūti:



Visi obligātie ievades lauki ir atzīmēti ar sarkanu zvaigznīti.

Warning:
Eksportējot e-pasta multivides tipa konfigurāciju, parole tiek eksportēta nešifrētā veidā.

Tālāk norādītie parametri attiecas uz e-pasta multivides tipu:

Parameter	Description
Email provider	Atlasiet e-pasta pakalpojumu sniedzēju: *Generic SMTP*, *Gmail*, *Gmail relay*, *Office365* vai *Office365 relay*. Ja atlasāt ar Gmail/Office365 saistītās opcijas, jums būs jānorāda tikai sūtītāja e-pasta adrese un parole; tādas opcijas kā *SMTP server*, *SMTP server port*, *SMTP helo* un *Connection security* Zabbix aizpildīs automātiski. Skatiet arī: Automated Gmail/Office365 media types.
SMTP server	Norādiet SMTP serveri izejošo ziņojumu apstrādei. Šis lauks ir pieejams, ja par e-pasta pakalpojumu sniedzēju ir atlasīts *Generic SMTP*.
SMTP server port	Norādiet SMTP servera portu izejošo ziņojumu apstrādei. Šis lauks ir pieejams, ja par e-pasta pakalpojumu sniedzēju ir atlasīts *Generic SMTP*.

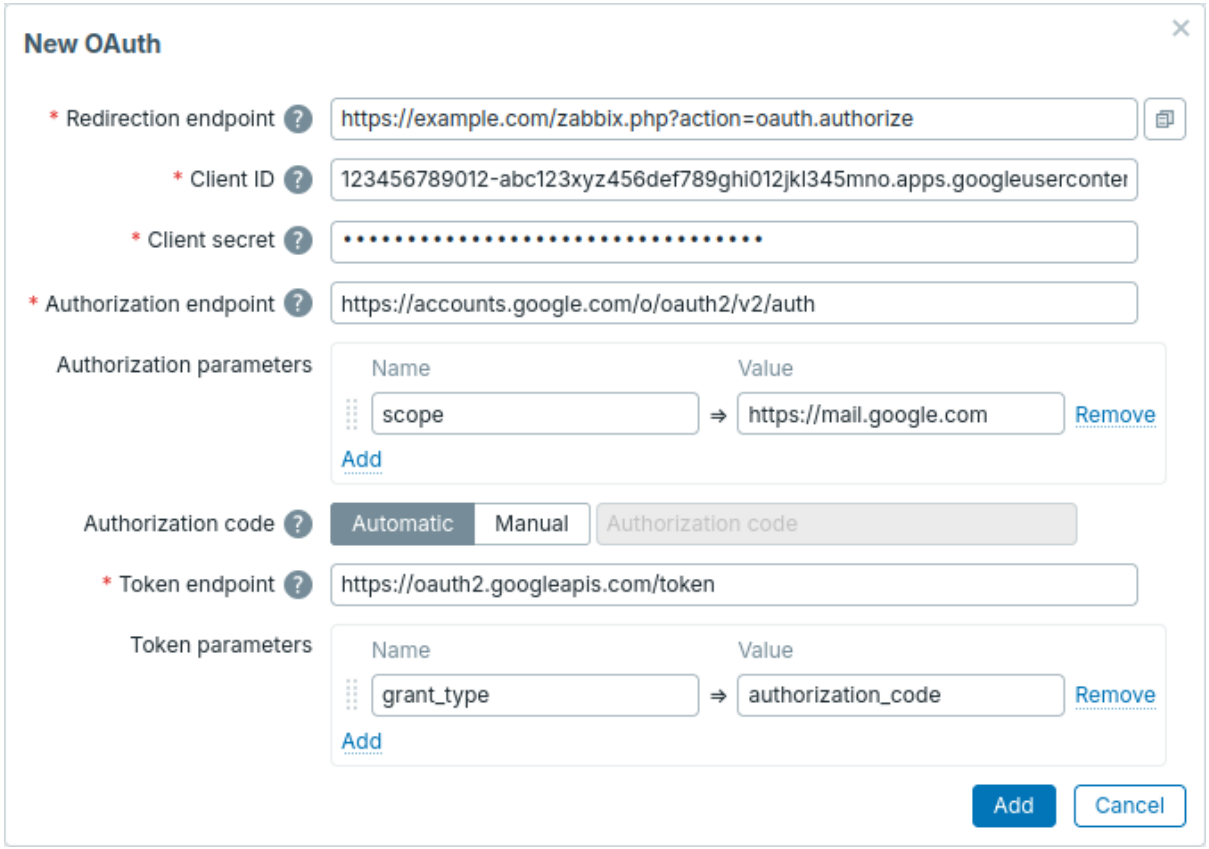
Parameter	Description
Email	Šeit ievadītā adrese tiks izmantota kā nosūtīto ziņojumu **From** adrese. Tiek atbalstīta sūtītāja parādāmā vārda pievienošana faktiskajai e-pasta adresei (piemēram, `Zabbix_info <zabbix@company.com>`). Zabbix e-pastos parādāmajiem vārdiem ir daži ierobežojumi salīdzinājumā ar tiem, ko atļauj RFC 5322, kā parādīts tālāk sniegtajos piemēros. Derīgi piemēri: `zabbix@company.com` (tikai e-pasta adrese; leņķiekavas nav nepieciešamas) `Zabbix_info <zabbix@company.com>` (parādāmais vārds, kam seko e-pasta adrese leņķiekavās) `Ω-monitoring <zabbix@company.com>` (UTF-8 rakstzīmes parādāmajā vārdā) Nederīgi piemēri: `Zabbix HQ zabbix@company.com` (parādāmais vārds ir norādīts, bet ap e-pasta adresi nav leņķiekavu) `"Zabbix@<H(comment)Q>" <zabbix@company.com>` (lai gan tas ir derīgs saskaņā ar RFC 5322, Zabbix e-pastos netiek atbalstīti citētie pāri un komentāri).
SMTP helo	Norādiet pareizu SMTP helo vērtību, parasti domēna nosaukumu. Ja lauks ir tukšs, tiks nosūtīts e-pasta domēna nosaukums (t. i., tas, kas laukā *Email* atrodas aiz `@`). Ja domēna nosaukumu nav iespējams iegūt, žurnālā tiks ierakstīts atkļūdošanas līmeņa brīdinājums un HELO komandai kā domēns tiks nosūtīts servera resursdatora nosaukums. Šis lauks ir pieejams, ja par e-pasta pakalpojumu sniedzēju ir atlasīts *Generic SMTP*.
Connection security	Atlasiet savienojuma drošības līmeni: **None** - neizmantot opciju CURLOPT_USE_SSL **STARTTLS** - izmantot opciju CURLOPT_USE_SSL ar vērtību CURLUSESSL_ALL **SSL/TLS** - CURLOPT_USE_SSL izmantošana nav obligāta.
SSL verify peer	Atzīmējiet izvēles rūtiņu, lai pārbaudītu SMTP servera SSL sertifikātu. Servera konfigurācijas direktīvas "SSLCALocation" vērtība sertifikāta validācijai jāievieto CURLOPT_CAPATH. Tādējādi tiek iestatīta cURL opcija CURLOPT_SSL_VERIFYPEER.
SSL verify host	Atzīmējiet izvēles rūtiņu, lai pārbaudītu, vai SMTP servera sertifikāta lauks *Common Name* vai *Subject Alternate Name* atbilst. Tādējādi tiek iestatīta cURL opcija CURLOPT_SSL_VERIFYHOST.
Authentication	Atlasiet autentifikācijas līmeni: **None** - cURL opcijas netiek iestatītas **Username and password** - nozīmē "AUTH=*", atstājot autentifikācijas mehānisma izvēli cURL ziņā **OAuth** - OAuth autentifikācija OAuth autentifikācija netiek atbalstīta *Office365 relay* e-pasta pakalpojumu sniedzējam.
Username	Autentifikācijai izmantojamais lietotājvārds. Tādējādi tiek iestatīta CURLOPT_USERNAME vērtība. Tiek atbalstīti lietotāja makro.
Password	Autentifikācijai izmantojamā parole. Tādējādi tiek iestatīta CURLOPT_PASSWORD vērtība. Tiek atbalstīti lietotāja makro. Šeit ievadītā vērtība tiek eksportēta tāda, kāda tā ir, eksportējot multivides tipa konfigurāciju. Lai eksportētajos failos neizpaustu sensitīvus akreditācijas datus, burtiskas paroles vietā izmantojiet lietotāja makro (piemēram, `{$EMAIL_SMTP_PASSWORD}`) - tiks eksportēta makro atsauce, savukārt slepenā vērtība galamērķa sistēmā ir jā(konfigurē atkārtoti).
OAuth tokens	Noklikšķiniet uz *Configure*, lai jaunā logā iestatītu parametrus OAuth tokens izgūšanai. Šis lauks ir pieejams tikai tad, ja laukā *Authentication* ir atlasīts "OAuth".
Message format	Atlasiet ziņojuma formātu: **HTML** - nosūtīt HTML formātā; **Plain text** - nosūtīt vienkārša teksta formātā.

Attention:
Lai iespējotu SMTP autentifikāciju, Zabbix serveris ir jākompilē ar `--with-libcurl` kompilēšanas opciju, kurai nepieciešama libcurl bibliotēka (informāciju par versijām skatiet servera prasībās).

Detalizētu informāciju par noklusējuma ziņojumu un trauksmju apstrādes opciju konfigurēšanu skatiet sadaļā common media type parameters.

OAuth tokeni

Lai izgūtu OAuth tokenus, ir nepieciešami šādi parametri:



Warning:
Klienta noslēpums tiek eksportēts atklātā tekstā, eksportējot e-pasta multivides tipa konfigurāciju.

Parameter	Description
Pāradresācijas galapunkts	Ievadiet Zabbix lietotāja saskarne URL, uz kuru OAuth pakalpojums pāradresēs OAuth autorizāciju atpakaļ (izmantojot formātu `https://<zabbix-frontend-url>/zabbix.php?action=oauth.authorize`). Tas tiek iestatīts automātiski jaunam e-pasta multivides tipam, ja ir definēts lietotāja saskarne URL.
Klienta ID	Ievadiet OAuth autorizācijas serverī reģistrētās klienta lietotnes unikālo identifikatoru.
Klienta noslēpums	Ievadiet OAuth autorizācijas serverī reģistrētās klienta lietotnes privāto noslēpumu.
Autorizācijas galapunkts	Ievadiet OAuth autorizācijas servera URL, lai pieprasītu lietotāja autorizāciju.
Autorizācijas parametri	Ievadiet autorizācijas galapunkta parametrus.
Autorizācijas kods	Ievadiet autorizācijas kodu: **Automātiski** - kods tiks izgūts automātiski, izmantojot pāradresācijas lapu; **Manuāli** - ievadiet kodu manuāli, ja automātiska izgūšana neizdodas.
Tokena galapunkts	Ievadiet OAuth autorizācijas servera URL, lai apmainītu autorizācijas kodu pret piekļuves un atsvaidzināšanas tokeniem.
Tokena parametri	Ievadiet piekļuves tokena parametrus.

Note:
OAuth tokenu izgūšanai ir automatizētas funkcijas e-pasta pakalpojumu sniedzējiem **Gmail**, **Gmail relay** un **Office365**. Šādā gadījumā ir jānorāda tikai *Pāradresācijas galapunkts*, *Klienta ID* un *Klienta noslēpums* parametru 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.

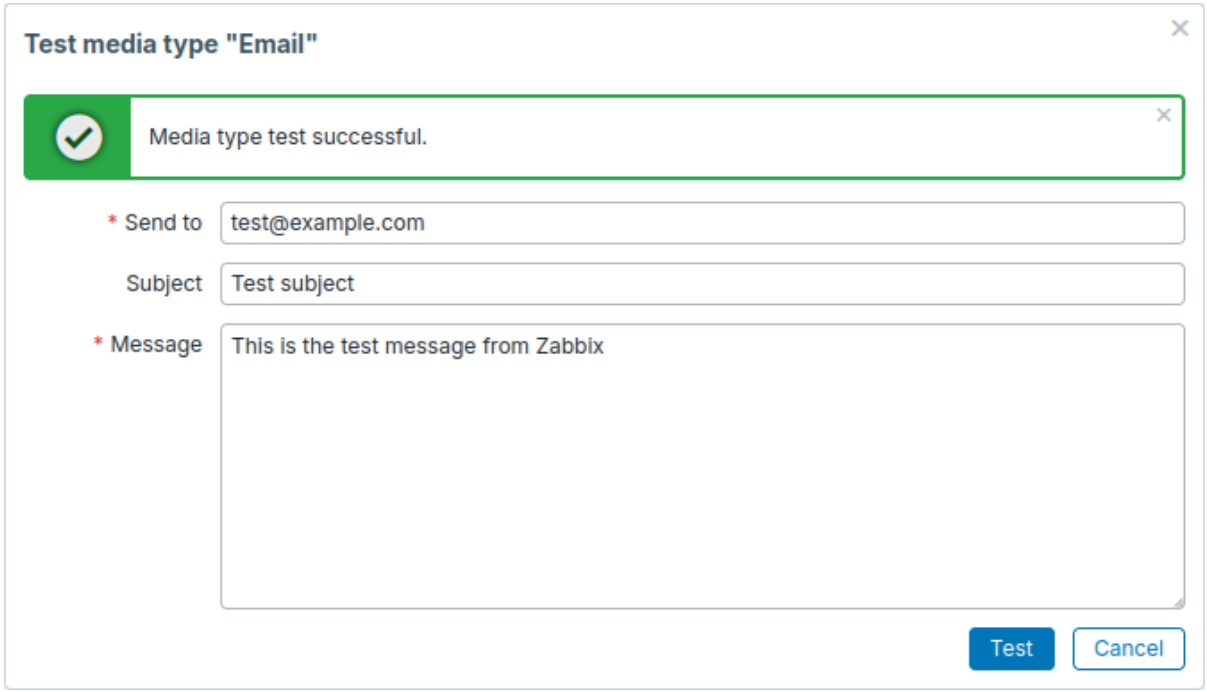
3. OAuth serviss novirza lietotāju uz Zabbix lietotāja saskarnes darbību `oauth.authorize` ar autorizācijas kodu un tvēruma vērtību.
4. Atbildot uz to, Zabbix veiks pieprasījumu uz *marķiera galapunktu*, lai apmainītu autorizācijas kodu pret piekļuves un atsvaidzināšanas marķieriem.

Testēšana

Lai pārbaudītu, vai konfigurētais e-pasta mediju tips darbojas pareizi:

1. Atrodiet attiecīgo e-pastu sarakstā ar mediju tipiem.
2. Noklikšķiniet uz *Test* saraksta pēdējā kolonnā (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.

Veiksmīga vai neveiksmīga testa ziņojuma rezultāts tiks parādīts tajā pašā logā:



Lietotāja mediji

Kad e-pasta mediju tips ir konfigurēts, dodieties uz *Lietotāji* > *Lietotāji* sadaļu un rediģējiet lietotāja profilu, lai piešķirtu lietotājam e-pasta mediju. Lietotāja mediju iestatīšanas soļi, kas ir kopīgi visiem mediju tipiem, ir aprakstīti lapā Mediju tipi.

Automatizētie Gmail/Office365 mediju tipi

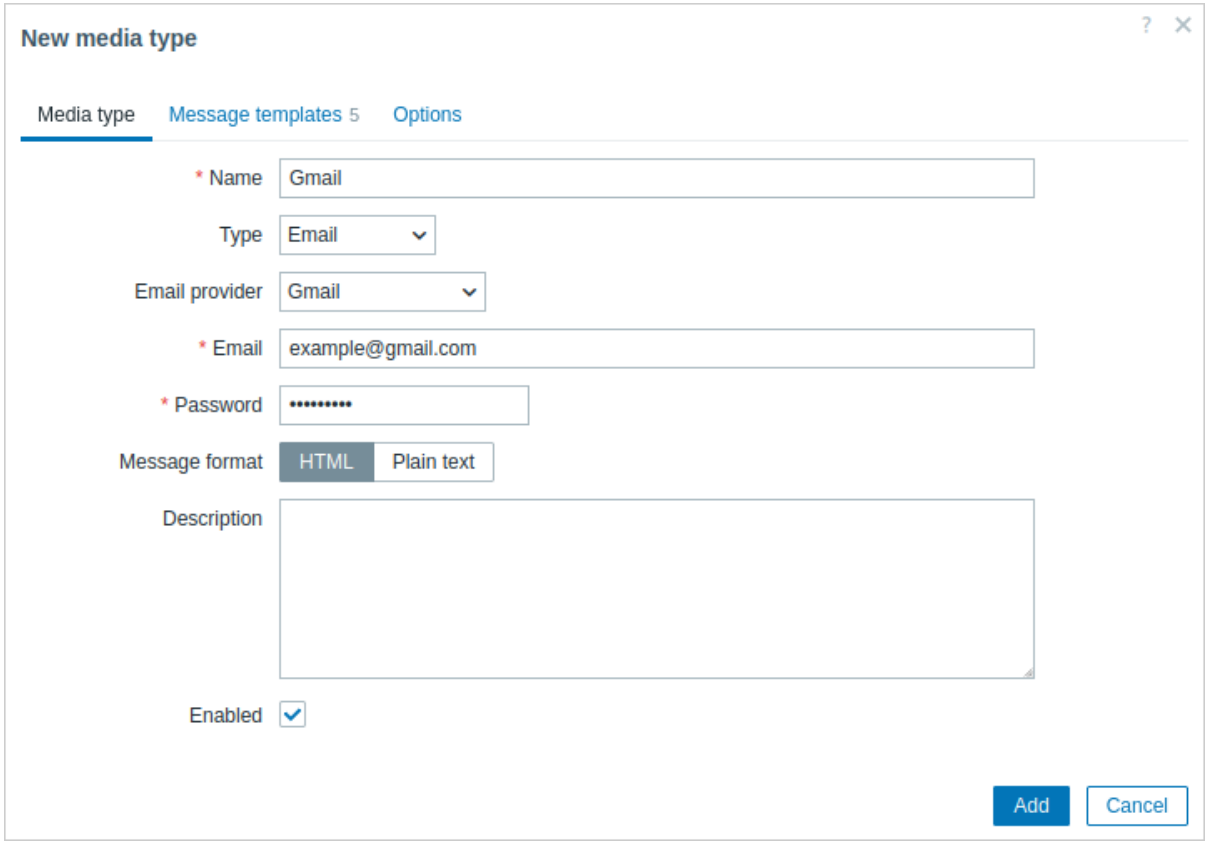
Pārskats

Gmail vai Office365 lietotāji var gūt labumu no automatizētām funkcijām mediju tipa konfigurācijā.

Konfigurācija

E-pasta multivides tipa konfigurācijas laukā *E-pasta pakalpojumu sniedzējs* varat atlasīt iepriekš konfigurētas opcijas Gmail un Office 365.

Atlasot ar Gmail/Office365 saistītās opcijas, lai izveidotu funkcionējošu multivides tipu, jānorāda tikai sūtītāja e-pasta adrese un parole.



Tiklīdz e-pasta adrese un parole ir norādīta, Zabbix var automātiski aizpildīt visus Gmail/Office365 multivides tipiem nepieciešamos iestatījumus ar faktiskajām/ieteicamajām vērtībām, proti, *SMTP serveris*, *SMTP servera ports*, *SMTP helo* un *Savienojuma drošība*. Šīs automatizācijas dēļ šie lauki pat netiek parādīti, tomēr multivides tipu sarakstā varat skatīt SMTP servera un e-pasta informāciju (skatiet kolonnu *Detalizēta informācija*).

Ņemiet vērā arī tālāk norādīto:

- Parole nav nepieciešama releja opcijām.
- Office365 relejam norādītās e-pasta adreses domēna nosaukums tiks izmantots, lai dinamiski aizpildītu SMTP serveri (t. i., `example-com.mail.protection.outlook.com` vērtība "example.com" tiks aizstāta ar faktisko 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
```

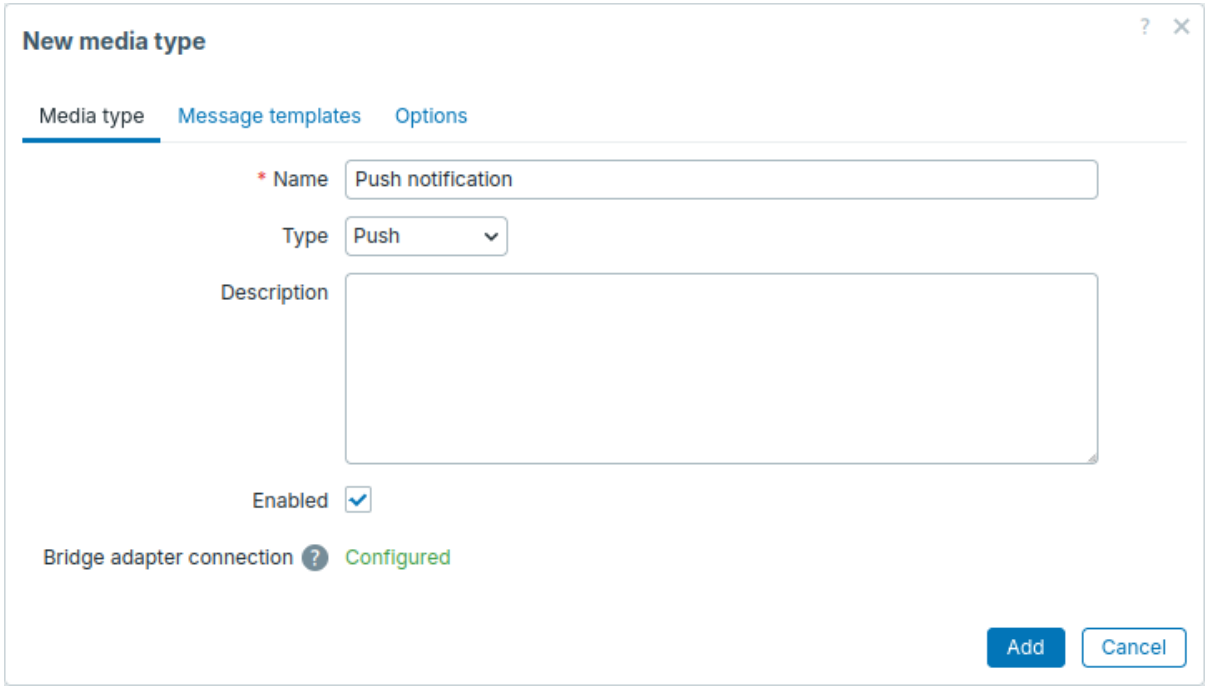
URL ar vaicājuma parametriem vietā, tādējādi nodrošinot saderību ar Office365 personīgajiem kontiem.

Push paziņojumi

Šis multivides tips ļauj Zabbix serverim nosūtīt push paziņojumus uz Zabbix mobilo lietotni jūsu mobilajā ierīcē.

Konfigurācija

Veidojot multivides tipu, kā tā *Type* atlasiet *Push*:



Konfigurācijas veidlapā ir ietverti vispārīgie multivides tipa parametri, ziņojumu veidnes un opcijas.

Veidlapā tiek parādīts arī, vai Zabbix bridge adapteris ir konfigurēts.
 Ja adapteris nav konfigurēts, multivides tipu joprojām var izveidot, taču tas nedarbosies.

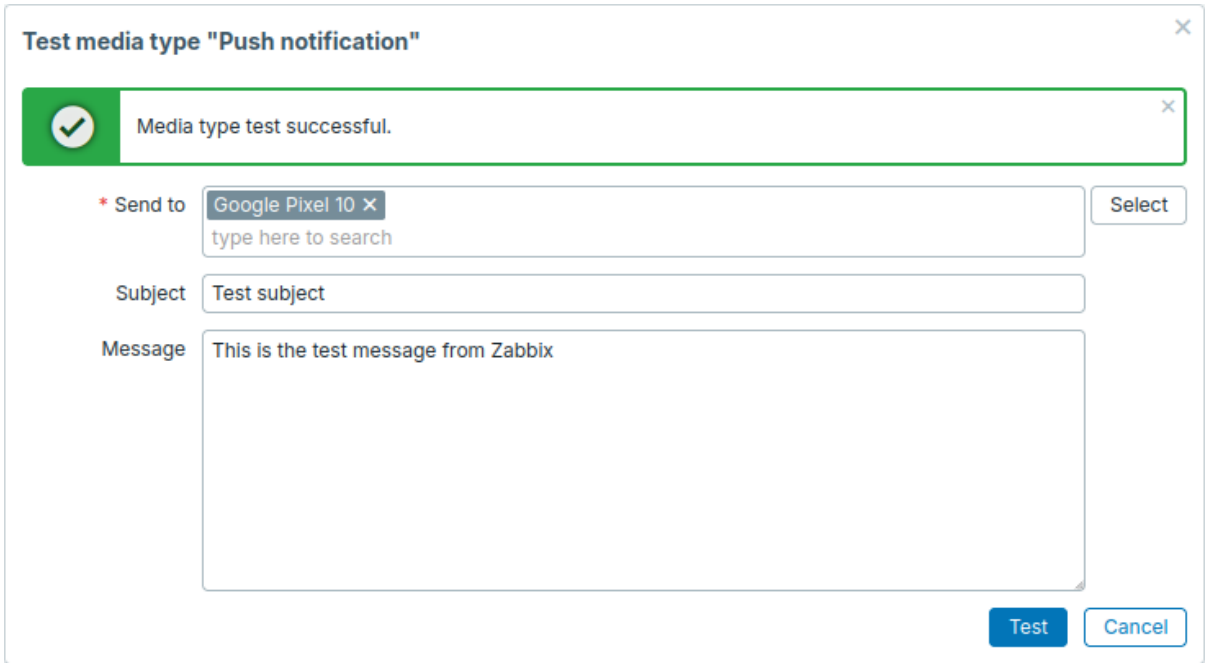
Kad multivides tips ir izveidots un iespējots, pievienojiet ierīci lietotājam, lai multivides tips tiktu automātiski pievienots šim lietotājam (vai manuāli pievienojiet lietotāja multividi). Tas ļauj lietotājam saņemt pašpiegādes paziņojumus savā mobilajā ierīcē.

Testēšana

Lai pārbaudītu, vai šis multivides tips darbojas:

1. Atrodiet multivides tipu multivides tipu sarakstā.
2. Saraksta pēdējā kolonnā noklikšķiniet uz *Testēt*, lai atvērtu testēšanas veidlapu.
3. Laukā *Sūtīt uz* atlasiet ierīci.
4. Noklikšķiniet uz *Testēt*, lai atlasītajai ierīcei nosūtītu testa pašpiegādes paziņojumu.

Tajā pašā logā tiks parādīts ziņojums par veiksmīgu vai neveiksmīgu darbību:



SMS

Pārskats

Zabbix atbalsta SMS ziņojumu sūtīšanu, izmantojot seriālo GSM modemu, kas pievienots Zabbix servera 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 apskatītu 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 šiem GSM modēmiem:

- Siemens MC35
- Teltonika ModemCOM/G10

Lai konfigurētu SMS kā ziņojumu piegādes kanālu, jums ir arī jākonfigurē SMS kā medija veids un lietotājiem jānorāda attiecīgie tālruņu numuri.

Konfigurēšana

Lai konfigurētu SMS kā multivides tipu:

1. Dodieties uz *Alerts* > *Media types*.
2. Noklikšķiniet uz *Create media type* (vai sarakstā ar iepriekš definētajiem multivides tipiem noklikšķiniet uz *SMS*).
3. Formā ievadiet SMS multivides tipa parametrus.

Tālāk norādītie parametri ir specifiski SMS multivides tipam:

Parameter	Description
GSM modem	Iestatiet GSM modema seriālās ierīces nosaukumu. Šeit ievadītais ceļš tiks validēts atbilstoši servera parametram SMSDevices (ja tas ir norādīts).

Detalizētu informāciju par noklusējuma ziņojumu un brīdinājumu apstrādes opciju konfigurēšanu skatiet sadaļā common media type parameters. Ņemiet vērā, ka SMS paziņojumu sūtīšana paralēlā režīmā nav iespējama.

Lietotāja multivide

Kad SMS multivides tips ir konfigurēts, dodieties uz sadaļu *Lietotāji* > *Lietotāji* un rediģējiet lietotāja profilu, lai piešķirtu lietotājam SMS multividi. Lietotāja multivides iestatīšanas darbības, kas ir kopīgas visiem multivides tipiem, ir aprakstītas lapā Multivides tipi.

Pielāgoti brīdinājumu skripti

Pārskats

Ja jūs neesat apmierināts ar esošajiem mediju tipiem brīdinājumu nosūtīšanai, ir alternatīvs veids, kā to izdarīt. Jūs varat izveidot skriptu, kas apstrādās paziņojumu jums vēlamajā veidā.

Pielāgotie brīdinājumu skripti tiek izpildīti Zabbix serverī. Šiem skriptiem jāatrodas direktorijā, kas norādīta servera konfigurācijas faila AlertScriptsPath parametrā.

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.

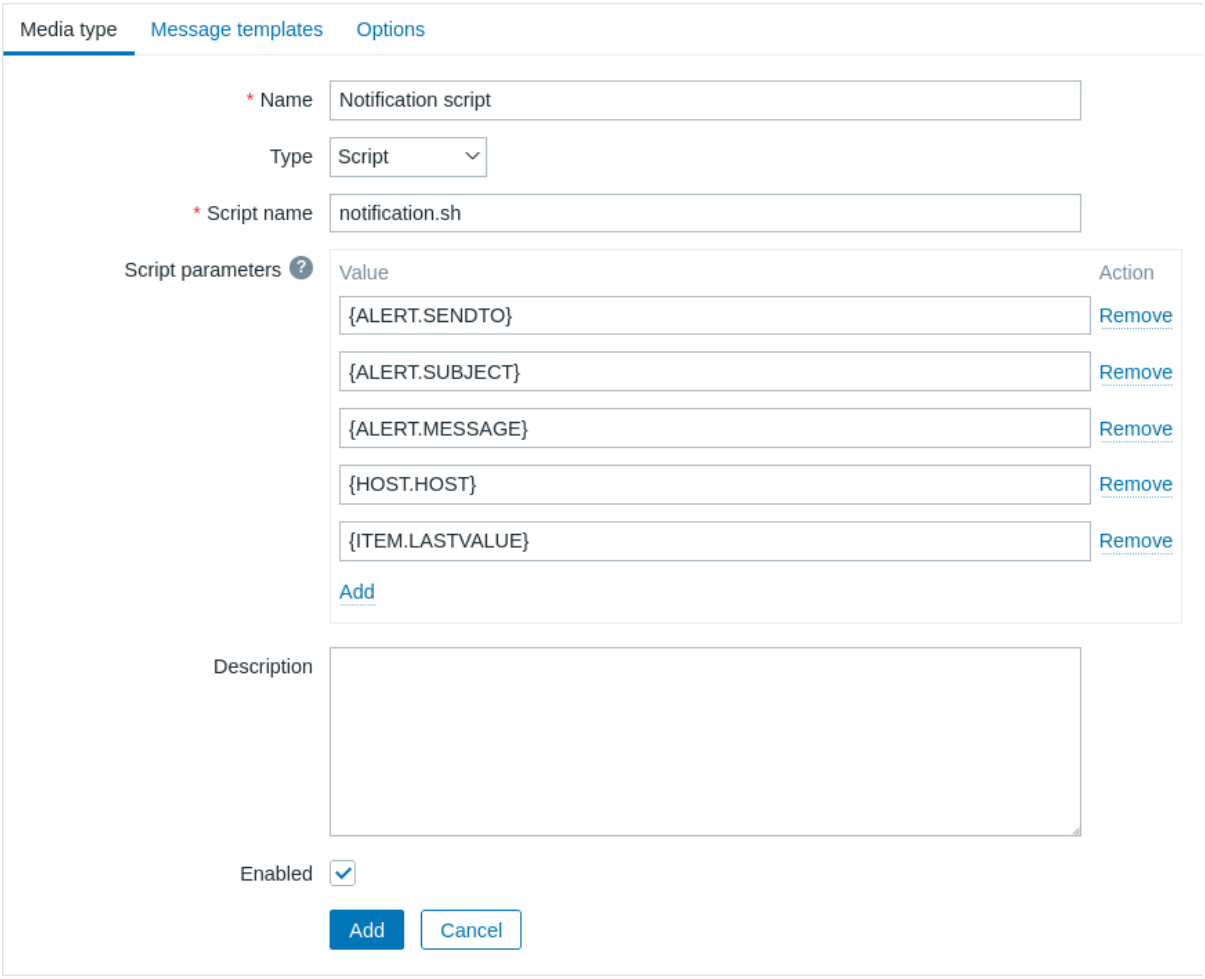
Vides mainīgie skriptam netiek saglabāti vai izveidoti, tāpēc tie ir jāapstrādā tieši.

Konfigurēšana

Lai konfigurētu pielāgotus brīdinājumu skriptus kā multivides tipu:

1. Dodieties uz *Alerts* > *Media types*.
2. Noklikšķiniet uz *Create media type*.
3. Veidlapā ievadiet skripta multivides tipa parametrus.

Cilnē **Media type** ir ietverti vispārīgie multivides tipa atribūti:



Visi obligātie ievades lauki ir atzīmēti ar sarkanu zvaigznīti.

Tālāk norādītie parametri attiecas uz skripta multivides tipu:

Parametrs	Apraksts
Script name	Ievadiet skripta faila nosaukumu (piemēram, notification.sh), kas atrodas direktorijā, kura norādīta servera konfigurācijas parametrā AlertScriptsPath.
Script parameters	Pievienojiet neobligātus skripta parametrus, kas tiks nodoti skriptam kā komandrindas argumenti to definēšanas secībā. Skripta parametri atbalsta makro {ALERT.SENDTO}, {ALERT.SUBJECT}, {ALERT.MESSAGE} un visus makro, kas tiek atbalstīti paziņojumos, kā arī lietotāja makro.

Detalizētu informāciju par noklusējuma ziņojumu un brīdinājumu apstrādes opciju konfigurēšanu skatiet sadaļā common media type parameters.

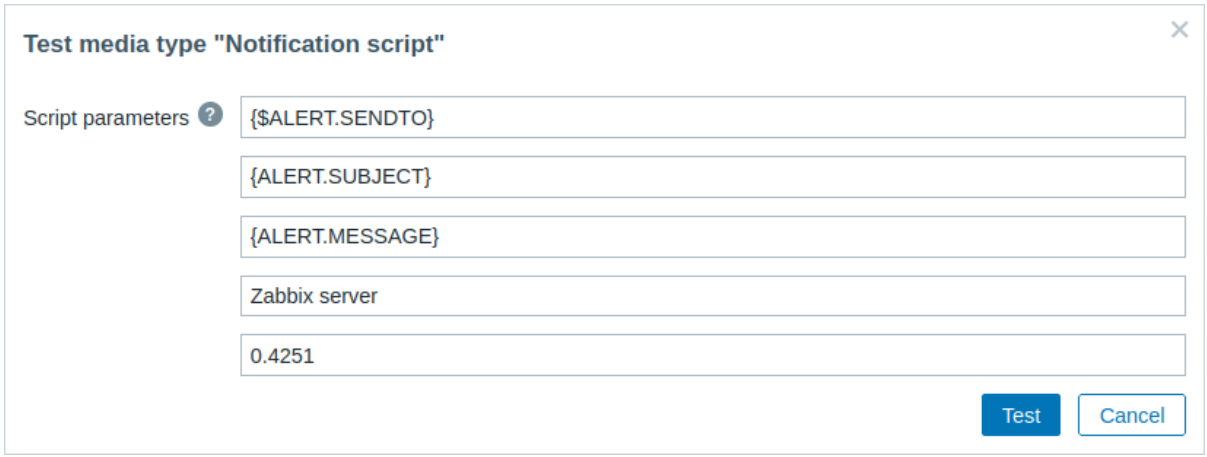
Warning:
Pat ja brīdinājuma skripts neizmanto noklusējuma ziņojumus, šim multivides tipam izmantotajiem darbības tipiem joprojām ir jādefinē ziņojumu veidnes. Pretējā gadījumā paziņojums netiks nosūtīts.

Attention:
Ja ir konfigurēts vairāk nekā viens skripta multivides tips, brīdinājumu apstrādes procesi šos skriptus var apstrādāt paralēli. Kopējo brīdinājumu apstrādes procesu skaitu ierobežo servera konfigurācijas faila parametrs `StartAlerters`.

Testēšana

Lai pārbaudītu konfigurētu skripta mediju tipu:

1. Atrodiet attiecīgo skriptu sarakstā ar mediju tipiem.
2. Noklikšķiniet uz *Test* saraksta pēdējā kolonnā; testēšanas forma tiks atvērta uznirstošajā logā. Testēšanas forma saturēs tikpat daudz parametru, cik ir konfigurēts skripta mediju tipam.
3. Ja nepieciešams, rediģējiet skripta parametru vērtības. Rediģēšana ietekmē tikai testēšanas procedūru; faktiskās vērtības netiks mainītas.
4. Noklikšķiniet uz *Test*.



Note:
Pārbaudot konfigurētu skripta mediju tipu, {ALERT.SENDTO}, {ALERT.SUBJECT}, {ALERT.MESSAGE} un lietotāja makrosi tiks aizstāti ar to vērtībām, taču makrosi, kas ir saistīti ar notikumiem (piem., {HOST.HOST}, {ITEM.LASTVALUE} u.c.), netiks aizstāti, jo testēšanas laikā nav saistīta notikuma, no kura iegūt detaļas. Ņemiet vērā, ka makrosi {ALERT.SUBJECT} un {ALERT.MESSAGE} iekšpusē arī netiks aizstāti. Piemēram, ja {ALERT.SUBJECT} vērtība ir veidota kā "Problem: {EVENT.NAME}", tad makross {EVENT.NAME} netiks aizstāts.

Lietotāja mediji

Kad mediju tips ir konfigurēts, dodieties uz sadaļu *Users* > *Users* un rediģējiet lietotāja profilu, piešķirot šim lietotājam šo mediju tipu. Lietotāja mediju iestatīšanas darbības, kas ir kopīgas visiem mediju tipiem, ir aprakstītas lapā Media types.

Ņemiet vērā, ka, definējot lietotāja medijus, lauks *Send to* nevar būt tukšs. Ja šis lauks netiek izmantots brīdinājuma skriptā, ievadiet jebkuru atbalstīto rakstzīmju kombināciju, lai apietu validācijas prasības.

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 tālāk norādītās integrācijas, kas ļauj izmantot iepriekš definētus webhook multivides 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 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 Zabbix paziņojumus pārveidotu par Spiceworks pieteikumiem, izveidojiet e-pasta multivides tipu un norādītā Zabbix lietotāja profila iestatījumos ievadiet Spiceworks palīdzības dienesta e-pasta adresi (piemēram, help@zabbix.on.spiceworks.com).

Konfigurēšana

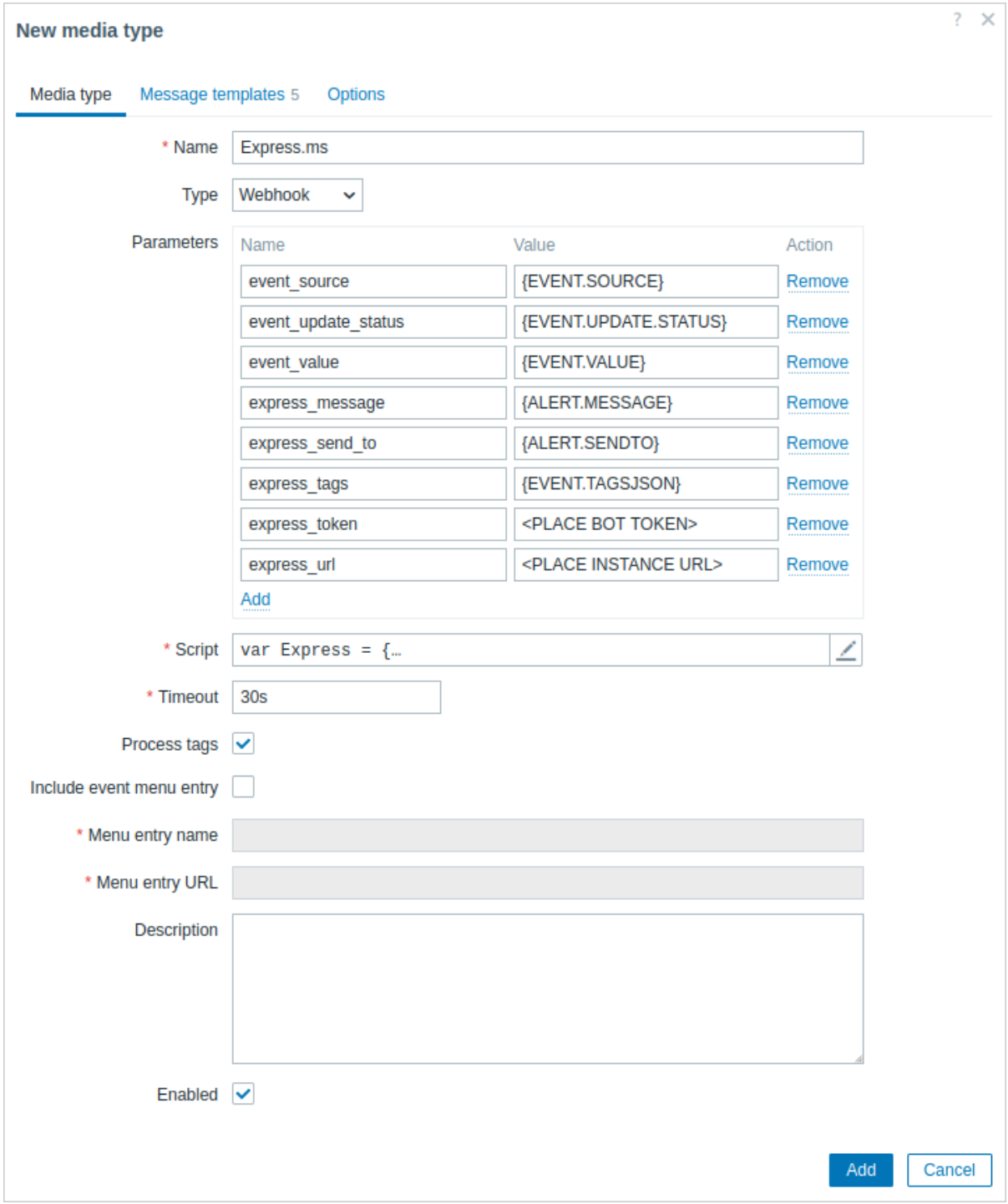
Lai sāktu izmantot webhook integrāciju:

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 tiks parādīts multivides tipu sarakstā.
3. Konfigurējiet webhook saskaņā ar `Readme.md` failā sniegtajiem norādījumiem (lai ātri piekļūtu `Readme.md`, varat noklikšķināt uz webhook nosaukuma iepriekš).

Lai izveidotu pielāgotu webhook no jauna:

1. Dodieties uz *Brīdinājumi* > *Multivides tipi*.
2. Noklikšķiniet uz *Izveidot multivides tipu*.
3. Formā ievadiet webhook multivides tipa parametrus.

Cilnē **Multivides tips** ir ietverti dažādi šim multivides tipam specifiski atribūti:



Visi obligātie ievades lauki ir atzīmēti ar sarkanu zvaigznīti.

Tālāk norādītie parametri ir specifiski webhook multivides tipam:

Parametrs	Apraksts
Parametri	Norādiet webhook mainīgos kā atribūtu un vērtību 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 saglabāt vai noņemt. Webhook parametri atbalsta lietotāja makro, visus makro, kas tiek atbalstīti problēmu paziņojumos, kā arī makro {ALERT.SENDTO}, {ALERT.SUBJECT} un {ALERT.MESSAGE}. Ja norādāt HTTP starpniekserveri, lauks atbalsta tādu pašu funkcionalitāti kā vienuma konfigurācijas HTTP starpniekservera lauks. Starpniekservera virknes sākumā var pievienot `[scheme]://`, lai norādītu izmantotā starpniekservera veidu (piemēram, https, socks4, socks5; skatiet dokumentāciju).
Skripts	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 parametra un vērtības pārus. Vērtības jāpārveido JSON objektos, izmantojot JSON.parse() metodi, piemēram: `var params = JSON.parse(value);`. Kodam ir piekļuve visiem parametriem; tas var veikt HTTP GET, POST, PUT un DELETE pieprasījumus, atbalstīt papildu metodes, piemēram, CONNECT, PATCH, HEAD, OPTIONS un TRACE, kā arī kontrolēt HTTP galvenes un pieprasījuma pamattekstu. Skriptā jābūt return operatoram, pretējā gadījumā tas nebūs derīgs. Tas var atgriezt statusu OK kopā ar neobligātu tagu un tagu vērtību sarakstu (skatiet opciju *Apstrādāt tagus*) vai kļūdas virkni. Atkopšanas notikumus (neatkarīgi no tā, vai tie ģenerēti automātiski vai manuālas aizvēršanas rezultātā) izveido serveris, un tie ietver atrisinātos notikuma tagus (tostarp tagus, kas mantoti no veidnēm, hostiem un trigeriem). Webhook skripti tiek izpildīti pēc brīdinājuma izveides; tādēļ webhook skripta atgrieztie tagi tiek pievienoti tikai pēc sākotnējās brīdinājuma izveides un nebūs pieejami sākotnējā problēmas ziņojuma vai tūlītējā atkopšanas ziņojuma {EVENT.TAGS} un `{EVENT.RECOVERY.TAGS}` makro. *Piezīme*: Ieteicams izmantot lokālos mainīgos (piemēram, `var local = 1`), nevis globālos mainīgos (piemēram, `global = 1`), lai nodrošinātu, ka katrs skripts darbojas ar saviem datiem, un izvairītos no konfliktiem vienlaicīgu izsaukumu laikā (skatiet zināmās problēmas). Skatiet arī: Webhook izstrādes vadlīnijas, Webhook skriptu piemēri, Papildu JavaScript objekti.
Noildze	JavaScript izpildes noildze (1–60 s, pēc noklusējuma 30 s).<brTiek atbalstīti laika sufiksi (piemēram, `30s`, `1m`).
Apstrādāt tagus	Atzīmējiet izvēles rūtiņu, lai atgriezto JSON rekvizītu vērtības apstrādātu kā tagus. Šie tagi tiek pievienoti visiem esošajiem problēmas tagiem. Ņemiet vērā, ka, izmantojot webhook tagus, webhook jāatgriež JSON objekts, kas satur vismaz tukšu tagu objektu: `var result = {tags: {}};` Atgriežamo tagu piemēri: *jira-id:prod-1234*, *responsible:John Smith*, *processed:<no value>*
Iekļaut notikumu izvēlnes ierakstu	Atzīmējiet izvēles rūtiņu, lai notikumu izvēlnē iekļautu ierakstu, kas novirza uz izveidoto ārējo pieteikumu. Ieraksts tiks iekļauts katram iespējotam webhook, kuram šī izvēles rūtiņa ir atzīmēta. Ņemiet vērā: ja parametri *Izvēlnes ieraksta nosaukums* un *Izvēlnes ieraksta URL* satur kādu no {EVENT.TAGS.<tag name>} makro, ieraksts tiks iekļauts tikai tad, ja šos makro var atrisināt (tas ir, notikumam ir definēti šie tagi). Ja šī opcija ir atzīmēta, webhook nevajadzētu izmantot paziņojumu sūtīšanai dažādiem lietotājiem (tā vietā apsveriet iespēju izveidot īpašu lietotāju), kā arī nevajadzētu to izmantot vairākās brīdinājumu darbībās vienam problēmas notikumam.
Izvēlnes ieraksta nosaukums	Norādiet izvēlnes ieraksta nosaukumu. Tiek atbalstīts {EVENT.TAGS.<tag name>} makro. Šis lauks ir obligāts tikai tad, ja ir atzīmēta opcija *Iekļaut notikumu izvēlnes ierakstu*.
Izvēlnes ieraksta URL	Norādiet izvēlnes ieraksta pamatā esošo URL. Tiek atbalstīts {EVENT.TAGS.<tag name>} makro. Šis lauks ir obligāts tikai tad, ja ir atzīmēta opcija *Iekļaut notikumu izvēlnes ierakstu*.

Detalizētu informāciju par noklusējuma ziņojumu un brīdinājumu apstrādes opciju konfigurēšanu skatiet sadaļā kopīgie multivides tipa parametri.

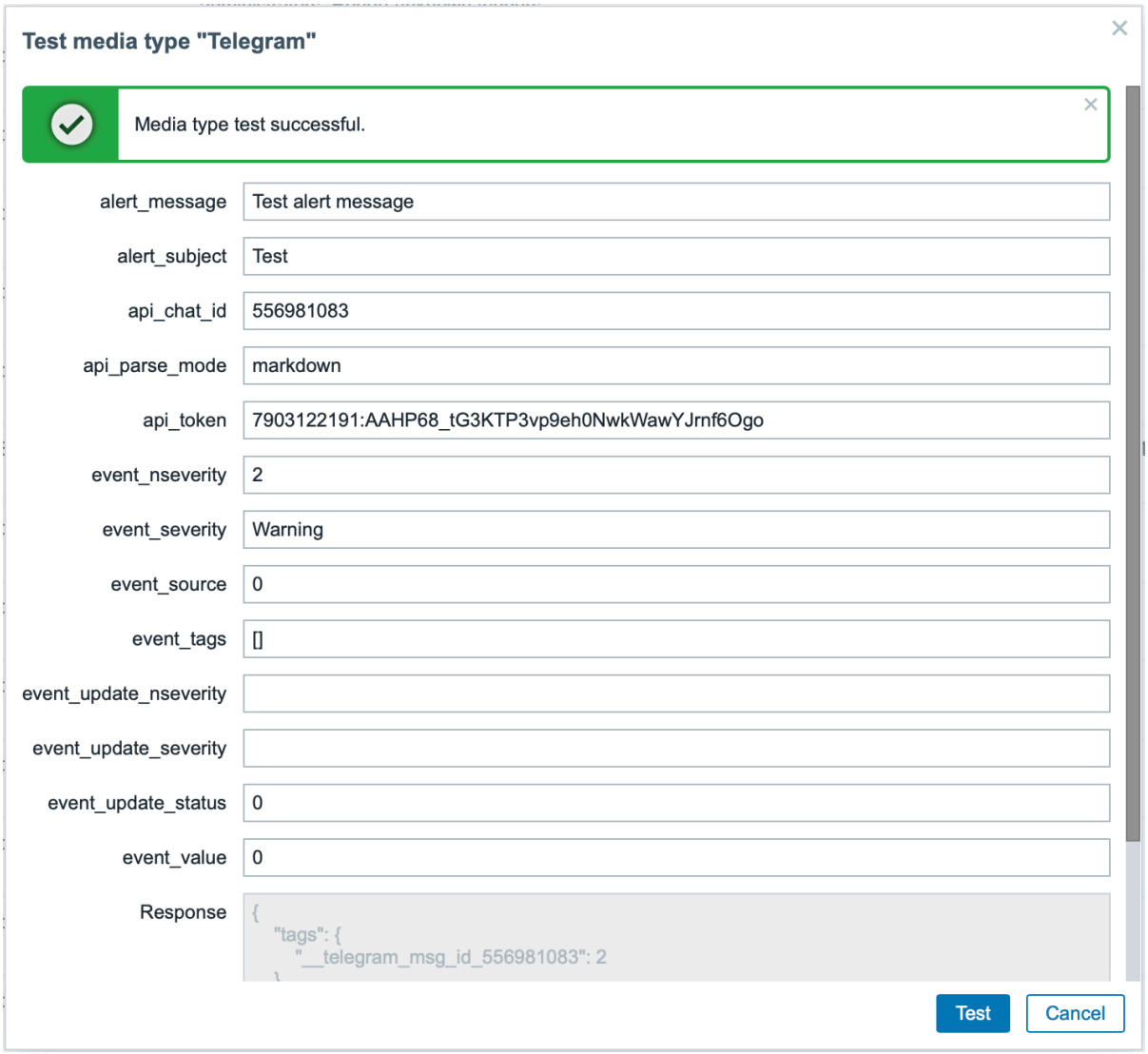
Warning:
Pat ja webhook neizmanto noklusējuma ziņojumus, ziņojumu veidnes darbību tipiem, ko izmanto šis webhook, joprojām ir jādefinē.

Testēšana

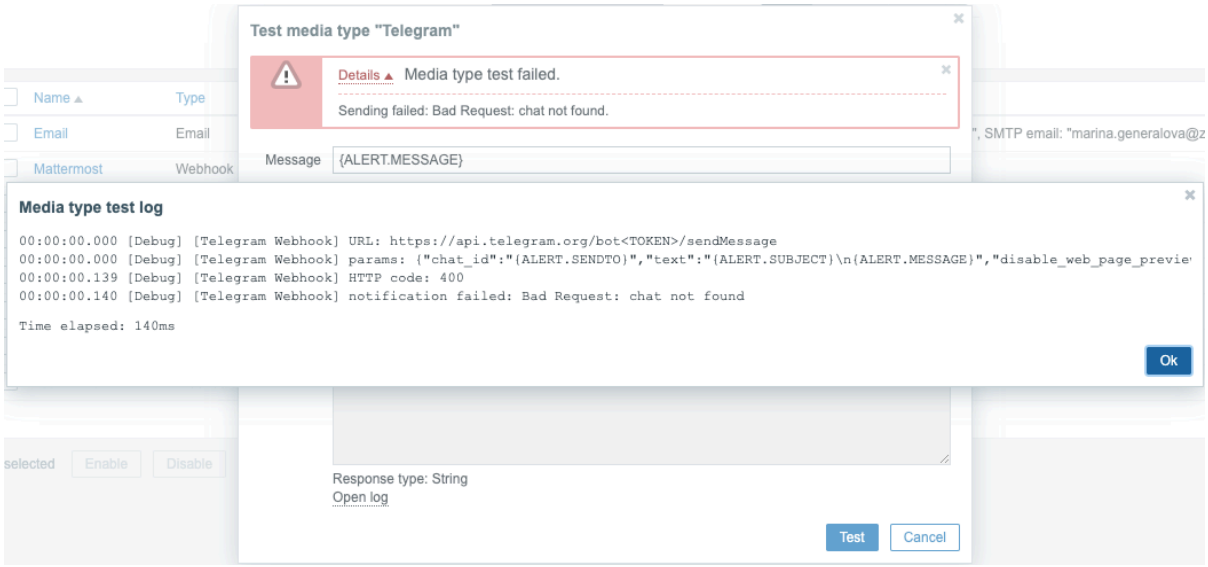
Lai pārbaudītu konfigurētu webhook multivides tipu:

1. Atrodiet attiecīgo webhook multivides tipu sarakstā.
2. Saraksta pēdējā kolonnā noklikšķiniet uz *Test* (tiks atvērts testēšanas logs).
3. Pēc nepieciešamības rediģējiet webhook parametru vērtības.
   Aizstājiet makro ar piemēra vērtībām; pretējā gadījumā makro 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 multivides tipa testa žurnāla ierakstus, neaizverot 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 multivide

Kad multivides tips ir konfigurēts, dodieties uz sadaļu *Users* > *Users* un piešķiriet webhook multividi esošam lietotājam vai izveidojiet jaunu lietotāju, kas pārstāvēs webhook.

Darbības lietotāja multivides iestatīšanai esošam lietotājam, kas ir kopīgas visiem multivides tipiem, ir aprakstītas lapā Media types.

Ja webhook izmanto tagus, lai saglabātu biļetes\ziņojuma ID, nepiešķiriet to pašu webhook kā multividi dažādiem lietotājiem, jo tas var izraisīt webhook kļūdas (tas attiecas uz lielāko daļu webhook, kas izmanto opciju *Include event menu entry*). Šādā gadījumā ieteicams izveidot īpašu lietotāju, kas pārstāvēs webhook:

1. Pēc webhook multivides tipa konfigurēšanas dodieties uz sadaļu *Users* > *Users* un izveidojiet īpašu Zabbix lietotāju, kas pārstāvēs webhook, piemēram, lietotāju ar lietotājvārdu *Slack* Slack webhook vajadzībām. Visus iestatījumus, izņemot multividi, var atstāt pēc noklusējuma, jo šis lietotājs nepieteiksies Zabbix.
2. Lietotāja profilā dodieties uz cilni *Media* un pievienojiet webhook, norādot nepieciešamo kontaktinformāciju. Ja webhook neizmanto lauku *Send to*, ievadiet jebkādu atbalstīto rakstzīmju kombināciju, lai apietu validācijas prasības.
3. Piešķiriet šim lietotājam vismaz lasīšanas atļaujas visiem hostiem, kuriem tam jānosūta brīdinājumi.

Konfigurējot brīdinājuma darbību, laukā *Send to users* sadaļā Operation details pievienojiet šo lietotāju — tādējādi Zabbix izmantos webhook paziņojumiem no šīs darbības.

Brīdinājumu darbību konfigurēšana

Darbības nosaka, kuri paziņojumi ir jānosūta, izmantojot webhook. Soļi darbību konfigurēšanai, kas saistīti ar 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 tā paša webhook izmantošanas vairākās brīdinājumu darbībās vienam problēmas notikumam. Ja {EVENT.TAGS.<tag name>} pastāv un tiek atjaunināts webhook, tā rezultējošā vērtība būs nenoteikta. Lai no tā izvairītos, webhook izmantojiet jaunu taga nosaukumu atjaunināto vērtību saglabāšanai. 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 viens webhook var tikt izmantots vairākās vienas un tās pašas darbības operācijās vai eskalācijas soļos, kā arī dažādās darbībās, kas netiks aktivizētas ar to pašu problēmas notikumu atšķirīgu nosacījumu dēļ.
- Izmantojot webhook darbībās iekšējiem notikumiem, noteikti atzīmējiet izvēles rūtiņu *Custom message* un darbības operācijas konfigurācijā definējiet pielāgotu ziņojumu. Pretējā gadījumā paziņojums netiks nosūtīts.

Webhook skriptu piemēri

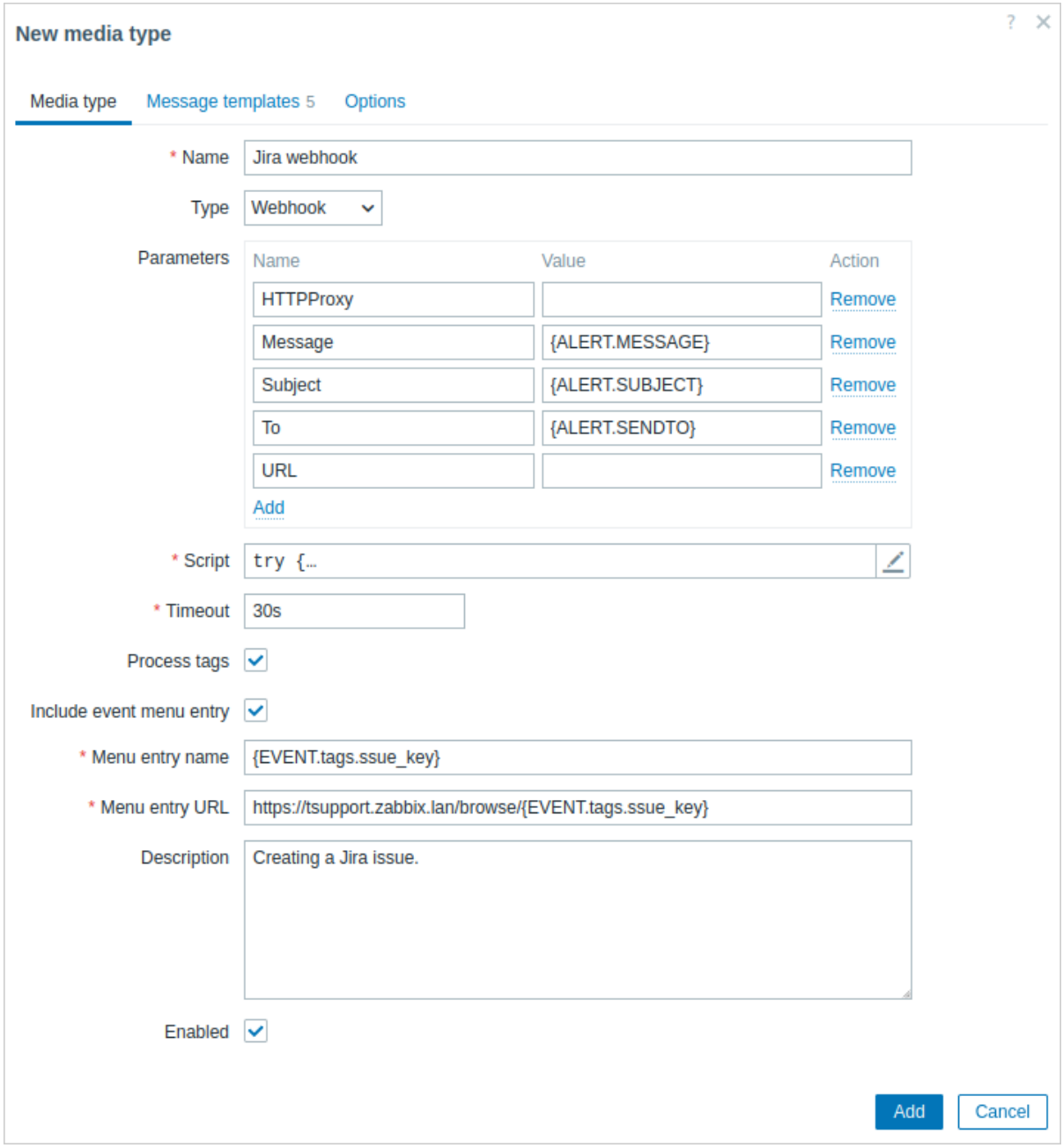
Pārskats

Lai gan Zabbix piedāvā lielu skaitu webhook integrāciju, kas pieejamas uzreiz pēc noklusējuma, iespējams, vēlēsieties izveidot arī savus webhookus. Šajā sadaļā ir sniegti pielāgotu webhook skriptu piemēri (izmantojami parametrā *Script*). Citu webhook parametru aprakstu skatiet sadaļā webhook.

Attention:
Preprocessing JavaScript neizmantojiet nedeklarētus piešķīrumus. Lai deklarētu lokālos mainīgos, izmantojiet `var`.

Atjaunošanas notikumi (neatkarīgi no tā, vai tie tiek ģenerēti automātiski vai pēc manuālas aizvēršanas) ietver atrisināto notikumu tagus (tostarp tagus, kas mantoti no veidnēm, hostiem un trigeriem). Webhook skripti tiek izpildīti pēc brīdinājuma izveides; tādēļ webhook skripta atgrieztie tagi tiek lietoti pēc sākotnējās brīdinājuma izveides un var nebūt klātesoši brīdinājumā, kas izraisīja webhook. Ja integrācijai ir nepieciešams, lai webhook ģenerētie tagi būtu klātesoši sākotnējā paziņojumā, iegūstiet notikuma tagus no servera (piemēram, izmantojot Event API) vai saglabājiet webhook ģenerētos tagus 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();
    }

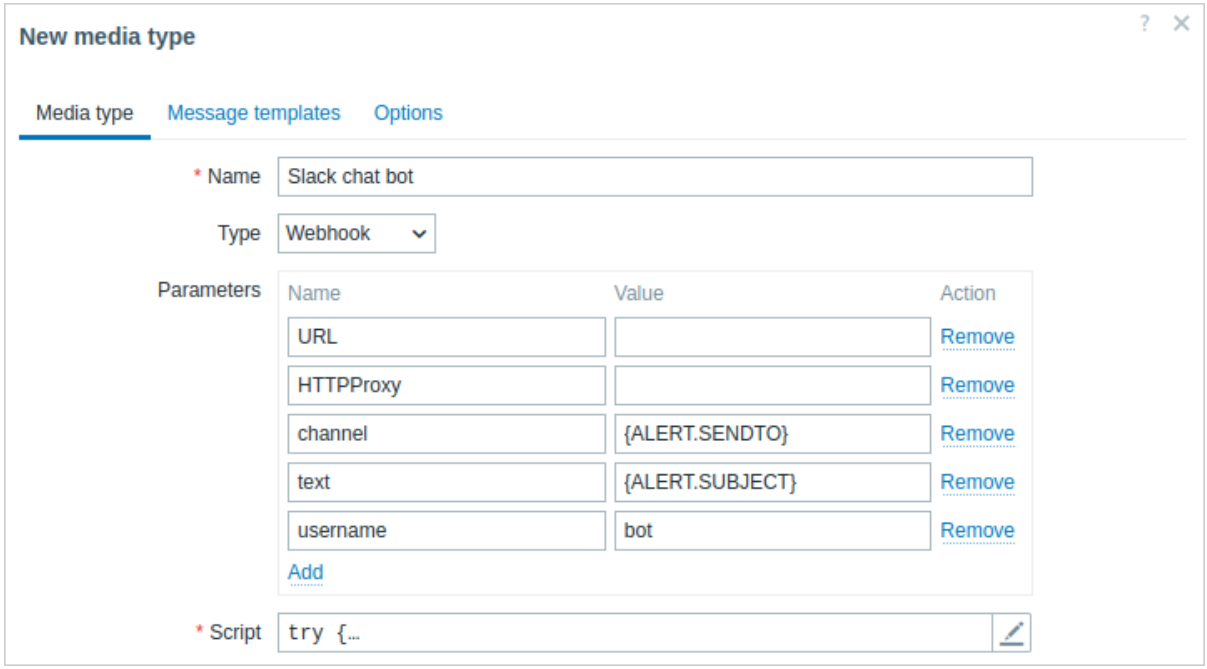
    resp = JSON.parse(resp);
    result.tags.issue_id = resp.id;
    result.tags.issue_key = resp.key;

    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.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;
}
```

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:

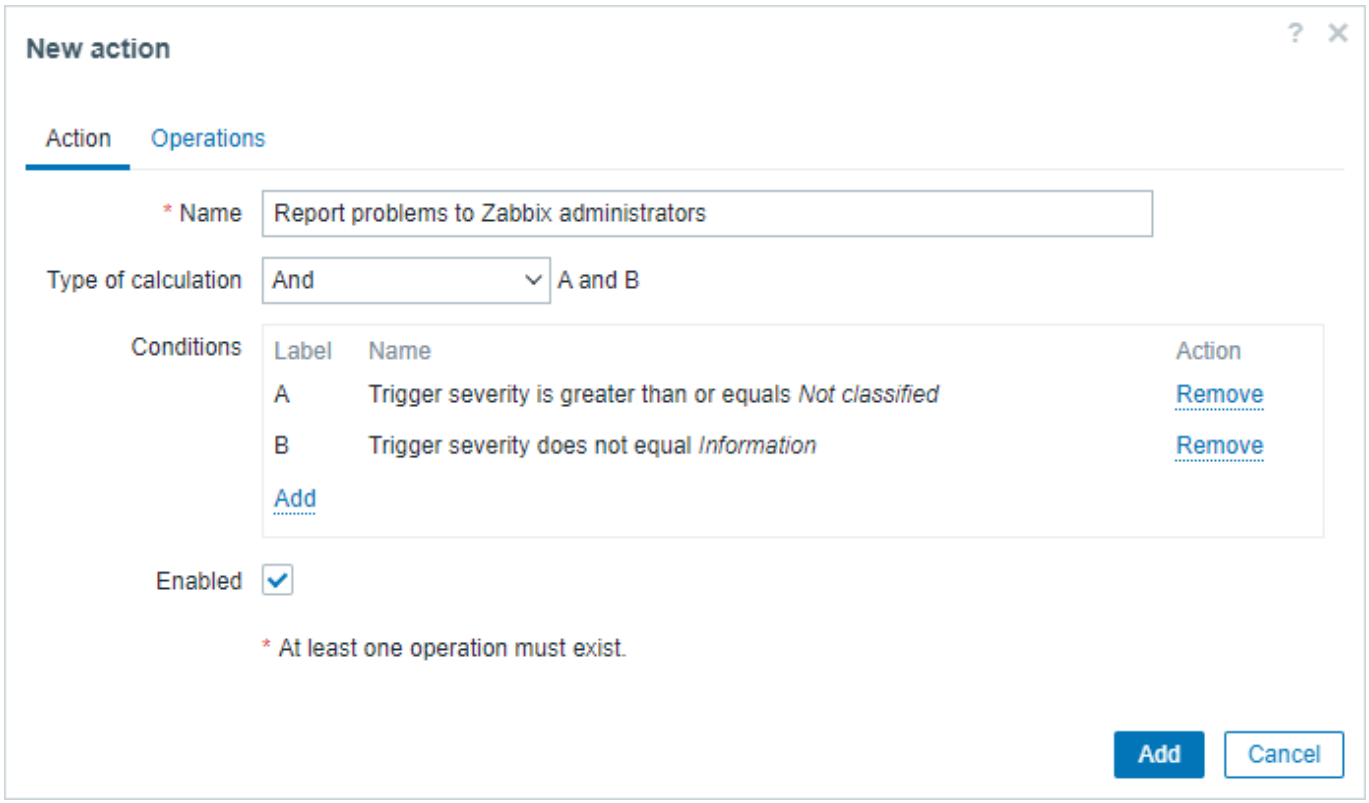
- Lietotāja piekļuve pakalpojumu darbībām ir atkarīga no piekļuves tiesībām pakalpojumiem, kas piešķirtas lietotāja lomai
- Pakalpojumu darbības atbalsta atšķirīgu nosacījumu kopu

Darbības konfigurēšana

Lai konfigurētu darbību, rīkojieties šādi:

1. Dodieties uz *Alerts* > *Actions* un apakšizvēlnē atlasiet nepieciešamo darbības tipu (vēlāk varat pārslēgties uz citu tipu, izmantojot virsraksta nolaižamo izvēlni).
2. Noklikšķiniet uz *Create action*.
3. Norādiet darbības nosaukumu.
4. Izvēlieties nosacījumus, pēc kuriem tiek veiktas operācijas.
5. Izvēlieties veicamās operācijas.

Vispārīgie darbības atribūti:



Visi obligātie ievades lauki ir atzīmēti ar sarkanu zvaigznīti.

Parameter	Description
Name	Unikāls darbības nosaukums.

Parameter	Description
Type of calculation	Atlasiet novērtēšanas opciju darbības nosacījumiem (ja ir vairāk nekā viens nosacījums): **And** - jāizpildās visiem nosacījumiem. **Or** - pietiek, ja izpildās viens nosacījums. **And/Or** - abu kombinācija: AND ar dažādiem nosacījumu tipiem un OR ar vienu un to pašu nosacījumu tipu. **Custom expression** - lietotāja definēta aprēķina formula darbības nosacījumu novērtēšanai.
Conditions	Darbības nosacījumu saraksts. Noklikšķiniet uz *Add*, lai pievienotu jaunu nosacījumu. Ja nosacījumi nav konfigurēti, darbība tiks izpildīta katram notikumam, kas atbilst konfigurētajam darbības tipam.
Enabled	Atzīmējiet izvēles rūtiņu, lai iespējotu darbību. Pretējā gadījumā tā būs atspējota.

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, ko izslēgt. **vienāds ar** - notikums pieder šim pakalpojumam. **nav vienāds ar** - notikums nepieder šim pakalpojumam. Norādot vecāka pakalpojumu, netieši tiek atlasīti visi pakārtotie pakalpojumi. Lai norādītu tikai vecāka pakalpojumu, visi ligzdotie pakalpojumi papildus jāiestata ar operatoru **nav vienāds ar**.
Pakalpojuma nosaukums	satur nesatur	Norādiet virkni pakalpojuma nosaukumā vai virkni, ko izslēgt. **satur** - notikumu ģenerē pakalpojums, kura nosaukumā ir šī virkne. **nesatur** - šo virkni nevar atrast pakalpojuma nosaukumā.
Pakalpojuma taga nosaukums	vienāds ar nav vienāds ar satur nesatur	Norādiet notikuma tagu vai notikuma tagu, ko izslēgt. Pakalpojuma notikumu tagus var definēt pakalpojuma konfigurācijas sadaļā *Tags*. **vienāds ar** - notikumam ir šis tags. **nav vienāds ar** - notikumam nav šī taga. **satur** - notikumam ir tags, kas satur šo virkni. **nesatur** - notikumam nav taga, kas satur šo virkni.
Pakalpojuma taga vērtība	vienāds ar nav vienāds ar satur nesatur	Norādiet notikuma taga un vērtības kombināciju vai taga un vērtības kombināciju, ko izslēgt. Pakalpojuma notikumu tagus var definēt pakalpojuma konfigurācijas sadaļā *Tags*. **vienāds ar** - notikumam ir šis tags un vērtība. **nav vienāds ar** - notikumam nav šī taga un vērtības. **satur** - notikumam ir tags un vērtība, kas satur šīs virknes. **nesatur** - notikumam nav taga un vērtības, kas satur šīs virknes.

Attention:
Pārliecinieties, ka pakalpojumu darbībām ir definētas ziņojumu veidnes izvēlnē *Alerts* > *Media types*. Pretējā gadījumā paziņojumi netiks nosūtīti.

Atklāšanas darbības

Atklāšanā balstītos notikumos var izmantot šādus nosacījumus:

Nosacījuma tips	Atbalstītie operatori	Apraksts
Hosts IP adrese	equals does not equal	Norādiet atklātā hosts IP adrešu diapazonu vai izslēdzamo diapazonu. **equals** - hosts IP adrese ir diapazonā. **does not equal** - hosts IP adrese nav diapazonā. Var izmantot šādus formātus: Atsevišķa 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 Saraksta formātā var izmantot atstarpes.
Pakalpojuma tips	equals does not equal	Norādiet atklātā pakalpojuma tipu vai izslēdzamo pakalpojuma tipu. **equals** - atbilst atklātajam pakalpojumam. **does not equal** - 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.
Pakalpojuma ports	equals does not equal	Norādiet atklātā pakalpojuma TCP portu diapazonu vai izslēdzamo diapazonu. **equals** - pakalpojuma ports ir diapazonā. **does not equal** - pakalpojuma ports nav diapazonā.
Atklāšanas noteikums	equals does not equal	Norādiet atklāšanas noteikumu vai izslēdzamo atklāšanas noteikumu. **equals** - tiek izmantots šis atklāšanas noteikums. **does not equal** - tiek izmantots jebkurš cits atklāšanas noteikums, izņemot šo.
Atklāšanas pārbaude	equals does not equal	Norādiet atklāšanas pārbaudi vai izslēdzamo atklāšanas pārbaudi. **equals** - tiek izmantota šī atklāšanas pārbaude. **does not equal** - tiek izmantota jebkura cita atklāšanas pārbaude, izņemot šo.
Atklāšanas objekts	equals	Norādiet atklāto objektu. **equals** - vienāds ar atklāto objektu (ierīci vai pakalpojumu).
Atklāšanas statuss	equals	**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'.
Darbības laiks/ dīkstāves laiks	is greater than or equals is less than or equals	Darbības laiks notikumiem 'Host Up' un 'Service Up'. Dīkstāves laiks notikumiem 'Host Down' un 'Service Down'. **is greater than or equals** - lielāks vai vienāds ar. Parametrs tiek norādīts sekundēs. **is less than or equals** - mazāks vai vienāds ar. Parametrs tiek norādīts sekundēs.
Saņemtā vērtība	equals does not equal is greater than or equals is less than or equals contains does not contain	Norādiet vērtību, kas saņemta no aģenta (Zabbix, SNMP) pārbaudes atklāšanas noteikumā. Tiek veikta virkņu salīdzināšana. Ja noteikumam ir konfigurētas vairākas Zabbix aģenta vai SNMP pārbaudes, tiek pārbaudītas katras pārbaudes saņemtās vērtības (katra pārbaude ģenerē jaunu notikumu, kas tiek salīdzināts ar visiem nosacījumiem). **equals** - vienāda ar vērtību. **does not equal** - nav vienāda ar vērtību. **is greater than or equals** - lielāka vai vienāda ar vērtību. **is less than or equals** - mazāka vai vienāda ar vērtību. **contains** - satur apakšvirkni. Parametrs tiek norādīts kā virkne. **does not contain** - nesatur apakšvirkni. Parametrs tiek norādīts kā virkne.
Starpniekserveris	equals does not equal	Norādiet starpniekserveri vai izslēdzamo starpniekserveri. **equals** - tiek izmantots šis starpniekserveris. **does not equal** - tiek izmantots jebkurš cits starpniekserveris, izņemot šo.

Nosacījuma tips	Atbalstītie operatori	Apraksts
Starpniekserveru grupa	equals does not equal	Norādiet starpniekserveru grupu vai izslēdzamo starpniekserveru grupu. **equals** - tiek izmantota šī starpniekserveru grupa. **does not equal** - tiek izmantota jebkura cita starpniekserveru grupa, izņemot šo.

Note:
Pakalpojumu pārbaudes atklāšanas noteikumā, kuru rezultātā tiek ģenerēti atklāšanas notikumi, nenotiek vienlaikus. Tāpēc, ja darbībā nosacījumiem `Service type`, `Service port` vai `Received value` ir konfigurētas **vairākas** vērtības, tās tiks salīdzinātas ar vienu atklāšanas notikumu vienlaikus, bet **ne** ar vairākiem notikumiem vienlaikus. Tādēļ darbības ar vairākām vērtībām vieniem un tiem pašiem pārbaudes tipiem var netikt izpildītas pareizi.

Automātiskās reģistrācijas darbības

Tālāk norādītos nosacījumus var izmantot darbībās, kuru pamatā ir aktīvā aģenta automātiskā reģistrācija:

Nosacījuma veids	Atbalstītie operatori	Apraksts
Hosts metadati	satur nesatur atbilst neatbilst	Norādiet hosts metadatus vai izslēdzamos hosts metadatus. **satur** - hosts metadati satur virkni. **nesatur** - hosts metadati nesatur virkni. Hosts metadatus var norādīt aģenta konfigurācijas failā. **atbilst** - hosts metadati atbilst regulārajai izteiksmei. **neatbilst** - hosts metadati neatbilst regulārajai izteiksmei.
Hosts nosaukums	satur nesatur atbilst neatbilst	Norādiet hosts nosaukumu vai izslēdzamo hosts nosaukumu. **satur** - hosts nosaukums satur virkni. **nesatur** - hosts nosaukums nesatur virkni. **atbilst** - hosts nosaukums atbilst regulārajai izteiksmei. **neatbilst** - hosts 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 (vai serveris), izņemot šo starpniekserveri.
Starpniekserveru grupa	ir vienāda ar nav vienāda ar	Norādiet starpniekserveru grupu vai izslēdzamo starpniekserveru grupu. **ir vienāda ar** - tiek izmantota šī starpniekserveru grupa. **nav vienāda ar** - tiek izmantota jebkura cita starpniekserveru grupa, 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

Condition type	Supported operators	Description
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).".

Operācijas

Pārskats

Visiem notikumiem var definēt šādas darbības:

- Nosūtīt ziņojumu
- Izpildīt attālinātu 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.

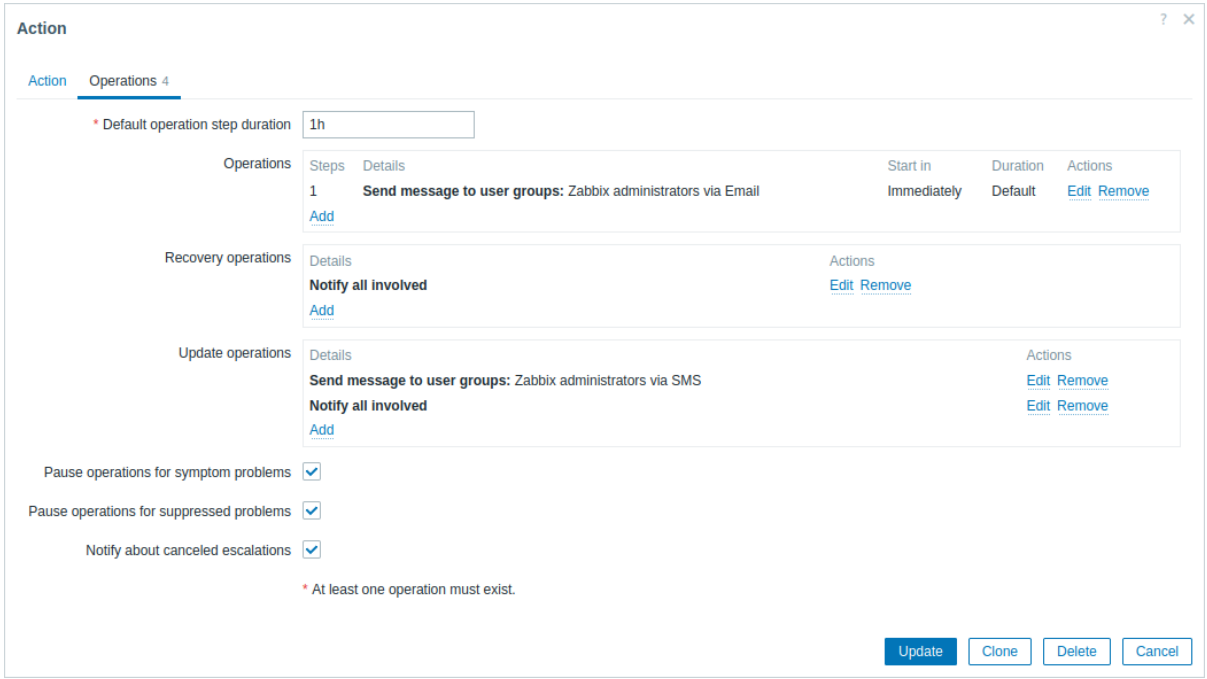
Atklāšanas un autoreģistrācijas notikumiem ir pieejamas papildu darbības:

- 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

Darbības konfigurēšana

Lai konfigurētu darbību:

1. Darbības konfigurācijā atveriet cilni *Operations* (action).
2. Konfigurējiet darbības informāciju.



Darbības vispārīgie atribūti:

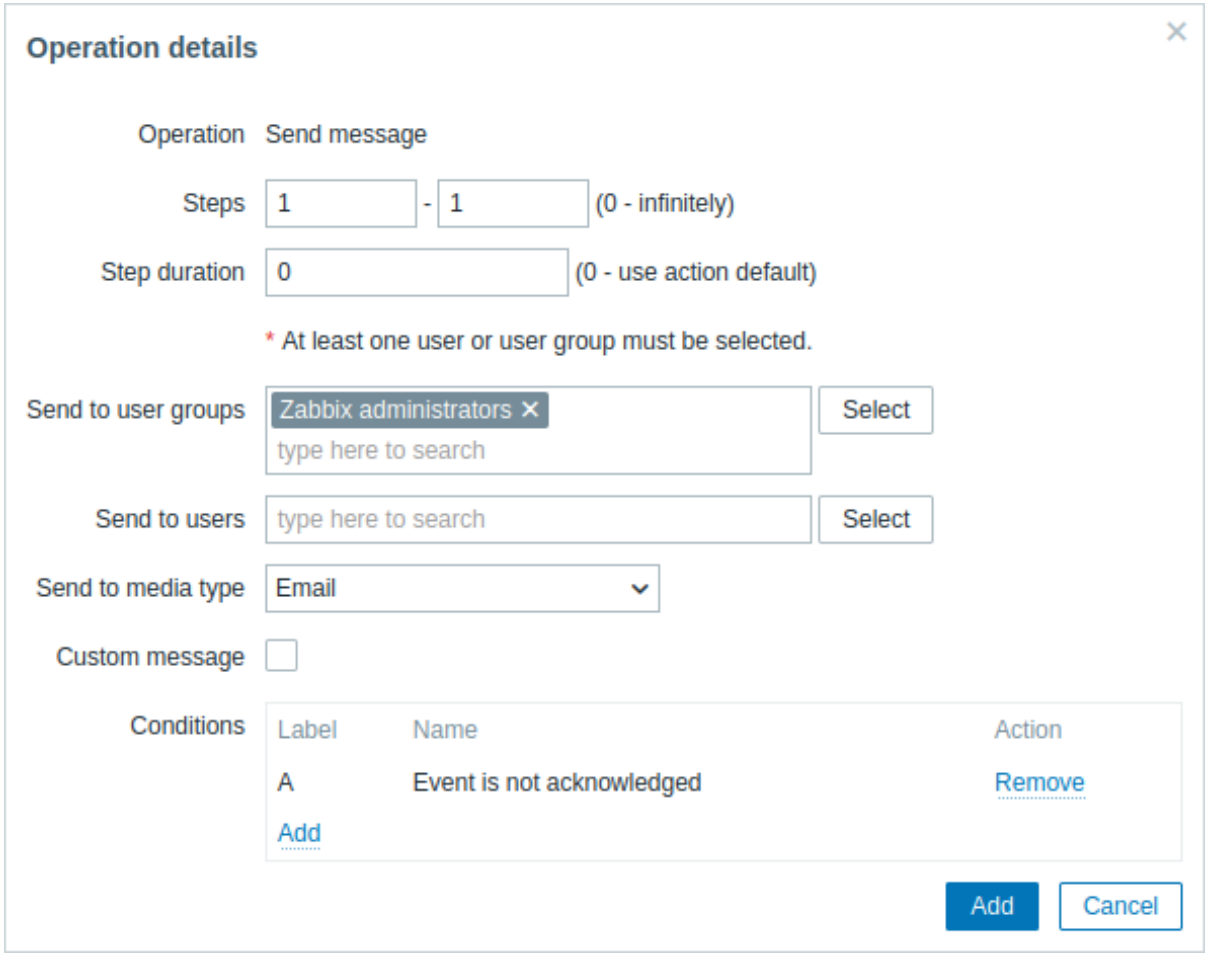
Parametrs	Apraksts
Default operation step duration	Vienas darbības darbības posma noklusējuma ilgums (no 60 sekundēm līdz 1 nedēļai). Piemēram, vienu stundu ilgs posms nozīmē, ka pēc darbības izpildes līdz nākamajam posmam paies viena stunda. Tiek atbalstīti laika sufiksi, piemēram, 60s, 1m, 2h, 1d. Tiek atbalstīti lietotāja makro.
Operations	Tiek parādītas darbības darbības (ja tādas ir) ar šādu informāciju: **Steps** - eskalācijas posms(-i), kuram(-iem) darbība ir piešķirta. **Details** - darbības veids un tās saņēmējs/mērķis. Darbību sarakstā tiek parādīts arī izmantotais multivides veids (e-pasts, SMS vai skripts), kā arī paziņojuma saņēmēja vārds un uzvārds (iekavās aiz lietotājvārda). **Start in** - cik ilgi pēc notikuma darbība tiek izpildīta. **Duration (sec)** - tiek parādīts posma ilgums. Ja posmam tiek izmantots noklusējuma ilgums, tiek parādīts *Default*, bet, ja tiek izmantots pielāgots ilgums, tiek parādīts laiks. **Actions** - tiek parādītas saites darbības rediģēšanai un noņemšanai.

Parametrs	Apraksts
Recovery operations	Tiek parādītas darbības darbības (ja tādas ir) ar šādu informāciju: **Details** - darbības veids un tās saņēmējs/mērķis. Darbību sarakstā tiek parādīts arī izmantotais multivides veids (e-pasts, SMS vai skripts), kā arī paziņojuma saņēmēja vārds un uzvārds (iekavās aiz lietotājvārda). **Actions** - tiek parādītas saites darbības rediģēšanai un noņemšanai.
Update operations	Tiek parādītas darbības darbības (ja tādas ir) ar šādu informāciju: **Details** - darbības veids un tās saņēmējs/mērķis. Darbību sarakstā tiek parādīts arī izmantotais multivides veids (e-pasts, SMS vai skripts), kā arī paziņojuma saņēmēja vārds un uzvārds (iekavās aiz lietotājvārda). **Actions** - tiek parādītas saites darbības rediģēšanai un noņemšanai.
Pause operations for symptom problems	Atzīmējiet šo izvēles rūtiņu, lai apturētu darbības (pēc pirmās darbības) simptomu problēmām. Ņemiet vērā, ka šis iestatījums ietekmē tikai problēmu eskalācijas; atkopšanas un atjaunināšanas darbības netiks ietekmētas. Šī opcija ir pieejama tikai *Trigger actions*.
Pause operations for suppressed problems	Atzīmējiet šo izvēles rūtiņu, lai atliktu darbību sākšanu uz uzturēšanas perioda laiku. Kad pēc uzturēšanas perioda darbības tiek sāktas, tiek izpildītas visas darbības, tostarp tās, kas attiecas uz uzturēšanas laikā notikušajiem notikumiem. Ņemiet vērā, ka šis iestatījums ietekmē tikai problēmu eskalācijas; atkopšanas un atjaunināšanas darbības netiks ietekmētas. Ja noņemsiet atzīmi no šīs izvēles rūtiņas, darbības tiks izpildītas bez aizkaves arī uzturēšanas perioda laikā. Šī opcija nav pieejama *Service actions*.
Notify about canceled escalations	Noņemiet atzīmi no šīs izvēles rūtiņas, lai atspējotu paziņojumus par atceltām eskalācijām (kad hosts, vienums, trigeris vai darbība ir atspējota).

Visi obligātie ievades lauki ir atzīmēti ar sarkanu zvaigznīti.

Lai konfigurētu jaunas darbības informāciju, blokā *Operations* noklikšķiniet uz Add. Lai rediģētu esošu darbību, noklikšķiniet uz Edit blakus darbībai. Tiks atvērts uznirstošais logs, kurā varēsiet rediģēt darbības posma informāciju.

Operācijas informācija



Parametrs	Apraksts
Operācija	Atlasiet operāciju: **Nosūtīt ziņojumu** - nosūtīt ziņojumu lietotājam. **<remote command name>** - izpildīt attālo komandu. Komandas ir pieejamas izpildei, ja tās iepriekš definētas sadaļā globālie skripti, kā tvērumu atlasot *Action operation*. Vairāk operāciju ir pieejamas atklāšanas un automātiskās reģistrācijas notikumiem (skatiet iepriekš).
Soļi	Atlasiet soli(-ļus), kuram(-iem) piešķirt operāciju eskalācijas grafikā: **No** - izpildīt, sākot ar šo soli. **Līdz** - izpildīt līdz šim solim (0 = bezgalība, izpilde netiks ierobežota).
Soļa ilgums	Pielāgots šo soļu ilgums (0 = izmantot noklusējuma soļa ilgumu). Tiek atbalstīti laika sufiksi, piemēram, 60s, 1m, 2h, 1d. Tiek atbalstīti lietotāja makro. Vienam un tam pašam solim var piešķirt vairākas operācijas. Ja šīm operācijām ir definēti atšķirīgi soļa ilgumi, tiek ņemts vērā īsākais ilgums un piemērots šim solim.
Operācijas tips: nosūtīt ziņojumu	
Nosūtīt lietotāju grupām	Atlasiet lietotāju grupas, kurām nosūtīt ziņojumu. Lai lietotāju grupa saņemtu paziņojumu, tai ir jābūt vismaz ”read” atļaujām attiecībā uz hostu.
Nosūtīt lietotājiem	Atlasiet lietotājus, kuriem nosūtīt ziņojumu. Lai lietotājs saņemtu paziņojumu, viņam ir jābūt vismaz ”read” atļaujām attiecībā uz hostu.

Parametrs	Apraksts
Nosūtīt multivides tipam	Nosūtīt ziņojumu visiem pieejamajiem (konfigurētajiem un iespējotajiem) multivides tipiem vai tikai konkrētam tipam.
Pielāgots ziņojums	Ja atlasīts, var konfigurēt pielāgotu ziņojumu. Paziņojumiem par iekšējiem notikumiem, kas tiek nosūtīti, izmantojot webhook, pielāgots ziņojums ir obligāts.
Temats	Pielāgotā ziņojuma temats. Tematā var būt makro. Tā garums ir ierobežots līdz 255 rakstzīmēm.
Ziņojums	Pielāgotais ziņojums. Ziņojumā var būt makro. Tā garums ir ierobežots līdz noteiktam rakstzīmju skaitam atkarībā no datubāzes tipa (plašāku informāciju skatiet sadaļā Ziņojuma nosūtīšana).
Operācijas tips: attālā komanda	
Mērķu saraksts	Atlasiet mērķus, kuros izpildīt komandu: **Pašreizējais hosts** - komanda tiek izpildīta tā hosta, kas izraisīja problēmas notikumu, trigerī. **Hosts** - atlasiet hostu(-s), kurā(-os) izpildīt komandu. **Hostu grupa** - atlasiet hostu grupu(-as), kurā(-ās) izpildīt komandu. Norādot vecāka hostu grupu, netieši tiek atlasītas visas ligzdotās hostu grupas. Tādējādi attālā komanda tiks izpildīta arī ligzdoto grupu hostos. Komanda hostā tiek izpildīta tikai vienu reizi, pat ja hosts atbilst vairāk nekā vienu reizi (piemēram, no vairākām hostu grupām; atsevišķi un no hostu grupas). Mērķu sarakstam nav nozīmes, ja pielāgots skripts tiek izpildīts Zabbix serverī. Šādā gadījumā, atlasot vairāk mērķu, skripts serverī tiks izpildīts vairāk reižu. Ņemiet vērā, ka globālajiem skriptiem mērķa atlase ir atkarīga arī no iestatījuma *Host group* globālā skripta konfigurācijā. Opcija *Mērķu saraksts* nav pieejama *Service actions*, jo šādā gadījumā attālās komandas vienmēr tiek izpildītas Zabbix serverī.
Nosacījumi	Nosacījums operācijas veikšanai: **Notikums nav apstiprināts** - tikai tad, ja notikums nav apstiprināts. **Notikums ir apstiprināts** - tikai tad, ja notikums ir apstiprināts. Opcija *Nosacījumi* ir pieejama tikai *Trigger actions*.

Kad esat pabeidzis, noklikšķiniet uz *Add*, lai pievienotu operāciju sarakstam *Operations*.

Ziņojuma nosūtīšana

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 nosūtīt un saņemt paziņojumus no Zabbix, jums ir:

- Jādefinē multivide, uz kuru nosūtīt ziņojumu.

Ja darbība notiek ārpus atlasītajai multividei lietotāja konfigurācijā definētā perioda Kad aktīvs, ziņojums netiks nosūtīts.

Noklusējuma trigeru smaguma pakāpe (“Nav klasificēts”) **ir jāatzīmē** lietotāja multivides konfigurācijā, ja vēlaties saņemt paziņojumus par notikumiem, kas nav saistīti ar trigeriem, piemēram, atklāšanu, aktīva aģenta automātisku reģistrāciju vai iekšējiem notikumiem.

- Jākonfigurē darbības operācija, kas nosūta ziņojumu uz kādu no definētajām multividēm.

Attention:
Zabbix nosūta paziņojumus tikai tiem lietotājiem, kuriem ir vismaz "lasīšanas" atļaujas uz hostu, kas ģenerēja notikumu. Vismaz vienam hostam trigeru izteiksmē ir jābūt pieejamam.

Varat konfigurēt pielāgotus ziņojumu nosūtīšanas scenārijus, izmantojot eskalācijas.

Lai veiksmīgi saņemtu un lasītu e-pasta ziņojumus no Zabbix, e-pasta serveriem/klientiem ir jāatbalsta standarta "SMTP/MIME e-pasta" formāts, jo Zabbix nosūta UTF-8 datus (ja tēma satur tikai ASCII rakstzīmes, tā netiek kodēta UTF-8 formātā). Ziņojuma tēma un pamatteksts tiek kodēti base64 formātā, lai atbilstu "SMTP/MIME e-pasta" formāta standartam.

Ziņojuma ierobežojums pēc visu makrosu izvēršanas ir tāds pats kā ziņojuma ierobežojums attālajām komandām.

Ziņojumu izsekošana

Jūs varat skatīt nosūtīto ziņojumu statusu sadaļā *Monitoring* > *Problems*.

Kolonnā *Actions* varat redzēt apkopotu informāciju par veiktajām darbībām. Tajā zaļie skaitļi norāda nosūtītos ziņojumus, sarkanie - neveiksmīgi nosūtītos ziņojumus. *In progress* norāda, ka darbība ir uzsākta. *Failed* informē, ka neviena darbība nav veiksmīgi izpildīta.

Ja noklikšķināsiet uz notikuma laika, lai skatītu notikuma informāciju, sadaļā *Actions* varēsiet redzēt informāciju par ziņojumiem, kas tika nosūtīti (vai netika nosūtīti) šī notikuma dēļ.

Sadaļā *Reports* > *Action log* redzēsiet informāciju par visām veiktajām darbībām tiem notikumiem, kuriem ir konfigurēta darbība.

Attālinātās komandas

Pārskats

Izmantojot attālās komandas, varat definēt, ka noteikts iepriekš definēts komandas tiek automātiski izpildīta uz uzraudzītā hosta, iestājoties noteiktam nosacījumam.

Tādējādi attālās komandas ir jaudīgs mehānisms viedai proaktīvai uzraudzībai.

Acīmredzamākajos šīs funkcionalitātes izmantošanas gadījumos varat:

- Automātiski restartēt kādu lietotni (tīmekļa serveri, starpprogrammatūru, CRM), ja tā nereaģē
- Izmantot IPMI komandu `reboot`, lai restartētu attālo serveri, ja tas neatbild uz pieprasījumiem
- Automātiski atbrīvot vietu diskā (dzēšot vecākus failus, iztīrot /tmp), ja diskā sāk trūkt vietas
- Migrēt virtuālo mašīnu no viena fiziskā datora uz citu atkarībā no CPU noslodzes
- Pievienot jaunus mezglus mākoņa vidē, ja nepietiek CPU (diska, atmiņas vai citu) resursu

Darbības konfigurēšana attālajām komandām ir līdzīga darbības konfigurēšanai ziņojuma nosūtīšanai; vienīgā atšķirība ir tā, ka Zabbix ziņojuma nosūtīšanas vietā izpilda komandu.

Attālās komandas var izpildīt Zabbix serveris, Zabbix starpniekserveris vai Zabbix aģents. Ņemiet vērā:

- Attālās komandas Zabbix aģentā var izpildīt tieši Zabbix serveris vai ar Zabbix starpniekservera starpniecību.
- Zabbix servera izpildītās attālās komandas tiek palaistas, kā aprakstīts sadaļā Komandu izpilde, tostarp pārbaudot izejas kodu.
- Attālās komandas tiek izpildītas arī tad, ja mērķa hosts atrodas uzturēšanas režīmā.

Pēc noklusējuma attālās komandas ir atspējotas gan Zabbix aģentā, gan Zabbix starpniekserverī. Tās var iespējot šādi:

- Attālajām komandām, ko izpilda Zabbix aģents, aģenta konfigurācijā katrai atļautajai komandai pievienojiet parametru `AllowKey=system.run[<command>,*]` vai `AllowKeyRegexp=^system\.run\[<command>,.*\]$`, kur `*` atbilst jebkuram izpildes režīmam (`wait` vai `nowait`).
- Attālajām komandām, ko izpilda Zabbix starpniekserveris, starpniekservera konfigurācijā iestatiet parametru `EnableRemoteCommands` uz 1. Ņemiet vērā, ka attālās komandas nav jāiespējo Zabbix starpniekserverī, ja tās Zabbix aģents izpilda uz hosta, kuru uzrauga starpniekserveris.

Attention:
Neizmantojiet `AllowKey=system.run[*]` vai `AllowKeyRegexp=^system\.run\[.*\]$`, ja vien tas nav nepieciešams, jo tādējādi tiek atļauts izpildīt jebkuru komandu uz hosta. Drošības apsvērumu dēļ ieteicams atļaut tikai precīzi norādītas komandas, savukārt `system.run[*]` pēc noklusējuma ir aizliegts (`DenyKey=system.run[*]`).

Skatiet arī sadaļu Aģenta pārbaužu ierobežošana.

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.

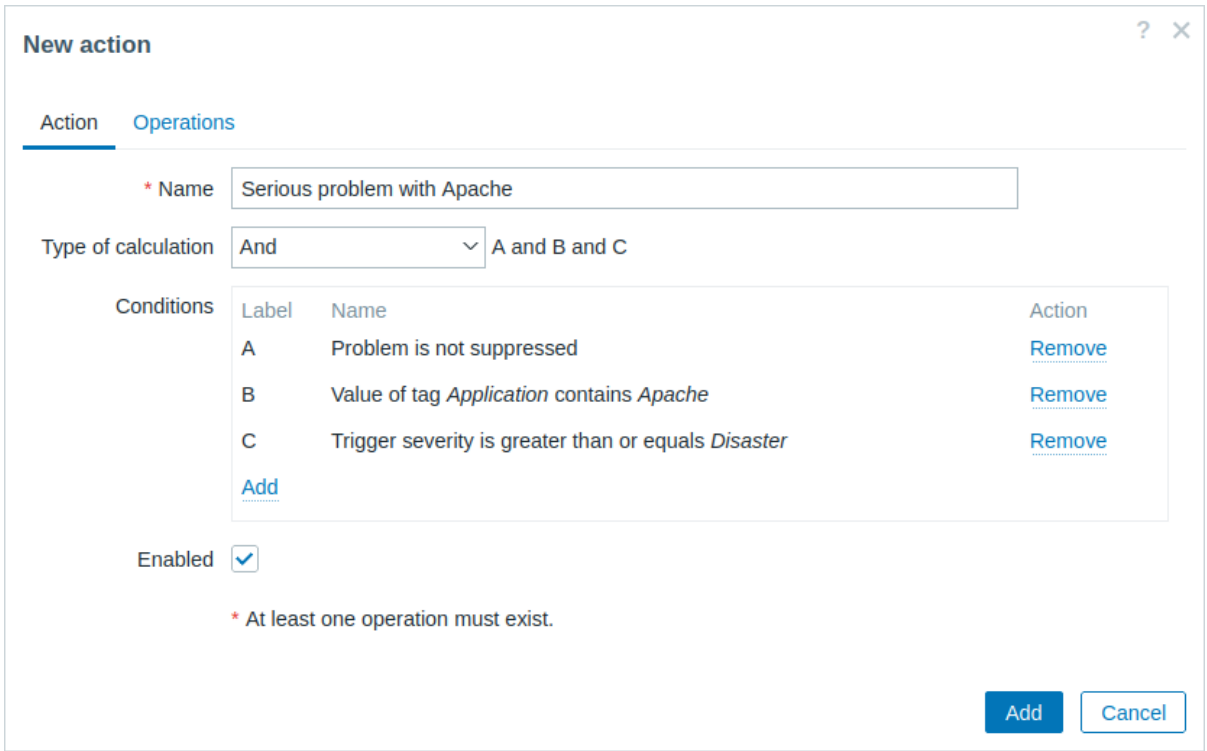
Konfigurēšana

Attālās komandas, ko izpilda Zabbix aģents, vispirms ir jāatļauj aģenta konfigurācijā, izmantojot parametru `AllowKey` vai `AllowKeyRegexp`.

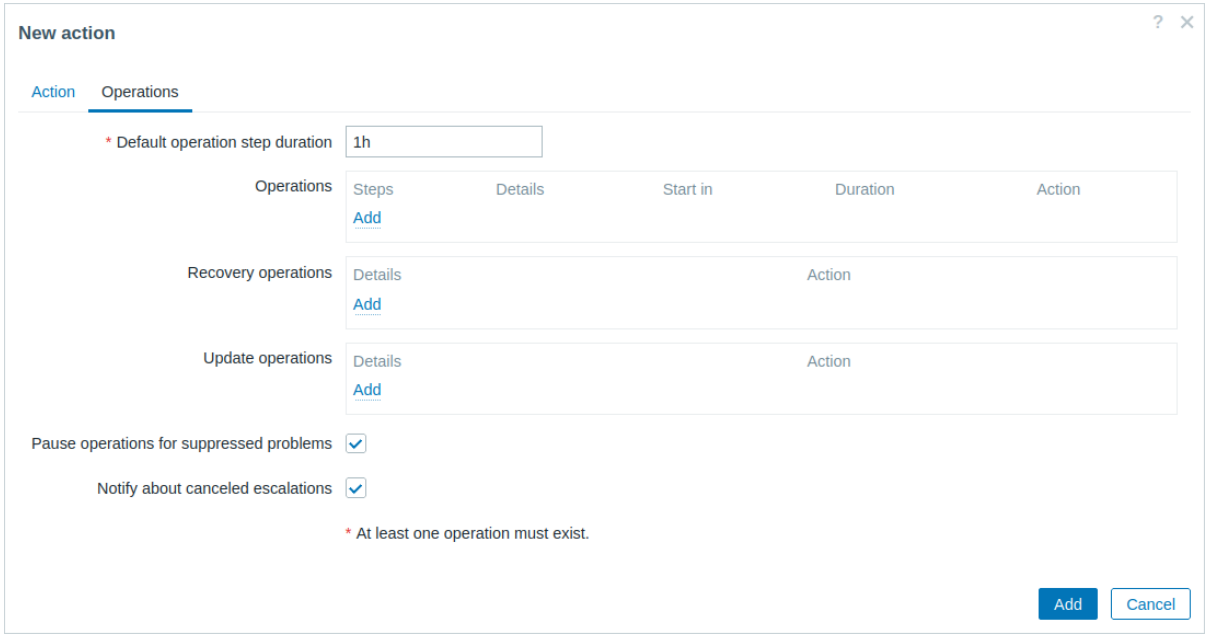
Pārliecinieties, ka aģenta konfigurācijā katrai atļautajai komandai ir pievienots parametrs `AllowKey=system.run[<command>,*]` vai `AllowKeyRegexp=^system\.run\[<command>,.*\]$`, lai atļautu tās izpildi `nowait` režīmā. Pēc parametra maiņas restartējiet aģenta dēmonu.

Pēc tam, konfigurējot jaunu darbību, dodieties uz *Alerts* > *Actions* > *Trigger actions*:

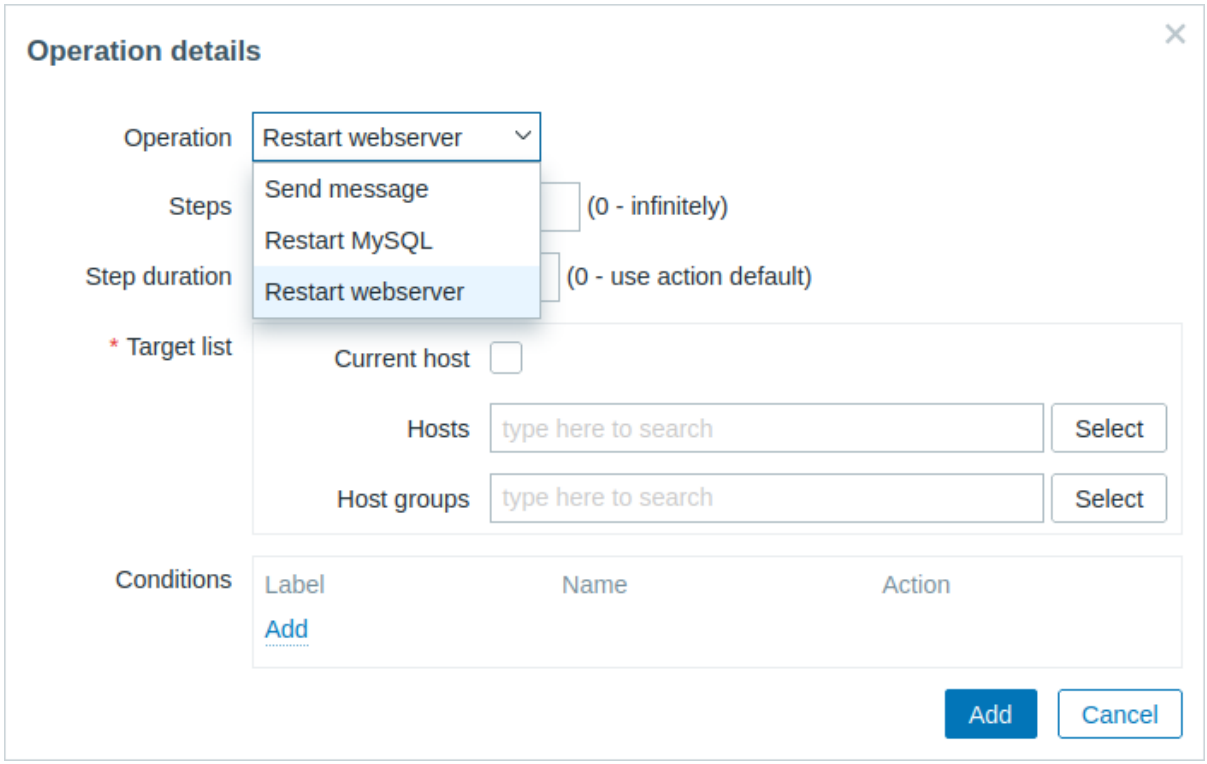
1. Definējiet atbilstošos nosacījumus, piemēram, iestatiet, lai darbība tiktu aktivizēta, ja kādai no Apache lietotnēm rodas jebkura katastrofas līmeņa problēma.



2. Cilnē *Operations* sadaļā *Operations*, *Recovery operations* vai *Update operations* noklikšķiniet uz *Add*.



3. Nolaižamajā sarakstā *Operation* atlasiet kādu no iepriekš definētajiem skriptiem un iestatiet skripta *Target list*.



Iepriekš definēti skripti

Skripti, kas ir pieejami darbību operācijām (webhook, skripts, SSH, Telnet, IPMI), ir definēti globālajos skriptos, kā darbības jomu izvēloties *Action operation*.

Piemēram:

```
sudo /etc/init.d/apache restart
```

Šajā gadījumā Zabbix mēģinās restartēt Apache procesu. Ar šo komandu pārliecinieties, ka komanda tiek izpildīta uz Zabbix aģents (noklikšķiniet uz pogas *Zabbix agent* pie *Execute on*).

Attention:
Ņemiet vērā sudo izmantošanu - Zabbix lietotājam pēc noklusējuma nav tiesību restartēt sistēmas pakalpojumus. Skatiet tālāk ieteikumus, kā konfigurēt sudo.

Note:
Sākot ar Zabbix aģents 7.0, attālinātās komandas var izpildīt arī uz aģents, kas darbojas aktīvajā režīmā. Zabbix aģents - neatkarīgi no tā, vai tas ir aktīvs vai pasīvs - jādarbojas attālajā hostā, un tas izpilda komandas fonā.

Attālinātās komandas uz Zabbix aģents tiek izpildītas bez noildzes, izmantojot `system.run[,nowait]` atslēgu, un to izpildes rezultāti netiek pārbaudīti. Uz Zabbix serveris un Zabbix starpniekserveris attālinātās komandas tiek izpildītas ar noildzi, kā noteikts `TrapperTimeout` parametrā failā `zabbix_server.conf` vai `zabbix_proxy.conf`, un to izpildes rezultāti tiek pārbaudīti. Papildu informāciju skatiet *Script timeout*.

Piekļuves atļaujas

Pārliecinieties, ka lietotājam `zabbix` ir izpildes atļaujas konfigurētajām komandām. Var būt noderīgi izmantot `sudo`, lai piešķirtu piekļuvi priviliģētām komandām. Lai konfigurētu piekļuvi, izpildiet kā root:

```
visudo
```

Piemēra rindas, ko var izmantot `sudoers` failā:

```
# ļauj lietotājam 'zabbix' palaist visas komandas bez paroles.
zabbix ALL=NOPASSWD: ALL

# ļauj lietotājam 'zabbix' restartēt apache bez paroles.
zabbix ALL=NOPASSWD: /etc/init.d/apache restart
```

Note:
Dažās sistēmās `sudoers` fails neļaus ne-lokāliem lietotājiem izpildīt komandas. Lai to mainītu, komentējiet `requiretty` opciju 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 noteiktos apstākļos.

Lai automātiski restartētu Windows, kad Zabbix konstatē problēmu, definējiet šādu skriptu:

Script parameter	Value
Scope	'Action operation'
Type	'Script'
Command	`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

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.

Makro 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
```

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

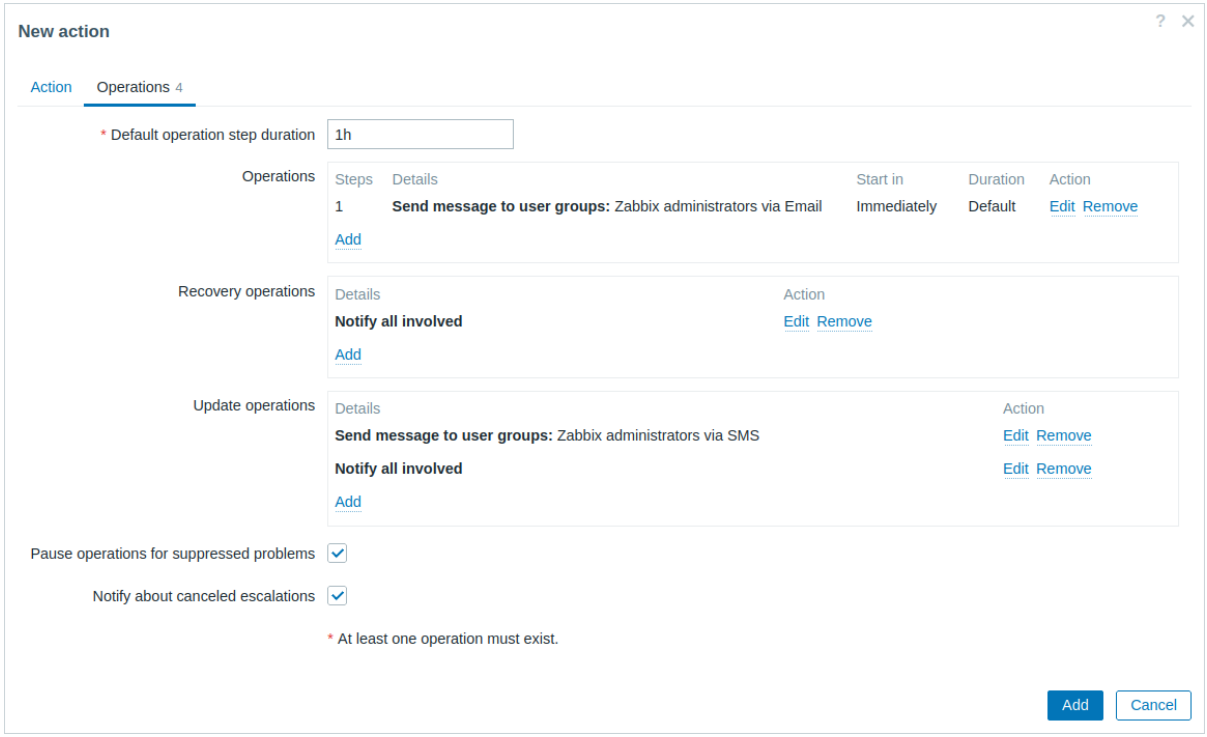
Daži atkopšanas operāciju lietošanas gadījumi ir šādi:

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.

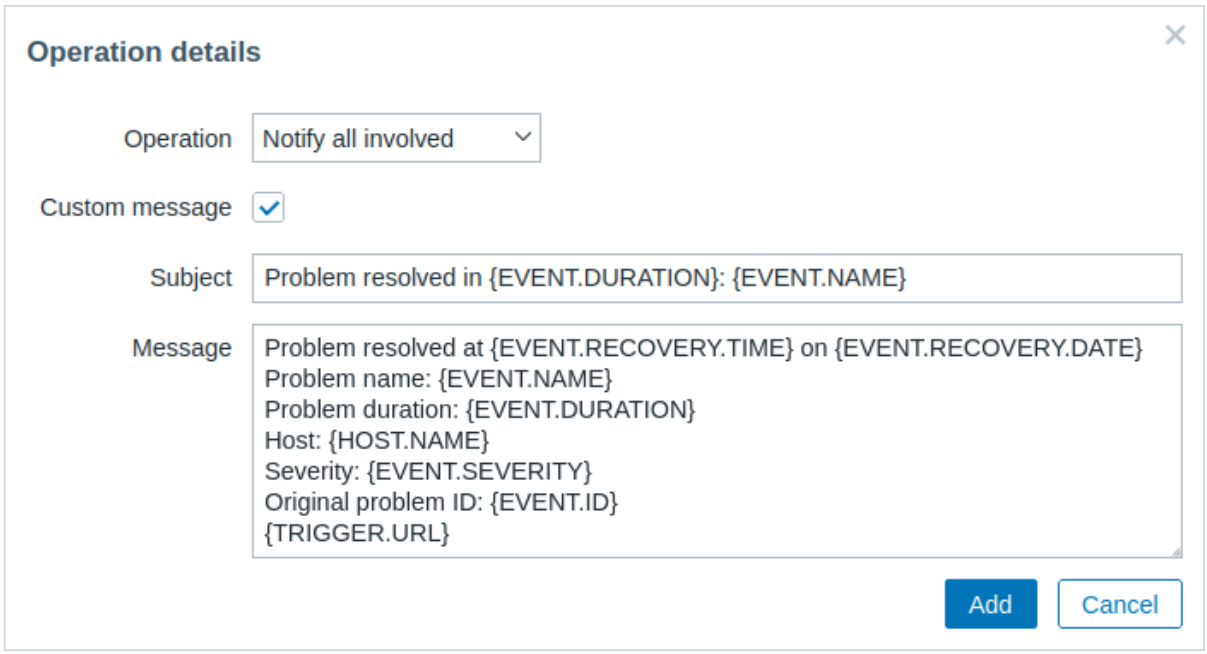
Atkopšanas operācijas konfigurēšana

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 darbības informācija



Atkopšanas notikumiem ir pieejami trīs darbību 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.

Katra darbību tipa parametri ir aprakstīti tālāk. Visi obligātie ievades lauki ir atzīmēti ar sarkanu zvaigznīti. Kad pabeigts, noklikšķiniet uz *Pievienot*, lai pievienotu darbību sarakstam *Atkopšanas darbības*.

Note:
Ņemiet vērā, ka, ja viens un tas pats saņēmējs ir definēts vairākos darbību 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.

Darbības veids: attālināta komanda

Parametrs	Apraksts
Mērķu saraksts	Atlasiet mērķus, kuros izpildīt komandu: **Pašreizējais hosts** - komanda tiek izpildīta tā trigeri izraisījušā hosta, kurā radies problēmas notikums, kontekstā. Šī opcija nedarbosies, ja trigerī ir vairāki hosti. **Hosts** - atlasiet hostu(s), kuros izpildīt komandu. **Hostu grupa** - atlasiet hostu grupu(as), kurās izpildīt komandu. Norādot vecāku hostu grupu, netieši tiek atlasītas visas ligzdotās hostu grupas. Tādējādi attālinātā komanda tiks izpildīta arī ligzdotajās grupās esošajos hostos. Komanda hostā tiek izpildīta tikai vienu reizi, pat ja hosts atbilst vairāk nekā vienu reizi (piemēram, no vairākām hostu grupām; atsevišķi un no hostu grupas). Mērķu sarakstam nav nozīmes, ja komanda tiek izpildīta Zabbix serverī. Šādā gadījumā, atlasot vairāk mērķu, komanda serverī tiks izpildīta tikai vairāk reižu. 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.

Atjaunināšanas darbības

Pārskats

Atjaunināšanas darbības ir pieejamas darbībās ar šādiem notikumu avotiem:

- *Trigeri* - kad problēmas ir atjaunināts citu lietotāju veiktu darbību rezultātā, t. i., tām ir pievienots komentārs, tās ir apstiprinātas, mainīts to smagums, aizvērtas (manuāli);
- *Pakalpojumi* - kad pakalpojuma smagums ir mainījies, bet pakalpojums joprojām nav atjaunots.

Lūdzu, ņemiet vērā, ka lietotāji nesaņem paziņojumus par saviem veiktajiem atjaunājumiem.

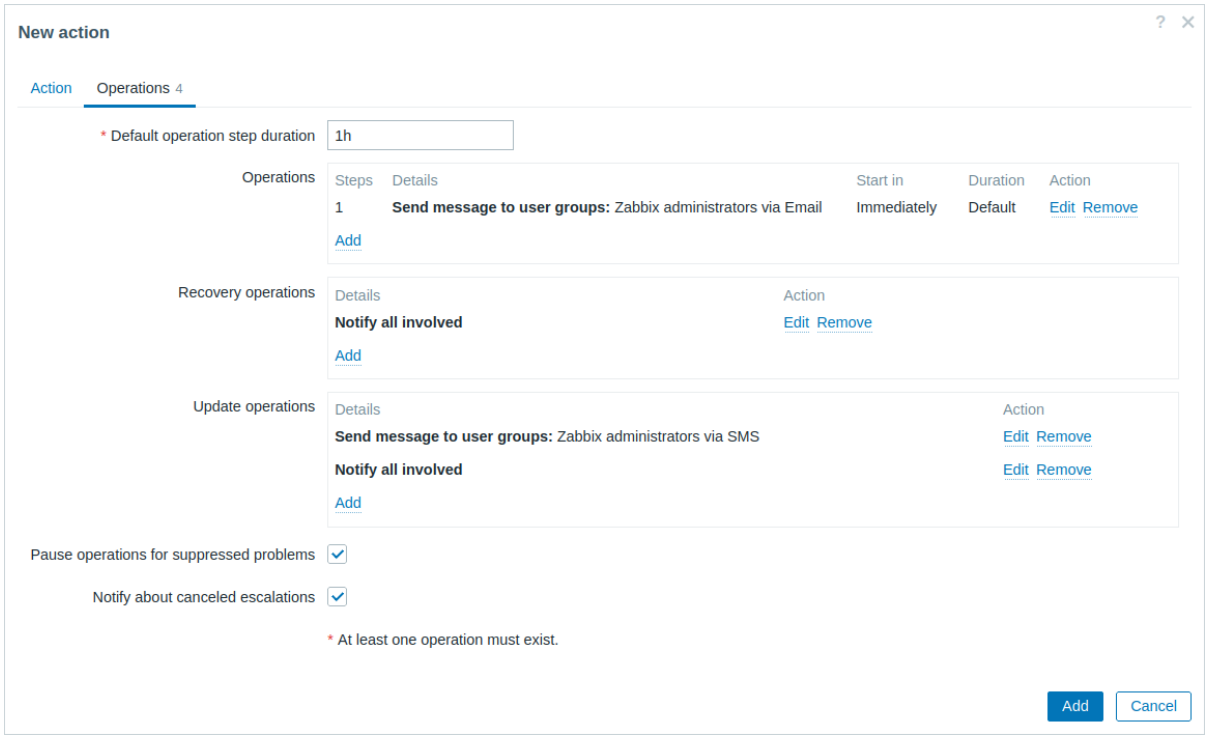
Atjaunināšanas darbībās tiek atbalstīti gan ziņojumi, gan attālinātās komandas. Lai gan var pievienot vairākas darbības, eskalācija netiek atbalstīta - visas darbības tiek piešķirtas vienam solim un tādēļ tiks izpildītas vienlaikus.

Atjaunināšanas operācijas konfigurēšana

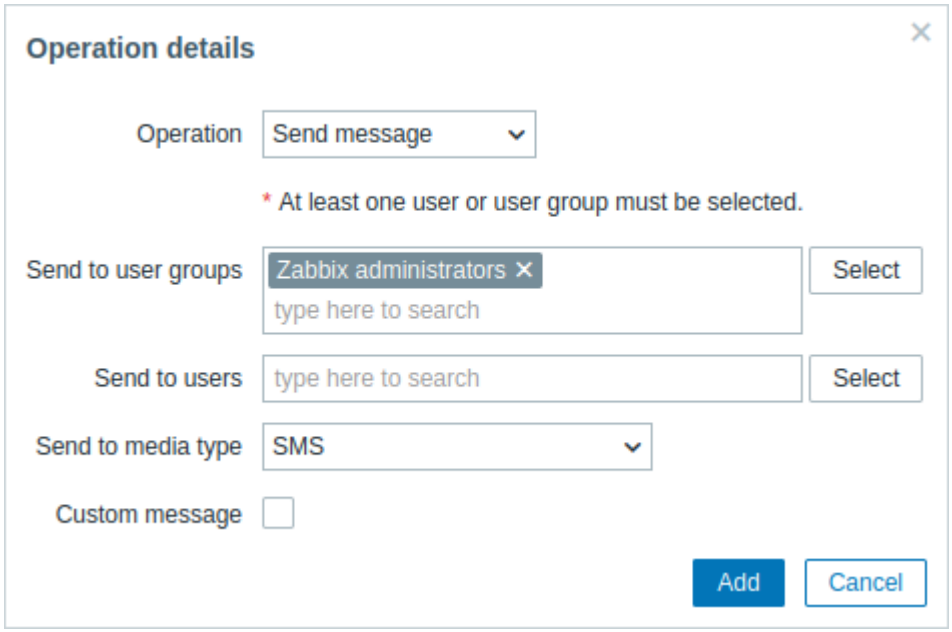
Lai konfigurētu atjaunināšanas operāciju, veiciet šādas darbības:

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.

Eskalācijas

Pārskats

Ar eskalācijām varat izveidot pielāgotus scenārijus paziņojumu sūtīšanai vai attālinātu komandu izpildei.

Praksē tas nozīmē, ka:

- Lietotājus var nekavējoties informēt par jaunām problēmām.
- Paziņojumus var atkārtot, līdz problēma ir novērsta.
- Paziņojuma nosūtīšanu var aizkavēt.
- Paziņojumus var eskalēt uz citu, ”augstāku” lietotāju grupu.

- Attālinātās komandas var izpildīt nekavējoties vai tad, ja problēma ilgstoši netiek novērsta.

Darbības tiek eskalētas, pamatojoties uz **eskalācijas soli**. Katram solim ir noteikts ilgums.

Varat definēt gan noklusējuma ilgumu, gan arī atsevišķa soļa pielāgotu ilgumu. Viena eskalācijas soļa minimālais ilgums ir 60 sekundes.

Varat sākt darbības, piemēram, paziņojumu sūtīšanu vai komandu izpildi, no jebkura soļa. Pirmais solis ir paredzēts tūlītējām darbībām. Ja vēlaties aizkavēt darbību, varat to piešķirt vēlākam solim. Katram solim var definēt vairākas darbības.

Eskalācijas soļu skaits nav ierobežots.

Eskalācijas tiek definētas, konfigurējot operāciju. Eskalācijas tiek atbalstītas tikai problēmu operācijām, nevis atkopšanai.

Dažādi eskalācijas darbības aspekti

Apskatīsim, kas notiek dažādos apstākļos, ja darbība satur vairākus eskalācijas soļus.

Situācija	Darbība
Attiecīgais hosts pāriet uzturēšanas režīmā 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.
*Laika periods, kas definēts darbības nosacījumā **Time period**, beidzas pēc sākotnējā paziņojuma nosūtīšanas*	Visi atlikušie eskalācijas soļi tiek izpildīti. Nosacījums *Time period* nevar apturēt operācijas; tas ietekmē to, kad darbības tiek sāktas vai netiek sāktas, nevis operācijas.
Problēma sākas uzturēšanas laikā un turpinās (netiek novērsta) pēc uzturēšanas beigām	Atkarībā no iestatījuma *Pause operations for suppressed problems* darbības konfigurācijā visi eskalācijas soļi tiek izpildīti vai nu no uzturēšanas beigšanās brīža, vai nekavējoties.
Problēma sākas uzturēšanas laikā bez datiem un turpinās (netiek novērsta) pēc uzturēšanas beigām *Dažādas eskalācijas seko cita citai ar nelielu intervālu un pārklājas*	Tai jāgaida, līdz trigeris tiek aktivizēts, un tikai pēc tam tiek izpildīti visi eskalācijas soļi. Katras jaunas eskalācijas izpilde aizstāj iepriekšējo eskalāciju, taču iepriekšējā eskalācijā vienmēr tiek izpildīts vismaz viens eskalācijas solis. Šāda darbība ir būtiska darbībām saistībā ar notikumiem, kas tiek izveidoti katrā trigera problēmas novērtēšanas reizē.
Notiekošas eskalācijas laikā (piemēram, kamēr tiek nosūtīts ziņojums), pamatojoties uz jebkāda veida notikumu: - darbība tiek atspējota Pamatojoties uz trigera notikumu: - trigeris tiek atspējots - hosts vai vienums tiek atspējots Pamatojoties uz iekšēju notikumu par trigeriem: - trigeris tiek atspējots Pamatojoties uz iekšēju notikumu par vienumiem/zemā līmeņa atklāšanas noteikumiem: - vienums tiek atspējots - hosts tiek atspējots	Notiekošais ziņojums tiek nosūtīts, un pēc tam eskalācijas ietvaros tiek nosūtīts vēl viens ziņojums. Turpmākā ziņojuma pamattekstā sākumā būs atcelšanas teksts (*NOTE: Escalation canceled*), kurā norādīts iemesls (piemēram, *NOTE: Escalation canceled: action '<Action name>' disabled*). Tādējādi adresāts tiek informēts, ka eskalācija ir atcelta un turpmākie soļi netiks izpildīti. Šis ziņojums tiek nosūtīts visiem, kuri iepriekš saņēma paziņojumus. Atcelšanas iemesls tiek ierakstīts arī servera žurnālfailā (sākot ar Debug Level 3=Warning). Ņemiet vērā, ka ziņojums *Escalation canceled* tiek nosūtīts arī tad, ja operācijas ir pabeigtas, bet ir konfigurētas atkopšanas operācijas un tās vēl nav izpildītas.
Notiekošas eskalācijas laikā (piemēram, kamēr tiek nosūtīts ziņojums) darbība tiek dzēsta	Vairāk ziņojumu netiek nosūtīts. Informācija tiek ierakstīta servera žurnālfailā (sākot ar Debug Level 3=Warning), piemēram: `escalation canceled: action id:334 deleted`

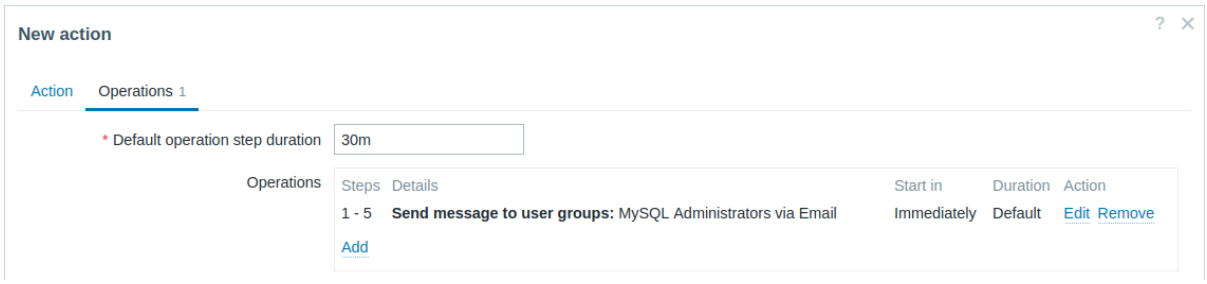
Eskalācijas piemēri

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).

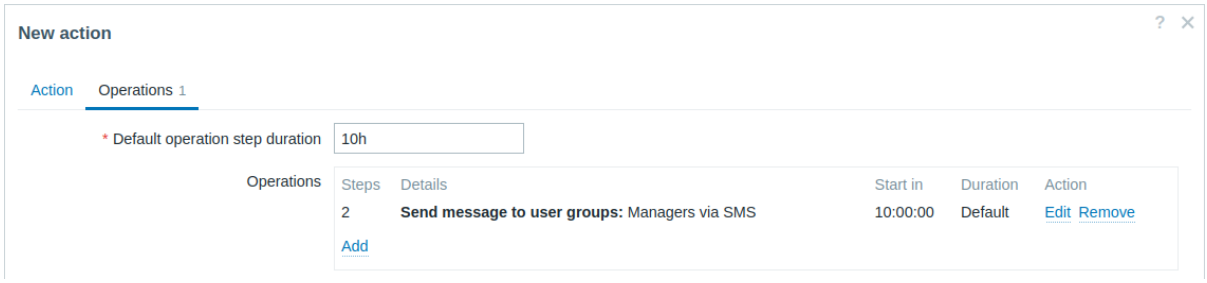
Ja problēma tiek atrisināta un ir konfigurēts atkopšanas ziņojums, tas tiks nosūtīts tiem, kuri šajā eskalācijas scenārijā saņēma vismaz vienu problēmas ziņojumu.

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”.



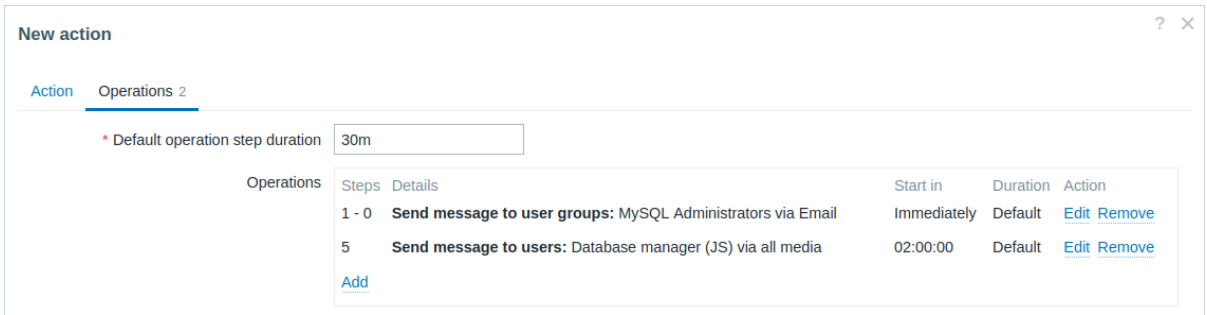
Paziņojums tiks nosūtīts tikai eskalācijas scenārija 2. solī, jeb 10 stundas pēc problēmas sākuma.

Varat pielāgot ziņojuma tekstu, piemēram: ”Problēma ilgst jau vairāk nekā 10 stundas”.

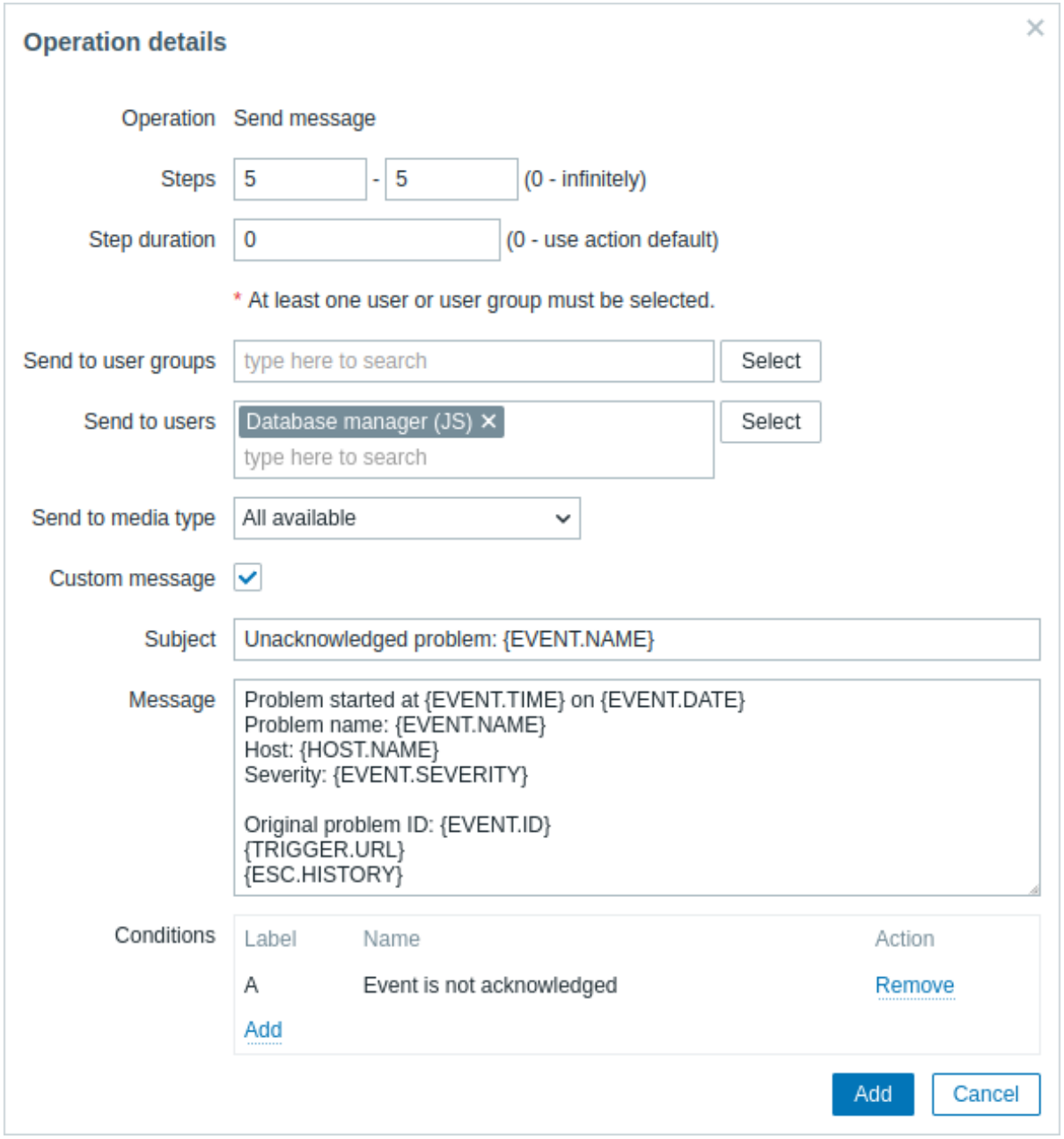
Piemērs 3

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:



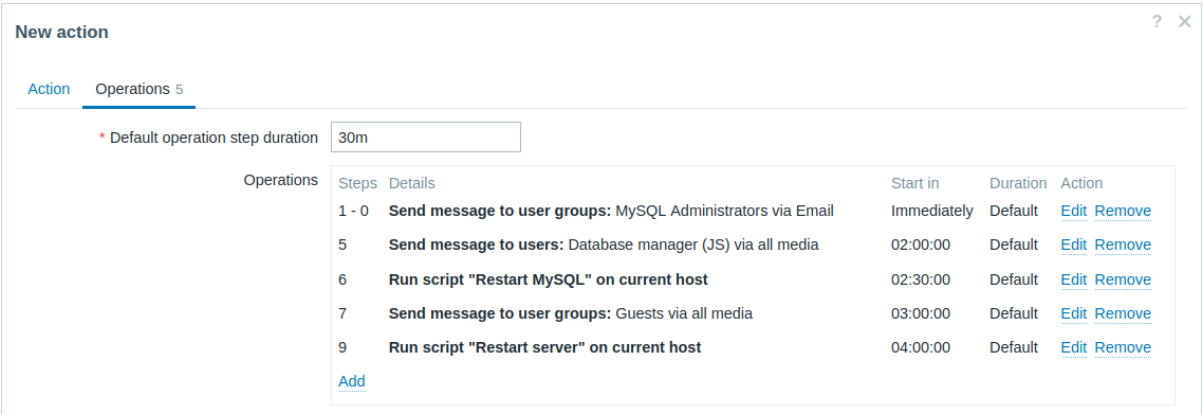
Ņemiet vērā pielāgotajā ziņojumā izmantoto makrosu {ESC.HISTORY}. Šis makross saturēs informāciju par visām iepriekš izpildītajām šīs eskalācijas darbībām, piemēram, nosūtītajiem paziņojumiem un izpildītajām komandām.

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.

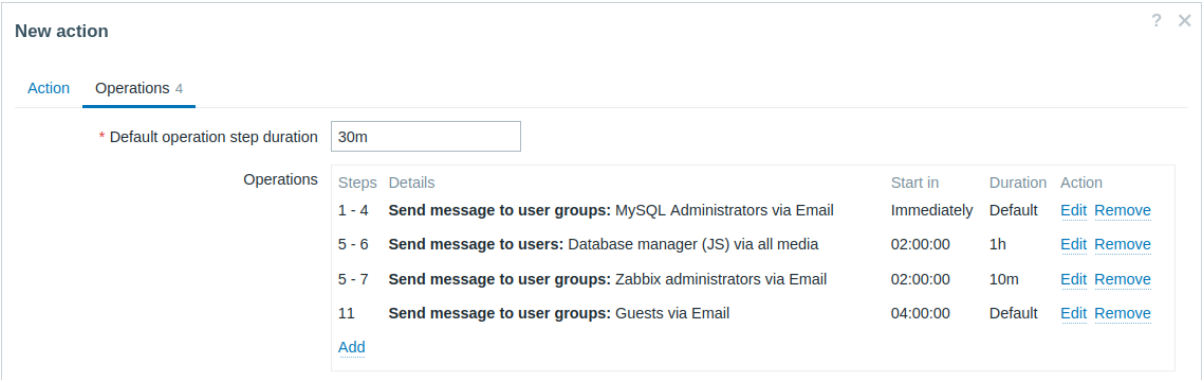
Ja problēma joprojām pastāvēs, pēc vēl 30 minūtēm Zabbix nosūtīs ziņojumu visiem vieslietotājiem.

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).

Paziņojuma saņemšana par neatbalstītiem vienumiem

Pārskats

Zabbix ir iespējams saņemt paziņojumus par neatbalstītiem vienumiem.

Tas ir daļa no Zabbix iekšējo notikumu koncepcijas, kas ļauj lietotājiem saņemt paziņojumus šādos gadījumos. Iekšējie notikumi atspoguļo stāvokļa maiņu:

- kad vienumi pāriet no ”normāls” uz ”neatbalstīts” (un atpakaļ);
- kad trigeri pāriet no ”normāls” uz ”nezināms” (un atpakaļ);
- kad zemā līmeņa atklāšanas noteikumi 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 mediju, piemēram, e-pastu, SMS vai skriptu, ko izmantot paziņojumiem. Lai veiktu šo uzdevumu, skatiet attiecīgās rokasgrāmatas sadaļas.

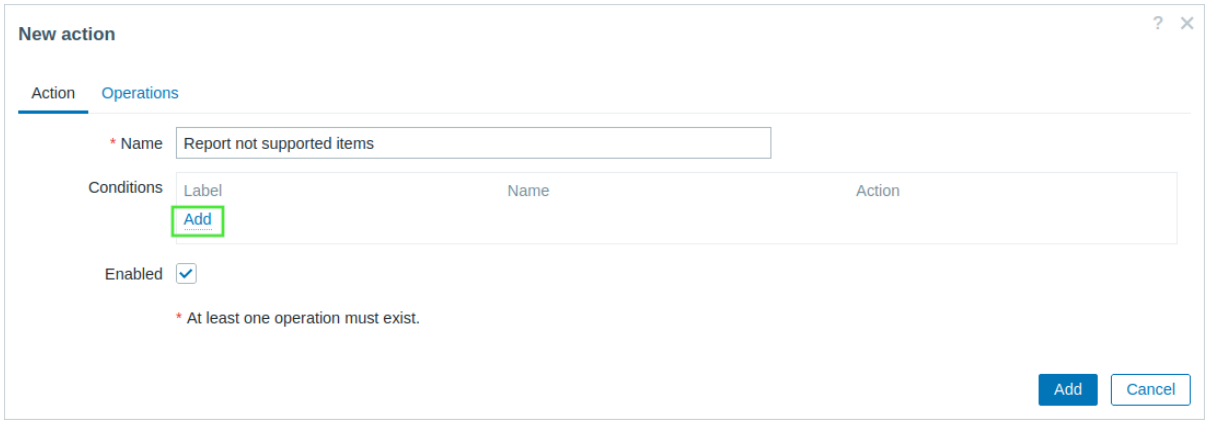
Attention:
Lai paziņotu par iekšējiem notikumiem, pēc noklusējuma tiek izmantota nopietnības pakāpe (“Neklasificēts”), tāpēc atstājiet to atzīmētu, konfigurējot lietotāja medijus, ja vēlaties saņemt paziņojumus par iekšējiem notikumiem.

2. solis

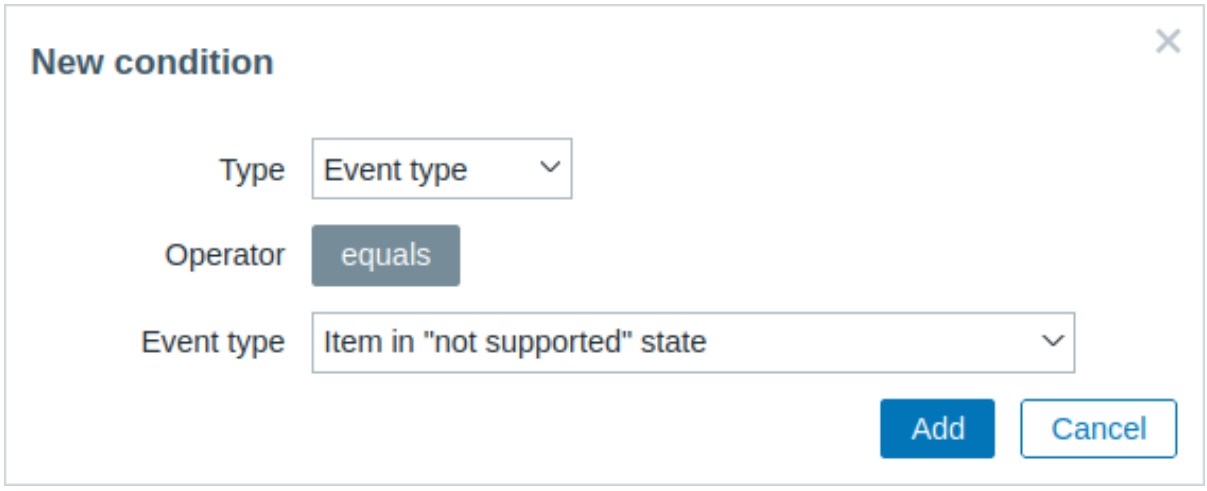
1. Dodieties uz *Alerts* > *Actions* > *Internal actions*.
2. Noklikšķiniet uz *Create action* lapas augšējā labajā stūrī, lai atvērtu darbības konfigurācijas formu.

3. darbība

1. Cilnē *Action* ievadiet darbības nosaukumu.
2. Blokā *Conditions* noklikšķiniet uz *Add*, lai pievienotu jaunu nosacījumu.



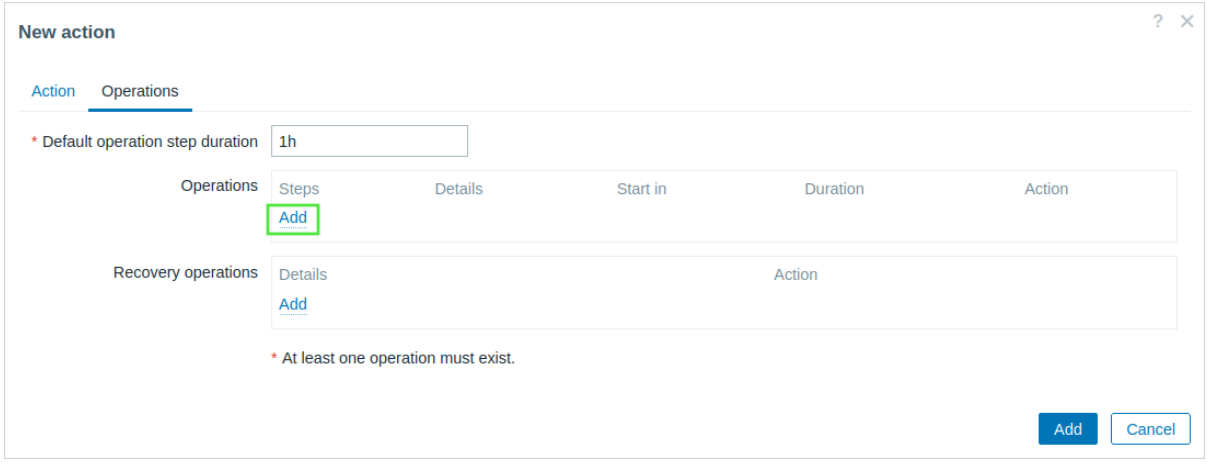
3. Uznirstošajā logā *New condition* kā nosacījuma tipu atlasiet “Event type” un pēc tam kā notikuma tipu atlasiet “Item in ’not supported’ state”.



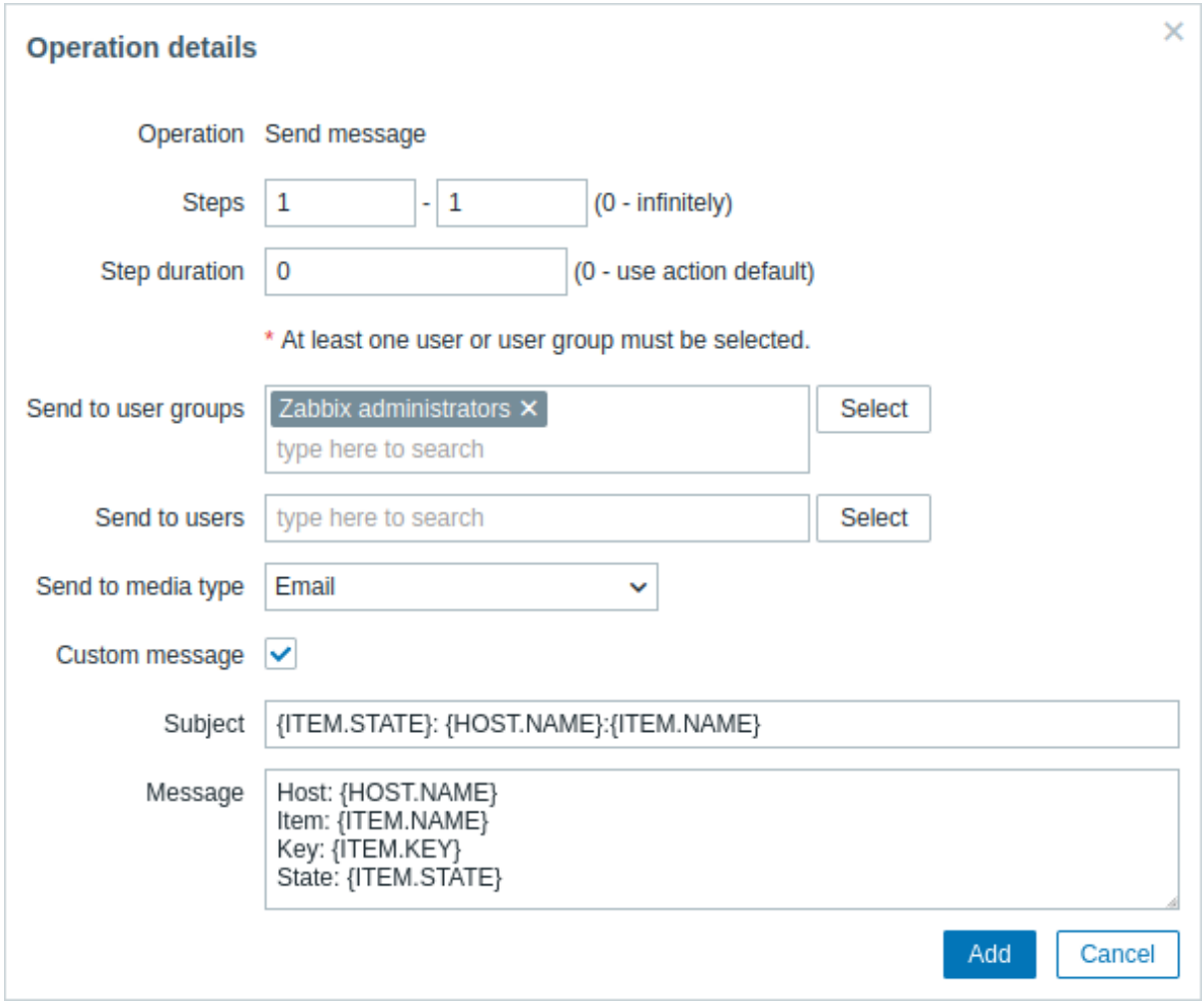
4. Noklikšķiniet uz *Add*, lai faktiski pievienotu nosacījumu blokam *Conditions*.

4. darbība

1. Cilnē *Operations* blokā *Operations* noklikšķiniet uz *Add*, lai pievienotu jaunu darbību.



2. Atlasiet ziņojuma adresātus (lietotāju grupas/lietotājus) un piegādei izmantojamo multivides tipu (vai *All*).
3. Atzīmējiet izvēles rūtiņu *Custom message*, ja vēlaties ievadīt problēmas ziņojuma pielāgoto tēmu/saturu.



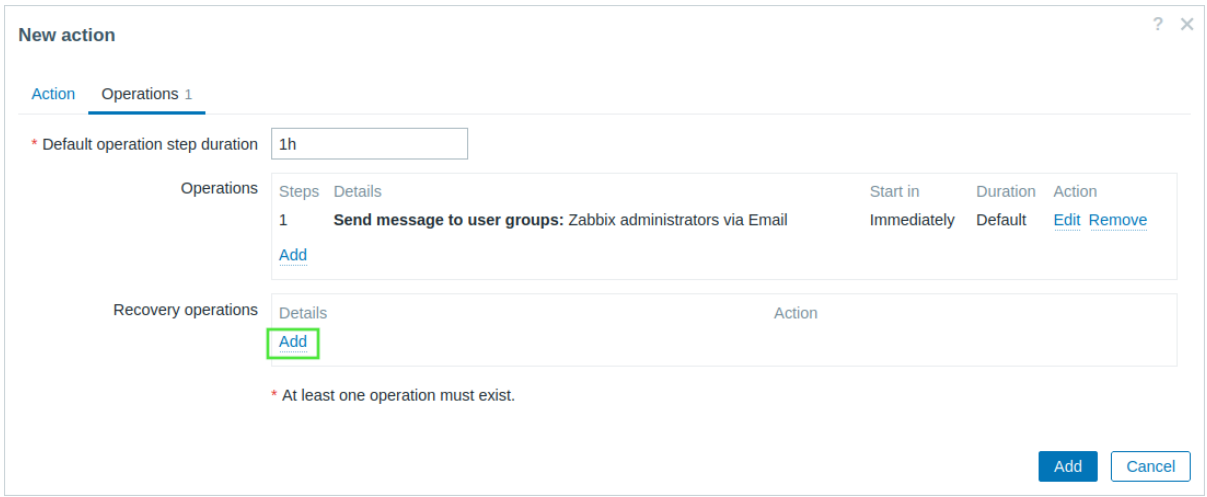
4. Noklikšķiniet uz *Add*, lai faktiski pievienotu darbību blokam *Operations*.

Ja vēlaties saņemt vairāk nekā vienu paziņojumu, iestatiet darbības soļa ilgumu (intervālu starp nosūtītajiem ziņojumiem) un pievienojiet vēl vienu soli.

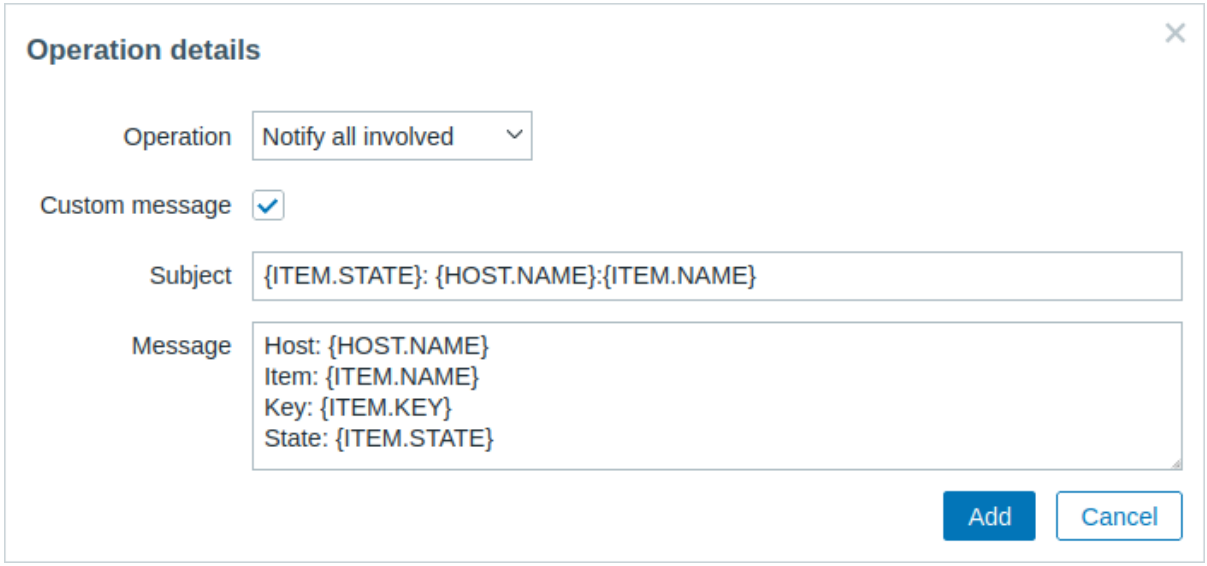
5. darbība

Atkopšanas darbību bloks ļauj konfigurēt atkopšanas paziņojumu, kad vienums atgriežas normālā stāvoklī.

1. Blokā *Atkopšanas darbības* noklikšķiniet uz *Pievienot*, lai pievienotu jaunu atkopšanas darbību.



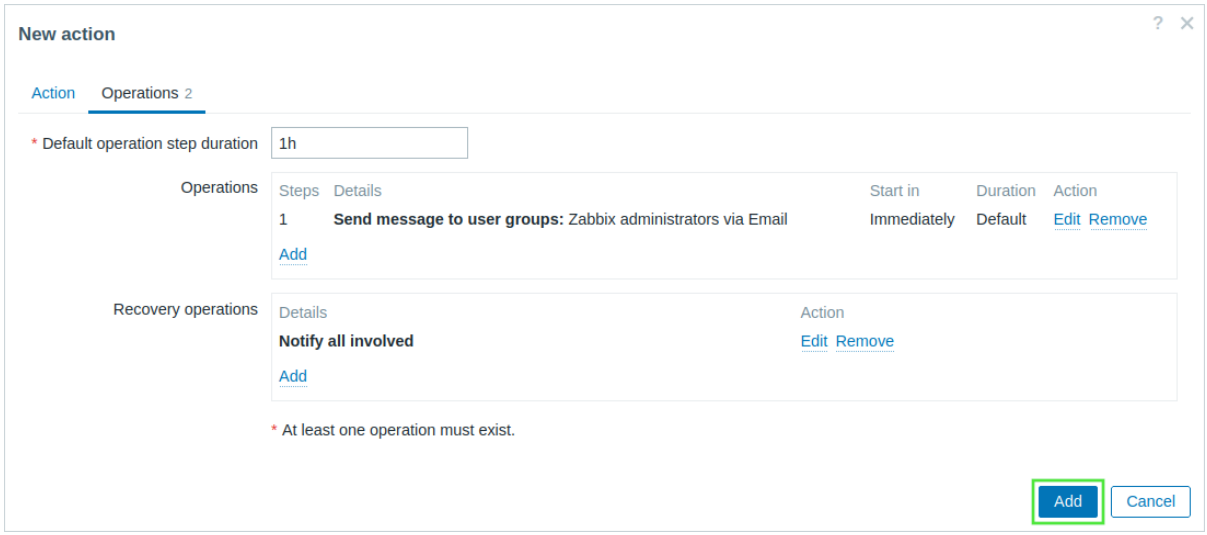
2. Atlasiet darbības tipu “Paziņot visiem iesaistītajiem”.
3. Atzīmējiet izvēles rūtiņu *Pielāgots ziņojums*, ja vēlaties ievadīt atkopšanas ziņojuma pielāgoto tēmu un saturu.



4. Uznirstošajā logā *Darbības informācija* noklikšķiniet uz *Pievienot*, lai faktiski iekļautu darbību blokā *Atkopšanas darbības*.

6. solis

Kad esat pabeidzis, formas apakšā noklikšķiniet uz pogas *Add*.



Un tas arī viss, esat pabeidzis! Tagad varat gaidīt savu pirmo paziņojumu no Zabbix, ja kāds vienums kļūst neatbalstīts.

Makrosi

Pārskats

Zabbix atbalsta vairākus iebūvētos makro, kurus var izmantot dažādās situācijās. Šie makro ir mainīgie, kurus identificē noteikta sintakse:

`{MACRO}`

Makro tiek aizstāti ar konkrētu vērtību atkarībā no konteksta.

Efektīva makro izmantošana ļauj ietaupīt laiku un padarīt Zabbix konfigurāciju pārskatāmāku.

Vienā no tipiskajiem izmantošanas veidiem makro var izmantot veidnē. Tādējādi trigeri veidnē var nosaukt par "Procesora noslodze ir pārāk augsta hostā {HOST.NAME}". Kad veidne tiek lietota hostam, piemēram, Zabbix serverim, nosaukums tiks aizstāts ar "Procesora noslodze ir pārāk augsta hostā Zabbix server" sadaļā Monitoring, kad tiek parādīts trigeris.

Makro var izmantot vienuma atslēgas parametros. Makro var izmantot tikai daļai parametra, piemēram, `item.key[server_{HOST.HOST}_l`
Parametru nav nepieciešams likt dubultpēdiņās, jo Zabbix parūpēsies par visiem neskaidrajiem speciālajiem simboliem, ja tādi būs aizstātajā makro.

Zabbix ir pieejami arī citi makro veidi.

Zabbix atbalsta šādus makro:

- {MACRO} - iebūvēts makro (skatiet pilnu sarakstu)
- {<macro>.<func>(<params>)} - makro funkcijas
- {$MACRO} - lietotāja definēts makro, pēc izvēles ar kontekstu
- {#MACRO} - makro zema līmeņa atklāšanai
- {?EXPRESSION} - izteiksmes makro

Makro funkcijas

Pārskats

Makro funkcijas nodrošina iespēju pielāgot makro vērtības (piemēram, saīsināt vai izvilkt noteiktas apakšvirknes), padarot tās ērtāk lietojamas.

Makro funkcijas sintakse ir:

```
{macro.func(params)}
```

kur

- **macro** - makro, ko pielāgot;
- **func** - funkcija, ko lietot (skatiet atbalstītās funkcijas);
- **params** - ar komatiem atdalīts funkcijas parametru saraksts, kam jābūt **iekļautam dubultpēdiņās**, ja:
  - 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 šādiem elementiem:

- Iebūvētie makro
- Lietotāja makro
- Zema līmeņa atklāšanas makro
- Izteiksmes makro

Makro funkcijas var izmantot visās vietās, kur tiek atbalstīti uzskaitītie makro. Tas attiecas uz gadījumiem, ja vien nav skaidri norādīts, ka tiek gaidīts tikai makro (piemēram, konfigurējot hosts makro vai zema līmeņa atklāšanas noteikuma filtrus).

Katram makro tiek atbalstīta viena funkcija; vairāku makro funkciju ķēde netiek atbalstīta.

> **Note:**
> Lūdzu, skatiet aizvietošanas piemērus gadījumiem, kad makro funkcijas tiek izmantotas citos kontekstos (funkcija, vienuma atslēga, cits 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 drukājamo ciparu skaitu.
fmttime	Laika formatēšana.
htmldecode	Makro vērtības dekodēšana no HTML kodējuma.
htmlencode	Makro vērtības kodēšana HTML kodējumā.
iregsub	Apakšvirknes izvilkšana, izmantojot regulārās izteiksmes atbilstību (reģistrjutība netiek ņemta vērā).
jsonpath	JSONPath rezultāta atgriešana.
lowercase	Makro vērtības rakstzīmju pārveidošana par mazajiem burtiem.
regrepl	Rakstzīmes/apakšvirknes aizstāšana makro vērtībā.
regsub	Apakšvirknes izvilkšana, izmantojot regulārās izteiksmes atbilstību (reģistrjutīga).

Funkcija	Apraksts
tr	Makro vērtības rakstzīmju transliterācija.
uppercase	Makro vērtības rakstzīmju pārveidošana par lielajiem burtiem.
urldecode	Makro vērtības dekodēšana no URL kodējuma.
urlencode	Makro vērtības kodēšana URL kodējumā.
xmlxpath	XML XPath rezultāta atgriešana.

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 makro, kas atrisina 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 formatējumu;
- **time_shift** (neobligāts) - laika nobīde, kas tiek piemērota laikam pirms formatēšanas; tai jāsākas ar `-<N><time_unit>` vai `+<N><time_unit>`, kur:
  - `N` - laika vienību skaits, ko pievienot vai atņemt;
  - `time_unit` - h (stunda), d (diena), w (nedēļa), M (mēnesis) vai y (gads).

Komentāri:

- `time_shift` parametrs atbalsta vairāku soļu laika darbības 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 darbības tiek aprēķinātas no kreisās uz labo pusi bez prioritātēm. Piemēram, `-1M/d+1h/w` tiks parsēts kā `((-1M/d)+1h)/w`.

Piemēri:

```
{{TIMESTAMP}.fmttime(%B)} - atgriezīs "October" no saņemtās vērtības "1633098961"
{{TIMESTAMP}.fmttime(%d %B,-1M/M)} - atgriezīs "1 September" no saņemtās vērtības "1633098961"
{{EVENT.TIMESTAMP}.fmttime("%x %X")} - atgriezīs lokālei atbilstošu datuma un laika attēlojumu; piemēram,
{{EVENT.TIME}.fmttime("%x %X")} - atgriezīs lokālei atbilstošu datuma un laika attēlojumu; piemēram, "07/1
```

htmldecode

Makro vērtības dekodēšana no HTML kodējuma.

Tiek atbalstītas šādas rakstzīmes:

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
```

jsonpath(pattern,<default>)

Atgriež JSONPath rezultātu.
 Atbalstītie vērtību tipi: *String*, *Text*, *Log*.

Parametri:

- **pattern** - obligātais raksts;
- **default** - neobligāta atkāpšanās vērtība, ja JSONPath vaicājums neatgriež datus. Ņemiet vērā, ka citu kļūdu gadījumā tiek atgriezta kļūda.

Komentāri:

- Funkcija neizdodas ar kļūdu, ja:
  - JSON nav derīgs.
  - JSONPath vaicājums neizdodas.
  - JSONPath rezultāts nesatur datus un parametrs `default` nav norādīts.

Piemērs:

```
{{ITEM.VALUE}.jsonpath("$.columns.servicename", "NA")}
```

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, kas jāsaskaņo;
- **replacement** - aizstāšanas virkne. **\1 - \9** vietturi tiek atbalstīti aizstāšanas virknēs, lai uztvertu grupas.

Komentāri:

- Tiek aizstātas visas atbilstības, kas sakrīt ar pattern.
- Pattern un replacement tiek apstrādāti secīgi, un katrs nākamais pāris tiek lietots atbilstoši iepriekšējās aizstāšanas rezultātam.
- Atsauces uz neesošām 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 nelatīņu rakstzīmes tiks ekrānētas ar atpakaļsvītru
{$THRESHOLD:"{{#FSNAME}.regrepl(\"\\$\",\"\")}"} - noņems beigu atpakaļsvītru (piemēram, lai aizstātu "C:\
{{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"
```

xmlxpath(pattern,<default>)

Atgriež XML XPath rezultātu.
 Atbalstītie vērtību tipi: *String*, *Text*, *Log*.

Parametri:

- **pattern** - obligātais šablons;
- **default** - neobligāta atkāpšanās vērtība, ja XPath vaicājums neatgriež datus. Ņemiet vērā, ka citu kļūdu gadījumā tiek atgriezta kļūme.

Komentāri:

- Funkcija neizdodas ar kļūdu, ja:
  - XML nav derīgs.
  - XPath vaicājums neizdodas.
  - XPath rezultāts nesatur datus un parametrs `default` nav norādīts.
  - Zabbix tika kompilēts bez XML atbalsta.

Piemērs:

```
{{ITEM.VALUE}.xmlxpath("/response/error/status","unknown")}
```

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`

Makro funkcija	Saņemtā vērtība	Izvade
{{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_1	1
{{#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.

Lietotāja makro

Pārskats

Lielākai elastībai Zabbix atbalsta lietotāja makro papildus makro, kas tiek atbalstīti jau pēc noklusējuma.

Lietotāja makro var definēt globālā, veidnes un hosts līmenī. Šiem makro ir īpaša sintakse:

{$MACRO}

Zabbix makro atrisina atbilstoši tālāk norādītajai prioritātei:

1. Hosts līmeņa makro (tiek pārbaudīti vispirms).
2. Makro, kas definēti hosts pirmā līmeņa veidnēs (t. i., veidnēs, kas tieši piesaistītas hosts), sakārtoti pēc veidnes ID.
3. Makro, kas definēti hosts otrā līmeņa veidnēs, sakārtoti pēc veidnes ID.
4. Makro, kas definēti hosts trešā līmeņa veidnēs, sakārtoti pēc veidnes ID utt.
5. Globālie makro (tiek pārbaudīti pēdējie).

Citiem vārdiem, ja hosts makro nepastāv, Zabbix mēģinās to atrast hosts veidnēs, sākot ar mazāko ligzdošanas dziļumu. Ja makro joprojām netiek atrasts, tiks izmantots globālais makro, ja tāds pastāv.

Warning:
Ja vairāku viena līmeņa piesaistīto veidņu makro ir ar **vienādu nosaukumu**, tiks izmantots makro no veidnes ar mazāko ID. Tāpēc vienāda nosaukuma makro izmantošana vairākās veidnēs rada konfigurācijas risku.

Ja Zabbix nevar atrast makro, makro netiks atrisināts.

Attention:
Makro (tostarp lietotāja makro) sadaļā Configuration netiek atrisināti (piemēram, trigeru sarakstā), lai pēc noklusējuma padarītu sarežģītu konfigurāciju pārskatāmāku.

Lietotāja makro 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.

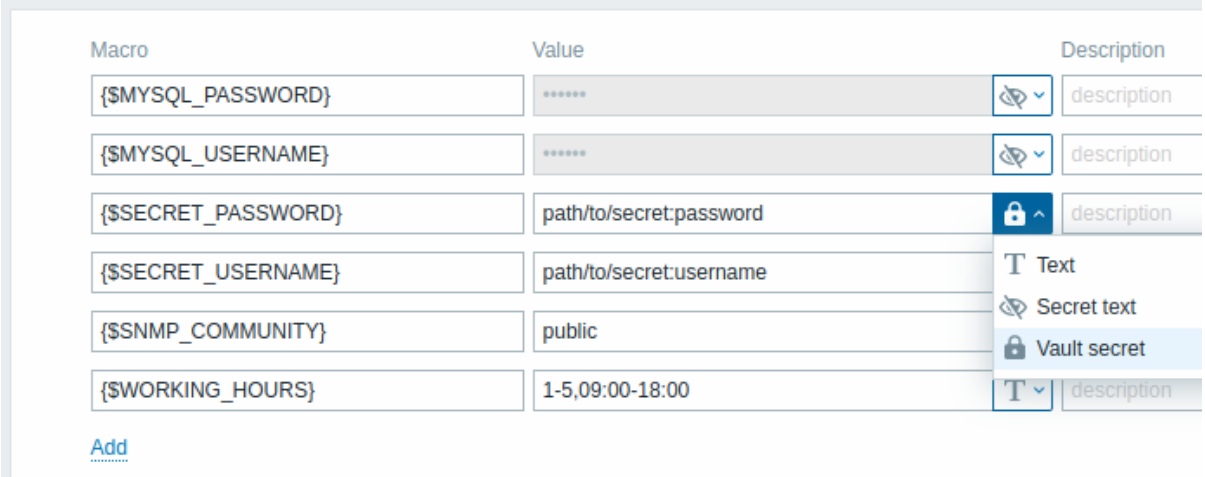
Note:
Ieteicams izmantot hosta makro globālo makro vietā, jo globālo makro pievienošana, atjaunināšana vai dzēšana piespiež veikt pakāpenisku konfigurācijas atjaunināšanu visiem hostiem. Papildinformāciju skatiet Pasīvie un aktīvie aģenta pārbaudes.

Konfigurācija

Lai definētu lietotāja makro, dodieties uz atbilstošo vietu lietotāja saskarnē:

- Globālajiem makro atveriet *Administration* > *Macros*.
- Host un veidnes līmeņa makro atveriet hosta vai veidnes rekvizītus un atrodiet cilni *Macros*.

Lietotāja makro ir šādi atribūti:



Parameter	Description
Macro	Makro nosaukums. Nosaukumam jābūt ietvertam figūriekavās un jāsākas ar dolāra zīmi. Piemērs: `{$FRONTEND_URL}`. Makro nosaukumos ir atļautas šādas rakstzīmes: **A-Z** (tikai lielie burti), **0-9**, _, **.**
Value	Makro 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. Maksimālais lietotāja makro vērtības garums ir 2048 rakstzīmes.
Description	Teksta lauks, ko izmanto, lai sniegtu papildu informāciju par šo makro.

Konfigurējot lietotāja makro, **ņemiet vērā** šādu no konteksta atkarīgu uzvedību:

- ja lietotāja makro tiek izmantoti veidnes vienumos vai trigeros, apsveriet iespēju pievienot šos makro arī veidnei (pat ja tie ir definēti globāli); tādā veidā *Text* tipa makro darbosies, kā paredzēts, pēc veidnes eksportēšanas uz XML un importēšanas citā sistēmā (slepeno makro vērtības netiek eksportētas)
- ja lietotāja makro tiek izmantoti trigeru izteiksmēs, šie makro 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 trigeru izteiksmi (slepenos makro nevar izmantot trigeru izteiksmēs)
- ja lietotāja makro tiek izmantoti hostā, kuram ir zemā līmeņa atklāšanas noteikums ar hostu prototipiem, atklātie hosti pārmantos visus šajā hostā definētos lietotāja makro

Piemēri

1. piemērs

Hosts līmeņa makro izmantošana vienuma "SSH dēmona statuss" atslēgā:

```
net.tcp.service[ssh,,{$SSH_PORT}]
```

Šo vienumu var piešķirt vairākiem hostiem, ja vien šajos hostos ir definēta `{$SSH_PORT}` vērtība.

2. piemērs

Hosts līmeņa makro izmantošana trigerī "CPU slodze ir pārāk augsta":

```
last(/ca_001/system.cpu.load[,avg1])>{$MAX_CPULOAD}
```

Šāds trigeris tiktu izveidots veidnē, nevis rediģēts atsevišķos hosts.

Note:
Ja vēlaties izmantot vērtību skaitu kā funkcijas parametru (piemēram, `max(/host/key,#3)`), makro definīcijā iekļaujiet restītes simbolu, piemēram: `SOME_PERIOD => #3`.

3. piemērs

Divu makro izmantošana trigerī "CPU slodze ir pārāk augsta":

```
min(/ca_001/system.cpu.load[,avg1],{$CPULOAD_PERIOD})>{$MAX_CPULOAD}
```

Ņemiet vērā, ka makro var izmantot kā trigeru funkcijas parametru, šajā piemērā — funkcijas `min()` parametru.

4. piemērs

Sinhronizējiet aģenta nepieejamības nosacījumu ar vienuma atjaunināšanas intervālu:

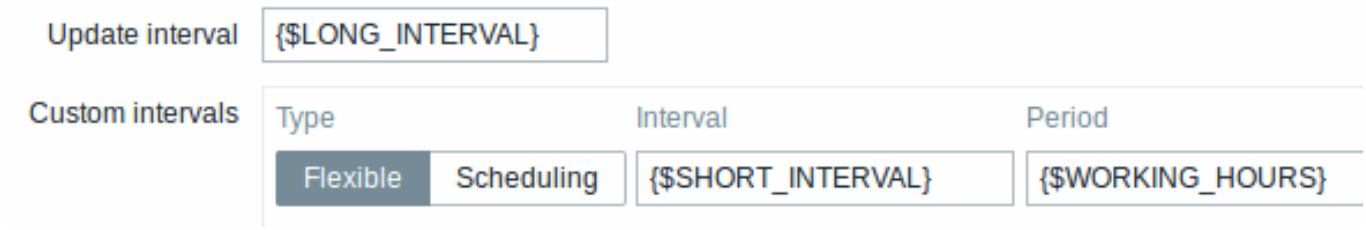
- definējiet makro `{$INTERVAL}` un izmantojiet to vienuma atjaunināšanas intervālā;
- izmantojiet `{$INTERVAL}` kā aģenta nepieejamības trigera parametru:

```
nodata(/ca_001/agent.ping,{$INTERVAL})=1
```

5. piemērs

Centralizējiet darba laika konfigurāciju:

- izveidojiet globālo makro `{$WORKING_HOURS}`, kura vērtība ir `1-5,09:00-18:00`;
- izmantojiet to laukā *Darba laiks* sadaļā *Administrēšana* > *Vispārīgi* > *Lietotāja saskarne*;
- izmantojiet to lietotāja cilnes *Multivide* laukā *Kad aktīvs* sadaļā *Lietotāji* > *Lietotāji*;
- izmantojiet to, lai iestatītu biežāku vienumu aptauju darba laikā:

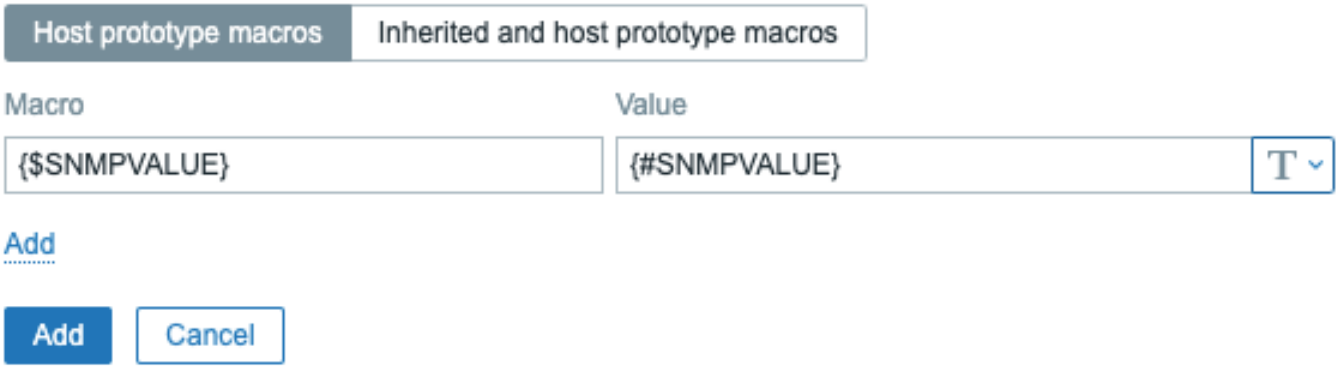


- izmantojiet to darbības nosacījumā *Laika periods*;
- ja nepieciešams, pielāgojiet darba laiku sadaļā *Administrēšana* > *Makro*.

6. piemērs

Izmantojiet hosts prototipa makro, lai konfigurētu vienumus atklātajiem hostiem:

1. Hosts prototipā definējiet lietotāja makro {$SNMPVALUE}, kā vērtību izmantojot zema līmeņa atklāšanas makro {#SNMPVALUE}:



2. Piešķiriet *Generic SNMPv2* veidni hosts prototipam.
3. Izmantojiet {$SNMPVALUE} lauka *SNMP OID* vērtībā *Generic SNMPv2* veidnes vienumiem.

Lietotāja makro konteksts

Skatiet lietotāja makro ar kontekstu.

Lietotāja makro ar kontekstu

Pārskats

lietotāja makros var izmantot neobligātu kontekstu, kas ļauj noklusējuma vērtību aizstāt ar kontekstam specifisku vērtību.

Konteksts tiek pievienots makro nosaukumam; sintakse ir atkarīga no tā, vai konteksts ir statiska teksta vērtība:

```
{$MACRO:"static text"}
```

vai regulāra izteiksme:

```
{$MACRO:regex:"regular expression"}
```

Ņemiet vērā, ka makro ar regulāras izteiksmes kontekstu var definēt tikai lietotāja makro konfigurācijā. Ja prefikss `regex:` tiek izmantots citur kā lietotāja makro konteksts, piemēram, trigeru izteiksmē, tas tiks interpretēts kā statisks konteksts.

Konteksta ietveršana pēdiņās nav obligāta (skatiet arī svarīgas piezīmes).

Makro konteksta piemēri:

Piemērs	Apraksts
{$LOW_SPACE_LIMIT}	Lietotāja makro bez konteksta.
{$LOW_SPACE_LIMIT:/tmp}	Lietotāja makro ar kontekstu (statiska virkne).
{$LOW_SPACE_LIMIT:regex:"^/tmp$"}	Lietotāja makro ar kontekstu (regulāra izteiksme). Tas ir tāds pats kā {$LOW_SPACE_LIMIT:/tmp}.
{$LOW_SPACE_LIMIT:regex:"^/var/log/.*$"}	Lietotāja makro ar kontekstu (regulāra izteiksme). Atbilst visām virknēm, kas sākas ar /var/log/.

Lietošanas gadījumi

Lietotāja makro ar kontekstu var definēt, lai izveidotu elastīgākus sliekšņus trigeru izteiksmēs (pamatojoties uz vērtībām, kas iegūtas ar zemā līmeņa atklāšanu). Piemēram, varat definēt šādus makro:

- {$LOW_SPACE_LIMIT} = 10
- {$LOW_SPACE_LIMIT:/home} = 20
- {$LOW_SPACE_LIMIT:regex:"^/[a-z]+$"} = 30

Pēc tam zemā līmeņa atklāšanas makro var izmantot kā makro kontekstu trigeru prototipā piemontētas failu sistēmas atklāšanai:

```
last(/host/vfs.fs.size[{#FSNAME},pfree])<{$LOW_SPACE_LIMIT:"{#FSNAME}"}
```

Pēc atklāšanas trigeros tiks piemēroti dažādi zema brīvās vietas sliekšņi atkarībā no atklātajiem montēšanas punktiem vai failu sistēmu tipiem. Problēmu notikumi tiks ģenerēti, ja:

- /home mapē būs mazāk nekā 20% brīvās diska vietas
- mapēm, kas atbilst regulārās izteiksmes paraugam (piemēram, /etc, /tmp vai /var), būs mazāk nekā 30% brīvās diska vietas

- mapēm, kas neatbilst regulārās izteiksmes paraugam un nav /home, būs mazāk nekā 10% brīvās 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.

Slepenie lietotāja makrosi

Pārskats

Zabbix nodrošina divas iespējas sensitīvas informācijas aizsardzībai lietotāja makro vērtībās:

- Slepens teksts
- Vault noslēpums

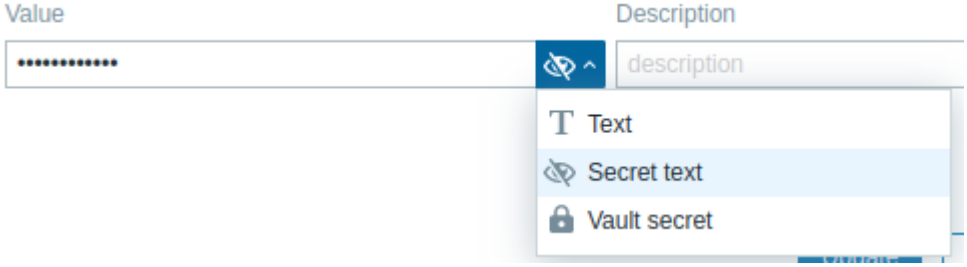
Note:
Lai gan slepena makro vērtība ir paslēpta, to var atklāt, izmantojot to vienumos. Piemēram, ārējā skriptā echo priekšrakstu, kas atsaucas uz slepenu makro, var izmantot, lai atklātu makro vērtību lietotāja saskarnē, jo Zabbix serveris ir piekļuve reālajai makro vērtībai. Skatiet vietas, kur slepeno makro vērtības tiek parādītas neatmaskotas.

Slepenos makro nevar izmantot trigeru izteiksmēs.

Slepenais teksts

Izmantojot Slepenā teksta makro, makro vērtība tiek maskēta ar zvaigznītēm.

Lai makro vērtību padarītu slepenu, noklikšķiniet uz pogas lauka *Vērtība* beigās un atlasiet opciju *Slepenais teksts*:



Kad konfigurācija ir saglabāta, vērtību vairs nebūs iespējams skatīt.

Lai mainītu makro vērtību, novietojiet kursoru virs lauka *Vērtība* un noklikšķiniet uz pogas *Iestatīt jaunu vērtību* (tā parādās, kad uzvedat kursoru):



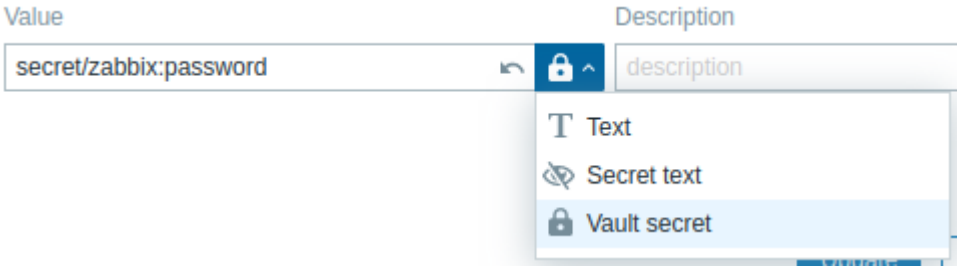
Kad noklikšķināt uz pogas *Iestatīt jaunu vērtību* (vai mainīt makro vērtības tipu), pašreizējā vērtība tiks dzēsta. Sākotnējo vērtību varat atjaunot, noklikšķinot uz bultiņas lauka *Vērtība* beigās (pieejama tikai pirms jaunās konfigurācijas saglabāšanas). Ņemiet vērā, ka sākotnējās vērtības atjaunošana to neatklās.

Note:
URL, kas satur slepeno makro, nedarbosies, jo tajos esošais makro tiks aizstāts ar "******".

Vault secret

Ar Vault secret makro makro vērtība tiek glabāta ārējā slepeno datu pārvaldības programmatūrā (vault).

Lai konfigurētu Vault secret makro, noklikšķiniet uz pogas lauka *Value* beigās un atlasiet opciju *Vault secret*:



Makro vērtībai ir jānorāda uz vault secret. Ievades formāts ir atkarīgs no vault nodrošinātāja. Konkrēta nodrošinātāja konfigurācijas piemērus skatiet:

- HashiCorp
- CyberArk

Vault slepeno makro vērtības no glabātuves izgūst 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 noslēpumus izgūst 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 izgūt, atbilstošais vienums, kas izmanto šo vērtību, kļūs neatbalstīts.

Nemaskētās atrašanās vietas

Šis saraksts norāda parametru atrašanās vietas, kur slepeno makro vērtības netiek maskētas.

Note:
Slepeno makro vērtības tālāk norādītajās atrašanās vietās paliks maskētas, ja uz tām atsauksies netieši. Piemēram, {ITEM.KEY}, {ITEM.KEY<1-9>}, {LLDRULE.KEY} iebūvētie makro, kas izmantoti mediju tipos (Script vai Webhook parametri), tiks atrisināti uz vienumu atslēgām, kurās ir maskēti slepenie makro, piemēram, `net.tcp.port[******,******]` nevis `net.tcp.port[192.0.2.1,80]`.

Konteksts		Parametrs
Vienumi, vienumu prototipi, LLD noteikumi		
	Vienums	*Vienuma atslēgas parametri*
	Vienuma prototips	*Vienuma prototipa atslēgas parametri*
	Zema līmeņa atklāšanas noteikums	*Atklāšanas vienuma atslēgas parametri*
	SNMP aģents	*SNMP kopiena* *Konteksta nosaukums* (SNMPv3) *Drošības nosaukums* (SNMPv3) *Autentifikācijas parole* (SNMPv3) *Privātuma parole* (SNMPv3)

Konteksts		Parametrs
	HTTP aģents	*URL*
		Vaicājuma lauki
		Pieprasījuma pamatteksts
		Galvenes
		Lietotājvārds
		Parole
		SSL atslēgas parole
	Script	*Parametri*
		Script
	Browser	*Parametri*
		Script
	Datubāzes monitors	*SQL vaicājums*
	TELNET aģents	*Script*
		Lietotājvārds
		Parole
	SSH aģents	*Script*
		Lietotājvārds
		Parole
	Vienkārša pārbaude	*Lietotājvārds*
		Parole
	JMX aģents	*Lietotājvārds*
		Parole
Vienuma vērtības pirmapstrāde		
	JavaScript pirmapstrādes solis	*Script*
Tīmekļa scenāriji		
	Tīmekļa scenārijs	*Mainīgā vērtība*
		Galvenes vērtība
		URL
		Vaicājuma lauka vērtība
		Post lauka vērtība
		Neapstrādāts post
	Tīmekļa scenārija autentifikācija	*Lietotājs*
		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
	Script	*Script*
Mediju tipi		
	Script	*Script parametri*
	Webhook	*Parametri*
IPMI pārvaldība		

Konteksts	Parametrs
Hosts	*Lietotājvārds* *Parole*

Zema līmeņa atklāšanas makro

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.

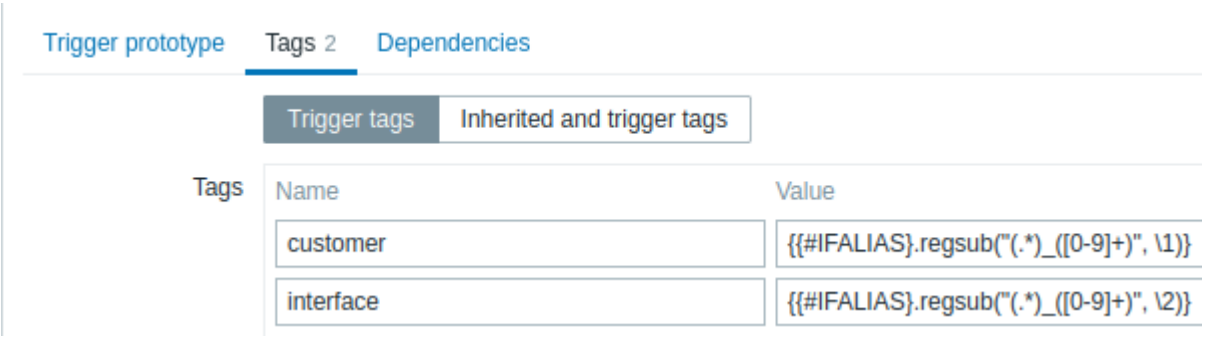
Makro funkciju izmantošana

Makro funkcijas tiek atbalstītas zema līmeņa atklāšanas makros (izņemot zema līmeņa atklāšanas noteikuma filtrā), ļaujot, izmantojot regulāro izteiksmi, iegūt noteiktu makro vērtības daļu.

Piemēram, notikumu marķēšanai var būt nepieciešams iegūt klienta nosaukumu un saskarnes numuru no tālāk norādītā LLD makro:

```
{#IFALIAS}=customername_1
```

Lai to izdarītu, trigerim prototipa notikuma taga vērtības laukā kopā ar makro var izmantot `regsub` makro funkciju:



Ņemiet vērā, ka neiekļautiem vienuma atslēgas parametriem komati nav atļauti, tāpēc parametrs, kas satur makro funkciju, ir jāiekļauj pēdiņās. Lai parametra iekšpusē aizsargātu dubultpēdiņas, jāizmanto atpakaļvērstā slīpsvītra (\). Piemērs:

```
net.if.in["{{#IFALIAS}.regsub(\"(.*)_([0-9]+)\", \1)}",bytes]
```

Plašāku informāciju par makro funkciju sintaksi skatiet šeit: 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.

Izteiksmes makrosi

Pārskats

Izteiksmes makro ļauj veikt aprēķinus laukos.

To vērtība tiek aprēķināta, vispirms atrisinot visus iekšējos makro un pēc tam izvērtējot 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}
```

Vietās, kur tiek atbalstīti izteiksmes makro, meklējiet "{?EXPRESSION}" atbalstīto makro tabulā.

Piezīmes par lietošanu:

- Izmantojiet {FUNCTION.*} makro, lai atsauktos uz trigeru izteiksmju/atjaunošanas izteiksmju funkciju vērtībām.
- Izmantojiet {HOST.HOST<1-9>} un {ITEM.KEY<1-9>} makro, lai atsauktos uz hostiem un vienumiem.
- Veidnēs izmantojiet {HOST.HOST<1-9>} makro vai pirmajam hostam vispār nenorādiet hostu — piemēram, {?avg(//item1,1h)} — nevis veidņu nosaukumus, jo veidņu nosaukumi veidnes saistīšanas laikā netiek aizstāti ar hostu nosaukumiem.

```
{?{FUNCTION.VALUE2} - {FUNCTION.VALUE3}}
{?max(/{HOST.HOST}/{ITEM.KEY},3h)}
```

Skatiet arī trigeru izteiksmju piemērus, kur parādīts izteiksmes makro lietojums notikuma nosaukumā.

Lietotāji un lietotāju grupas

Pārskats

Visi Zabbix lietotāji piekļūst Zabbix lietotnei, izmantojot tīmekļa lietotāja saskarni. Katram lietotājam tiek piešķirts unikāls pieteikšanās vārds un parole.

Visas lietotāju paroles tiek 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 pārlūkprogrammu var aizsargāt, izmantojot SSL.

Izmantojot elastīgu lietotāju atļauju shēmu, varat ierobežot un diferencēt tiesības uz:

- Piekļuvi Zabbix administratīvajām lietotāja saskarnes funkcijām.
- Veikt noteiktas darbības lietotāja saskarnē.
- Piekļuvi uzraudzītajiem hostiem hostu grupās.
- Izmantot noteiktas API metodes.

Lietotāja konfigurēšana

Pārskats

Sākotnējā Zabbix instalācijā ir divi iepriekš definēti lietotāji:

- *Admin* - Zabbix superlietotājs ar pilnām atļaujām.
- *guest* - īpašs Zabbix lietotājs.

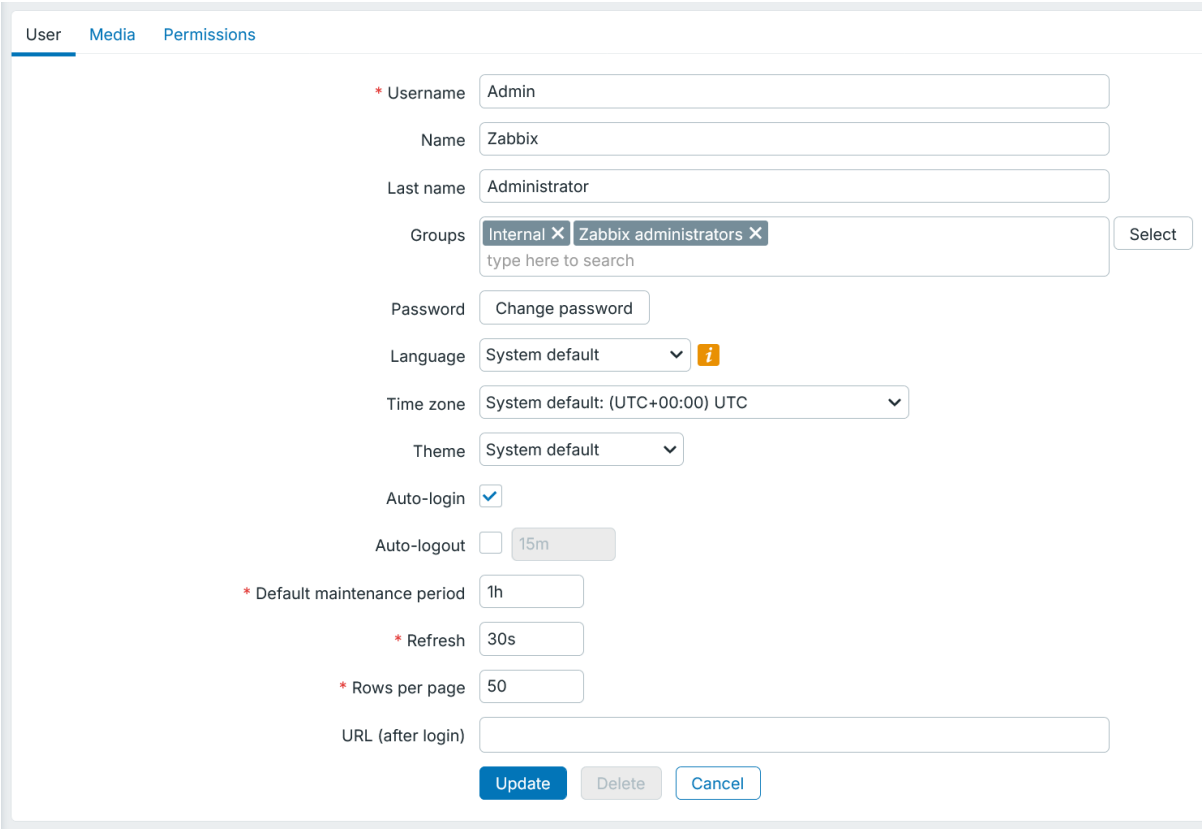
Lietotājs 'guest' pēc noklusējuma ir atspējots.
Lai iespējotu lietotāju, no lietotāju grupām noņemiet iepriekš definēto grupu *Disabled*.
Varat pieteikties ar šo lietotāju un piekļūt Zabbix uzraudzības lapām.
Ņemiet vērā, ka pēc noklusējuma lietotājam 'guest' nav atļauju piekļūt Zabbix objektiem.

Lai konfigurētu lietotāju:

1. Atveriet *Users*> *Users*.
2. Noklikšķiniet uz *Create user* (vai uz lietotāja vārda, lai rediģētu esošu lietotāju).
3. Rediģējiet lietotāja atribūtus veidlapā.

Vispārīgie atribūti

Cilnē *Lietotājs* ir ietverti vispārīgie lietotāja atribūti:



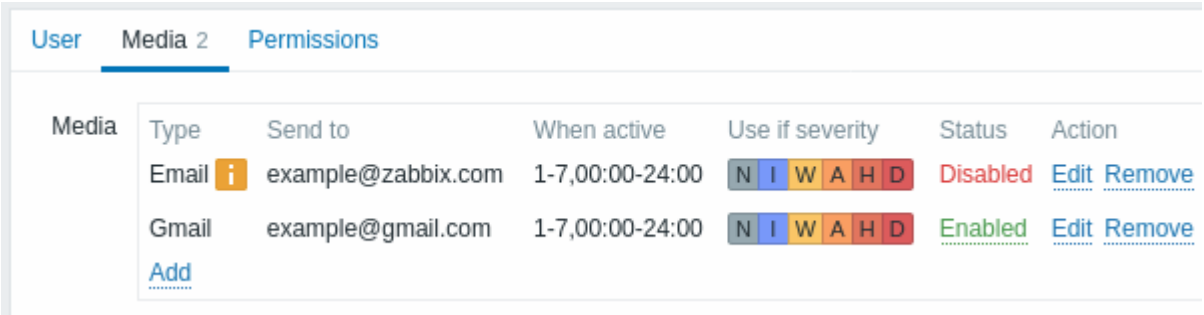
Visi obligātie ievades lauki ir atzīmēti ar sarkanu zvaigznīti.

Parametrs	Apraksts
Lietotājvārds	Unikāls lietotājvārds, kas tiek izmantots kā pieteikšanās vārds.
Vārds	Lietotāja vārds (neobligāts). Ja nav tukšs, tas ir redzams apstiprinājuma informācijā un paziņojuma saņēmēja informācijā.
Uzvārds	Lietotāja uzvārds (neobligāts). Ja nav tukšs, tas ir redzams apstiprinājuma informācijā un paziņojuma saņēmēja informācijā.
Grupas	Atlasiet lietotāju grupas, kurām lietotājs pieder. Šis lauks atbalsta automātisko pabeigšanu, tāpēc, sākot rakstīt lietotāju grupas nosaukumu, tiks parādīts atbilstošo grupu nolaižamais saraksts. Ritiniet uz leju, lai atlasītu grupu. Varat arī noklikšķināt uz *Atlasīt*, lai pievienotu grupas. Noklikšķiniet uz "x", lai noņemtu atlasīto grupu. Piederība lietotāju grupām nosaka, kurām resursdatoru grupām un hostiem lietotājam būs piekļuve.
Parole	Divi lauki lietotāja paroles ievadīšanai vai poga *Mainīt paroli*, ja lietotājs jau pastāv. Noklikšķinot uz pogas *Mainīt paroli*, tiek atvērti divi lauki jaunas paroles ievadīšanai. Lietotājam ar *Super administratora lomu*, mainot savu paroli, noklikšķinot uz pogas *Mainīt paroli*, 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.
Valoda	Zabbix lietotāja saskarnes valoda. Lai tulkojumi darbotos, ir nepieciešams php gettext paplašinājums.
Laika josla	Atlasiet laika joslu, lai lietotāja līmenī aizstātu globālo laika joslu, vai atlasiet **Sistēmas noklusējums**, lai izmantotu globālos laika joslas iestatījumus.

Parametrs	Apraksts
Motīvs	Nosaka lietotāja saskarnes izskatu: **Sistēmas noklusējums** - izmantot sistēmas noklusējuma iestatījumus **Zils** - standarta zilais motīvs **Zils (klasiskais)** - standarta zilais motīvs, izmantojot mantotos fontus **Tumšs** - alternatīvs tumšais motīvs **Tumši zils** - tumši zilais motīvs **Tumšs (klasiskais)** - alternatīvs tumšais motīvs, izmantojot mantotos fontus **Gaišs ar augstu kontrastu** - gaišs motīvs ar augstu kontrastu **Tumšs ar augstu kontrastu** - tumšs motīvs ar augstu kontrastu
Automātiskā pieteikšanās	Atzīmējiet šo izvēles rūtiņu, lai Zabbix atcerētos lietotāju un automātiski pieteiktu viņu 30 dienas. Piesakoties ar opciju *Atcerēties 30 dienas*: - Automātiskā izrakstīšanās tiek atiestatīta (sesija tiek saglabāta 30 dienas). - Automātiskā pieteikšanās tiek iespējota netraucētai atkārtotai autentifikācijai. Piesakoties bez opcijas *Atcerēties 30 dienas*: - Automātiskā pieteikšanās tiek iespējota netraucētai atkārtotai autentifikācijai. - Automātisko izrakstīšanos joprojām nosaka standarta taimauta iestatījums. Šim nolūkam tiek izmantoti pārlūkprogrammas sīkfaili.
Automātiskā izrakstīšanās	Ja šī izvēles rūtiņa ir atzīmēta, lietotājs tiks automātiski izrakstīts pēc iestatītā sekunžu skaita (vismaz 90 sekundes, ne vairāk kā 1 diena). Ņemiet vērā, ka šis iestatījums tiek ignorēts, ja ir iespējota opcija *Atcerēties 30 dienas*, jo sesija tiek pagarināta uz visu šo periodu. Tiek atbalstīti laika sufiksi, piemēram, 90s, 5m, 2h, 1d. Ņemiet vērā, ka šī opcija nedarbosies: * Ja ir iespējota globālās konfigurācijas opcija “Rādīt brīdinājumu, ja Zabbix serveris nedarbojas” un Zabbix lietotāja saskarne ir atstāta atvērta. * Kad uzraudzības izvēlnes lapas veic informācijas atsvaidzināšanu fonā. * Ja piesakāties ar atzīmētu opciju *Atcerēties mani 30 dienas*.
Noklusējuma uzturēšanas periods	Iestatiet noklusējuma uzturēšanas periodu ad hoc uzturēšanai.
Atsvaidzināšana	Iestatiet atsvaidzināšanas biežumu, ko izmanto grafikiem, vienkārša teksta datiem u. c. Var iestatīt vērtību 0, lai to atspējotu. Tiek atbalstīti laika sufiksi, piemēram, 90s, 5m, 1h.
Rindas lapā	Varat noteikt, cik rindu lapā tiks parādīts sarakstos.
URL (pēc pieteikšanās)	Varat iestatīt, lai Zabbix pēc veiksmīgas pieteikšanās pārsūtītu lietotāju uz noteiktu URL, piemēram, uz lapu *Problēmas*.

Lietotāja multivide

Cilnē *Multivide* ir uzskaitīta visa lietotājam definētā multivide.
Multivide tiek izmantota paziņojumu nosūtīšanai.



Noklikšķiniet uz *Pievienot*, lai piešķirtu lietotājam multividi.

Ja multivides tips ir atspējots:

- Pēc nosaukuma tiek parādīta dzeltena informācijas ikona.
- Statusa kolonnā tiek parādīts *Atspējots*.

Note:
Lietotāja atļaujas mainīt savas multivides informāciju var piešķirt vai atsaukt, pamatojoties uz viņa lietotāja lomu (skatiet opciju *Izveidot un rediģēt savu multividi*). Super administratora lietotāja atļaujas mainīt citu lietotāju multivides informāciju arī var piešķirt vai atsaukt, pamatojoties uz viņa lietotāja lomu (skatiet opciju *Izveidot un rediģēt lietotāju multividi*).

Detalizētu informāciju par lietotāja multivides konfigurēšanu skatiet sadaļā Multivides 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.

Atļaujas

Pārskats

Atļaujas Zabbix ir atkarīgas no lietotāja tipa, pielāgotām lietotāju lomām un piekļuves hostiem, kas tiek noteikta, pamatojoties uz lietotāju grupu.

Lietotāju tipi

Zabbix atļaujas galvenokārt ir atkarīgas no lietotāja tipa:

- *Lietotājs* - ir ierobežotas piekļuves tiesības izvēlnes sadaļām (skatiet tālāk) un pēc noklusējuma nav piekļuves nevienam resursam. Jebkādas atļaujas hosts vai veidņu grupām ir jāpiešķir tieši.
- *Administrators* - ir nepilnīgas piekļuves tiesības izvēlnes sadaļām (skatiet tālāk). Pēc noklusējuma lietotājam nav piekļuves nevienai hosts grupai. Jebkādas atļaujas hosts vai veidņu grupām ir jāpiešķir tieši.
- *Superadministrators* - ir piekļuve visām izvēlnes sadaļām. Lietotājam ir lasīšanas un rakstīšanas piekļuve visām hosts un veidņu grupām. Atļaujas nevar atsaukt, liedzot piekļuvi konkrētām grupām.

Piekļuve izvēlnei

Note:
Ierobežota piekļuve dažiem lietotāja saskarnes elementiem tikai neļauj atvērt attiecīgo lapu — tā nenoņem iespēju piekļūt pamatā esošajiem datiem citās saskarnes daļās.

Šajā tabulā ir parādīta piekļuve Zabbix izvēlnes sadaļām atkarībā no lietotāja tipa:

Izvēlnes sadaļa		Lietotājs	Administrators	Superadministrators
Informācijas paneļi		+	+	+
Uzraudzība		+	+	+
	Problēmas	+	+	+
	Hosts	+	+	+
	Jaunākie dati	+	+	+
	Kartes	+	+	+
	Atklāšana		+	+
APM		+	+	+

Izvēlnes sadaļa		Lietotājs	Administrators	Superadministrators
	Trases	+	+	+
	Metrikas	+	+	+
	Žurnāli	+	+	+
Pakalpojumi		+	+	+
	Pakalpojumi	+	+	+
	SLA		+	+
	SLA pārskats	+	+	+
Inventārs		+	+	+
	Pārskats	+	+	+
	Hosts	+	+	+
Pārskati		+	+	+
	Sistēmas informācija		+	+
	Ieplānotie pārskati		+	+
	Pieejamības pārskats	+	+	+
	100 populārākie trigeri	+	+	+
	Audita žurnāls			+
	Darbību žurnāls			+
	Paziņojumi		+	+
Datu apkopošana			+	+
	Veidņu grupas		+	+
	Hosts grupas		+	+
	Veidnes		+	+
	Hosts		+	+
	Uzturēšana		+	+
	Notikumu korelācija			+
	Atklāšana		+	+
Brīdinājumi			+	+
	Trigeru darbības		+	+
	Pakalpojumu darbības		+	+
	Atklāšanas darbības		+	+
	Automātiskās reģistrācijas darbības		+	+
	Iekšējās darbības		+	+
	Multivides tipi			+
	Skripti			+
Lietotāji				+
	Lietotāju grupas			+
	Lietotāju lomas			+
	Lietotāji			+
	API marķieri			+
	Autentifikācija			+
	Ierīces			+
Administrēšana				+
	Datu avots			+
	Vispārīgi			+
	Audita žurnāls			+
	Apkopes darbi			+
	Starpniekserveru grupas			+
	Starpniekserveri			+
	Makrosi			+
	Rinda			+

Lietotāju lomas

Lietotāju lomas ļauj pielāgot lietotāja tipam definētās atļaujas.
Lai gan atļaujas nevar pievienot (tas pārsniegtu lietotāja tipam piešķirtās 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 Super admin lietotāji konfigurē sadaļā *Users* > *User roles*.

Super admin lietotāji lietotāju lomām piešķir lietotājus lietotāja konfigurācijas veidlapas cilnē *Permissions*.

Users

User | Media | Permissions

* Role: Admin role ✕ | Select

User type: Admin

Permissions

User group permissions

Group	Type	Permissions
All groups	Hosts	None
All groups	Templates	None

Proxies: None

Proxy groups: None

Permissions can be assigned for user groups only.

Access to UI elements

Dashboards: Dashboards

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

APM: Traces | Metrics | Logs

Services: Services | SLA | SLA report

Inventory: Overview | Hosts

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

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

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

User profile settings

Redirect URL after login

Access to services

Read-write access to services: None

Read-only access to services: All

Access to modules

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

Access to API

Disabled

Devices

Enabled | Manage own devices

Access to actions

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

Add | Cancel

Piekļuve hostiem

Piekļuve jebkura hosta un veidnes datiem Zabbix sistēmā 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, kurā atrodas šis hosts.

Līdzīgi lietotājam piekļuvi veidnei var piešķirt tikai tad, ja viņš ir daļa no lietotāju grupas, kurai ir piešķirta piekļuve veidņu grupai, kurā atrodas šī veidne.

Lietotāju grupas

Pārskats

Lietotāju grupas ļauj grupēt lietotājus gan organizatoriskos nolūkos, gan datu piekļuves tiesību piešķiršanai. Tiesības skatīt un konfigurēt hostu grupu un veidņu grupu datus tiek piešķirtas lietotāju grupām, nevis atsevišķiem lietotājiem.

Bieži vien ir lietderīgi nošķirt, kāda informācija ir pieejama vienai lietotāju grupai un kāda — citai. To var panākt, grupējot lietotājus un pēc tam piešķirot atšķirīgas tiesības piekļūt hostu un veidņu grupām.

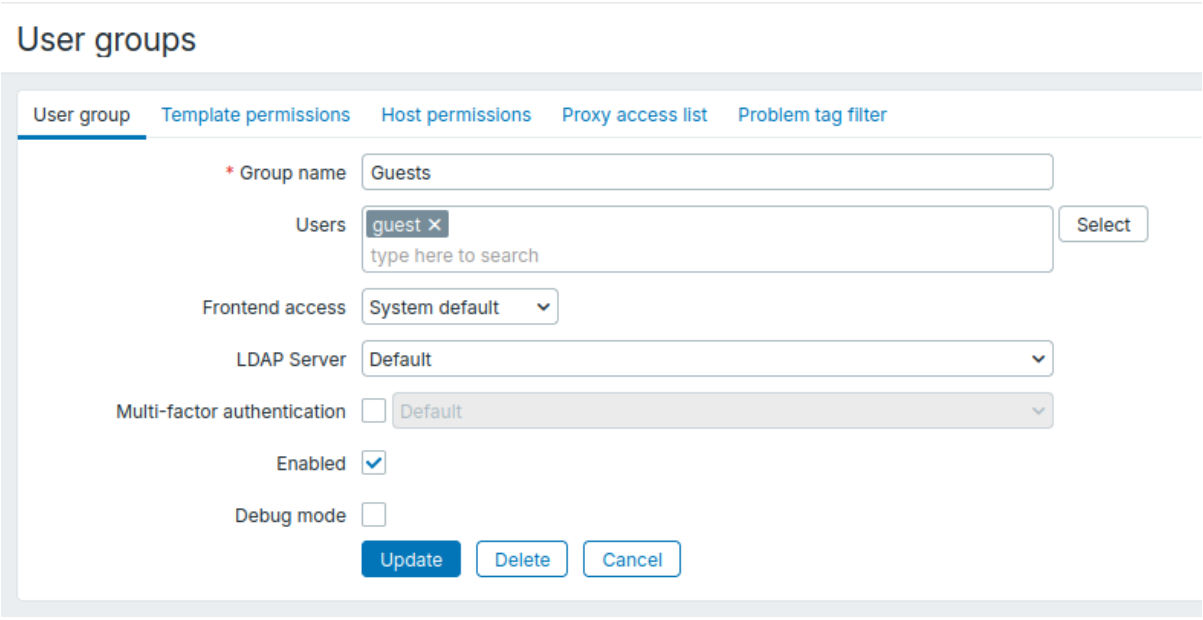
Lietotājs var piederēt neierobežotam skaitam grupu.

Konfigurēšana

Lai konfigurētu lietotāju grupu:

1. Atveriet *Users* > *User groups*.
2. Noklikšķiniet uz *Create user group* (vai uz grupas nosaukuma, lai rediģētu esošu grupu).
3. Rediģējiet grupas atribūtus veidlapā.

Cilnē **User group** ir ietverti vispārīgie grupas atribūti:

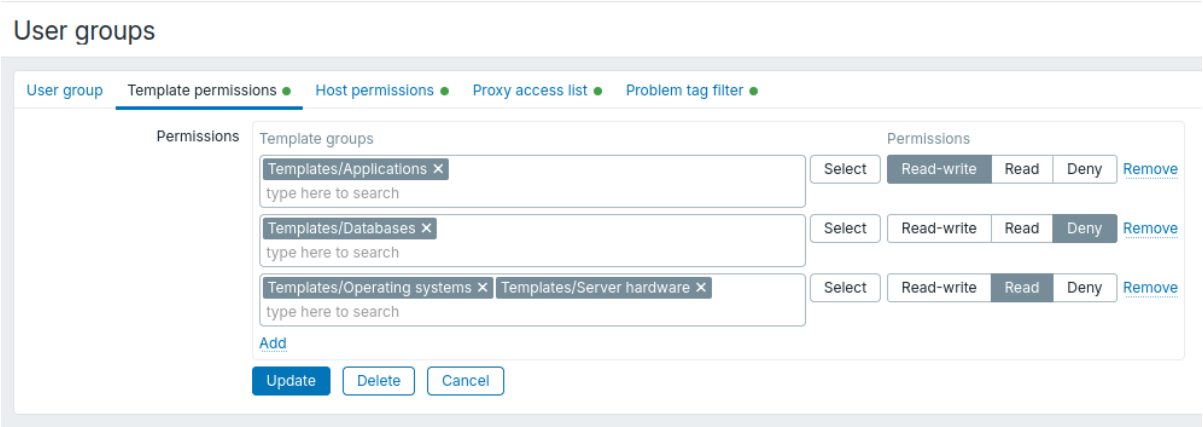


Visi obligātie ievades lauki ir atzīmēti ar sarkanu zvaigznīti.

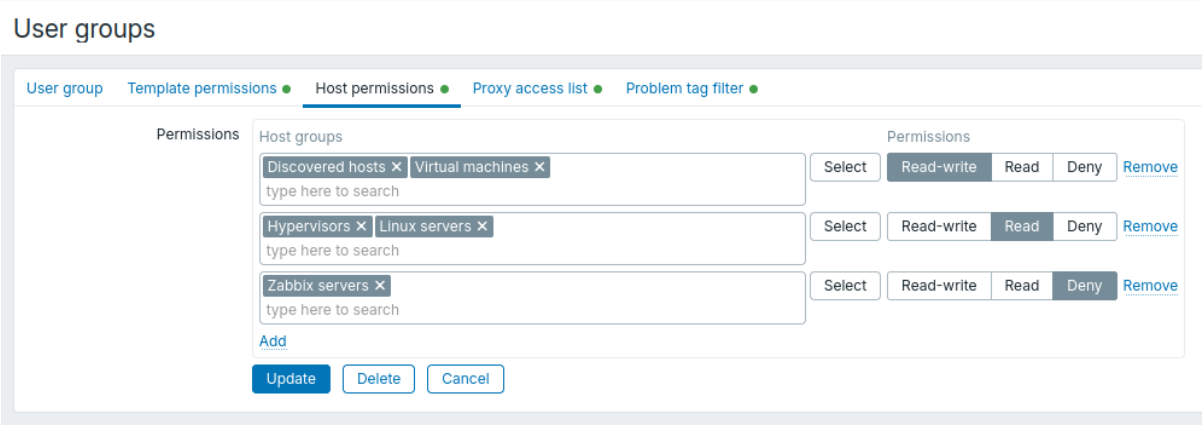
Parameter	Description
Group name	Unikāls grupas nosaukums.
Users	Lai pievienotu lietotājus grupai, sāciet rakstīt esoša lietotāja vārdu. Kad tiek parādīts nolaižamais saraksts ar atbilstošajiem lietotāju vārdiem, ritiniet uz leju, lai veiktu atlasi. Varat arī noklikšķināt uz pogas *Select*, lai uznirstošajā logā atlasītu lietotājus.
Frontend access	Kā tiek autentificēti grupas lietotāji. **System default** - izmantot noklusējuma autentifikācijas metodi (iestatīta globāli) **Internal** - izmantot Zabbix iekšējo autentifikāciju (pat ja globāli tiek izmantota LDAP autentifikācija). Ignorēts, ja HTTP autentifikācija ir globālā noklusējuma metode. **LDAP** - izmantot LDAP autentifikāciju (pat ja globāli tiek izmantota iekšējā autentifikācija). Ignorēts, ja HTTP autentifikācija ir globālā noklusējuma metode. **Disabled** - piekļuve Zabbix lietotāja saskarnei šai grupai ir aizliegta

Parameter	Description
LDAP server	Atlasiet, kuru LDAP serveri izmantot lietotāja autentificēšanai. Šis lauks ir iespējots tikai tad, ja *Frontend access* ir iestatīts uz LDAP vai System default.
Multi-factor authentication	Atlasiet, kuru daudzfaktoru autentifikācijas metodi izmantot lietotāja autentificēšanai: **Default** - izmantot MFA konfigurācijā kā noklusējuma iestatīto metodi; šī opcija pēc noklusējuma ir atlasīta jaunām lietotāju grupām, ja MFA ir iespējots; **<Method name>** - izmantot atlasīto metodi (piemēram, "Zabbix TOTP"); **Disabled** - MFA šai grupai ir atspējots; šī opcija pēc noklusējuma ir atlasīta jaunām lietotāju grupām, ja MFA ir atspējots. Ņemiet vērā: ja lietotājs pieder vairākām lietotāju grupām, kurās MFA ir iespējots (vai vismaz vienā grupā MFA ir iespējots), tiek piemēroti šādi autentifikācijas noteikumi: ja kāda grupa izmanto MFA metodi "Default", tā tiks izmantota lietotāja autentificēšanai; pretējā gadījumā autentificēšanai tiks izmantota pirmā metode (alfabētiskā secībā).
Enabled	Lietotāju grupas un grupas dalībnieku statuss. *Checked* - lietotāju grupa un lietotāji ir iespējoti *Unchecked* - lietotāju grupa un lietotāji ir atspējoti
Debug mode	Atzīmējiet šo izvēles rūtiņu, lai lietotājiem aktivizētu atkļūdošanas režīmu.

Cilnē **Template permissions** varat norādīt lietotāju grupas piekļuvi veidņu grupām un, izmantojot tās, veidņu datiem:



Cilnē **Host permissions** varat norādīt lietotāju grupas piekļuvi hostu grupām un, izmantojot tās, hostu datiem:



Noklikšķiniet uz Add, lai izvēlētos veidņu/hostu grupas (neatkarīgi no tā, vai tā ir vecākgrupa vai ligzdota grupa) un piešķirtu tām atļaujas. Sāciet rakstīt grupas nosaukumu (tiks parādīts nolaižamais saraksts ar atbilstošajām grupām) vai noklikšķiniet uz *Select*, lai atvērtu uznirstošo logu ar visu grupu sarakstu.

Pēc tam izmantojiet opciju pogas, lai izvēlētajām grupām piešķirtu atļaujas. Iespējamās atļaujas ir šādas:

- **Read-write** - lasīšanas un rakstīšanas piekļuve grupai
- **Read** - tikai lasīšanas piekļuve grupai
- **Deny** - piekļuve grupai liegta

Ja viena un tā pati veidņu/hostu grupa ir pievienota vairākās rindās ar atšķirīgām iestatītām atļaujām, tiks piemērota stingrākā atļauja.

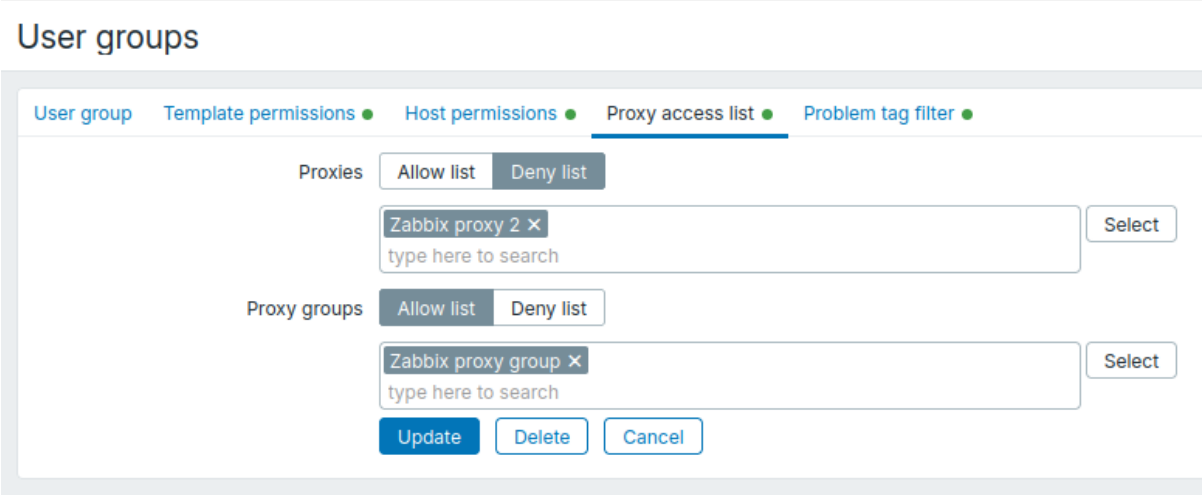
Ņemiet vērā, ka *Super admin* lietotājs var piespiest ligzdotajām grupām izmantot tādu pašu atļauju līmeni kā vecākgrupai; to var izdarīt hostu/veidņu grupas konfigurācijas veidlapā.

Cilnes **Template permissions** un **Host permissions** atbalsta vienādu parametru kopu.

Pašreizējās grupu atļaujas tiek parādītas blokā *Permissions*, un tās var modificēt vai noņemt.

Note:
Ja lietotāju grupai ir **Read-write** atļaujas hostam, bet **Deny** vai nav atļauju veidnei, kas saistīta ar šo hostu, šīs grupas lietotāji nevarēs rediģēt veidnes vienumus hostā, un veidnes nosaukums tiks parādīts kā *Inaccessible template*.

Cilnē **Proxy access list** varat norādīt lietotāju grupas piekļuvi starpniekserveriem — atsevišķi vai izmantojot starpniekserveru grupas:



Izmantojiet lauku *Proxy groups*, lai piemērotu atļauju atlasītajām grupām. Starpniekserveri, kas pieder grupai, var atļaut vai liegt tikai ar šīs grupas starpniecību. Izmantojiet lauku *Proxies*, lai pievienotu atsevišķus starpniekserverus, kas nepieder nevienai starpniekserveru grupai.

Proxies un *Proxy groups* atbalsta vienādu parametru kopu.

Noklikšķiniet uz pogas *Select* blakus laukam *Proxies* vai *Proxy groups*, lai attiecīgi atlasītu starpniekserverus vai starpniekserveru grupas. Varat arī sākt rakstīt starpniekservera vai starpniekserveru grupas nosaukumu attiecīgajā laukā. Tiks parādīts nolaižamais saraksts ar atbilstošajiem starpniekserveriem vai starpniekserveru grupām.

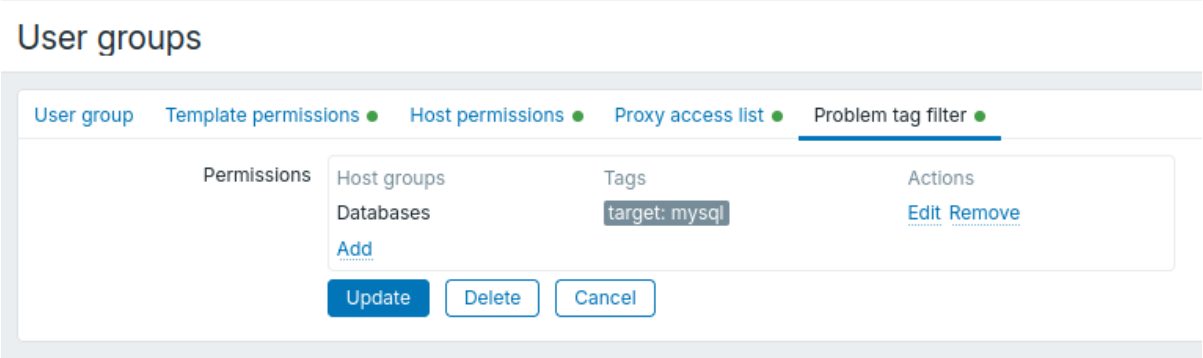
Atlasiet *Allow list*, lai ļautu lietotāju grupai piekļūt tikai atlasītajiem starpniekserveriem un starpniekserveru grupām.
 Ja *Proxies* atlase laukam *Allow list* ir tukša, lietotāju grupai nav piekļuves nevienam starpniekserverim, kas nepieder starpniekserveru grupai. Ja *Proxy groups* atlase laukam *Allow list* ir tukša, lietotāju grupai nav piekļuves nevienai starpniekserveru grupai.

Atlasiet *Deny list*, lai liegtu lietotāju grupai piekļuvi atlasītajiem starpniekserveriem un starpniekserveru grupām. Piekļuve visiem pārējiem starpniekserveriem un starpniekserveru grupām ir atļauta.
 Ja *Proxies* atlase laukam *Deny list* ir tukša, lietotāju grupai ir pilna piekļuve visiem starpniekserveriem, kas nepieder starpniekserveru grupai. Ja *Proxy groups* atlase laukam *Deny list* ir tukša, lietotāju grupai ir pilna piekļuve visām starpniekserveru grupām.

Atļauju prioritātes noteikumi lietotājiem, kuri pieder vairākām lietotāju grupām:

- *Deny list* ignorē *Allow list*. Ja kāda grupa liedz piekļuvi starpniekserverim/starpniekserveru grupai, lietotājs tam nevar piekļūt.
- *Allow list* atļaujas tiek apvienotas. Ja vairākas grupas atļauj piekļuvi, lietotājs var piekļūt visiem starpniekserveriem, kuriem šīs grupas ir atļāvušas piekļuvi.

Cilnē **Problem tag filter** varat iestatīt uz tagiem balstītas atļaujas lietotāju grupām, lai skatītu problēmas, kas filtrētas pēc taga nosaukuma un vērtības:

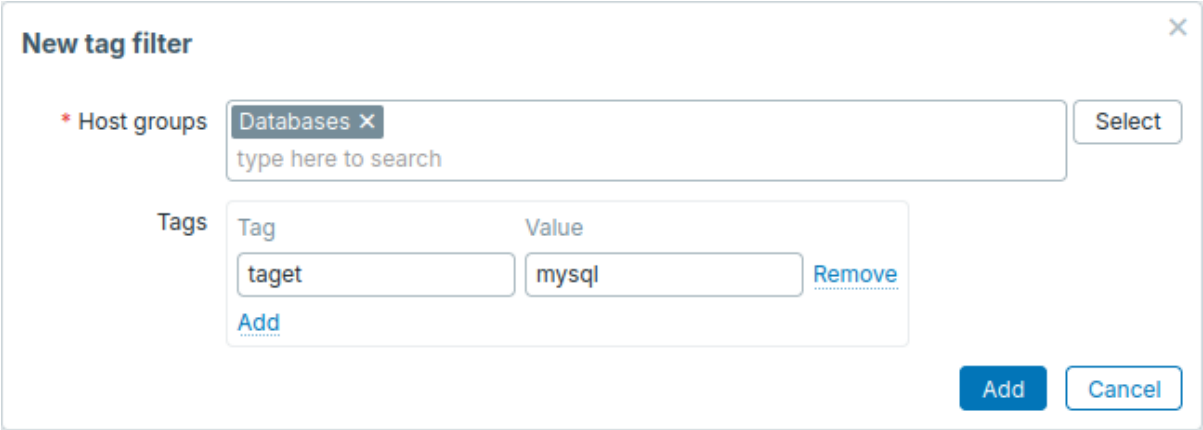


Noklikšķiniet uz Add, lai izvēlētos hostu grupas. Lai atlasītu hostu grupu, kurai piemērot taga filtru, noklikšķiniet uz *Select*, lai iegūtu pilnu esošo hostu grupu sarakstu, vai sāciet rakstīt hostu grupas nosaukumu, lai iegūtu nolaižamo sarakstu ar atbilstošajām grupām. Tiks parādītas tikai hostu grupas, jo problēmas taga filtru nevar piemērot veidņu grupām.

Tagu nosaukumus bez vērtībām var pievienot, bet vērtības bez nosaukumiem nevar. Blokā *Permissions* tiek parādīti tikai pirmie trīs tagi (ja tiem ir vērtības); ja tagu ir vairāk, tos var skatīt, noklikšķinot uz ikonas ••• vai novietojot virs tās kursoru.

Taga filtrs ļauj nošķirt piekļuvi hostu grupai no iespējas skatīt problēmas.

Piemēram, ja datubāzes administratoram jāredz tikai "MySQL" datubāzes problēmas, vispirms jāizveido datubāzes administratoru lietotāju grupa un pēc tam jānorāda taga nosaukums "target" un vērtība "mysql".



Ja ir norādīts taga nosaukums "target", bet vērtības lauks ir atstāts tukšs, lietotāju grupa atlasītajai hostu grupai redzēs visas problēmas ar taga nosaukumu "target".

Pārliecinieties, ka taga nosaukums un taga vērtība ir norādīti pareizi, pretējā gadījumā lietotāju grupa neredzēs nevienu problēmu.

Apskatīsim piemēru, kad lietotājs ir atlasīto vairāku lietotāju grupu dalībnieks. Šādā gadījumā tagiem tiks izmantots nosacījums VAI.

User group A			**User group B**			**Visible result for a user (member) of both groups**
Tag filter						
Host group	*Tag name*	*Tag value*	*Host group*	*Tag name*	*Tag value*	
Databases	target	mysql	Databases	target	oracle	target:mysql or target:oracle problems visible
Not configured in the **Problem tag filter**			Databases	target	oracle	target:oracle problems visible

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**.

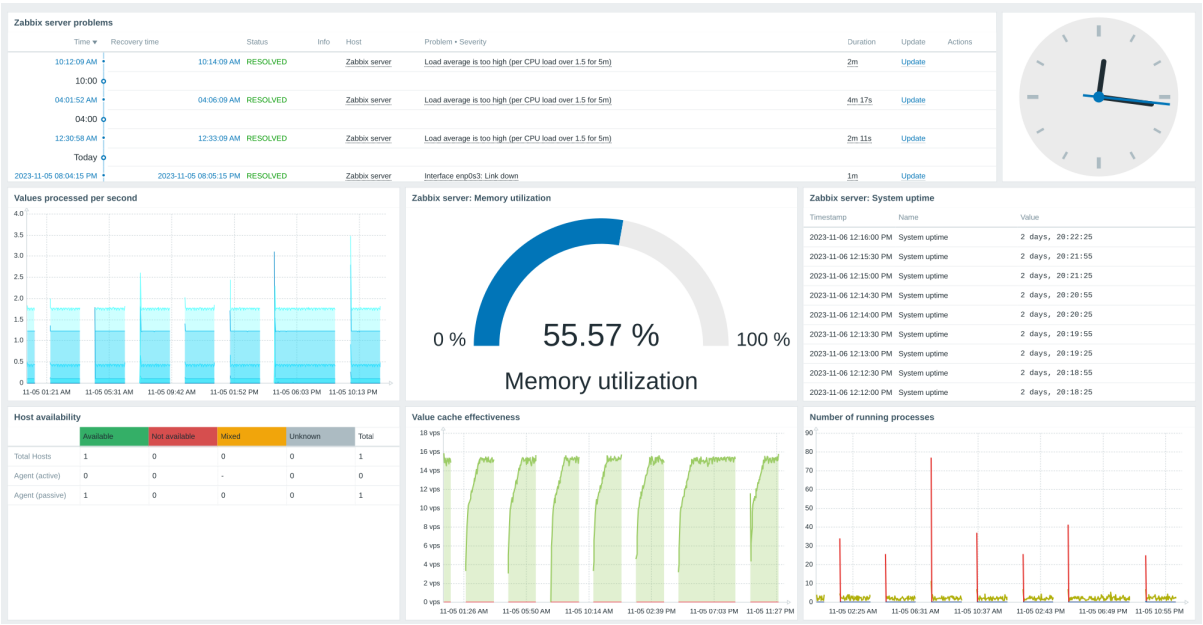
Cita informācija

- 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".

Plānotie pārskati

Pārskats

Izmantojot funkciju *Ieplānotie pārskati*, varat iestatīt, lai noteikta informācijas paneļa PDF versija tiktu nosūtīta norādītajiem adresātiem regulāri.



Priekšnosacījumi:

- Lai iespējotu ieplānotu pārskatu ģenerēšanu, ir jābūt instalētam un pareizi konfigurētam Zabbix tīmekļa servisam. Norādījumus skatiet sadaļā Ieplānotu pārskatu iestatīšana.
- Lietotājam ir jābūt lietotāja lomai ar tipu *Administrators* vai *Superadministrators* un tālāk norādītajām atļaujām:
  - *Ieplānotie pārskati* blokā *Piekļuve lietotāja saskarnes elementiem* (lai skatītu pārskatu iestatījumus)
  - *Ieplānotu pārskatu pārvaldība* blokā *Piekļuve darbībām* (lai izveidotu vai rediģētu pārskatus)

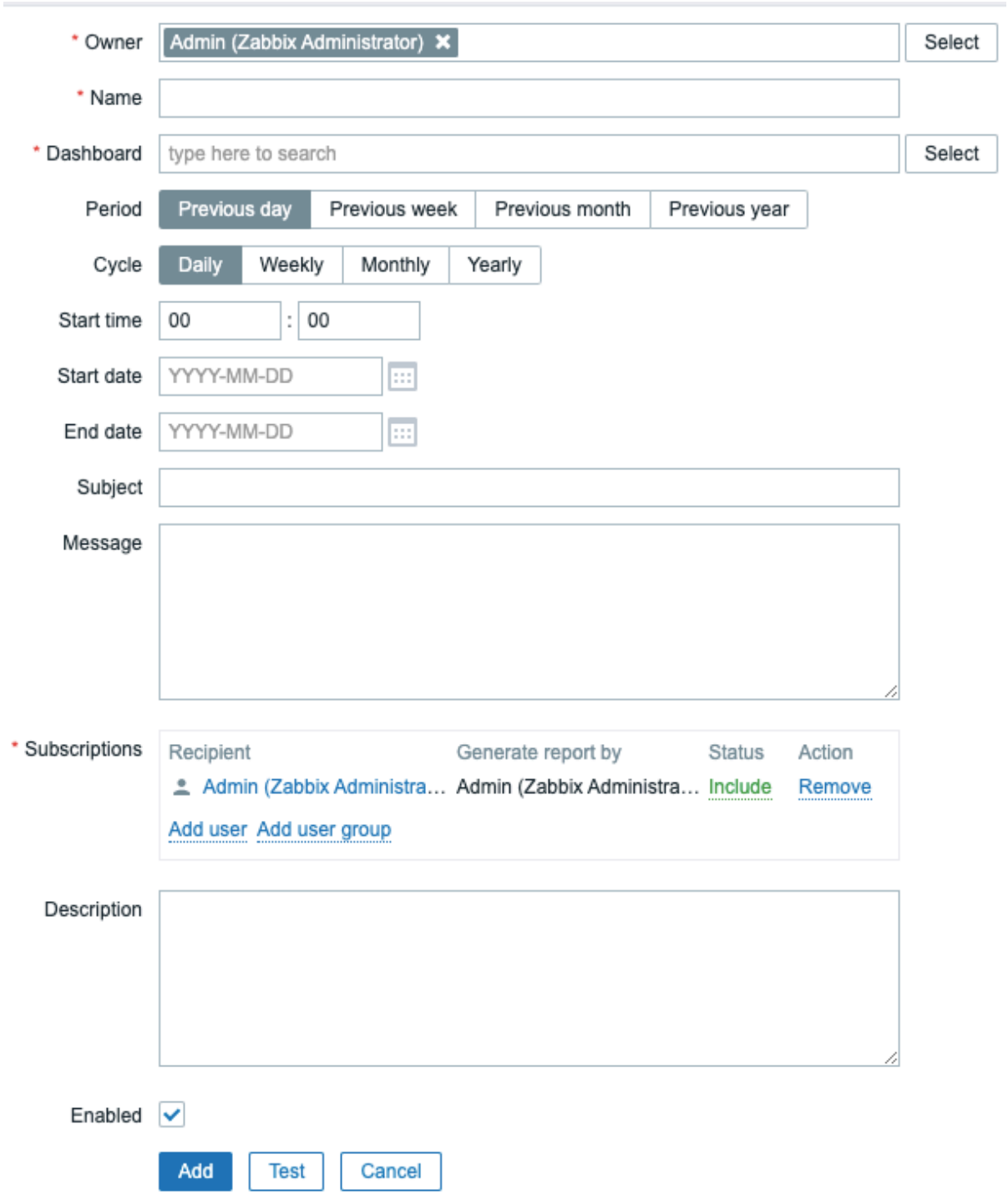
Lai izveidotu ieplānotu pārskatu Zabbix lietotāja saskarnē, rīkojieties, kā norādīts tālāk.

1. Atveriet *Pārskati* > *Ieplānotie pārskati*.
2. Ekrāna augšējā labajā stūrī noklikšķiniet uz *Izveidot pārskatu*.
3. Veidlapā ievadiet pārskata parametrus.

Varat arī izveidot pārskatu, atverot esošu pārskatu, noklikšķinot uz pogas *Klonēt* un saglabājot to ar citu nosaukumu.

Konfigurācija

Cilnē *Ieplānotie pārskati* ir ietverti vispārīgie pārskata atribūti.



Visi obligātie ievades lauki ir atzīmēti ar sarkanu zvaigznīti.

Parametrs	Apraksts
Īpašnieks	Lietotājs, kurš izveido pārskatu. Lietotājiem ar *Super administratora* līmeni ir atļauts mainīt īpašnieku. Lietotājiem ar *Administratora* līmeni šis lauks ir tikai lasāms.
Nosaukums	Pārskata nosaukums; tam jābūt unikālam.

Parametrs	Apraksts
Informācijas panelis	Informācijas panelis, uz kura balstīts pārskats; vienlaikus var atlasīt tikai vienu informācijas paneli. Lai atlasītu informācijas paneli, sāciet rakstīt tā nosaukumu - tiks parādīts atbilstošo informācijas paneļu saraksts; ritiniet uz leju, lai atlasītu vajadzīgo. Varat arī noklikšķināt uz *Atlasīt* blakus laukam un atlasīt informācijas paneli parādītajā sarakstā.
Periods	Laika periods, par kuru tiks sagatavots pārskats. Atlasiet iepriekšējo dienu, nedēļu, mēnesi vai gadu.
Cikls	Pārskata ģenerēšanas biežums. Pārskatus var nosūtīt katru dienu, nedēļu, mēnesi vai gadu. Režīmā "Nedēļas" var atlasīt nedēļas dienas, kurās pārskats tiks nosūtīts.
Sākuma laiks	Dienas laiks formātā hh:mm, kad tiks sagatavots pārskats. Ņemiet vērā, ka tiks izmantota Zabbix servera laika josla.
Atkārtot	Nedēļas dienas, kurās tiks nosūtīts pārskats. Šis lauks ir pieejams tikai tad, ja laukam *Cikls* ir iestatīta vērtība "Nedēļas".
Sākuma datums	Datums, kad jāsāk regulāra pārskatu ģenerēšana.
Beigu datums	Datums, kad jāpārtrauc regulāra pārskatu ģenerēšana.
Temats	Pārskata e-pasta ziņojuma temats. Atbalstītās makrokomandas: {TIME}, {TIMESTAMP}.
Ziņojums	Pārskata e-pasta ziņojuma pamatteksts. Atbalstītās makrokomandas: {TIME}, {TIMESTAMP}.
Abonementi	Pārskata saņēmēju saraksts. Pēc noklusējuma tajā ir iekļauts tikai pārskata īpašnieks. Par pārskata saņēmēju var norādīt jebkuru Zabbix lietotāju, kuram ir konfigurēts e-pasta multivides tips. Noklikšķiniet uz *Pievienot lietotāju* vai *Pievienot lietotāju grupu*, lai pievienotu citus saņēmējus. Noklikšķiniet uz lietotājvārda, lai rediģētu iestatījumus: *Ģenerēt pārskatu, pamatojoties uz* - vai pārskata dati jāģenerē, pamatojoties uz pašreizējā lietotāja vai saņēmēja informācijas paneļa atļaujām. *Statuss* - atlasiet "Iekļaut", lai nosūtītu pārskatu lietotājam, vai "Izslēgt", lai nepieļautu 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, kuriem nav pietiekamu atļauju (tas ir, lietotāji ar lomu, kas balstīta uz *Administratora* lietotāja tipu un kuri nav tās pašas lietotāju grupas dalībnieki, kurā ir saņēmējs vai pārskata īpašnieks), laukos *Saņēmējs* un *Ģenerēt pārskatu, pamatojoties uz* faktiskā nosaukuma vietā redzēs "Nepieejams lietotājs" vai "Nepieejama lietotāju grupa"; lauki *Statuss* un *Darbība* tiks parādīti kā tikai lasāmi.
Iespējots	Pārskata statuss. Notīrot šo izvēles rūtiņu, 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 pārskatu. Šī poga ir pieejama tikai jauniem pārskatiem.
Update	Atjaunināt pārskata rekvizītus.
Clone	Izveidot citu pārskatu, pamatojoties uz pašreizējā pārskata rekvizītiem.
Test	Pārbaudīt, vai pārskata konfigurācija ir pareiza, nosūtot pārskatu pašreizējam lietotājam.
Delete	Dzēst pārskatu.
Cancel	Atcelt pārskata rekvizītu rediģēšanu.

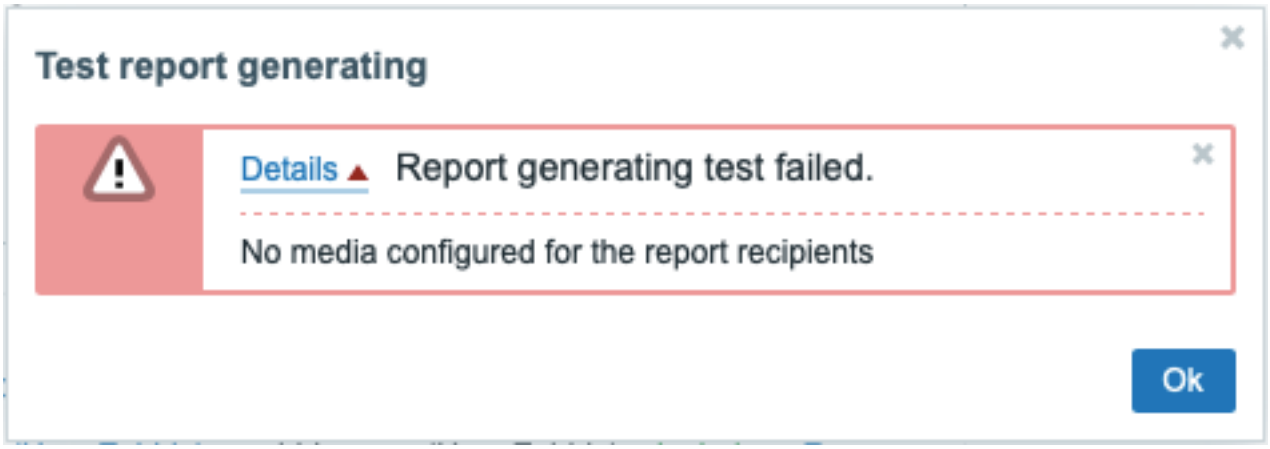
Testēšana

Lai pārbaudītu pārskatu, pārskata konfigurācijas veidlapas apakšdaļā noklikšķiniet uz pogas *Testēt*.

Note:
Poga *Testēt* nav pieejama, ja pārskata konfigurācijas veidlapa ir atvērta no informācijas paneļa darbību izvēlnes.

Ja konfigurācija ir pareiza, testa pārskats tiek nekavējoties nosūtīts pašreizējam lietotājam. Testa pārskatiem abonenti un lietotāja iestatījums *Ģenerēt pārskatu pēc* tiek ignorēti.

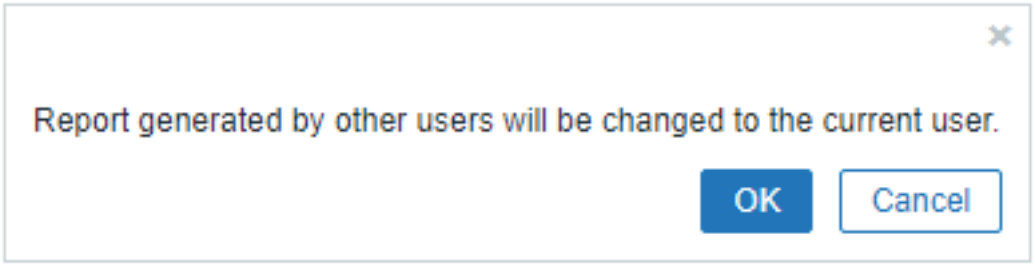
Ja konfigurācija ir nepareiza, tiek parādīts kļūdas ziņojums, kurā aprakstīts iespējamais iemesls.



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*.

Datu eksports

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)

Eksportēt uz failiem

Pārskats

Ir iespējams konfigurēt trigeru notikumu, vienumu vērtību un tendenču reāllaika eksportēšanu newline-delimited JSON formātā.

Eksportēšana tiek veikta failos, kur katra eksporta faila rinda ir JSON objekts. Vērtību kartējumi netiek piemēroti.

Kļūdu gadījumā (datus nevar ierakstīt eksporta failā, eksporta failu nevar pārdēvēt vai pēc 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 fails ir lasāms īpašnieka grupai. Visas pārējās atļaujas ir liegtas.

Ņemiet vērā, ka host/item var nebūt metadatu (hostu grupas, hosta nosaukuma, vienuma nosaukuma), 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 eksportēšanas failu direktoriju - skatiet ExportDir parametru servera konfigurācijā.

Ir pieejami vēl divi parametri:

- `ExportFileSize` var izmantot, lai iestatītu viena eksportēšanas faila maksimālo atļauto 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 vienam eksportēšanas failam ir 1G, lielu eksportēšanas failu saglabāšana var ātri aizpildīt diska vietu.

- `ExportType` ļauj norādīt, kuri entītiju tipi (notikumi, vēsture, tendences) tiks eksportēti.

Jaunas rindas atdalīts JSON eksporta protokols

Šajā sadaļā ir sniegta informācija par eksporta protokolu newline-delimited JSON formātā, ko izmanto:

- datu eksportēšanai uz failiem
- straumēšanai uz ārējām sistēmām

Visiem failiem ir `.ndjson` paplašinājums. Katra eksporta faila rinda ir JSON objekts.

Trigeru notikuma eksports

Tālāk norādītā informācija tiek eksportēta problēmas notikumam:

Field		Type	Description
clock		number	Sekunžu skaits kopš Epoch līdz brīdim, kad problēma tika konstatēta (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 - Katastrofāls).
hosts		array	To hosts saraksts, kas iesaistīti trigeris izteiksmē; masīvā jābūt vismaz vienam elementam.
-		object	
	host	string	Hosta nosaukums.
	name	string	Redzamais hosta nosaukums.
groups		array	Visu to hosts grupu saraksts, kas iesaistīti trigeris izteiksmē; masīvā jābūt vismaz vienam elementam.
-		string	Hostu grupas nosaukums.

Field	Type	Description
tags	array	Problēmas tagu saraksts (var būt tukšs).
-	object	
tag	string	Taga nosaukums.
value	string	Taga vērtība (var būt tukša).

Tālāk norādītā informācija tiek eksportēta atjaunošanās notikumam:

Field	Type	Description
clock	number	Sekunžu skaits kopš Epoch līdz brīdim, kad problēma tika atrisināta (veselā daļa).
ns	number	Nanosekunžu skaits, kas jāpieskaita `clock`, lai iegūtu precīzu problēmas atrisināšanas laiku.
value	number	0 (vienmēr).
eventid	number	Atjaunošanās notikuma ID.
p_eventid	number	Problēmas notikuma ID.

Trigeru notikumu eksportēšanas piemēri

Problēma:

```
{"clock":1519304285,"ns":123456789,"value":1,"name":"Either Zabbix aģents is unreachable on Host B or poll
```

Atjaunošana:

```
{"clock":1519304345,"ns":987654321,"value":0,"eventid":43,"p_eventid":42}
```

Problēma (vairāku problēmu notikumu ģenerēšana):

```
{"clock":1519304286,"ns":123456789,"value":1,"eventid":43,"name":"Either Zabbix aģents is unreachable on H
```

```
{"clock":1519304286,"ns":123456789,"value":1,"eventid":43,"name":"Either Zabbix aģents is unreachable on H
```

Atjaunošana:

```
{"clock":1519304346,"ns":987654321,"value":0,"eventid":44,"p_eventid":43}
```

```
{"clock":1519304346,"ns":987654321,"value":0,"eventid":44,"p_eventid":42}
```

Vienuma vērtības eksports

Tālāk norādītā informācija tiek eksportēta savāktai vienuma vērtībai:

Field	Type	Description
host	object	Vienuma 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 tagu saraksts (var būt tukšs).
-	object	
tag	string	Taga nosaukums.
value	string	Taga vērtība (var būt tukša).
itemid	number	Vienuma ID.
name	string	Redzamais vienuma nosaukums.
clock	number	Sekunžu skaits kopš Epoch brīža 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.

Field	Type	Description
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 - skaitliskais peldošais punkts, 1 - rakstzīme, 2 - žurnāls, 3 - skaitliskais bez zīmes, 4 - teksts, 5 - binārs, 6 - JSON

Vienuma vērtības eksportēšanas piemērs

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":[{"
```

Trendu eksports

Tālāk norādītā informācija tiek eksportēta aprēķinātai trendu vērtībai:

Field	Type	Description
host	object	Vienuma 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 tagu saraksts (var būt tukšs).
-	object	
tag	string	Taga nosaukums.
value	string	Taga vērtība (var būt tukša).
itemid	number	Vienuma ID.
name	string	Redzamais vienuma nosaukums.
clock	number	Sekunžu skaits kopš Unix laikmeta brīža, kad vērtība tika savākta (veselā daļa).
count	number	Savākto vērtību skaits noteiktā stundā.
min	number	Minimālā vienuma vērtība noteiktā stundā.
avg	number	Vidējā vienuma vērtība noteiktā stundā.
max	number	Maksimālā vienuma vērtība noteiktā stundā.
type	number	Vērtības tips: 0 - skaitlis ar peldošo komatu, 3 - bez zīmes skaitlis

Trendu eksporta 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":[{"

Straumēšana uz ārējām sistēmām

Pārskats

Ir iespējams straumēt vienumu vērtības un notikumus no Zabbix uz ārējām sistēmām, izmantojot HTTP (skatiet protokola informāciju).

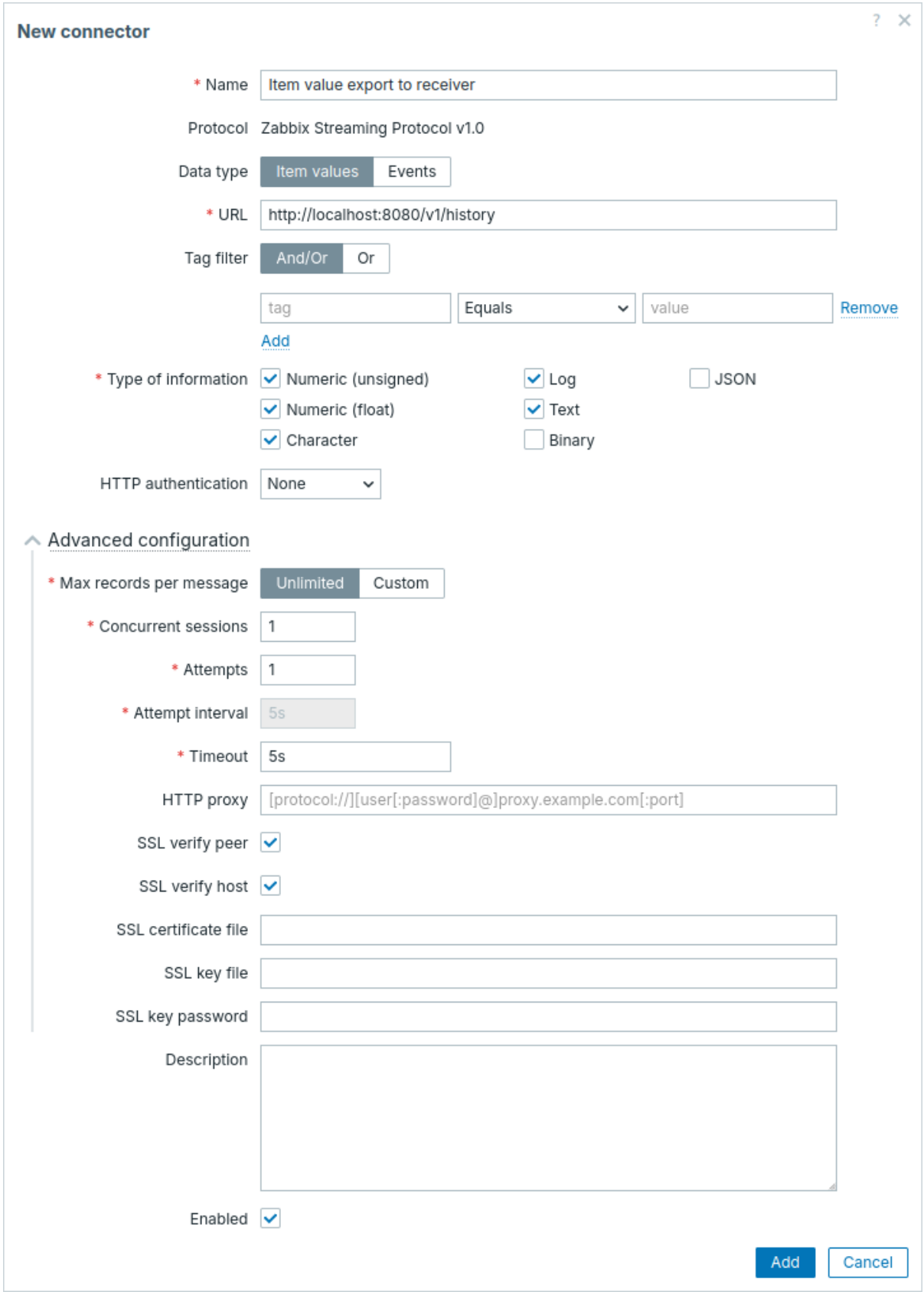
Tagu filtrs ļauj straumēt vienumu vērtību vai notikumu apakškopas.

Par datu straumēšanu ir atbildīgi divi Zabbix servera procesu tipi: `connector manager` un `connector worker`. Zabbix iekšējais vienums `zabbix[connector_queue]` ļauj pārraudzīt savienotāja rindā ievietoto vērtību skaitu.

Konfigurēšana

Lai konfigurētu datu straumēšanu uz ārēju sistēmu, jāveic tālāk norādītās darbības:

1. Iestatiet attālo sistēmu datu saņemšanai no Zabbix. Šim nolūkam ir pieejami šādi rīki:
   - Vienkārša saņēmēja piemērs, kas saņemto informāciju reģistrē failos `events.ndjson` un `history.ndjson`.
   - Kafka connector for Zabbix server - viegls Go valodā rakstīts serveris, kas paredzēts vienumu vērtību un notikumu pārsūtīšanai no Zabbix servera uz Kafka brokeri.
2. Iestatiet nepieciešamo savienotāju workeru skaitu Zabbix, pielāgojot `StartConnectors` parametru failā `zabbix_server.conf`. Savienotāju workeru skaitam jāatbilst (vai jābūt lielākam, ja vienlaicīgo sesiju skaits ir lielāks par 1) Zabbix lietotāja saskarnē konfigurētajam savienotāju skaitam. Pēc tam restartējiet Zabbix serveri.
3. Konfigurējiet jaunu savienotāju Zabbix lietotāja saskarnē (*Administration* > *General* > *Connectors*) un atkārtoti ielādējiet servera kešatmiņu, izmantojot 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 saņēmēja URL. Tiek atbalstīti lietotāja makro.

Parameter	Description
Tag filter	Eksportējiet tikai vienumu vērtības vai notikumus, kas atbilst tagu filtram. Ja filtrs nav iestatīts, tiek eksportēts viss. Var iekļaut vai izslēgt konkrētus tagus un tagu vērtības. Var iestatīt vairākus nosacījumus. Tagu nosaukumu 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, nav reģistrjutīga); **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, nav reģistrjutīga). Nosacījumiem ir pieejami divi aprēķina tipi: **And/Or** - jāizpilda visi nosacījumi; nosacījumi ar vienādu taga nosaukumu tiks grupēti, izmantojot nosacījumu Or; **Or** - pietiek, ja tiek izpildīts viens nosacījums.
Type of information	Atlasiet informācijas tipu (numeric (unsigned), numeric (float), character u. c.), pēc kura filtrēt vienumu vērtības, kas 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: Kerberos konfigurēšana ar Zabbix); **Digest** - tiek izmantota Digest autentifikācija; **Bearer** - tiek izmantota Bearer autentifikācija.
Username	Ievadiet lietotājvārdu (līdz 255 rakstzīmēm). Tiek atbalstīti lietotāja makro. 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 makro. Šis lauks ir pieejams, ja *HTTP authentication* ir iestatīts uz ”Basic”, ”NTLM”, ”Kerberos” vai ”Digest”.
Bearer token	Ievadiet Bearer tokenu. Tiek atbalstīti lietotāja makro. Šis lauks ir pieejams un obligāts, ja *HTTP authentication* ir iestatīts uz ”Bearer”.
Advanced configuration	Noklikšķiniet uz galvenes *Advanced configuration*, lai parādītu papildu konfigurācijas opcijas (skatiet tālāk).
Max records per message	Norādiet maksimālo vērtību vai notikumu skaitu, ko var straumēt vienā ziņojumā.
Concurrent sessions	Atlasiet šim savienotājam palaižamo sūtītāja procesu skaitu. 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 10 s; 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 HTTP atbildes kods nav 200, 201, 202, 203 vai 204. Atkārtoti mēģinājumi tiek **aktivizēti** sakaru kļūdu gadījumā vai tad, ja HTTP atbildes kods nav 200, 201, 202, 203, 204, 400, 401, 403, 404, 405, 415 vai 422. Novirzīšana tiek izpildīta, tāpēc 302 -> 200 ir pozitīva atbilde, savukārt 302 -> 503 aktivizēs atkārtotu mēģinājumu.
Timeout	Norādiet ziņojuma taimautu (1-60 sekundes, noklusējuma vērtība - 5 sekundes). Tiek atbalstīti laika sufiksi (piemēram, 30s, 1m). Tiek atbalstīti lietotāja makro.

Parameter	Description
HTTP proxy	Varat norādīt izmantojamo HTTP starpniekserveri šādā formātā: `[protocol://][username[:password]@]proxy.example.com[:port]` Tiek atbalstīti lietotāja makro. 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 ports 1080. Ja *HTTP proxy* ir norādīts, starpniekserveris pārrakstīs ar starpniekserveri saistītos vides mainīgos, piemēram, `http_proxy`, `HTTPS_PROXY`. Ja tas nav norādīts, starpniekserveris ar starpniekserveri saistītos vides mainīgos nepārrakstīs. Ievadītā vērtība tiek nodota tālāk bez izmaiņām, un tās pareizība netiek pārbaudīta. Varat ievadīt arī SOCKS starpniekservera adresi. Ja norādīsiet nepareizu protokolu, savienotājs nespēs straumēt vienumu vērtības vai notikumus no Zabbix. Ņemiet vērā, ka ar HTTP starpniekserveri tiek atbalstīta tikai vienkārša autentifikācija.
SSL verify peer	Atzīmējiet izvēles rūtiņu, lai pārbaudītu tīmekļa servera SSL sertifikātu. Servera sertifikāts tiek pārbaudīts, izmantojot sistēmas mēroga CA atrašanās vietā esošos uzticamās sertificēšanas iestādes (CA) sertifikātus. CA failu atrašanās vietu varat mainīt, izmantojot Zabbix servera vai starpniekservera konfigurācijas parametru `SSLCALocation`.
SSL verify host	Atzīmējiet izvēles rūtiņu, lai pārbaudītu, vai tīmekļa servera sertifikāta lauks *Common Name* vai *Subject Alternate Name* atbilst. Tādējādi tiek iestatīta cURL opcija `CURLOPT_SSL_VERIFYHOST`.
SSL certificate file	SSL sertifikāta faila nosaukums, kas tiek izmantots klienta autentifikācijai. Sertifikāta failam jābūt PEM[1] formātā. Tiek atbalstīti lietotāja makro. Ja sertifikāta failā ir arī privātā atslēga, atstājiet lauku *SSL key file* tukšu. Ja atslēga ir šifrēta, laukā *SSL key password* norādiet paroli. Direktoriju, kurā atrodas šis fails, norāda Zabbix servera vai starpniekservera konfigurācijas parametrs `SSLCertLocation`.
SSL key file	SSL privātās atslēgas faila nosaukums, kas tiek izmantots klienta autentifikācijai. Privātās atslēgas failam jābūt PEM[1] formātā. Tiek atbalstīti lietotāja makro. Direktoriju, kurā atrodas šis fails, norāda Zabbix servera vai starpniekservera konfigurācijas parametrs `SSLKeyLocation`.
SSL key password	SSL privātās atslēgas faila parole. Tiek atbalstīti lietotāja makro.
Description	Ievadiet savienotāja aprakstu.
Enabled	Atzīmējiet izvēles rūtiņu, lai iespējotu savienotāju.

Ja Kafka savienotājs ir konfigurēts ar komatiem atdalītu bootstrap brokeru adrešu sarakstu (piemēram, `Kafka.URL=kafka1.example.com:9` Kafka klients izveido savienojumu ar brokeri(-iem), kas atbild pirmais, un izmanto to klastera metadatus. Ja sarakstā ir dažādiem Kafka klasteriem piederošas adreses, tiks izmantots tikai visātrāk atbildējušais klasteris, bet pārējās adreses tiks reģistrētas kā nepieejamas; tādēļ var tikt parādīti tālāk norādītie startēšanas brīdinājumi, pat ja savienotājs ir savienots:

```
kafka cluster connected, but broker(s) "kafka1.example.com:9093, kafka2.example.com:9093" unavailable; wil
```

Dažās vidēs (privātajos tīklos, konteineru tīklos vai nestandarta DNS/hosts konfigurācijās) resursdatoru nosaukumi vai IP adreses var tikt atrisinātas uz cilpas atgriezeniskās saites adresēm (piemēram, `127.0.0.1/localhost`) vai arī klients tās var normalizēt, tādējādi šādi brīdinājumi var būt maldinoši. Lai mazinātu neskaidrības, pārliecinieties, ka visas `Kafka.URL` adreses pieder vienam Kafka klasterim, pārbaudiet DNS izšķiršanu no savienotāja resursdatora un brokeru `advertised.listeners`, kā arī dodiet priekšroku adresēm, kas tiek atrisinātas uz brokera reklamēto adresi.

Protokols

Saziņa starp serveri un uztvērēju notiek, izmantojot HTTP, REST API, NDJSON un "Content-Type: application/x-ndjson".

Plašāku informāciju skatiet Newline-delimited JSON export protocol.

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, bet citam kodam — neveiksmīgiem pieprasījumiem.

Veiksmīgas atbildes piemērs:

```
HTTP/1.1 200 OK
Content-Type: application/json
X-Content-Type-Options: nosniff
Date: Tue, 21 Apr 2026 10:13:04 GMT
Content-Length: 23

{"response":"success"}
```

Piemērs ar kļūdām:

```
HTTP/1.1 422 Unprocessable Entity
Content-Type: application/json
X-Content-Type-Options: nosniff
Date: Tue, 21 Apr 2026 12:15:01 GMT
Content-Length: 55

{"error":"invalid character '{' after top-level value"}
```

SNMP gateway

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`)

Konfigurācijas eksportēšana/importēšana

Pārskats

Zabbix eksportēšanas/importēšanas funkcionalitāte ļauj apmainīties ar dažādiem konfigurācijas objektiem 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 Zabbix Community templates. Pēc tam citi var lejupielādēt veidni un importēt failu Zabbix.
- integrēties ar trešo pušu rīkiem - universālie YAML, XML un JSON formāti padara iespējamu integrāciju un datu importēšanu/eksportēšanu ar trešo pušu rīkiem un lietojumprogrammām

Ko var eksportēt/importēt

Objekti, kurus var eksportēt/importēt, ir:

- Host grupas (*tikai caur Zabbix API*)
- Veidņu grupas (*tikai caur Zabbix API*)
- Veidnes
- Hosts
- Tīkla kartes
- Multivides tipi
- Attēli
- Informācijas paneļi

- Regulārās izteiksmes

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.
- Hosts un veidnes entītijas (vienumi, trigeri, grafiki, atklāšanas noteikumi), kas ir mantotas no saistītajām veidnēm, netiek eksportētas. Jebkādas izmaiņas, kas veiktas šīm entītijām hosta līmenī (piemēram, mainīts vienuma intervāls, modificēta regulārā izteiksme vai pievienoti prototipi zemā līmeņa atklāšanas noteikumam), eksportēšanas laikā tiks zaudētas; importēšanas laikā visas entītijas no saistītajām veidnēm tiek atkārtoti izveidotas tāpat kā sākotnējā saistītajā veidnē.
- Entītijas, kas izveidotas ar zemā līmeņa atklāšanu, un jebkuras entītijas, kas no tām ir atkarīgas, netiek eksportētas. Piemēram, trigeris, kas izveidots vienumam, ko ģenerējis LLD noteikums, 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 apstājas pie pirmās kļūdas.
- Atjauninot esošos attēlus attēlu importēšanas laikā, lauks "imagetype" tiek ignorēts, t. i., importēšanas laikā nav iespējams mainīt attēla tipu.
- Importējot hosts/veidnes, izmantojot opciju "Delete missing", hosta/veidnes makro, kas nav iekļauti importēšanas failā, pēc importēšanas tiks dzēsti no hosta/veidnes.
- Tukši tagi vienumiem, trigeriem, grafikiem, discoveryRules, itemPrototypes, triggerPrototypes, graphPrototypes nav nozīmīgi, t. i., tas ir tas pats, kas tad, ja tie būtu izlaisti.
- Ja importētā hosta/veidnes entītijām ir konfigurēti savi noildzes laiki, tiks lietoti tie; pretējā gadījumā tiks lietoti starpniekservera/globalie noildzes laiki.
- Importēšana atbalsta YAML, XML un JSON, importēšanas failam ir jābūt ar pareizu faila paplašinājumu: .yaml un .yml YAML formātam, .xml XML formātam un .json JSON formātam. Skatiet saderības informāciju par atbalstītajām XML versijām.
- Importēšana atbalsta konfigurācijas failus tikai UTF-8 kodējumā (ar BOM vai bez tā); citi kodējumi (UTF16LE, UTF16BE, UTF32LE, UTF32BE u. c.) izraisīs importēšanas konvertēšanas kļūdu.

YAML pamata formāts

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 galvene XML dokumentiem
- Saknes tags Zabbix XML eksportam
- Eksporta versija

```
<?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:

- Saknes objekts Zabbix JSON eksportam
- Eksporta versija

```
{
    "zabbix_export": {
        "version": "8.0"
    }
}
```

Citi objekti ir atkarīgi no eksportētajiem objektiem.

Veidņu grupas

Pārskats

Lietotāja saskarnē veidņu grupas var tikt eksportētas tikai kopā ar veidnes eksportu. Kad veidne tiek eksportēta, visas grupas, kurām tā pieder, tiek eksportētas automātiski 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.

Hostu grupas

Pārskats

Lietotāja saskarnē hostu grupas var eksportēt tikai kopā ar hosta eksportu. Kad hosts tiek eksportēts, visas grupas, kurām tas pieder, tiek eksportētas kopā ar to automātiski.

API ļauj eksportēt hostu grupas neatkarīgi no hostiem.

Eksporta formāts

```
host_groups:
  - uuid: 6f6799aa69e844b4b3918f779f2abf08
    name: 'Zabbix serveri'
```

Eksportētie elementi

Element	Type	Description
uuid	string	Unikāls identifikators šai hostu grupai.
name	string	Grupas nosaukums.

Veidnes

Pārskats

Veidnes tiek eksportētas kopā ar daudziem saistītajiem objektiem un objektu attiecībām.

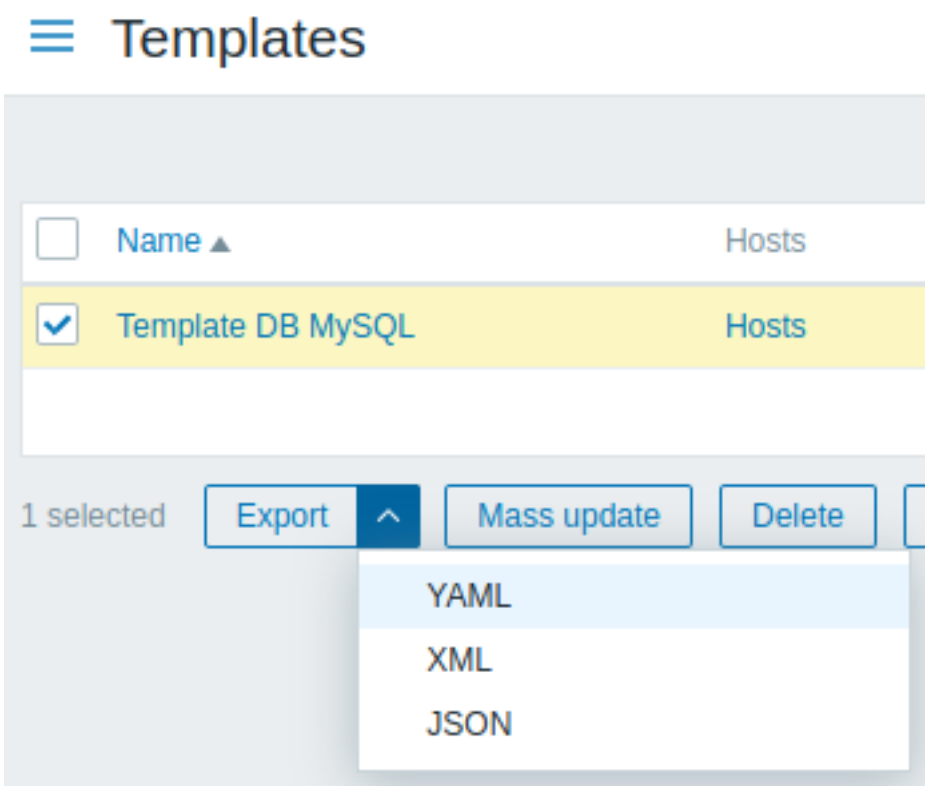
Veidnes eksportā ir iekļauts:

- Saistītās veidņu grupas
- Saistītās hostu grupas (ja tās tiek izmantotas host prototipa konfigurācijā)
- Veidnes dati
- Saite uz citām veidnēm
- Saite uz veidņu grupām
- Tieši saistītie vienumi
- Tieši saistītie trigeri
- Tieši saistītās diagrammas
- Tieši saistītie paneļi
- Tieši saistītie atklāšanas noteikumi ar visiem prototipiem
- Tieši saistītie tīmekļa scenāriji
- Vērtību kartes

Eksportēšana

Lai eksportētu veidnes, rīkojieties šādi:

1. Dodieties uz *Datu vākšana* > *Veidnes*.
2. Atzīmējiet eksportējamo veidņu izvēles rūtiņas.
3. Noklikšķiniet uz *Eksportēt* zem saraksta, lai eksportētu noklusējuma YAML formātā, vai noklikšķiniet uz bultiņas, lai atlasītu citu eksporta formātu.



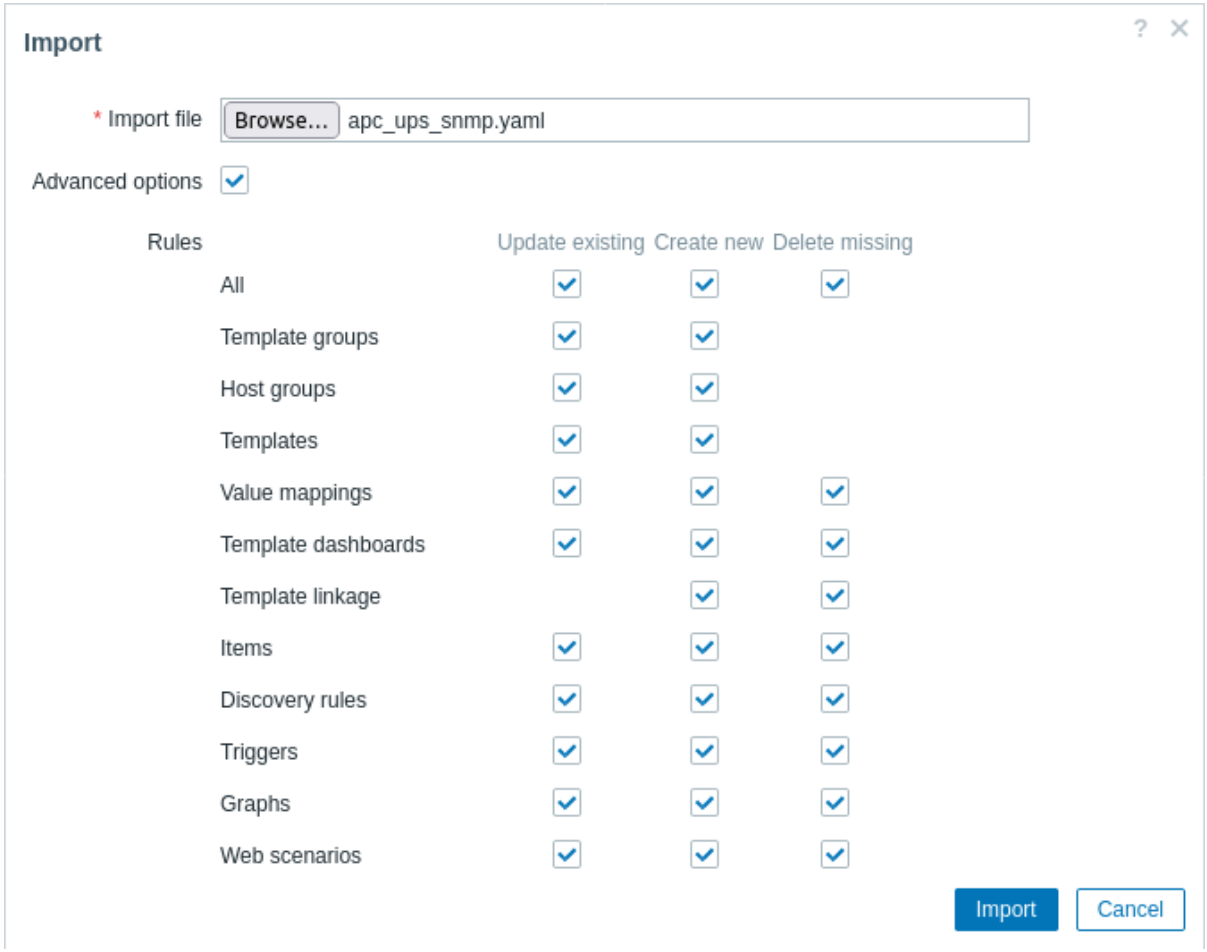
Atkarībā no atlasītā formāta veidnes tiek eksportētas uz lokālu failu ar noklusējuma nosaukumu:

- `zabbix_export_templates.yaml` - YAML eksportā;
- `zabbix_export_templates.xml` - XML eksportā;
- `zabbix_export_templates.json` - JSON eksportā.

Importēšana

Lai importētu veidnes, rīkojieties, kā norādīts tālāk:

1. Atveriet *Data collection* > *Templates*.
2. Augšējā labajā stūrī noklikšķiniet uz *Import*.
3. Atlasiet importēšanas failu.
4. Atzīmējiet to importēšanas noteikumu izvēles rūtiņas, kurus vēlaties lietot (skatiet tālāk).
5. Konfigurācijas formas apakšējā labajā stūrī noklikšķiniet uz *Import*.



Ja atzīmēsiet izvēles rūtiņu *Advanced options*, tiks parādīts detalizēts visu importējamo elementu saraksts — atzīmējiet vai noņemiet atzīmi katram importēšanas noteikumam atbilstoši nepieciešamībai.

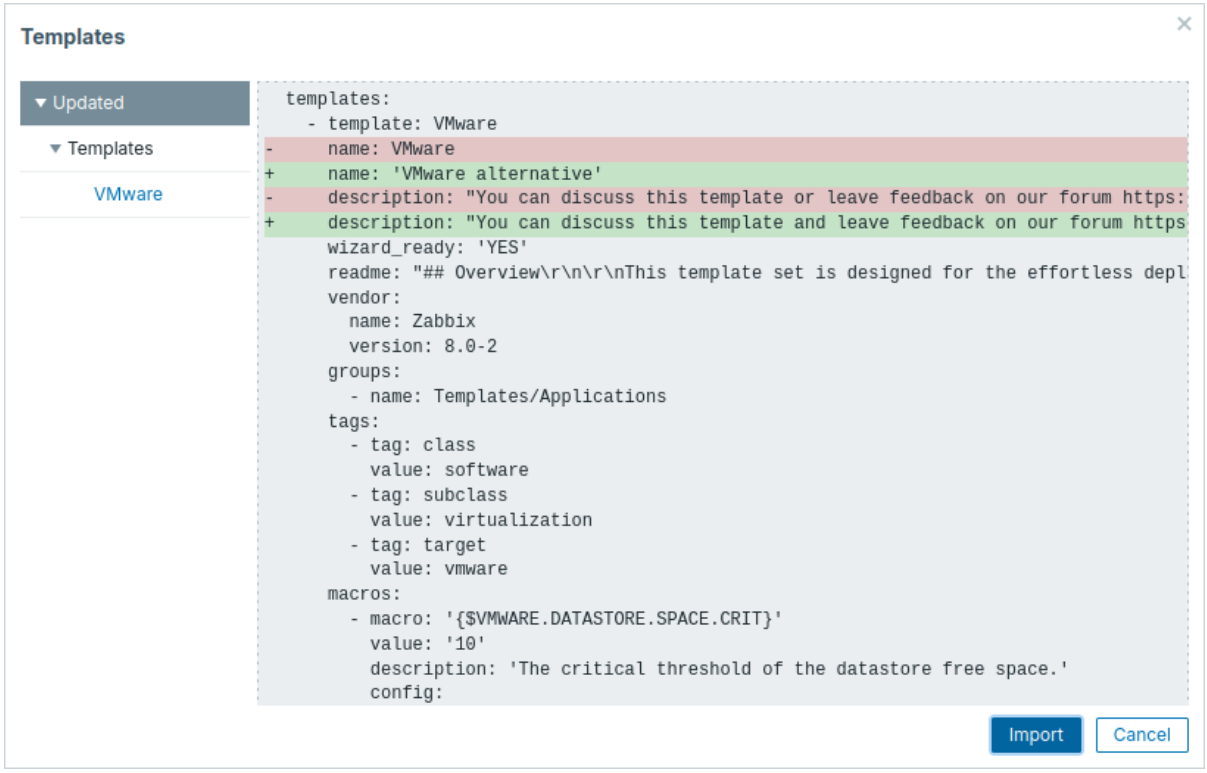
Ja noklikšķināsiet uz izvēles rūtiņas rindā *All*, visi zem tās esošie elementi tiks atzīmēti vai arī tiem tiks noņemta atzīme.

Importēšanas noteikumi:

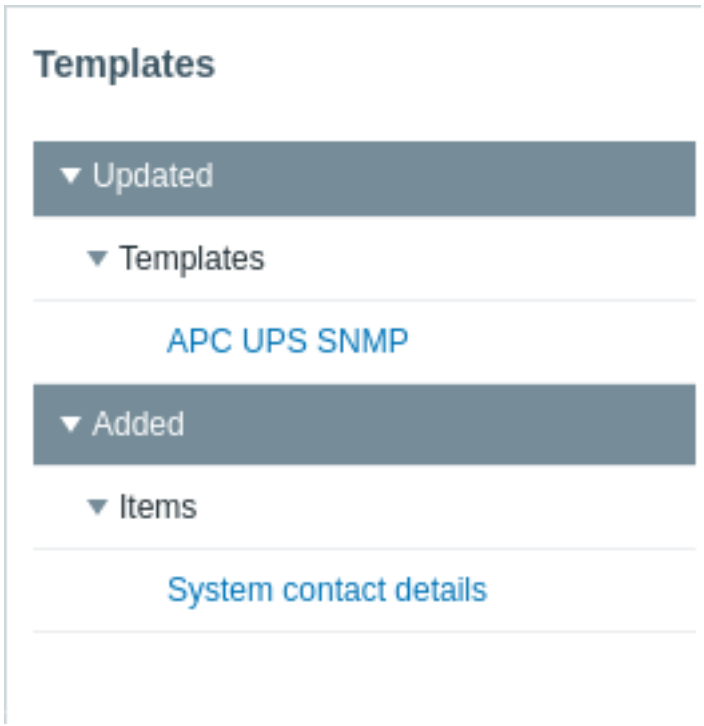
Noteikums	Apraksts
Update existing	Esošie elementi tiks atjaunināti, izmantojot importēšanas faila datus. Pretējā gadījumā tie netiks atjaunināti.
Create new	Jauni elementi tiks izveidoti, izmantojot importēšanas faila datus. Pretējā gadījumā tie netiks izveidoti.
Delete missing	Esošie elementi, kas nav ietverti importēšanas failā, tiks noņemti. Pretējā gadījumā tie netiks noņemti. Ja vienumam *Template linkage* ir atzīmēts *Delete missing*, pašreizējā veidnes saite, kas nav ietverta importēšanas failā, tiks atsaistīta. No atsaistītajām veidnēm mantotās entītijas (vienumi, trigeri, grafiki u. c.) netiks noņemtas (ja vien katrai entītijai nav atlasīta arī opcija *Delete missing*).

Attention:
Ja veidnes ar tādiem pašiem nosaukumiem jau pastāv, importēšanas laikā jāatzīmē opcijas *Delete missing*, lai nodrošinātu tīru importēšanu. Tādējādi tiks noņemti vecie vienumi, kas vairs nav ietverti atjauninātajā veidnē (ņemiet vērā, ka tādējādi tiks zaudēta šo veco vienumu vēsture).

Nākamajā ekrānā varēsiet skatīt importējamās veidnes saturu. Ja tā ir jauna veidne, visi elementi tiks parādīti zaļā krāsā. Ja tiek atjaunināta esoša veidne, jaunie veidnes elementi tiks izcelti zaļā krāsā, noņemtie veidnes elementi — sarkanā krāsā, bet nemainītie elementi tiks parādīti uz pelēka fona.



Kreisajā pusē esošo izvēlni var izmantot, lai pārvietotos pa izmaiņu sarakstu. Sadaļā *Updated* tiek izceltas visas esošo veidnes elementu izmaiņas. Sadaļā *Added* ir uzskaitīti jaunie veidnes elementi. Katrā sadaļā elementi ir sagrupēti pēc elementa tipa; noklikšķiniet uz pelēkās bultiņas, lai izvērstu vai sakļautu elementu grupu.



Pārskatiet veidnes izmaiņas un pēc tam noklikšķiniet uz *Import*, lai veiktu veidnes importēšanu. Lietotāja saskarnē tiks parādīts ziņojums par importēšanas veiksmīgu vai neveiksmīgu izpildi.

Eksporta formāts Kad veidne tiek eksportēta, Zabbix ģenerē strukturētu YAML, JSON vai XML formātu. Eksportā ir iekļauti veidnes elementi, piemēram, veidnes metadati, vienumi, makrosi, trigeri, informācijas paneļi un citi.

Katram elementam ir noteikts mērķis, un tas var saturēt ligzdotus elementus.

Turpmākajās sadaļās ir aprakstīts katrs elements eksporta formātā. Piemēros izmantota veidne Linux by Zabbix agent.

Daudzpunkte (`...`) norāda uz elementiem, kas izlaisti īsuma labad.

Piezīme (`skatīt tabulu zemāk`) tiek izmantota, ja elements ir izskaidrots detalizētāk vēlākajās sadaļās.

```
zabbix_export:
  version: '8.0'
  template_groups:
    - uuid: 846977d1dfed4968bc5f8bdb363285bc
      name: 'Templates/Operating systems'
  templates: (skatīt tabulu zemāk)
  triggers: (skatīt tabulu zemāk)
  graphs: (skatīt tabulu zemāk)
```

Veidnes metadati

Element	Type	Apraksts
version	string	(obligāts) Eksporta versija.
template_groups		(obligāts) Saknes elements veidņu grupām.
uuid	string	(obligāts) Unikāls šīs veidņu grupas identifikators.
name	string	(obligāts) Veidņu grupas nosaukums.
host_groups		Saknes elements hostu grupām, ko izmanto hostu prototipi.
uuid	string	(obligāts) Unikāls šīs hostu grupas identifikators.
name	string	(obligāts) Hostu grupas nosaukums.
templates		Saknes elements veidnēm.
triggers		Saknes elements veidnes trigeru elementiem, kas ir tādi paši kā veidnes vienuma trigeriem.
graphs		Saknes elements veidnes grafiku elementiem, kas ir tādi paši kā hostu grafikiem.

Veidnes

```
zabbix_export:
  (...)
  templates:
    - uuid: f8f7908280354f2abeed07dc788c3747
      template: 'Linux by Zabbix aģents'
      name: 'Linux by Zabbix aģents'
      description: |
        Šī ir oficiāla Linux veidne. Tai nepieciešams Zabbix aģents 8.0 vai jaunāks. (...)
      wizard_ready: 'YES'
      readme: |
        ## Pārskats

        Šī ir oficiāla Linux veidne. Tai nepieciešams Zabbix aģents 8.0 vai jaunāks. (...)
      vendor:
        name: Zabbix
        version: 8.0-2
      groups:
        - name: 'Templates/Operating systems'
      items: (see table below)
      discovery_rules: (see table below)
      tags:
        - tag: class
          value: os
        - tag: target
          value: linux
      macros:
        - macro: '{$AGENT.TIMEOUT}'
          value: 3m
          description: 'Noildze, pēc kuras aģents tiek uzskatīts par nepieejamu. Darbojas tikai aģentiem,
          config: (see table below)
        - macro: '{$CPU.UTIL.CRIT}'
          value: '90'
          description: 'Kritiska CPU noslodzes robežvērtība procentos.'
          config: (see table below)
        (...)
```

```
      dashboards: (see table below)
      valuemaps: (see table below)
    (...)
```

Element	Type	Description
uuid	string	(required) Unikāls šīs veidnes identifikators.
template	string	(required) Unikāls veidnes nosaukums.
name	string	Redzamais veidnes nosaukums.
description	text	Veidnes apraksts.
wizard_ready	text	Vai veidne ir pieejama izvēlei Host Wizard. Iespējamās vērtības:[1] NO (0, noklusējums), YES (1). Skatiet arī: Template object (`wizard_ready`).
readme	text	Veidnei specifiskas konfigurācijas instrukcijas, kas jāattēlo Host Wizard. Atbalsta Markdown formatējumu.
vendor		Saknes elements veidnes piegādātājam (ir klātesošs, ja eksportētajā veidnē ir piegādātāja dati).
name	string	(required) Veidnes piegādātāja nosaukums.
version	string	(required) Veidnes versija. Priekš out-of-the-box templates versija tiek attēlota šādi: Zabbix galvenā versija, atdalītājs (”-”), revīzijas numurs (tiek palielināts ar katru jaunu veidnes versiju un atiestatīts ar katru Zabbix galveno versiju). Piemēram, 7.0-0, 7.0-3, 8.0-0, 8.0-3.
templates		Saknes elements saistītajām veidnēm.
name	string	(required) Veidnes nosaukums.
groups		Saknes elements veidņu grupām.
name	string	(required) Veidnes grupas nosaukums.
items		Saknes elements template items.
discovery_rules		Saknes elements template low-level discovery rules.
httptests		Saknes elements template web scenarios.
tags		Saknes elements veidnes tagiem.
tag	string	(required) Taga nosaukums.
value	string	Taga vērtība.
macros		Saknes elements veidnes lietotāja makro.
macro	string	(required) Lietotāja makro nosaukums.
type	string	Lietotāja makro tips. Iespējamās vērtības:[1] TEXT (0, noklusējums), SECRET_TEXT (1), VAULT (2). Skatiet arī: User macro object (`type`).
value	string	Lietotāja makro vērtība.
description	string	Lietotāja makro apraksts.
config		Saknes elements template macro configuration, kas nosaka, kā makro tiek attēlots Host Wizard.
dashboards		Saknes elements template dashboards.
valuemaps		Saknes elements template value maps.

Veidnes vienumi

```
zabbix_export:
  (...)
  templates:
    (...)
      items:
        - uuid: f94f9f4699e94c369e6c98b2a2f485ce
          name: 'Zabbix aģenta ping'
          key: agent.ping
          description: 'Šim vienumam aģents vienmēr atgriež "1". To var izmantot kopā ar `nodata()`, lai p
          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 tālāk)
      tags:
        - tag: component
          value: os
      triggers: (skatiet tabulu tālāk)
    - 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 tālāk)
      master_item:
        key: 'vm.memory.size[pavailable]'
      tags:
        - tag: component
          value: memory
      triggers: (skatiet tabulu tālāk)
    (...)
```

Element	Type	Description
uuid	string	(obligāts) Šī vienuma unikālais 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), TELEMETRY_QUERY (24).
snmp_oid	string	(obligāts SNMP_AGENT vienumiem) SNMP objekta ID.
key	string	(obligāts) Vienuma atslēga.
delay	string	Vienuma atjaunināšanas intervāls. Noklusējums: 1m. TRAP vienumiem vērtība vienmēr būs 0.
history	string	Laika periods (izmantojot laika sufiksu, lietotāja makro vai LLD makro), cik ilgi jāsaglabā vēstures dati. Noklusējums: 31d.
trends	string	Laika periods (izmantojot laika sufiksu, lietotāja makro vai LLD makro), 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 to hostu IP adrešu saraksts, kuriem atļauts nosūtīt datus šim vienumam. Atbalstīts TRAP un HTTP_AGENT vienumiem.
units	string	Saņemtās vērtības mērvienības (bps, B u. c.).
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	Description
username	string	(obligāts SSH un TELNET vienumiem) Lietotājvārds autentifikācijai. Atbalstīts SIMPLE, ODBC, JMX un HTTP_AGENT vienumiem. Izmantojot JMX vienumiem, jānorāda arī `password` (skatiet tālāk) vai abiem elementiem jābūt tukšiem.
password	string	(obligāta SSH un TELNET vienumiem) Parole autentifikācijai. Atbalstīta SIMPLE, ODBC, JMX un HTTP_AGENT vienumiem. Izmantojot JMX vienumiem, jānorāda arī `username` (skatiet iepriekš) vai abiem elementiem jābūt tukšiem.
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, kuru aizpilda vienums. Iespējamās vērtības:[1] NONE (0), ALIAS (4) u. c. (atbalstītos laukus skatiet sadaļā Hosta inventarizācija).
valuemap		Saknes elements vienuma vērtību kartēm.
name	string	(obligāts) Vērtību kartes nosaukums, ko izmantot vienumam.
logtimefmt	string	Laika formāts žurnāla ierakstos. Atbalstīts vienumiem ar LOG vērtības tipu.
preprocessing		Saknes elements vienuma vērtību priekšapstrādei.
step		Saknes elements veidnes vienuma vērtību priekšapstrādes darbībām.
jmx_endpoint	string	JMX galapunkts. Atbalstīts JMX vienumiem.
master_item		(obligāts DEPENDENT vienumiem) Atkarīgā vienuma galvenā vienuma saknes elements.
key	string	(obligāts) Atkarīgā vienuma galvenā vienuma atslēga.
timeout	string	Vienuma datu aptaujas pieprasījuma noildze. Atbalstīts Noildzes sarakstā norādītajiem vienumu tipiem.
url	string	(obligāts HTTP_AGENT vienumiem) URL virkne.
query_fields		Vaicājuma parametru saknes elements. Atbalstīts HTTP_AGENT vienumiem.
name	string	(obligāts HTTP_AGENT vienumiem) Vaicājuma parametra nosaukums.
value	string	Vaicājuma parametra vērtība. Atbalstīts HTTP_AGENT vienumiem.
parameters		Lietotāja definētu parametru saknes elements. 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	Ar komatiem atdalīti nepieciešamo HTTP statusa kodu diapazoni. Atbalstīts HTTP_AGENT vienumiem.
follow_redirects	string	Datu aptaujas laikā sekot atbildes novirzēm. 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		HTTP(S) pieprasījuma galveņu saknes elements. 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	Description
output_format	string	Kā apstrādāt atbildi. Atbalstīts HTTP_AGENT vienumiem. Iespējamās vērtības:[1] RAW (0, noklusējums), JSON (1).
allow_traps	string	Atļaut vērtību aizpildīt līdzīgi trapera 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 nosaukumam. Atbalstīts HTTP_AGENT vienumiem. Iespējamās vērtības:[1] NO (0, noklusējums), YES (1).
time_shift	string	Cik tālu pagātnē (pirms pašreizējā laika) beidzas laika diapazons, par kuru vienums veic vaicājumu. Atbalstīts TELEMETRY_QUERY vienumiem.
lookback_limit	string	Cik tālu pagātnē (pirms pašreizējā laika) vienums meklē datus, ja nav iepriekšēja vaicājuma, no kura turpināt (piemēram, vienuma pirmajā izpildes reizē), vai ja iepriekšējais vaicājums ir vecāks par `delay` (piemēram, tāpēc, ka Zabbix serveris nedarbojās). Atbalstīts TELEMETRY_QUERY vienumiem.
granularity	string	Segmenta garums, kādā tiek sadalīts viss vaicājuma laika diapazons. Atbalstīts TELEMETRY_QUERY vienumiem.
query		(obligāts TELEMETRY_QUERY vienumiem) Telemetrijas vaicājuma saknes elements. Nosaka, ko vienums vaicā datu avotā, kā tas filtrē datus un kā apkopo rezultātu.
signal_type	string	(obligāts) Signāla tips, par kuru vienums veic vaicājumu. Atbalstīts TELEMETRY_QUERY vienumiem. Iespējamās vērtības:[1] TRACES (0), METRICS (1), LOGS (2).
metric_point_type	string	(obligāts METRICS signāla tipam) Metrikas punkta tips, par kuru vienums veic vaicājumu. Atbalstīts TELEMETRY_QUERY vienumiem. Iespējamās vērtības:[1] SUM (0), GAUGE (1), HISTOGRAM (2), EXPONENTIAL_HISTOGRAM (3).
columns		(obligāts) Telemetrijas datu kolonnu saknes elements, par kurām vienums veic vaicājumu. Šim elementam ir šādi elementi: `column (string)` - (obligāts) Telemetrijas datu kolonnas nosaukums, par kuru vienums veic vaicājumu. `attribute_key (string)` - (obligāts kolonnām, kurās ir atslēgu un vērtību pāri) Atslēgas nosaukums, par kuru vienums veic vaicājumu, kolonnā, kurā ir atslēgu un vērtību pāri.
aggregated_columns		(obligāts) Telemetrijas datu apkopoto kolonnu saknes elements, par kurām vienums veic vaicājumu. Šim elementam ir šādi elementi: `column (string)` - (obligāts MIN, MAX, AVG, SUM un PERCENTILE funkcijām) Telemetrijas datu kolonnas nosaukums, par kuru vienums veic vaicājumu. `function (string)` - (obligāts) Funkcija, ko vienums lieto kolonnas ierakstiem. Iespējamās vērtības:[1] MIN (1), MAX (2), AVG (3), COUNT (4), SUM (5), PERCENTILE (8). `parameters (array)` - (obligāts PERCENTILE funkcijām) Percentile, ko vienums aprēķina no kolonnas ierakstiem. `alias (string)` - (obligāts) Aprēķinātās vērtības unikālais nosaukums, ko izmanto tās identificēšanai vienuma vērtībā.

Element	Type	Description
filter		(obligāts) Telemetrijas vaicājuma filtra saknes elements vaicājuma kolonnu ierakstiem. Šim elementam ir šādi elementi: `evaltype` (`string`) - (obligāts) Filtra nosacījumu novērtēšanas metode. Iespējamās vērtības:[1] AND_OR (0), AND (1), OR (2), FORMULA (3). `formula` (`string`) - (obligāts FORMULA novērtēšanas metodēm) Lietotāja definēta izteiksme filtru nosacījumu novērtēšanai ar pielāgotu izteiksmi. `conditions` (`string`) - (obligāts) Telemetrijas datu filtra nosacījumu saknes elements, kuriem jāatbilst telemetrijas datu kolonnas ierakstam. Elementam `conditions` ir šādi elementi: `column` (`string`) - (obligāts) Telemetrijas datu kolonnas nosaukums, uz kuru attiecas filtra nosacījumi. `attribute_key` (`string`) - (obligāts kolonnām, kurās ir atslēgu un vērtību pāri) Atslēgas nosaukums, uz kuru attiecas nosacījums, kolonnā, kurā ir atslēgu un vērtību pāri. `operator` (`string`) - (obligāts) Filtra nosacījuma operators. Iespējamās vērtības kolonnām ar vienu vērtību:[1] EQUAL (0), NOT_EQUAL (1), LIKE (2), NOT_LIKE (3). Iespējamās vērtības kolonnām, kurās ir atslēgu un vērtību pāri:[1] EQUAL (0), NOT_EQUAL (1), EXISTS (12). `formulaid` (`string`) - (obligāts FORMULA novērtēšanas metodēm) Patvaļīgs unikāls ID, ko izmanto, lai atsauktos uz nosacījumu pielāgotā izteiksmē. `value` (`string`) - (obligāts EQUAL, NOT_EQUAL, LIKE un NOT LIKE darbībām) Vērtība, ar kuru salīdzināt.
tags		Vienuma tagu saknes elements.
tag	string	(obligāts) Taga nosaukums.
value	string	Taga vērtība.
triggers		Veidnes vienuma trigeru saknes elements.

Note:
Skatiet arī: Vienuma objekts (attiecīgo īpašību ar atbilstošu nosaukumu).

Veidnes vienuma vērtības priekšapstrādes soļi

```
zabbix_export:
  (...)
  templates:
    (...)
      items:
        (...)
        - uuid: 58818005e76d46dda14d6592f601ab00
          (...)
          preprocessing:
            - type: JSONPATH
              parameters:
                - $.length()
            - type: DISCARD_UNCHANGED_HEARTBEAT
              parameters:
                - 12h
          (...)
```

Element	Type	Apraksts
type	string	(obligāts) 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, calculated: received value - previous value), CHANGE_PER_SECOND (10, calculated: (received value - previous value)/(time now - time of last check)), 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		(obligāts) 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, default), DISCARD_VALUE (1), CUSTOM_VALUE (2), CUSTOM_ERROR (3).
error_handler_params	string	Kļūdu apstrādātāja parametri.

Note:
Skatiet arī: Vienuma priekšapstrādes objekts (skatiet attiecīgo īpašību ar atbilstošu 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
        (...)
```

Element	Type	Description
uuid	string	(required) Unikāls šī trigeris identifikators.
expression	string	(required) Trigeris izteiksme.
recovery_mode	string	Pamats OK notikumu ģenerēšanai. Iespējamās vērtības:[1] EXPRESSION (0, default), RECOVERY_EXPRESSION (1), NONE (2).
recovery_expression	string	Trigeris atjaunošanās izteiksme.
correlation_mode	string	Korelācijas režīms (bez notikumu korelācijas vai notikumu korelācija pēc taga). Iespējamās vērtības:[1] DISABLED (0, default), TAG_VALUE (1).
correlation_tag	string	Taga nosaukums, ko izmantot notikumu korelācijai.
name	string	(required) Trigeris nosaukums.
event_name	string	Notikuma nosaukums.
opdata	string	Operatīvie dati.
url_name	string	URL etiķete, kas saistīta ar trigeris.
url	string	Ar trigeris saistītais URL.
status	string	Trigeris statuss. Iespējamās vērtības:[1] ENABLED (0, default), DISABLED (1).
priority	string	Trigeris smagums. Iespējamās vērtības:[1] NOT_CLASSIFIED (0, default), INFO (1), WARNING (2), AVERAGE (3), HIGH (4), DISASTER (5).
description	text	Trigeris apraksts.
type	string	Notikumu ģenerēšanas veids (viens problēmas notikums vai vairāki problēmas notikumi). Iespējamās vērtības:[1] SINGLE (0, default), MULTIPLE (1).
manual_close	string	Manuāla problēmas notikumu aizvēršana. Iespējamās vērtības:[1] NO (0, default), YES (1).
dependencies		Atvases elements atkarībām.
name	string	(required) Atkarīgā trigeris nosaukums.
expression	string	(required) Atkarīgā trigeris izteiksme.
recovery_expression	string	Atkarīgā trigeris atjaunošanās izteiksme.
tags		Atvases elements trigeris tagiem.
tag	string	(required) Taga nosaukums.
value	string	Taga vērtība.

Note:
Skatiet arī: Trigger object (skatiet attiecīgo īpašību ar atbilstošu nosaukumu).

Veidnes zemā līmeņa atklāšanas noteikumi

```
zabbix_export:
  (...)
  templates:
    (...)
      discovery_rules:
        - uuid: acfdea9c46ef48c68e6636f43b8f96a2
          name: 'Network interface discovery'
          key: net.if.discovery
          delay: 1h
          filter: (see table below)
          description: 'The discovery of network interfaces.'
          item_prototypes: (see table below)
          trigger_prototypes: (see table below)
          graph_prototypes: (see table below)
        (...)
```

Attention:
Lielākā daļa veidnes zemā līmeņa atklāšanas noteikuma 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, default), TRAP (2), SIMPLE (3), INTERNAL (5), ZABBIX_ACTIVE (7), EXTERNAL (10), ODBC (11), IPMI (12), SSH (13), TELNET (14), JMX (16), DEPENDENT (18), HTTP_AGENT (19), SNMP_AGENT (20), ITEM_TYPE_SCRIPT (21), ITEM_TYPE_BROWSER (22).
key	string	(obligāts) Zemā līmeņa atklāšanas noteikuma atslēga.
filter		Saknes elements veidnes zemā līmeņa atklāšanas noteikuma filtriem.
lifetime	string	Laika periods (izmantojot sekundes, laika sufiksu vai lietotāja makro), 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 makro), 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 veidnes vienuma prototipa elementiem, kas ir tādi paši kā veidnes vienumiem.
trigger_prototypes		Saknes elements veidnes trigeru prototipa elementiem, kas ir tādi paši kā veidnes vienuma trigeriem.
graph_prototypes		Saknes elements veidnes grafiku prototipa elementiem, kas ir tādi paši kā hosts grafikiem.
host_prototypes		Saknes elements veidnes hostu prototipa elementiem, kas ir tādi paši kā hosts.
parent_discovery_rule		Saknes elements zemā līmeņa atklāšanas noteikuma prototipa vecākajam zemā līmeņa atklāšanas noteikumam (vai noteikuma prototipam). Šis īpašums norāda, ka tas ir LLD noteikuma prototips, tiešs norādītā noteikuma/noteikuma prototipa bērnelements.
key	string	(obligāts) Vecākā zemā līmeņa atklāšanas noteikuma (vai noteikuma prototipa) atslēga.
master_item	string	(obligāts DEPENDENT noteikumiem) Saknes elements atkarīgā noteikuma galvenajam vienumam.
lld_macro_paths		Saknes elements zemā līmeņa atklāšanas noteikuma makro ceļiem.
lld_macro	string	(obligāts) Zemā līmeņa atklāšanas noteikuma makro nosaukums.
path	string	(obligāts) Vērtības atlasītājs, kas tiks piešķirta attiecīgajam makro.
preprocessing		Saknes elements zemā līmeņa atklāšanas noteikuma vērtības pirmapstrādei.
step		Saknes elements zemā līmeņa atklāšanas noteikuma vērtības pirmapstrādes soļa elementiem, kas ir tādi paši kā veidnes vienuma vērtības pirmapstrādes soļi, izņemot to, ka pieejams mazāks iespējamo vērtību skaits. Skatiet arī: LLD noteikuma pirmapstrādes objekts.
overrides		Saknes elements zemā līmeņa atklāšanas noteikuma pārrakstīšanas noteikumiem.
name	string	(obligāts) Unikāls pārrakstīšanas nosaukums.
step	string	(obligāts) Unikāls pārrakstīšanas kārtas numurs.
stop	string	Pārtraukt nākamo pārrakstīšanu apstrādi, ja atbilst.
filter		Saknes elements veidnes zemā līmeņa atklāšanas noteikuma pārrakstīšanas noteikuma filtra elementiem, kas ir tādi paši kā veidnes zemā līmeņa atklāšanas noteikuma filtriem.
operations		Saknes elements veidnes zemā līmeņa atklāšanas noteikuma pārrakstīšanas darbībām.

Note:
Skatiet arī: LLD noteikuma objekts (skatiet attiecīgo īpašību ar atbilstošu nosaukumu).

Veidnes zemā līmeņa atklāšanas noteikuma 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
      (...)
```

Element	Type	Description
evaltype	string	Pārraksta filtra nosacījumu izvērtēšanas metodi. Iespējamās vērtības:[1] AND_OR (0, noklusējums), AND (1), OR (2), FORMULA (3).
formula	string	Pielāgota aprēķina formula filtra nosacījumiem.
conditions		Filtra nosacījumu saknes elements.
macro	string	(obligāts) Zemā līmeņa atklāšanas makro nosaukums, uz kuru 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	(obligāts) Patvaļīgs unikāls ID, ko izmanto, lai pielāgotajā izteiksmē atsauktos uz nosacījumu. Var saturēt tikai lielos burtus. ID lietotājam jādefinē, modificējot filtra nosacījumus, taču, pieprasot tos vēlāk, tas tiks ģenerēts no jauna.

Note:
Skatiet arī: LLD noteikuma filtra objekts (skatiet attiecīgo īpašību ar atbilstošu nosaukumu).

Veidnes zema līmeņa atklāšanas noteikuma pārrakstīšanas darbības

Element	Type	Description
operationobject	string	Objekts, kuram tiek lietota darbība. Iespējamās vērtības:[1] ITEM_PROTOTYPE (0), TRIGGER_PROTOTYPE (1), GRAPH_PROTOTYPE (2), HOST_PROTOTYPE (3).
operator	string	Pārrakstīšanas 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 pārrakstīšanas nosacījuma operatoram.
status	string	Objekta statuss pēc pārrakstīšanas darbības.
discover	string	Vai objekts tiek pievienots atklāšanas rezultātā.
delay	string	Atjaunināšanas intervāls, kas iestatīts vienuma prototipam pēc pārrakstīšanas darbības.
history	string	Vēstures glabāšanas periods, kas iestatīts vienuma prototipam pēc pārrakstīšanas darbības.
trends	string	Tendenču glabāšanas periods, kas iestatīts vienuma prototipam pēc pārrakstīšanas darbības.
severity	string	Trigeris prototipa smaguma pakāpe, kas iestatīta pēc pārrakstīšanas darbības.
tags		Saknes elements tagu kopai, kas objektam iestatīta pēc pārrakstīšanas darbības.
tag	string	(obligāts) Taga nosaukums.
value	string	Taga vērtība.
templates		Saknes elements veidnēm, kas saistītas ar hosts prototipu pēc pārrakstīšanas darbības.
name	string	(obligāts) Veidnes nosaukums.
inventory_mode	string	Hosts prototipa inventarizācijas režīms, kas iestatīts pēc pārrakstīšanas darbības.

Note:
Skatiet arī: LLD noteikuma pārrakstīšanas darbības objekts (skatiet attiecīgo īpašību ar atbilstošu nosaukumu).

Veidnes web scenāriji

Element	Type	Description
uuid	string	(required) Unique identifier for this web scenario.
name	string	(required) Web scenario name.
delay	string	Frequency (using seconds, time suffix, or user macro) of executing web scenario. Default: 1m.
attempts	integer	The number of attempts for executing web scenario steps. Possible values:[1] 1-10 (default: 1).
agent	string	Klienta aģents. Zabbix izliksies par izvēlēto pārlūkprogrammu. Tas ir noderīgi, ja vietne atgriež atšķirīgu saturu dažādām pārlūkprogrammām. Default: Zabbix.
http_proxy	string	Starpniekserveris, kas tiks izmantots web scenārijam, norādīts kā: `http://[username[:password]@]proxy.example.com[:port]`
variables		Saknes elements web scenārija mainīgajiem, kurus var izmantot scenārija soļos.
name	string	(required) Mainīgā nosaukums.
value	text	(required) Mainīgā vērtība.
headers		Saknes elements HTTP galvenēm, kas tiks nosūtītas, veicot pieprasījumu. Galvenes jāuzskaita, izmantojot to pašu sintaksi, kāda tām būtu HTTP protokolā.
name	string	(required) Galvenes nosaukums.
value	text	(required) Galvenes vērtība.
status	string	Web scenārija statuss. Possible values:[1] ENABLED (0, default), DISABLED (1).
authentication	string	Autentifikācijas metode. Possible values:[1] NONE (0, default), BASIC (1), NTLM (2), Kerberos (3), or Digest (4).
http_user	string	Lietotājvārds, ko izmanto BASIC (HTTP), NTLM, Kerberos vai Digest autentifikācijai.
http_password	string	Parole, ko izmanto BASIC (HTTP), NTLM, Kerberos vai Digest autentifikācijai.
verify_peer	string	Pārbaudīt tīmekļa servera SSL sertifikātu. Possible values:[1] NO (0, default), YES (1).
verify_host	string	Pārbaudīt, vai tīmekļa servera sertifikāta lauks *Common Name* vai *Subject Alternate Name* atbilst. Possible values:[1] NO (0, default), YES (1).
ssl_cert_file	string	SSL sertifikāta faila nosaukums, ko izmanto klienta autentifikācijai (tam jābūt PEM formātā).
ssl_key_file	string	SSL privātās atslēgas faila nosaukums, ko izmanto klienta autentifikācijai (tam jābūt PEM formātā).
ssl_key_password	string	SSL privātās atslēgas faila parole.
steps		(required) Saknes elements veidnes web scenārija soļiem.
tags		Saknes elements web scenārija tagiem.
tag	string	(required) Taga nosaukums.
value	string	Taga vērtība.

Note:
Skatīt arī: Web scenario object (skatiet attiecīgo īpašību ar atbilstošu nosaukumu).

Veidnes tīmekļa scenārija soļi

Element	Type	Description
name	string	(required) Tīmekļa scenārija soļa nosaukums.
url	string	(required) Uzraudzībai paredzētais URL.
query_fields		Vaicājuma parametru saknes elements (HTTP lauku masīvs, kas pieprasījuma izpildes laikā tiek pievienots URL).
name	string	(required) Vaicājuma parametra nosaukums.
value	string	Vaicājuma parametra vērtība.
posts		HTTP POST mainīgo saknes elements (virknes veidā neapstrādāti POST dati vai HTTP lauku masīvs (formas lauku dati)).

Element	Type	Description
name	string	(required) POST lauka nosaukums.
value	string	(required) POST lauka vērtība.
variables		Soļa līmeņa mainīgo (makro) saknes elements, kas jāpiemēro pēc šī soļa. Ja mainīgā vērtībai ir prefikss 'regex:', tās vērtība tiek iegūta no datiem, ko atgriež šis solis, atbilstoši regulārās izteiksmes paraugam, kas seko prefiksam 'regex:'
name	string	(required) Mainīgā nosaukums.
value	text	(required) Mainīgā vērtība.
headers		HTTP galveņu saknes elements, kas jānosūta pieprasījuma izpildes laikā.
name	string	(required) Galvenes nosaukums.
value	text	(required) Galvenes vērtība.
follow_redirects	string	Sekot HTTP novirzēm. Iespējamās vērtības:[1] NO (0), YES (1, noklusējums).
retrieve_mode	string	HTTP atbildes iegūšanas režīms. Iespējamās vērtības:[1] BODY (0, noklusējums), HEADERS (1), BOTH (2).
timeout	string	Soļa izpildes noildze (sekundēs, laika sufikss vai lietotāja makro). Noklusējums: 15s.
required	string	Teksts, kam jābūt atbildē (ja tukšs, tiek ignorēts).
status_codes	string	Ar komatiem atdalīts pieņemamo HTTP statusa kodu saraksts (piem., 200-201,210-299; ja tukšs, tiek ignorēts).

Note:
Skatiet arī: Web scenario step object (skatiet atbilstošo īpašību ar sakritīgu nosaukumu).

Veidnes makro konfigurācija

```
zabbix_export:
  (...)
  templates:
    - uuid: f8f7908280354f2abeed07dc788c3747
      (...)
      macros:
        - macro: '{$AGENT.TIMEOUT}'
          (...)
          config:
            type: TEXT
            priority: '1'
            label: 'Sekundes kopš pēdējā redzētā Zabbix aģenta'
            description: 'Noildze, pēc kuras aģents tiek uzskatīts par nepieejamu.'
        - macro: '{$CPU.UTIL.CRIT}'
          (...)
          config:
            type: TEXT
            priority: '2'
            section_name: Thresholds
            label: 'Norādītais CPU noslodzes slieksnis'
            description: 'Kritiskais CPU noslodzes slieksnis procentos. Diapazonā no 0 līdz 100 ieskaitot.
            regex: '^-?([0-9]+|(([0-9]+)\.([0-9]+)))$'
        (...)
        - macro: '{$IFCONTROL}'
          (...)
          config:
            type: CHECKBOX
            priority: '19'
            label: 'Saskarņu vadība'
            description: 'Izveidot 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 makriem) Makro etiķete.
description	text	Makro palīdzības teksts. Atbalsta Markdown formatējumu.
required	string	Norāda, ka makro ir obligāts. Iespējamās vērtības:[1] NO (0), YES (1). Atbalstīts TEXT un LIST makriem.
regex	string	Regulārā izteiksme lietotāja ievades validēšanai teksta laukā. Atbalstīts TEXT makriem.
options		Saknes elements LIST vienumiem vai CHECKBOX vērtībām.
value	string	(obligāts LIST makriem) LIST vienuma vērtība.
text	string	(obligāts LIST makriem) LIST vienuma teksts.
checked	string	(obligāts CHECKBOX makriem) Vērtība, kas apzīmē atzīmētu stāvokli. Iespējamās vērtības:[1] 0 (false), 1 (true). Atbalstīts CHECKBOX makriem.
unchecked	string	(obligāts CHECKBOX makriem) Vērtība, kas apzīmē neatzīmētu stāvokli. Iespējamās vērtības:[1] 0 (false), 1 (true). Atbalstīts CHECKBOX makriem.

Note:
Skatiet arī: Makro konfigurācijas objekts (skatiet atbilstošo īpašību ar sakritīgu nosaukumu).

Veidnes informācijas paneļi

```
zabbix_export:
  (...)
  templates:
    (...)
      dashboards:
        - uuid: c689ad3115fd46a4b927d1f70ee2e5a4
          name: Filesystems
          pages:
            - name: Overview
              widgets: (see table below)
        (...)
```

Element	Type	Description
uuid	string	(obligāts) Šī informācijas paneļa unikāls identifikators.
name	string	(obligāts) Veidnes informācijas paneļa nosaukums.
display pe- riod	integer	Informācijas paneļa lapu attēlošanas periods.
auto_start	string	Slaidrādes automātiskā palaišana. Iespējamās vērtības:[1] NO (0), YES (1, noklusējums).
pages		Veidnes informācijas paneļa lapu saknes elements.
name	string	Lapas nosaukums.
display period	integer	Slaidrādes intervāls.
sortorder	integer	Lapas kārtošanas secība.
widgets		Saknes elements veidnes informācijas paneļa logrīkiem.

Note:
Skatiet arī: Veidnes informācijas paneļa objekts (skatiet attiecīgo īpašību ar atbilstošu nosaukumu).

Veidnes informācijas paneļa logrīki

```yaml
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 aģents'
                        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 aģents'
                        name: 'FS [{#FSTYPE}({#FSNAME})]: Space usage graph, in % (relative to max availab
                    - type: STRING
                      name: reference
                      value: AAAEC
        (...)
```

Element	Type	Apraksts
type	string	(obligāts) Logrīka tips.
name	string>Logrīka nosaukums.	
x	integer>Horizontālā pozīcija no veidnes informācijas paneļa kreisās puses. Iespējamās vērtības:[1] 0-71.	

Element	Type	Apraksts
y	integer>Vertikālā pozīcija no veidnes informācijas paneļa augšas. 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, noklusējums), YES (1).	
fields		Saknes elements veidnes informācijas paneļa logrīka laukiem.
type	string	(obligāts) 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	(obligāts) Logrīka lauka nosaukums.
value	mixed	(obligāts) Logrīka lauka vērtība atkarībā no lauka tipa.

Note:
Skatiet arī: Template dashboard widget object (skatiet attiecīgo īpašību ar atbilstošu nosaukumu).

Veidnes vērtību kartējumi

```
zabbix_export:
  (...)
  templates:
    (...)
      valuemaps:
        - uuid: 8c048c6cca8248f2860c208e8db0f59e
          name: ifOperStatus
          mappings:
            - value: '0'
              newvalue: unknown
            - value: '1'
              newvalue: notpresent
            - value: '2'
              newvalue: down
            - value: '3'
              newvalue: lowerlayerdown
            - value: '4'
              newvalue: testing
            - value: '5'
              newvalue: dormant
            - value: '6'
```

```
            newvalue: up
        (...)
```

Element	Type	Apraksts
uuid	string	(obligāts) Unikāls šī vērtību kartējuma identifikators.
name	string	(obligāts) Vērtību kartējuma nosaukums.
mapping		Kartējumu saknes elements.
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 sakritīgu nosaukumu).

Piezīmes [1] API veselo skaitļu vērtības iekavās, piemēram, ENABLED (0), ir minētas tikai uzziņai. Papildu informāciju skatiet saistītajā API objekta lapā tabulas ierakstā vai katras sadaļas beigās.

Hosts

Pārskats

Hosts tiek eksportēti kopā ar daudziem saistītiem objektiem un objektu attiecībām.

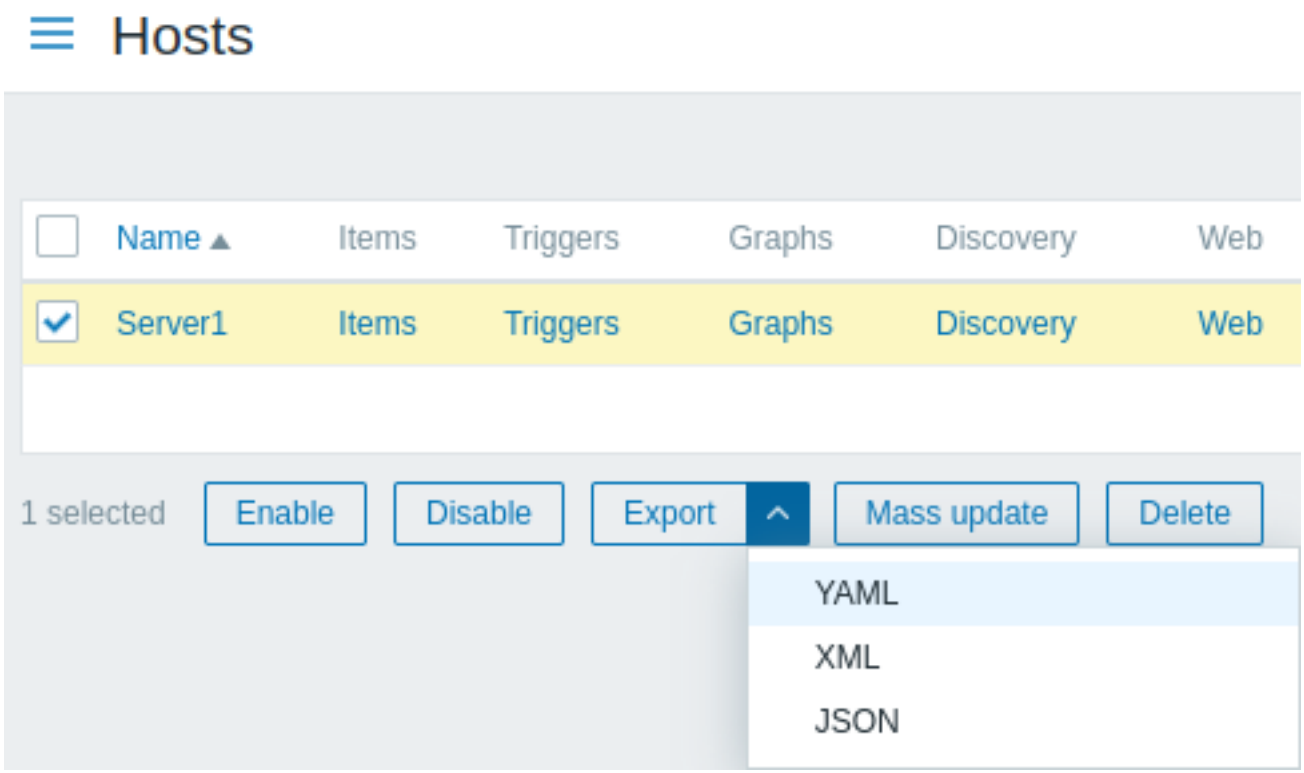
Host eksportā ir iekļauts:

- Saistītas host grupas
- Host dati
- Saistījums ar veidnēm
- Saistījums ar host grupām
- Host saskarnes
- Tieši saistīti vienumi
- Tieši saistīti trigeri
- Tieši saistīti atklāšanas noteikumi ar visiem prototipiem
- Tieši saistīti tīmekļa scenāriji
- Host makro
- Host inventāra dati
- Vērtību kartes
- Saistītie grafiki

Eksportēšana

Lai eksportētu hosts, rīkojieties šādi:

1. Dodieties uz *Datu vākšana* > *Hosts*.
2. Atzīmējiet to hosts izvēles rūtiņas, kurus vēlaties eksportēt.
3. Noklikšķiniet uz *Eksportēt* zem saraksta, lai eksportētu noklusējuma YAML formātā, vai noklikšķiniet uz bultiņas, lai atlasītu citu eksporta formātu.



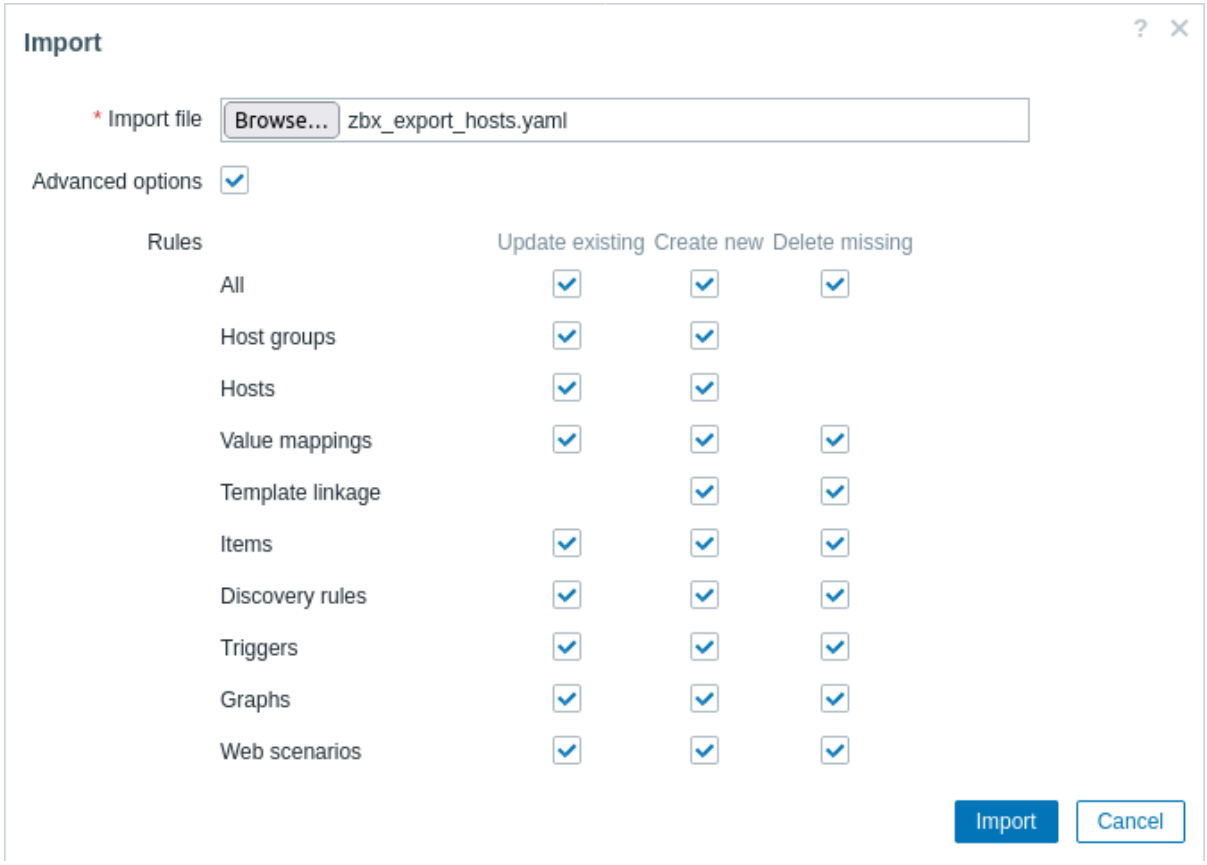
Atkarībā no atlasītā formāta hosts tiek eksportēti uz lokālu failu ar noklusējuma nosaukumu:

- `zabbix_export_hosts.yaml` - YAML eksportā;
- `zabbix_export_hosts.xml` - XML eksportā;
- `zabbix_export_hosts.json` - JSON eksportā.

Importēšana

Lai importētu hostus, rīkojieties, kā norādīts tālāk:

1. Atveriet *Data collection* > *Hosts*.
2. Augšējā labajā stūrī noklikšķiniet uz *Import*.
3. Atlasiet importēšanas failu.
4. Atzīmējiet to importēšanas noteikumu izvēles rūtiņas, kurus vēlaties lietot (skatiet tālāk).
5. Konfigurācijas formas apakšējā labajā stūrī noklikšķiniet uz *Import*.



Ja atzīmēsiet izvēles rūtiņu *Advanced options*, tiks parādīts detalizēts visu importējamo elementu saraksts — atzīmējiet vai noņemiet atzīmi katram importēšanas noteikumam atbilstoši nepieciešamībai.

Ja noklikšķināsiet uz izvēles rūtiņas rindā *All*, visi zem tās esošie elementi tiks atzīmēti vai atzīme tiem tiks noņemta.

Importēšanas noteikumi:

Rule	Description
Update existing	Esošie elementi tiks atjaunināti, izmantojot importēšanas faila datus. Pretējā gadījumā tie netiks atjaunināti.
Create new	Jauni elementi tiks izveidoti, izmantojot importēšanas faila datus. Pretējā gadījumā tie netiks izveidoti.
Delete missing	Esošie elementi, kas nav ietverti importēšanas failā, tiks noņemti. Pretējā gadījumā tie netiks noņemti. Ja sadaļā *Template linkage* ir atzīmēts *Delete missing*, pašreizējās veidņu saites, kas nav ietvertas importēšanas failā, tiks atsaistītas. No atsaistītajām veidnēm mantotās entītijas (vienumi, trigeri, grafiki u. c.) netiks noņemtas (ja vien katrai entītijai nav atlasīta arī opcija *Delete missing*).

Importēšanas laikā nevar pārskatīt izmaiņas hostos. Importēšanas veiksmes vai kļūmes ziņojums tiks parādīts lietotāja saskarnē.

Eksporta formāts

Kad hosts tiek eksportēts, Zabbix ģenerē strukturētu YAML, JSON vai XML formātu. Eksportā ir iekļauti hosta elementi, piemēram, hosta metadati, vienumi, makrosi, trigeri un citi.

Katram elementam ir noteikts mērķis, un tas var saturēt ligzdotus elementus.

Turpmākajās 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).

Elipse (`...`) norāda uz īsuma labad izlaistiem elementiem. Piezīme (`skatīt tabulu zemāk`) tiek izmantota, ja elements vēlākajās sadaļās ir izskaidrots detalizētāk.

```
zabbix_export:
  version: '8.0'
```

```
  host_groups:
    - uuid: 748ad4d098d447d492bb935c907f652f
      name: Databases
  hosts: (skatīt tabulu zemāk)
  graphs: (skatīt tabulu zemāk)
```

Hosts metadati

Element		Type	Description
version		string	(obligāts) Zabbix versija.
host_groups			(obligāts) Saknes elements host grupām.
	uuid	string	(obligāts) Unikāls identifikators šai host grupai.
	name	string	(obligāts) Host grupas nosaukums.
hosts			Saknes elements hosts.
graphs			Saknes elements host grafikiem.

Hosts

```
zabbix_export:
  (...)
  hosts:
    - host: 'MySQL server'
      name: 'MySQL server'
      groups:
        - name: Databases
      interfaces: (see table below)
      items: (see table below)
      discovery_rules: (see table below)
      tags:
        - tag: location
          value: Riga
      macros:
        (...)
        - macro: '{$MYSQL.DSN}'
          value: 192.0.2.1
        (...)
        - macro: '{$MYSQL.PASSWORD}'
          type: SECRET_TEXT
          description: 'MySQL lietotāja parole.'
        (...)
      valuemaps: (see table below)
  (...)
```

Element	Type	Description
host	string	(required) Unikāls hosts nosaukums.
name	string	Redzamais hosts nosaukums.
description	text	Hosts apraksts.
monitored_by	string	Kā hosts tiek uzraudzīts. Iespējamās vērtības:[1] SERVER (0, noklusējums), PROXY (1) vai PROXY_GROUP (2).
proxy		Starpniekservera saknes elements.
name	string	(required) Starpniekservera nosaukums (ja tāds ir), kas uzrauga hostu.
proxy_group		Starpniekserveru grupas saknes elements.
name	string	(required) Starpniekserveru grupas nosaukums (ja tāds ir), kas tiek izmantots 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).
ipmi_privilege	string	IPMI sesijas privilēģiju līmenis. Iespējamās vērtības:[1] CALLBACK (1), USER (2, noklusējums), OPERATOR (3), ADMIN (4), OEM (5).

Element	Type	Description
ipmi_username	string	IPMI pārbaužu lietotājvārds.
ipmi_password	string	IPMI pārbaužu parole.
templates		Saistīto veidņu saknes elements.
name	string	(required) Veidnes nosaukums.
groups		Hostu grupu, kurām hosts pieder, saknes elements.
name	string	(required) Hostu grupas nosaukums.
interfaces		hosta saskarņu saknes elements.
items		hosta vienumu saknes elements.
discovery_rules		hosta zemā līmeņa atklāšanas noteikumu saknes elements.
http_tests		hosta tīmekļa scenāriju saknes elements.
tags		Hosta tagu saknes elements.
tag	string	(required) Taga nosaukums.
value	string	Taga vērtība.
macros		Hosta makro saknes elements.
macro	string	(required) Lietotāja makro nosaukums.
type	string	Lietotāja makro tips. Iespējamās vērtības:[1] TEXT (0, noklusējums), SECRET_TEXT (1), VAULT (2).
value	string	Lietotāja makro vērtība.
description	string	Lietotāja makro apraksts.
inventory		Hosta inventāra saknes elements.
<inventory_property>	string	Inventāra īpašība. Visām īpašībām ir savs atbilstošais elements (`type`, `name`, `os` utt.; piemēram, skatiet eksporta formātu).
inventory_mode	string	Inventāra režīms. Iespējamās vērtības:[1] DISABLED (-1), MANUAL (0, noklusējums), AUTOMATIC (1).
valuemaps		hosta vērtību karšu saknes elements.

Note:
Skatiet arī: Host object (skatiet attiecīgo īpašību ar atbilstošu nosaukumu).

Hosts saskarnes

```
zabbix_export:
  (...)
  hosts:
    - host: 'MySQL server'
      (...)
      interfaces:
        - ip: 192.0.2.1
          interface_ref: if1
      (...)
```

Element	Type	Apraksts
default	string	Vai šī ir primārā hosts saskarne. Ņemiet vērā, ka vienam hostam var būt tikai viena primārā saskarne viena tipa ietvaros. 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 savienojumam ar hostu izmantot IP adresi (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 detaļām.
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.
max_repetitions	string	Maksimālā atkārtojumu vērtība vietējiem SNMP lielapjoma pieprasījumiem (GetBulkRequest-PDU). Atbalstīts SNMPv2 un SNMPv3 vienumiem (`discovery[]` un `walk[]` vienumi). Noklusējums: 10.

Element	Type	Apraksts
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).

Host vienumi

```
zabbix_export:
  (...)
  hosts:
    - host: 'MySQL server'
      (...)
      items:
        (...)
        - name: 'Binlog kešatmiņas diska izmantojums'
          type: DEPENDENT
          key: mysql.binlog_cache_disk_use
          value_type: FLOAT
          description: 'Transakciju skaits, kurās tika izmantota pagaidu diska kešatmiņa, jo tās neietilpa
          preprocessing: (skatiet tabulu tālāk)
          master_item:
            key: 'mysql.get_status_variables["{$MYSQL.DSN}","{$MYSQL.USER}","{$MYSQL.PASSWORD}"]'
          tags:
            - tag: component
              value: cache
        (...)
        - name: 'Bufera pūla izmantojums'
          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 bufera pūla lapu attiecība pret kopējo lapu skaitu.'
          tags:
```

```
        - tag: component
          value: memory
      triggers: (skatiet tabulu tālāk)
    (...)
    - name: Darbspējas laiks
      type: DEPENDENT
      key: mysql.uptime
      units: uptime
      description: 'Sekunžu skaits, cik ilgi serveris darbojas.'
      preprocessing: (skatiet tabulu tālāk)
      master_item:
        key: 'mysql.get_status_variables["{$MYSQL.DSN}","{$MYSQL.USER}","{$MYSQL.PASSWORD}"]'
      tags:
        - tag: component
          value: application
      triggers: (skatiet tabulu tālāk)
    (...)
```

Element	Type	Description
name	string	(obligāts) Vienuma nosaukums.
type	string	Vienuma tips. Iespējamās vērtības:[1] ZABBIX_PASSIVE (0, noklusējuma), TRAP (2), SIMPLE (3), INTERNAL (5), ZABBIX_ACTIVE (7), EXTERNAL (10), ODBC (11), IPMI (12), SSH (13), TELNET (14), CALCULATED (15), JMX (16), SNMP_TRAP (17), DEPENDENT (18), HTTP_AGENT (19), SNMP_AGENT (20), ITEM_TYPE_SCRIPT (21), ITEM_TYPE_BROWSER (22), ITEM_TYPE_NESTED (23), TELEMETRY_QUERY (24).
snmp_oid	string	(obligāts SNMP_AGENT vienumiem) SNMP objekta ID.
key	string	(obligāts) Vienuma atslēga.
delay	string	Vienuma atjaunināšanas intervāls. Noklusējums: 1m. TRAP vienumiem vērtība vienmēr būs 0.
history	string	Laika periods (izmantojot laika sufiksu, lietotāja makro vai LLD makro), cik ilgi jāglabā vēstures dati. Noklusējums: 31d.
trends	string	Laika periods (izmantojot laika sufiksu, lietotāja makro vai LLD makro), cik ilgi jāglabā tendenču dati. Noklusējums: 365d.
status	string	Vienuma statuss. Iespējamās vērtības:[1] ENABLED (0, noklusējuma), DISABLED (1).
value_type	string	Saņemtās vērtības tips. Iespējamās vērtības:[1] FLOAT (0), CHAR (1), LOG (2), UNSIGNED (3, noklusējuma), TEXT (4), BINARY (5), JSON (6).
allowed_hosts	string	Ar komatiem atdalīts to hostu IP adrešu saraksts, kuriem atļauts nosūtīt vienuma datus. Atbalstīts TRAP un HTTP_AGENT vienumiem.
units	string	Saņemtās vērtības mērvienības (bps, B u. c.).
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, noklusējuma), PUBLIC_KEY (1). Iespējamās vērtības HTTP_AGENT vienumiem:[1] NONE (0, noklusējuma), 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. Izmantojot JMX vienumiem, jānorāda arī `password` (skatiet tālāk) vai abi elementi jāatstāj tukši.
password	string	(obligāts SSH un TELNET vienumiem) Parole autentifikācijai. Atbalstīta SIMPLE, ODBC, JMX un HTTP_AGENT vienumiem. Izmantojot JMX vienumiem, jānorāda arī `username` (skatiet iepriekš) vai abi elementi jāatstāj tukši.

Element	Type	Description
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	Hostu inventarizācijas lauks, ko aizpilda vienums. Iespējamās vērtības:[1] NONE (0), ALIAS (4) u. c. (atbalstītos laukus skatiet sadaļā Hostu inventarizācija).
valuemap		Vienuma vērtību kartējumu saknes elements.
name	string	(obligāts) Vērtību kartējuma nosaukums, ko izmantot vienumam.
logtimefmt	string	Laika formāts žurnāla ierakstos. Atbalstīts LOG vērtības tipa vienumiem.
preprocessing		Vienuma vērtību priekšapstrādes saknes elements.
step		Hosta vienuma vērtību priekšapstrādes soļa saknes elements.
interface_ref	string	Atsauce uz hosta saskarni (formāts: `if<N>`).
jmx_endpoint	string	JMX galapunkts. Atbalstīts JMX vienumiem.
master_item		(obligāts DEPENDENT vienumiem) Atkarīgā vienuma galvenā vienuma saknes elements.
key	string	(obligāts) Atkarīgā vienuma galvenā vienuma atslēga.
timeout	string	Vienuma datu aptaujas pieprasījuma taimauts. Atbalstīts Taimautu sarakstā norādītajiem vienumu tipiem.
url	string	(obligāts HTTP_AGENT vienumiem) URL virkne.
query_fields		Vaicājuma parametru saknes elements. Atbalstīts HTTP_AGENT vienumiem.
name	string	(obligāts HTTP_AGENT vienumiem) Vaicājuma parametra nosaukums.
value	string	Vaicājuma parametra vērtība. Atbalstīts HTTP_AGENT vienumiem.
parameters		Lietotāja definētu parametru saknes elements. 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 pamatdati. Atbalstīts HTTP_AGENT vienumiem.
status_codes	string	Ar komatiem atdalīti nepieciešamo HTTP statusa kodu diapazoni. Atbalstīts HTTP_AGENT vienumiem.
follow_redirects	string	Datu aptaujas laikā sekot atbildes novirzēm. Atbalstīts HTTP_AGENT vienumiem. Iespējamās vērtības:[1] NO (0, noklusējuma), YES (1, noklusējuma).
post_type	string	POST datu pamatteksta tips. Atbalstīts HTTP_AGENT vienumiem. Iespējamās vērtības:[1] RAW (0, noklusējuma), JSON (2), XML (3).
http_proxy	string	HTTP(S) starpniekservera savienojuma virkne. Atbalstīts HTTP_AGENT vienumiem.
headers		HTTP(S) pieprasījuma galveņu saknes elements. 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 jāglabā. Atbalstīts HTTP_AGENT vienumiem. Iespējamās vērtības:[1] BODY (0, noklusējuma), HEADERS (1), BOTH (2).
request_method	string	Pieprasījuma metodes tips. Atbalstīts HTTP_AGENT vienumiem. Iespējamās vērtības:[1] GET (0, noklusējuma), POST (1), PUT (2), HEAD (3).
output_format	string	Kā apstrādāt atbildi. Atbalstīts HTTP_AGENT vienumiem. Iespējamās vērtības:[1] RAW (0, noklusējuma), JSON (1).
allow_traps	string	Atļaut aizpildīt vērtību tāpat kā trapper vienumam. Atbalstīts HTTP_AGENT vienumiem. Iespējamās vērtības:[1] NO (0, noklusējuma), YES (1).
ssl_cert_file	string	Publiskās SSL atslēgas faila ceļš. Atbalstīts HTTP_AGENT vienumiem.

Element	Type	Description
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, noklusējuma), YES (1).
verify_host	string	Vai pārbaudīt, ka savienojuma hosta nosaukums atbilst hosta sertifikātā norādītajam nosaukumam. Atbalstīts HTTP_AGENT vienumiem. Iespējamās vērtības:[1] NO (0, noklusējuma), YES (1).
time_shift	string	Cik tālu pagātnē (pirms pašreizējā laika) beidzas laika diapazons, par kuru vienums veic vaicājumu. Atbalstīts TELEMETRY_QUERY vienumiem.
lookback_limit	string	Cik tālu pagātnē (pirms pašreizējā laika) vienums meklē datus, ja nav iepriekšēja vaicājuma, no kura turpināt (piemēram, vienuma pirmās palaišanas laikā), vai ja iepriekšējais vaicājums ir vecāks par `delay` (piemēram, tāpēc, ka Zabbix serveris nedarbojās). Atbalstīts TELEMETRY_QUERY vienumiem.
granularity	string	Tā segmenta garums, kurā tiek sadalīts viss vaicājuma laika diapazons. Atbalstīts TELEMETRY_QUERY vienumiem.
query		(obligāts TELEMETRY_QUERY vienumiem) Telemetrijas vaicājuma saknes elements. Nosaka, ko vienums vaicā datu avotā, kā tas filtrē datus un kā apkopo rezultātu.
signal_type	string	(obligāts) Signāla tips, par kuru vienums veic vaicājumu. Atbalstīts TELEMETRY_QUERY vienumiem. Iespējamās vērtības:[1] TRACES (0), METRICS (1), LOGS (2).
metric_point_type	string	(obligāts METRICS signāla tipam) Metrikas punkta tips, par kuru vienums veic vaicājumu. Atbalstīts TELEMETRY_QUERY vienumiem. Iespējamās vērtības:[1] SUM (0), GAUGE (1), HISTOGRAM (2), EXPONENTIAL_HISTOGRAM (3).
columns		(obligāts) Telemetrijas datu kolonnu saknes elements, par kurām vienums veic vaicājumu. Šim elementam ir šādi elementi: `column (string)` - (obligāts) Telemetrijas datu kolonnas nosaukums, par kuru vienums veic vaicājumu. `attribute_key (string)` - (obligāts kolonnām, kurās ir atslēgu un vērtību pāri) Atslēgas nosaukums, par kuru vienums veic vaicājumu, kolonnā, kurā ir atslēgu un vērtību pāri.
aggregated_columns		(obligāts) Apkopoto telemetrijas datu kolonnu saknes elements, par kurām vienums veic vaicājumu. Šim elementam ir šādi elementi: `column (string)` - (obligāts MIN, MAX, AVG, SUM un PERCENTILE funkcijām) Telemetrijas datu kolonnas nosaukums, par kuru vienums veic vaicājumu. `function (string)` - (obligāts) Funkcija, ko vienums lieto kolonnas ierakstiem. Iespējamās vērtības:[1] MIN (1), MAX (2), AVG (3), COUNT (4), SUM (5), PERCENTILE (8). `parameters (array)` - (obligāts PERCENTILE funkcijām) Percentile, ko vienums aprēķina no kolonnas ierakstiem. `alias (string)` - (obligāts) Unikāls aprēķinātās vērtības nosaukums, ko izmanto tās identificēšanai vienuma vērtībā.

Element	Type	Description
filter		(obligāts) Telemetrijas vaicājuma filtra saknes elements vaicājuma kolonnu ierakstiem. Šim elementam ir šādi elementi: `evaltype` (`string`) - (obligāts) Filtra nosacījumu novērtēšanas metode. Iespējamās vērtības:[1] AND_OR (0), AND (1), OR (2), FORMULA (3). `formula` (`string`) - (obligāts FORMULA novērtēšanas metodēm) Lietotāja definēta izteiksme, ko izmanto filtra nosacījumu novērtēšanai ar pielāgotu izteiksmi. `conditions` (`string`) - (obligāts) Telemetrijas datu filtra nosacījumu saknes elements, kuriem jāatbilst telemetrijas datu kolonnas ierakstam. Elementam `conditions` ir šādi elementi: `column` (`string`) - (obligāts) Telemetrijas datu kolonnas nosaukums, uz kuru attiecas filtra nosacījumi. `attribute_key` (`string`) - (obligāts kolonnām, kurās ir atslēgu un vērtību pāri) Atslēgas nosaukums, uz kuru attiecas nosacījums, kolonnā, kurā ir atslēgu un vērtību pāri. `operator` (`string`) - (obligāts) Filtra nosacījuma operators. Iespējamās vērtības kolonnām ar vienu vērtību:[1] EQUAL (0), NOT_EQUAL (1), LIKE (2), NOT_LIKE (3). Iespējamās vērtības kolonnām, kurās ir atslēgu un vērtību pāri:[1] EQUAL (0), NOT_EQUAL (1), EXISTS (12). `formulaid` (`string`) - (obligāts FORMULA novērtēšanas metodēm) Patvaļīgs unikāls ID, ko izmanto, lai atsauktos uz nosacījumu pielāgotā izteiksmē. `value` (`string`) - (obligāts EQUAL, NOT_EQUAL, LIKE un NOT LIKE darbībām) Vērtība, ar kuru salīdzināt.
tags		Vienuma tagu saknes elements.
tag	string	(obligāts) Taga nosaukums.
value	string	Taga vērtība.
triggers		Hosta vienuma trigeru saknes elements.

Note:
Skatiet arī: Vienuma objekts (skatiet attiecīgo īpašību ar atbilstošu nosaukumu).

Hosts vienuma vērtības pirmapstrādes soļi

```
zabbix_export:
  (...)
  hosts:
    - host: 'MySQL serveris'
      (...)
      items:
        (...)
        - name: 'Binlog cache disk use'
          (...)
          preprocessing:
            - type: JSONPATH
              parameters:
                - $.Binlog_cache_disk_use
            - type: DISCARD_UNCHANGED_HEARTBEAT
              parameters:
                - 6h
          (...)
```

Element	Type	Apraksts
type	string	(obligāts) Vienuma vērtības pirmapstrādes 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		(obligāts) Saknes elements vienuma vērtības pirmapstrādes soļa parametriem.
parameter	string	Atsevišķs vienuma vērtības pirmapstrādes soļa parametrs.
error_handler	string	Darbības tips, ko izmanto pirmapstrādes soļa kļūmes gadījumā. Iespējamās vērtības:[1] ORIGINAL_ERROR (0, noklusējums), DISCARD_VALUE (1), CUSTOM_VALUE (2), CUSTOM_ERROR (3).
error_handler_params	string	Kļūdu apstrādātāja parametri.

Note:
Skatiet arī: Vienuma pirmapstrādes objekts (skatiet atbilstošo īpašību ar tādu pašu nosaukumu).

Hosts vienuma trigeri

```
zabbix_export:
  (...)
  hosts:
    - host: 'MySQL serveris'
      (...)
      items:
        (...)
        - name: Uptime
          (...)
          triggers:
            - expression: 'nodata(/MySQL serveris/mysql.uptime,30m)=1'
              name: 'MySQL: Neizdevās iegūt informācijas datus'
              event_name: 'MySQL: Neizdevās iegūt informācijas datus (vai nav datu 30 minūtes)'
              priority: INFO
              description: 'Zabbix pēdējo 30 minūšu laikā nav saņēmis nekādus datus par vienumiem.'
              dependencies:
                - name: 'MySQL: Pakalpojums nedarbojas'
                  expression: 'last(/MySQL serveris/mysql.ping["{$MYSQL.DSN}","{$MYSQL.USER}","{$MYSQL.PAS
              tags:
                - tag: scope
                  value: availability
            - expression: 'last(/MySQL serveris/mysql.uptime)<10m'
              name: 'MySQL: Pakalpojums ir restartēts'
              event_name: 'MySQL: Pakalpojums ir restartēts (darbības laiks < 10 min)'
              priority: INFO
              description: 'MySQL darbības laiks ir mazāks par 10 minūtēm.'
              tags:
                - tag: scope
                  value: notice
        (...)
```

Element	Type	Description
uuid	string	(required) Unikāls identifikators šim trigerim.
expression	string	(required) Trigeris izteiksme.

Element	Type	Description
recovery_mode	string	Pamats OK notikumu ģenerēšanai. Iespējamās vērtības:[1] EXPRESSION (0, noklusējums), RECOVERY_EXPRESSION (1), NONE (2).
recovery_expression	string	Trigeris atjaunošanas 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, ko izmantot notikumu korelācijai.
name	string	(required) Trigeris 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	Trigeris statuss. Iespējamās vērtības:[1] ENABLED (0, noklusējums), DISABLED (1).
priority	string	Trigeris smagums. Iespējamās vērtības:[1] NOT_CLASSIFIED (0, noklusējums), INFO (1), WARNING (2), AVERAGE (3), HIGH (4), DISASTER (5).
description	text	Trigeris apraksts.
type	string	Notikumu ģenerēšanas veids (viens problēmas notikums vai vairāki problēmas notikumi). Iespējamās vērtības:[1] SINGLE (0, noklusējums), MULTIPLE (1).
manual_close	string	Manuāla problēmas notikumu aizvēršana. Iespējamās vērtības:[1] NO (0, noklusējums), YES (1).
dependencies		Atkarību saknes elements.
name	string	(required) Atkarības trigeris nosaukums.
expression	string	(required) Atkarības trigeris izteiksme.
recovery_expression	string	Atkarības trigeris atjaunošanas izteiksme.
tags		Trigeris tagu saknes elements.
tag	string	(required) Taga nosaukums.
value	string	Taga vērtība.

Note:
Skatiet arī: Trigger object (skatiet attiecīgo īpašību ar atbilstošu nosaukumu).

Hostu zemā līmeņa atklāšanas noteikumi

```
zabbix_export:
  (...)
  hosts:
    - host: 'MySQL serveris'
      (...)
      discovery_rules:
        - name: 'Datubāzes atklāšana'
          key: 'mysql.db.discovery["{$MYSQL.DSN}","{$MYSQL.USER}","{$MYSQL.PASSWORD}"]'
          delay: 1h
          filter: (skatīt tabulu zemāk)
          description: 'Datubāzu skenēšana DBMS.'
          interface_ref: if1
          item_prototypes:
            - name: 'Datubāzes {#DATABASE} izmērs'
              key: 'mysql.db.size["{$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 zemā līmeņa atklāšanas noteikumu elementu ir tādi paši kā host vienumiem. Tālāk esošajā tabulā ir aprakstīti tie elementi, kas atšķiras no host 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	(obligāts) Zemā līmeņa atklāšanas noteikuma atslēga.
filter		Saknes elements hostu zemā līmeņa atklāšanas noteikumu filtriem.
lifetime	string	Laika periods (izmantojot sekundes, laika sufiksu vai lietotāja makro), 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 makro), 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 host vienumu prototipu elementiem, kas ir tādi paši kā host vienumiem.
trigger_prototypes		Saknes elements host trigeru prototipu elementiem, kas ir tādi paši kā host vienumu trigeriem.
graph_prototypes		Saknes elements host grafiku prototipiem, kas ir tādi paši kā host grafikiem.
host_prototypes		Saknes elements hostu prototipiem, kas ir tādi paši kā hostiem.
parent_discovery_rule		Saknes elements zemā līmeņa atklāšanas noteikuma prototipa vecākajam zemā līmeņa atklāšanas noteikumam (vai noteikuma prototipam). Šis īpašums norāda, ka tas ir LLD noteikuma prototips, tiešs norādītā noteikuma/noteikuma prototipa bērnelements.
key	string	(obligāts) Vecākā zemā līmeņa atklāšanas noteikuma (vai noteikuma prototipa) atslēga.
master_item	string	(obligāts DEPENDENT noteikumiem) Saknes elements atkarīgā noteikuma galvenajam vienumam.
lld_macro_paths		Saknes elements zemā līmeņa atklāšanas noteikuma makro ceļiem.
lld_macro	string	(obligāts) Zemā līmeņa atklāšanas noteikuma makro nosaukums.
path	string	(obligāts) Vērtības atlasītājs, kas tiks piešķirta attiecīgajam makro.
preprocessing		Saknes elements zemā līmeņa atklāšanas noteikuma vērtības pirmapstrādei.
step		Saknes elements zemā līmeņa atklāšanas noteikuma vērtības pirmapstrādes soļa elementiem, kas ir tādi paši kā host vienuma vērtības pirmapstrādes soļi, izņemot to, ka pieejamo vērtību ir mazāk. Skatīt arī: LLD noteikuma pirmapstrādes objekts.
overrides		Saknes elements zemā līmeņa atklāšanas noteikuma pārrakstīšanas noteikumiem.
name	string	(obligāts) Unikāls pārrakstīšanas nosaukums.
step	string	(obligāts) Unikāls pārrakstīšanas kārtas numurs.
stop	string	Pārtraukt nākamo pārrakstīšanu apstrādi, ja atbilst.
filter		Saknes elements zemā līmeņa atklāšanas noteikuma pārrakstīšanas noteikumu filtra elementiem, kas ir tādi paši kā hostu zemā līmeņa atklāšanas noteikumu filtriem.
operations		Saknes elements hostu zemā līmeņa atklāšanas noteikumu pārrakstīšanas darbībām.

Note:
Skatīt arī: LLD noteikuma objekts (skatiet attiecīgo īpašību ar atbilstošu nosaukumu).

Hosts zema līmeņa atklāšanas noteikuma filtri

```
zabbix_export:
  (...)
  hosts:
    - host: 'MySQL serveris'
      (...)
      discovery_rules:
        - name: 'Datubāzes atklāšana'
          (...)
          filter:
            evaltype: AND
            conditions:
              - macro: '{#DATABASE}'
                value: '{$MYSQL.DBNAME.MATCHES}'
              - macro: '{#DATABASE}'
                value: '{$MYSQL.DBNAME.NOT_MATCHES}'
                operator: NOT_MATCHES_REGEX
          (...)
```

Element	Type	Description
evaltype	string	Filtra nosacījumu izvērtēšanas metodes pārrakstīšana. Iespējamās vērtības:[1] AND_OR (0, noklusējuma), AND (1), OR (2), FORMULA (3).
formula	string	Pielāgota aprēķina formula filtra nosacījumiem.
conditions		Filtra nosacījumu saknes elements.
macro	string	(obligāts) Zema līmeņa atklāšanas makro nosaukums, uz kura veikt pārbaudi.
value	string	Vērtība, ar kuru salīdzināt.
operator	string	Nosacījuma operators. Iespējamās vērtības:[1] MATCHES_REGEX (8, noklusējuma), NOT_MATCHES_REGEX (9).
formulaid	string	(obligāts) Patvaļīgs unikāls ID, kas tiek izmantots, lai atsauktos uz nosacījumu no pielāgotās izteiksmes. Var saturēt tikai lielos burtus. ID lietotājam ir jādefinē, modificējot filtra nosacījumus, taču, pieprasot tos vēlāk, tas tiks ģenerēts no jauna.

Note:
Skatiet arī: LLD noteikuma filtra objekts (skatiet atbilstošo īpašību ar tādu pašu nosaukumu).

Hosts zema līmeņa atklāšanas noteikuma pārrakstīšanas operācijas

Element	Type	Description
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	Pārrakstīšanas 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 pārrakstīšanas nosacījuma operatoram.
status	string	Objekta statuss pēc pārrakstīšanas operācijas.
discover	string	Vai objekts tiek pievienots atklāšanas rezultātā.
delay	string	Atjaunināšanas intervāls, kas iestatīts vienuma prototipam pēc pārrakstīšanas operācijas.
history	string	Vēstures glabāšanas periods, kas iestatīts vienuma prototipam pēc pārrakstīšanas operācijas.
trends	string	Tendenču glabāšanas periods, kas iestatīts vienuma prototipam pēc pārrakstīšanas operācijas.
severity	string	Trigeris prototipa smaguma pakāpe, kas iestatīta pēc pārrakstīšanas operācijas.
tags		Saknes elements tagiem, kas iestatīti objektam pēc pārrakstīšanas operācijas.
tag	string	(obligāts) Taga nosaukums.

Element	Type	Description
value	string	Taga vērtība.
templates		Saknes elements veidnēm, kas saistītas ar host prototipu pēc pārrakstīšanas operācijas.
name	string	(obligāts) Veidnes nosaukums.
inventory_mode	string	Host prototipa inventarizācijas režīms, kas iestatīts pēc pārrakstīšanas operācijas.

Note:
Skatiet arī: LLD noteikuma pārrakstīšanas operācijas objekts (skatiet atbilstošo īpašību ar tādu pašu nosaukumu).

Hosts web scenāriji

Element	Type	Description
uuid	string	(required) Unikāls identifikators šim web scenārijam.
name	string	(required) Web scenārija nosaukums.
delay	string	Izpildes biežums (sekundēs, laika sufikss vai lietotāja makro) web scenārijam. Default: 1m.
attempts	integer	Mēģinājumu skaits web scenārija soļu izpildei. Iespējamās vērtības: 1-10 (default: 1).
agent	string	Klienta aģents. Zabbix izliksies par atlasīto pārlūkprogrammu. Tas ir noderīgi, ja vietne dažādām pārlūkprogrammām atgriež atšķirīgu saturu. Default: Zabbix.
http_proxy	string	Starpniekserveris, kas tiks izmantots web scenārijam, norādīts kā: `http://[username[:password]@]proxy.example.com[:port]`
variables		Saknes elements web scenārija mainīgajiem, ko var izmantot scenārija soļos.
name	string	(required) Mainīgā nosaukums.
value	text	(required) Mainīgā vērtība.
headers		Saknes elements HTTP galvenēm, kas tiks nosūtītas pieprasījuma izpildes laikā. Galvenes jāuzskaita, izmantojot to pašu sintaksi, kāda tām būtu HTTP protokolā.
name	string	(required) Galvenes nosaukums.
value	text	(required) Galvenes vērtība.
status	string	Web scenārija statuss. Iespējamās vērtības:[1] ENABLED (0, default), DISABLED (1).
authentication	string	Autentifikācijas metode. Iespējamās vērtības:[1] NONE (0, default), BASIC (1), NTLM (2), Kerberos (3) vai Digest (4).
http_user	string	Lietotājvārds, ko izmanto BASIC (HTTP), NTLM, Kerberos vai Digest autentifikācijai.
http_password	string	Parole, ko izmanto BASIC (HTTP), NTLM, Kerberos vai Digest autentifikācijai.
verify_peer	string	Pārbaudīt tīmekļa servera SSL sertifikātu. Iespējamās vērtības:[1] NO (0, default), YES (1).
verify_host	string	Pārbaudīt, vai tīmekļa servera sertifikāta lauks *Common Name* vai *Subject Alternate Name* atbilst. Iespējamās vērtības:[1] NO (0, default), YES (1).
ssl_cert_file	string	SSL sertifikāta faila nosaukums, ko izmanto klienta autentifikācijai (tam jābūt PEM formātā).
ssl_key_file	string	SSL privātās atslēgas faila nosaukums, ko izmanto klienta autentifikācijai (tam jābūt PEM formātā).
ssl_key_password	string	SSL privātās atslēgas faila parole.
steps		(required) Saknes elements hosts web scenārija soļiem.
tags		Saknes elements web scenārija tagiem.
tag	string	(required) Taga nosaukums.
value	string	Taga vērtība.

Note:
Skatiet arī: Web scenārija objekts (skatiet attiecīgo īpašību ar atbilstošu nosaukumu).

Hosts tīmekļa scenārija soļi

Element	Type	Description
name	string	(required) Tīmekļa scenārija soļa nosaukums.
url	string	(required) Uzraudzības URL.
query_fields		Vaicājuma parametru saknes elements (HTTP lauku masīvs, kas tiks pievienots URL, veicot pieprasījumu).
name	string	(required) Vaicājuma parametra nosaukums.
value	string	Vaicājuma parametra vērtība.
posts		HTTP POST mainīgo saknes elements (virkne (raw post data) vai HTTP lauku masīvs (form field data)).
name	string	(required) POST lauka nosaukums.
value	string	(required) POST lauka vērtība.
variables		Soļa līmeņa mainīgo (makro) saknes elements, kas jāpiemēro pēc šī soļa. Ja mainīgā vērtībai ir prefikss 'regex:', tad tās vērtība tiek iegūta no datiem, ko atgriež šis solis, saskaņā ar regulārās izteiksmes paraugu, kas seko prefiksam 'regex:'
name	string	(required) Mainīgā nosaukums.
value	text	(required) Mainīgā vērtība.
headers		HTTP galvenes saknes elements, kas jānosūta, veicot pieprasījumu.
name	string	(required) Galvenes nosaukums.
value	text	(required) Galvenes vērtība.
follow_redirects	string	Sekot HTTP novirzēm. Iespējamās vērtības:[1] NO (0), YES (1, default).
retrieve_mode	string	HTTP atbildes iegūšanas režīms. Iespējamās vērtības:[1] BODY (0, default), HEADERS (1), BOTH (2).
timeout	string	Soļa izpildes noildze (sekundēs, laika sufikss vai lietotāja makro). Noklusējums: 15s.
required	string	Teksts, kam jābūt atbildē (ja tukšs, tiek ignorēts).
status_codes	string	Ar komatiem atdalīts pieņemamo HTTP statusa kodu saraksts (piem., 200-201,210-299; ja tukšs, tiek ignorēts).

Note:
Skatiet arī: Web scenario step object (skatiet atbilstošo īpašību ar sakritīgu nosaukumu).

Hostu grafiki

```
zabbix_export:
  (...)
  graphs:
    - name: 'MySQL: Bandwidth'
      graph_items: (skatīt tabulu zemāk)
    (...)
```

Element	Type	Description
uuid	string	Grafika unikālais identifikators.
name	string	(obligāts) Grafika nosaukums.
width	integer	Grafika platums pikseļos. Tiek izmantots priekšskatījumam un pīrāga/izkliedētajiem grafikiem. Iespējamās vērtības:[1] 20-65535 (noklusējums: 900).
height	integer	Grafika augstums pikseļos. Tiek izmantots priekšskatījumam un pīrāga/izkliedētajiem grafikiem. Iespējamās vērtības:[1] 20-65535 (noklusējums: 900).
yaxismin	double	Y ass minimālā vērtība. Atbalstīts Y ass FIXED minimālajai vērtībai. Noklusējums: 0.
yaxismax	double	Y ass maksimālā vērtība. Atbalstīts X ass FIXED minimālajai vērtībai. Noklusējums: 0.
show_work_period	string	Izcelt nestrādājošās stundas. Atbalstīts NORMAL un STACKED grafikiem. Iespējamās vērtības:[1] NO (0), YES (1, noklusējums).

Element	Type	Description
show_triggers	string	Parādīt vienkāršu 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	Parādīt 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).
percent_left	double	Parādīt procentiles līniju kreisajai asij. Atbalstīts NORMAL grafikiem. Noklusējums: 0.
percent_right	double	Parādīt 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 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 informācijai.
host	string	(obligāts) Vienuma hosts.
key	string	(obligāts) Vienuma atslēga.
graph_items		(obligāts) Saknes elements host grafika vienumiem.

Note:
Skatīt arī: Graph object (skatiet atbilstošo īpašību ar tādu pašu nosaukumu).

Host graph items

```
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	Tips	Apraksts
sortorder	integer	Zīmēšanas secība. Mazākā vērtība tiek zīmēta pirmā. Var izmantot, lai zīmētu līnijas vai apgabalus aiz citas līnijas vai apgabala (vai to priekšā).

Elements	Tips	Apraksts
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, heksadecimāls).
yaxisside	string	Grafika puse, kurā tiks zīmēta grafika vienuma Y ass mērogs. Atbalstīts NORMAL un STACKED grafikiem. Iespējamās vērtības:[1] LEFT (0, noklusējums), RIGHT (1).
calc_fnc	string	Dati, ko zīmēt, ja vienumam ir 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/izvērstiem sektoru grafikiem).
type	string	Grafika vienuma tips. Iespējamās vērtības:[1] SIMPLE (0, noklusējums), GRAPH_SUM (2; vienuma vērtība attēlo visu sektoru; atbalstīts sektoru/izvērstiem sektoru grafikiem).
item		(obligāts) Atsevišķs vienums.
host	string	(obligāts) Vienuma hosts.
key	string	(obligāts) Vienuma atslēga.

Note:
Skatiet arī: Graph item object (skatiet atbilstošo īpašību ar tādu pašu nosaukumu).

Hosts vērtību kartes

```
zabbix_export:
  (...)
  hosts:
    - host: 'MySQL server'
      (...)
      valuemaps:
        - name: Example value map
          mappings:
            - value: '1'
              newvalue: Example value
  (...)
```

Element	Type	Description
uuid	string	(required) Unikāls šīs vērtību kartes identifikators.
name	string	(required) Vērtību kartes nosaukums.
mapping		Kartējumu saknes elements.
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	(required) Vērtība, uz kuru sākotnējā vērtība tiek kartēta.

Piezīmes [1] API veselo skaitļu vērtības iekavās, piemēram, ENABLED (0), ir minētas tikai uzziņai. Papildinformāciju skatiet saistītajā API objekta lapā tabulas ierakstā vai katras sadaļas beigās.

Tīkla kartes

Pārskats

Tīkla karte eksports satur:

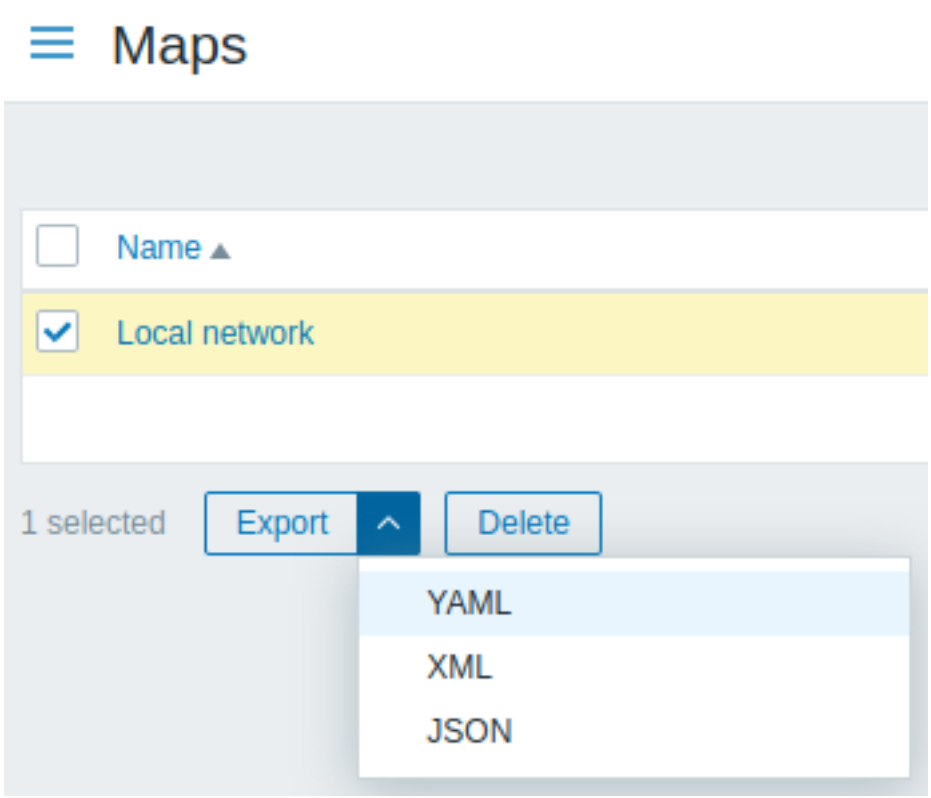
- Visus saistītos attēlus
- Kartes struktūru (visi kartes iestatījumi, visi tajā iekļautie elementi ar to iestatījumiem, kartes saites un kartes saišu statusa indikatori)

Warning:
Netiek eksportētas nevienas hostu grupas, hosts, trigeri, citas kartes vai citi elementi, kas var būt saistīti ar eksportēto karti. Tāpēc, ja trūkst vismaz viens no elementiem, uz kuru karte atsaucas, tās importēšana neizdosies.

Eksportēšana

Lai eksportētu tīkla kartes, rīkojieties šādi:

1. Dodieties uz *Monitoring* > *Maps*.
2. Atzīmējiet eksportējamās tīkla kartes izvēles rūtiņas.
3. Noklikšķiniet uz *Export* zem saraksta, lai eksportētu noklusējuma YAML formātā, vai noklikšķiniet uz bultiņas, lai atlasītu citu eksporta formātu.



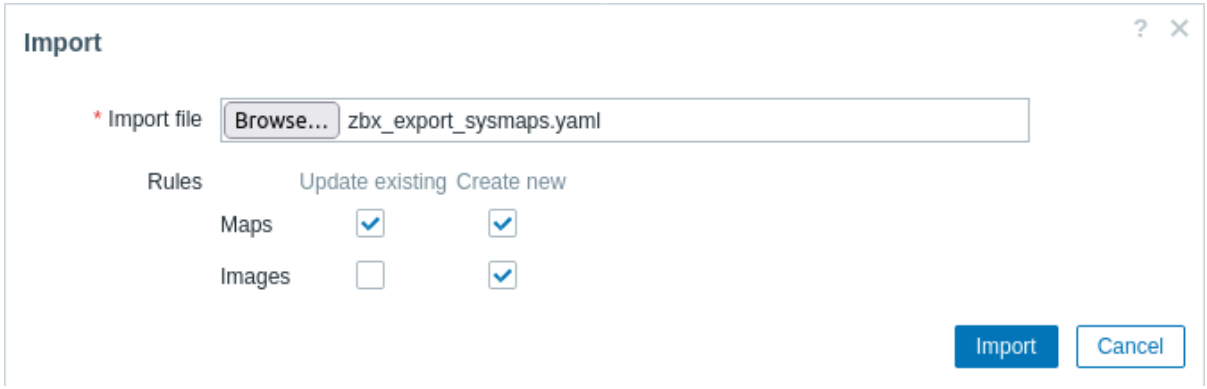
Atkarībā no atlasītā formāta kartes tiek eksportētas uz lokālu failu ar noklusējuma nosaukumu:

- `zabbix_export_maps.yaml` - YAML eksportā;
- `zabbix_export_maps.xml` - XML eksportā;
- `zabbix_export_maps.json` - JSON eksportā.

Importēšana

Lai importētu tīkla kartes, rīkojieties, kā norādīts tālāk:

1. Atveriet *Monitoring* > *Maps*.
2. Augšējā labajā stūrī noklikšķiniet uz *Import*.
3. Atlasiet importēšanas failu.
4. Atzīmējiet to importēšanas noteikumu izvēles rūtiņas, kurus vēlaties lietot (skatiet tālāk).
5. Konfigurācijas formas apakšējā labajā stūrī noklikšķiniet uz *Import*.



Importēšanas noteikumi:

Noteikums	Apraksts
Update existing	Esošās kartes tiks atjauninātas, izmantojot importēšanas faila datus. Pretējā gadījumā tās netiks atjauninātas.
Create new	Izmantojot importēšanas faila datus, tiks izveidotas jaunas kartes. 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.

Importēšanas laikā tīkla karšu izmaiņas nevar pārskatīt. Importēšanas veiksmes vai kļūmes ziņojums tiks parādīts lietotāja saskarnē.

Warning:
Ja aizstāsiet esošu attēlu, tas ietekmēs visas kartes, kurās tiek izmantots šis attēls.

Eksporta formāts

Eksportēt uz YAML:

```
zabbix_export:
  version: '8.0'
  images:
    - name: Zabbix_server_3D_(128)
      imagetype: '1'
      encodedImage: iVBOR...5CYII=
  maps:
    - name: 'Local network'
      width: '680'
      height: '200'
      label_type: '0'
      label_location: '0'
      highlight: '1'
      expandproblem: '1'
      markelements: '1'
      show_unack: '0'
      severity_min: '0'
      show_suppressed: '0'
      grid_size: '50'
      grid_show: '1'
      grid_align: '1'
      label_format: '0'
      label_type_host: '2'
      label_type_hostgroup: '2'
      label_type_trigger: '2'
      label_type_map: '2'
      label_type_image: '2'
      label_string_host: ''
      label_string_hostgroup: ''
```

```
    label_string_trigger: ''
    label_string_map: ''
    label_string_image: ''
    expand_macros: '1'
    background: {  }
    iconmap: {  }
    urls: {  }
    selements:
      - elementtype: '0'
        elements:
          - host: 'Zabbix server'
        label: |
          {HOST.NAME}
          {HOST.CONN}
        label_location: '0'
        x: '111'
        'y': '61'
        elementsubtype: '0'
        areatype: '0'
        width: '200'
        height: '200'
        viewtype: '0'
        use_iconmap: '0'
        selementid: '1'
        icon_off:
          name: Zabbix_server_3D_(128)
        icon_on: {  }
        icon_disabled: {  }
        icon_maintenance: {  }
        urls: {  }
        evaltype: '0'
    shapes:
      - type: '0'
        x: '0'
        'y': '0'
        width: '680'
        height: '15'
        text: '{MAP.NAME}'
        font: '9'
        font_size: '11'
        font_color: '000000'
        text_halign: '0'
        text_valign: '0'
        border_type: '0'
        border_width: '0'
        border_color: '000000'
        background_color: ''
        zindex: '0'
    lines: {  }
    links: {  }
```

Eksportētie elementi

Eksportētie elementi ir paskaidroti tabulā zemāk.

Elements	Tips	Apraksts
images		Saknes elements attēliem.
name	string	Unikāls attēla nosaukums.
imagetype	integer	Attēla tips. Iespējamās vērtības: 1 - Attēls; 2 - Fons.
encodedImage	string	Base64 kodēts attēls.

Elements	Tips	Apraksts
maps		Saknes elements kartēm.

Kartes

Element	Type	Apraksts
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 - hosts IP adrese; 2 - Elementa nosaukums; 3 - Tikai statuss; 4 - Nekas.
label_location	integer	Kartes elementa etiķetes atrašanās vieta pēc noklusējuma. Iespējamās vērtības: 0 - Apakšā; 1 - Pa kreisi; 2 - Pa labi; 3 - Augšā.
highlight	integer	Iespējot ikonu izcelšanu aktīviem trigeriem un host 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, kuru statuss nesen ir mainījies. 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 trigeru nopietnība, kas pēc noklusējuma tiek rādīta kartē. Iespējamās vērtības: 0 - Neklasificēts; 1 - Informācija; 2 - Brīdinājums; 3 - Vidējs; 4 - Augsts; 5 - Katastrofāls.
show_suppressed	integer	Rādīt problēmas, kas citādi būtu slēptas (nerādītas) host 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.

Element	Type	Apraksts
grid_align	integer	Automātiski izlīdzināt ikonas kartes konfigurācijā. Iespējamās vērtības: 0 - Jā; 1 - Nē.
label_format	integer	Izmantot paplašinātu etiķešu konfigurāciju. Iespējamās vērtības: 0 - Nē; 1 - Jā.
label_type_host	integer	Rādīt etiķeti kā host etiķeti. Atbalstīts, ja `label_format` ir iestatīts uz 1. Iespējamās vērtības: 0 - Etiķete; 1 - hosts IP adrese; 2 - Elementa nosaukums; 3 - Tikai statuss; 4 - Nekas; 5 - Pielāgota etiķete.
label_type_hostgroup	integer	Rādīt etiķeti kā host 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 trigeris 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 host elementiem. Atbalstīts, ja `label_type_host` ir iestatīts uz 5.
label_string_hostgroup	string	Pielāgota etiķete host grupas elementiem. Atbalstīts, ja `label_type_hostgroup` ir iestatīts uz 5.
label_string_trigger	string	Pielāgota etiķete trigeris 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 makro kartes konfigurācijas etiķetēs. Iespējamās vērtības: 0 - Nē; 1 - Jā.

Element	Type	Apraksts
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 - Host grupa; 4 - Attēls.
selements		Saknes elements map selements.
shapes		Saknes elements kartes figūrām.
type	integer	Figūras tips. Iespējamās vērtības: 0 - Taisnstūris; 1 - Elipse.
x	integer	Figūras X koordinātas pikseļos.
y	integer	Figūras Y koordinātas pikseļos.
width	integer	Figūras platums.
height	integer	Figūras augstums.
text	string	Teksts figūras 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 lielums pikseļos.
font_color	string	Fonta krāsa heksadecimālā kodā.
text_halign	integer	Teksta horizontālā izlīdzināšana. Iespējamās vērtības: 0 - Centrā; 1 - Pa kreisi; 2 - Pa labi.
text_valign	integer	Teksta vertikālā izlīdzināšana. Iespējamās vērtības: 0 - Vidū; 1 - Augšā; 2 - Apakšā.
border_type	integer	Figūras apmales tips. Iespējamās vērtības: 0 - Nav; 1 - Trekns līnijas stils; 2 - Punktota līnija; 3 - Svītrota līnija.
border_width	integer	Apmales platums pikseļos.
border_color	string	Apmales krāsa heksadecimālā kodā.

Element	Type	Apraksts
background_color	string	Fona (aizpildījuma) krāsa heksadecimālā kodā.
zindex	integer	Vērtība visu figūru un līniju kārtošanai (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 - Trekns līnijas stils; 2 - Punktota līnija; 3 - Svītrota līnija.
line_width	integer	Līnijas platums pikseļos.
line_color	string	Līnijas krāsa heksadecimālā kodā.
zindex	integer	Vērtība visu figūru un līniju kārtošanai (z-index).
links		Saknes elements saitēm starp kartes elementiem.
drawtype	integer	Saites stils. Iespējamās vērtības: 0 - Līnija; 2 - Trekns līnijas stils; 3 - Punktota līnija; 4 - Svītrota līnija.
color	string	Saites krāsa (6 simboli, heksadecimāli).
label	string	Saites etiķete.
selementid1	id	Viena elementa ID, ko savienot.
selementid2	id	Otra elementa ID, ko savienot.
linktriggers		Saknes elements saites statusa indikatoriem.

Note:
Skatiet arī: Kartes objekts (skatiet atbilstošo īpašību ar tādu pašu nosaukumu).

Kartes elementi

Element	Type	Description
elementtype	integer	Kartes elementa tips. Iespējamās vērtības: 0 - Hosts; 1 - Karte; 2 - Trigeris; 3 - Hostu grupa; 4 - Attēls.
elements		Zabbix entītiju (hosts, hostu grupa, karte u. c.) saknes elements, kas attēlotas kartē. Visām entītijām ir attiecīgais elements (`host`, u. tml.; skatiet, piemēram, Eksporta formāts).
label	string	Ikonas etiķete.
label_location	integer	Etiķetes novietojums. Iespējamās vērtības: -1 - Izmantot kartes noklusējumu; 0 - Apakšā; 1 - Pa kreisi; 2 - Pa labi; 3 - Augšā.
x	integer	Atrašanās vieta X asī.
y	integer	Atrašanās vieta Y asī.

Element	Type	Description
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 - Vienmērīgi izvietot apgabalā.
use_iconmap	integer	Šim elementam izmantot ikonu kartējumu. Attiecas tikai tad, ja ikonu kartējums ir aktivizēts kartes līmenī. Iespējamās vērtības: 0 - Nē; 1 - Jā.
selementid	id	Unikāls elementa ieraksta ID.
icon_off		Saknes elements attēlam, ko izmantot, ja elements ir `OK` stāvoklī.
name	string	Unikāls attēla nosaukums.
icon_on		Saknes elements attēlam, ko izmantot, ja elements ir `Problem` stāvoklī.
name	string	Unikāls attēla nosaukums.
icon_disabled		Saknes elements attēlam, ko izmantot, ja elements ir atspējots.
name	string	Unikāls attēla nosaukums.
icon_maintenance		Saknes elements attēlam, ko izmantot, ja elements ir apkopē.
name	string	Unikāls attēla nosaukums.
urls		Saknes elements URL, ko izmanto kartes vai katra kartes elementa vajadzībām.
name	string	Saites nosaukums.
url	string	Saites URL.
evaltype	integer	Tagu novērtēšanas tips.
tags		Saknes elements problēmu tagiem (host un hostu grupas elementiem). Ja tagi ir norādīti, kartē tiks rādītas tikai problēmas ar šiem tagiem.
tag	string	Taga nosaukums.
value	string	Taga vērtība.
operator	integer	Operators.

Note:
Skatiet arī: Kartes elementa objekts (skatiet attiecīgo īpašību ar atbilstošu nosaukumu).

Kartes saites statusa indikatori

Elements	Type	Description
drawtype	integer	Saites stils, kad trigeris ir stāvoklī "Problem". Iespējamās vērtības: 0 - līnija; 2 - trekna līnija; 3 - punktēta līnija; 4 - pārtraukta līnija.
color	string	Saites krāsa (6 simboli, hex), kad trigeris ir stāvoklī "Problem".
trigger		Trigerim paredzētais saknes elements, ko izmanto saites statusa norādīšanai.
description	string	Trigeris nosaukums.
expression	string	Trigeris izteiksme.
recovery_expression	string	Trigeris atjaunošanas izteiksme.

Note:
Skatiet arī: Map link trigger object (skatiet atbilstošo īpašību ar tādu pašu nosaukumu).

Multivides tipi

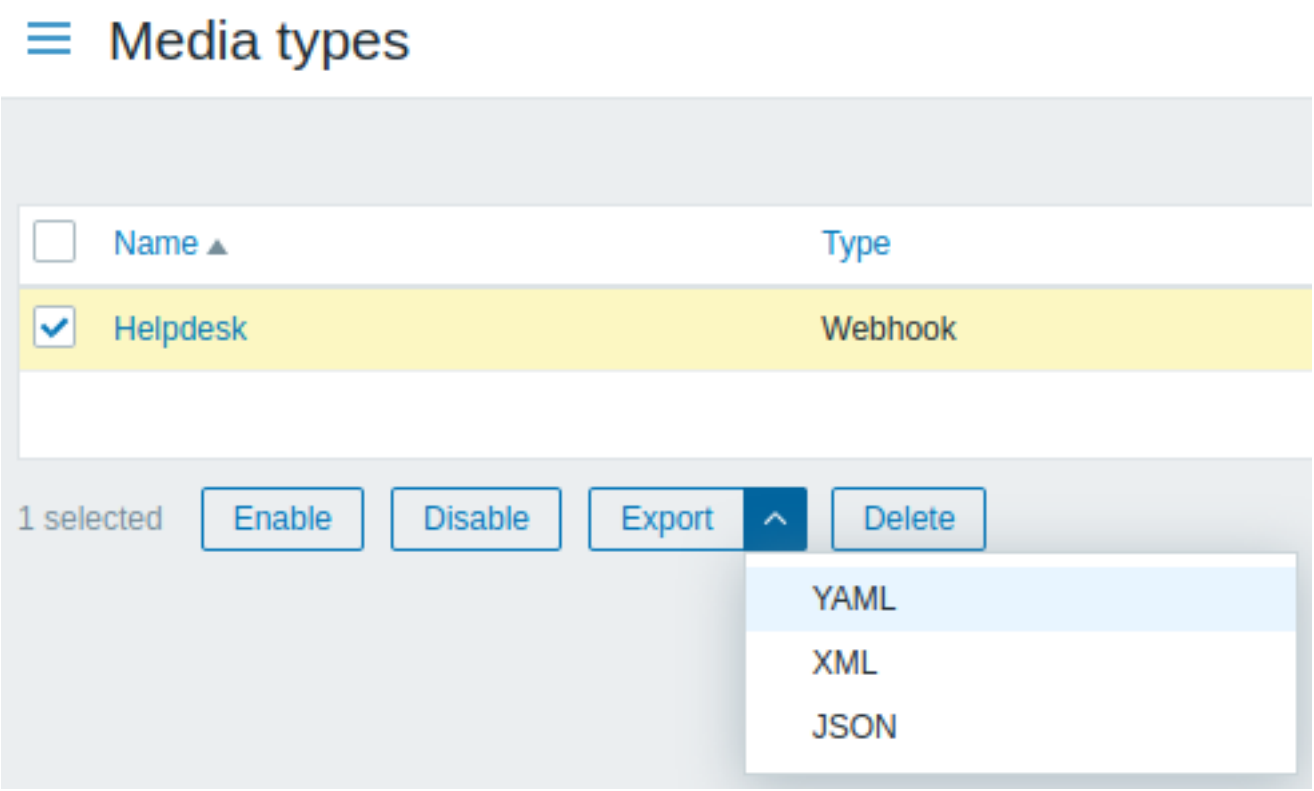
Pārskats

Multivides tipi tiek eksportēti kopā ar visiem saistītajiem objektiem un objektu attiecībām.

Eksportēšana

Lai eksportētu mediju tipus, rīkojieties šādi:

1. Dodieties uz *Alerts* > *Media types*.
2. Atzīmējiet eksportējamo mediju tipu izvēles rūtiņas.
3. Noklikšķiniet uz *Export* zem saraksta, lai eksportētu noklusējuma YAML formātā, vai noklikšķiniet uz bultiņas, lai atlasītu citu eksporta formātu.



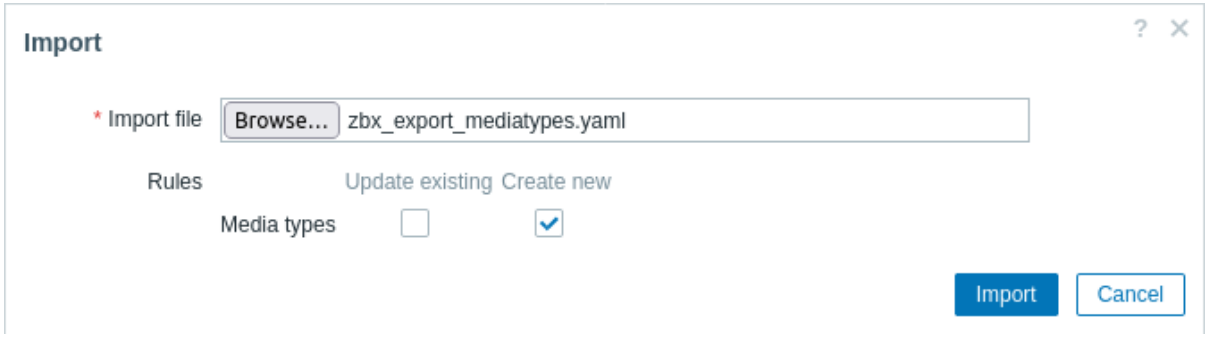
Atkarībā no atlasītā formāta mediju tipi tiek eksportēti uz lokālu failu ar noklusējuma nosaukumu:

- `zabbix_export_mediatypes.yaml` - YAML eksportā;
- `zabbix_export_mediatypes.xml` - XML eksportā;
- `zabbix_export_mediatypes.json` - JSON eksportā.

Importēšana

Lai importētu multivides tipus, rīkojieties, kā norādīts tālāk:

1. Atveriet *Alerts* > *Media types*.
2. Augšējā labajā stūrī noklikšķiniet uz *Import*.
3. Atlasiet importēšanas failu.
4. Atzīmējiet to importēšanas noteikumu izvēles rūtiņas, kurus vēlaties lietot (skatiet tālāk).
5. Konfigurācijas formas apakšējā labajā stūrī noklikšķiniet uz *Import*.



Importēšanas noteikumi:

Rule	Description
Update existing	Atzīmējiet šo izvēles rūtiņu, lai atjauninātu esošos multivides tipus, izmantojot importēšanas faila datus.
Create new	Atzīmējiet šo izvēles rūtiņu, lai izveidotu jaunus multivides tipus, izmantojot importēšanas faila datus. Ja importēšanas failā ir multivides tips, kas jau pastāv (piemēram, *Email*), jauns tips netiks izveidots, un esošais multivides tips arī netiks atjaunināts.

Importēšanas laikā nevar pārskatīt multivides tipu izmaiņas. Lietotāja saskarnē tiks parādīts importēšanas veiksmes vai kļūmes ziņojums.

Eksporta formāts

Eksportēt uz 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 'Nekorekts "event_source" parametrs: "' + params.event_source + '".\nTam jābūt 0-3.'
          }

          if (params.event_value !== '0' && params.event_value !== '1'
              && (params.event_source === '0' || params.event_source === '3')) {
              throw 'Nekorekts "event_value" parametrs: ' + params.event_value + '\nTam jābūt 0 vai 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" jābūt -2..2';
          }

          if (params.event_source === '0' && isNaN(params.triggerid)) {
              throw 'lauks "triggerid" nav skaitlis';
          }
```

```
if (isNaN(params.eventid)) {
    throw 'lauks "eventid" nav skaitlis';
}

if (typeof params.message !== 'string' || params.message.trim() === '') {
    throw 'lauks "message" nevar būt tukšs';
}

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 'lauks "retry" jābūt skaitlim, kura vērtība ir vismaz 30, ja "priority" ir iesta
    }

    if (isNaN(params.expire) || params.expire > 10800) {
        throw 'lauks "expire" jābūt skaitlim, kura vērtība ir ne vairāk kā 10800, ja "priority
    }

    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 rokasgrāmatu šeit: https://git.zabbix.com/projects/ZBX/repos/zabbix/br

    Iestatiet token parametru ar savu Pushover lietotnes atslēgu.
    Piešķirot Pushover mediju Zabbix lietotājam, laukā "send to" pievienojiet lietotāja atslēgu.
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}
      Smagums: {EVENT.SEVERITY}
      Operatīvie dati: {EVENT.OPDATA}
      Sākotnējās problēmas ID: {EVENT.ID}
      {TRIGGER.URL}
  - event_source: TRIGGERS
    operation_mode: RECOVERY
    subject: 'Atrisināts {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}
      Smagums: {EVENT.SEVERITY}
      Sākotnējās problēmas ID: {EVENT.ID}
      {TRIGGER.URL}
  - event_source: TRIGGERS
    operation_mode: UPDATE
    subject: 'Atjaunināta problēma {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}, apstiprināts: {EVEN
  - event_source: DISCOVERY
    operation_mode: PROBLEM
    subject: 'Atklāšana: {DISCOVERY.DEVICE.STATUS} {DISCOVERY.DEVICE.IPADDRESS}'
    message: |
      Atklāšanas noteikums: {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 pakalpojuma nosaukums: {DISCOVERY.SERVICE.NAME}
      Ierīces pakalpojuma ports: {DISCOVERY.SERVICE.PORT}
      Ierīces pakalpojuma statuss: {DISCOVERY.SERVICE.STATUS}
      Ierīces pakalpojuma darbības laiks: {DISCOVERY.SERVICE.UPTIME}
  - event_source: AUTOREGISTRATION
    operation_mode: PROBLEM
    subject: 'Automātiska reģistrācija: {HOST.HOST}'
    message: |
      Hosts nosaukums: {HOST.HOST}
      Hosts IP: {HOST.IP}
```

```
Aģenta ports: {HOST.PORT}
```

Eksportētie elementi

Eksportētie elementi ir izskaidroti tabulā zemāk.

Element	Type	Description
name	string	(obligāts) Mediju tipa nosaukums.
type	string	(obligāts) Mediju tipa izmantotais transports. Iespējamās vērtības:[1] EMAIL (0), SMS (1), SCRIPT (2), WEBHOOK (4).
status	string	Vai mediju tips ir iespējots. Iespējamās vērtības:[1] ENABLED (0, noklusējums), DISABLED (1)
max_sessions	integer	Maksimālais brīdinājumu skaits, ko var apstrādāt paralēli. Iespējamās vērtības SMS:[1] 1 (noklusējums). Iespējamās vērtības citiem mediju tipiem:[1] 0-100 (kur 0 - neierobežots).
attempts	integer	Maksimālais mēģinājumu skaits brīdinājuma nosūtīšanai. Iespējamās vērtības:[1] 1-10 (noklusējums: 3).
attempt_interval	string	Intervāls (izmantojot sekundes vai laika sufiksu) starp atkārtotiem mēģinājumiem. Iespējamās vērtības:[1] 0-60s (noklusējums: 10s).
description	string	Mediju tipa apraksts.
message_templates		Saknes elements mediju tipa ziņojumu veidnēm.
event_source	string	(obligāts) Notikuma avots. Iespējamās vērtības:[1] TRIGGERS (0), DISCOVERY (1), AUTOREGISTRATION (2), INTERNAL (3), SERVICE (4).
operation_mode	string	Darbības režīms. Iespējamās vērtības:[1] PROBLEM (0), RECOVERY (1), UPDATE (2).
subject	string	Ziņojuma temats.
message	string	Ziņojuma saturs.

Note:
Skatiet arī: Mediju tipa objekts (skatiet attiecīgo īpašību ar atbilstošu nosaukumu).

Email

Tālāk norādītie papildu elementi tiek eksportēti tikai *Email* mediju tipam.

Element	Type	Description
provider	string	E-pasta pakalpojuma sniedzējs.
smtp_server	string	SMTP serveris.
smtp_port	integer	SMTP servera ports, ar kuru 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	SSL verifikācija SMTP resursdatoram. Iespējamās vērtības:[1] NO (0, noklusējums), YES (1).
smtp_verify_peer	string	SSL verifikācija SMTP vienaudžam. 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 OAuth autorizāciju.
client_id	string	Klienta identifikators, kas reģistrēts OAuth autorizācijas serverī.
authorization_url	string	OAuth URL ar parametriem piekļuves un atsvaidzināšanas žetonu iegūšanai.
token_url	string	OAuth URL autorizācijas žetona apmaiņai pret piekļuves un atsvaidzināšanas žetoniem.
message_format	string	Ziņojuma formāts. Iespējamās vērtības:[1] TEXT (0), HTML (1, noklusējums).

Note:
Skatiet arī: Media type object (skatiet attiecīgo īpašību ar atbilstošu nosaukumu).

SMS

Tālāk norādītie papildu elementi tiek eksportēti tikai *SMS* multivides tipam.

Elements	Type	Apraksts
gsm_modem	string	(obligāts) GSM modema seriālās ierīces nosaukums.

Note:
Skatiet arī: Multivides tipa objekts (skatiet attiecīgo īpašību ar atbilstošu nosaukumu).

Skripts

Tālāk norādītie papildu elementi tiek eksportēti tikai *Skripts* multivides tipam.

Elements	Tips	Apraksts
skripta nosaukums	string	(obligāts) Skripta nosaukums.
parametri		Saknes elements skripta parametriem.
sortorder	string	(obligāts) Skripta parametru secība, kas tiek nodota skriptam kā komandrindas argumenti.
value	string	Skripta parametra vērtība.

Note:
Skatiet arī: Multivides tipa objekts (attiecībā uz atbilstošo īpašību ar tādu pašu nosaukumu).

webhook

Tālāk norādītie papildu elementi tiek eksportēti tikai *webhook* multivides tipam.

Element	Type	Description
script	string	Skripts.
timeout	string	Javascript skripta HTTP pieprasījuma taimauta 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 {EVENT.TAGS.*} makro ir 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 {EVENT.TAGS.*} makro.
event_menu_name	string	Notikuma izvēlnes ieraksta nosaukums. Atbalsta {EVENT.TAGS.*} makro.
parameters		Saknes elements webhook multivides tipa parametriem.
name	string	(obligāts) Webhook parametra nosaukums.
value	string	Webhook parametra vērtība.

Note:
Skatiet arī: Media type object (skatiet atbilstošo īpašību ar sakritīgu nosaukumu).

Piezīmes

[1] API veselo skaitļu vērtības iekavās, piemēram, ENABLED (0), ir minētas tikai uzziņai. Papildu informāciju skatiet saistītajā API objekta lapā tabulas ierakstā vai katras sadaļas beigās.

Informācijas paneļi

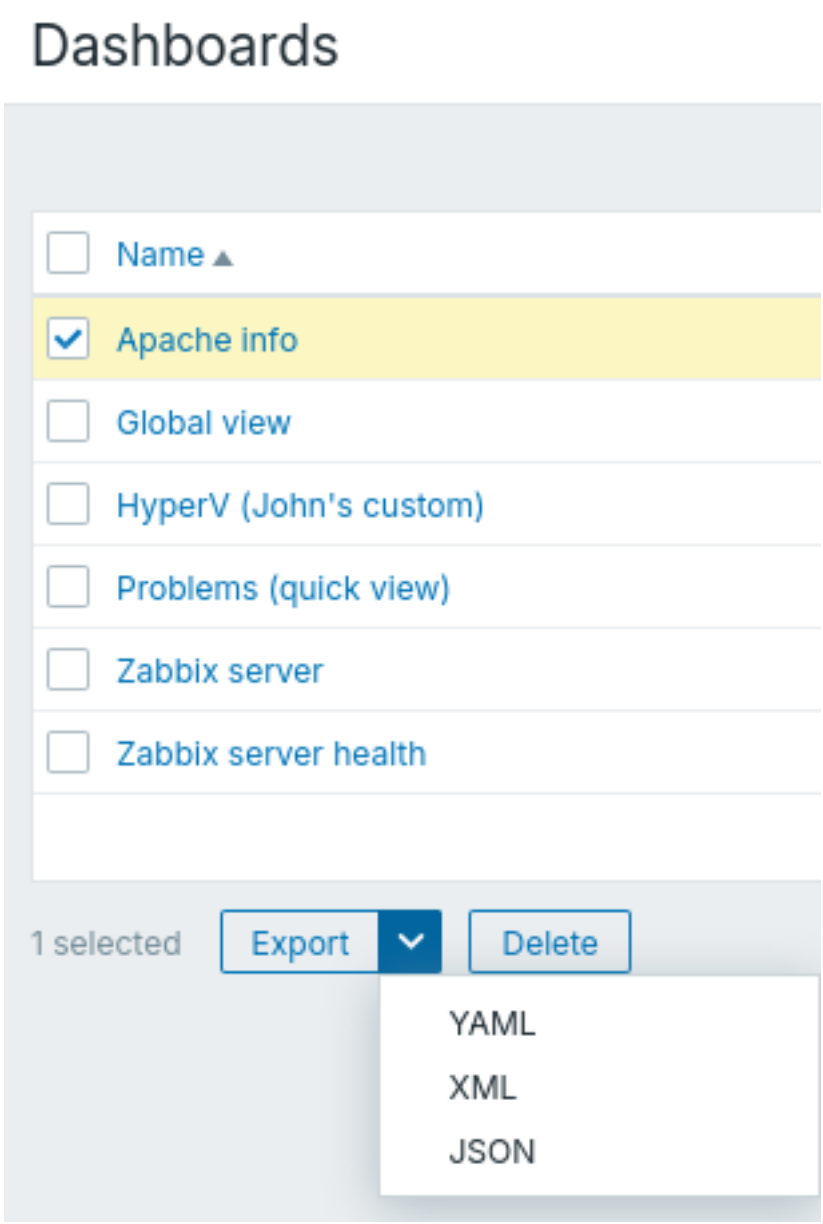
Pārskats

Informācijas paneļi tiek eksportēti kopā ar visiem saistītajiem objektiem un objektu attiecībām.

Eksportēšana

Lai eksportētu paneļus, rīkojieties šādi:

1. Dodieties uz *Paneļi* > *Visi paneļi*.
2. Atzīmējiet eksportējamo paneļu izvēles rūtiņas.
3. Noklikšķiniet uz *Eksportēt* zem saraksta, lai eksportētu noklusējuma YAML formātā, vai noklikšķiniet uz bultiņas, lai atlasītu citu eksporta formātu.



Atkarībā no atlasītā formāta paneļi tiek eksportēti uz lokālu failu ar noklusējuma nosaukumu:

- `zbx_export_dashboards.yaml` - YAML eksportā;
- `zbx_export_dashboards.xml` - XML eksportā;
- `zbx_export_dashboards.json` - JSON eksportā.

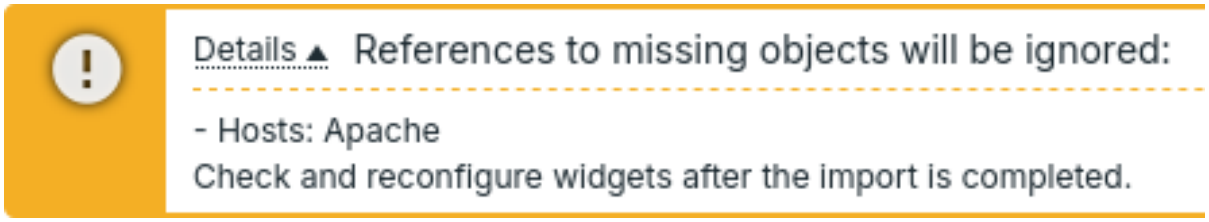
Importēšana

Lai importētu informācijas paneļus, rīkojieties, kā norādīts tālāk:

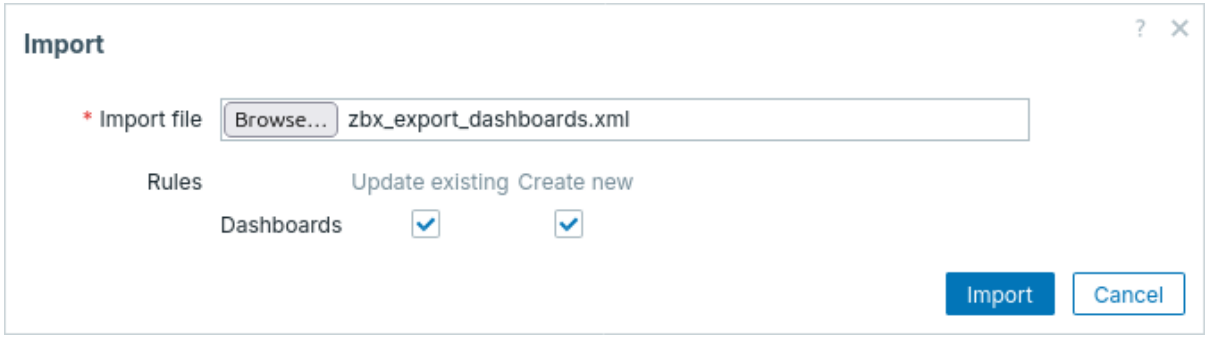
1. Dodieties uz *Informācijas paneļi* > *Visi informācijas paneļi*.
2. Augšējā labajā stūrī noklikšķiniet uz *Importēt*. Šī opcija ir atspējota, ja lietotājam nav atļaujas *Izveidot un rediģēt informācijas paneļus*.

3. Atlasiet importēšanas failu.
4. Atzīmējiet to importēšanas noteikumu izvēles rūtiņas, kurus vēlaties lietot (skatiet tālāk).
5. Konfigurācijas formas apakšējā labajā stūrī noklikšķiniet uz *Importēt*.

Ja importēšanas failā ir atsauces uz objektiem, kas mērķa sistēmā nepastāv, tiek parādīts brīdinājuma ziņojums. Tas ir paredzēts logrīkiem, kas attiecas uz konkrētiem hostiem, vienumiem vai citiem objektiem. Šādos gadījumos Zabbix ļauj importēt informācijas paneli, notīra trūkstošās atsauces un vai nu atzīmē logrīku kā nepilnīgi konfigurētu, ja dati ir obligāti, vai noņem filtrēšanas kritērijus, ja lauki tiek izmantoti tikai filtrēšanai.



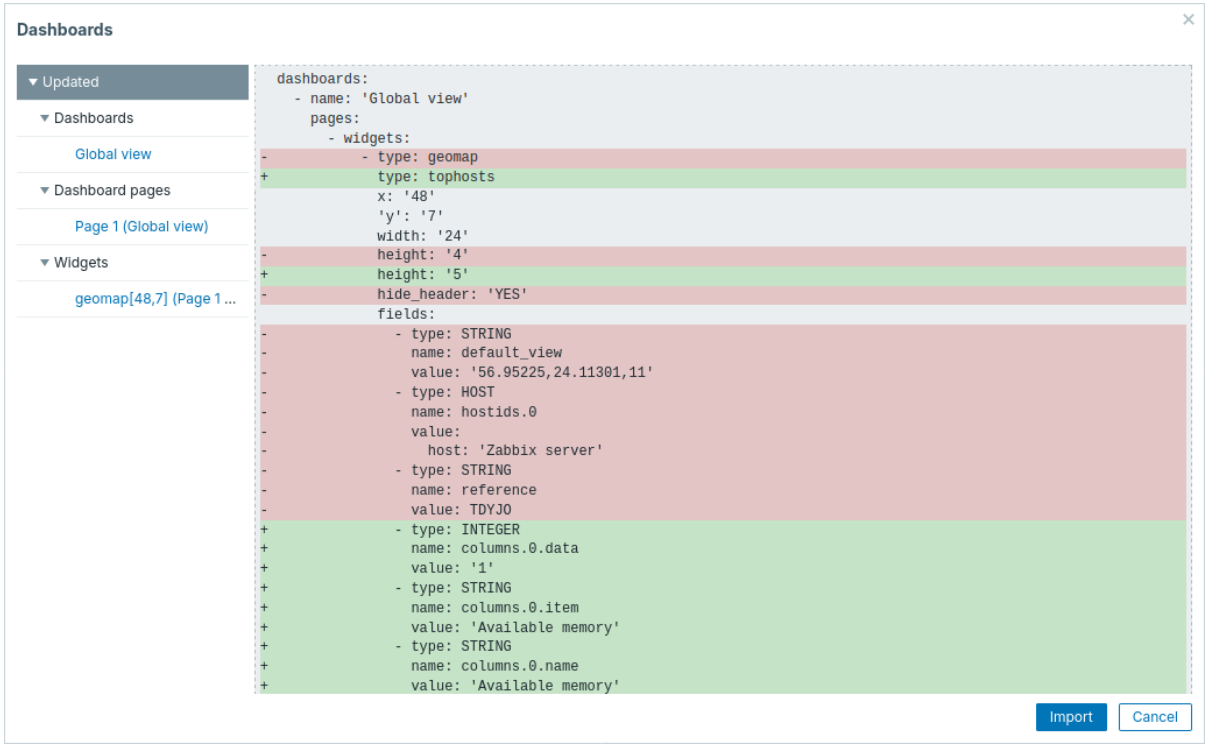
Vēlreiz noklikšķinot uz pogas *Importēt*, informācijas panelis tiks pievienots un/vai atjaunināts.



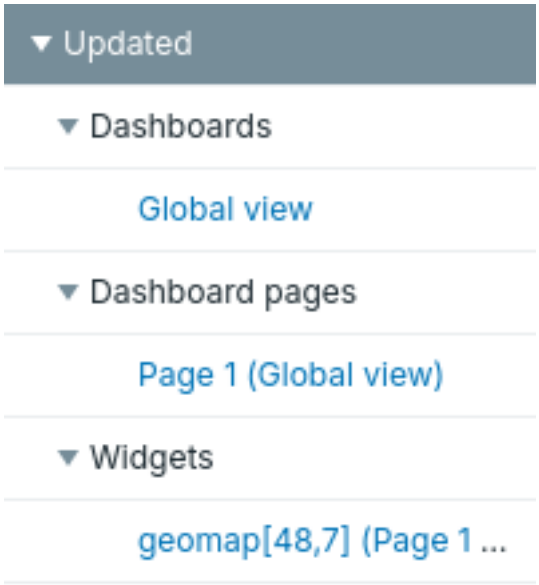
Importēšanas noteikumi:

Noteikums	Apraksts
Atjaunināt esošos	Esošie elementi tiks atjaunināti, izmantojot importēšanas faila datus. Pretējā gadījumā tie netiks atjaunināti.
Izveidot jaunus	Jauni elementi tiks izveidoti, izmantojot importēšanas faila datus. Pretējā gadījumā tie netiks izveidoti.

Nākamajā ekrānā varēsiet skatīt importējamo informācijas paneļu saturu. Ja tas ir jauns informācijas panelis, visi elementi tiks parādīti zaļā krāsā. Atjauninot esošu informācijas paneli, jaunie informācijas paneļa elementi tiks izcelti zaļā krāsā, noņemtie informācijas paneļa elementi tiks izcelti sarkanā krāsā, bet elementi, kas nav mainīti, tiks parādīti uz pelēka fona.



Kreisajā pusē esošo izvēlni var izmantot, lai pārvietotos izmaiņu sarakstā. Sadaļā *Atjaunināts* ir izceltas visas esošo informācijas paneļa elementu izmaiņas. Sadaļā *Pievienots* ir uzskaitīti jaunie informācijas paneļa elementi. Katrā sadaļā elementi ir sagrupēti pēc elementa tipa; noklikšķiniet uz pelēkās bultiņas, lai izvērstu vai sakļautu elementu grupu.



Pārskatiet informācijas paneļa izmaiņas un pēc tam noklikšķiniet uz *Importēt*, lai veiktu informācijas paneļa importēšanu. Lietotāja saskarnē tiks parādīts importēšanas veiksmes vai kļūmes ziņojums.

Eksporta formāts

Eksportēt uz 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
            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: 'Pašreizējās problēmas'
        '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: 'Problēmas pēc nopietnības'
```

```
      x: '25'
      'y': '4'
      width: '23'
      fields:
        - type: STRING
          name: reference
          value: LBSVB
        - type: INTEGER
          name: show_type
          value: '1'
    - type: gauge
      name: 'Zabbix serveris'
      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 serveris'
            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 serveris'
          - type: STRING
            name: ds.0.items.0
            value: 'Pieejamā atmiņa'
          - 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 serveris'
          - 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 serveris'
          - type: STRING
            name: reference
            value: TDYJO
      - type: clock
        name: 'Lokālais laiks'
        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: 'Servera veiktspēja'
      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: 'Vērtības sekundē'
        - type: INTEGER
          name: desc_h_pos
          value: '0'
        - type: INTEGER
          name: desc_v_pos
          value: '0'
        - type: ITEM
          name: itemid.0
          value:
            host: 'Zabbix serveris'
            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'
```

Eksportētie elementi

Eksportētie elementi ir paskaidroti tālāk esošajā tabulā.

Elements	Tips	Apraksts
name	string	(obligāts) Paneļa nosaukums.
pages		Paneļa lapu saknes elements
widgets		Paneļa lapu saknes elements

Note:
Skatiet arī: Dashboard object (skatiet atbilstošo īpašību ar atbilstošu nosaukumu).

Paneļa sīkrīks

Element	Type	Description
type	string	(obligāts) Paneļa sīkrīka tips. Iespējamās vērtības: actionlog - Darbību žurnāls; clock - Pulkstenis; discovery - Atklāšanas statuss; favgraphs - Izlases grafiki; favmaps - Izlases kartes; gauge - Mērītājs; geomap - Ģeokarte; graph - Grafiks (klasisks); graphprototype - Grafika prototips; honeycomb - Honeycomb; hostavail - Hosta 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 - Problēmu hosti; problems - Problēmas; problemsbysv - Problēmas pēc nopietnības; scatterplot - Punktu diagramma; slareport - SLA pārskats; svggraph - Grafiks; systeminfo - Sistēmas informācija; tophosts - Top hosti; topitems - Top vienumi; toptriggers - Top trigeri; trigover - Trigeru pārskats; url - URL; web - Tīmekļa uzraudzība.
x	integer	Horizontālā pozīcija no paneļa kreisās malas. Iespējamās vērtības:[1] 0-71.
y	integer	Vertikālā pozīcija no paneļa augšas. Iespējamās vērtības:[1] 0-63.
name	string	Pielāgots sīkrīka nosaukums.
width	integer	Sīkrīka platums. Iespējamās vērtības:[1] 1-72.
height	integer	Sīkrīka augstums. Iespējamās vērtības:[1] 1-64.
hide_header	string	Paslēpt sīkrīka galveni. Iespējamās vērtības:[1] NO (0, noklusējums), YES (1).
fields		Paneļa sīkrīka lauku saknes elements.

Note:
Skatiet arī: Dashboard widget (skatiet attiecīgo īpašību ar atbilstošu nosaukumu).

Informācijas paneļa sīkrīka lauks

Element	Type	Description
type	string	(required) Sīkrīka lauka tips. Iespējamās vērtības:[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) Sīkrīka lauka nosaukums.
value	mixed	(required) Sīkrīka lauka vērtība atkarībā no lauka tipa.
host	string	Vienuma hosts.
key	string	Vienuma atslēga.

Note:
Skatiet arī: Informācijas paneļa sīkrīka lauks (skatiet atbilstošo īpašību ar sakrītošu nosaukumu).

Piezīmes

[1] API veselo skaitļu vērtības iekavās, piemēram, ENABLED (0), ir minētas tikai uzziņai. Papildu informāciju skatiet saistītajā API objekta lapā tabulas ierakstā vai katras sadaļas beigās.

Regulārās izteiksmes

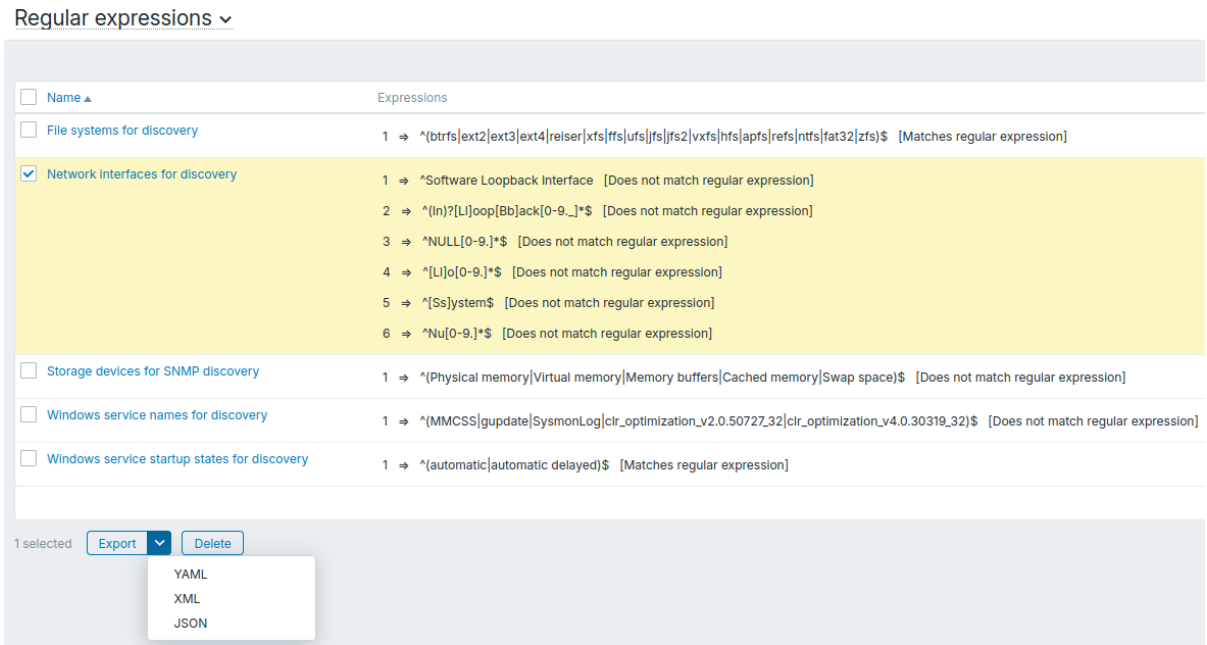
Pārskats

Regulārās izteiksmes tiek eksportētas kopā ar visiem saistītajiem objektiem un objektu attiecībām.

Eksportēšana

Lai eksportētu regulārās izteiksmes, rīkojieties šādi:

1. Dodieties uz *Administration* > *General* > *Regular expressions*.
2. Atzīmējiet eksportējamās regulārās izteiksmes izvēles rūtiņas.
3. Noklikšķiniet uz *Export* zem saraksta, lai eksportētu noklusējuma YAML formātā, vai noklikšķiniet uz bultiņas, lai atlasītu citu eksporta formātu.



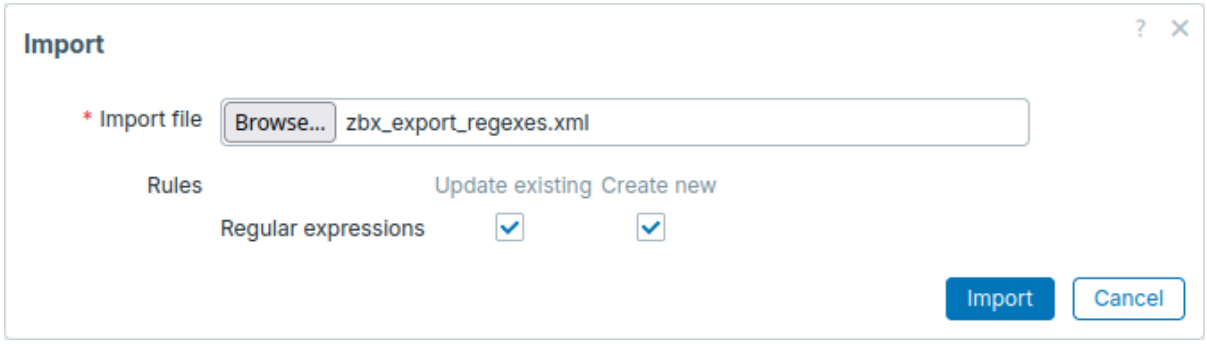
Atkarībā no atlasītā formāta regulārās izteiksmes tiek eksportētas uz lokālu failu ar noklusējuma nosaukumu:

- `zbx_export_regexes.yaml` - YAML eksportā;
- `zbx_export_regexes.xml` - XML eksportā;
- `zbx_export_regexes.json` - JSON eksportā.

Importēšana

Lai importētu regulārās izteiksmes, rīkojieties, kā norādīts tālāk:

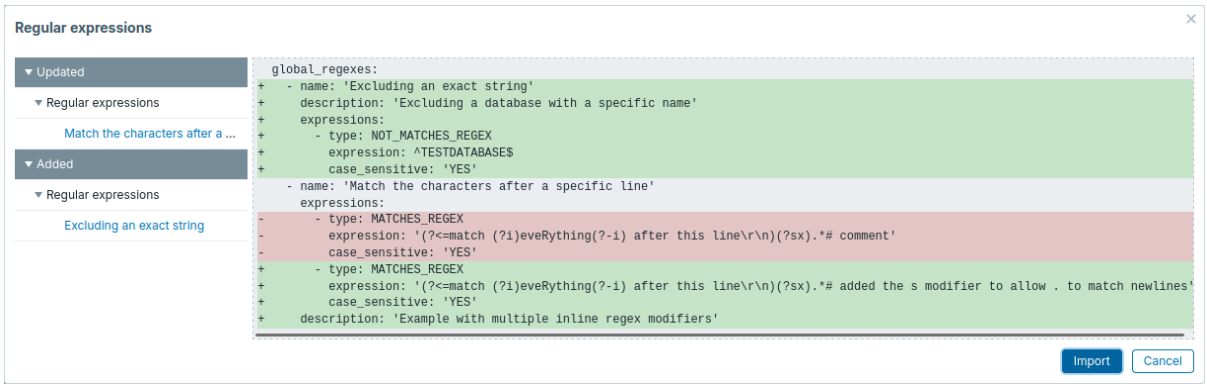
1. Atveriet *Administration* > *General* > *Regular expressions*.
2. Augšējā labajā stūrī noklikšķiniet uz *Import*.
3. Atlasiet importēšanas failu.
4. Atzīmējiet to importēšanas noteikumu izvēles rūtiņas, kurus vēlaties lietot (skatiet tālāk).
5. Konfigurācijas formas apakšējā labajā stūrī noklikšķiniet uz *Import*.



Importēšanas noteikumi:

Noteikums	Apraksts
Update existing	Esošās regulārās izteiksmes tiks atjauninātas, izmantojot importēšanas faila datus, pilnībā pārrakstot esošās izteiksmes ar tādu pašu nosaukumu. Pretējā gadījumā tās netiks atjauninātas.
Create new	Izmantojot importēšanas faila datus, tiks izveidotas jaunas regulārās izteiksmes. Pretējā gadījumā tās netiks izveidotas.

Nākamajā ekrānā varēsiet skatīt importējamo regulāro izteiksmju saturu. Ja tā ir jauna regulārā izteiksme, visi dati tiks parādīti zaļā krāsā. Atjauninot esošu regulāro izteiksmi, jaunie regulārās izteiksmes dati tiks izcelti zaļā krāsā, noņemtie dati tiks izcelti sarkanā krāsā, bet nemainītie dati tiks parādīti uz pelēka fona.



Kreisajā pusē esošo izvēlni var izmantot, lai pārvietotos pa izmaiņu sarakstu. Sadaļā *Updated* ir izceltas visas esošajās regulārajās izteiksmēs veiktās izmaiņas. Sadaļā *Added* ir uzskaitītas jaunās regulārās izteiksmes. Noklikšķiniet uz bultiņas, lai izvērstu vai sakļautu regulāro izteiksmju grupu.

Pārskatiet regulāro izteiksmju izmaiņas un pēc tam noklikšķiniet uz *Import*, lai importētu regulārās izteiksmes. Lietotāja saskarnē tiks parādīts importēšanas veiksmes vai kļūmes ziņojums.

Eksporta formāts

Eksportēt uz YAML:

```
zabbix_export:
  version: '8.0'
  global_regexes:
    - name: 'Failu sistēmas atklāšanai'
      description: 'Globāla regulārā izteiksme, ko izmanto, lai atklātu un identificētu failu sistēmas pār
      expressions:
        - type: MATCHES_REGEX
          expression: ^(btrfs|ext2|ext3|ext4|reiser|xfs|ffs|ufs|jfs|jfs2|vxfs|hfs|apfs|refs|ntfs|fat32|zfs
    - name: 'Krātuves ierīces SNMP atklāšanai'
      expressions:
        - type: NOT_MATCHES_REGEX
          expression: '^(Physical memory|Virtual memory|Memory buffers|Cached memory|Swap space)$'
          case_sensitive: 'YES'
```

Eksportētie elementi

Regulāro izteiksmju eksportētie elementi:

Element	Type	Description
global_regexes	object	Regulāro izteiksmju saknes elements.
name	string	(required) Regulārās izteiksmes nosaukums.
description	text	Regulārās izteiksmes apraksts.
expressions	object	Regulāro izteiksmju saknes elements.
type	string	Regulārās izteiksmes tips.
expression	string	Regulārā izteiksme.
exp_delimiter	string	Regulārās izteiksmes atdalītājs; ir pieejams, ja tips ir *Contains any substring from list*.
case_sensitive	string	Reģistrjutības opcija regulārajai izteiksmei.

Note:
Skatiet arī: Regular expression object (skatiet attiecīgo īpašību ar atbilstošu nosaukumu).

Regulārās izteiksmes

Pārskats Perl Compatible Regular Expressions (PCRE, PCRE2) tiek atbalstītas Zabbix.

Zabbix regulārās izteiksmes var izmantot divos veidos:

- Manuāli ievadot regulāro izteiksmi
- Izmantojot globālo regulāro izteiksmi, kas izveidota Zabbix

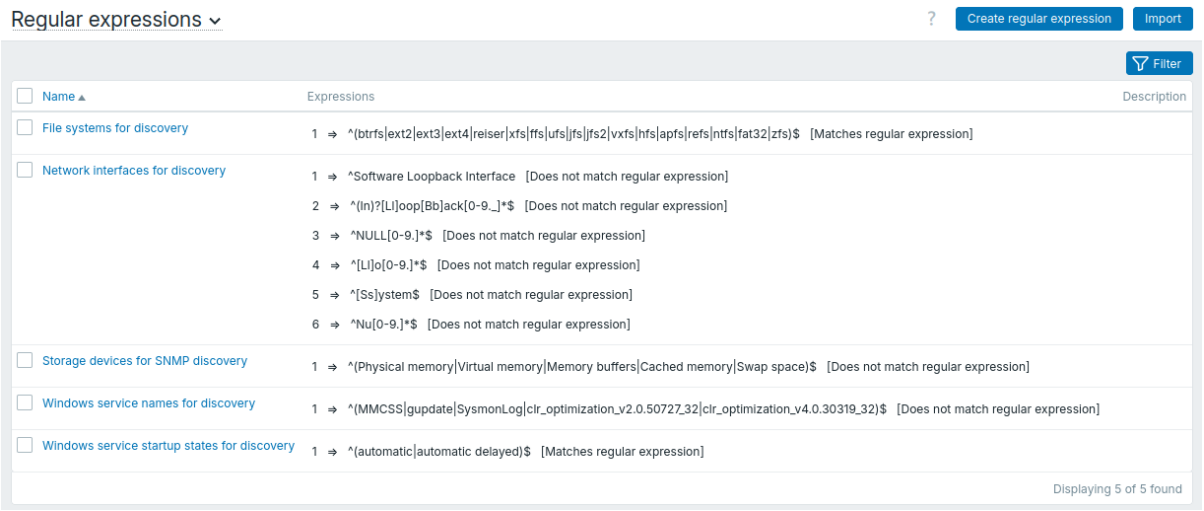
Regulārās izteiksmes Jūs varat manuāli ievadīt regulāro izteiksmi atbalstītajās vietās.
 Ņemiet vērā, ka izteiksme nedrīkst sākties ar @, jo šis simbols Zabbix tiek izmantots globālo regulāro izteiksmju atsaucei.

Attention:
Izmantojot regulārās izteiksmes, var beigties steka atmiņa. Plašāku informāciju skatiet PCRE PERFORMANCE.

Ņemiet vērā, ka vairākrindu atbilstības gadījumā enkuri ^ un $ atbilst attiecīgi katras rindas sākumam un beigām, nevis visas virknes sākumam un beigām.

Skatiet arī piemērus par pareizu simbolu atsoļošanu dažādos kontekstos.

Globālie regulārie izteicieni Zabbix lietotāja saskarne piedāvā papildu redaktoru sarežģītu regulāro izteicienu izveidei un testēšanai.
 Lai tam piekļūtu, dodieties uz *Administration* > *General* > *Regular expressions*.
 Zabbix noklusējuma datu kopā ir iekļauts vairāku iepriekš definētu globālo regulāro izteicienu kopums.



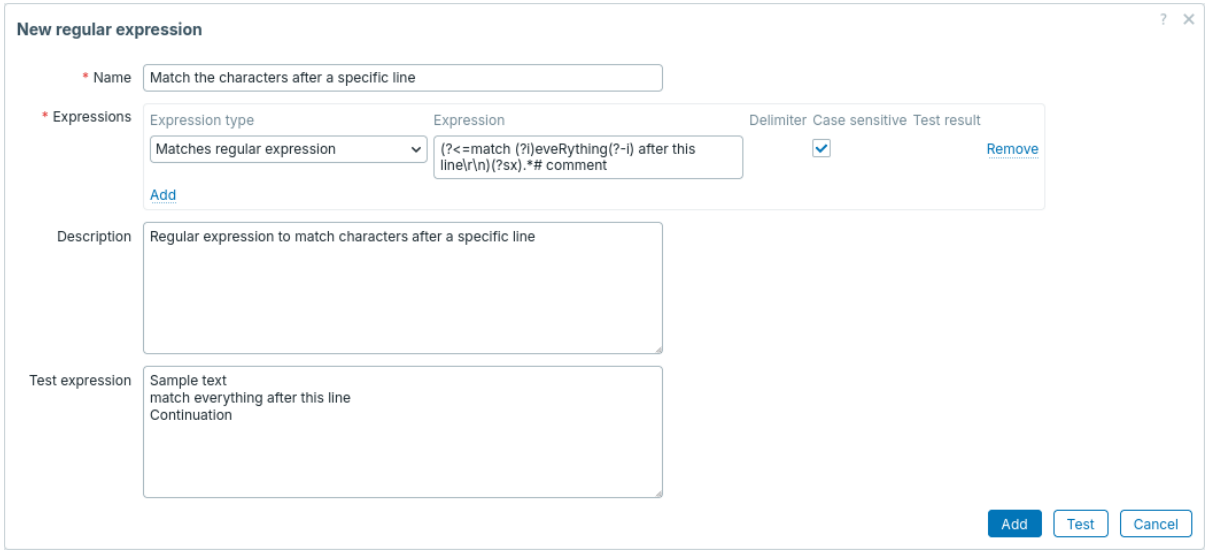
Jūs varat filtrēt regulāro izteicienu sarakstu pēc nosaukuma un apraksta, lai atrastu nepieciešamos izteicienus.

Konfigurācija

Lai izveidotu globālu regulāro izteiksmi:

1. Noklikšķiniet uz *Create regular expression*.

2. Formā ievadiet regulārās izteiksmes parametrus.



Visi obligātie ievades lauki ir atzīmēti ar sarkanu zvaigznīti.

Parametrs	Apraksts
Name	Iestatiet regulārās izteiksmes nosaukumu. Ir atļautas jebkādas Unicode rakstzīmes.
Expressions	Sadaļā *Expressions* noklikšķiniet uz *Add*, lai pievienotu jaunu apakšizteiksmi.
Expression type	Atlasiet izteiksmes tipu: **Contains string** - atbilst apakšvirknei. **Contains any substring from list** - atbilst jebkurai apakšvirknei no saraksta, kura elementi ir atdalīti ar norobežotāju. Saraksta elementi tiek atdalīti ar komatu `,`, punktu `.` vai slīpsvītru `/`. **Does not contain string** - atbilst jebkurai virknei, izņemot apakšvirkni. **Matches regular expression** - atbilst regulārajai izteiksmei. **Does not match regular expression** - neatbilst regulārajai izteiksmei.
Expression	Ievadiet apakšvirkni/regulāro izteiksmi. Maksimālais garums: 2048 rakstzīmes.
Delimiter	Komats `,`, punkts `.` vai slīpsvītra `/`, lai regulārajā izteiksmē atdalītu teksta virknes. Šis parametrs ir aktīvs tikai tad, ja ir atlasīts izteiksmes tips *Contains any substring from list*.
Case sensitive	Izvēles rūtiņa, lai norādītu, vai regulārā izteiksme ir reģistrjutīga.
Description	Ievadiet regulārās izteiksmes aprakstu.
Test expression	Vērtība, ko izmantot regulārās izteiksmes testēšanai.

Slīpsvītra `/` izteiksmē tiek interpretēta burtiski, nevis kā norobežotājs. Tas ļauj saglabāt izteiksmes, kurās ir slīpsvītra, neradot kļūdas.

Attention:
Pielāgotas regulārās izteiksmes nosaukums Zabbix var saturēt komatus, atstarpes u. c. Gadījumos, kad tas var izraisīt nepareizu interpretāciju, veidojot atsauci (piemēram, komats vienuma atslēgas parametrā), visa atsauce var tikt ietverta pēdiņās, piemēram: *"@My custom regexp for purpose1, purpose2"*.
 Citos gadījumos regulāro izteiksmju nosaukumus nedrīkst likt pēdiņās (piemēram, LLD kārtulas rekvizītos).

Kad regulārā izteiksme ir izveidota, to var izmantot vairākās lietotāja saskarnes vietās, atsaucoties uz tās nosaukumu ar prefiksu `@`, piemēram, `@mycustomregexp`.

Veidlapas pogas

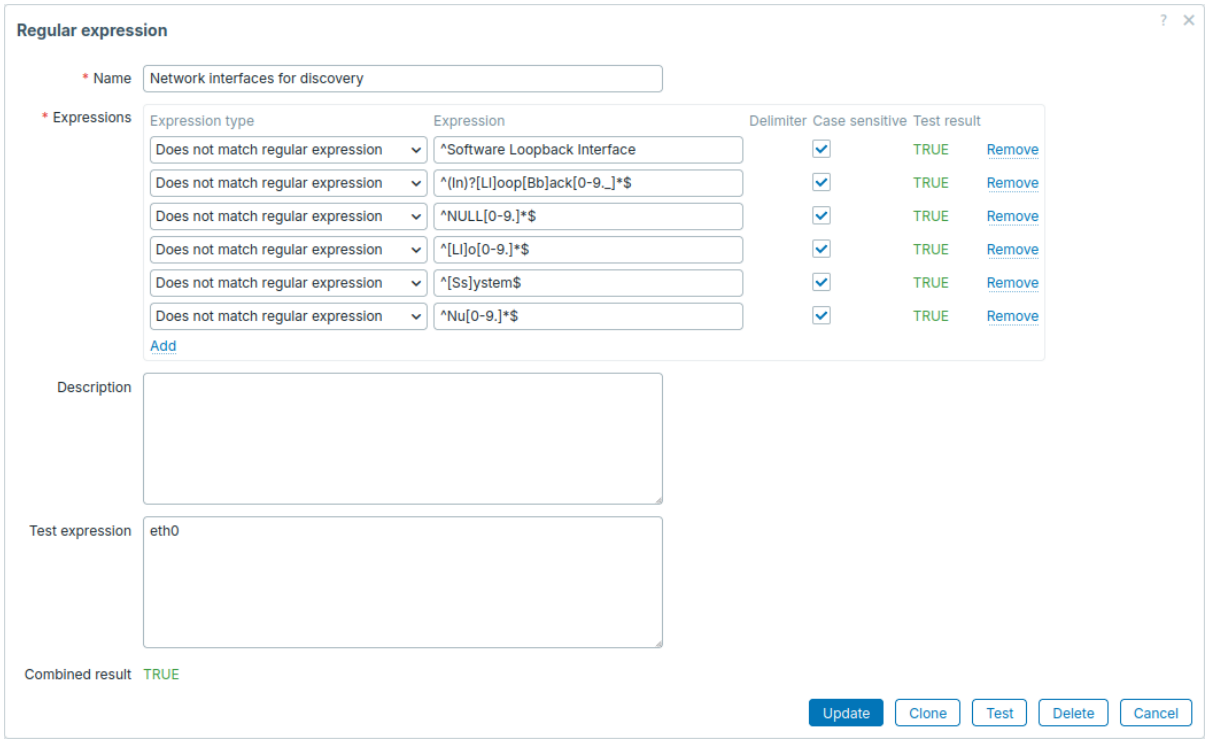
Pogas veidlapas apakšdaļā ļauj veikt dažādas darbības:

Poga	Apraksts
Add	Pievienot regulāro izteiksmi. Šī poga ir pieejama tikai jaunām regulārajām izteiksmēm.

Poga	Apraksts
Update	Atjaunināt regulārās izteiksmes īpašības.
Clone	Izveidot citu regulāro izteiksmi, pamatojoties uz pašreizējās regulārās izteiksmes īpašībām.
Test	Pārbaudīt regulāro izteiksmi un tās apakšizteiksmes pret norādīto vērtību.
Delete	Dzēst regulāro izteiksmi.
Cancel	Atcelt regulārās izteiksmes īpašību rediģēšanu.

Regulārās izteiksmes testēšana

Jūs varat pārbaudīt regulāro izteiksmi un tās apakšizteiksmes, ievadot vērtību laukā *Test expression* un noklikšķinot uz *Test* pogas.



Kolonnā *Test result* tiek parādīts katras apakšizteiksmes statuss. Lauks *Combined result* parāda regulārās izteiksmes kopējo statusu. Ja ir definētas vairākas apakšizteiksmes, Zabbix izmanto loģisko AND operatoru, lai aprēķinātu *Combined result*; tas ir, ja vismaz viens rezultāts ir `False`, tad arī *Combined result* ir `False`.

Noklusējuma globālie regulārie izteicieni

Noklusējuma datu kopā ir iekļauti šādi iepriekš definēti globālie regulārie izteicieni:

Name	Expression	Matches
Failu sistēmas atklāšanai	`^(btrfs|ext2|ext3|ext4|reiser|xfs|ffs|ufs|jfs|jfs2|vxfs|hfs|apfs|refs|ntfs|fat32|zfs)$`	”btrfs” vai ”ext2” vai ”ext3” vai ”ext4” vai ”reiser” vai ”xfs” vai ”ffs” vai ”ufs” vai ”jfs” vai ”jfs2” vai ”vxfs” vai ”hfs” vai ”apfs” vai ”refs” vai ”ntfs” vai ”fat32” vai ”zfs”.
Tīkla saskarnes atklāšanai	`^Software Loopback Interface`	Virknes, kas sākas ar ”Software Loopback Interface”.
	`^lo$`	”lo”.
	`^(In)?[Ll]oop[Bb]ack[0-9._]*$`	Virknes, kas pēc izvēles sākas ar ”In”, kam seko ”L” vai ”l”, pēc tam ”oop”, pēc tam ”B” vai ”b”, pēc tam ”ack”, un pēc izvēles seko jebkurš ciparu, punktu vai pasvītrojumu skaits.
	`^NULL[0-9.]*$`	Virknes, kas sākas ar ”NULL” un kam pēc izvēles seko jebkurš ciparu vai punktu skaits.
	`^[Ll]o[0-9.]*$`	Virknes, kas sākas ar ”Lo” vai ”lo” un kam pēc izvēles seko jebkurš ciparu vai punktu skaits.
	`^[Ss]ystem$`	”System” vai ”system”.
	`^Nu[0-9.]*$`	Virknes, kas sākas ar ”Nu” un kam pēc izvēles seko jebkurš ciparu vai punktu skaits.

Name	Expression	Matches
SNMP atklāšanas krātuves ierīcēm	`^(Physical memory|Virtual memory|Memory buffers|Cached memory|Swap space)$`	”Physical memory” vai ”Virtual memory” vai ”Memory buffers” vai ”Cached memory” vai ”Swap space”.
Windows pakalpojumu nosaukumiem atklāšanai	`^(MMCSS|gupdate|SysmonLog|clr_optimization_v2.0.50727_32|clr_optimization_v4.0.30319_32)$`	”MMCSS” vai ”gupdate” vai ”SysmonLog” vai virknes, kas atbilst ”clr_optimization_v2.0.50727_32” vai ”clr_optimization_v4.0.30319_32”, kur katru punktu var aizstāt ar jebkuru rakstzīmi, izņemot jaunu rindu.
Windows pakalpojumu startēšanas stāvokļiem atklāšanai	`^(automatic|automatic delayed)$`	”automatic” vai ”automatic delayed”.

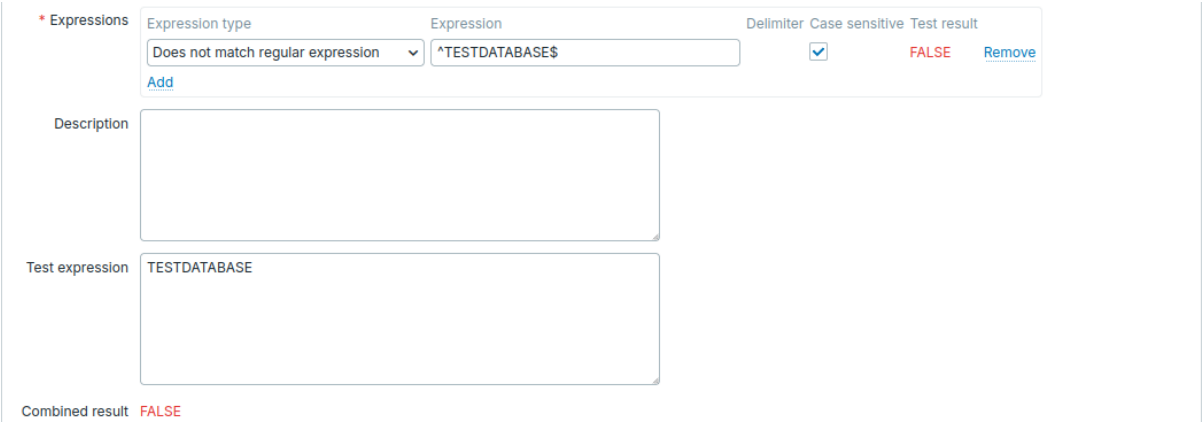
Regulārās izteiksmes importēšana/eksportēšana

Globālās regulārās izteiksmes var importēt vai eksportēt, kā aprakstīts Regulārās izteiksmes eksportēšanas/importēšanas konfigurācijas lapā.

Piemēri Precīzas virknes izslēgšanas piemērs

Lai zemā līmeņa atklāšanā atrastu datubāzes, izņemot datubāzi ar noteiktu nosaukumu, izmantojiet šādu izteiksmi:

```
^TESTDATABASE$
```

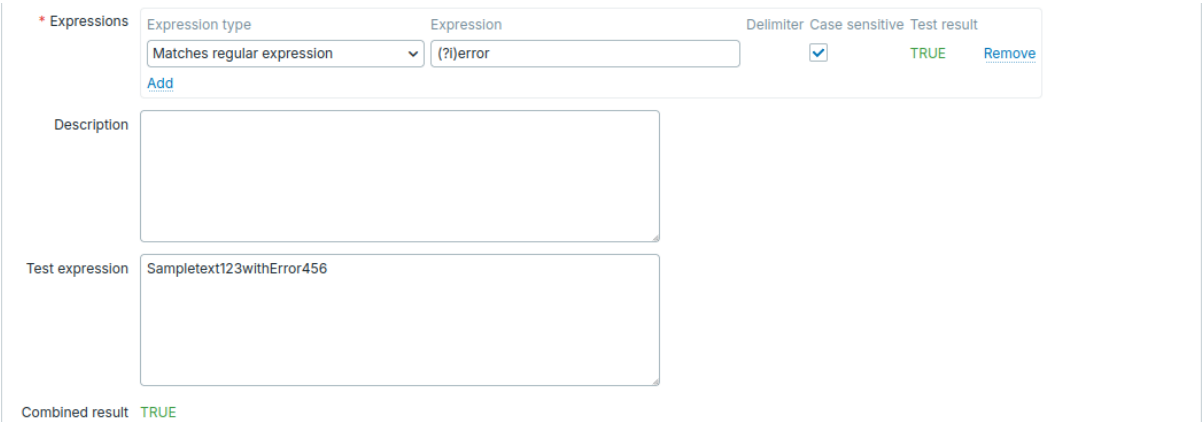


Gan *Test result*, gan *Combined result* rāda FALSE: nosaukums atbilst ”TESTDATABASE”, tāpēc nosacījums *Does not match regular expression* nav izpildīts.

Piemērs ar iekļautu regulārās izteiksmes modifikatoru reģistrjutības ignorēšanai

Izmantojiet šādu regulāro izteiksmi, kurā ir iekļauts iekšējais modifikators `(?i)`, lai veiktu simbolu ”error” reģistrneatkarīgu atbilstību:

```
(?i)error
```



Gan *Test result*, gan *Combined result* rāda TRUE: simboli "error" tiek atrasti pat tad, ja ir atlasīts *Case sensitive*, jo iekšējais modifikators `(?i)` padara atbilstību reģistrneatkarīgu.

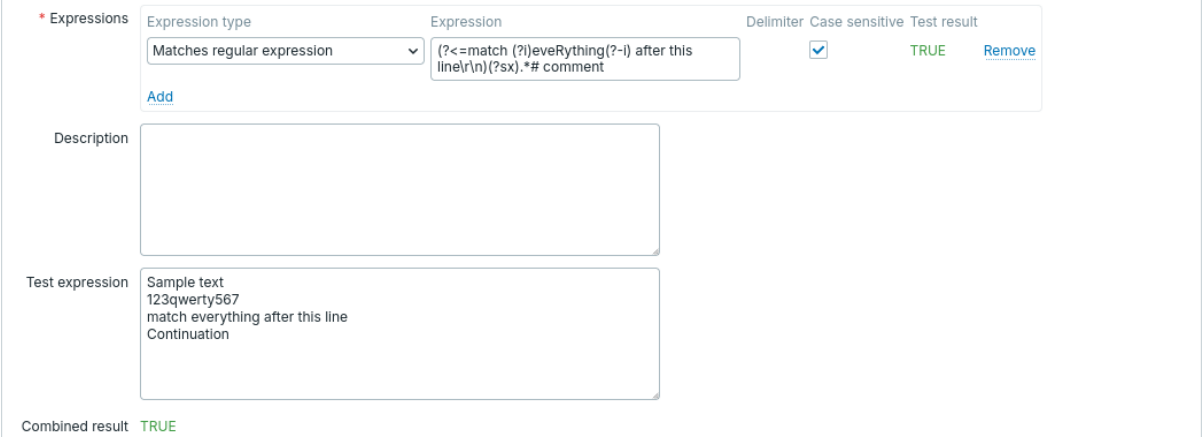
Piemērs ar vairākiem inline regex modifikatoriem

Izmantojiet šādu regulāro izteiksmi, kas ietver vairākus inline modifikatorus, lai saskaņotu rakstzīmes pēc noteiktas rindas:

```
(?<=match (?i)eveRything(?-i) after this line\r\n)(?sx).*# comment
```

kur:

- `(?i)` - ieslēdz reģistrjutības ignorēšanu;
- `(?-i)` - izslēdz reģistrjutības ignorēšanu;
- `(?s)` - iespējo dotall režīmu, ļaujot `.` saskaņot jaunas rindas rakstzīmes;
- `(?x)` - iespējo paplašināto režīmu, ļaujot regulārajā izteiksmē ignorēt atstarpes un komentārus.



Gan *Test result*, gan *Combined result* rāda TRUE: tiek saskaņotas rakstzīmes, kas seko konkrētajai rindai.

Attention:
g modifikatoru nevar norādīt inline. Pieejamo modifikatoru sarakstu var atrast pcresyntax man page. Lai iegūtu vairāk informācijas par PCRE sintaksi, lūdzu, skatiet PCRE HTML dokumentāciju.

Regulārās izteiksmes atbalsts pēc atrašanās vietas

Atrašanās vieta		Regulārā izteiksme	Globālā regulārā izteiksme	Daudzrindu 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
	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

Atrašanās vieta		Regulārā izteiksme	Globālā regulārā izteiksme	Daudzrindu atbilstība	Komentāri
	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 slazdi					
	snmptrap[]	Jā	Jā	Nē	`regexp` parametrs
Vienuma vērtības pirmapstrāde		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	*Regulārā izteiksme* lauks
	Pārrakstīšanas	Jā	Nē		Opcijās *atbilst*, *neatbilst* sadaļā *Operation* nosacījumi
Darbību nosacījumi		Jā	Nē	Nē	Opcijās *atbilst*, *neatbilst* sadaļā *Host name* un *Host metadata* automātiskās reģistrācijas nosacījumi
Skripti		Jā	Jā	Nē	*Input validation rule* lauks
Tīmekļa uzraudzība		Jā	Nē	Jā	*Variables* ar prefiksu **regex:** *Required string* lauks
Lietotāja makro konteksts		Jā	Nē	Nē	Makro kontekstā ar prefiksu **regex:**
Makro funkcijas					
	regsub()	Jā	Nē	Nē	`pattern` parametrs
	iregsub()				
Saišu indikatori kartēs		Jā	Nē	Nē	*Pattern* lauks (teksta vienumiem)
Ikonu kartēšana		Jā	Jā	Nē	*Expression* lauks
Vērtību kartēšana		Jā	Nē	Nē	*Value* lauks, ja kartēšanas tips ir `regexp`

Atrašanās vieta		Regulārā izteiksme	Globālā regulārā izteiksme	Daudzrindu atbilstība	Komentāri
Aģenta konfig-urācijas parametri					
	AllowKeyRegexp DenyKeyRegexp	Jā	Nē	Nē	Raksts

Notikumi un atrisināšana

Pārskats Zabbix tiek ģenerēti vairāku veidu notikumi:

- 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 atklāti hosts vai pakalpojumi
- automātiskās reģistrācijas notikumi - kad serveris automātiski reģistrē aktīvos aģentus
- iekšējie notikumi - kad vienums/zemā līmeņa atklāšanas noteikums kļūst neatbalstīts vai trigeris nonāk nezināmā stāvoklī

Notikumiem tiek pievienots laika zīmogs, un tie var kalpot par pamatu tādām darbībām kā paziņojuma e-pasta nosūtīšana, biļešu izveide vai attālo komandu izpilde.

Lai lietotāja saskarnē skatītu notikuma informāciju, dodieties uz *Monitoring* > *Problems*. Tur varat noklikšķināt uz notikuma datuma un laika, lai atvērtu notikuma informācijas lapu, kur varat redzēt papildu informāciju par notikumu, tostarp to, vai tas tika slāpēts apkopes dēļ vai manuāli slāpēts lietotāja.

Problēmu atrisināšana Ar trigeriem izveidotie problēmu notikumi tiek atrisināti automātiski, ja nosacījums, kas izraisīja problēmu, vairs nepastāv.

Lai nodrošinātu lielāku elastību un ērtāku problēmu notikumu pārvaldību, ir pieejamas šādas opcijas:

- Notikumu korelācija
- Notikumu apstiprināšana un slāpēšana
- Manuāla problēmu aizvēršana
- Uzturēšanas periodi - definē periodus, kad problēmas tiek slāpētas
- Marķēšana - marķēt problēmas pēc kopīgas pazīmes

Trigeru notikumi Trigeru stāvokļa maiņa ir visbiežākais un vissvarīgākais notikumu avots. Katru reizi, kad trigeris maina savu stāvokli, tiek ģenerēts notikums. Notikums satur informāciju par trigeris stāvokļa maiņu - kad tā notika un kāds ir jaunais stāvoklis.

Trigeri rada divu veidu notikumus - `Problem` un `OK`.

Problēmu notikumi

Problēmu notikums tiek izveidots:

- kad trigeris izteiksme tiek novērtēta kā TRUE, ja trigeris ir OK stāvoklī;
- katru reizi, kad trigeris 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 ir manuāli aizvērts

Trigeriem ir iestatījums ”OK notikuma ģenerēšana”, kas kontrolē, kā tiek ģenerēti OK notikumi:

- *Izteiksme* - OK notikums tiek ģenerēts trigerim problēmas stāvoklī, kad tā izteiksme tiek novērtēta kā FALSE. Šis ir vienkāršākais iestatījums, kas pēc noklusējuma ir iespējots.
- *Atjaunošanas izteiksme* - OK notikums tiek ģenerēts trigerim problēmas stāvoklī, kad tā izteiksme tiek novērtēta kā FALSE un atjaunošanas izteiksme tiek novērtēta kā TRUE. To var izmantot, ja trigera atjaunošanas kritēriji atšķiras no problēmas kritērijiem.

- *Nav* - 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 notikums aizver”, kas kontrolē, kuri problēmu notikumi tiek aizvērti:

- *Visas problēmas* - OK notikums aizvērs visas atvērtās problēmas, ko izveidojis trigeris
- *Visas problēmas, ja tagu vērtības sakrīt* - OK notikums aizvērs atvērtās problēmas, ko izveidojis trigeris un kurām ir vismaz viena sakrītoša taga vērtība. Tags ir definēts ar trigera iestatījumu ”Tag for matching”. Ja nav problēmu notikumu, ko aizvērt, tad OK notikums netiek ģenerēts. To bieži sauc par trigera līmeņa notikumu korelāciju.

Notikumu korelācija

Notikumu korelācija (saukta arī par globālo notikumu korelāciju) ir veids, kā iestatīt pielāgotus notikumu aizvēršanas noteikumus (kuru rezultātā tiek ģenerēts OK notikums).

Noteikumi nosaka, kā jaunie problēmu notikumi tiek savienoti pārī ar esošajiem problēmu notikumiem, ļaujot aizvērt jauno notikumu vai atbilstošos notikumus, ģenerējot attiecīgos OK notikumus.

Tomēr notikumu korelācija ir jākonfigurē ļoti rūpīgi, jo tā var negatīvi ietekmēt notikumu apstrādes veiktspēju vai, ja tā ir nepareizi konfigurēta, aizvērt vairāk notikumu, nekā paredzēts (sliktākajā gadījumā var tikt aizvērti pat visi problēmu notikumi). Daži konfigurēšanas padomi:

1. vienmēr samaziniet korelācijas tvērumu, iestatot unikālu tagu kontroles notikumam (notikumam, kas tiek savienots pārī ar vecajiem notikumiem), un izmantojiet korelācijas nosacījumu “new event tag”
2. izmantojot darbību “close old event”, neaizmirstiet pievienot nosacījumu, kas balstīts uz veco notikumu, pretējā gadījumā var tikt aizvērtas visas esošās problēmas
3. izvairieties no bieži izmantotiem tagu nosaukumiem, kurus izmanto dažādas korelācijas konfigurācijas

Uzdevumu pārvaldnieks

Ja trigerim ir iespējots iestatījums ”Atļaut manuālu aizvēršanu”, tad ir iespējams manuāli aizvērt problēmu notikumus, ko ģenerē trigeris. To veic lietotāja saskarnē, kad tiek atjaunināta problēma. Notikums netiek aizvērts tieši - tā vietā tiek izveidots ”notikuma aizvēršanas” uzdevums, ko drīzumā apstrādā uzdevumu pārvaldnieks. Uzdevumu pārvaldnieks ģenerēs atbilstošu OK notikumu, un problēmu notikums tiks aizvērts.

Pakalpojuma notikumi Pakalpojuma notikumi tiek ģenerēti tikai tad, ja šo notikumu pakalpojuma darbības ir iespējotas. Šādā gadījumā katras pakalpojuma stāvokļa izmaiņas rezultātā tiek izveidots jauns notikums:

- Problēmas notikums - kad pakalpojuma stāvoklis tiek mainīts no `OK` uz `Problem`
- OK notikums - kad pakalpojuma stāvoklis tiek mainīts no `Problem` uz `OK`

Notikums satur informāciju par pakalpojuma stāvokļa maiņu - kad tā notika un kāds ir jaunais stāvoklis.

Atklāšanas notikumi Zabbix periodiski skenē IP diapazonus, kas definēti tīkla atklāšanas noteikumos. Pārbaudes biežumu var konfigurēt katram noteikumam atsevišķi. Kad tiek atklāts hosts vai pakalpojums, tiek ģenerēts atklāšanas notikums (vai vairāki notikumi).

Zabbix ģenerē šādus notikumus:

Notikums	Kad tiek ģenerēts
Service Up	Katru reizi, kad Zabbix konstatē aktīvu pakalpojumu.
Service Down	Katru reizi, kad Zabbix nevar konstatēt pakalpojumu.
Host Up	Ja IP ir pieejams vismaz viens no pakalpojumiem.
Host Down	Ja visi pakalpojumi neatbild.
Service Discovered	Ja pakalpojums atkal ir pieejams pēc dīkstāves vai tiek atklāts pirmo reizi.
Service Lost	Ja pakalpojums tiek zaudēts pēc tam, kad tas bija pieejams.
Host Discovered	Ja hosts atkal ir pieejams pēc dīkstāves vai tiek atklāts pirmo reizi.
Host Lost	Ja hosts tiek zaudēts pēc tam, kad tas bija pieejams.

Aktīva aģenta automātiskās reģistrācijas notikumi Aktīva aģenta automātiskā reģistrācija Zabbix izveido notikumus.

Ja tā ir konfigurēta, aktīva aģenta automātiskās reģistrācijas notikums tiek izveidots, kad iepriekš nezināms aktīvs aģents pieprasa pārbaudes vai ja ir mainījušies hosts metadati. Serveris pievieno jaunu automātiski reģistrētu hostu, izmantojot saņemto aģenta IP adresi un portu.

Papildinformāciju skatiet lapā aktīva aģenta automātiskā reģistrācija.

Iekšējie notikumi Iekšējie notikumi notiek, kad:

- vienums maina stāvokli no ’normal’ uz ’unsupported’
- vienums maina stāvokli no ’unsupported’ uz ’normal’
- zemā līmeņa atklāšanas noteikums maina stāvokli no ’normal’ uz ’unsupported’
- zemā līmeņa atklāšanas noteikums 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, 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, kad vienumi kļūst neatbalstīti), atspējojiet visas iekšējo notikumu darbības sadaļā *Alerts* > *Actions* > *Internal actions*.

> **Note:**
> Ja iekšējās darbības ir atspējotas, kamēr objekts atrodas stāvoklī ’unsupported’, šim objektam joprojām tiks izveidots atjaunošanās notikums.
> Ja iekšējās darbības ir iespējotas, kamēr objekts atrodas stāvoklī ’unsupported’, šim objektam tiks izveidots atjaunošanās notikums, pat ja objektam nav izveidots ’problem event’.

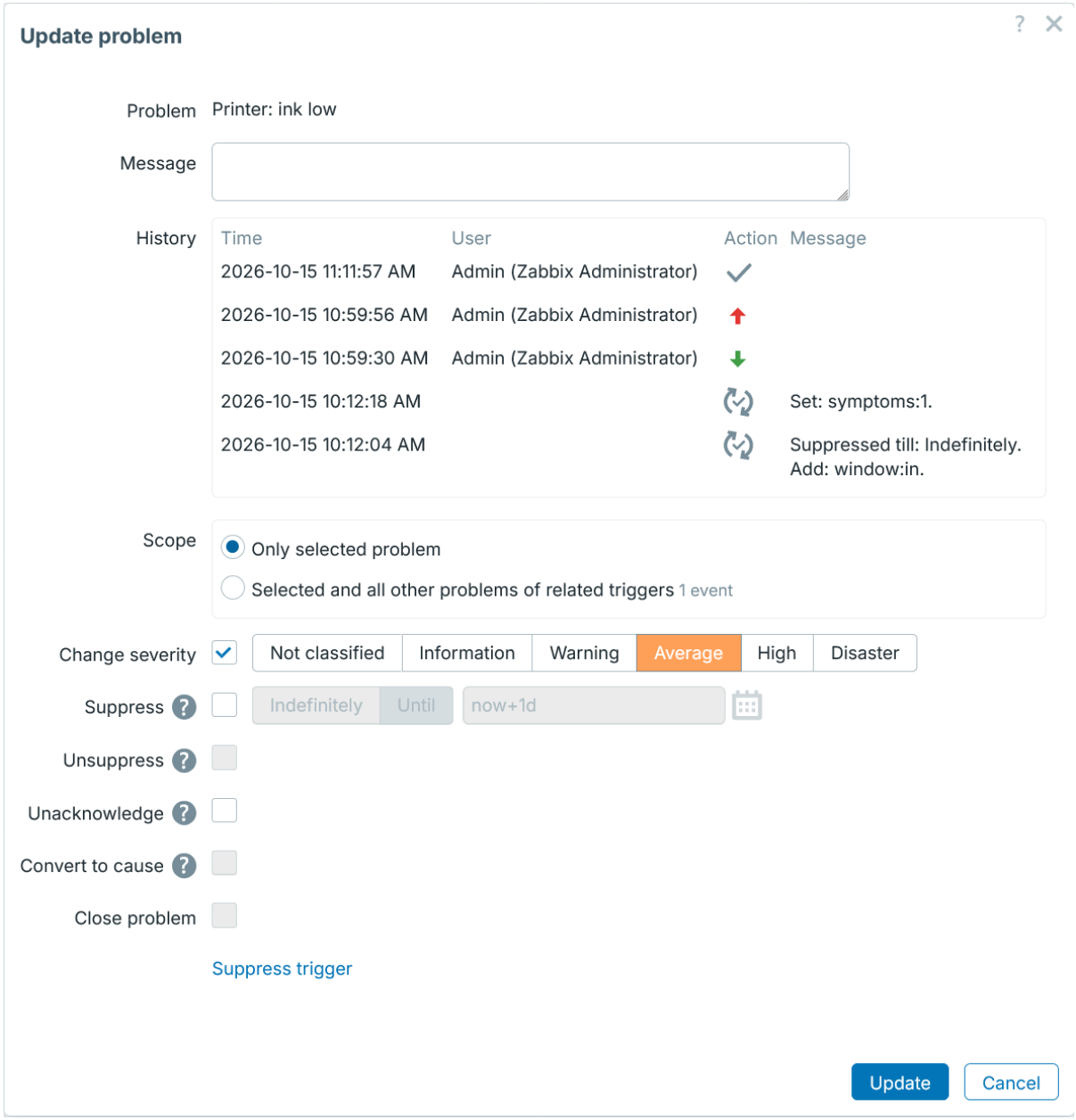
Skatiet arī: Paziņojumu saņemšana par neatbalstītiem vienumiem

Problēmas apstiprināšana un slāpēšana

Problēmu notikumus Zabbix var apstiprināt vai slāpēt lietotāji.

Problēmu atjaunināšana Problēmas atjaunināšanas uznirstošais logs ļauj:

- Pievienot komentāru par problēmu.
- Skatīt līdzšinējos komentārus un darbības.
- Mainīt problēmas nopietnību.
- Apturēt/atcelt problēmas apturēšanu.
- Apstiprināt/atcelt problēmas apstiprināšanu.
- Mainīt simptomātisko problēmu uz cēloņa problēmu.
- Manuāli aizvērt problēmu.
- Izveidot trigerim ad hoc uzturēšanu.



Visi obligātie ievades lauki ir atzīmēti ar sarkanu zvaigznīti.

Parametrs	Apraksts
Problēma	Ja atlasīta tikai viena problēma, tiek parādīts problēmas nosaukums. Ja atlasītas vairākas problēmas, tiek parādīts *Atlasītas N problēmas*.
Ziņojums	Ievadiet tekstu, lai pievienotu komentāru par problēmu (ne vairāk kā 2048 rakstzīmes).
Vēsture	Tiek uzskaitītas iepriekšējās darbības un komentāri par problēmu, kā arī laiks un lietotāja informācija. Informāciju par ikonām, kas tiek izmantotas lietotāja darbību apzīmēšanai, skatiet lapā notikuma informācija. Ņemiet vērā, ka vēsture tiek parādīta tikai tad, ja atjaunināšanai atlasīta viena problēma.
Tvērums	Definējiet tādu darbību tvērumu kā nopietnības mainīšana, problēmu apstiprināšana vai manuāla aizvēršana: **Tikai atlasītā problēma** - ietekmēs tikai šo notikumu **Atlasītā un visas pārējās saistīto trigeru problēmas** - problēmas apstiprināšanas/aizvēršanas gadījumā ietekmēs šo notikumu un visas pārējās problēmas, kas līdz šim nav apstiprinātas/aizvērtas. Ja tvērumā ir iekļautas jau apstiprinātas vai aizvērtas problēmas, tās netiks atkārtoti apstiprinātas/aizvērtas. Savukārt ziņojumu un nopietnības mainīšanas darbību skaits nav ierobežots.

Parametrs	Apraksts
Mainīt nopietnību	Atzīmējiet izvēles rūtiņu un noklikšķiniet uz nopietnības pogas, lai atjauninātu problēmas nopietnību. Nopietnības mainīšanas izvēles rūtiņa ir pieejama, ja vismaz vienai no atlasītajām problēmām ir lasīšanas un rakstīšanas atļaujas. Noklikšķinot uz *Atjaunināt*, tiks atjauninātas tikai tās problēmas, kurām ir lasīšanas un rakstīšanas atļaujas. Ja nevienam no atlasītajiem trigeriem nav lasīšanas un rakstīšanas atļauju, izvēles rūtiņa ir atspējota.
Apturēt	Atzīmējiet izvēles rūtiņu, lai apturētu problēmu: **Bez termiņa** - apturēt bez termiņa **Līdz** - apturēt līdz norādītajam laikam. Tiek atbalstīti gan absolūtie, gan relatīvie laika formāti, piemēram: `now+1d` - vienu dienu no šī brīža (noklusējums) `now/w` - līdz pašreizējās nedēļas beigām `2022-05-28 12:00:00` - līdz absolūtam datumam/laikam Ņemiet vērā, ka vienkāršs periods (piemēram, `1d`, `1w`) netiek atbalstīts. Šīs opcijas pieejamība ir atkarīga no lietotāja lomas iestatījuma "Apturēt problēmas". Skatiet arī: Problēmu apturēšana
Atcelt apturēšanu	Atzīmējiet izvēles rūtiņu, lai atceltu problēmas apturēšanu. Šī izvēles rūtiņa ir aktīva tikai tad, ja vismaz viena no atlasītajām problēmām ir apturēta. Šīs opcijas pieejamība ir atkarīga no lietotāja lomas iestatījuma "Apturēt problēmas".
Apstiprināt	Atzīmējiet izvēles rūtiņu, lai apstiprinātu problēmu. Šī izvēles rūtiņa ir pieejama, ja starp atlasītajām problēmām ir vismaz viena neapstiprināta problēma. Jau apstiprinātai problēmai nav iespējams pievienot vēl vienu apstiprinājumu (tomēr ir iespējams pievienot vēl vienu komentāru).
Atcelt apstiprināšanu	Atzīmējiet izvēles rūtiņu, lai atceltu problēmas apstiprināšanu. Šī izvēles rūtiņa ir pieejama, ja starp atlasītajām problēmām ir vismaz viena apstiprināta problēma.
Pārvērst par cēloni	Atzīmējiet izvēles rūtiņu, lai simptomātiskās problēmas pārvērstu par cēloņa problēmām.
Aizvērt problēmu	Atzīmējiet izvēles rūtiņu, lai manuāli aizvērtu atlasītās problēmas. Problēmas aizvēršanas izvēles rūtiņa ir pieejama, ja vismaz vienai no atlasītajām problēmām trigera konfigurācijā ir atzīmēta opcija *Atļaut manuālu aizvēršanu*. Noklikšķinot uz *Atjaunināt*, tiks aizvērtas tikai tās problēmas, kuras drīkst aizvērt. Ja nevienu problēmu nevar aizvērt manuāli, izvēles rūtiņa ir atspējota. Jau aizvērtas problēmas netiks atkārtoti aizvērtas.
Apturēt trigeri	Noklikšķiniet, lai sāktu trigerim ad hoc uzturēšanu. Tiks atvērta uzturēšanas perioda konfigurācijas veidlapa. Tajā būs iepriekš atlasīts vienreizējs 1 h ilgs uzturēšanas periods, kas sākas no šī brīža (ilgumu var konfigurēt lietotāja iestatījumos un profilā). Tiks iepriekš atlasīts arī attiecīgais trigeris. Uzturēšanas nosaukums tiks aizpildīts iepriekš formātā `Ad-hoc: <event name>`.

Problēmas apstiprināšana Ja lietotājs saņem paziņojumu par problēmas notikumu, viņš var doties uz Zabbix lietotāja saskarne, atvērt šīs problēmas atjaunināšanas uznirstošo logu, izmantojot kādu no tālāk norādītajiem veidiem, un apstiprināt problēmu. Apstiprināšanas laikā viņš var ievadīt savu komentāru, norādot, ka strādā pie tās, vai jebko citu, ko vēlas par to pateikt.

Tādējādi, ja cits sistēmas lietotājs pamana to pašu problēmu, viņš uzreiz redz, vai tā ir apstiprināta, un līdzšinējos komentārus.

Šādā veidā problēmu novēršanas process, kurā iesaistīts vairāk nekā viens sistēmas lietotājs, var noritēt koordinēti.

Apstiprināšanas statuss tiek izmantots arī, definējot darbību operācijas. Piemēram, varat definēt, ka paziņojums tiek nosūtīts augstāka līmeņa vadītājam tikai tad, ja notikums kādu laiku nav apstiprināts.

Lai apstiprinātu notikumus un pievienotu tiem komentārus, lietotājam ir jābūt vismaz lasīšanas tiesībām attiecīgajiem trigeriem. Lai mainītu problēmas nopietnību vai aizvērtu problēmu, lietotājam ir jābūt lasīšanas un rakstīšanas tiesībām attiecīgajiem trigeriem.

Ir **vairāki** veidi, kā piekļūt problēmas atjaunināšanas uznirstošajam logam, kas ļauj apstiprināt problēmu.

- Varat atlasīt problēmas sadaļā *Monitoring* > *Problems* un pēc tam zem saraksta noklikšķināt uz *Mass update*
- Varat noklikšķināt uz *Update* problēmas kolonnā *Update* šeit:
  - *Dashboards* (*Problems* un *Problems by severity* sīkrīki)
  - *Monitoring* > *Problems*
  - *Monitoring* > *Problems* > *Event details*
- Varat noklikšķināt uz neatrisinātas problēmas šūnas šeit:
  - *Dashboards* (*Trigger overview* sīkrīks)

Uznirstošajā izvēlnē ir opcija *Update*, kas jūs aizvedīs uz problēmas atjaunināšanas logu.

Pamatojoties uz apstiprinājuma informāciju, ir iespējams konfigurēt, kā problēmu skaits tiek parādīts informācijas panelī vai kartēs. Lai to izdarītu, opcijā *Problēmu attēlošana*, kas pieejama gan kartes konfigurācijā, gan informācijas paneļa logrīkā *Problēmas pēc smaguma pakāpes* Problems by severity, ir jāveic atbilstošas izvēles. Ir iespējams parādīt kopējo problēmu skaitu, neapstiprināto problēmu skaitu atsevišķi no kopējā skaita vai tikai neapstiprināto problēmu skaitu.

Pamatojoties uz informāciju par problēmas atjauninājumiem (apstiprināšanu u. c.), ir iespējams konfigurēt atjaunināšanas darbības — nosūtīt ziņojumu vai izpildīt attālās komandas.

Problēmas atlikšana Problēmas atlikšana nodrošina iespēju īslaicīgi paslēpt problēmu, kuru var novērst vēlāk. Tas ir noderīgi, lai sakārtotu problēmu sarakstu un piešķirtu augstāko prioritāti vissteidzamākajām problēmām. Piemēram, dažkārt nedēļas nogalē var rasties problēma, kas nav pietiekami steidzama, lai to novērstu nekavējoties, tāpēc to var atlikt līdz pirmdienas rītam.

Problēmas atlikšana ļauj paslēpt *vienu* problēmu, atšķirībā no atlikšanas, izmantojot hosts uzturēšanu, kad tiek paslēptas visas uzturēšanā esošā hosts problēmas.

Trigeru darbību operācijas atliktajām problēmām tiks apturētas tāpat kā hosts uzturēšanas gadījumā.

Problēmu var apspiest, izmantojot logu **problēmas atjaunināšana**, kur apspiešana ir viena no problēmas atjaunināšanas opcijām līdzās komentēšanai, smaguma pakāpes maiņai, apstiprināšanai u. c.

Problēmu var arī atcelt no apspiešanas, izmantojot to pašu problēmas atjaunināšanas logu.

Attēlošana

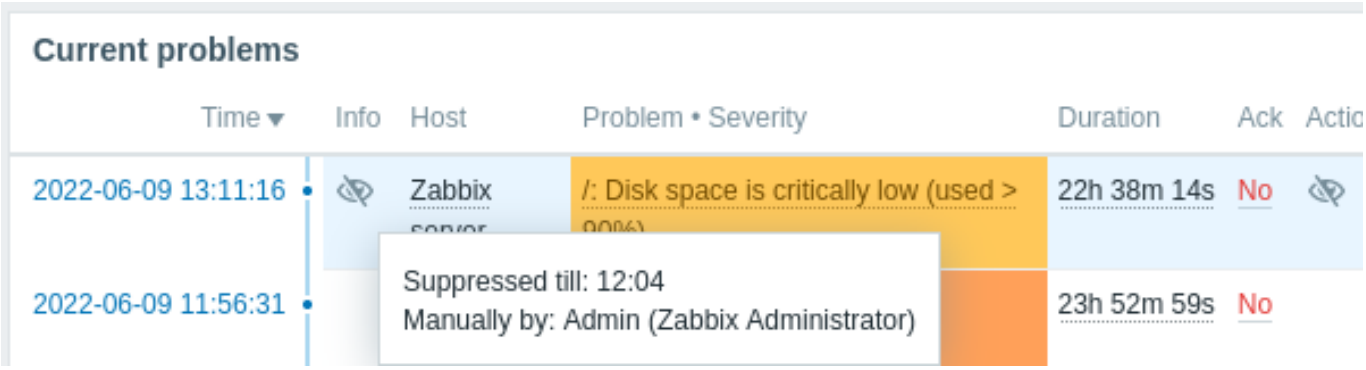
Kad problēma ir apspiesta, tā pirms paslēpšanas tiek atzīmēta ar mirgojošu  apspiešanas ikonu kolonnā *Info*.

Ja apspiešana ir saistīta ar hosts apkopi, apspiešanas ikona kolonnā *Info* tiek rādīta oranžā krāsā. Ja vienlaikus ir spēkā gan apkope, gan manuāla apspiešana, ikonas krāsā prioritāte ir manuālajai apspiešanai.

Apspiešanas ikona mirgo, kamēr apspiešanas uzdevums atrodas gaidīšanas sarakstā. Kad uzdevumu pārvaldnieks ir apspiedis problēmu, ikona pārstāj mirgot. Ja apspiešanas ikona ilgstoši turpina mirgot, tas var norādīt uz servera problēmu, piemēram, ja serveris nedarbojas un uzdevumu pārvaldnieks nevar pabeigt uzdevumu. Tas pats attiecas uz apspiešanas atcelšanu. Īsajā periodā pēc uzdevuma iesniegšanas un pirms serveris to ir pabeidzis, apspiešanas atcelšanas ikona mirgo.

Apspiesta problēma atkarībā no problēmu filtra/logdaļas iestatījumiem var būt gan paslēpta, gan parādīta.

Kad tā tiek rādīta problēmu sarakstā, apspiesta problēma ir atzīmēta ar apspiešanas ikonu, un apspiešanas informācija tiek parādīta, uzvedot peli virs tās:



Apspiešanas informācija tiek rādīta arī uznirstošajā logā, novietojot peli virs apspiešanas ikonas kolonnā *Actions*.

Manuāla problēmu slēgšana

Pārskats Lai gan parasti problēmu notikumi tiek atrisināti automātiski, kad trigeris stāvoklis mainās no `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 jānoregulē optimālai veiktspējai. Šādā gadījumā par problēmu tiek informēti Linux administratori, viņi to novērš un pēc tam manuāli aizver problēmu.

Problēmas var manuāli aizvērt tikai tiem trigeriem, kuriem ir iespējota opcija *Allow manual close*.

Kad problēma tiek ”manuāli aizvērta”, Zabbix serveris ģenerē jaunu iekšējo uzdevumu. Pēc tam process *task manager* izpilda šo uzdevumu un ģenerē OK notikumu, tādējādi aizverot problēmas notikumu.

Ģenerētais *OK* notikums ietver pilnu notikuma tagu kopu, kas ir atrisināta šim notikumam (tostarp tagus, kas mantoti no veidnēm, hostiem un trigeriem). Šie tagi ir pieejami paziņojumos un makrosos, piemēram, `{EVENT.RECOVERY.TAGS}` un `{EVENT.RECOVERY.TAGSJSON}`.

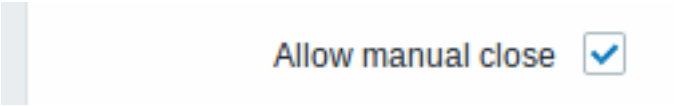
Manuāli aizvērta problēma nenozīmē, ka pamatā esošais trigeris nekad vairs nepāries *Problem* stāvoklī. Trigera izteiksme tiek pārvērtēta un var atkal izraisīt problēmu:

- Kad tiek saņemti jauni dati jebkuram vienumam, kas ir iekļauts trigera izteiksmē (ņemiet vērā, ka vērtības, kas ir atmestas ar throttling pirmapstrādes soli, netiek uzskatītas par saņemtām un neizraisīs trigera izteiksmes pārvērtēšanu);
- Kad izteiksmē tiek izmantotas datuma un laika un/vai nodata() funkcijas.

Konfigurācija Lai manuāli aizvērtu problēmu, ir nepieciešami divi soļi.

Trigeru konfigurācija

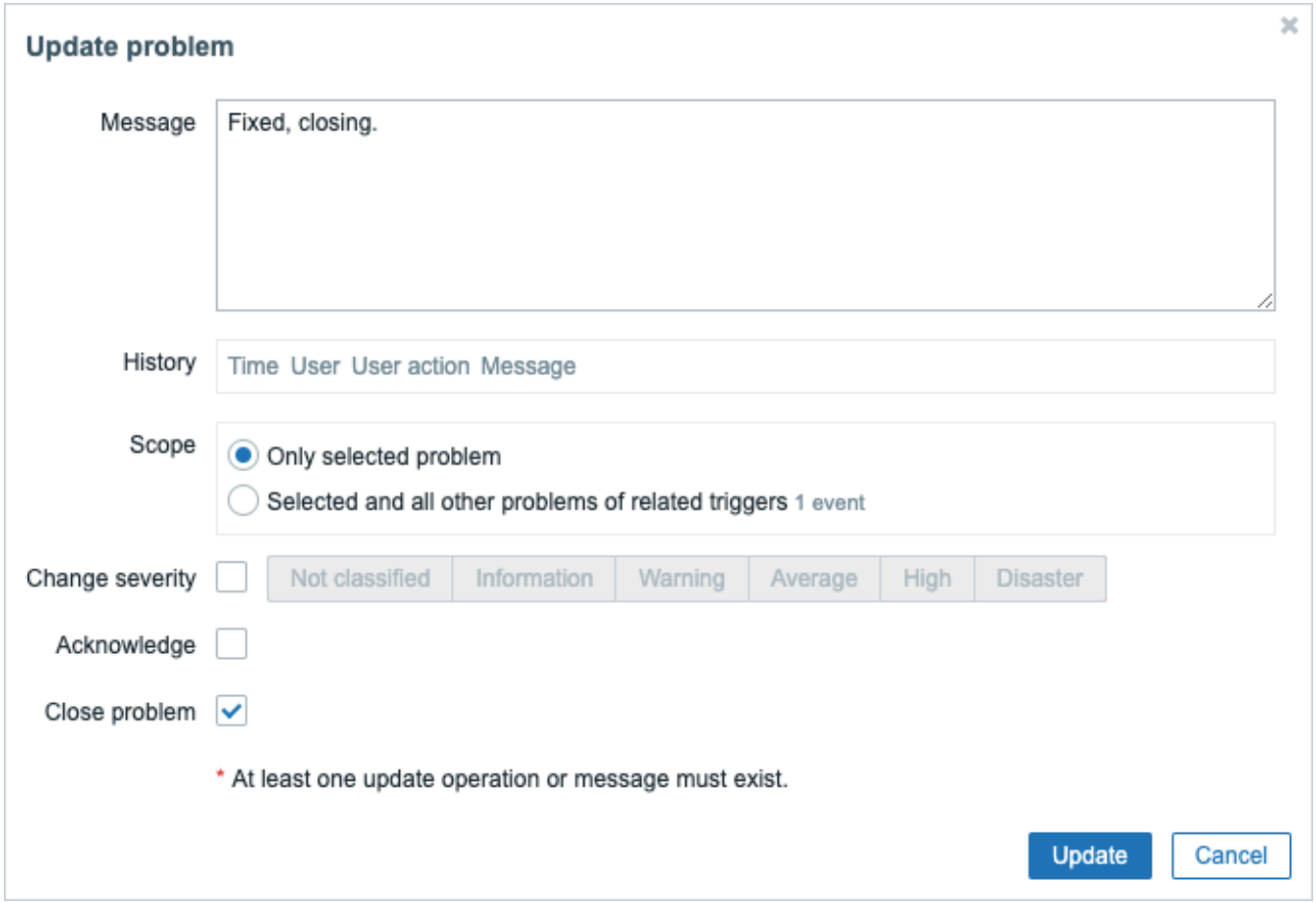
Trigeru konfigurācijā iespējojiet opciju *Allow manual close*.



Problēmas atjaunināšanas logs

Ja rodas problēma trigerim ar *Manual close* karodziņu, varat atvērt šīs problēmas problēmas atjaunināšanas uznirstošo logu un aizvērt problēmu manuāli.

Lai aizvērtu problēmu, atzīmējiet veidlapā 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 *CLOSING* tiek rādīts sadaļā *Monitoring* > *Problems* kā problēmas statuss.

Pārbaude 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.

Uzturēšanas periodi

Pārskats

Apkopes režīmu izmanto, lai iepriekš noteiktos laika periodos apspiestu problēmas.

Apkopes periodus var definēt šīm entītēm:

- Hosts grupas
- Hosts
- Trigeri
- Notikumu nosaukumi
- Tagi

Ja apkope ir definēta konkrētiem tagiem vai notikumu nosaukumiem, apkope tiks aktivizēta tikai atbilstošajiem trigeriem; visi pārējie hosta vai hostu grupas trigeri apkopes režīmā nebūs.

Pastāv divi apkopes veidi: *Ar datu apkopošanu* un *Bez datu apkopošanas*.

Apkopes *Ar datu apkopošanu* laikā trigeri tiek apstrādāti kā parasti un nepieciešamības gadījumā tiek izveidoti notikumi. Tomēr, ja darbības konfigurācijā ir atzīmēta opcija *Apturēt darbības apspiestām problēmām*, problēmu eskalācijas apkopes režīmā esošajām entītēm tiek apturētas. Šādā gadījumā eskalācijas soļi, kas var ietvert paziņojumu nosūtīšanu vai attālinātas komandas, netiks izpildīti, kamēr turpinās apkopes periods. Ņemiet vērā, ka problēmu atkopšanas un atjaunināšanas darbības apkopes laikā netiek apspiestas — tiek apspiestas tikai eskalācijas. Atkopšanas paziņojumi netiek nosūtīti, ja problēma sākās apkopes laikā.

Piemēram, ja eskalācijas soļi ir ieplānoti 0, 30 un 60 minūtes pēc problēmas sākuma un pēc reālas problēmas rašanās tiek aktivizēta pusstundu ilga apkope, kas ilgst no 10. līdz 40. minūtei, otrais un trešais solis tiks izpildīts pusstundu vēlāk — attiecīgi 60. un 90. minūtē (ja problēma joprojām pastāv). Līdzīgi, ja problēma rodas apkopes laikā, eskalācija sāksies pēc apkopes beigām.

Lai apkopes laikā saņemtu problēmu paziņojumus kā parasti (bez aizkaves), darbības konfigurācijā ir jānoņem atzīme no opcijas *Apturēt darbības apspiestām problēmām*.

> **Note:**
> Ja vismaz viens hosts (kas tiek izmantots trigera izteiksmē) neatrodas apkopes režīmā, Zabbix nosūtīs problēmas paziņojumu.

Apkopes laikā Zabbix serverim ir jādarbojas. Apkopes tiek pārrēķinātas katru minūti vai tiklīdz konfigurācijas kešatmiņa tiek atkārtoti ielādēta, ja apkopes periodā ir veiktas izmaiņas.

Taimera procesi pārbauda, vai hosta statuss ir jāmaina uz apkopes režīmu vai no tā katras minūtes 0. sekundē. Turklāt katru sekundi taimera process pārbauda, vai kāda apkope ir jāsāk vai jāaptur, pamatojoties uz izmaiņām [apkopes periodos] pēc konfigurācijas atjaunināšanas. Tādēļ apkopes periodu sākšanas un apturēšanas ātrums ir atkarīgs no konfigurācijas atjaunināšanas intervāla (pēc noklusējuma — 10 sekundes). Ņemiet vērā, ka apkopes perioda izmaiņas neietver iestatījumus *Aktīvs no/Aktīvs līdz*. Turklāt, ja entītija (hostu grupa/hosts/trigeris u. c.) tiek pievienota jau aktīvam apkopes periodam, izmaiņas taimera process aktivizēs tikai nākamās minūtes sākumā.

Ņemiet vērā, ka, hostam pārejot apkopes režīmā, Zabbix servera taimera procesi nolasīs visas atvērtās problēmas, lai pārbaudītu, vai tās ir jāapslāpē. Tas var ietekmēt veiktspēju, ja ir daudz atvērtu problēmu. Zabbix serveris visas atvērtās problēmas nolasīs arī startēšanas laikā, pat ja tajā brīdī nav konfigurēta neviena apkope.

Ņemiet vērā, ka Zabbix serveris (vai starpniekserveris) vienmēr apkopo datus neatkarīgi no apkopes veida (tostarp apkopes *Bez datu apkopošanas* laikā). Ja ir iestatīta opcija *Bez datu apkopošanas*, serveris šos datus vēlāk ignorē.

Kad beidzas apkope *Bez datu apkopošanas*, trigeri, kas izmanto funkciju `nodata()`, neaktivizēsies līdz nākamajai pārbaudei periodā, kuru tie pārbauda.

Ja žurnāla vienums tiek pievienots laikā, kad hosts atrodas apkopes režīmā, un apkope beidzas, tiks apkopoti tikai jaunie žurnālfaila ieraksti, kas pievienoti pēc apkopes beigām.

Ja hostam, kas atrodas apkopes *Bez datu apkopošanas* režīmā, tiek nosūtīta vērtība ar laika zīmogu (piemēram, izmantojot Zabbix sender), šī vērtība tiks atmesta; tomēr ir iespējams nosūtīt vērtību ar laika zīmogu par beigušos apkopes periodu, un tā tiks pieņemta.

Ja lietotājs maina apkopes periodu vai apkopes opciju (hostu grupu, hostu, trigeri u. c.), izmaiņas stāsies spēkā tikai pēc konfigurācijas kešatmiņas sinhronizācijas.

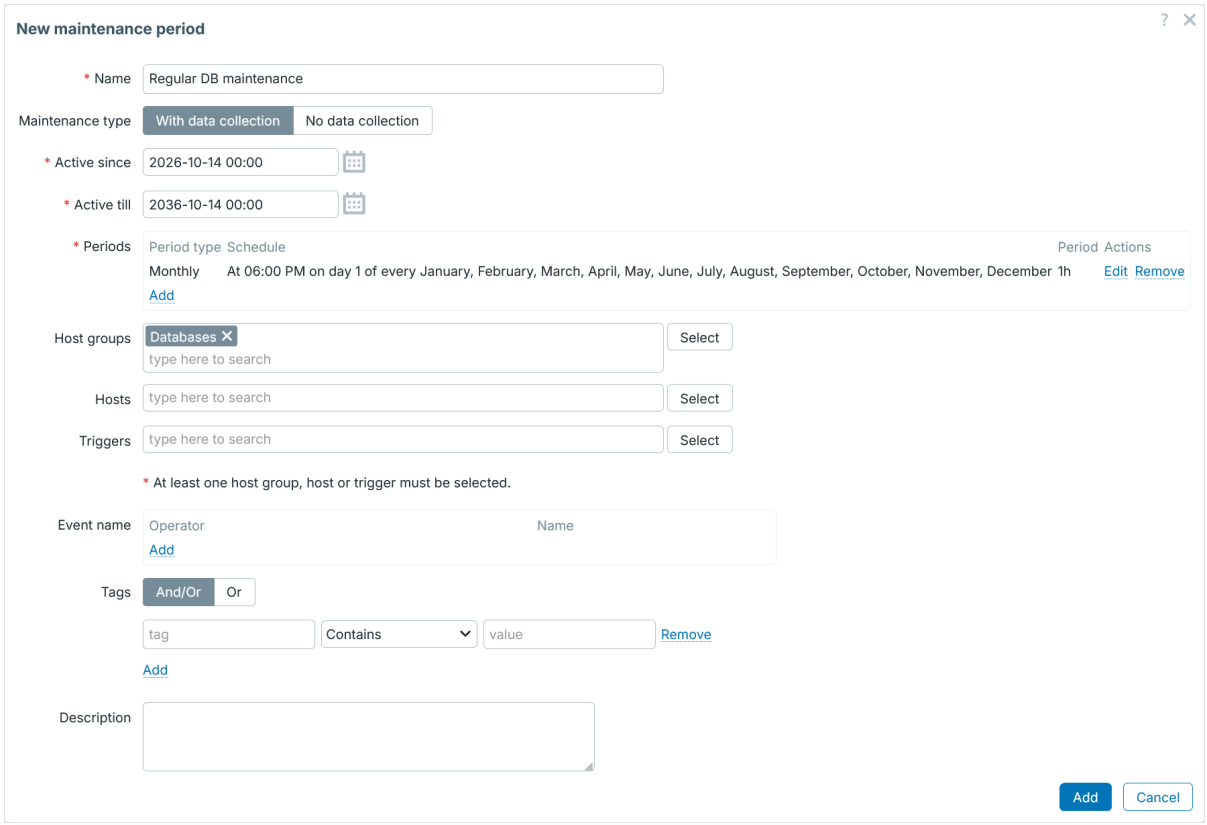
Konfigurēšana

Lai nekavējoties sāktu uzturēšanas periodu:

1. Dodieties uz *Monitoring* > *Problems*.
2. Noklikšķiniet uz problēmas nosaukuma.
3. Notikumu izvēlnē atlasiet uzturēšanas opciju (hosts, trigeris, notikuma nosaukums, tagi).
4. Veidlapā mainiet uzturēšanas parametrus.

Lai konfigurētu plānoto uzturēšanas periodu:

1. Dodieties uz *Data collection* > *Maintenance*.
2. Noklikšķiniet uz *Create maintenance period* (vai uz esoša uzturēšanas perioda nosaukuma).
3. Veidlapā ievadiet uzturēšanas parametrus.



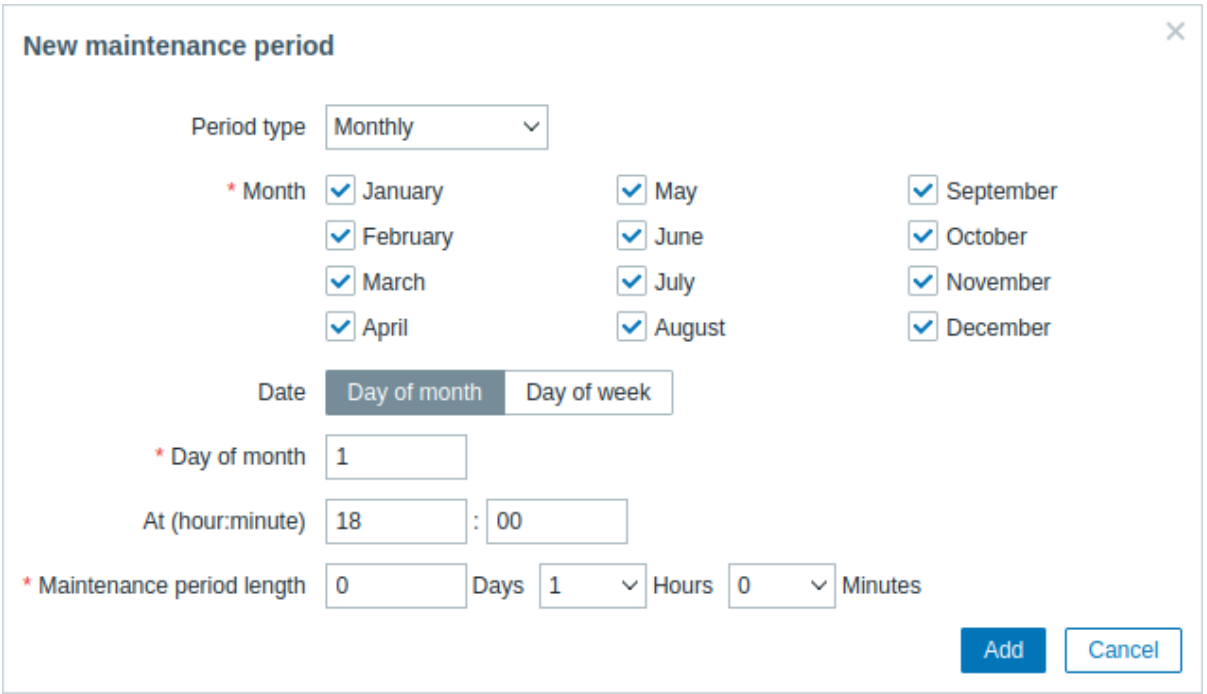
Visi obligātie ievades lauki ir atzīmēti ar sarkanu zvaigznīti.

Parameter	Description
Name	Uzturēšanas perioda nosaukums.
Maintenance type	Var iestatīt divus uzturēšanas veidus: **With data collection** - serveris uzturēšanas laikā apkopo datus, un trigeri tiek apstrādāti; **No data collection** - dati joprojām var tikt apkopoti, taču uzturēšanas laikā tie netiek saglabāti datubāzē un trigeri (tostarp funkcija `nodata()`) netiek aktivizēti. Informāciju par to, kā katrs veids ietekmē pieejamības pārskatu, skatiet sadaļā Uzturēšanas periodu ietekme.
Active since	Datums un laiks, kad uzturēšanas periodu izpilde kļūst aktīva. *Piezīme:* Tikai šī laika iestatīšana neaktivizē uzturēšanas periodu; uzturēšanas periodi ir jākonfigurē sadaļā *Periods* (skatiet tālāk).
Active till	Datums un laiks, kad uzturēšanas periodu izpilde vairs nav aktīva.
Periods	Šis bloks ļauj definēt precīzas dienas un stundas, kad notiek uzturēšana. Noklikšķinot uz Add, tiek atvērts uznirstošais logs ar elastīgu *Maintenance period* veidlapu, kurā varat definēt uzturēšanas grafiku. Detalizētu aprakstu skatiet sadaļā Uzturēšanas periodi.
Host groups	Atlasiet hosts grupas, kurām tiks aktivizēta uzturēšana. Uzturēšana tiks aktivizēta visiem hosts no norādītās hosts grupas vai grupām. Šis lauks atbalsta automātisko pabeigšanu, tāpēc, sākot tajā rakstīt, tiks parādīts visu pieejamo hosts grupu nolaižamais saraksts. Norādot vecāka hosts grupu, netieši tiek atlasītas visas ligzdotās hosts grupas. Tādējādi uzturēšana tiks aktivizēta arī hosts no ligzdotajām grupām.

Parameter	Description
Hosts	Atlasiet hosts, kuriem tiks aktivizēta uzturēšana. Šis lauks atbalsta automātisko pabeigšanu, tāpēc, sākot tajā rakstīt, tiks parādīts visu pieejamo hosts nolaižamais saraksts.
Triggers	Atlasiet trigerus, kuriem tiks aktivizēta uzturēšana. Šis lauks atbalsta automātisko pabeigšanu, tāpēc, sākot tajā rakstīt, tiks parādīts visu pieejamo trigeru nolaižamais saraksts.
Event name	Norādiet notikumu nosaukumus, kuriem tiks aktivizēta uzturēšana. Var iestatīt vairākus nosacījumus. Katram nosacījumam ir pieejami divi operatori: **Contains** - iekļaut norādītos tagu nosaukumus, ja tagu vērtības satur ievadīto virkni (apakšvirknes atbilstība, reģistrjutība netiek ņemta vērā) **Does not contain** - izslēgt norādītos tagu nosaukumus, ja tagu vērtības satur ievadīto virkni (apakšvirknes atbilstība, reģistrjutība netiek ņemta vērā)
Tags	Norādiet tagus, kuriem tiks aktivizēta uzturēšana. Var iestatīt vairākus nosacījumus. Tagu nosaukumu atbilstība vienmēr ir reģistrjutīga. Katram nosacījumam ir pieejami vairāki operatori: **Contains** - iekļaut norādītos tagu nosaukumus, ja tagu vērtības satur ievadīto virkni (apakšvirknes atbilstība, reģistrjutīga); **Equals** - iekļaut norādītos tagu nosaukumus un vērtības (reģistrjutīgi). **Does not equal** - izslēgt norādītos tagu nosaukumus un vērtības (reģistrjutīgi) **Does not contain** - izslēgt norādītos tagu nosaukumus, ja tagu vērtības satur ievadīto virkni (apakšvirknes atbilstība, reģistrjutība netiek ņemta vērā) Nosacījumiem ir pieejami divi aprēķina veidi: **And/Or** - jāizpilda visi nosacījumi; nosacījumi ar vienādu taga nosaukumu tiks grupēti, izmantojot nosacījumu Or; **Or** - pietiek, ja tiek izpildīts viens nosacījums. Tagus var norādīt tikai tad, ja ir atlasīts uzturēšanas veids *With data collection*.
Description	Uzturēšanas perioda apraksts.

Uzturēšanas periodi

Uzturēšanas perioda logs ir paredzēts atkārtotas vai vienreizējas uzturēšanas plānošanai. Veidlapa ir dinamiska, un pieejamie lauki mainās atkarībā no atlasītā *Perioda veida*.



Perioda veids	Apraksts
Tikai vienu reizi	Konfigurējiet vienreizēju uzturēšanas periodu: *Datums* - uzturēšanas perioda datums un laiks; *Uzturēšanas perioda ilgums* - cik ilgi uzturēšana būs aktīva.
Katru dienu	Konfigurējiet ikdienas uzturēšanas periodu: *Ik pēc dienām* - uzturēšanas biežums (1 - *(noklusējums)* katru dienu, 2 - ik pēc divām dienām utt.); *Plkst. (stunda:minūte)* - diennakts laiks, kad sākas uzturēšana; *Uzturēšanas perioda ilgums* - cik ilgi uzturēšana būs aktīva. Ja parametrs *Ik pēc dienām* ir lielāks par ”1”, sākuma diena ir diena, kurā iekrīt laiks *Aktīvs no*. Piemēri: - ja *Aktīvs no* ir iestatīts uz ”2021-01-01 12:00”, *Ik pēc dienām* ir iestatīts uz ”2”, un *Plkst. (stunda:minūte)* ir iestatīts uz ”23:00”, tad pirmais uzturēšanas periods sāksies 1. janvārī plkst. 23:00, savukārt otrais uzturēšanas periods sāksies 3. janvārī plkst. 23:00; - ja *Aktīvs no* ir iestatīts uz ”2021-01-01 12:00”, *Ik pēc dienām* ir iestatīts uz ”2”, un *Plkst. (stunda:minūte)* ir iestatīts uz ”01:00”, tad pirmais uzturēšanas periods sāksies 3. janvārī plkst. 01:00, savukārt otrais uzturēšanas periods sāksies 5. janvārī plkst. 01:00.
Katru nedēļu	Konfigurējiet iknedēļas uzturēšanas periodu: *Ik pēc nedēļām* - uzturēšanas biežums (1 - *(noklusējums)* katru nedēļu, 2 - ik pēc divām nedēļām utt.); *Nedēļas diena* - kurā dienā uzturēšanai jānotiek; *Plkst. (stunda:minūte)* - diennakts laiks, kad sākas uzturēšana; *Uzturēšanas perioda ilgums* - cik ilgi uzturēšana būs aktīva. Ja parametrs *Ik pēc nedēļām* ir lielāks par ”1”, sākuma nedēļa ir nedēļa, kurā iekrīt laiks *Aktīvs no*. Piemērus skatiet iepriekšējā parametra *Katru dienu* aprakstā.
Katru mēnesi	Konfigurējiet ikmēneša uzturēšanas periodu: *Mēnesis* - atlasiet visus mēnešus, kuros tiek veikta regulārā uzturēšana; *Datums:* ***Mēneša diena*** - atlasiet šo opciju, ja uzturēšanai jānotiek katru mēnesi vienā un tajā pašā datumā (piemēram, katra mēneša 1. datumā), un pēc tam laukā *Mēneša diena*, kas parādās, atlasiet vajadzīgo dienu; *Datums:* ***Nedēļas diena*** - atlasiet šo opciju, ja uzturēšanai jānotiek tikai noteiktās dienās (piemēram, katra mēneša pirmajā pirmdienā), pēc tam nolaižamajā sarakstā atlasiet vajadzīgo mēneša nedēļu (pirmā, otrā, trešā, ceturtā vai pēdējā) un pēc tam atzīmējiet uzturēšanas dienas izvēles rūtiņas; *Plkst. (stunda:minūte)* - diennakts laiks, kad sākas uzturēšana; *Uzturēšanas perioda ilgums* - cik ilgi uzturēšana būs aktīva.

Attention:
Veidojot apkopes periodu, tiek izmantota tā lietotāja laika josla, kurš to izveido. Tomēr, plānojot atkārtotus apkopes periodus (*Daily*, *Weekly*, *Monthly*), tiek izmantota Zabbix servera laika josla. Lai nodrošinātu paredzamu atkārtotu apkopes periodu darbību, ir nepieciešams visām Zabbix daļām izmantot vienu kopīgu laika joslu.

Kad tas ir izdarīts, noklikšķiniet uz *Add*, lai pievienotu uzturēšanas periodu sadaļai *Periods*.

Ņemiet vērā, ka vasaras laika (DST) izmaiņas 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 DST maiņa 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 DST maiņa un pašreizējais laiks mainās no 03:00 uz 02:00, uzturēšana tiks pārtraukta, jo ir pagājušas divas stundas;
- ja uzturēšanas periods sākas stundā, kas tiek izlaista DST maiņas dēļ, tad uzturēšana nesāksies.

Ja uzturēšanas periods ir iestatīts uz ”1 day” (faktiskais uzturēšanas periods ir 24 stundas, jo Zabbix aprēķina dienas stundās), sākas plkst. 00:00 un beidzas plkst. 00:00 nākamajā dienā:

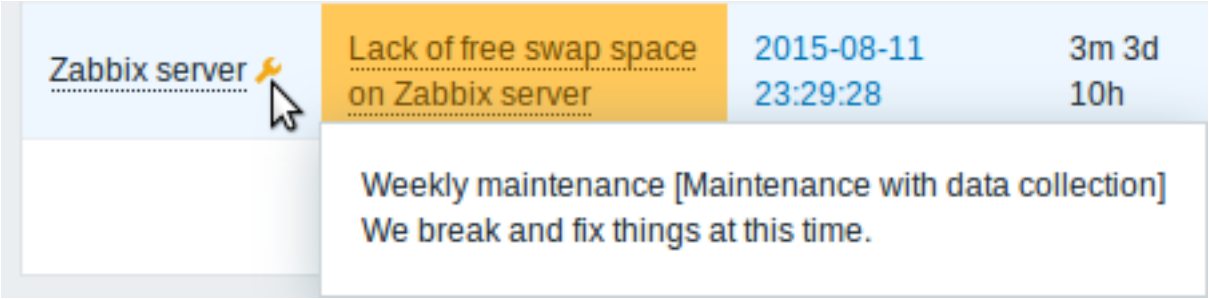
- uzturēšana beigsies plkst. 01:00 nākamajā dienā, ja pašreizējais laiks tiek pārbīdīts par vienu stundu uz priekšu;
- uzturēšana beigsies plkst. 23:00 tajā pašā dienā, ja pašreizējais laiks tiek pārbīdīts par vienu stundu atpakaļ.

Attēlošana

Hosts attēlošana uzturēšanas režīmā

Oranža uzgriežņu atslēgas ikona blakus hosta nosaukumam norāda, ka šis hosts ir uzturēšanas režīmā šādās vietās:

- *Paneļi*
- *Uzraudzība* > *Problēmas*
- *Inventārs* > *Hosts* > *Hosta inventāra informācija*
- *Datu vākšana* > *Hosts* (skatiet kolonnu *Statuss*)



Uzturēšanas informācija tiek parādīta, kad peles rādītājs tiek novietots virs ikonas.

Turklāt hostiem uzturēšanas režīmā *Uzraudzība* > *Kartes* sadaļā tiek piešķirts oranžs fons.

Slēpto problēmu attēlošana

Parasti problēmas hosts uzturēšanas režīmā tiek slēptas, tas ir, tās netiek rādītas lietotāja saskarnē.

Slēptās problēmas var būt arī paslēptas tāpēc, ka tās tika slēptas manuāli.

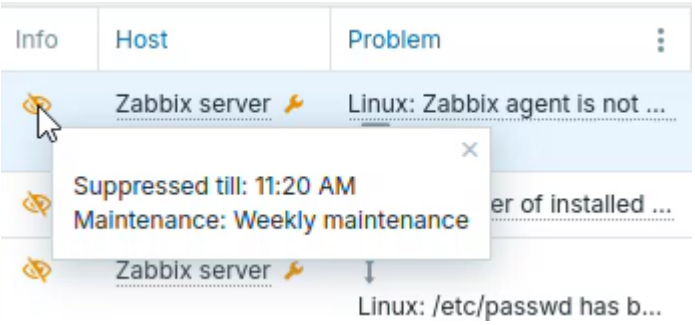
Slēptās problēmas ir iespējams attēlot, atlasot opciju *Rādīt slēptās problēmas* šajās vietās:

- *Informācijas paneļi* (sadaļā *Problemātiskie hosts*, *Problēmas*, *Problēmas pēc smaguma pakāpes*, *Trigeru pārskats* logrīka konfigurācijā)
- *Uzraudzība* > *Problēmas* (filtrā)
- *Uzraudzība* > *Kartes* (kartes konfigurācijā)
- Globālās paziņojumi (lietotāja profila konfigurācijā)

Kad slēptās problēmas tiek attēlotas, tiek rādīta šāda ikona: . Ja slēpšana ir saistīta ar hosta uzturēšanas režīmu, tiek rādītas šādas ikonas:

- - problēma ir slēpta hosta uzturēšanas režīma dēļ.
- - problēma pēc hosta uzturēšanas režīma ir vairs neslēpta.

Novietojot kursoru virs ikonas, tiek parādīta papildu informācija. Notikuma detaļās var pārbaudīt, vai notikums tika slēpts uzturēšanas režīma dēļ vai slēpts manuāli. Ja attiecas gan uzturēšanas režīms, gan manuāla slēpšana, ikonas krāsā prioritāte ir manuālai slēpšanai.



Rindas apkopes laikā

Rindas, kas tiek parādītas Zabbix lietotāja saskarnē (Administration > Queue), aprēķina Zabbix serveris. Tās neietver vienumus apkopes režīmā *No data collection*, un šo vienumu rindas garums vienmēr ir nulle, pat ja to vērtības kavējas. Aizkavētie vienumi apkopes režīmā *With data collection* joprojām tiek ieskaitīti rindā.

Zabbix starpniekserveris nezina par apkopes periodiem, jo starp Zabbix serveri un starpniekserveri nenotiek apkopes konfigurācijas sinhronizācija. Iekšējās pārbaudes, kas tiek aprēķinātas Zabbix starpniekserveros (piemēram, `zabbix[queue,,]` un `zabbix[stats,,,queue,,]`), ziņo par aizkavētiem vienumiem neatkarīgi no apkopes statusa Zabbix serverī.

Tāpēc Zabbix lietotāja saskarne un iekšējās pārbaudes Zabbix starpniekserveros var ziņot par atšķirīgu rindas garumu tiem pašiem vienumiem apkopes režīmā *No data collection*.

Tagošana

Tagi kalpo vairākiem mērķiem, jo īpaši, lai atzīmētu notikumus. Kad entītijas ir apzīmētas ar tagiem, jebkurš jauns notikums, kas saistīts ar apzīmētu entītiju, pārmantos tās tagus. Piemēram:

- ar apzīmētām veidnēm - jebkura hosts problēma (ko izveido trigeri no veidnes) pārmantos veidnes tagus.
- ar apzīmētiem hostiem - jebkura hosts problēma pārmantos hosta tagus.
- ar apzīmētiem vienumiem/web scenārijiem - jebkura vienuma/web scenārija problēma pārmantos vienuma/web scenārija tagus.
- ar apzīmētiem trigeriem - jebkura problēma, ko izveido trigeris, pārmantos trigera tagus.

Problēmas notikums pārmanto visus tagus no visas entītiju ķēdes - veidnēm, hostiem, vienumiem/web scenārijiem, trigeriem. Identiskas `tag:value` kombinācijas (pēc makro atrisināšanas) tiek apvienotas vienā, tādējādi novēršot dublēšanos.

Atjaunošanas notikumi, kas ģenerēti ar manuālu aizvēršanu, arī ietver atrisinātos notikuma tagus, kas pārmantoti no veidnēm, hostiem, vienumiem/web scenārijiem un trigeriem. Šie tagi ir pieejami paziņojumos un, izmantojot makro, piemēram, `{EVENT.RECOVERY.TAGS}` un `{EVENT.RECOVERY.TAGSJSON}`.

Pielāgotie notikumu tagi piedāvā lielāku elastību. Piemēram:

- notikumu korelāciju var konfigurēt, pamatojoties uz notikumu tagiem.
- darbību nosacījumus var konfigurēt, pamatojoties uz notikumu tagiem.
- vienumu problēmas var grupēt, pamatojoties uz notikumu tagiem.
- problēmu tagus var izmantot, lai kartētu problēmas uz pakalpojumiem.

Entītijas var tikt apzīmētas ar vienu un to pašu taga nosaukumu, bet ar atšķirīgām taga vērtībām (piemēram, `component:memory` un `component:storage`). Tāpat entītijai var būt tags bez vērtības un tas pats tags ar vērtību (piemēram, `database` un `database:postgresql`). Šādi tagi netiek uzskatīti par dublikātiem.

Tagi sastāv no taga nosaukuma un taga vērtības. Apzīmējot entītijas, var izmantot tikai nosaukumu vai arī pievienot tam vērtību (piemēram, `mysql`, `jira`, `target:mysql`, `service:jira` u. c.).

Tagus var definēt dažādām entītijām:

- Veidnes
- Hosts
- Vienumi
- Web scenāriji
- Trigeri
- Pakalpojumi
- Veidņu vienumi un trigeri
- Hostu, vienumu un trigera prototipi

Sadaļās Veidnes, *Hosts*, *Vienumi*, *Trigeri*, *Web scenāriji* un to prototipos kolonna *Tagi* parāda gan parastos, gan pārmantotos tagus. Novietojot kursoru virs *pārmantota* taga vai noklikšķinot uz tā, tiek parādīts JavaScript padoms ar tekstu "Inherited tag". Ja pārmantots tags pastāv arī kā parasts tags, tiek rādīts atšķirīgs padoma teksts atkarībā no saraksta, kurā tas parādās (piemēram, "Inherited and template tag" veidņu sarakstā vai "Inherited and host tag" hostu sarakstā). Citās lietotāja saskarnes vietās parastie un pārmantotie tagi tiek rādīti kopā (dublikāti tiek noņemti), bez ikonām un bez papildu padoma teksta.

> **Note:**
> Skatiet oficiālās Zabbix vadlīnijas par vispārīgiem ieteikumiem tagu definēšanai, kā arī konkrētus norādījumus par veidnēm, vienumiem, trigeriem un zemā līmeņa atklāšanas noteikumiem.

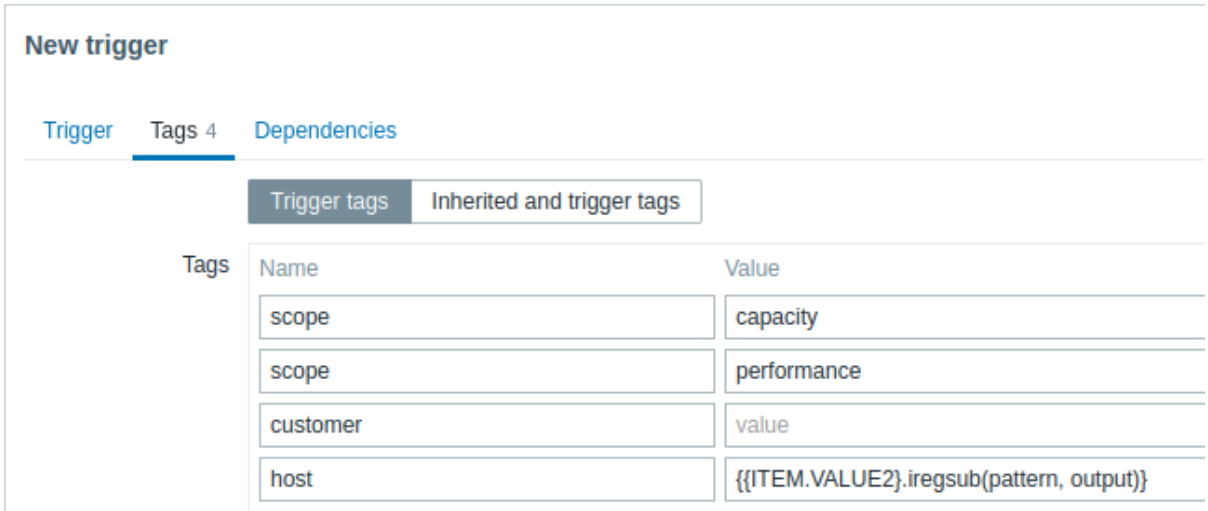
Lietošanas gadījumi Daži biežāk sastopami tagu lietošanas gadījumi ir šādi:

1. Atzīmēt trigeru notikumus:
   - Definējiet trigeris tagu (piemēram, `scope:performance`).
   - Problēmām, kas izveidotas ar šo trigeris, būs trigeris tags.
2. Atzīmēt no veidnes mantotās problēmas:
   - Definējiet veidnes tagu (piemēram, `target:mysql`).
   - Problēmām, kas izveidotas ar trigeriem no šīs veidnes, būs veidnes tags.
3. Atzīmēt hosts problēmas:
   - Definējiet hosta tagu (piemēram, `service:jira`).
   - Problēmām, kas izveidotas ar trigeriem no šī hosta, būs hosta tags.
4. Filtrēt saistītos vienumus:

- Definējiet vienuma tagu (piemēram, `component:cpu`).
- Sadaļā *Monitoring* > *Latest data* vienumus var filtrēt pēc taga `component:cpu`.

5. Izmantot no vienuma vērtības iegūtu informāciju kā taga vērtību:
   - Definējiet tagu ar makro kā taga vērtību (piemēram, `tag-name:{{ITEM.VALUE<N>}.regsub()}`).
   - Sadaļā *Monitoring* > *Problems* problēmām taga vērtība tiks aizstāta ar datiem, kas iegūti no vienuma vērtības.
6. Identificēt problēmas žurnāla failā un aizvērt tās atsevišķi:
   - Definējiet trigeris tagu žurnāla uzraudzības vienuma trigerim, kas, izmantojot makro, no vienuma vērtības izvilks vērtības (piemēram, `service:{{ITEM.VALUE<N>}.regsub()}`).
   - trigeris konfigurācijā iestatiet notikumu korelāciju:
     - iestatiet *PROBLEM event generation mode* uz "Multiple";
     - iestatiet *OK event closes* uz "All problems if tag values match";
     - iestatiet atbilstības tagu.
   - Problēmām, kas izveidotas ar žurnāla vienuma trigeri, būs trigeris tags, un tās tiks aizvērtas atsevišķi.
7. Filtrēt paziņojumus:
   - Definējiet trigeris tagus (piemēram, `scope:security` trigerim1 un `scope:availability` trigerim2).
   - Izmantojiet tagu filtrēšanu darbību nosacījumos, lai saņemtu paziņojumus tikai par notikumiem, kas atbilst tagu datiem.
8. Identificēt problēmas paziņojumos:
   - Definējiet trigeris tagus.
   - Paziņojumā par problēmu izmantojiet makro {EVENT.TAGS}.
   - Paziņojums par problēmu saturēs trigeris tagus, padarot vieglāku noteikt, kurai lietotnei/pakalpojumam paziņojums pieder.
9. Vienkāršot konfigurācijas uzdevumus, izmantojot veidnes tagus:
   - Definējiet veidnes trigeris tagu.
   - Trigeriem, kas izveidoti no šī veidnes trigeris, būs tā tags.
10. Izveidot trigerus ar tagiem no zemā līmeņa atklāšanas (LLD):
    - Definējiet trigeru prototipa tagu ar LLD makro taga nosaukumā vai vērtībā (piemēram, `scope:{#FSNAME}`).
    - Trigeriem, kas izveidoti no trigeru prototipa, būs tā tags.
11. Saskaņot pakalpojumus, izmantojot pakalpojumu tagus:
    - Definējiet pakalpojumu tagus.
    - Konfigurējiet pakalpojumu darbības pakalpojumiem ar atbilstošiem tagiem.
    - Papildus izmantojiet pakalpojumu tagus, lai saistītu pakalpojumu ar SLA SLA aprēķiniem.
12. Saistīt pakalpojumus ar problēmām, izmantojot pakalpojumu problēmu tagus:
    - Definējiet problēmas tagu pakalpojuma konfigurācijā (piemēram, `target:mysql`).
    - Problēmas ar atbilstošu tagu tiks automātiski korelētas ar pakalpojumu, un pakalpojuma statuss mainīsies atbilstoši konfigurētajiem pakalpojuma statusa aprēķina noteikumiem.
13. Slāpēt problēmas, kad hosts ir uzturēšanas režīmā:
    - Definējiet tagus uzturēšanas perioda konfigurācijā.
    - Problēmas ar definētajiem tagiem tiks slāpētas.
14. Piešķirt piekļuvi lietotāju grupām:
    - Definējiet tagus lietotāju grupas konfigurācijā.
    - Lietotāji lietotāju grupā varēs skatīt tikai problēmas ar definētajiem tagiem.

Konfigurācija Tagus var definēt atsevišķā cilnē, piemēram, trigeru konfigurācijā:



Makro atbalsts Iebūvētie un lietotāja makro tagos tiek atrisināti notikuma laikā.
Līdz notikuma iestāšanās brīdim šie makro Zabbix lietotāja saskarne tiks rādīti kā neatrisināti.

Zema līmeņa atklāšanas makro tiek atrisināti atklāšanas procesa laikā.

Tālāk norādītos makro 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 makro
- {INVENTORY.*} iebūvētie makro (host inventāra vērtību atsaukšanai no viena vai vairākiem hosts trigeru izteiksmē)
- Lietotāja makro un lietotāja makro ar kontekstu (konteksts var ietvert zema līmeņa atklāšanas makro)
- Zema līmeņa atklāšanas makro (tikai trigeru prototipa tagos)

Tālāk norādītos makro var izmantot veidnes, host un vienuma/web scenārija tagu nosaukumos un vērtībās:

- {HOST.HOST}, {HOST.NAME}, {HOST.CONN}, {HOST.DNS}, {HOST.IP}, {HOST.PORT} un {HOST.ID} iebūvētie makro
- {INVENTORY.*} iebūvētie makro
- Lietotāja makro
- Zema līmeņa atklāšanas makro (tikai host un vienuma prototipa tagos)

Tālāk norādītos makro var izmantot paziņojumos, kas balstīti uz trigeriem:

- {EVENT.TAGS} un {EVENT.RECOVERY.TAGS} iebūvētie makro (šie makro tiks atrisināti kā ar komatiem atdalīts notikuma tagu vai atkopšanas notikuma tagu saraksts)
- {EVENT.TAGSJSON} un {EVENT.RECOVERY.TAGSJSON} iebūvētie makro (šie makro tiks atrisināti kā JSON masīvs, kas satur notikuma taga objektus vai atkopšanas notikuma taga objektus)

Apakšvirknes izvilkšana trigeru tagos

Apakšvirknes izvilkšana tiek atbalstīta taga nosaukuma vai taga vērtības aizpildīšanai, 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)}
```

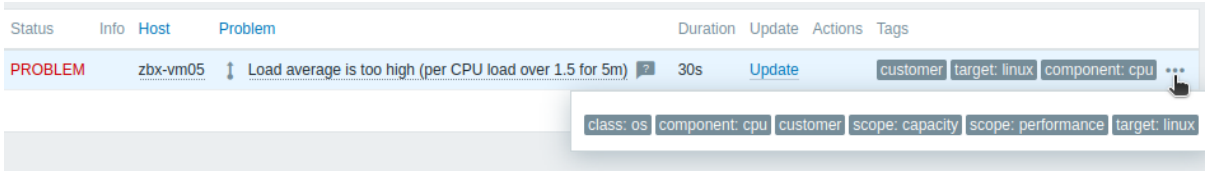
Ja taga nosaukums vai vērtība pēc makro atrisināšanas pārsniedz 255 rakstzīmes, tā tiks saīsināta līdz 255 rakstzīmēm.

Skatiet arī: Makro funkciju izmantošana zemlīmeņa atklāšanas makro notikumu tagu pievienošanai.

Notikumu tagu skatīšana Tagus, ja tie ir definēti, var skatīt kopā ar jauniem notikumiem šeit:

- *Monitoring* > *Problems*
- *Monitoring* > *Problems* > *Event details*
- *Dashboards* > *Problems* widget

Parādīto tagu secību un skaitu nosaka filtrēšanas opcijas *Tag display priority* un *Show tags* sadaļā *Monitoring* > *Problems* vai *Problems* informācijas paneļa logrīkā. Ņemiet vērā, ka var tikt parādīti ne vairāk kā trīs tagi; ja tagu ir vairāk, virzot kursoru virs trim punktiem, visi tagi tiek parādīti uznirstošajā logā.



Notikumu korelācija

Pārskats

Notikumu korelācija ļauj ļoti precīzi un elastīgi sasaistīt problēmu notikumus ar to atrisinājumu.

Notikumu korelāciju var definēt:

- Trigeru līmenī - vienu trigeri var izmantot, lai sasaistītu atsevišķas problēmas ar to risinājumu;
- Globāli - problēmas var sasaistīt ar to risinājumu, izmantojot citu trigeri vai aptaujas metodi un globālos korelācijas noteikumus.

Uz trigeriem balstīta notikumu korelācija

Pārskats

Uz trigeriem balstīta notikumu korelācija ļauj korelēt atsevišķas problēmas, par kurām ziņo viens trigeris.

Parasti OK notikums var aizvērt visus problēmu notikumus, ko izveidojis viens trigeris, taču ir gadījumi, kad nepieciešama detalizētāka pieeja. Piemēram, pārraugot žurnālfailus, iespējams, vēlēsities atklāt noteiktas problēmas žurnālfailā un aizvērt tās atsevišķi, nevis visas kopā.

Tas attiecas uz trigeriem, kuru parametram *PROBLEM event generation mode* ir iestatīta vērtība *Multiple*. Šādus trigerus parasti izmanto žurnālu pārraudzībai, trapu apstrādei u. c.

Zabbix ir iespējams saistīt problēmu notikumus, pamatojoties uz tagging. Tagus izmanto vērtību izgūšanai un problēmu notikumu identificēšanai. Izmantojot šo iespēju, problēmas var arī aizvērt atsevišķi, pamatojoties uz atbilstošu tagu.

Citiem vārdiem, viens un tas pats trigeris var izveidot atsevišķus notikumus, kas identificēti pēc notikuma taga. Tādējādi problēmu notikumus var identificēt pa vienam un aizvērt atsevišķi, pamatojoties uz identifikāciju pēc notikuma taga.

Kā tas darbojas

Žurnālu uzraudzībā jūs varat sastapties ar rindām, kas līdzīgas šīm:

```
1. rinda: Pakalpojums 1 apturēts
2. rinda: Pakalpojums 2 apturēts
3. rinda: Pakalpojums 1 tika restartēts
4. rinda: Pakalpojums 2 tika restartēts
```

Notikumu korelācijas ideja ir spēt sasaistīt problēmas notikumu no 1. rindas ar atrisinājumu no 3. rindas un problēmas notikumu no 2. rindas ar atrisinājumu no 4. rindas, un aizvērt šīs problēmas pa vienai:

```
1. rinda: Pakalpojums 1 apturēts
3. rinda: Pakalpojums 1 tika restartēts #problēma no 1. rindas aizvērta

2. rinda: Pakalpojums 2 apturēts
4. rinda: Pakalpojums 2 tika restartēts #problēma no 2. rindas aizvērta
```

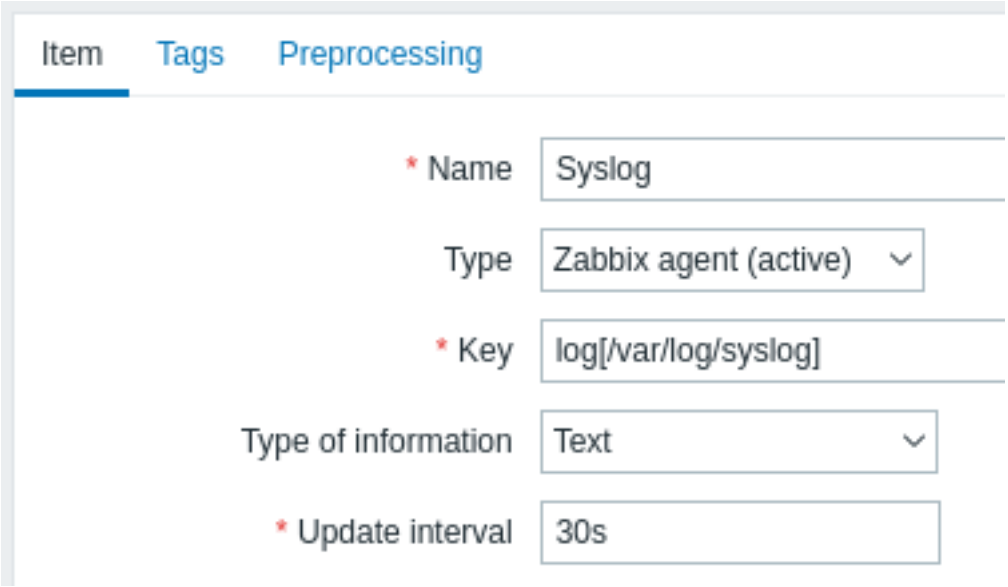
Lai to izdarītu, jums šie saistītie notikumi ir jāmarķē, piemēram, kā "Pakalpojums 1" un "Pakalpojums 2". To var izdarīt, loga rindai piemērojot regulāro izteiksmi, lai iegūtu taga vērtību. Pēc tam, kad notikumi tiek izveidoti, tiem attiecīgi tiek piešķirti tagi "Pakalpojums 1" un "Pakalpojums 2", un problēmu var sasaistīt ar atrisinājumu.

Konfigurācija

Vienums

Lai sāktu, iespējams, vēlēsities izveidot vienumu, kas uzrauga žurnālfailu, piemēram:

```
log[/var/log/syslog]
```



Kad vienums ir izveidots, uzgaidiet minūti, līdz konfigurācijas izmaiņas tiek pārņemtas, un pēc tam dodieties uz Latest data, lai pārliecinātos, ka vienums ir sācis vākt datus.

Trigeris

Kad vienums darbotos, jums ir jākonfigurē trigeris. Ir svarīgi izlemt, kuri ieraksti žurnālfailā ir pietiekami nozīmīgi, lai tiem pievērstu uzmanību. Piemēram, šāds trigeris izteiksmes veids meklēs virkni, piemēram, 'Stopping', lai signalizētu par iespējamām problēmām:

```
find(/My host/log[/var/log/syslog],,"regexp","Stopping")=1
```

Attention:
Lai pārliecinātos, ka katra rinda, kurā ir virkne "Stopping", tiek uzskatīta par problēmu, trigeris konfigurācijā *Problem event generation mode* iestatiet uz 'Multiple'.

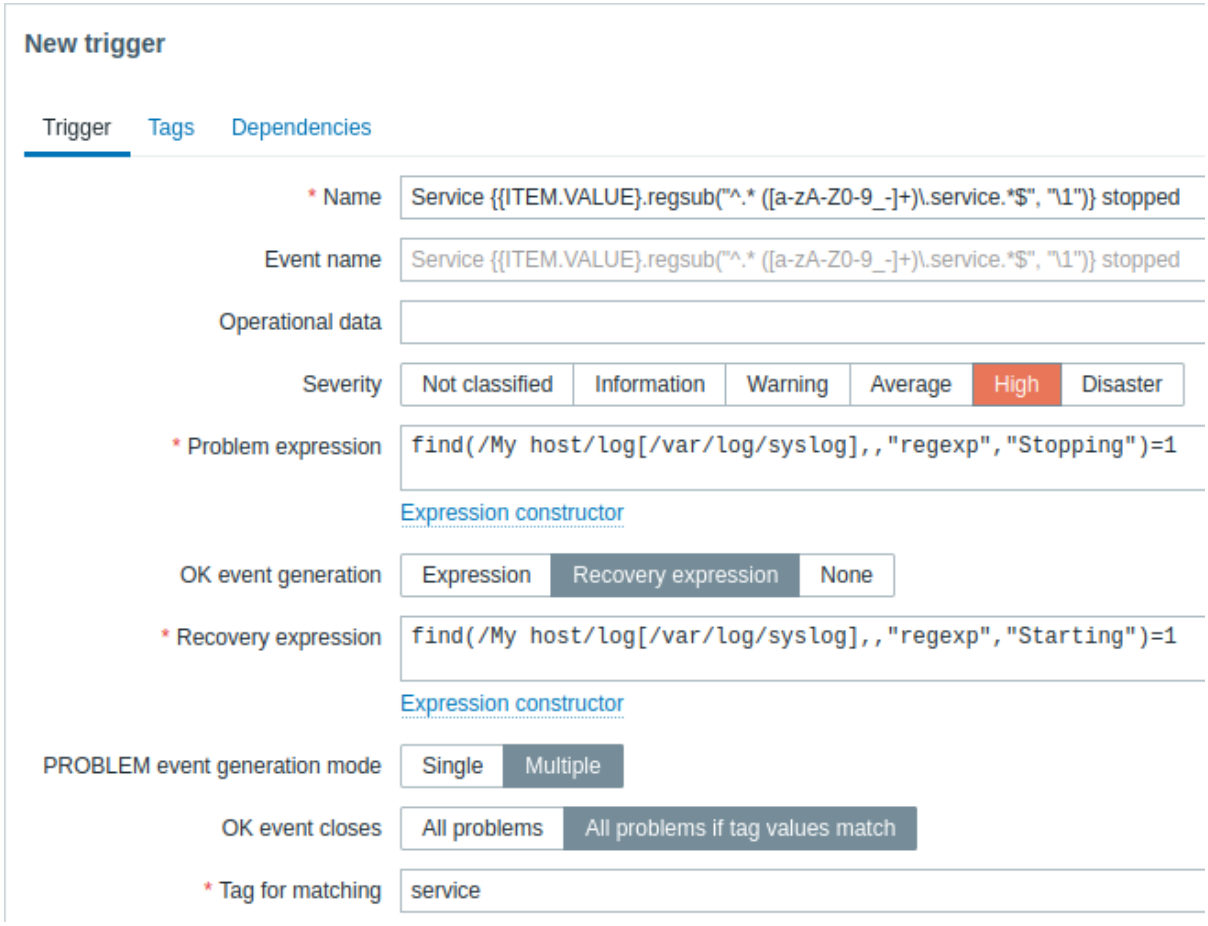
Pēc tam definējiet atkopšanas izteiksmi. Šāda atkopšanas izteiksme atrisinās visas problēmas, ja tiks atrasta žurnāla rinda, kurā ir virkne "Starting":

```
find(/My host/log[/var/log/syslog],,"regexp","Starting")=1
```

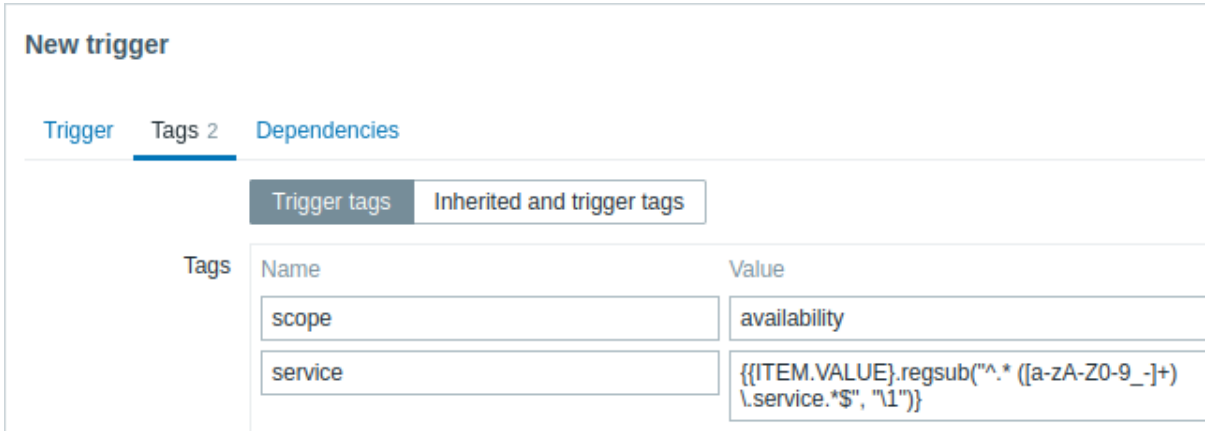
Tā kā mēs nevēlamies, lai tā notiktu, ir svarīgi kaut kā nodrošināt, ka attiecīgās pamatproblēmas tiek aizvērtas, nevis tikai visas problēmas. Šeit var palīdzēt tagošana.

Problēmas un atrisinājumus var saskaņot, norādot tagu trigeris konfigurācijā. Jāveic šādi iestatījumi:

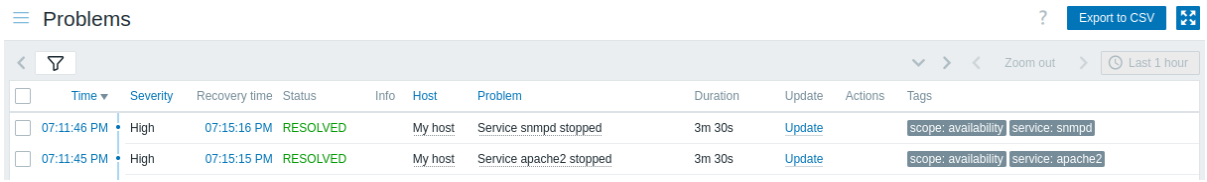
- *Problem event generation mode*: Multiple
- *OK event closes*: All problems if tag values match
- Ievadiet taga nosaukumu notikumu saskaņošanai



- konfigurējiet tagus, lai no žurnāla rindām iegūtu taga vērtības



Ja konfigurācija ir veiksmīga, sadaļā *Monitoring* > *Problems* varēsiet redzēt problēmu notikumus, kas ir atzīmēti pēc lietojumprogrammas un saskaņoti ar to atrisinājumu.



Warning:
Tā kā ir iespējama nepareiza konfigurācija, ja līdzīgi notikumu tagi var tikt izveidoti **nesaistītām** problēmām, lūdzu, pārskatiet tālāk aprakstītos gadījumus!

- Indeksētie makro vienmēr attiecas uz trigeris konfigurācijas lauku *Expression*, nevis uz *Recovery expression*. Piemēram, atkopšanas notikumā **{ITEM.VALUE1}** tiks aizstāts ar pirmā vienuma jaunāko vērtību **problēmas izteiksmē** atkopšanas brīdī. Ja atkopšanas izteiksme ir balstīta uz citu vienumu un problēmas izteiksmes vienuma vērtība līdz atkopšanas brīdim mainās, notikumiem būs atšķirīgi tagi, un tie netiks korelēt i.

- Ja divas lietojumprogrammas raksta kļūdu un atkopšanas ziņojumus vienā un tajā pašā žurnālfailā, lietotājs var nolemt izmantot divus *service* tagus vienā un tajā pašā trigeris ar atšķirīgām taga vērtībām, izmantojot atsevišķas regulārās izteiksmes taga vērtībās, lai no {ITEM.VALUE} makro iegūtu, piemēram, pakalpojuma A un pakalpojuma B nosaukumus (piemēram, ja ziņojumu formāti atšķiras). Tomēr tas var nedarboties, kā plānots, ja regulārajām izteiksmēm nav atbilstības. Regulārajām izteiksmēm bez atbilstības tiks piešķirtas tukšas taga vērtības, un pietiek ar vienu tukšu taga vērtību gan problēmas, gan OK notikumos, lai tos korelētu. Tādēļ atkopšanas ziņojums no pakalpojuma A var nejauši aizvērt kļūdas ziņojumu no pakalpojuma B.
- Faktiskie tagi un taga vērtības kļūst redzami tikai tad, kad trigeris nostrādā. Ja izmantotā regulārā izteiksme ir nederīga, tā klusi tiek aizstāta ar virkni *UNKNOWN*. Ja sākotnējais problēmas notikums ar *UNKNOWN* taga vērtību tiek palaists garām, var parādīties turpmāki OK notikumi ar to pašu *UNKNOWN* taga vērtību, kas var aizvērt problēmu notikumus, kurus tiem nevajadzēja aizvērt.
- Ja lietotājs kā taga vērtību izmanto {ITEM.VALUE} makro bez makro funkcijām, tiek piemērots 255 rakstzīmju ierobežojums. Ja žurnāla ziņojumi ir gari un pirmās 255 rakstzīmes nav specifiskas, tas var arī radīt līdzīgus notikumu tagus nesaistītām problēmām.

Globāla notikumu korelācija

Pārskats

Globālā notikumu korelācija ļauj aptvert visus Zabbix uzraudzītos rādītājus 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 loģikas, kurā problēmas un to novēršana ir balstīta uz vienu trigeri. Līdz šim vienu problēmas notikumu izveidoja viens trigeris, un problēmas novēršanai bijām atkarīgi no šī paša trigera. Mēs nevarējām novērst viena trigera izveidotu problēmu ar citu trigeri. Taču, izmantojot uz notikumu marķēšanu balstītu notikumu korelāciju, tas ir iespējams.

Piemēram, žurnāla trigeris var ziņot par lietojumprogrammas problēmām, savukārt aptaujas trigeris var ziņot, ka lietojumprogramma darbojas. Izmantojot notikumu marķēšanu, žurnāla trigerim var pievienot marķējumu *status:down*, bet aptaujas trigerim - marķējumu *status:up*. Pēc tam globālās korelācijas noteikumā šos trigerus var sasaistīt un šai korelācijai piešķirt atbilstošu darbību, piemēram, aizvērt vecos notikumus.

Citā gadījumā globālā korelācija var identificēt līdzīgus trigerus un piemērot tiem vienādu darbību. Kā būtu, ja mēs varētu saņemt tikai vienu ziņojumu par katru tīkla porta problēmu? Nebūtu nepieciešams ziņot par visām. Ar globālo notikumu korelāciju arī tas ir iespējams.

Globālā notikumu korelācija tiek konfigurēta **korelācijas noteikumos**. Korelācijas noteikums nosaka, kā jaunie problēmu notikumi tiek sasaistīti ar esošajiem problēmu notikumiem un kā rīkoties atbilstības gadījumā (aizvērt jauno notikumu, aizvērt atbilstošos vecos notikumus, ģenerējot atbilstošus OK notikumus). Ja problēma tiek aizvērta, izmantojot globālo korelāciju, tā tiek parādīta *Monitoring* > *Problems* sadaļas kolonnā *Info*.

Globālās korelācijas noteikumus konfigurēt drīkst tikai lietotāji ar Super Admin līmeņa tiesībām.

Attention:
Notikumu korelācija ir jākonfigurē ļoti rūpīgi, jo tā var negatīvi ietekmēt notikumu apstrādes veiktspēju vai, ja tā ir konfigurēta nepareizi, aizvērt vairāk notikumu, nekā paredzēts (sliktākajā gadījumā var tikt aizvērti pat visi problēmu notikumi).

Lai globālās korelācijas noteikumus konfigurētu **droši**, ievērojiet šos svarīgos ieteikumus:

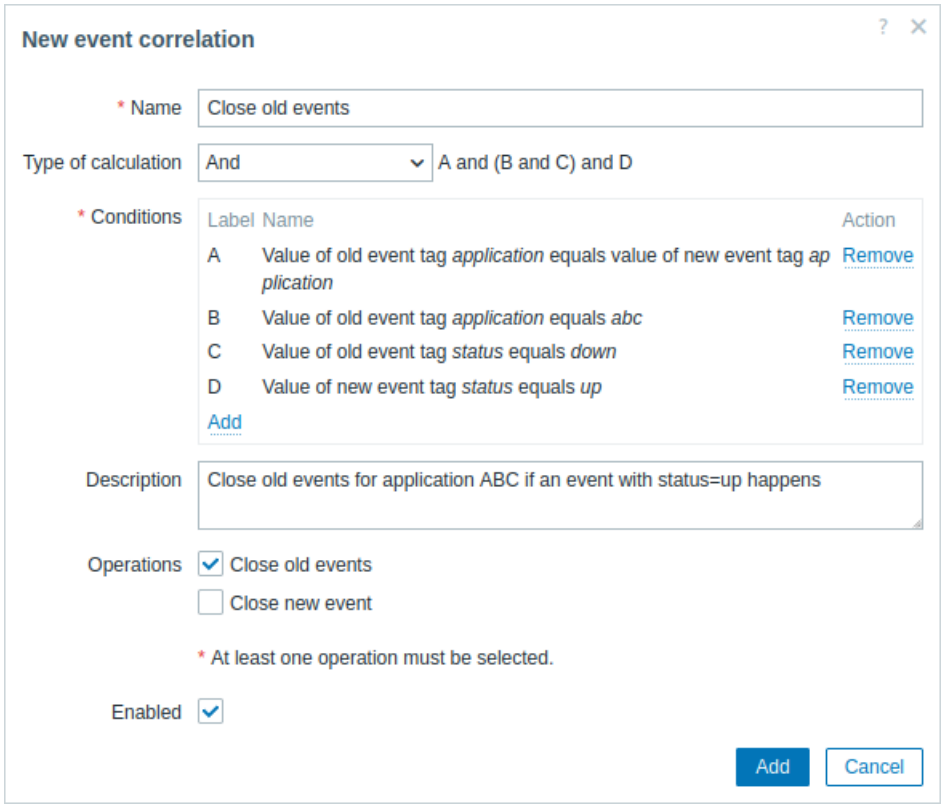
- Samaziniet korelācijas tvērumu. Vienmēr iestatiet unikālu marķējumu jaunajam notikumam, kas tiek sasaistīts ar vecajiem notikumiem, un izmantojiet korelācijas nosacījumu *New event tag name*.
- Izmantojot **Close old events**, pievienojiet skaidru vecā notikuma nosacījumu. Izvēloties **Close old events**, vienmēr pievienojiet vismaz vienu *Old event* nosacījumu (piemēram, *Old event tag name*, *Old event tag value* vai *Event tag pair*); pretējā gadījumā noteikums var atbilst nesaistītiem esošajiem problēmu notikumiem un tos aizvērt (sliktākajā gadījumā - visas problēmas). Izmantojiet *Event tag pair*, lai atbilstību noteiktu pēc izpildlaika vērtībām (host:port, sesijas ID u. c.), un, ja iespējams, papildus sašauriniet atbilstību pēc hosts vai hostu grupas.
- Neizmantojiet izplatītus marķējumu nosaukumus, kurus varētu izmantot dažādas korelācijas konfigurācijas.
- Ierobežojiet korelācijas noteikumu skaitu līdz tiem, kas jums patiešām ir nepieciešami.

Skatiet arī: zināmās problēmas.

Konfigurācija

Lai globāli konfigurētu notikumu korelācijas noteikumus:

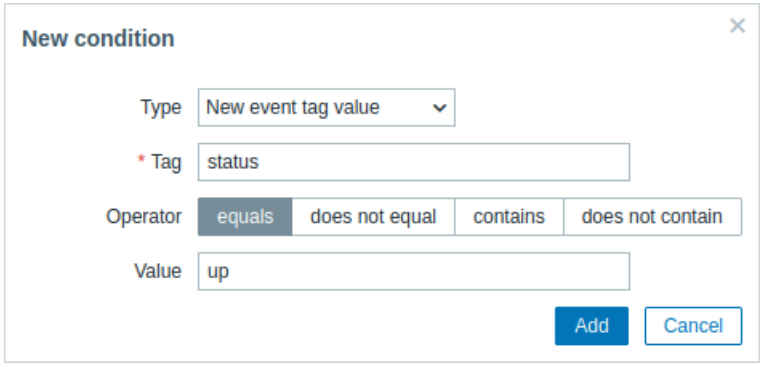
1. Dodieties uz *Data collection* > *Event correlation*.
2. Noklikšķiniet uz *Create event correlation* labajā pusē (vai uz korelācijas nosaukuma, lai rediģētu esošu noteikumu).
3. Formā ievadiet korelācijas noteikuma parametrus.



Visi obligātie ievades lauki ir atzīmēti ar sarkanu zvaigznīti.

Parameter	Description
Name	Unikāls korelācijas noteikuma nosaukums.
Type of calculation	Pieejamas šādas nosacījumu aprēķināšanas opcijas: **And** - jāizpildās visiem nosacījumiem **Or** - pietiek, ja izpildās viens nosacījums **And/Or** - AND ar dažādiem nosacījumu tipiem un OR ar vienādiem nosacījumu tipiem **Custom expression** - lietotāja definēta aprēķina formula darbību nosacījumu izvērtēšanai. Tai jāietver visi nosacījumi (attēloti ar lielajiem burtiem A, B, C, ...) un var ietvert atstarpes, tabulācijas, iekavas ( ), **and** (reģistrjutīgs), **or** (reģistrjutīgs), **not** (reģistrjutīgs).
Conditions	Nosacījumu saraksts. Skatiet tālāk, lai uzzinātu vairāk par nosacījuma konfigurēšanu.
Description	Korelācijas noteikuma apraksts.
Operations	Atzīmējiet tās darbības izvēles rūtiņu, kas jāveic, kad notikums tiek korelēts. Pieejamas šādas darbības: **Close old events** - aizvērt vecos notikumus, kad notiek jauns notikums. Vienmēr pievienojiet nosacījumu, kas balstīts uz veco notikumu, izmantojot darbību *Close old events*, pretējā gadījumā var tikt aizvērti visi esošie problēmu notikumi. **Close new event** - aizvērt jauno notikumu, kad tas notiek. **Brīdinājums!** Nepalieciet vecā/jaunā notikuma nosacījumus tukšus, izmantojot *Close old events*/*Close new event*. Ja izvēlaties darbību *Close old events* bez nosacījuma, 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*, saskaņos visus vecos notikumus, kas atbilst (trūkstošajiem) vecā notikuma kritērijiem - faktiski aizverot vecās problēmas.
Enabled	Ja atzīmējat šo izvēles rūtiņu, korelācijas noteikums būs iespējots.

Lai konfigurētu jauna nosacījuma detaļas, Conditions blokā noklikšķiniet uz Add. Tiks atvērts uznirstošais logs, kurā varēsiet rediģēt nosacījuma detaļas.



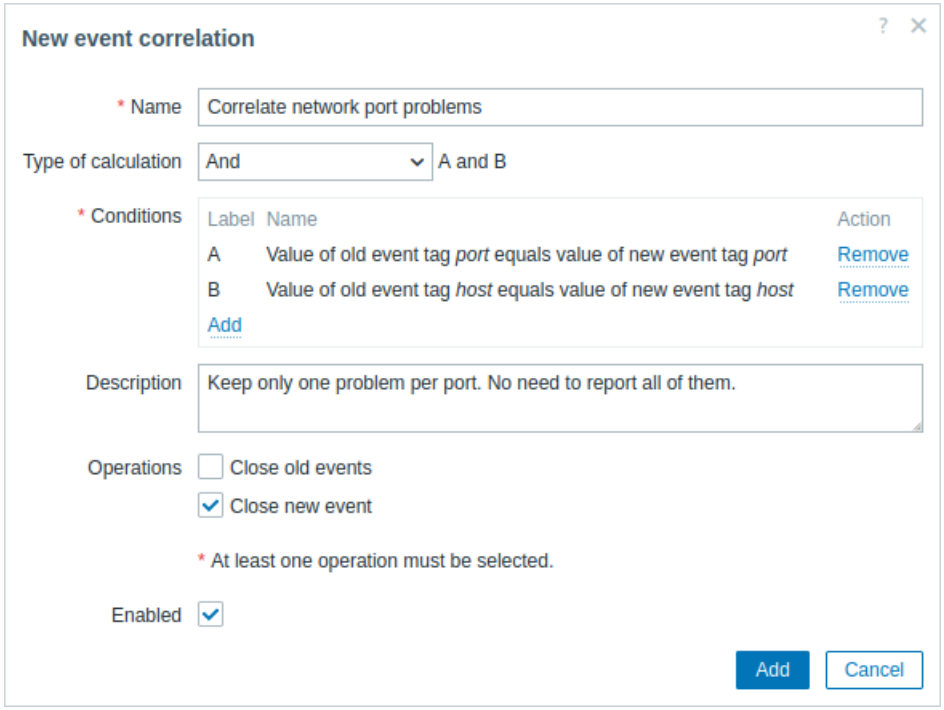
Parameter	Description
New condition	Atlasiet nosacījumu notikumu korelācijai. *Piezīme*, ka, ja nav norādīts vecā notikuma nosacījums, var tikt saskaņoti un aizvērti visi vecie notikumi. Līdzīgi, ja nav norādīts jaunā notikuma nosacījums, var tikt saskaņoti un aizvērti visi jaunie notikumi. Pieejami šādi nosacījumi: **Old event tag name** - norādiet vecā notikuma taga nosaukumu saskaņošanai. **New event tag name** - norādiet jaunā notikuma taga nosaukumu saskaņošanai. **New event host group** - norādiet jaunā notikuma hostu grupu saskaņošanai. **Event tag pair** - norādiet jaunā notikuma taga nosaukumu un vecā notikuma taga nosaukumu saskaņošanai. Šajā gadījumā atbilstība būs tad, ja abu notikumu tagu **vērtības** sakritīs. Tagu *nosaukumiem* nav obligāti jāsakrīt. Šī opcija ir noderīga izpildlaika vērtību saskaņošanai, kas konfigurēšanas brīdī var nebūt zināmas (skatiet arī Example). **Old event tag value** - norādiet vecā notikuma taga nosaukumu un vērtību saskaņošanai, izmantojot šādus operatorus: *equals* - ir vecā notikuma taga vērtība *does not equal* - nav vecā notikuma taga vērtība *contains* - satur virkni vecā notikuma taga vērtībā *does not contain* - nesatur virkni vecā notikuma taga vērtībā **New event tag value** - norādiet jaunā notikuma taga nosaukumu un vērtību saskaņošanai, izmantojot šādus operatorus: *equals* - ir jaunā notikuma taga vērtība *does not equal* - nav jaunā notikuma taga vērtība *contains* - satur virkni jaunā notikuma taga vērtībā *does not contain* - nesatur virkni jaunā notikuma taga vērtībā

Warning:
Tā kā iespējama nepareiza konfigurācija, ja līdzīgi notikumu tagi var tikt izveidoti **nesaistītām** problēmām, lūdzu, pārskatiet tālāk aprakstītos gadījumus!

- Faktiskie tagi un tagu vērtības kļūst redzami tikai tad, kad tiek aktivizēts trigeris. Ja izmantotā regulārā izteiksme ir nederīga, tā klusi tiek aizstāta ar virkni *UNKNOWN*. Ja sākotnējais problēmas notikums ar *UNKNOWN* taga vērtību tiek palaists garām, var parādīties turpmāki OK notikumi ar to pašu *UNKNOWN* taga vērtību, kas var aizvērt problēmu notikumus, kurus tiem nevajadzētu aizvērt.
- Ja lietotājs kā taga vērtību izmanto {ITEM.VALUE} makro bez makro funkcijām, tiek piemērots 255 rakstzīmju ierobežojums. Ja žurnāla ziņojumi ir gari un pirmās 255 rakstzīmes nav specifiskas, tas var arī radīt līdzīgus notikumu tagus nesaistītām problēmām.

Piemērs

Apturiet atkārtotus problēmu notikumus no tā paša tīkla porta.



Šis globālais korelācijas noteikums sasaistīs problēmas, ja trigerī pastāv *host* un *port* tagu vērtības un tās ir vienādas sākotnējā notikumā un jaunajā notikumā.

Šī darbība aizvērs jaunus problēmu notikumus tajā pašā tīkla portā, atstājot atvērtu tikai sākotnējo problēmu.

Atklāšana

Pārskats Atklāšana nodrošina automātiskus veidus, kā pievienot hostus un konfigurēt uzraudzību bez manuālas iestatīšanas. Zabbix atbalsta vairākas atklāšanas metodes, lai pielāgotos dažādiem scenārijiem.

Atklāšanas metodes

- Tīkla atklāšana - automātiski noteikt jūsu tīklā pieejamos hostus un pakalpojumus.
- Aktīvā aģenta automātiskā reģistrācija - ļaut aktīvajiem aģentiem automātiski reģistrēties serverī.
- Zema līmeņa atklāšana - automātiski izveidot vienumus, trigerus, grafikus un hostus atklātajām entītijām, piemēram, failu sistēmām vai tīkla saskarnēm, izmantojot prototipus un ligzdotus atklāšanas noteikumus.

Tīkla atklāšana

Pārskats

Zabbix piedāvā automātisku tīkla atklāšanas funkcionalitāti, kas ir efektīva un ļoti elastīga.

Pareizi konfigurējot tīkla atklāšanu, jūs varat:

- Paātrināt Zabbix ieviešanu.
- Vienkāršot administrēšanu.
- Lietot 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 aģents (tiek atbalstīts tikai nešifrēts režīms)
- Informācija, kas saņemta no SNMP aģents

Tā NENODROŠINA tīkla topoloģijas atklāšanu.

Tīkla atklāšana pamatā sastāv no divām fāzēm: atklāšanas un darbībām.

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 IP diapazonam.

Atklāšanas noteikumus apstrādā atklāšanas pārvaldnieks. Atklāšanas pārvaldnieks izveido vienu darbu katram noteikumam ar uzdevumu sarakstu (tīkla pārbaudēm). Tīkla pārbaudes paralēli veic pieejamie atklāšanas darbinieki (to skaits ir konfigurējams lietotāja saskarnē katram noteikumam). Izņēmums ir SNMPv3 pārbaudes, kuras apstrādā viens darbinieks. Secīgi tiek ieplānotas tikai pārbaudes ar vienu un to pašu IP un portu, jo dažas ierīces nepieņem paralēlus savienojumus uz viena porta.

Tīkla pārbaužu rindas izmērs ir ierobežots līdz 2000000 vai aptuveni 4 GB atmiņas. Ja rinda kļūst pilna, atklāšanas noteikums tiks izlaists, un žurnālā tiks ierakstīts 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 pārbaude neatrod pakalpojumu (vai neizdodas), pārējās pārbaudes joprojām tiks apstrādātas.

Note:
Ja atklāšanas noteikums izpildes laikā tiek mainīts, pašreizējā atklāšanas izpilde tiks pārtraukta.

Katras pakalpojuma un hosts (IP) pārbaudes, 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 tiek atklāts pirmo reizi.
Pakalpojums ir pieejams	Pakalpojums ir 'up', pēc tam, kad tas jau bija 'up'.
Pakalpojums zaudēts	Pakalpojums ir 'down' pēc tam, kad tas bija 'up'.
Pakalpojums nav pieejams	Pakalpojums ir 'down', pēc tam, kad tas jau bija 'down'.
Hosts atklāts	Vismaz viens hosts pakalpojums ir 'up' pēc tam, kad visi šī hosta pakalpojumi bija 'down', vai arī tiek atklāts pakalpojums, kas pieder nereģistrētam hostam.
Hosts ir pieejams	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 nav pieejams	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.

Hostu izveide

Hosts tiek pievienots, ja ir atlasīta darbība *Pievienot hostu*. Hosts tiek pievienots arī tad, ja darbība *Pievienot hostu* nav norādīta, ja atlasāt darbības, kuru rezultātā tiek veiktas darbības ar hostu. Šādas darbības ir:

- Iespējot hostu.
- Atspējot hostu.
- Pievienot hostu hostu grupai.
- Saistīt veidni ar hostu.

Izveidotie hosti tiek pievienoti grupai *Discovered hosts* (pēc noklusējuma, konfigurējams sadaļā *Administration* > *General* > *Other*). Ja vēlaties, lai hosti tiktu pievienoti citai grupai, pievienojiet darbību *Remove from host groups* (norādot "Discovered hosts") un arī darbību *Add to host groups* (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 interfeisa tipu tiek izmantota kā kritērijs hosta atrašanai sistēmā. Ja hostam ar to pašu IP adresi, interfeisa tipu un atklāšanas avotu jau ir, šis hosts tiks izmantots darbību veikšanai. Ja atklāšanas avots atšķiras, atklātais objekts tiek uzskatīts par citu hostu, un var tikt izveidots jauns hosts.

Ja atklātā hosta IP adrese tiek mainīta vai interfeiss tiek dzēsts, 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.

Interfeisa izveide, pievienojot hostus

Kad hosti tiek pievienoti tīkla atklāšanas rezultātā, tiem tiek izveidoti interfeisi saskaņā ar šādiem noteikumiem:

- Tiek noteikti pakalpojumi - piemēram, ja SNMP pārbaude bija veiksmīga, tiks izveidots SNMP interfeiss.
- Ja hosts atbildēja gan uz Zabbix aģents, gan SNMP pieprasījumiem, tiks izveidoti abu tipu interfeisi.
- Ja unikalitātes kritēriji ir Zabbix aģents vai SNMP atgrieztie dati, pirmais atrastais hosta interfeiss tiks izveidots kā noklusējuma interfeiss. Citas IP adreses tiks ignorētas, ja vien noklusējuma interfeiss nedarbosies. Ja noklusējuma interfeiss nedarbojas, kā aizstājējs noklusējuma interfeisam tiks izmantota mazākā no pārējām IP adresēm. Darbības nosacījumi (piemēram, Host IP) neietekmē interfeisu pievienošanu. *Piezīme*, ka tas darbosies, ja visi interfeisi tiek atklāti ar vienu un to pašu atklāšanas noteikumu. Ja cits atklāšanas noteikums atklāj cita tā paša hosta interfeisu, tiks pievienots papildu hosts.
- Ja hosts atbildēja tikai uz aģents pārbaudēm, tas tiks izveidots tikai ar aģents interfeisu. Ja vēlāk tas sāks atbildēt uz SNMP, tiks pievienoti papildu SNMP interfeisi.
- Ja sākotnēji tika izveidoti 3 atsevišķi hosti, kas tika atklāti, izmantojot "IP" unikalitātes kritērijus, un pēc tam atklāšanas noteikums tiek modificēts tā, ka hostiem A, B un C ir identisks unikalitātes kritēriju rezultāts, B un C tiek izveidoti kā papildu interfeisi A, pirmajam hostam. Atsevišķie hosti B un C paliek. Sadaļā *Monitoring* > *Discovery* pievienotie interfeisi tiks parādīti kolonnā "Discovered device" ar melnu fontu un ar atkāpi, bet kolonna "Monitored host" rādīs tikai A, pirmo izveidoto hostu. "Uptime/Downtime" netiek mērīts IP adresēm, kas tiek uzskatītas par papildu interfeisiem.

Starpniekservera iestatījuma maiņa

Hosts, ko atklāj dažādi starpniekserveri, ne vienmēr tiek uzskatīti par atšķirīgiem hosts. Atklāšana un unikalitātes pārbaudes ir atkarīgas no starpniekserveru grupas struktūras: kad starpniekserveris izpilda atklāšanas noteikumu un izveido hostu, šis hosts tiek pievienots starpniekservera vecākgrupai, nevis piešķirts pašam starpniekserverim. Kad Zabbix atklāšanas laikā izvērtē IP

unikalitāti, tas pārbauda hostus, ko uzrauga vecākgrupa. Hosti, ko šajā grupā uzrauga atsevišķi starpniekserveri (tostarp starpniekserveris, kas veica atklāšanu), unikalitātes pārbaudē tiek ignorēti, kas var izraisīt dublētus hostus, ja vairāki starpniekserveri uzrauga pārklājošos apakštīklus.

Lai gan šāda darbība ļauj atklāšanai darboties pāri pārklājošiem IP diapazoniem, ko izmanto dažādi apakštīkli, jau uzraudzīta apakštīkla starpniekservera maiņa ir sarežģītāka, jo starpniekservera izmaiņas ir konsekventi jāpiemēro gan atklātajiem hostiem, gan vecākgrupas dalībai, lai izvairītos no dublikātiem.

Piemēram, darbības, lai aizstātu starpniekserveri atklāšanas noteikumā:

1. Atspējojiet atklāšanas noteikumu.
2. Sinhronizējiet starpniekservera konfigurāciju.
3. Aizstājiet starpniekserveri atklāšanas noteikumā.
4. Aizstājiet starpniekserveri visiem hostiem, kas atklāti ar šo noteikumu (pārliecinieties, ka hosti vecākgrupā un jebkuri hosti, ko šajā grupā uzrauga atsevišķi starpniekserveri, ir atjaunināti, lai izvairītos no dublikātiem).
5. Iespējojiet atklāšanas noteikumu.

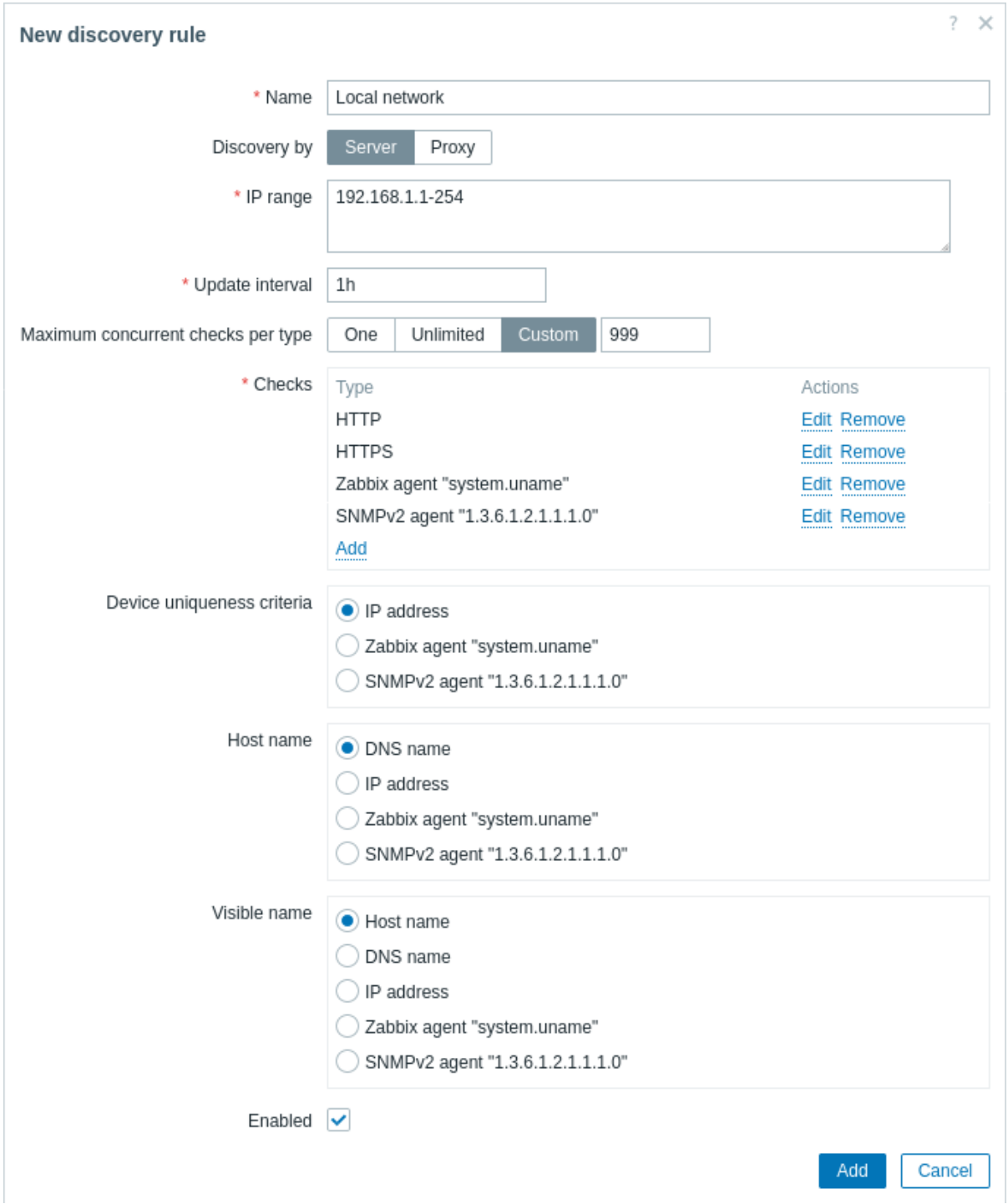
Tīkla atklāšanas noteikuma konfigurēšana

Pārskats

Lai konfigurētu tīkla atklāšanas noteikumu, ko Zabbix izmanto hosts un pakalpojumu atklāšanai:

1. Dodieties uz *Datu vākšana > Atklāšana*.
2. Noklikšķiniet uz *Izveidot atklāšanas noteikumu* (vai uz noteikuma nosaukuma, lai rediģētu esošu).
3. Rediģējiet atklāšanas noteikuma atribūtus.

Noteikumu atribūti



Visi obligātie ievades lauki ir atzīmēti ar sarkanu zvaigznīti.

Parameter	Description
Name	Unikāls noteikuma nosaukums. Piemēram, ”Lokālais tīkls”.
Discovery by	Atklāšana tiek veikta ar: **Serveris** - ar Zabbix serveris **Starpniekserveris** - ar Zabbix starpniekserveris (atlasīts starpniekservera nosaukuma laukā)
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. Diapazons ir ierobežots ar kopējo aptverto adrešu skaitu (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.

Parameter	Description
Update interval	Šis parametrs nosaka, cik bieži Zabbix izpildīs noteikumu. Intervāls tiek mērīts pēc iepriekšējā atklāšanas gadījuma izpildes beigām, tāpēc pārklāšanās nenotiek. Laika sufiksi tiek atbalstīti, piemēram, 30s, 1m, 2h, 1d. Lietotāja makro tiek atbalstīti. *Piezīme*, ka, ja tiek izmantots lietotāja makro un tā vērtība tiek mainīta (piem., 1w → 1h), nākamā pārbaude tiks izpildīta atbilstoši iepriekšējai vērtībai (ar dotajiem piemēra datiem - tālā nākotnē).
Maximum concurrent checks per type	Iestatiet maksimālo atklāšanas pavedienu (darbinieku) skaitu uz vienu pakalpojuma pārbaudi paralēlai atklāšanas pārbaužu apstrādei: **One** - viens pavediens **Unlimited** - neierobežots pavedienu skaits (bet ne vairāk kā StartDiscoverers parametrā) **Custom** - iestatiet pielāgotu pavedienu skaitu (0-999). SNMPv1 un SNMPv2 asinhronās pakalpojuma pārbaudes tiek apstrādātas paralēli pieejamajiem darbiniekiem. Visi atklāšanas noteikumi ar SNMPv3 asinhronajām pakalpojuma pārbaudēm tiek apstrādāti ar vienu darbinieku libsnmp ieviešanas īpatnību dēļ; tāpēc darbinieku skaita palielināšana nepalielinās atklāšanas ātrumu.
Checks	Zabbix izmantos šo pārbaužu sarakstu atklāšanai. Noklikšķiniet uz Add, lai konfigurētu jaunu pārbaudi uznirstošajā logā. Atbalstītās pārbaudes: SSH, LDAP, SMTP, FTP, HTTP, HTTPS, POP, NNTP, IMAP, TCP, Telnet, Zabbix aģents, SNMPv1 aģents, SNMPv2 aģents, SNMPv3 aģents, ICMP ping. Ar protokolu balstīta atklāšana izmanto **net.tcp.service[]** funkcionalitāti, lai pārbaudītu katru hostu, izņemot SNMP, kur tiek vaicāts SNMP OID. Zabbix aģents tiek pārbaudīts, vaicājot vienumu nešifrētā režīmā. Plašāku informāciju skatiet aģenta vienumi. Parametrs 'Ports' var būt viens no šiem: Viens ports: 22 Portu diapazons: 22-45 Saraksts: 22-45,55,60-70 Kopš Zabbix 7.0 visas pakalpojuma pārbaudes tiek veiktas asinhroni, izņemot LDAP pārbaudes. Kopš 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 adrese** - neapstrādāt vairākas ierīces ar vienu IP adresi. Ja ierīce ar to pašu IP jau pastāv, tā tiks uzskatīta par jau atklātu, un jauns hosts netiks pievienots. **<discovery check>** - vai nu Zabbix aģents, vai SNMP aģenta pārbaude. Ņemiet vērā, ka atklāšanā izmantotie unikalitātes kritēriji nav tie paši, kas hosta identificēšana sistēmā, veicot darbības. Unikalitātes kritēriji atklāšanas laikā nosaka, vai divas vai vairāk atklātas ierīces ir vienas un tās pašas (vai atšķirīgas), savukārt Zabbix hosta identificēšanai kritērijs ir tikai IP adrese (skatiet Host creation).
Host name	Iestatiet izveidotā hosta tehnisko hosta nosaukumu, izmantojot: **DNS name** - DNS nosaukums (noklusējums) **IP address** - IP adrese **<discovery check>** - saņemtā atklāšanas pārbaudes virknes vērtība (piem., Zabbix aģents, SNMP aģenta pārbaude) Skatiet arī: Host naming.
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 atklāšanas pārbaudes virknes vērtība (piem., Zabbix aģents, SNMP aģenta pārbaude) Skatiet arī: Host naming.
Enabled	Ja izvēles rūtiņa ir atzīmēta, noteikums ir aktīvs un to izpildīs Zabbix serveris. Ja nav atzīmēta, noteikums nav aktīvs. Tas netiks izpildīts.

Faila deskriptoru limita pārsniegšana

Ja vienlaicīgo pārbaužu skaits ir liels, ir iespējams izsmelt faila deskriptoru limitu atklāšanas pārvaldniekam.

Faila deskriptoru skaits, kas nepieciešams noteikšanai, ir vienāds ar atklāšanas darbinieku skaitu * 1000. Pēc noklusējuma ir 5 atklāšanas darbinieki, savukārt sistēmas mīkstais limits ir aptuveni 1024.

Ja šis limits tiek sasniegts, Zabbix samazinās noklusējuma vienlaicīgo pārbaužu skaitu katram tipam katram darbiniekam un ierakstīs brīdinājumu žurnālfailā. Taču, 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

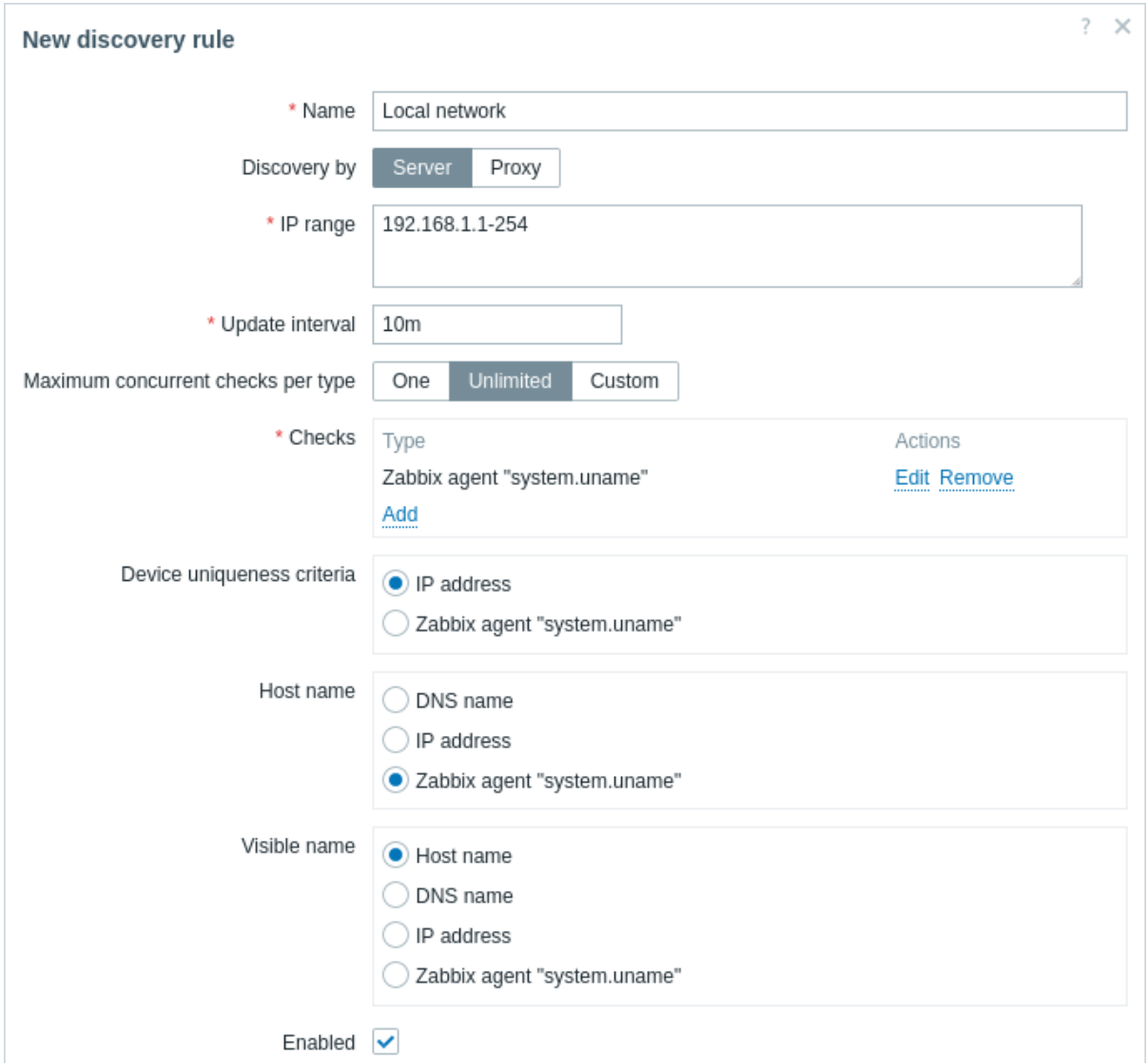
Šajā piemērā mēs vēlamies iestatīt tīkla atklāšanu lokālajam tīklam ar IP diapazonu 192.168.1.1-192.168.1.254.

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.



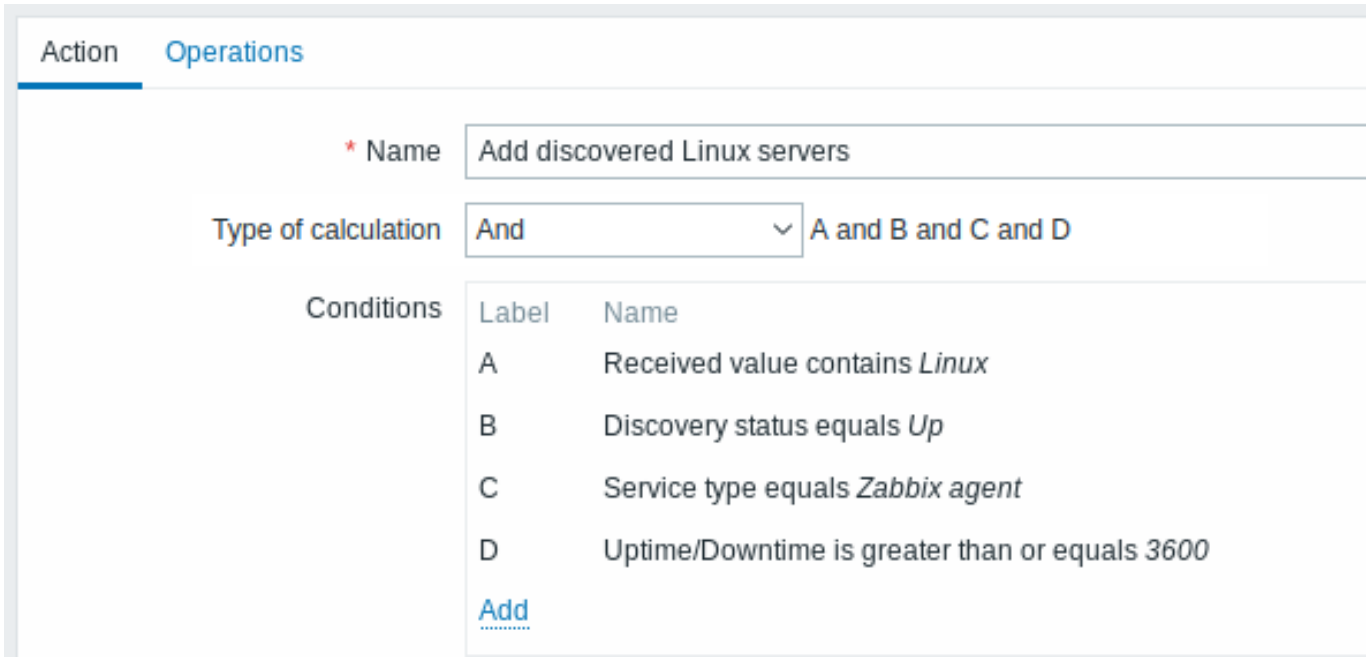
Zabbix mēģinās atklāt hostus IP diapazonā 192.168.1.1-192.168.1.254, pieslēdzoties Zabbix aģentiem un iegūstot vērtību no **system.uname** atslēgas. No aģenta saņemto vērtību var izmantot hostu nosaukšanai, kā arī dažādu darbību piemērošanai dažādām operētājsistēmām. Piemēram, saistīt Windows serverus 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.

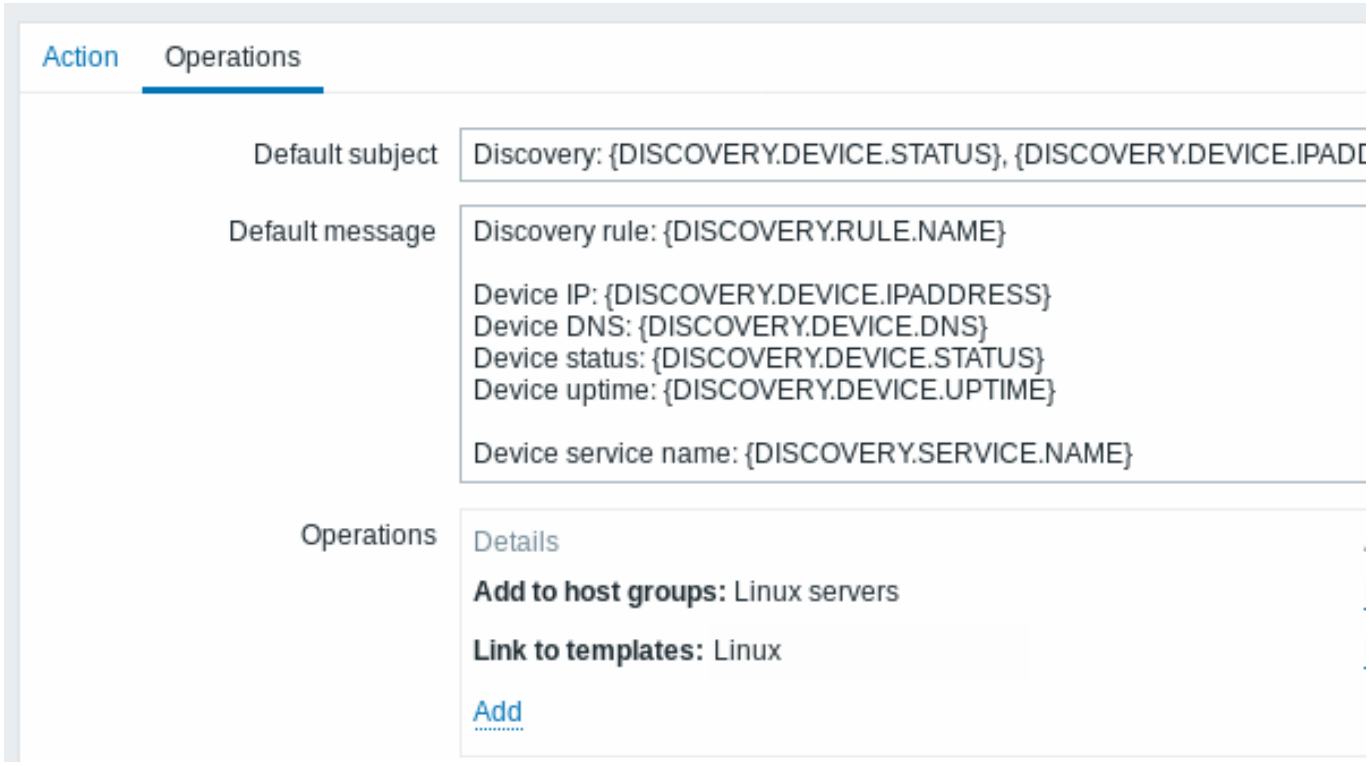
2. solis

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

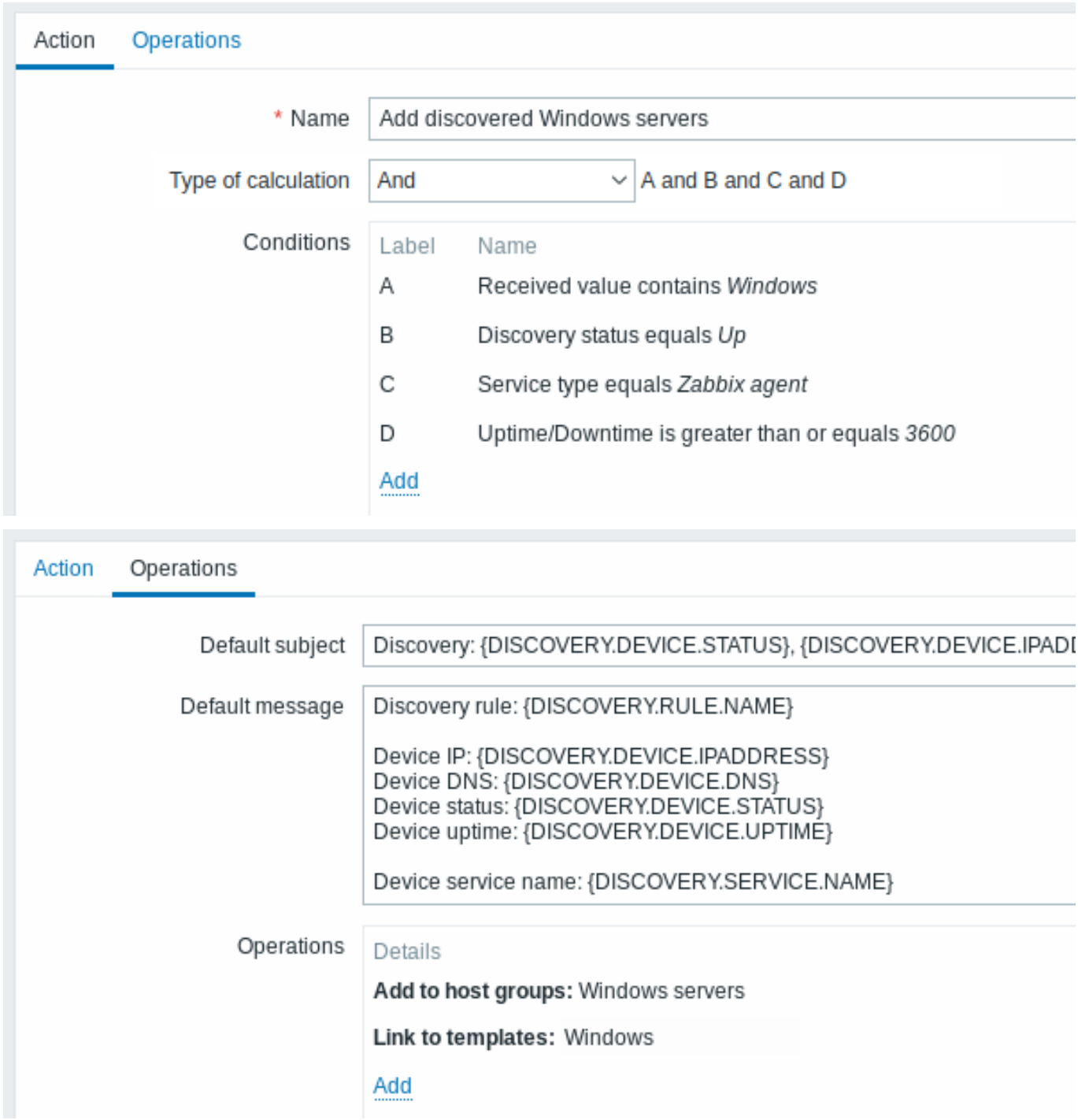


Darbība izpildīs šādas operācijas:

- 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”.

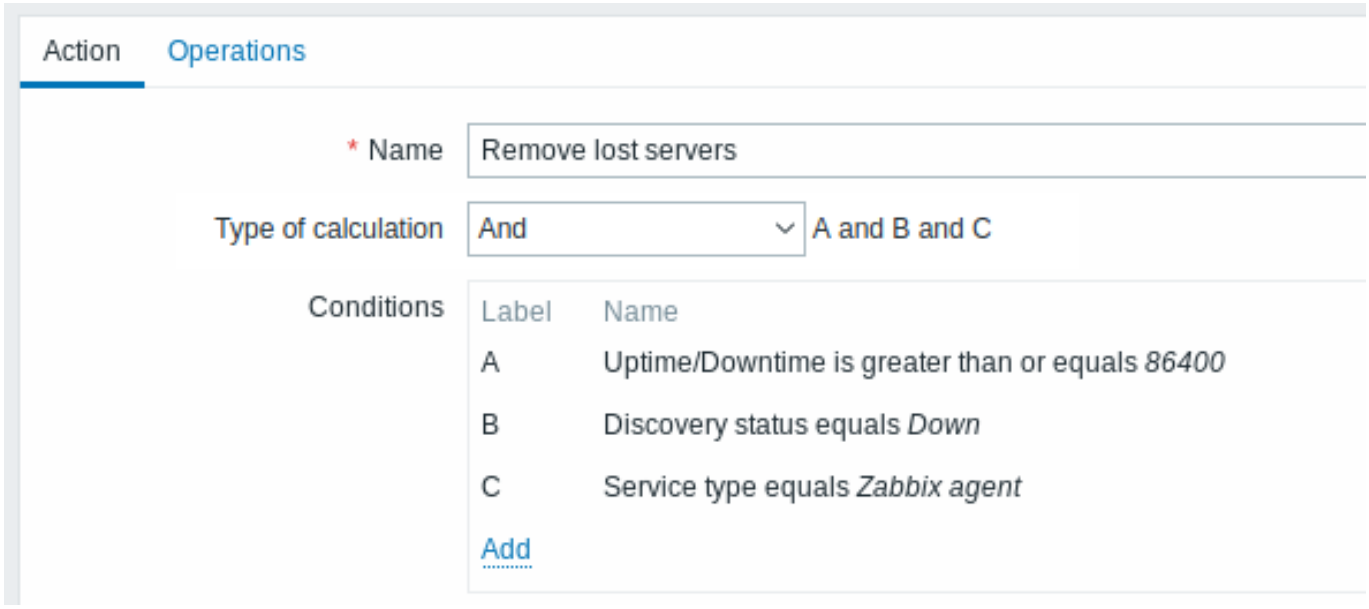
3. solis

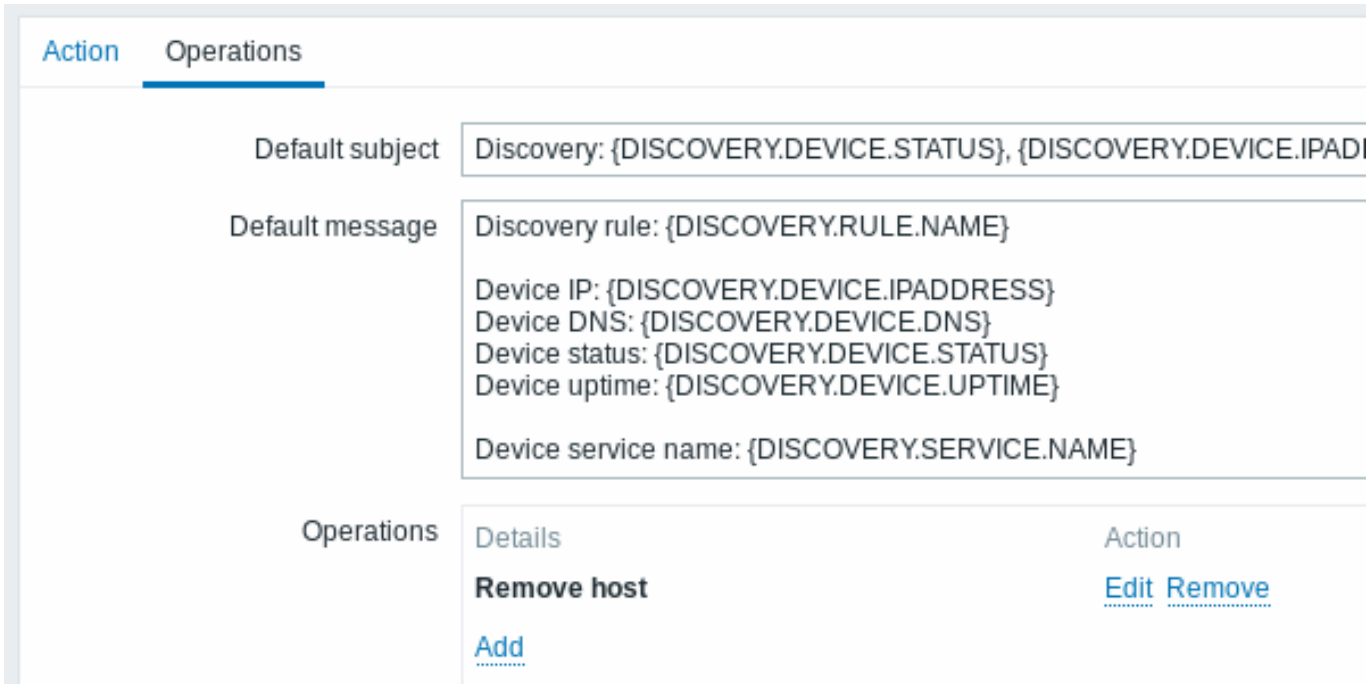
Atklāšanas darbības definēšana, lai pievienotu atklātos Windows serverus attiecīgajai grupai/veidnei.



4. solis

Atklāšanas darbības definēšana zaudētu serveru noņemšanai.





Serveris tiks noņemts, ja pakalpojums ”Zabbix aģents” būs ’down’ ilgāk par 24 stundām (86400 sekundēm).

Aktīva aģenta automātiskā reģistrācija

Pārskats

Ir iespējams atļaut aktīva Zabbix aģenta automātisko reģistrāciju, pēc kuras serveris var sākt tos uzraudzīt. Tādējādi jaunus hosts var pievienot uzraudzībai, tos manuāli nekonfigurējot serverī.

Automātiskā reģistrācija var notikt, kad iepriekš nezināms aktīvs aģents pieprasa pārbaudes.

Šī funkcija var būt ļoti noderīga jaunu mākoņa mezglu automātiskai uzraudzībai. Tiklīdz Zabbix Cloud parādās jauns mezgls, tiks automātiski sākta hosta veiktspējas un pieejamības datu vākšana.

Aktīva aģenta automātiskā reģistrācija atbalsta arī pievienotu hosts uzraudzību ar pasīvajām pārbaudēm. Kad aktīvais aģents pieprasa pārbaudes, un konfigurācijas failā ir definēti `ListenIP` vai `ListenPort` konfigurācijas parametri, tie tiek nosūtīti serverim. Ja ir norādītas vairākas IP adreses, serverim tiek nosūtīta pirmā.

Serveris, pievienojot jauno automātiski reģistrēto hostu, izmanto saņemto IP adresi un portu, lai konfigurētu aģentu. Ja IP adreses vērtība netiek saņemta, tiek izmantota tā, kas tika lietota ienākošajam savienojumam. Ja porta vērtība netiek saņemta, tiek izmantots 10050.

Ir iespējams norādīt, ka hostam jābūt automātiski reģistrētam ar DNS nosaukumu kā noklusējuma aģenta saskarni.

Automātiskā reģistrācija tiek atkārtota:

- ja mainās hosta metadatu informācija:
  - HostMetadata izmaiņu un aģenta restartēšanas dēļ
  - HostMetadataItem atgrieztās vērtības izmaiņu dēļ
- manuāli izveidotiem hosts, kuriem trūkst metadatu
- ja hostam manuāli tiek mainīts uzraudzības starpniekserveris uz citu Zabbix starpniekserveri
- ja par to pašu hostu automātiskā reģistrācija tiek saņemta no jauna Zabbix starpniekservera

Aktīva aģenta automātiskās reģistrācijas sirdspukstu intervāls Zabbix serverim un Zabbix starpniekserverim ir 120 sekundes. Tāpēc, ja atklātais hosts tiek dzēsts, automātiskā reģistrācija tiks atkārtota 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 hosts nosaukšanai izmantos aģenta atrašanās vietas sistēmas resursdatora nosaukumu. Sistēmas resursdatora nosaukumu Linux sistēmā var iegūt, izpildot komandu `hostname`.

Ja *Hostname* Zabbix aģenta konfigurācijā ir definēts kā ar komatiem atdalīts hostu saraksts, katram norādītajam resursdatora nosaukumam tiks izveidots hosts.

Pēc izmaiņu veikšanas konfigurācijas failā restartējiet aģentu.

Darbība aktīvo aģentu automātiskai reģistrācijai

Kad serveris saņem automātiskās reģistrācijas pieprasījumu no aģenta, tas izsauc darbību. Aģentu automātiskajai reģistrācijai ir jākonfigurē darbība ar notikumu avotu "Autoregistration".

> **Note:**
> Lai aktīvie aģenti tiktu automātiski reģistrēti, tīkla atklāšana nav nepieciešama.

Zabbix lietotāja saskarnē:

1. Dodieties uz *Alerts* > *Actions*.
2. Atlasiet *Autoregistration actions*.
3. Noklikšķiniet uz *Create action*.
4. Cilnē *Action* piešķiriet darbībai nosaukumu.
5. Pēc izvēles norādiet nosacījumus. Nosacījumos resursdatora nosaukumam/resursdatora metadatiem varat izmantot apakšvirknes vai regulārās izteiksmes atbilstību. Ja plānojat izmantot nosacījuma tipu *Host metadata*, skatiet sadaļu Resursdatora metadatu izmantošana.
6. Cilnē *Operations* pievienojiet atbilstošās darbības, piemēram, *Add host*, *Add to host group* (piemēram, *Discovered hosts*), *Link templates* u. c.

> **Note:**
> Ja hosts, kas tiks automātiski reģistrēti, visticamāk, tiks atbalstīti tikai aktīvajai uzraudzībai (piemēram, hosts, kurus no jūsu Zabbix servera atdala ugunsmūris), iespējams, vēlēsieties izveidot īpašu veidni, piemēram, *Template_Linux-active*, ar ko tos saistīt.

Izveidotie hosts tiek pievienoti grupai *Discovered hosts* (pēc noklusējuma; šo iestatījumu var mainīt sadaļā *Administration* > *General* > *Other*). Ja vēlaties, lai hosts tiktu pievienoti citai grupai, pievienojiet darbību *Remove from host group* (norādot "Discovered hosts") un arī darbību *Add to host group* (norādot citu hostu grupu), jo hostam ir jāpieder kādai hostu grupai.

Droša automātiskā reģistrācija

Droša automātiskās reģistrācijas metode ir iespējama, konfigurējot uz PSK balstītu autentifikāciju ar šifrētiem savienojumiem.

Šifrēšanas līmenis tiek konfigurēts globāli sadaļā *Administration* > *General* > *Autoregistration*. Var izvēlēties, vai neizmantot šifrēšanu, izmantot TLS šifrēšanu ar PSK autentifikāciju vai izmantot abas iespējas (tādējādi daži hosti var reģistrēties bez šifrēšanas, bet citi — izmantojot šifrēšanu).

PSK autentifikāciju pirms hosta pievienošanas pārbauda Zabbix serveris. Ja pārbaude ir veiksmīga, hosts tiek pievienots, un *Connections from/to host* tiek iestatīti tikai uz "PSK", izmantojot tādu pašu identitāti un iepriekš koplietoto atslēgu kā globālajā automātiskās reģistrācijas iestatījumā.

Attention:
Lai nodrošinātu automātiskās reģistrācijas drošību instalācijās, kurās tiek izmantoti starpniekserveri, jāiespējo šifrēšana starp Zabbix serveri un starpniekserveri.

DNS izmantošana kā noklusējuma saskarne

HostInterface un HostInterfaceItem konfigurācijas parametri ļauj automātiskās reģistrācijas laikā norādīt pielāgotu hosts saskarnes vērtību.

Precīzāk, tie ir noderīgi, ja hosts jāreģistrē automātiski ar DNS nosaukumu kā noklusējuma aģenta saskarni, nevis ar tā IP adresi. Šādā gadījumā DNS nosaukums jānorāda vai jāatgriež kā viena no parametru HostInterface vai HostInterfaceItem vērtībām. Ja kāda no šo parametru vērtībām mainās, piemēram, no IP adreses uz DNS nosaukumu vai otrādi, automātiski reģistrētā hosts noklusējuma saskarne tiks attiecīgi atjaunināta. Šis atjauninājums tiek piemērots esošajam hosts, nevis izveidots jauns hosts. Lai nosūtītu jauno vērtību, aģents ir jārestartē, lai tas atkārtoti sāktu automātiskās reģistrācijas procesu.

Note:
Ja parametri HostInterface vai HostInterfaceItem nav konfigurēti, tā vietā tiek izmantots parametrs listen_dns. Šī vērtība tiek noteikta, veicot aģenta IP adreses reverso DNS uzmeklēšanu. Ja reversā DNS konfigurācija nav pareizi iestatīta vai atgriež nederīgu nosaukumu, nederīgas saskarnes vērtības dēļ var tikt veikta nepareiza vai neveiksmīga automātiskā reģistrācija.

Hostu metadatu izmantošana

Kad aģents nosūta serverim automātiskās reģistrācijas pieprasījumu, tas nosūta sava hosta nosaukumu. Dažos gadījumos (piemēram, Amazon mākoņa mezgliem) ar hosta nosaukumu nepietiek, lai Zabbix serveris atšķirtu atklātos hostus. Hosta metadatus var pēc izvēles izmantot, lai no aģenta nosūtītu serverim citu informāciju.

Hosta metadati tiek konfigurēti aģenta konfigurācijas failā (`zabbix_agentd.conf`). Hosta metadatus konfigurācijas failā var norādīt divos veidos:

```
HostMetadata
HostMetadataItem
```

Skatiet opciju aprakstu iepriekš norādītajā saitē.

Parametrs HostMetadataItem var atgriezt līdz 65535 UTF-8 koda punktiem. Garāka vērtība tiks saīsināta.

Ņemiet vērā, ka MySQL gadījumā maksimālais efektīvais garums rakstzīmēs būs mazāks, ja atgrieztā vērtība satur daudzbaitu rakstzīmes. Piemēram, vērtība, kas satur tikai 3 baitu rakstzīmes, kopumā būs ierobežota līdz 21844 rakstzīmēm, savukārt vērtība, kas satur tikai 4 baitu rakstzīmes, būs ierobežota līdz 16383 simboliem.

Attention:
Automātiskās reģistrācijas mēģinājums notiek ikreiz, kad aktīvs aģents nosūta serverim pieprasījumu aktīvo pārbaužu atsvaidzināšanai. Laika intervāls starp pieprasījumiem ir norādīts 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.

Zema līmeņa atklāšana

Pārskats Zema līmeņa atklāšana (LLD) automātiski atklāj entītijas (piem., failu sistēmas, tīkla saskarnes) uz hosta un izveido atbilstošos vienumus, trigerus un grafikus bez manuālas iestatīšanas katram atsevišķi.

Lai izmantotu LLD, jūs izveidojat **atklāšanas noteikumu**, kas apkopo JSON datus, kuri apraksta entītijas, un **prototipus** (vienums, trigeris un grafiks) katrai entītijai.

Noteikuma atgrieztajam JSON jābūt objektu masīvam, kur katrs objekts attēlo vienu atklāto entītiju, izmantojot atslēgu-vērtību pārus. Piemēram, atklāšanas noteikums ar `net.if.discovery` var atgriezt:

```
[
  {"{#IFNAME}": "lo"},
  {"{#IFNAME}": "eth0"}
]
```

Pēc tam no prototipiem tiek izveidots atbilstošs vienumu, trigeru un grafiku kopums — pa vienam katrai saskarnei (`lo` un `eth0`).

Ja objekti satur atslēgas formātā `{#MACRO}`, šīs makro tiek izmantotas tieši prototipos. Ja nē, vai ja ir nepieciešamas papildu/pielāgotas makro, tās var definēt manuāli un, izmantojot JSONPath, piesaistīt JSON laukiem.

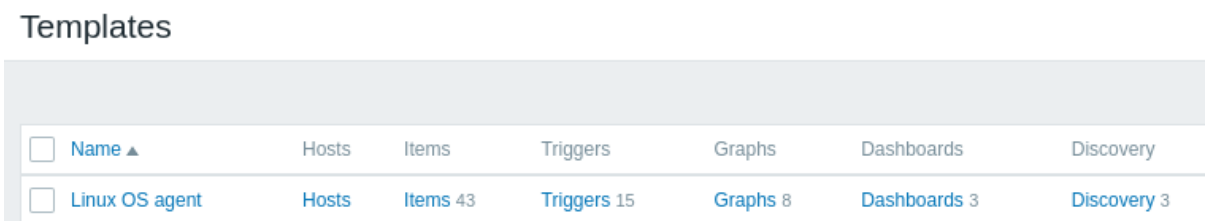
LLD var arī izveidot hosts (piem., virtuālajām mašīnām, kas atklātas uz hipervizora), un atbalsta ligzdotus atklāšanas noteikumus vairāklīmeņu atklāšanai.

Entītijas, kas vairs netiek atklātas, var automātiski atspējot vai dzēst.

Zema līmeņa atklāšanas konfigurēšana Mēs ilustrēsim zema līmeņa atklāšanu, izmantojot failu sistēmas atklāšanas piemēru.

Lai konfigurētu atklāšanu, veiciet šādas darbības:

1. Dodieties uz *Datu vākšana* > *Veidnes* vai *Hosts*.
2. Atbilstošas veidnes/hosta rindā noklikšķiniet uz *Atklāšana*.



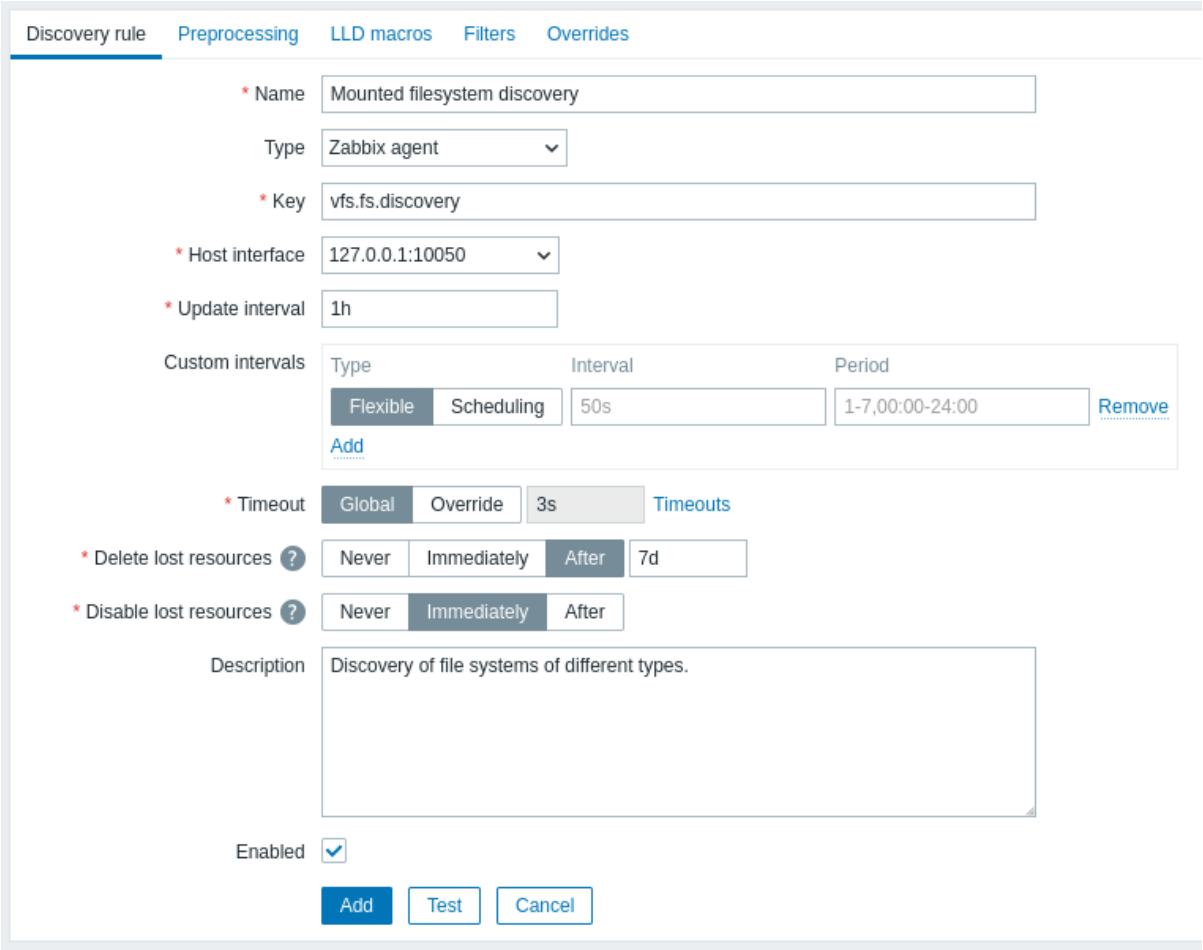
3. Ekrāna augšējā labajā stūrī noklikšķiniet uz *Izveidot atklāšanas noteikumu*.
4. Aizpildiet atklāšanas noteikuma veidlapu ar nepieciešamo informāciju.

Atklāšanas noteikums

Atklāšanas noteikuma veidlapā ir piecas cilnes, kas no kreisās uz labo pusi attēlo datu plūsmu atklāšanas laikā:

- *Atklāšanas noteikums* - norādiet, galvenokārt, iebūvēto vienumu vai pielāgoto skriptu atklāšanas datu iegūšanai.
- *Priekšapstrāde* - veiciet atklāto datu priekšapstrādi.
- *LLD makrosi* - ļauj iegūt makrosu vērtības izmantošanai atklātajos vienumos, trigeros u. c.
- *Filtri* - ļauj filtrēt atklātās vērtības.
- *Ignorēšanas noteikumi* - ļauj modificēt vienumus, trigerus, grafikus vai hostu prototipus, piemērojot tos konkrētiem atklātajiem objektiem.

Cilnē **Atklāšanas noteikums** ir norādīta atklāšanai izmantojamā vienuma atslēga (kā arī daži vispārīgi atklāšanas noteikuma atribūti):



Visi obligātie ievades lauki ir atzīmēti ar sarkanu zvaigznīti.

Parametrs	Apraksts
Nosaukums	Atklāšanas noteikuma nosaukums.
Tips	Pārbaudes tips, kas jāveic atklāšanai. Šajā piemērā tiek izmantots *Zabbix aģenta* vienuma tips. Atklāšanas noteikums var būt arī atkarīgs vienums, kas ir atkarīgs no parasta vienuma. Tas nevar būt atkarīgs no cita atklāšanas noteikuma. Atkarīgam vienumam atlasiet attiecīgo tipu (*Atkarīgs vienums*) un laukā "Galvenais vienums" norādiet galveno vienumu. Galvenajam vienumam ir jāpastāv.
Atslēga	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 vēl viena failu sistēmu atklāšanas iespēja ir izmantot atklāšanas rezultātus, ko nodrošina aģenta atslēga "vfs.fs.get" (skatiet piemēru).
Atjaunināšanas intervāls	Šis lauks norāda, cik bieži Zabbix veic atklāšanu. Sākumā, kad tikai iestatāt failu sistēmu atklāšanu, varat iestatīt nelielu intervālu, bet, kad zināt, ka tā darbojas, varat to iestatīt uz 30 minūtēm vai ilgāk, 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 tas ir iestatīts uz "0" un pastāv pielāgots intervāls (elastīgs vai ieplānots) ar vērtību, kas nav nulle, vienums tiks aptaujāts pielāgotā intervāla laikā. Jauni atklāšanas noteikumi tiks pārbaudīti 60 sekunžu laikā pēc to izveides, ja vien tiem nav iestatīts ieplānots vai elastīgs atjaunināšanas intervāls un *Atjaunināšanas intervāls* nav iestatīts uz 0. *Piezīme*: Esošam atklāšanas noteikumam atklāšanu var veikt nekavējoties, nospiežot *Izpildīt tūlīt* pogu.
Pielāgoti intervāli	Varat izveidot pielāgotus noteikumus vienuma pārbaudei: **Elastīgs** - izveidot izņēmumu *Atjaunināšanas intervālam* (intervālu ar citu biežumu) **Ieplānošana** - izveidot pielāgotu aptaujas grafiku. Detalizētu informāciju skatiet sadaļā Pielāgoti intervāli.

Parametrs	Apraksts
Noildze	Iestatiet atklāšanas pārbaudes noildzi. Atlasiet noildzes opciju: **Globāla** - tiek izmantota starpniekservera/globālā noildze (tā tiek parādīta pelēkajā laukā *Noildze*); **Ignorēt** - tiek izmantota pielāgota noildze (iestatīta laukā *Noildze*; atļautais diapazons: 1–600 s). Tiek atbalstīti laika sufiksi, piemēram, 30s, 1m, un lietotāja makrosi. Noklikšķinot uz saites *Noildzes*, varat konfigurēt starpniekservera noildzes vai globālās noildzes (ja starpniekserveris netiek izmantots). Ņemiet vērā, ka saite *Noildzes* ir redzama tikai lietotājiem ar *Super administratora* tipu, kuriem ir atļaujas sadaļā *Administrēšana* > *Vispārīgi* vai *Administrēšana* > *Starpniekserveri* lietotāja saskarnes sadaļās.
Dzēst zaudētos resursus	Norādiet, cik drīz atklātā entītija tiks dzēsta, kad tās atklāšanas statuss kļūs "Vairs nav atklāta": *Nekad* - tā netiks dzēsta; *Nekavējoties* - tā tiks dzēsta nekavējoties; *Pēc* - tā tiks dzēsta pēc norādītā laika perioda. Vērtībai jābūt lielākai par *Atspējot zaudētos resursus* vērtību. Tiek atbalstīti laika sufiksi, piemēram, 2h, 1d. Tiek atbalstīti lietotāja makrosi. *Piezīme*: Nav ieteicams izmantot opciju "Nekavējoties", jo, vienkārši nepareizi rediģējot filtru, entītija var tikt dzēsta kopā ar visiem vēsturiskajiem datiem. Ņemiet vērā, ka manuāli atspējotus resursus zema līmeņa atklāšana neizdzēsīs.
Atspējot zaudētos resursus	Norādiet, cik drīz atklātā entītija tiks atspējota, kad tās atklāšanas statuss kļūs "Vairs nav atklāta": *Nekad* - tā netiks atspējota; *Nekavējoties* - tā tiks atspējota nekavējoties; *Pēc* - tā tiks atspējota pēc norādītā laika perioda. Vērtībai jābūt lielākai par atklāšanas noteikuma atjaunināšanas intervālu. Ņemiet vērā, ka automātiski atspējoti resursi atkal tiks iespējoti, ja zema līmeņa atklāšana tos atklās atkārtoti. Manuāli atspējoti resursi atkārtotas atklāšanas gadījumā atkal netiks iespējoti. Šis lauks netiek parādīts, ja *Dzēst zaudētos resursus* ir iestatīts uz "Nekavējoties". Tiek atbalstīti laika sufiksi, piemēram, 2h, 1d. Tiek atbalstīti lietotāja makrosi.
Apraksts	Ievadiet aprakstu.
Iespējots	Ja ir atzīmēts, noteikums tiks apstrādāts.

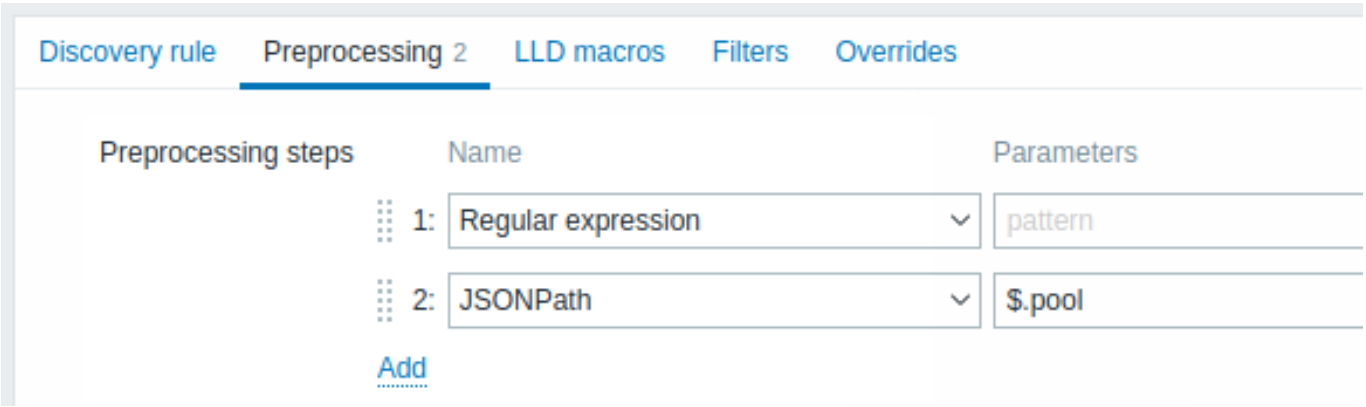
Note:
Atklāšanas noteikumu vēsture netiek saglabāta.

Iepriekšēja apstrāde

Cilne **Iepriekšēja apstrāde** ļauj definēt pārveidošanas noteikumus, kas jāpiemēro atklāšanas rezultātam. Šajā darbībā var izmantot vienu vai vairākas pārveidošanas darbības. Pārveidošanas darbības tiek izpildītas to definēšanas secībā. Visu iepriekšējo apstrādi veic Zabbix serveris.

Skatiet arī:

- Iepriekšējas apstrādes informācija
- Iepriekšējas apstrādes testēšana



Tips	
Pārveidošana	Apraksts
Teksts	
Regulārā izteiksme	Saskaņojiet saņemto vērtību ar <pattern> regulāro izteiksmi un aizstājiet vērtību ar iegūto <output>. Regulārā izteiksme atbalsta ne vairāk kā 10 tverto grupu iegūšanu, izmantojot secību \N. Parametri: **pattern** - regulārā izteiksme **output** - izvades formatēšanas veidne. Escape secība \N (kur N=1...9) tiek aizstāta ar N-to atrasto grupu. Escape secība \0 tiek aizstāta ar atrasto tekstu. Ja atzīmējat izvēles rūtiņu *Pielāgota darbība kļūmes gadījumā*, varat norādīt pielāgotas kļūdu apstrādes opcijas: atmest vērtību, iestatīt norādītu vērtību vai iestatīt norādītu kļūdas ziņojumu.
Aizstāšana	Atrodiet meklējamo virkni un aizstājiet to ar citu virkni (vai neko). Visas meklējamās virknes instances tiks aizstātas. Parametri: **search string** - atrodamā un aizstājamā virkne, reģistrjutīga (obligāts) **replacement** - virkne, ar ko aizstāt meklējamo virkni. Aizstāšanas virkne var būt arī tukša, tādējādi ļaujot atrasto meklējamo virkni dzēst. Lai meklētu vai aizstātu rindu pārtraukumus, rakstatgriezes, tabulācijas un atstarpes, var izmantot escape secības ”\n \r \t \s”; slīpsvītru var aizsargāt kā ”\\” un escape secības var aizsargāt kā ”\\n”. Rindu pārtraukumu, rakstatgriezes un tabulāciju aizsargāšana zema līmeņa atklāšanas laikā tiek veikta automātiski.
Strukturētie dati	
JSONPath	Iegūstiet vērtību vai fragmentu no JSON datiem, izmantojot JSONPath funkcionalitāti. Ja atzīmējat izvēles rūtiņu *Pielāgota darbība kļūmes gadījumā*, varat norādīt pielāgotas kļūdu apstrādes opcijas: atmest vērtību, iestatīt norādītu vērtību vai iestatīt norādītu kļūdas ziņojumu.

XML XPath	Iegūstiet 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)` iegūs 10 no `<document><item><value>10</value></item></document>`
	`number(/document/item/@attribute)` iegūs 10 no `<document><item attribute="10"></item></document>`
	`/document/item` iegūs `<item><value>10</value></item>` no `<document><item><value>10</value></item></document>`
	Ņemiet vērā, ka nosaukumvietas netiek atbalstītas.
	Ja atzīmējat izvēles rūtiņu *Pielāgota darbība kļūmes gadījumā*, varat norādīt pielāgotas kļūdu apstrādes opcijas: atmest vērtību, iestatīt norādītu vērtību vai iestatīt norādītu kļūdas ziņojumu.

CSV uz JSON	Pārveidojiet CSV faila datus JSON formātā. Papildinformāciju skatiet: CSV uz JSON iepriekšēja apstrāde.
XML uz JSON	Pārveidojiet XML formāta datus JSON formātā. Papildinformāciju skatiet: Serializācijas noteikumi. Ja atzīmējat izvēles rūtiņu *Pielāgota darbība kļūmes gadījumā*, varat norādīt pielāgotas kļūdu apstrādes opcijas: atmest vērtību, iestatīt norādītu vērtību vai iestatīt norādītu kļūdas ziņojumu.
SNMP	
SNMP walk vērtība	Iegūstiet vērtību, izmantojot norādīto OID/MIB nosaukumu, un lietojiet formatēšanas opcijas: **Nemainīts** - atgriezt Hex-STRING kā neaizsargātu heksadecimālu virkni (*ņemiet vērā*, ka displeja norādes joprojām tiek lietotas); **UTF-8 no Hex-STRING** - pārveidot Hex-STRING par UTF-8 virkni; **MAC no Hex-STRING** - pārveidot Hex-STRING par MAC adreses virkni (kur `' '` tiks aizstāta ar `':'`); **Integer no BITS** - pārveidot bitu virknes pirmos 8 baitus, kas izteikti kā heksadecimālu rakstzīmju secība (piemēram, ”1A 2B 3C 4D”), par 64 bitu unsigned integer. Bitu virknēs, kas garākas par 8 baitiem, nākamie baiti tiks ignorēti. Ja atzīmējat izvēles rūtiņu *Pielāgota darbība kļūmes gadījumā*, varat norādīt pielāgotas kļūdu apstrādes opcijas: atmest vērtību, iestatīt norādītu vērtību vai iestatīt norādītu kļūdas ziņojumu.

Tips	
SNMP walk uz JSON	Pārveidojiet SNMP vērtības JSON formātā. Norādiet lauka nosaukumu JSON un atbilstošo SNMP OID ceļu. Lauku vērtības tiks aizpildītas ar vērtībām no norādītā SNMP OID ceļa. Šo iepriekšējas apstrādes darbību var izmantot SNMP OID atklāšanai. Ir pieejamas līdzīgas vērtību formatēšanas opcijas kā darbībā *SNMP walk vērtība*. Ja atzīmējat izvēles rūtiņu *Pielāgota darbība kļūmes gadījumā*, varat norādīt pielāgotas kļūdu apstrādes opcijas: atmest vērtību, iestatīt norādītu vērtību vai iestatīt norādītu kļūdas ziņojumu.
SNMP get vērtība	Lietojiet formatēšanas opcijas SNMP get vērtībai: **UTF-8 no Hex-STRING** - pārveidot Hex-STRING par UTF-8 virkni; **MAC no Hex-STRING** - pārveidot Hex-STRING par MAC adreses virkni (kur `' '` tiks aizstāta ar `':'`); **Integer no BITS** - pārveidot bitu virknes pirmos 8 baitus, kas izteikti kā heksadecimālu rakstzīmju secība (piemēram, ”1A 2B 3C 4D”), par 64 bitu unsigned integer. Bitu virknēs, kas garākas par 8 baitiem, nākamie baiti tiks ignorēti. Ja atzīmējat izvēles rūtiņu *Pielāgota darbība kļūmes gadījumā*, varat norādīt pielāgotas kļūdu apstrādes opcijas: atmest vērtību, iestatīt norādītu vērtību vai iestatīt norādītu kļūdas ziņojumu.
Pielāgoti skripti	
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 iepriekšēja apstrāde
Validācija	
Neatbilst regulārajai izteiksmei	Norādiet regulāro izteiksmi, kurai vērtība nedrīkst atbilst. Piemēram, `Error:(.*?)\.` Ja atzīmējat izvēles rūtiņu *Pielāgota darbība kļūmes gadījumā*, varat norādīt pielāgotas kļūdu apstrādes opcijas: atmest vērtību, iestatīt norādītu vērtību vai iestatīt norādītu kļūdas ziņojumu.
Pārbaudīt kļūdu JSON	Pārbaudiet lietojumprogrammas līmeņa kļūdas ziņojumu, kas atrodas JSONPath norādītajā vietā. Pārtrauciet apstrādi, ja pārbaude ir veiksmīga un ziņojums nav tukšs; pretējā gadījumā turpiniet apstrādi ar vērtību, kas bija pirms šīs iepriekšējas apstrādes darbības. Ņemiet vērā, ka šīs ārējā pakalpojuma kļūdas lietotājam tiek parādītas tādas, kādas tās ir, nepievienojot informāciju par iepriekšējas apstrādes darbību. Piemēram, `$.errors`. Ja tiek saņemts JSON, piemēram, `{"errors":"e1"}`, nākamā iepriekšējas apstrādes darbība netiks izpildīta. Ja atzīmējat izvēles rūtiņu *Pielāgota darbība kļūmes gadījumā*, varat norādīt pielāgotas kļūdu apstrādes opcijas: atmest vērtību, iestatīt norādītu vērtību vai iestatīt norādītu kļūdas ziņojumu.
Pārbaudīt kļūdu XML	Pārbaudiet lietojumprogrammas līmeņa kļūdas ziņojumu, kas atrodas XPath norādītajā vietā. Pārtrauciet apstrādi, ja pārbaude ir veiksmīga un ziņojums nav tukšs; pretējā gadījumā turpiniet apstrādi ar vērtību, kas bija pirms šīs iepriekšējas apstrādes darbības. Ņemiet vērā, ka šīs ārējā pakalpojuma kļūdas lietotājam tiek parādītas tādas, kādas tās ir, nepievienojot informāciju par iepriekšējas apstrādes darbību. Kļūda netiks ziņota, ja neizdosies parsēt nederīgu XML. Ja atzīmējat izvēles rūtiņu *Pielāgota darbība kļūmes gadījumā*, varat norādīt pielāgotas kļūdu apstrādes opcijas: atmest vērtību, iestatīt norādītu vērtību vai iestatīt norādītu kļūdas ziņojumu.
Atbilst regulārajai izteiksmei	Norādiet regulāro izteiksmi, kurai vērtībai jāatbilst. Ja atzīmējat izvēles rūtiņu *Pielāgota darbība kļūmes gadījumā*, varat norādīt pielāgotas kļūdu apstrādes opcijas: atmest vērtību, iestatīt norādītu vērtību vai iestatīt norādītu kļūdas ziņojumu.
Ierobežošana	
Atmest nemainīto ar sirdspukstu	Atmetiet vērtību, ja tā nav mainījusies noteiktajā laika periodā (sekundēs). Sekunžu norādīšanai tiek atbalstītas pozitīvas integer vērtības (minimums - 1 sekunde). Šajā laukā var izmantot laika sufiksus (piemēram, `30s`, `1m`, `2h`, `1d`). Šajā laukā var izmantot lietotāja makro un zema līmeņa atklāšanas makro. Atklāšanas vienumam var norādīt tikai vienu ierobežošanas opciju. Piemēram, `1m`. Ja identisks teksts šim noteikumam tiek nodots divreiz 60 sekunžu laikā, tas tiks atmests. *Piezīme*: Vienumu prototipu mainīšana neatiestata ierobežošanu. Ierobežošana tiek atiestatīta tikai tad, kad tiek mainītas iepriekšējas apstrādes darbības.

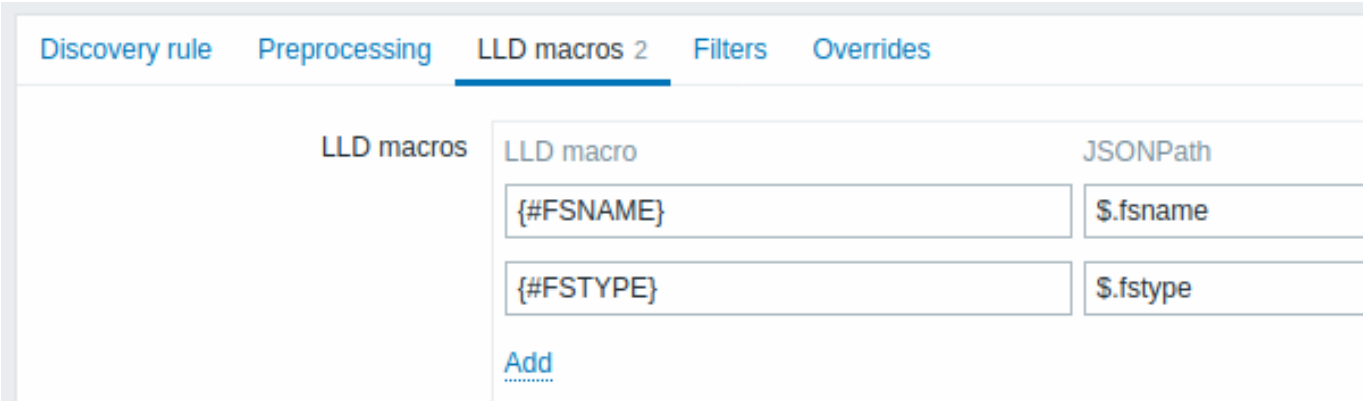
Tips	
Prometheus	
Prometheus uz JSON	Pārveidojiet nepieciešamos Prometheus metrikas datus JSON formātā. Papildinformāciju skatiet sadaļā Prometheus pārbaudes.

Ņemiet vērā, ka, ja atklāšanas noteikums hosts ir lietots, izmantojot veidni, šīs cilnes saturs ir tikai lasāms.

Pielāgotie makro

Cilne **LLD makro** ļauj norādīt pielāgotus LLD makro, ja atgrieztajā JSON jau nav atslēgu formātā `{#MACRO}`.

Piemēram, aģenta vienuma atslēga `vfs.fs.get` atgriež JSON failu sistēmas datus bez jebkādiem iepriekš definētiem LLD makro.
Šeit jūs manuāli definējat makro un piesaistāt tos vērtībām JSON, izmantojot JSONPath izteiksmes:



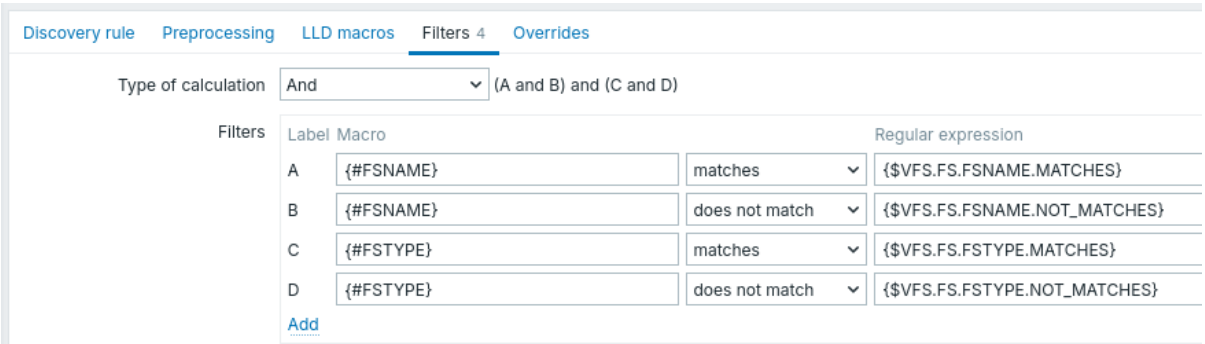
Parameter	Description
LLD macro	LLD makro nosaukums formātā `{#MACRO}`.
JSONPath	Izteiksme, kas nosaka, kāda vērtība jāizvelk no katras atklātās entītijas JSON masīvā. Piemēram, `$.foo` no `[{"foo":"bar"}, {"foo":"baz"}]` izvelk `"bar"` un `"baz"`. JSONPath atbalsta gan punktu, gan iekavu notāciju. Iekavu notācija jāizmanto atslēgām, kas satur speciālās rakstzīmes vai Unicode, piemēram, `$['unicode + special chars #1']['unicode + special chars #2']`.

Atklāšanas noteikumam saknē jāatgriež JSON masīvs (`$.`).
Lai nodrošinātu atpakaļsavietojamību, tiek pieņemts arī JSON objekts, kas satur vienu `"data"` masīvu. Šādā gadījumā `"data"` masīvs tiek automātiski atpakots:

```
{"data":[{"foo":"bar"}, {"foo":"baz"}]}
```

Filtrs

Filtru var izmantot, lai ģenerētu reālus vienumus, trigerus un grafikus tikai tām entītijām, kas atbilst kritērijiem. Cilnē **Filtri** ir ietvertas atklāšanas noteikumu filtru definīcijas, kas ļauj filtrēt atklāšanas vērtības:



Parametrs	Apraksts
Aprēķina veids	Ir pieejamas šādas filtru aprēķināšanas opcijas: **And** - jāizpilda visi filtri; **Or** - pietiek, ja tiek izpildīts viens filtrs; **And/Or** - izmanto *And* ar atšķirīgiem makro nosaukumiem un *Or* ar vienādu makro nosaukumu; **Pielāgota izteiksme** - ļauj definēt pielāgotu filtru aprēķinu. Formulā jāiekļauj visi sarakstā esošie filtri. Ierobežojums - 255 simboli.
Filtri	Ir pieejami šādi filtra nosacījumu operatori: *atbilst*, *neatbilst*, *pastāv*, *nepastāv*. Operatoriem *atbilst* un *neatbilst* ir nepieciešama Perl saderīga regulārā izteiksme (PCRE). Laukā *Regulārā izteiksme* var ievadīt regulāro izteiksmi vai izmantot atsauci uz globālu regulāro izteiksmi. Operatori *pastāv* un *nepastāv* ļauj filtrēt entītijas, pamatojoties uz norādītā LLD makro esamību vai neesamību atbildē. Ņemiet vērā: ja atbildē trūkst filtra makro, atrastā entītija tiks ignorēta, izņemot gadījumu, kad šim makro ir norādīts nosacījums "nepastāv". Ja makro neesamība ietekmē izteiksmes rezultātu, tiks parādīts brīdinājums. Piemēram, ja `{#B}` trūkst šādā izteiksmē: `{#A}` atbilst 1 un `{#B}` atbilst 2 - tiks parādīts brīdinājums `{#A}` atbilst 1 vai `{#B}` atbilst 2 - brīdinājums netiks parādīts.

Piemēram, ja jūs interesē tikai C:, D: un E: failu sistēmas, laukā *Makro* varat ievadīt `{#FSNAME}`, bet laukā *Regulārā izteiksme* - `^C|^D|^E.<br>` Filtrēšana ir iespējama arī pēc failu sistēmu tipiem, izmantojot makro `{#FSTYPE}` (piemēram, `^ext|^reiserfs`), kā arī pēc disku tipiem (to atbalsta tikai Windows aģents), izmantojot makro `{#FSDRIVETYPE}` (piemēram, `fixed`).

Lai pārbaudītu regulāro izteiksmi, varat izmantot `grep -E`, piemēram:

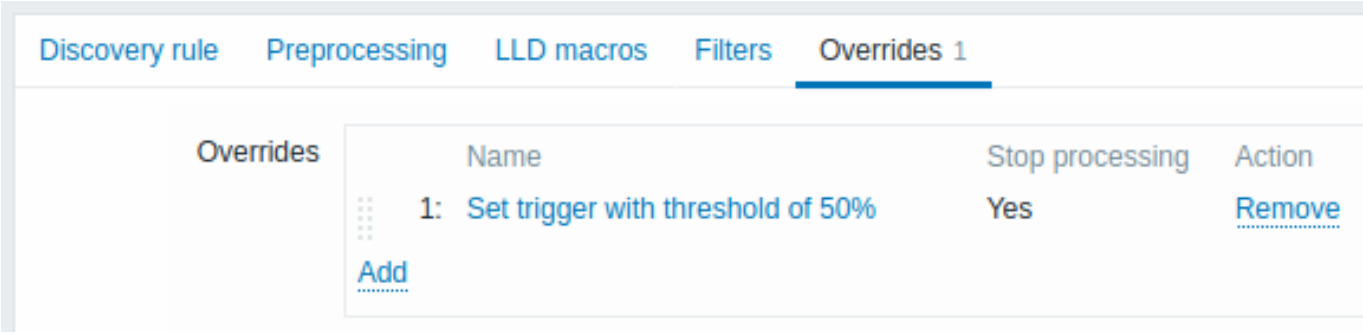
```
for f in ext2 nfs reiserfs smbfs;
do echo $f | grep -E '^ext|^reiserfs' || echo "SKIP: $f";
done
```

Warning:
Kļūda vai drukas kļūda regulārajā izteiksmē, kas izmantota LLD noteikumā (piemēram, nepareiza regulārā izteiksme "Failu sistēmas atklāšanai"), var izraisīt tūkstošiem konfigurācijas elementu, vēsturisko vērtību un notikumu dzēšanu daudziem hostiem.

Attention:
Zabbix datubāze MySQL vidē ir jāizveido, ņemot vērā reģistrjutību, lai pareizi tiktu atklāti failu sistēmu nosaukumi, kas atšķiras tikai pēc reģistra.

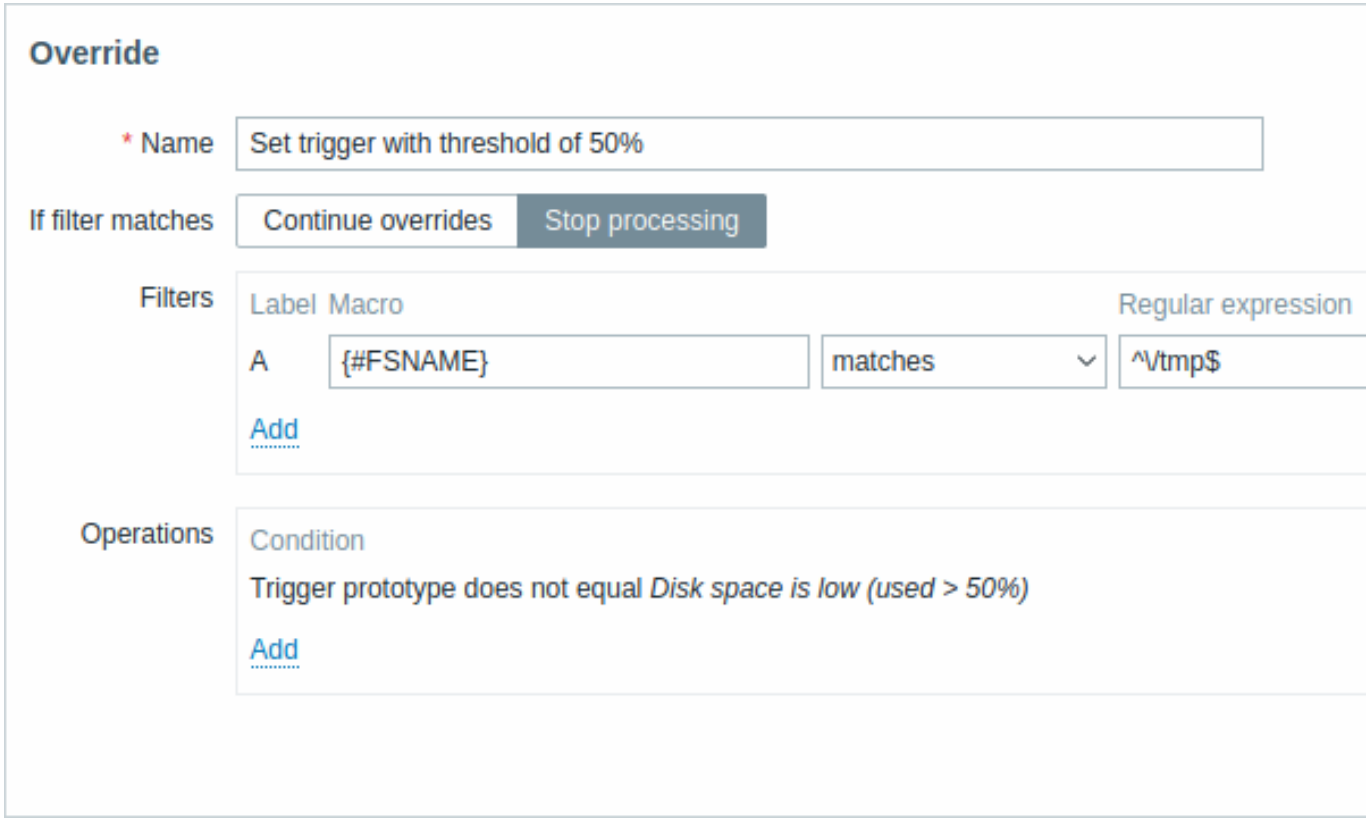
Pārrakstīšana

Cilne **Pārrakstīšanas** ļauj iestatīt noteikumus, lai modificētu vienumu, trigeru, grafiku, hostu un atklāšanas prototipu sarakstu vai to atribūtus atklātajiem objektiem, kas atbilst norādītajiem kritērijiem.



Pārrakstīšanas noteikumi (ja tādi ir) tiek parādīti pārkārtojamā vilkšanas un nomešanas sarakstā un izpildīti to definēšanas secībā.

Lai konfigurētu jaunas pārrakstīšanas detalizētu informāciju, blokā *Pārrakstīšanas* noklikšķiniet uz Add. Lai rediģētu esošu pārrakstīšanas noteikumu, noklikšķiniet uz tā nosaukuma. Tiks atvērts uznirstošais logs, kurā varēsiet rediģēt pārrakstīšanas noteikuma detalizēto informāciju.

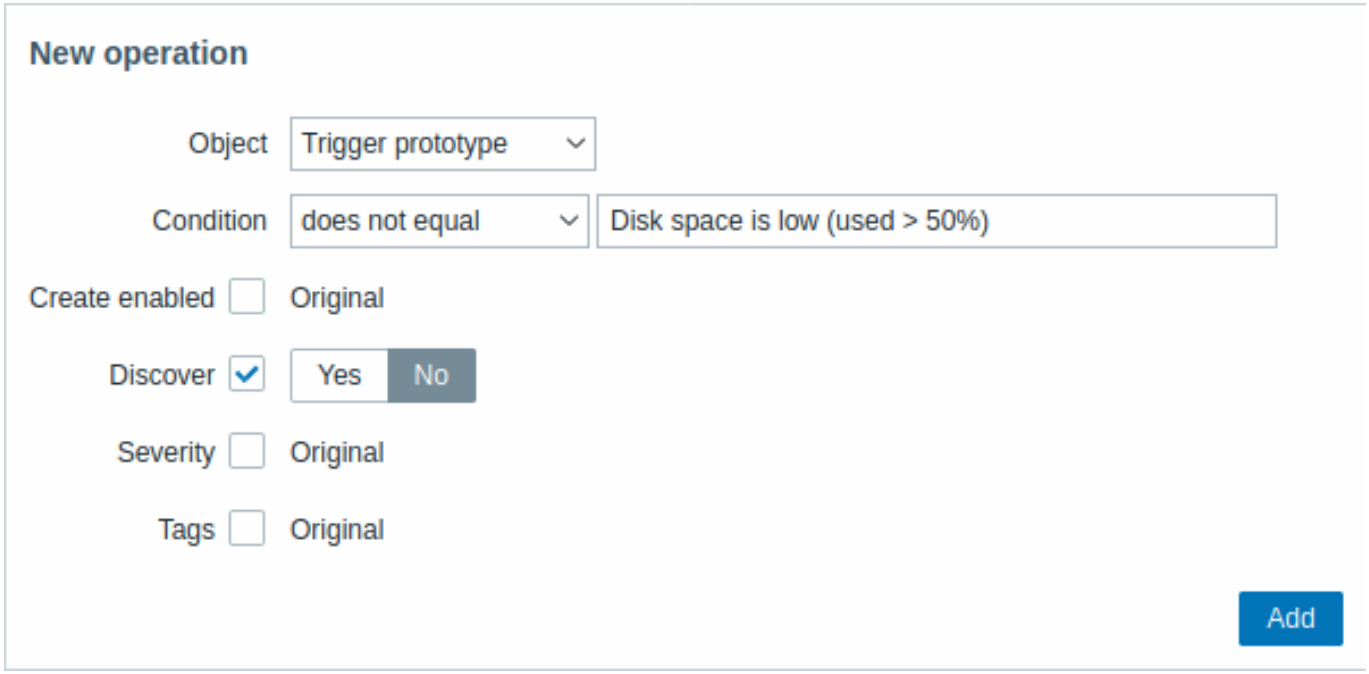


Visi obligātie parametri ir atzīmēti ar sarkanām zvaigznītēm.

Parametrs	Apraksts
Nosaukums	Unikāls (viena LLD noteikuma ietvaros) pārrakstīšanas nosaukums.
Ja filtrs atbilst	Nosaka, vai nākamās pārrakstīšanas ir jāapstrādā, kad filtra nosacījumi ir izpildīti: **Turpināt pārrakstīšanu** - tiks apstrādātas nākamās pārrakstīšanas. **Pārtraukt apstrādi** - tiks izpildītas iepriekšējo pārrakstīšanu (ja tādas ir) un šīs pārrakstīšanas darbības; nākamās pārrakstīšanas atbilstošajām LLD rindām tiks ignorētas.
Filtri	Nosaka, kurām atklātajām entītēm pārrakstīšana jāpiemēro. Pārrakstīšanas filtri tiek apstrādāti pēc atklāšanas noteikuma filtriem, un tiem ir tāda pati funkcionalitāte.
Darbības	Pārrakstīšanas darbības tiek parādītas ar šādu detalizētu informāciju: **Nosacījums** - objekta tips un nosacījums, kuram jāatbilst objekta nosaukumam; piemēram: trigera prototips nav vienāds ar *Disk space is low (used > 50%)*. **Darbības** - tiek parādītas saites darbības rediģēšanai un noņemšanai.

Darbības konfigurēšana

Lai konfigurētu jaunas darbības detalizētu informāciju, blokā Darbības noklikšķiniet uz Add. Lai rediģētu esošu darbību, blakus darbībai noklikšķiniet uz Edit. Tiks atvērts uznirstošais logs, kurā varēsiet rediģēt darbības detalizēto informāciju.



Parametrs	Apraksts
Objekts	Ir pieejami pieci objektu tipi: Vienuma prototips Trigera prototips Grafika prototips Hosta prototips Atklāšanas prototips
Nosacījums	Ļauj filtrēt entītijas, kurām darbība jāpiemēro.
Operators	Atbalstītie operatori: **ir vienāds ar** - piemērot šim prototipam **nav vienāds ar** - piemērot visiem prototipiem, izņemot šo **satur** - piemērot, ja prototipa nosaukums satur šo virkni **nesatur** - piemērot, ja prototipa nosaukums nesatur šo virkni **atbilst** - piemērot, ja prototipa nosaukums atbilst regulārajai izteiksmei **neatbilst** - piemērot, ja prototipa nosaukums neatbilst regulārajai izteiksmei
Modelis	Regulārā izteiksme vai meklējamā virkne.
Objekts: *Vienuma prototips*	
Izveidot iespējotu	Kad izvēles rūtiņa ir atzīmēta, tiks parādītas pogas, kas ļauj pārrakstīt sākotnējos vienuma prototipa iestatījumus: *Jā* - vienums tiks pievienots iespējotā stāvoklī. *Nē* - vienums tiks pievienots atklātai entītijai, bet atspējotā stāvoklī.
Atklāt	Kad izvēles rūtiņa ir atzīmēta, tiks parādītas pogas, kas ļauj pārrakstīt sākotnējos vienuma prototipa iestatījumus: *Jā* - vienums tiks pievienots. *Nē* - vienums netiks pievienots.
Atjaunināšanas intervāls	Kad izvēles rūtiņa ir atzīmēta, tiks parādītas divas opcijas, kas ļauj iestatīt vienumam citu intervālu: *Aizkave* - vienuma atjaunināšanas intervāls. Tiek atbalstīti lietotāja makro, LLD makro un laika sufiksi (piemēram, 30s, 1m, 2h, 1d). Ja tiek izmantots *Pielāgots intervāls*, jāiestata 0. *Pielāgots intervāls* - noklikšķiniet uz Add, lai norādītu elastīgus/plānošanas intervālus. Detalizētu informāciju skatiet sadaļā Pielāgoti intervāli.

Parametrs	Apraksts
Vēsture	Kad izvēles rūtiņa ir atzīmēta, tiks parādītas pogas, kas ļauj iestatīt atšķirīgus vienuma vēstures glabāšanas periodus: *Neglabāt* - ja atlasīts, vēsture netiks glabāta. *Glabāt līdz* - ja atlasīts, labajā pusē tiks parādīts ievades lauks glabāšanas perioda norādīšanai. Tiek atbalstīti lietotāja makro un LLD makro.
Tendences	Kad izvēles rūtiņa ir atzīmēta, tiks parādītas pogas, kas ļauj iestatīt atšķirīgus vienuma tendenču glabāšanas periodus: *Neglabāt* - ja atlasīts, tendences netiks glabātas. *Glabāt līdz* - ja atlasīts, labajā pusē tiks parādīts ievades lauks glabāšanas perioda norādīšanai. Tiek atbalstīti lietotāja makro un LLD makro.
Tagi	Kad izvēles rūtiņa ir atzīmēta, tiks parādīts jauns bloks, kurā varēsiet norādīt taga un vērtības pārus. Tiek atbalstīti lietotāja makro un LLD makro. Šie tagi tiks pievienoti vienuma prototipā norādītajiem tagiem, pat ja tagu nosaukumi sakrīt.
Objekts: *Trigera prototips*	
Izveidot iespējotu	Kad izvēles rūtiņa ir atzīmēta, tiks parādītas pogas, kas ļauj pārrakstīt sākotnējos trigera prototipa iestatījumus: *Jā* - trigeris tiks pievienots iespējotā stāvoklī. *Nē* - trigeris tiks pievienots atklātai entītijai, bet atspējotā stāvoklī.
Atklāt	Kad izvēles rūtiņa ir atzīmēta, tiks parādītas pogas, kas ļauj pārrakstīt sākotnējos trigera prototipa iestatījumus: *Jā* - trigeris tiks pievienots. *Nē* - trigeris netiks pievienots.
Nopietnība	Kad izvēles rūtiņa ir atzīmēta, tiks parādītas trigera nopietnības pogas, kas ļauj mainīt trigera nopietnību.
Tagi	Kad izvēles rūtiņa ir atzīmēta, tiks parādīts jauns bloks, kurā varēsiet norādīt taga un vērtības pārus. Tiek atbalstīti lietotāja makro un LLD makro. Šie tagi tiks pievienoti trigera prototipā norādītajiem tagiem, pat ja tagu nosaukumi sakrīt.
Objekts: *Grafika prototips*	
Atklāt	Kad izvēles rūtiņa ir atzīmēta, tiks parādītas pogas, kas ļauj pārrakstīt sākotnējos grafika prototipa iestatījumus: *Jā* - grafiks tiks pievienots. *Nē* - grafiks netiks pievienots.
Objekts: *Hosta prototips*	
Izveidot iespējotu	Kad izvēles rūtiņa ir atzīmēta, tiks parādītas pogas, kas ļauj pārrakstīt sākotnējos hosta prototipa iestatījumus: *Jā* - hosts tiks izveidots iespējotā stāvoklī. *Nē* - hosts tiks izveidots atspējotā stāvoklī.
Atklāt	Kad izvēles rūtiņa ir atzīmēta, tiks parādītas pogas, kas ļauj pārrakstīt sākotnējos hosta prototipa iestatījumus: *Jā* - hosts tiks atklāts. *Nē* - hosts netiks atklāts.
Saistīt veidnes	Kad izvēles rūtiņa ir atzīmēta, tiks parādīts ievades lauks veidņu norādīšanai. Sāciet rakstīt veidnes nosaukumu vai noklikšķiniet uz *Atlasīt* blakus laukam un uznirstošajā logā atlasiet veidnes no saraksta. Šīs pārrakstīšanas veidnes tiks pievienotas visām veidnēm, kas jau ir saistītas ar hosta prototipu.
Tagi	Kad izvēles rūtiņa ir atzīmēta, tiks parādīts jauns bloks, kurā varēsiet norādīt taga un vērtības pārus. Tiek atbalstīti lietotāja makro un LLD makro. Šie tagi tiks pievienoti hosta prototipā norādītajiem tagiem, pat ja tagu nosaukumi sakrīt.

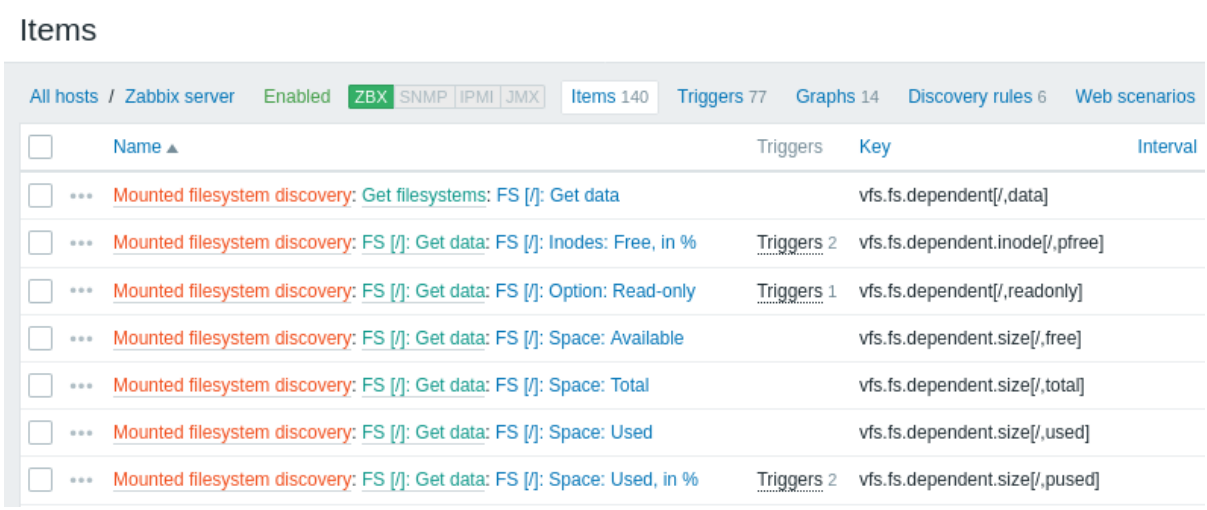
Parametrs	Apraksts
Hosta inventārs	Kad izvēles rūtiņa ir atzīmēta, tiks parādītas pogas, kas ļauj atlasīt citu hosta prototipa inventāra režīmu: *Atspējots* - neaizpildīt hosta inventāru *Manuāls* - manuāli norādīt detalizētu informāciju *Automatizēts* - automātiski aizpildīt hosta inventāra datus, pamatojoties uz apkopotajiem rādītājiem.
Objekts: *Atklāšanas prototips*	
Izveidot iespējotu	Kad izvēles rūtiņa ir atzīmēta, tiks parādītas pogas, kas ļauj pārrakstīt sākotnējos atklāšanas prototipa iestatījumus: *Jā* - atklāšanas noteikums tiks pievienots iespējotā stāvoklī. *Nē* - atklāšanas noteikums tiks pievienots atspējotā stāvoklī.
Atklāt	Kad izvēles rūtiņa ir atzīmēta, tiks parādītas pogas, kas ļauj pārrakstīt sākotnējos atklāšanas prototipa iestatījumus: *Jā* - atklāšanas noteikums tiks pievienots. *Nē* - atklāšanas noteikums netiks pievienots.
Atjaunināšanas intervāls	Kad izvēles rūtiņa ir atzīmēta, tiks parādītas divas opcijas, kas ļauj iestatīt atklāšanas noteikumam citu intervālu: *Aizkave* - noteikuma atjaunināšanas intervāls. Tiek atbalstīti lietotāja makro, LLD makro un laika sufiksi (piemēram, 30s, 1m, 2h, 1d). Ja tiek izmantots *Pielāgots intervāls*, jāiestata 0. *Pielāgots intervāls* - noklikšķiniet uz Add, lai norādītu elastīgus/plānošanas intervālus. Detalizētu informāciju skatiet sadaļā Pielāgoti intervāli.

Veidlapas pogas

Pogas veidlapas apakšdaļā ļauj veikt vairākas darbības.

Add	Pievienot atklāšanas noteikumu. Šī poga ir pieejama tikai jauniem atklāšanas noteikumiem.
Update	Atjaunināt atklāšanas noteikuma rekvizītus. Šī poga ir pieejama tikai esošiem atklāšanas noteikumiem.
Clone	Izveidot citu atklāšanas noteikumu, pamatojoties uz pašreizējā atklāšanas noteikuma rekvizītiem.
Execute now	Nekavējoties veikt atklāšanu, pamatojoties uz atklāšanas noteikumu. Atklāšanas noteikumam jau ir jābūt izveidotam. Skatiet papildinformāciju. *Piezīme*: veicot atklāšanu nekavējoties, konfigurācijas kešatmiņa netiek atjaunināta, tādēļ rezultāts neatspoguļos pavisam nesen veiktās izmaiņas atklāšanas noteikuma konfigurācijā.
Test	Pārbaudīt atklāšanas noteikuma konfigurāciju. Izmantojiet šo pogu, lai pārbaudītu konfigurācijas iestatījumus (piemēram, savienojamību un parametru pareizību), neatgriezeniski nepiemērojot izmaiņas.
Delete	Dzēst atklāšanas noteikumu.
Cancel	Atcelt atklāšanas noteikuma 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.

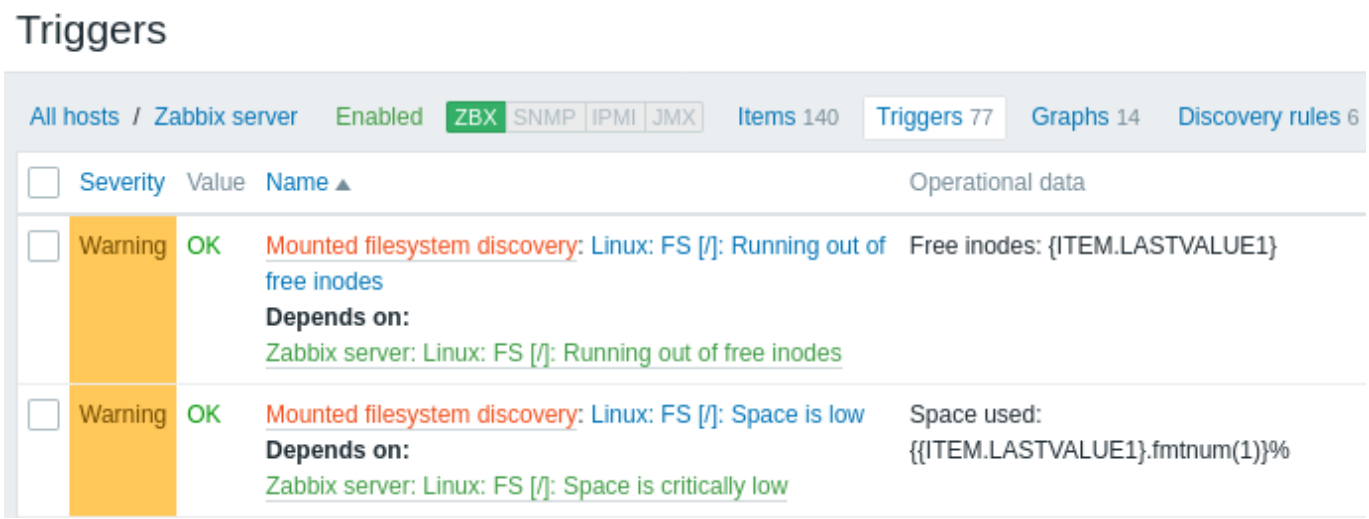
Zaudētie resursi var tikt automātiski dzēsti atkarībā no parametra *Delete lost resources* vērtības. Tas attiecas uz zaudētajiem hostiem, hostu grupām, vienumiem, trigeriem un grafikiem.

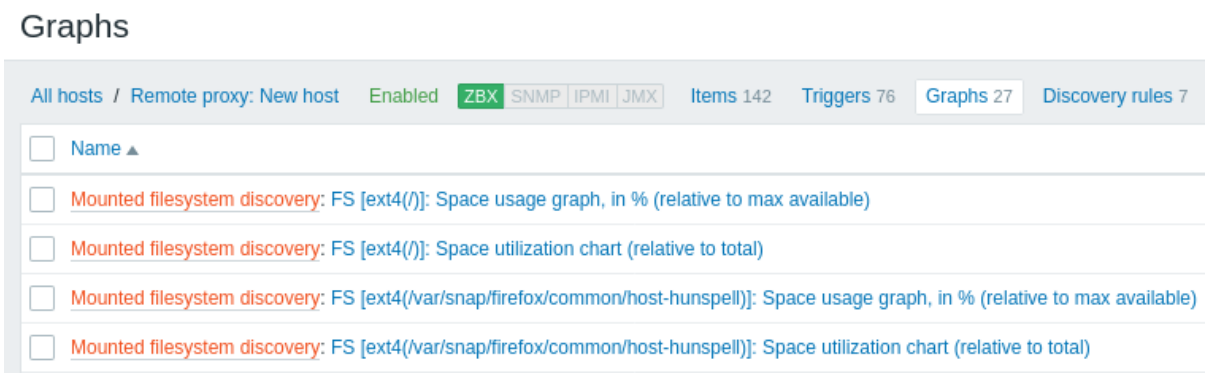
Kad atklātās entītijas kļūst par "Vairs netiek atklātas", entītiju sarakstā tiek parādīts dzīves ilguma indikators. Pārvietojiet peles kursoru virs tā, un tiks parādīts ziņojums ar detalizētu informāciju par statusu.



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:

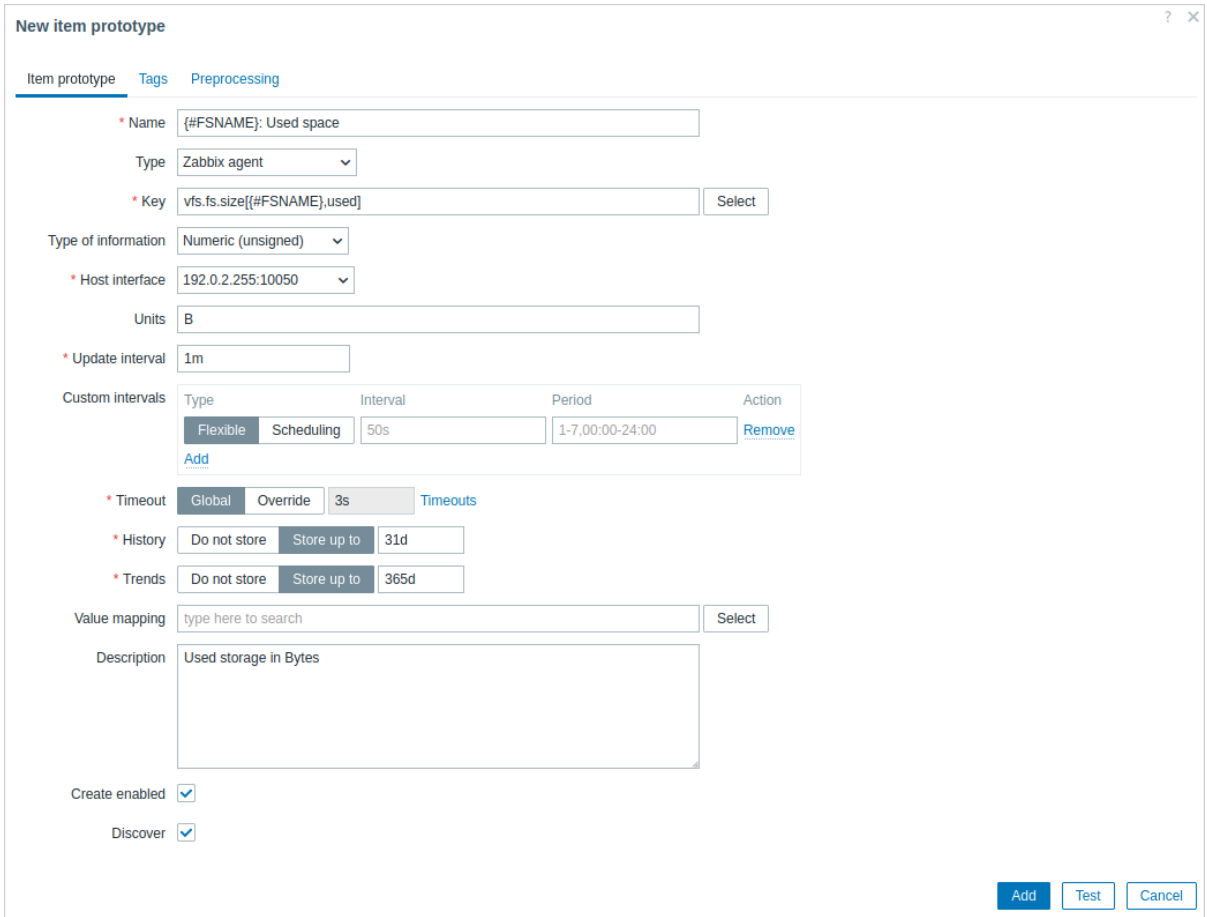
- tīkla saskarņu atklāšana
- CPU un CPU kodolu atklāšana
- SNMP OID atklāšana
- JMX objektu atklāšana;
- atklāšana, izmantojot ODBC SQL vaicājumus
- Windows pakalpojumu atklāšana
- hostu saskarņu atklāšana Zabbix vidē

Plašāku informāciju par JSON formātu atklāšanas vienumiem un piemēru, kā ieviest savu failu sistēmas atklājēju kā Perl skriptu, skatiet sadaļā pielāgotu LLD noteikumu izveide.

Vienumu prototipi

Kad noteikums ir izveidots, pārejiet uz šī noteikuma vienumiem un noklikšķiniet uz *Izveidot vienuma prototipu*, lai izveidotu vienuma prototipu.

Ņemiet vērā, ka `{#FSNAME}` makro tiek izmantots vietās, kur nepieciešams failu sistēmas nosaukums. Zema līmeņa atklāšanas makro izmantošana vienuma atslēgā ir obligāta, lai nodrošinātu pareizu atklāšanas apstrādi. Kad atklāšanas noteikums tiek apstrādāts, šis makro tiks aizstāts ar atklāto failu sistēmu.



Note:
Ja vienuma atslēgas parametri netiek izmantoti, ievietojiet LLD makro vienuma atslēgas iekavās `[...]` (piemēram, `v_[{#MACRO}]`).

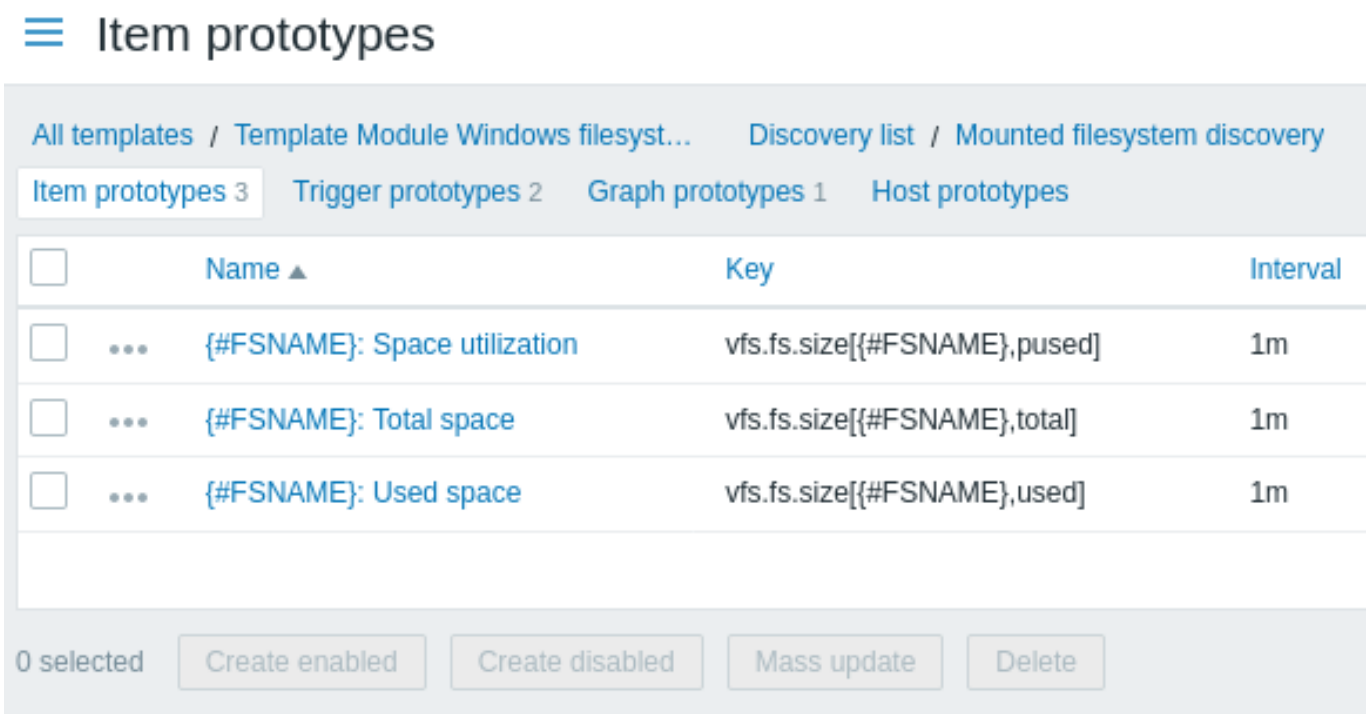
Zema līmeņa atklāšanas makro un lietotāja makro tiek atbalstīti vienuma prototipa konfigurācijā un vienuma vērtības priekšapstrādes parametros. Ņemiet vērā, ka, izmantojot makro atjaunināšanas intervālos, vienam makro ir jāaizpilda viss lauks. Vairāki makro vienā laukā vai makro kopā ar tekstu netiek atbalstīti.

Note:
Zema līmeņa atklāšanas makro kontekstam atbilstoša aizstājējrakstzīmju izmantošana tiek veikta, lai tos droši izmantotu regulārās izteiksmes un XPath priekšapstrādes parametros.

Vienumu prototipiem specifiskie atribūti:

Parametrs	Apraksts
Izveidot iespējotu	Ja ir atzīmēts, vienums tiks pievienots iespējotā stāvoklī. Ja nav atzīmēts, vienums tiks pievienots atklātai entītijai, bet atspējotā stāvoklī.
Atklāt	Ja ir atzīmēts (noklusējums), vienums tiks pievienots atklātai entītijai. Ja nav atzīmēts, vienums netiks pievienots atklātai entītijai, ja vien šis iestatījums atklāšanas noteikumā netiks ignorēts.

Katrai failu sistēmas metrikai, kas mūs interesē, varam izveidot vairākus vienumu prototipus:



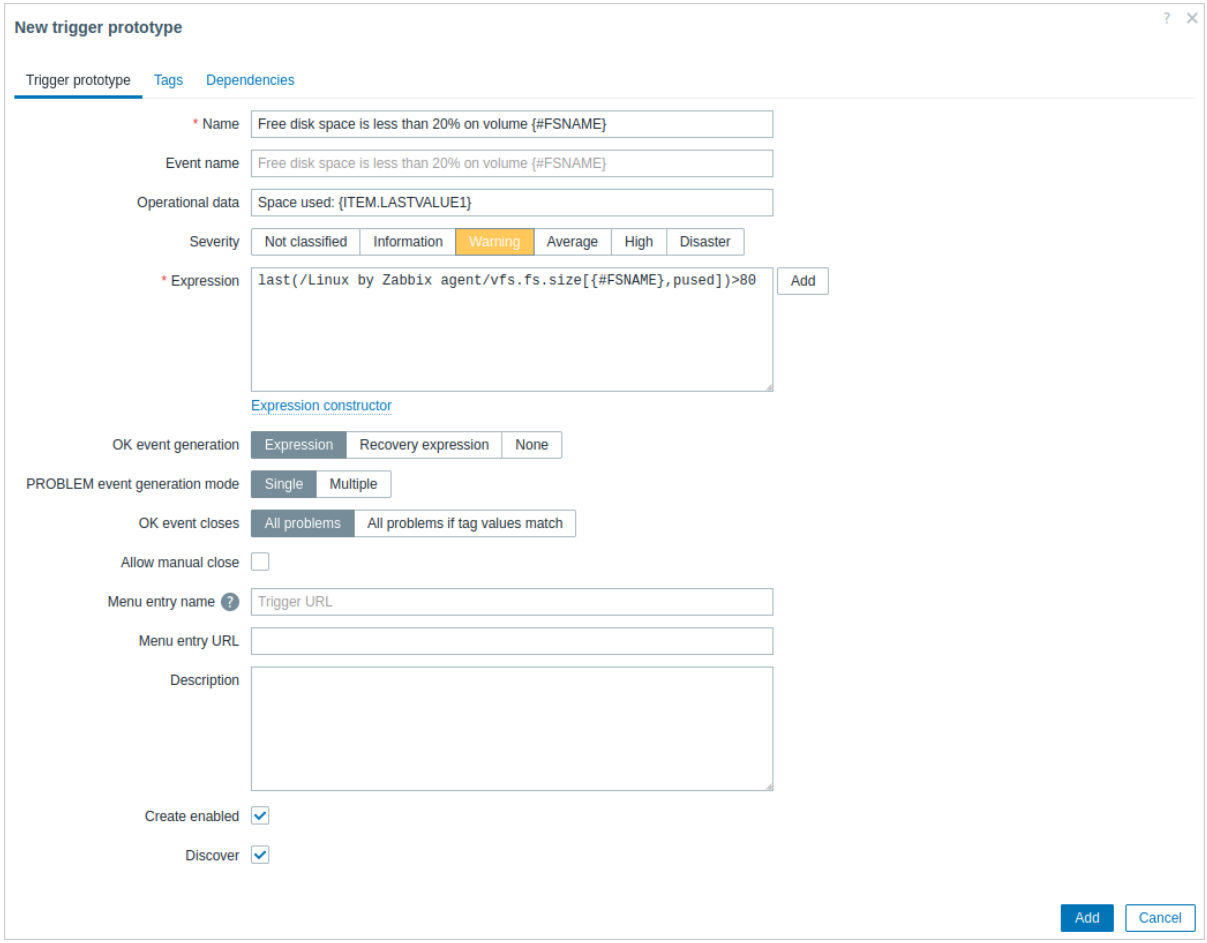
Noklikšķiniet uz trīs punktu ikonas, lai atvērtu konkrētā vienuma prototipa izvēlni ar šādām opcijām:

- *Izveidot trigera prototipu* - izveidot trigera prototipu, pamatojoties uz šo vienuma prototipu
- *Trigeru prototipi* - noklikšķināt, lai skatītu sarakstu ar saitēm uz jau konfigurētajiem šī vienuma prototipa trigeru prototipiem
- *Izveidot atkarīgo vienumu* - izveidot atkarīgo vienumu šim vienuma prototipam

Masveida atjaunināšanas opcija ir pieejama, ja vēlaties vienlaikus atjaunināt vairāku vienumu prototipu rekvizītus.

Trigeru prototipi

Mēs veidojam trigeru prototipus līdzīgā veidā kā vienumu prototipus:

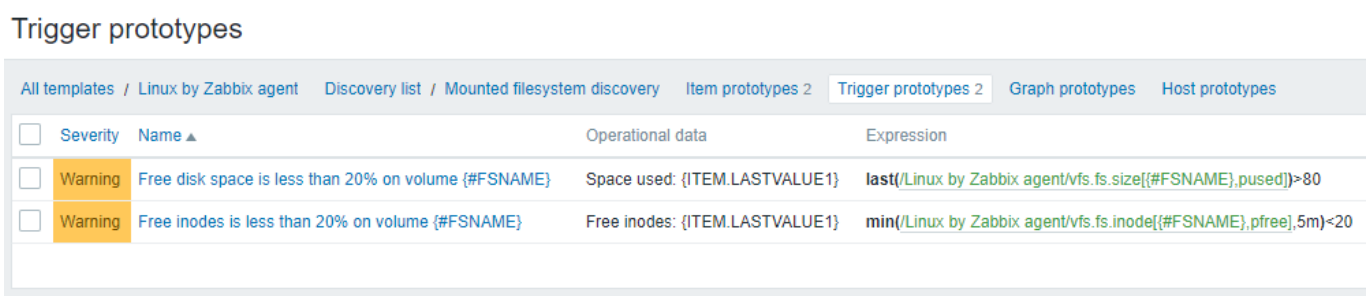


Atribūti, kas ir specifiski trigeru prototipiem:

Parameter	Description
Create enabled	Ja atzīmēts, trigeris tiks pievienots iespējotā stāvoklī. Ja nav atzīmēts, trigeris tiks pievienots atklātajam objektam, bet atspējotā stāvoklī.
Discover	Ja atzīmēts (noklusējums), trigeris tiks pievienots atklātajam objektam. Ja nav atzīmēts, trigeris netiks pievienots atklātajam objektam, ja vien šis iestatījums nav pārrakstīts atklāšanas noteikumā.

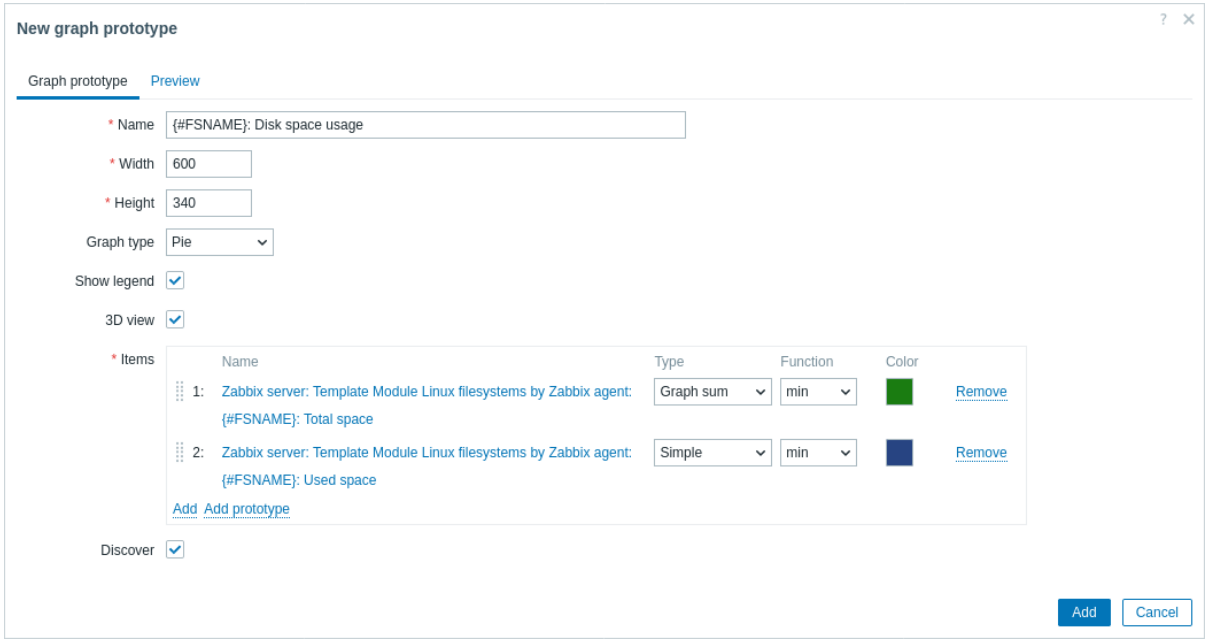
Kad no prototipiem tiek izveidoti reālie trigeri, var rasties nepieciešamība būt elastīgiem attiecībā uz to, kura konstante ('20' mūsu piemērā) tiek izmantota salīdzināšanai izteiksmē. Skatiet, kā lietotāja makro ar kontekstu var būt noderīgi, lai panāktu šādu elastību.

Jūs varat definēt atkarības starp trigeru prototipiem. Lai to izdarītu, dodieties uz cilni *Dependencies*. Trigeru prototips var būt atkarīgs no cita trigeru prototipa no tā paša zemā līmeņa atklāšanas (LLD) noteikuma vai no parasta trigeris. Trigeru prototips nevar būt atkarīgs no trigeru prototipa no cita LLD noteikuma vai no trigeris, kas izveidots no trigeru prototipa. Hosts trigeru prototips nevar būt atkarīgs no trigeris veidnē.



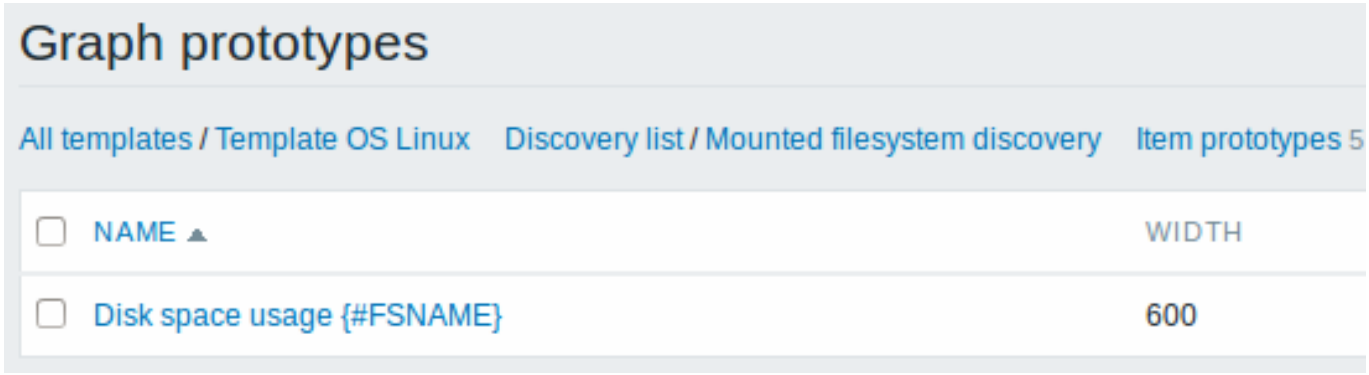
Grafu prototipi

Mēs varam izveidot arī grafu prototipus:

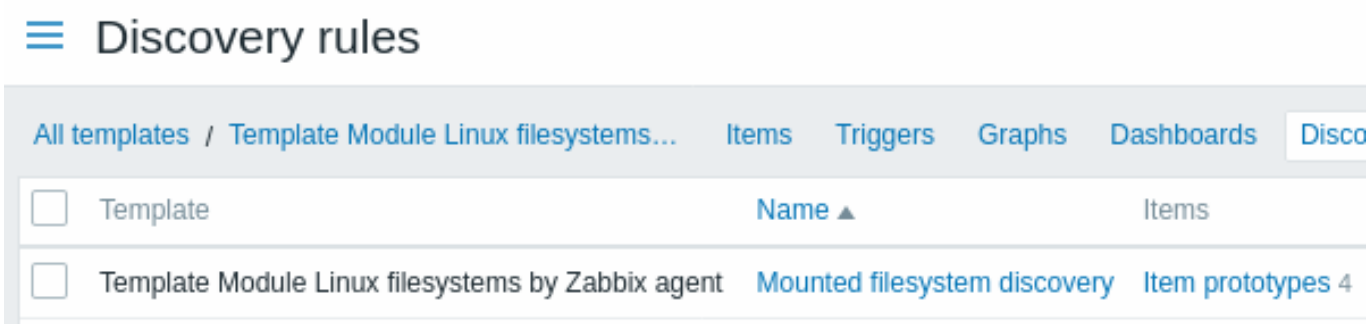


Grafu prototipiem specifiskie atribūti:

Parameter	Description
Discover	Ja atzīmēts (noklusējums), grafiks tiks pievienots atklātajai entītijai. Ja nav atzīmēts, grafiks netiks pievienots atklātajai entītijai, ja vien šis iestatījums nav pārrakstīts atklāšanas noteikumā.



Visbeidzot, esam izveidojuši atklāšanas noteikumu, kas izskatās kā parādīts zemāk. Tam ir pieci vienumu prototipi, divi trigeru prototipi un viens grafu prototips.



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

Hosts prototipi tiek konfigurēti sadaļā zema līmeņa atklāšanas noteikumi.

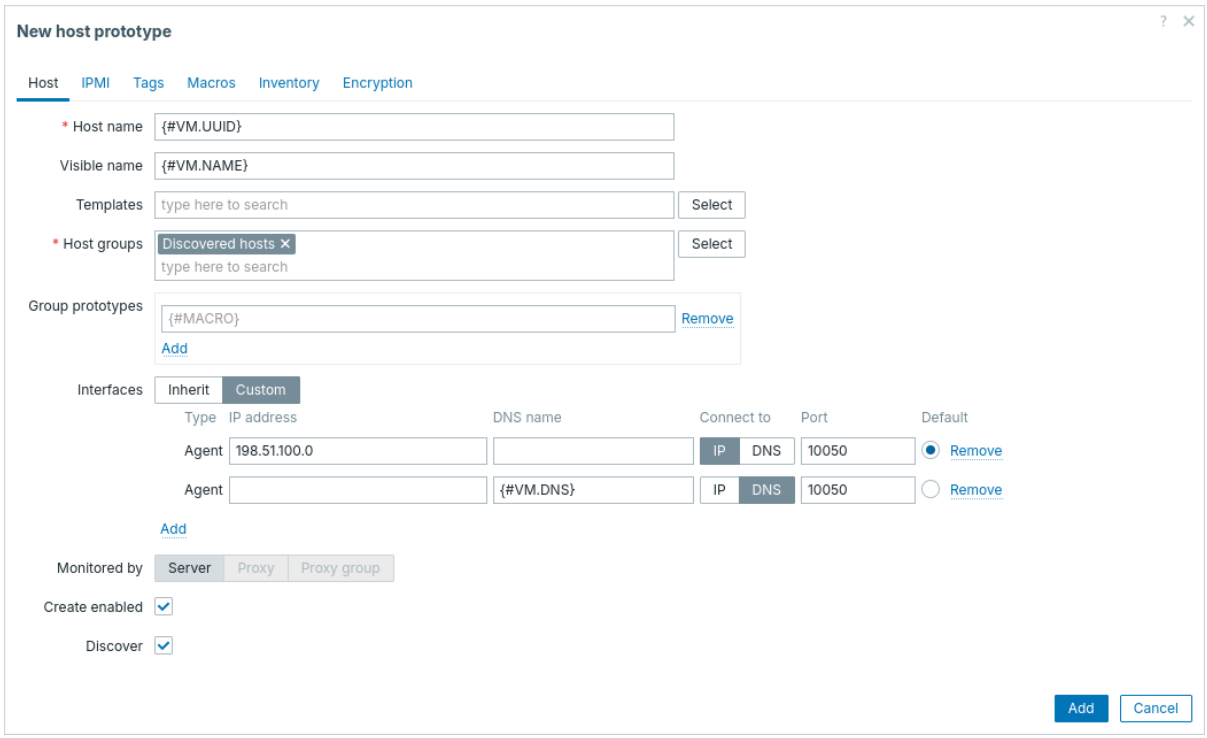
Lai izveidotu hosts prototipu:

1. Dodieties uz *Datu vākšana > Hosts*.

2. Noklikšķiniet uz *Discovery* attiecīgajam hostam, lai pārietu uz šim hostam konfigurēto zema līmeņa atklāšanas noteikumu sarakstu.

3. Noklikšķiniet uz *Host prototypes* attiecīgajam atklāšanas noteikumam.

4. Noklikšķiniet uz pogas *Create host prototype* augšējā labajā stūrī un aizpildiet konfigurācijas formu.



Visi obligātie ievades lauki ir atzīmēti ar sarkanu zvaigznīti.

Parameter	Description
Host name	Šim parametram jāsatur vismaz viens zema līmeņa atklāšanas makro, lai nodrošinātu unikālus hosts nosaukumus izveidotajiem hosts.
Visible name	Tiek atbalstīti zema līmeņa atklāšanas makro.
Group prototypes	Ļauj norādīt hostu grupu prototipus, izmantojot zema līmeņa atklāšanas makro. Balstoties uz norādītajiem grupu prototipiem, tiks atklātas, izveidotas un piesaistītas izveidotajiem hosts host groups; atklātās grupas, kas jau ir izveidotas ar citiem zema līmeņa atklāšanas noteikumiem, arī tiks piesaistītas izveidotajiem hosts. Tomēr atklātās hostu grupas, kas atbilst manuāli izveidotām hostu grupām, netiks piesaistītas izveidotajiem hosts.
Interfaces	Norādiet, vai atklātie hosts pārmanto IP no hosta, kuram pieder atklāšanas noteikums (noklusējums), vai arī saņem pielāgotas saskarnes. Tiek atbalstīti zema līmeņa atklāšanas makro un lietotāja makro.
Monitored by	Šis lauks tiek automātiski sinhronizēts ar attiecīgo lauku hosta konfigurācijā.
Create enabled	Iestatiet atklāto hosts statusu; ja nav atzīmēts, hosts tiks izveidoti kā atspējoti.
Discover	Iestatiet, vai hosts tiks izveidoti no hosta prototipa; ja nav atzīmēts, hosts netiks izveidoti no hosta prototipa (ja vien šis iestatījums nav pārrakstīts zema līmeņa atklāšanas noteikumā).

Note:
Zema līmeņa atklāšanas makro tiek atbalstīti arī tagu vērtībām un hosta prototipa lietotāja makro vērtībām.
 Vērtību kartes hostu prototipiem netiek atbalstītas.

Piemēru, kā konfigurēt hosta prototipu, skatiet sadaļā *Virtuālās mašīnas 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).

Atklāšanas prototipi

Pārskats

Atklāšanas prototipi ir **ligzdoti zema līmeņa atklāšanas** noteikumi "vecākatklāšanas" noteikuma ietvaros, kas ļauj veikt daudzlīmeņu objektu atklāšanu ar tiem piederošajiem vienumiem, trigeriem u. c. Piemēram, iespējams atklāt visus datubāzes servera datubāzu eksemplārus, pēc tam katram eksemplāram atklāt tabulvietas un pēc tam katrai tabulvietai atklāt tabulas.

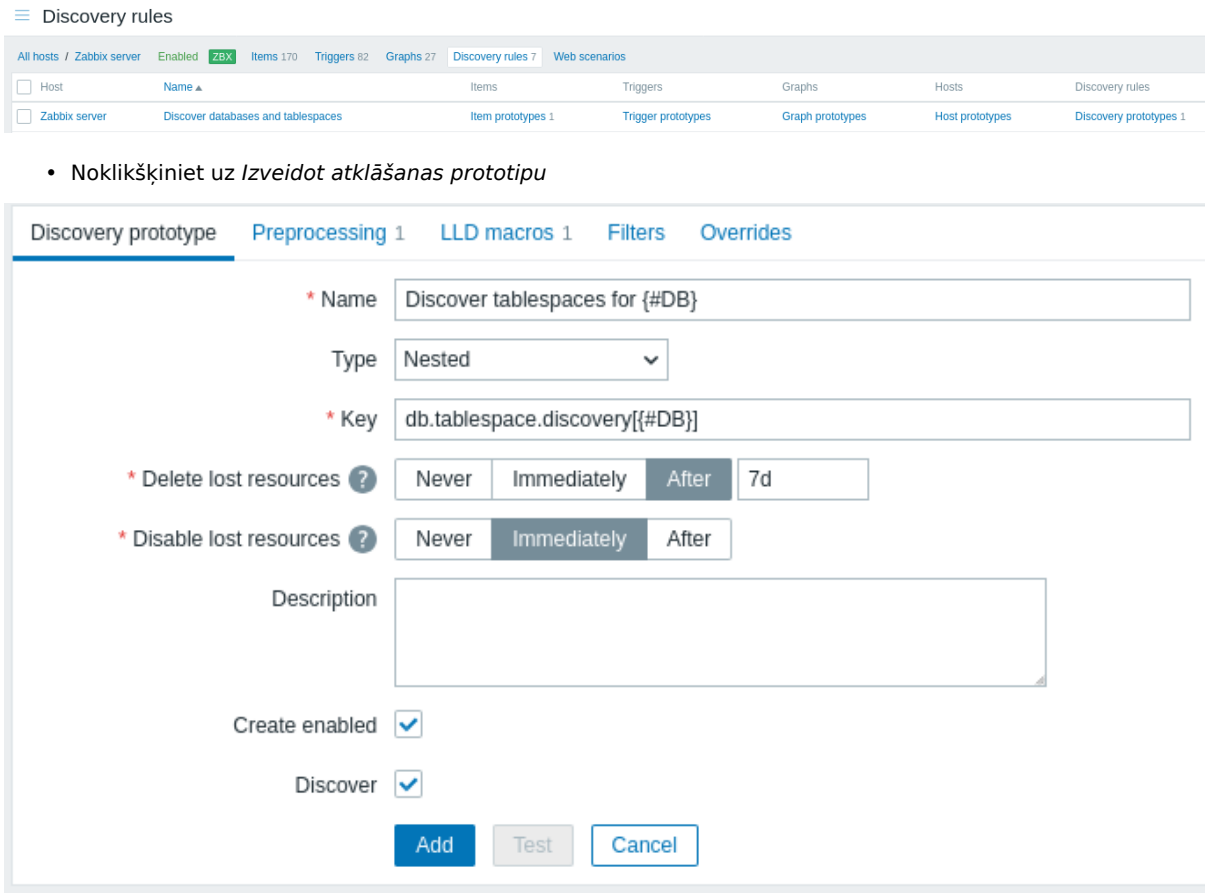
Atklāšanas prototipiem ir savi vienumu, trigeru, grafiku, hostu un atklāšanas prototipi. Ligzdots atklāšanas prototips izmantos to pašu JSON vērtību, ko vecākatklāšanas noteikums, ja norādīsiet tipu *Nested*.

Atklāšanas prototipu ligzdošanas līmeņu skaits nav ierobežots.

Konfigurācija

Lai izveidotu atklāšanas prototipu:

- Noklikšķiniet uz *Atklāšanas prototipi* esošas atklāšanas noteikuma rindā



- Noklikšķiniet uz *Izveidot atklāšanas prototipu*

Šīs formas konfigurācijas lauki ir koplietoti ar parasto zemā 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 vecāknoteikumam. Piemēram, ja sākotnējais JSON ir [<object A>, <object B>] un ir viens ligzdots atklāšanas noteikuma prototips, tad tiks ģenerēti divi atklāšanas noteikumi, attiecīgi balstīti uz object A un object B datiem.

Šajā gadījumā atklāšanas prototips tiek aktivizēts vienlaikus ar vecāknoteikumu. Tādējādi ligzdotais noteikums var izmantot priekšapstrādi, lai strādātu ar citu tās pašas datu "šķēli", ko jau ir ieguvis vecāknoteikums.

LLD makro 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" }
    ]
  }
]
```

1. gadījums

Datubāzes instanču atklāšana datubāzes serverī, pēc tam tabulvietu atklāšana katrai instancei.

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. Pārslēdzieties uz šīs kārtulas cilni *LLD Macros* 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: Nested, Atslēga: `db.tablespace.discovery[{#DB}]`).
7. Pārslēdzieties uz šī atklāšanas prototipa cilni *Preprocessing* un pievienojiet soli `JSONPath=$.tablespaces`.
8. Pārslēdzieties uz šī atklāšanas prototipa cilni *LLD Macros* un pievienojiet makrosu `{#TSNAME}=$.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 *Priekšapstrāde* un pievienojiet soli `JSONPath=$.tablespaces`.
6. Pārslēdzieties uz šīs kārtulas cilni *LLD makrosi*, pievienojiet makrosu `{#TSNAME}=$.name`.
7. Izveidojiet šai kārtulai vienuma prototipu ar nosaukumu *Tabulvietas {#TSNAME} lielums 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 makrosi*, pievienojiet makrosu `{#DB}=$.database`.

10. Pievienojiet šai kārtulai hosta prototipu ar nosaukumu *Hosts datubāzei {#DB}*.
11. Pārslēdzieties uz šī hosta prototipa cilni *Makrosi*, 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 lielums datubāzei db1
	Tabulvietas ts2 lielums datubāzei db1
	Tabulvietas ts3 lielums datubāzei db1
Hosts datubāzei db2	Aktīvie savienojumi ar db2
	Tabulvietas ts1 lielums datubāzei db2
	Tabulvietas ts2 lielums datubāzei db2
	Tabulvietas ts3 lielums datubāzei db2
Hosts datubāzei db3	Aktīvie savienojumi ar db3
	Tabulvietas ts1 lielums datubāzei db3
	Tabulvietas ts2 lielums datubāzei db3
	Tabulvietas ts3 lielums datubāzei db3

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, jādefinē Alias aģenta parametrs, kas ļauj dažādos atklāšanas noteikumos izmantot modificētas atklāšanas vienuma atslēgas, piemēram, `vfs.fs.discovery[foo]`, `vfs.fs.discovery[bar]` u. c.

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.

Atklāšanas noteikumu piemēri

Šī sadaļa sniedz praktiskus piemērus, kā konfigurēt atklāšanas noteikumus dažādiem entītiju veidiem, piemēram, failu sistēmām, tīkla saskarnēm, pakalpojumiem, sensoriem, SNMP objektiem, Java lietotnēm un datubāzēm.

Katrs piemērs parāda, kā var strukturēt atklāšanas noteikumu, tostarp izmantoto vienuma atslēgu vai vaicājumu, JSON izvadi ar LLD makro un to, kā šie rezultāti tiek izmantoti, lai izveidotu vienuma, trigeru un grafiku prototipus.

- Bloka ierīču atklāšana
- CPU un CPU kodolu atklāšana
- hosts saskarņu atklāšana Zabbix
- IPMI sensoru atklāšana
- JMX objektu atklāšana
- Piemontēto failu sistēmu atklāšana
- Tīkla saskarņu atklāšana
- SNMP OID atklāšana
- SNMP OID atklāšana (mantotā)
- systemd pakalpojumu atklāšana
- Windows veiktspējas skaitītāju instanču atklāšana
- Windows pakalpojumu atklāšana

- Atklāšana, izmantojot WMI vaicājumus
- Atklāšana, izmantojot ODBC SQL vaicājumus
- Atklāšana, izmantojot Prometheus datus

Pielāgotie 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 uz vienu entītiju nav ierobežots — kamēr iebūvētās atklāšanas kārtulas atgriež vienu vai divus makrosus (piemēram, divus failu sistēmu atklāšanai), ir iespējams atgriezt vairāk.

Piemērs

Nepieciešamo JSON virknes formātu vislabāk ilustrē piemērs. Pieņemsim, ka izmantojat vecu Zabbix 1.8 aģentu (tādu, kas neatbalsta atslēgu `vfs.fs.discovery`), bet jums joprojām ir nepieciešams atklāt failu sistēmas. Tālāk ir parādīts 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 makro nosaukumiem ir **0-9** , **A-Z** , **_** , **.**. Mazie burti nosaukumos 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 makro nosaukumiem; alternatīva ir izvilkt LLD makro 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 lauka *Filter* vērtībā varat norādīt `{#FSTYPE}` kā makro un `rootfs|ext3` kā regulāro izteiksmi.

Note:
Jums nav obligāti jāizmanto tādi makro nosaukumi kā FSNAME/FSTYPE ar pielāgotām LLD kārtulām; 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, atgrieztā vērtība ir ierobežota līdz 16 MB. Papildinformāciju skatiet sadaļā datu ierobežojumi LLD atgrieztajām vērtībām.

Virtuālās mašīnas 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 ir jākompilē ar konfigurācijas opcijām `--with-libcurl --with-libxml2`, lai iespējotu virtuālo mašīnu uzraudzību. Zabbix pakotnes jau ir kompilētas ar šīm opcijām.

Virtuālo mašīnu uzraudzībai var modificēt šādus Zabbix servera konfigurācijas faila parametrus:

- `StartVMwareCollectors`

Note:
Ieteicams iespējot vairāk kolektoru nekā uzraudzīto VMware pakalpojumu skaits; pretējā gadījumā VMware veiktspējas skaitītāju statistikas izgūšanu var aizkavēt VMware konfigurācijas datu izgūšana (lielās instalācijās tas aizņem kādu laiku).

 Parasti `StartVMwareCollectors` vērtībai nevajadzētu būt mazākai par 2 un tā nedrīkst pārsniegt divkāršu uzraudzīto VMware pakalpojumu skaitu: Pakalpojumu skaits < `StartVMwareCollectors` < (Pakalpojumu skaits * 2). Piemēram, uzraugot vienu VMware pakalpojumu, iestatiet `StartVMwareCollectors` uz 2; uzraugot trīs pakalpojumus, iestatiet `StartVMwareCollectors` uz 5.

 Ņemiet vērā, ka nepieciešamais kolektoru skaits ir atkarīgs arī no VMware vides apjoma, kā arī no 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. Papildu informāciju 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 atklātajiem objektiem. Papildinformāciju skatiet *Hostu prototipi*.

Konfigurācijas piemēri

Pamata piemēru, kā iestatīt Zabbix VMware uzraudzībai, izmantojot *VMware FQDN* veidni, skatiet *Monitor VMware with Zabbix*.

Detalizētāku piemēru, kā izveidot hostu, zema līmeņa atklāšanas noteikumu un hosta prototipu VMware uzraudzībai, skatiet *Setup example*.

Paplašināta reģistrēšana *vmware collector* procesu apkopotos datus var reģistrēt detalizētai atkļūdošanai, izmantojot 5. atkļūdošanas līmeni. Atkļūdošanas līmeni var konfigurēt servera un starpniekservera 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 uz 5 visiem *vmware collector* procesiem, izpildiet šo komandu:

```
zabbix_server -R log_level_increase="vmware collector"
```

Lai palielinātu atkļūdošanas līmeni no 4 uz 5 otrajam *vmware collector* procesam, izpildiet šo komandu:

```
zabbix_server -R log_level_increase="vmware collector,2"
```

Kad VMware kolektora datu paplašināta reģistrēšana vairs nav nepieciešama, ieteicams samazināt atkļūdošanas līmeni uz noklusējuma vērtību (3), izpildot komandu `-R log_level_decrease`.

Problēmu novēršana

- Ja metrikas nav pieejamas, pārliecinieties, ka tās nav padarītas nepieejamas vai pēc noklusējuma izslēgtas jaunākajās VMware vSphere versijās, kā arī ka nav noteikti ierobežojumi veiktspējas metrikas datubāzes vaicājumiem. Plašāku informāciju skatiet ZBX-12094.
- Ja `config.vpxd.stats.maxQueryMetrics` ir nederīgs vai pārsniedz atļauto rakstzīmju maksimālo skaitu, pievienojiet `config.vpxd.stats.maxQueryMetrics` parametru 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 Profiling excessive memory usage with tcmalloc.

VMware monitoringa vienumu atslēgas

Pārskats Šajā lapā ir sniegta informācija par vienkāršajām pārbaudēm, kuras var izmantot VMware vides uzraudzībai. Metrikas ir sagrupētas pēc uzraudzības mērķa.

Atbalstītie 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.	Vispārējais pakalpojums
vmware.fullname	VMware pakalpojuma pilns nosaukums.	
vmware.version	VMware pakalpojuma 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 tagu 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 tagu masīvs.	
vmware.datastore.write	Laiks, kas nepieciešams rakstīšanas operācijai uz datu krātuvi.	
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 tagu masīvs.	
vmware.dvswitch.discovery	VMware vSphere Distributed Switch atklāšana.	vSphere Distributed Switch
vmware.dvswitch.fetchportstate	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.	
vmware.hv.datacenter.name	VMware hipervizora datu centra nosaukums.	

Vienuma atslēga	Apraksts	Vienumu grupa
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 uz datu krātuvi.	
vmware.hv.discovery	VMware hipervizoru atklāšana.	
vmware.hv.diskinfo.get	VMware hipervizora diska 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 ražotāja 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 ievades statistika.	
vmware.hv.network.linkspeed	VMware hipervizora tīkla saskarnes ātrums.	
vmware.hv.network.out	VMware hipervizora tīkla izvades 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 apkopotais sensors.	
vmware.hv.sensors.get	VMware hipervizora HW ražotāja stāvokļa sensori.	
vmware.hv.status	VMware hipervizora statuss.	
vmware.hv.tags.get	VMware hipervizora tagu masīvs.	
vmware.hv.uptime	VMware hipervizora darbības 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 Resource Pool.	Resursu pūls
vmware.rp.memory	VMware resursu pūla atmiņas metrikas.	
vmware.alarms.get	VMware virtuālā centra 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	Procentuālais laiks, kurā virtuālā mašīna nevar darboties, jo tā sacenšas par piekļuvi fiziskajam CPU.	
vmware.vm.cpu.num	Procesoru skaits VMware virtuālajā mašīnā.	
vmware.vm.cpu.readiness	Procentuālais laiks, kurā virtuālā mašīna bija gatava, bet nevarēja tikt ieplānota darbībai uz fiziskā CPU.	
vmware.vm.cpu.ready	Laiks, kurā virtuālā mašīna bija gatava, 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	Viesu fiziskās atmiņas apjoms, kas ir izmainīts uz maiņas vietu.	
vmware.vm.guest.osuptime	Kopējais laiks, kas pagājis kopš pēdējās operētājsistēmas palaišanas.	
vmware.vm.hv.maintenance	VMware virtuālās mašīnas hipervizora apkopes statuss.	

Vienuma atslēga	Apraksts	Vienumu grupa
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 viesu fiziskās atmiņas lapu atbalstam.	
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 izmainītās atmiņas apjoms.	
vmware.vm.memory.size.usage.guest	VMware virtuālās mašīnas viesu atmiņas lietojums.	
vmware.vm.memory.size.usage.host	VMware virtuālās mašīnas hosta atmiņas lietojums.	
vmware.vm.memory.usage	Procentuālā daļa no hosta fiziskās atmiņas, kas ir patērēta.	
vmware.vm.net.if.discovery	VMware virtuālās mašīnas tīkla saskarņu atklāšana.	
vmware.vm.net.if.in	VMware virtuālās mašīnas tīkla saskarnes ievades statistika.	
vmware.vm.net.if.out	VMware virtuālās mašīnas tīkla saskarnes izvades 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 apstiprinātā krātuves vieta.	
vmware.vm.storage.readoio	Vidējais neizpildīto lasīšanas pieprasījumu skaits virtuālajam diskam apkopošanas intervāla laikā.	
vmware.vm.storage.totalreadlatency	Vidējais laiks, kas nepieciešams lasīšanai no virtuālā diska.	
vmware.vm.storage.totalwritelatency	Vidējais laiks, kas nepieciešams rakstīšanai uz virtuālā diska.	
vmware.vm.storage.uncommitted	VMware virtuālās mašīnas neapstiprinātā krātuves vieta.	
vmware.vm.storage.unshared	VMware virtuālās mašīnas nekoplietotā krātuves vieta.	
vmware.vm.storage.writeoio	Vidējais neizpildīto rakstīšanas pieprasījumu skaits virtuālajam diskam apkopošanas intervāla laikā.	
vmware.vm.tags.get	VMware virtuālās mašīnas tagu masīvs.	
vmware.vm.tools	VMware virtuālās mašīnas viesu rīku stāvoklis vai versija.	
vmware.vm.uptime	VMware virtuālās mašīnas darbības laiks.	
vmware.vm.vfs.dev.discovery	VMware virtuālās mašīnas diska 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 servisa 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 (piemēram,`"error,warning,info,user"`). Pēc noklusējuma atspējots.

Komentāri:

- Katram URL drīkst būt tikai viena `vmware.eventlog` vienuma atslēga;
- Skatiet arī filtrēšanas piemēru VMware notikumu žurnāla ierakstiem.

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 tagu 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 krātuves trauksmju dati.
 Atgrieztā vērtība: JSON object.

Parametri:

- **url** - VMware pakalpojuma URL;
- **uuid** - VMware datu krātuves globālais 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 object.

Parametri:

- **url** - VMware pakalpojuma URL;
- **uuid** - VMware datu krātuves globāli 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 object.

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 tagu masīvs.
 Atgrieztā vērtība: JSON objekts.

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 objekts.

Parametri:

- **url** - VMware pakalpojuma URL;
- **uuid** - VMware DVSwitch globālais unikālais identifikators.
- **filter** - viena virkne ar ar komatiem atdalītiem kritērijiem portu atlasīšanai;
- **mode** - *state* (visi XML bez ”config” XML mezgliem, noklusējums) 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 pakalpojuma URL;
- **uuid** - VMware hipervizora globālais 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 globālais 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 objekts.

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 HW ražotāja stāvokļa sensori.
 Atgrieztā vērtība: JSON objekts.

Parametri:

- **url** - VMware pakalpojuma URL;
- **uuid** - VMware hipervizora globālais 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 tagu masīvs.
 Atgrieztā vērtība: JSON object.

Parametri:

- **url** - VMware pakalpojuma URL;
- **uuid** - VMware hipervizora globālais 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 virtuālā centra 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 globālais 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 globālais 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 veiktspējas skaitītāju nosaukumu izveidei VMware.

[2] Šo vienumu vērtības tiek iegūtas 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.

Virtuālās mašīnas atklāšanas atslēgu lauki

Tālāk esošajā tabulā ir uzskaitīti lauki, ko atgriež ar virtuālo mašīnu saistītās atklāšanas atslēgas.

Item key		
Description	**Field**	**Retrieved content**
vmware.cluster.discovery		
Veic klastera 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, virtuālo mašīnu skaitu.
		
		Masīva struktūra:
		`[{`
		`"rpid":"resource group id",`
		`"tags":[{}],`
		`"rpath":"resource group path",`
		`"vm_count":0`
		`}]`

	”tags”	Masīva `"tags"` struktūru skatiet laukā ”tags”.
		Masīvs, kas satur tagus ar taga nosaukumu, aprakstu un kategoriju.
		
		Masīva struktūra:
		`[{`
		`"tag":"tag name",`
		`"tag_description":"tag description",`
		`"category":"tag category"`
		`}]`

vmware.datastore.discovery		
Veic datu krātuves atklāšanu.	{#DATASTORE}	Datu krātuves nosaukums.

	{#DATASTORE.EXTENT}	Masīvs, kas satur datu krātuves extent nodalījuma ID un instances nosaukumu.
		
		Masīva struktūra:
		`[{`
		`"partitionid":1,`
		`"instance":"name"`
		`}]`

	{#DATASTORE.TYPE}	Datu krātuves tips. Vērtību piemēri: `VMFS`, `NFS`, `vsan` u. c.
	{#DATASTORE.UUID}	Datu krātuves identifikators.

Item key		
	”tags”	Masīvs, kas satur tagus ar taga nosaukumu, aprakstu un kategoriju. Masīva struktūra: [{ "tag":"tag name", "tag_description":"tag description", "category":"tag category" }]
vmware.dc.discovery		
Veic datu centra atklāšanu.	{#DATACENTER}	Datu centra nosaukums.
	{#DATACENTERID}	Datu centra identifikators.
	”tags”	Masīvs, kas satur tagus ar taga nosaukumu, aprakstu un kategoriju. Masīva struktūra: [{ "tag":"tag name", "tag_description":"tag description", "category":"tag category" }]
vmware.dvswitch.discovery		
Veic vSphere sadalīto slēdžu atklāšanu.	{#DVS.NAME}	Slēdža nosaukums.
	{#DVS.UUID}	Slēdža identifikators.
vmware.hv.discovery		
Veic hipervizora atklāšanu.	{#HV.UUID}	Unikāls hipervizora identifikators.
	{#HV.ID}	Hipervizora identifikators (HostSystem pārvaldītā objekta nosaukums).
	{#HV.NAME}	Hipervizora nosaukums.
	{#HV.NETNAME}	Hipervizora tīkla resursdatora nosaukums.
	{#HV.IP}	Hipervizora IP adrese, var būt tukša. HA konfigurācijas gadījumā ar vairākām tīkla saskarnēm tiek ievērota šāda saskarnes atlases prioritāte: - priekšroka IP, kas atrodas tajā pašā IP apakštīklā, kur vCenter IP; - priekšroka IP no IP apakštīkla ar noklusējuma vārteju; - priekšroka IP no saskarnes ar zemāko 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” nozīmē nezināmu konteinera tipu.
	”resource_pool”	Masīvs, kas satur resursu pūla datus, tostarp resursu grupas ID, tagu masīvu, resursu pūla ceļu, virtuālo mašīnu skaitu. Masīva struktūra: [{ "rpid":"resource group id", "tags":[{}], "rpath":"resource group path", "vm_count":0 }] Masīva `"tags"` struktūru skatiet laukā ”tags”.

Item key		
	”tags”	Masīvs, kas satur tagus ar taga nosaukumu, aprakstu un kategoriju. Masīva struktūra: [{ "tag":"tag name", "tag_description":"tag description", "category":"tag category" }]
vmware.hv.datastore.discovery		
Veic hipervizora datu krātuves atklāšanu. Ņemiet vērā, ka vairākus hipervizorus var izmantot vienu un to pašu datu krātuvi.	{#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}	Diskā esošo diska nodalījumu skaits.
	”datastore_extent”	Masīvs, kas satur datu krātuves extent instances nosaukumu un nodalījuma ID. Masīva struktūra: [{ "partitionid":1, "instance":"name" }]
	”tags”	Masīvs, kas satur tagus ar taga nosaukumu, aprakstu un kategoriju. Masīva struktūra: [{ "tag":"tag name", "tag_description":"tag description", "category":"tag category" }]
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 virtuālās mašīnas 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}	Viesu virtuālās mašīnas OS saime, var būt tukša.
	{#VM.GUESTFULLNAME}	Pilns viesu virtuālās mašīnas OS nosaukums, var būt tukšs.

Item key		
	{#VM.FOLDER}	Virtuālās mašīnas vecākmapeņu ķēde, var tikt izmantota kā vērtība ligzdotām grupām; mapju nosaukumi tiek apvienoti ar "/". Var būt tukša.
	{#VM.TOOLS.STATUS}	VMware virtuālās mašīnas rīku statuss.
	{#VM.POWERSTATE}	VMware virtuālās mašīnas barošanas stāvoklis (`poweredOff`, `poweredOn` vai `suspended`).
	{#VM.RPOOL.ID}	Resursu pūla identifikators.
	{#VM.RPOOL.PATH}	Pilns resursu pūla ceļš, izņemot "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":"tag name",`
		`"tag_description":"tag description",`
		`"category":"tag category"`
		`}]`

	"vm_customattribute"	Virtuālās mašīnas pielāgoto atribūtu masīvs (ja definēti).
		
		Masīva struktūra:
		`[{`
		`"name":"custom field name",`
		`"value":"custom field value"`
		`}]`

	"net_if"	Virtuālās mašīnas tīkla saskarņu masīvs.
		
		Masīva struktūra:
		`[{`
		`"ifname": "interface name",`
		`"ifdesc": "interface description",`
		`"ifmac": "00:00:00:00:00:00",`
		`"ifconnected": true,`
		`"iftype": "interface type",`
		`"ifbackingdevice": "interface backing device",`
		`"ifdvswitch_uuid": "interface switch uuid",`
		`"ifdvswitch_portgroup": "interface switch port group",`
		`"ifdvswitch_port": "interface switch port",`
		`"ifip": ["interface ip addresses"]`
		`}]`
		
		Atgriezto datu aprakstu skatiet vienuma atslēgā "vmware.vm.net.if.discovery".

vmware.vm.net.if.discovery		
Veic virtuālās mašīnas tīkla saskarņu atklāšanu.	{#IFNAME}	Tīkla saskarnes nosaukums.
	{#IFDESC}	Saskarnes apraksts.
	{#IFMAC}	Saskarnes MAC adrese.
	{#IFCONNECTED}	Saskarnes savienojuma statuss (`false` - atvienota; `true` - savienota).
	{#IFTYPE}	Saskarnes tips.
	{#IFBACKINGDEVICE}	Atbalsta ierīces nosaukums.
	{#IFDVSWITCH.UUID}	Unikāls vSphere Distributed Switch identifikators.
	{#IFDVSWITCH.PORTGROUP}	Sadalītā portu grupa.
	{#IFDVSWITCH.PORT}	vSphere Distributed Switch ports.
	"ifip"	Saskarņu adrešu masīvs.
vmware.vm.vfs.dev.discovery		
Veic virtuālās mašīnas diska ierīces atklāšanu.	{#DISKNAME}	Diska ierīces nosaukums.
vmware.vm.vfs.fs.discovery		

Item key		
Veic virtuālās mašīnas failu sistēmas atklāšanu.	{#FSNAME}	Failu sistēmas nosaukums.

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 norādītas 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,
            "entity": {
                "id": "host-2013",
                "uuid": "4201b707-f737-7cf8-881f-49215d1aad2a",
                "type": "HostSystem",
                "name": "esx-riga-01.prod.local"
            }
        },
        {
            "name": "Host memory usage",
            "system_name": "alarm.HostMemoryUsageAlarm",
            "description": "Default alarm to monitor host memory usage",
            "enabled": true,
            "key": "alarm-4.host-1004",
            "time": "2022-05-16T13:32:42.47863Z",
            "overall_status": "yellow",
            "acknowledged": false,
            "entity": {
                "id": "host-1004",
                "uuid": "5315c818-g848-8dg9-992g-50326e2bbe3a",
                "type": "HostSystem",
                "name": "esx-riga-02.prod.local",
            }
        },
        {
            // other alarms
        }
    ]
}
```

vmware.*.tags.get 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
}
```


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.

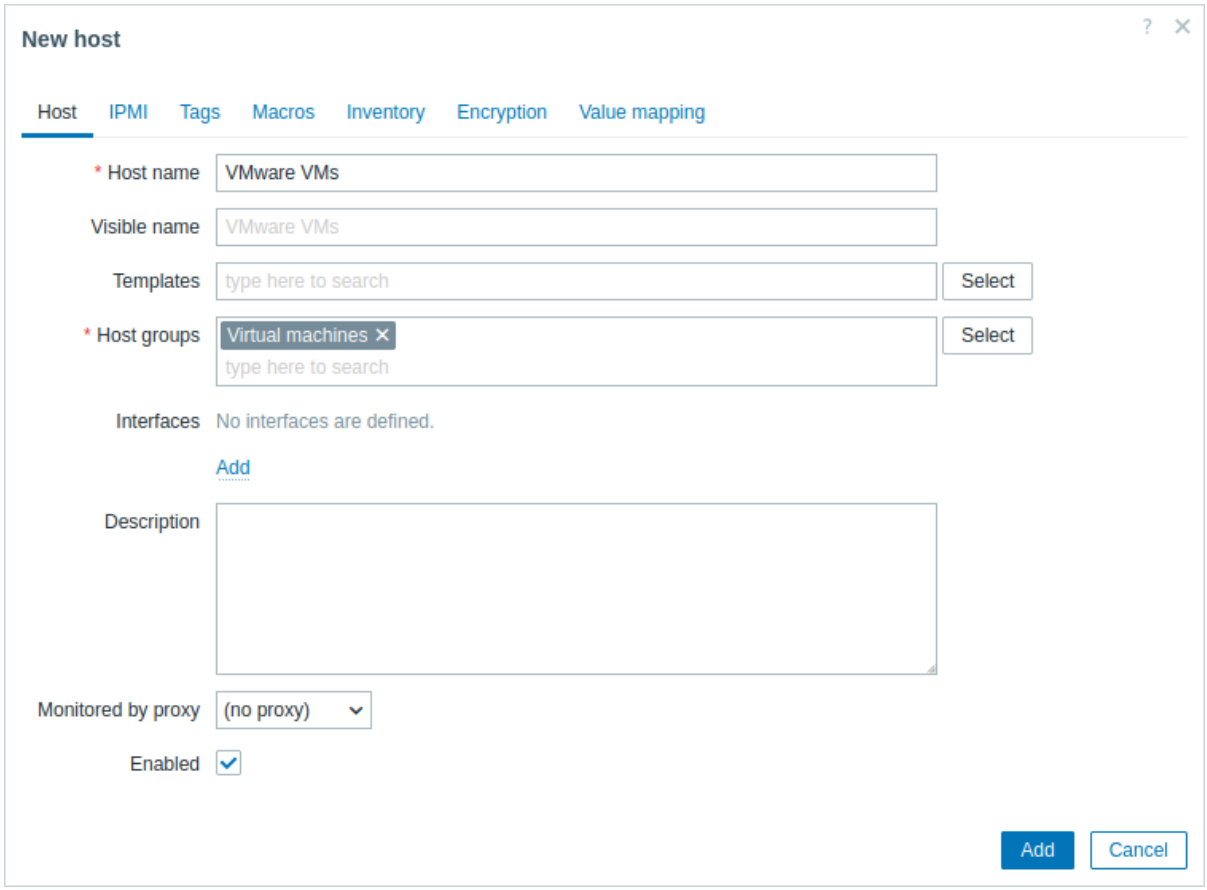
Pirms turpināt, iestatiet `StartVMwareCollectors` parametru Zabbix servera konfigurācijas failā uz 2 vai vairāk (noklusējuma vērtība ir 0).

Izveidot hostu

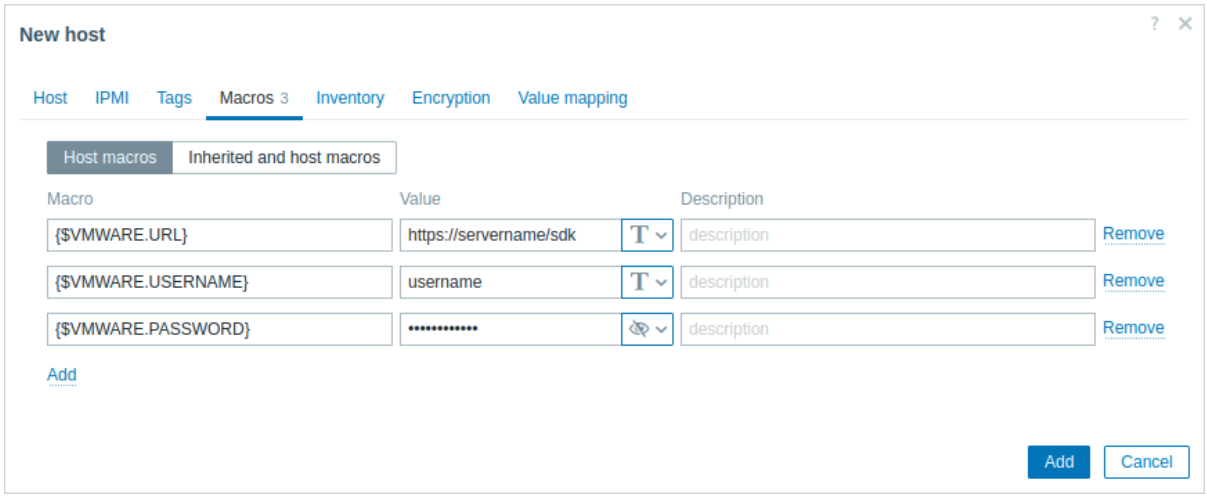
1. Dodieties uz *Datu vākšana* > *Hosts*.

2. Izveidojiet hostu:

- Laukā *Host name* ievadiet hosta nosaukumu (piemēram, "VMware VMs").
- Laukā *Host groups* ievadiet vai atlasiet hostu grupu (piemēram, "Virtual machines").



- Cilnē *Macros* iestatiet šādus hosta makro:
  - {$VMWARE.URL} - VMware pakalpojuma (ESXi hipervizora) SDK URL (https://servername/sdk)
  - {$VMWARE.USERNAME} - VMware pakalpojuma lietotājvārds
  - {$VMWARE.PASSWORD} - VMware pakalpojuma {$VMWARE.USERNAME} lietotāja parole



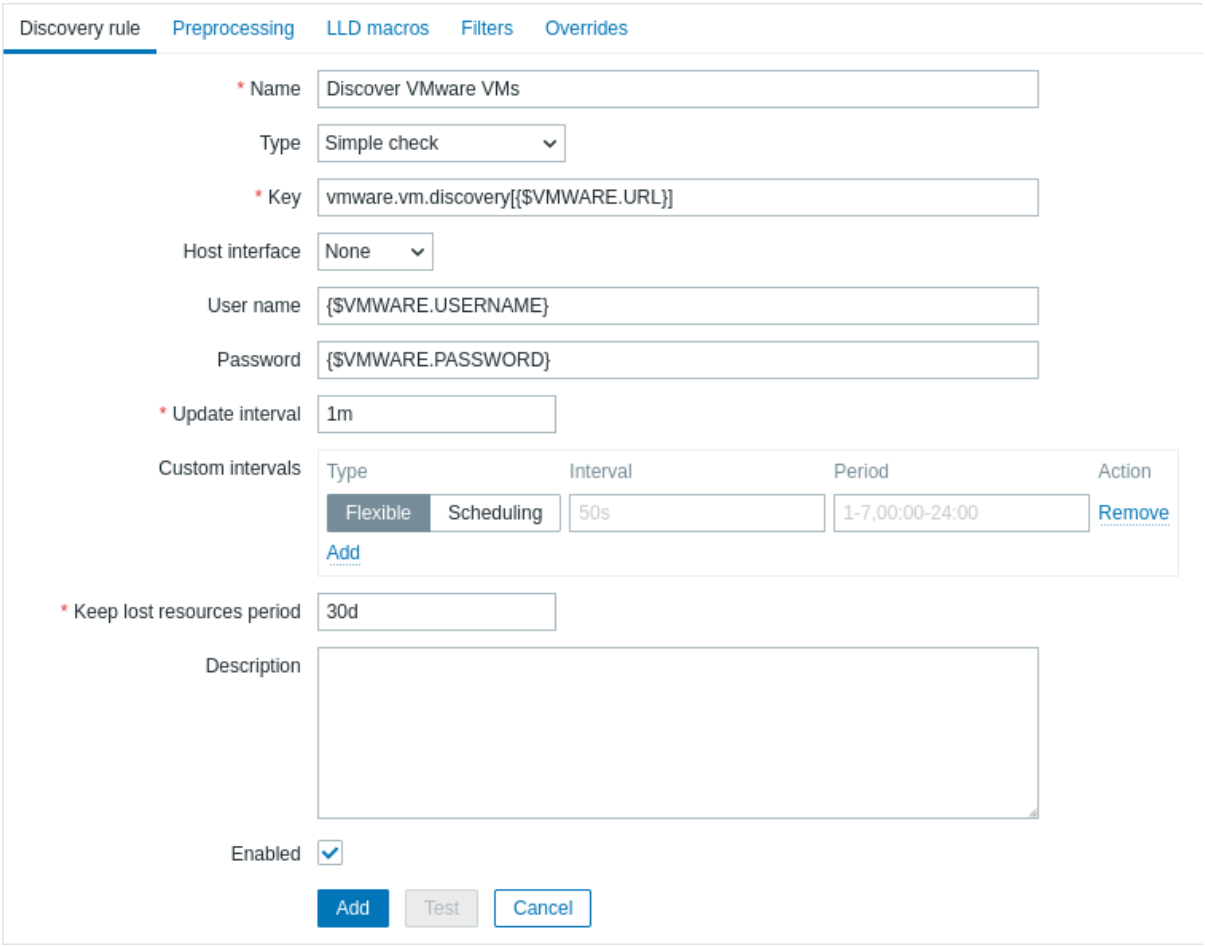
3. Noklikšķiniet uz pogas *Add*, lai izveidotu hostu. Šis hosts reprezentēs jūsu VMware vidi.

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.



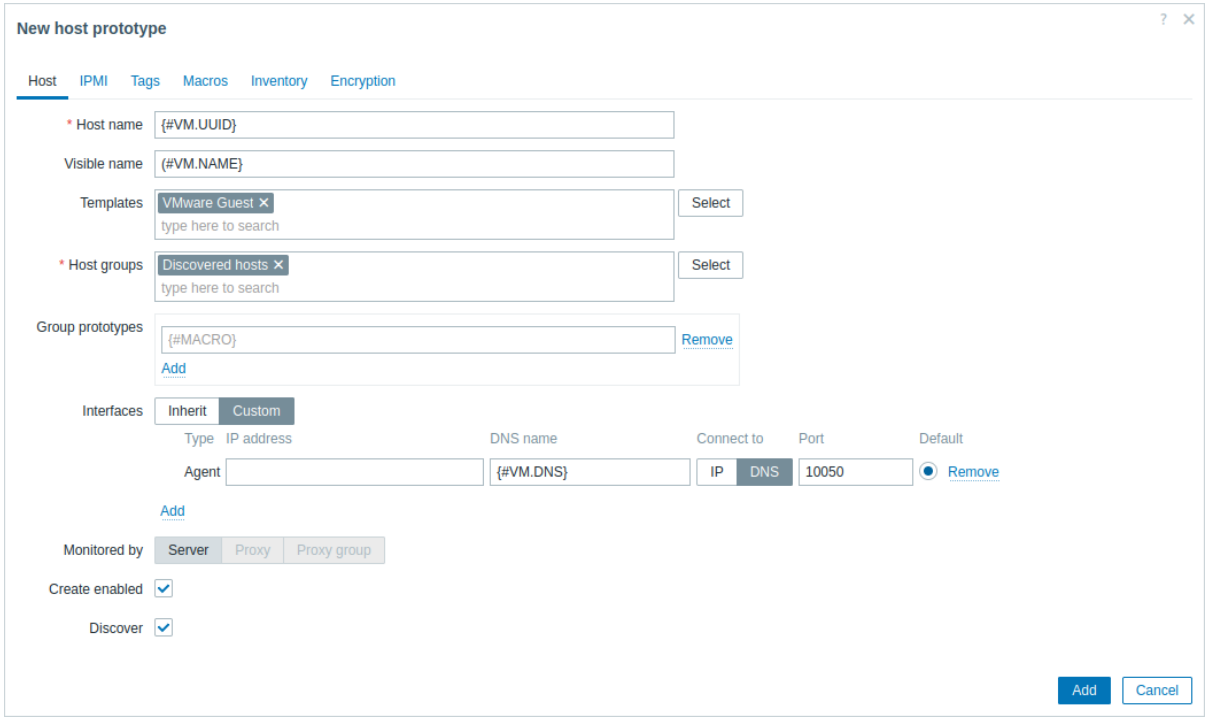
3. Noklikšķiniet uz pogas *Add*, lai izveidotu zema līmeņa atklāšanas kārtulu. Šī atklāšanas kārtula atklās virtuālās mašīnas jūsu VMware vidē.

Izveidot hosta prototipu

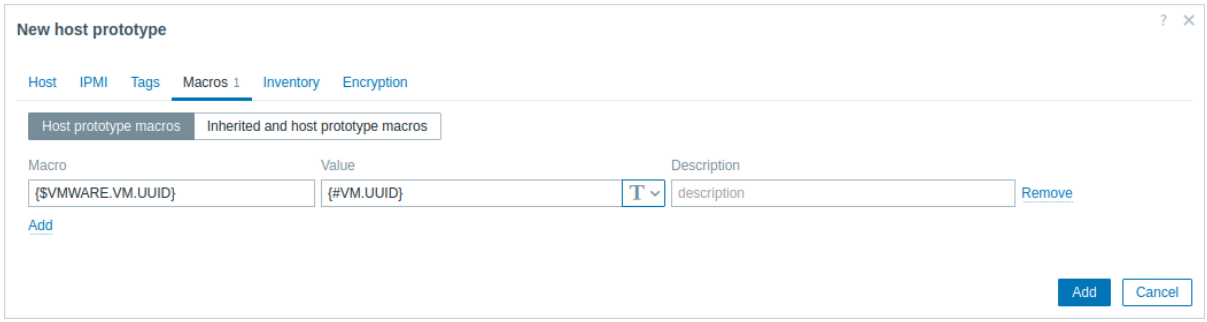
1. Zema līmeņa atklāšanas noteikumu sarakstā noklikšķiniet uz *Host prototypes* iepriekš izveidotajam zema līmeņa atklāšanas noteikumam.

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}]*).

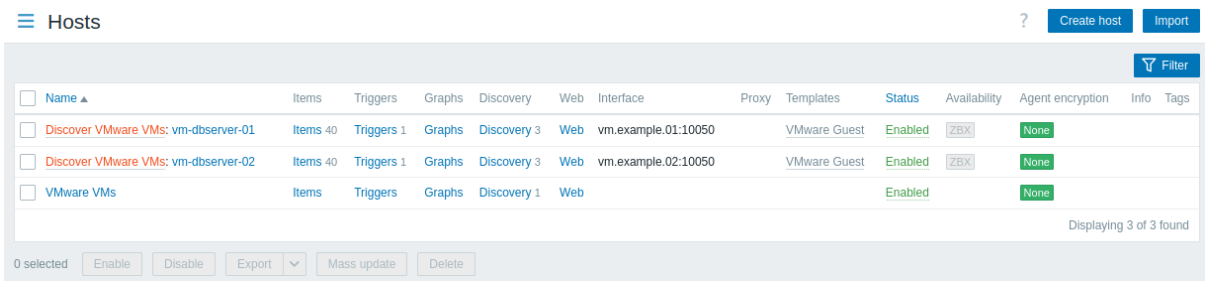


3. Noklikšķiniet uz pogas *Add*, lai izveidotu hosta prototipu. Šis hosta prototips tiks izmantots, lai izveidotu hostus virtuālajām mašīnām, ko atklāja iepriekš izveidotais zema līmeņa atklāšanas noteikums.

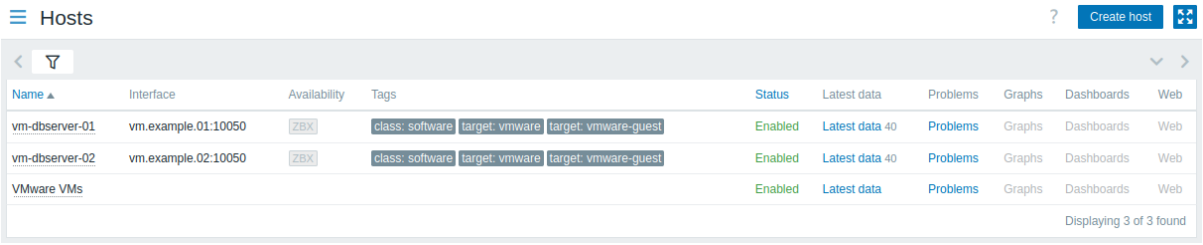
Skatīt hosts un metriku

Pēc tam, kad hosts prototips ir izveidots, zema līmeņa atklāšanas noteikums izveidos hosts atklātajām VMware virtuālajām mašīnām, un Zabbix sāks tās uzraudzīt. Ņemiet vērā, ka hosts atklāšanu un izveidi, ja nepieciešams, var arī izpildīt manuāli.

Lai skatītu izveidotos hosts, dodieties uz izvēlnes sadaļu *Datu vākšana > Hosts*.



Lai skatītu savāktās metrikas, dodieties uz izvēlnes sadaļu *Uzraudzība > Hosts* un noklikšķiniet uz *Jaunākie dati* vienam no hosts.



Paplašināta hosta saskarnes konfigurācija

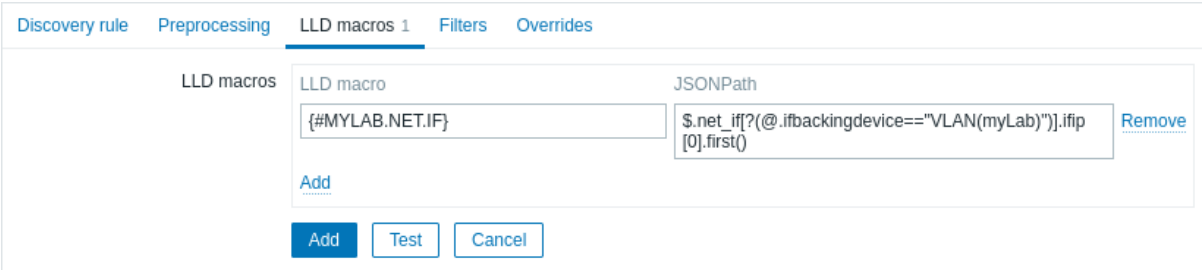
`vmware.vm.discovery[{$VMWARE.URL}]` vienuma atslēga, kas konfigurēta sadaļā *Izveidot zema līmeņa atklāšanas kārtulu*, atgriež tīkla saskarņu datus laukā "net_if":

```
"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": []
      }
    ]
```

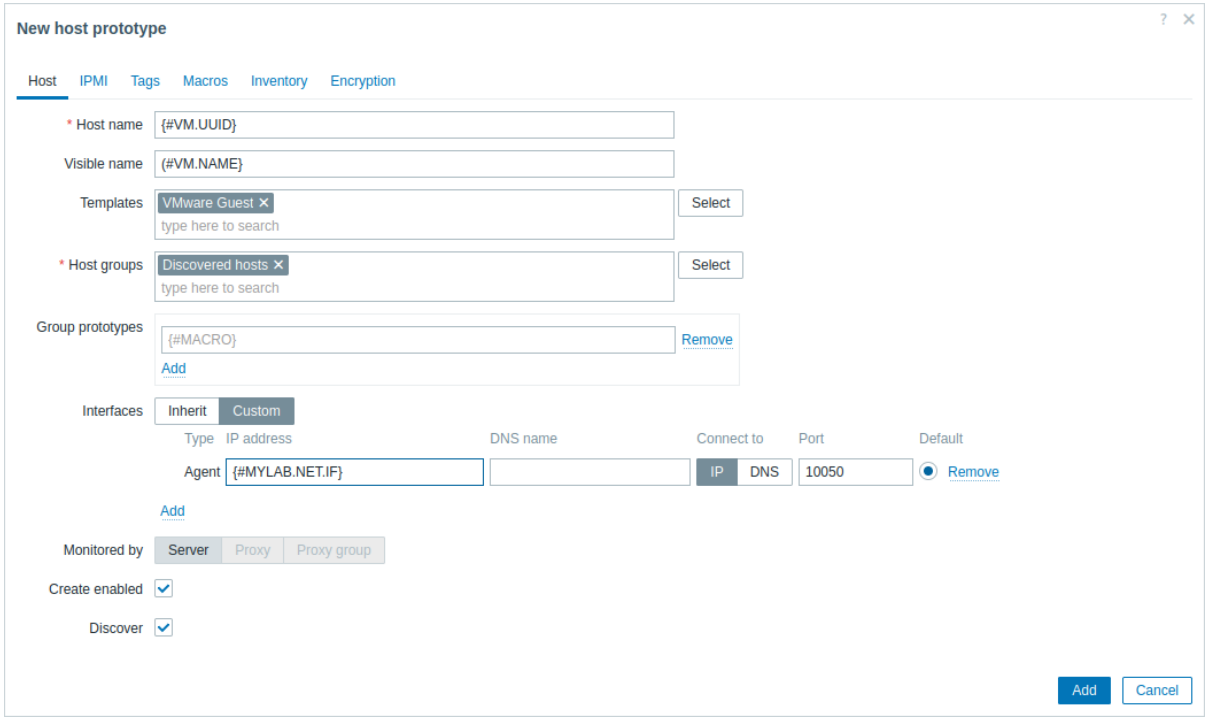
Šos datus var izmantot pielāgotas hosta saskarnes konfigurēšanai.

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*.



Tīmekļa uzraudzība

Pārskats Ar Zabbix 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 tiek periodiski izpildīti ar Zabbix serveris iepriekš noteiktā secībā. Ja hosts tiek uzraudzīts ar starpniekserveris, soļus izpilda starpniekserveris.

Tīmekļa scenāriji tiek piesaistīti hosts/veidnēm tādā pašā veidā kā vienumi, trigeri u. c. Tas nozīmē, ka tīmekļa scenārijus var izveidot arī veidnes līmenī un pēc tam vienā piegājienā piemērot vairākiem hosts.

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.

No tīmekļa scenāriju izpildes iegūtie dati tiek glabāti datubāzē. Šie dati automātiski tiek izmantoti diagrammām, trigeriem un paziņojumiem.

Zabbix var arī pārbaudīt, vai iegūtā HTML lapa satur iepriekš noteiktu virkni. Tas var izpildīt simulētu pieteikšanos un sekot simulētu peles klikšķu ceļam lapā.

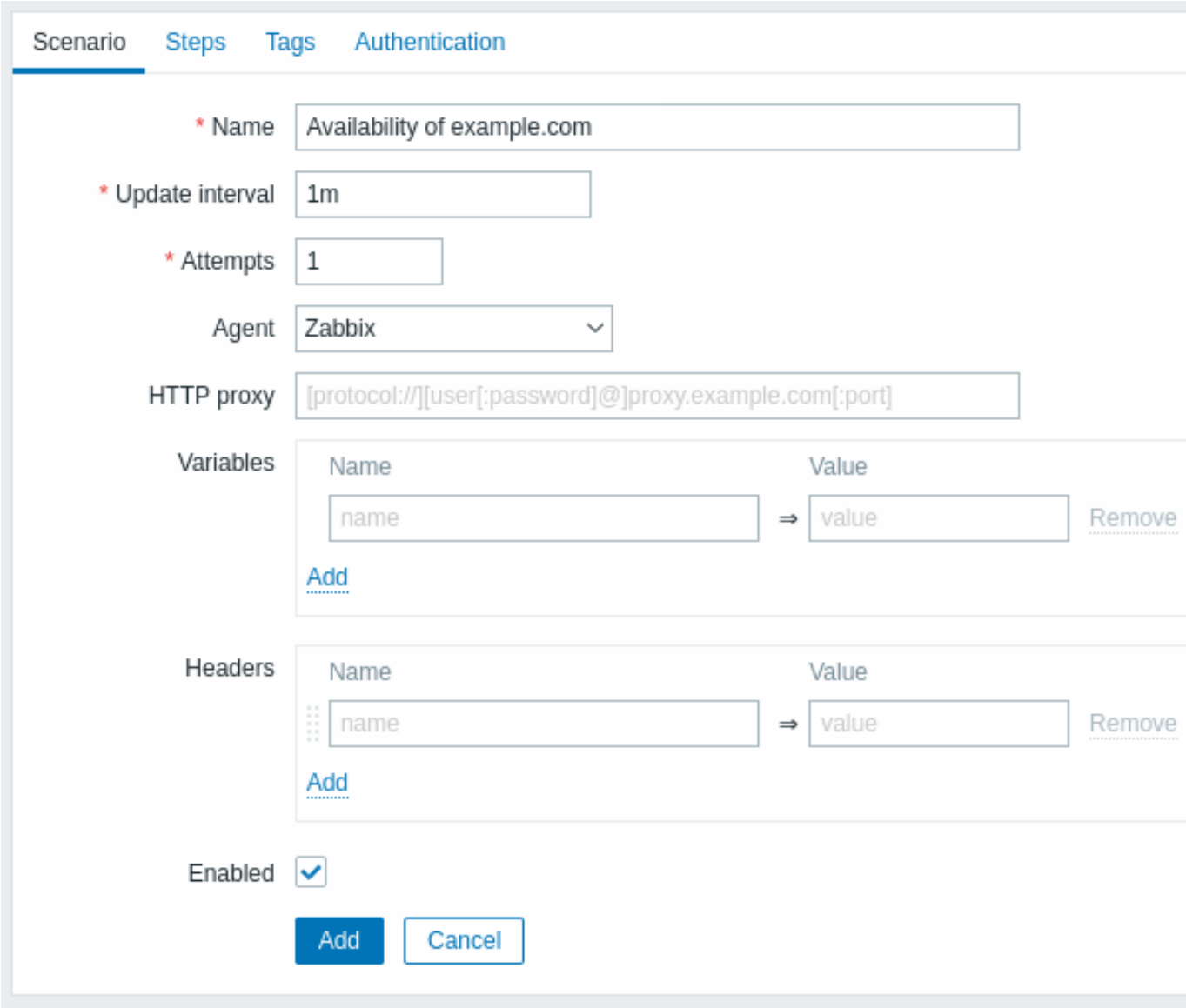
Zabbix tīmekļa uzraudzība atbalsta gan HTTP, gan HTTPS. Palaižot tīmekļa scenāriju, Zabbix pēc izvēles sekos pāradresācijām (skatiet tālāk opciju *Sekot pāradresācijām*). Maksimālais pāradresāciju skaits ir iebūvēts un ir 10 (izmantojot cURL opciju CURLOPT_MAXREDIRS). Visas sīkdatnes tiek saglabātas viena scenārija izpildes laikā.

Tīmekļa scenārija konfigurēšana Lai konfigurētu tīmekļa scenāriju:

1. Dodieties uz: *Data collection* > *Hosts* (vai *Templates*).

2. Hosts/veidnes rindā noklikšķiniet uz *Web*.
3. Labajā pusē noklikšķiniet uz *Create web scenario* (vai uz scenārija nosaukuma, lai rediģētu esošu scenāriju).
4. Veidlapā ievadiet scenārija parametrus.

Cilnē **Scenario** var konfigurēt tīmekļa scenārija vispārīgos parametrus.



Visi obligātie ievades lauki ir atzīmēti ar sarkanu zvaigznīti.

Scenārija parametri:

Parametrs	Apraksts
Name	Unikāls scenārija nosaukums. Tiek atbalstīti lietotāja makro. *Ņemiet vērā*, ka, ja tiek izmantoti lietotāja makro, šie makro tīmekļa uzraudzības vienumu nosaukumos paliks neatrisināti.
Update interval	Cik bieži scenārijs tiks izpildīts. Tiek atbalstīti laika sufiksi, piemēram, 30s, 1m, 2h, 1d. Tiek atbalstīti lietotāja makro. *Ņemiet vērā*, ka, ja tiek izmantots lietotāja makro un tā vērtība tiek mainīta (piemēram, 5m → 30s), nākamā pārbaude tiks veikta atbilstoši iepriekšējai vērtībai (piemērā ar šīm vērtībām — vēlāk). Jauni tīmekļa scenāriji tiks pārbaudīti 60 sekunžu laikā pēc to izveides.
Attempts	Mēģinājumu skaits tīmekļa scenārija darbību izpildei. Tīkla problēmu gadījumā (noildze, savienojuma trūkums u. c.) Zabbix var atkārtot darbības izpildi vairākas reizes. Iestatītais skaits vienādi ietekmēs katru scenārija darbību. Var norādīt līdz 10 mēģinājumiem; noklusējuma vērtība ir 1. *Piezīme*: Zabbix neatkārtos darbību nepareiza atbildes koda vai nepieciešamās virknes neatbilstības dēļ.

Parametrs	Apraksts
Agent	Atlasiet klienta aģentu. Zabbix uzdosies par atlasīto pārlūkprogrammu. Tas ir noderīgi, ja tīmekļa vietne dažādām pārlūkprogrammām atgriež atšķirīgu saturu. Šajā laukā var izmantot lietotāja makro.
HTTP proxy	Varat norādīt izmantojamo HTTP starpniekserveri, izmantojot formātu `[protocol://][username[:password]@]proxy.example.com[:port]`. Tādējādi tiek iestatīta cURL opcija 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 ports 1080. Ja starpniekserveris ir norādīts, tas aizstās ar starpniekserveri saistītos vides mainīgos, piemēram, http_proxy un HTTPS_PROXY. Ja starpniekserveris nav norādīts, tas ar starpniekserveri saistītos vides mainīgos neaizstās. Ievadītā vērtība tiek nodota "tāda, kāda tā ir"; tās pareizība netiek pārbaudīta. Varat ievadīt arī SOCKS starpniekservera adresi. Ja norādīsiet nepareizu protokolu, savienojums neizdosies un vienums kļūs neatbalstīts. *Piezīme*: HTTP starpniekserverim tiek atbalstīta tikai vienkārša autentifikācija. Šajā laukā var izmantot lietotāja makro.
Variables	Mainīgie, kurus var izmantot scenārija darbībās (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 makro 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 makro var atsaukties darbībās, piemēram, {username}, {password}, {hostid} u. c. Zabbix tos automātiski aizstās ar faktiskajām vērtībām. Ņemiet vērā, ka mainīgajiem ar `regex:` ir nepieciešama viena darbība, lai iegūtu regulārās izteiksmes vērtību, tāpēc iegūto vērtību var izmantot tikai nākamajā darbībā. Ja vērtības daļa sākas ar `regex:`, daļa pēc tā tiek uzskatīta par regulāro izteiksmi, kas meklē atbilstību tīmekļa lapā un, ja tā tiek atrasta, saglabā atbilstību mainīgajā. Lai varētu iegūt atbilstošo vērtību, jābūt vismaz vienai apakšgrupai. Tiek atbalstīti lietotāja makro un `{HOST.*}` makro. Izmantojot mainīgos vaicājuma laukos vai veidlapas datos POST mainīgajiem, tie tiek automātiski kodēti URL formātā, taču, izmantojot tos neapstrādātā POST pieprasījumā vai tieši URL, tie manuāli jāiekodē URL formātā.
Headers	HTTP galvenes tiek izmantotas pieprasījuma izpildes laikā. Var izmantot noklusējuma un pielāgotās galvenes. Galvenes tiks piešķirtas, izmantojot noklusējuma iestatījumus atkarībā no aģenta tipa, kas scenārija līmenī atlasīts nolaižamajā sarakstā, un tiks lietotas visām darbībām, ja vien tās nav pielāgotas darbības līmenī. **Ņemiet vērā, ka galvenes definēšana darbības līmenī automātiski atmet visas iepriekš definētās galvenes, izņemot noklusējuma galveni, kas piešķirta, scenārija līmenī nolaižamajā sarakstā atlasot "User-Agent".** Tomēr arī noklusējuma galveni "User-Agent" var aizstāt, norādot to darbības līmenī. Lai atceltu galveni scenārija līmenī, darbības līmenī galvene jānosauc un jānorāda bez vērtības. Galvenes jānorāda, izmantojot tādu pašu sintaksi, kāda tiek izmantota HTTP protokolā, pēc izvēles izmantojot dažas papildu funkcijas, 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 makro un `{HOST.*}` makro.

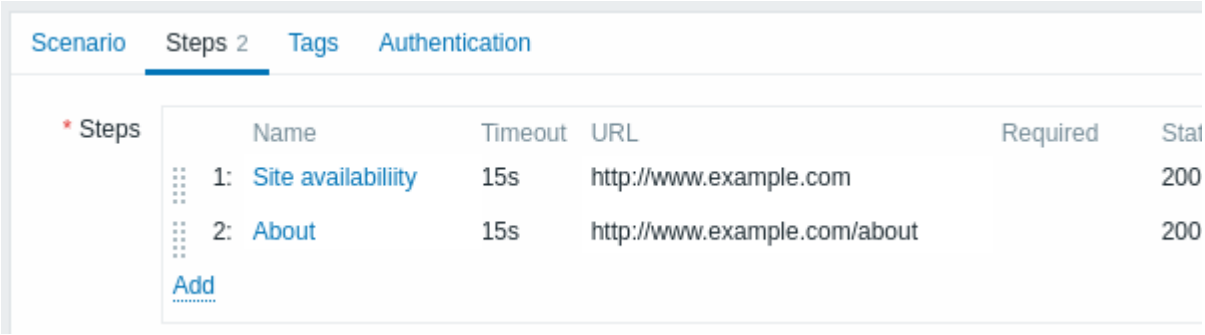
Parametrs	Apraksts
Enabled	Ja šī izvēles rūtiņa ir atzīmēta, scenārijs ir aktīvs; pretējā gadījumā tas ir atspējots.

Ņemiet vērā, ka, rediģējot esošu scenāriju, veidlapā ir pieejamas vēl divas pogas:

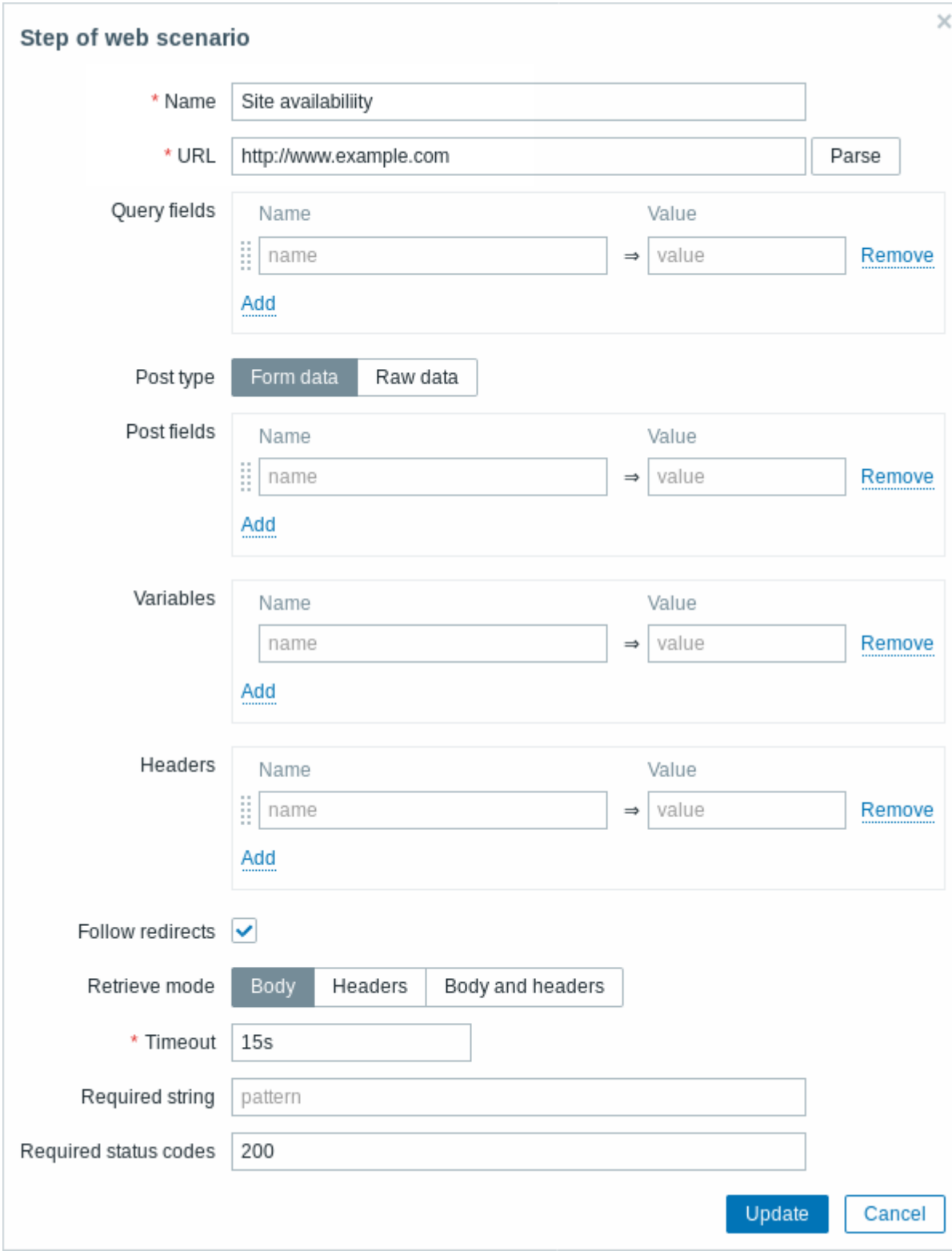
Clone	Izveidot citu scenāriju, pamatojoties uz esošā scenārija parametriem.
Clear history and trends	Dzēst scenārija vēstures un tendenču datus. Pēc datu dzēšanas serveris scenāriju izpildīs nekavējoties.

Note:
Ja lauks *HTTP proxy* ir atstāts tukšs, cits 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`.
Plašāku informāciju var iegūt, izpildot čaulas komandu: `# man curl`.

Cilnē **Steps** var konfigurēt tīmekļa scenārija darbības. Lai pievienotu tīmekļa scenārija darbību, blokā *Steps* noklikšķiniet uz *Add*.



Note:
Slepenus lietotāja makro nedrīkst izmantot URL, jo tie tiks aizstāti ar "******".



Darbību konfigurēšana

Darbības parametri:

Parametrs	Apraksts
Nosaukums	Unikāls darbības nosaukums. Tiek atbalstīti lietotāja makro. *Ņemiet vērā*, ka, ja tiek izmantoti lietotāja makro, šie makro tīmekļa uzraudzības vienumu nosaukumos paliks neatrisināti.

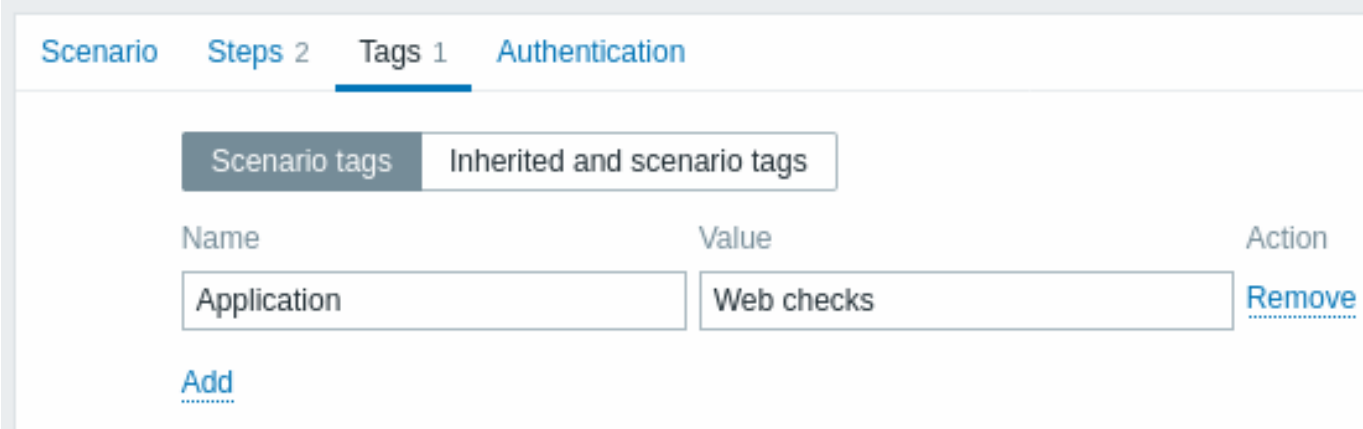
Parametrs	Apraksts
URL	URL, ar kuru izveidot savienojumu un no kura izgūt datus. Piemēram: https://www.example.com http://www.example.com/download Domēna nosaukumus var norādīt Unicode rakstzīmēs. Izpildot tīmekļa scenārija darbību, tie tiek automātiski pārveidoti par ASCII, izmantojot punycode. Pogu *Parsēt* var izmantot, lai no URL atdalītu neobligātos vaicājuma laukus (piemēram, `?key=value&limit=10`), pārvietojot atribūtus un vērtības uz *Vaicājuma laukiem*, lai tos automātiski kodētu URL formātā. URL var izmantot mainīgos, izmantojot sintaksi {macro}. Mainīgos var manuāli kodēt URL formātā, izmantojot sintaksi {{macro}.urlencode()}. Tiek atbalstīti lietotāja makro un `{HOST.*}` makro. Drošības apsvērumu dēļ URL neiekļaujiet faktiskos lietotājvārdus vai paroles. Tā vietā izmantojiet lietotāja makro, lai nepieļautu akreditācijas datu atklāšanu. Ierobežots līdz 2048 rakstzīmēm.
Vaicājuma lauki	HTTP GET mainīgie URL. Tie tiek norādīti kā atribūtu un vērtību pāri. Vērtības tiek automātiski kodētas URL formātā. Tīmekļa scenārija mainīgo, lietotāja makro vai `{HOST.*}` makro vērtības tiek atrisinātas un pēc tam automātiski kodētas URL formātā. Izmantojot sintaksi {{macro}.urlencode()}, tās tiks kodētas URL formātā divreiz. Tiek atbalstīti lietotāja makro un `{HOST.*}` makro.
Post	HTTP POST mainīgie. Režīmā **Formas dati** tie tiek norādīti kā atribūtu un vērtību pāri. Vērtības tiek automātiski kodētas URL formātā. Tīmekļa scenārija mainīgo, lietotāja makro vai `{HOST.*}` makro vērtības tiek atrisinātas un pēc tam automātiski kodētas URL formātā. Režīmā **Neapstrādāti dati** atribūti/vērtības tiek parādītas vienā rindā un savienotas ar simbolu **&**. Neapstrādātas vērtības var manuāli kodēt/dekodēt URL formātā, izmantojot sintaksi {{macro}.urlencode()} vai {{macro}.urldecode()}. Piemēram: id=2345&userid={user} Ja {user} ir definēts kā tīmekļa scenārija mainīgais, darbības izpildes laikā tas tiks aizstāts ar tā vērtību. Ja vēlaties mainīgo kodēt URL formātā, aizstājiet {user} ar {{user}.urlencode()}. Tiek atbalstīti lietotāja makro un `{HOST.*}` makro.
Mainīgie	Darbības līmeņa mainīgie, kurus var izmantot GET un POST funkcijās. Tie tiek norādīti kā atribūtu un vērtību pāri. Darbības līmeņa mainīgie aizstāj scenārija līmeņa mainīgos vai iepriekšējās darbības mainīgos. Tomēr darbības līmeņa mainīgā vērtība ietekmē tikai nākamo darbību, nevis pašreizējo darbību. Tiem ir šāds formāts: **{macro}**=value **{macro}**=regex:<regular expression> Papildinformāciju skatiet mainīgo aprakstā scenārija līmenī. Izmantojot mainīgos vaicājuma laukos vai formas datos POST mainīgajiem, tie tiek automātiski kodēti URL formātā, bet, izmantojot tos neapstrādātā POST pieprasījumā vai tieši URL, tie ir manuāli jākodē URL formātā.
Galvenes	Pielāgotas HTTP galvenes, kas tiks nosūtītas pieprasījuma izpildes laikā. Tās tiek norādītas kā atribūtu un vērtību pāri. Darbības līmenī definēta galvene tiks izmantota konkrētajai darbībai. **Ņemiet vērā, ka galvenes definēšana darbības līmenī automātiski atmet visas iepriekš definētās galvenes, izņemot noklusējuma galveni, kas piešķirta, scenārija līmenī nolaižamajā sarakstā atlasot "User-Agent".** Tomēr pat noklusējuma galveni "User-Agent" var aizstāt, norādot to darbības līmenī. Piemēram, piešķirot galvenei nosaukumu, bet nenorādot vērtību, tiks noņemta noklusējuma galvene scenārija līmenī. Tiek atbalstīti lietotāja makro un `{HOST.*}` makro. Tādējādi tiek iestatīta cURL opcija `CURLOPT_HTTPHEADER`.
Sekot novirzēm	Atzīmējiet izvēles rūtiņu, lai sekotu HTTP novirzēm. Tādējādi tiek iestatīta cURL opcija CURLOPT_FOLLOWLOCATION.
Izgūšanas režīms	Atlasiet izgūšanas režīmu: **Pamatteksts** - izgūt tikai HTTP atbildes pamattekstu **Galvenes** - izgūt tikai HTTP atbildes galvenes **Pamatteksts un galvenes** - izgūt HTTP atbildes pamattekstu un galvenes

Parametrs	Apraksts
Noildze	Zabbix URL apstrādei nepatērēs 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. Tāpēc Zabbix darbībai nepatērēs vairāk par **2 x Timeout** sekundēm. Tiek atbalstīti laika sufiksi, piemēram, 30s, 1m, 1h. Tiek atbalstīti lietotāja makro.
Nepieciešamā virkne	Nepieciešamais regulārās izteiksmes paraugs. Ja izgūtais saturs (HTML) neatbilst nepieciešamajam paraugam, darbība neizdosies. Ja lauks ir tukšs, nepieciešamās 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 makro un `{HOST.*}` makro.
Nepieciešamie statusa kodi	Sagaidāmo HTTP statusa kodu saraksts. Ja Zabbix saņem kodu, kas nav sarakstā, darbība neizdosies. Ja lauks ir tukšs, statusa kodu pārbaude netiek veikta. Piemēram: 200,201,210-299 Tiek atbalstīti lietotāja makro.

Note:
Jebkādas izmaiņas tīmekļa scenārija darbībās tiks saglabātas tikai tad, kad tiks saglabāts viss scenārijs.

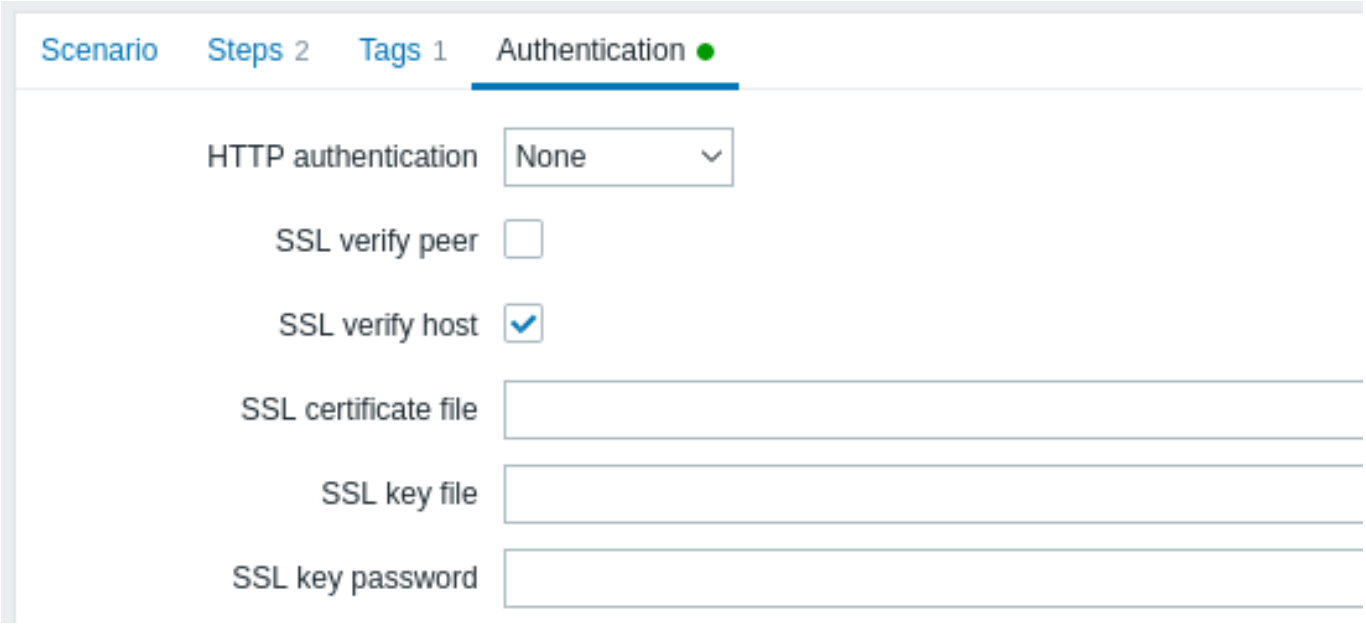
Skatiet arī praktisku piemēru, kurā parādīts, kā konfigurēt tīmekļa uzraudzības darbības.

Tagu konfigurēšana Cilne **Tagi** ļauj definēt scenārija līmeņa tagus.



Tagu 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:

Parameter	Description
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.
User	Ievadiet lietotājvārdu (līdz 255 rakstzīmēm). Šis lauks ir pieejams, ja *HTTP authentication* ir iestatīts uz Basic, NTLM, Kerberos vai Digest. Tiek atbalstīti lietotāja makro.
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 makro.
SSL verify peer	Atzīmējiet izvēles rūtiņu, lai pārbaudītu tīmekļa servera SSL sertifikātu. Servera sertifikāts tiek pārbaudīts, izmantojot sistēmas mēroga CA atrašanās vietā esošos uzticamo sertificēšanas iestāžu (CA) sertifikātus. CA failu atrašanās vietu var mainīt, izmantojot Zabbix servera vai starpniekservera konfigurācijas parametru SSLCALocation. Tādējādi tiek iestatīta cURL opcija CURLOPT_SSL_VERIFYPEER.
SSL verify host	Atzīmējiet izvēles rūtiņu, lai pārbaudītu, vai tīmekļa servera sertifikāta lauks *Common Name* vai *Subject Alternate Name* atbilst. Tādējādi tiek iestatīta cURL opcija CURLOPT_SSL_VERIFYHOST.
SSL certificate file	Klienta autentifikācijai izmantotā SSL sertifikāta faila nosaukums. Sertifikāta failam jābūt PEM[1] formātā. Ja sertifikāta failā ir arī privātā atslēga, atstājiet lauku *SSL key file* tukšu. Ja atslēga ir šifrēta, laukā *SSL key password* norādiet paroli. Direktoriju, kurā atrodas šis fails, nosaka Zabbix servera vai starpniekservera konfigurācijas parametrs SSLCertLocation. Šajā laukā var izmantot `HOST.*` makro un lietotāja makro. Tādējādi tiek iestatīta cURL opcija CURLOPT_SSLCERT.
SSL key file	Klienta autentifikācijai izmantotā SSL privātās atslēgas faila nosaukums. Privātās atslēgas failam jābūt PEM[1] formātā. Direktoriju, kurā atrodas šis fails, nosaka Zabbix servera vai starpniekservera konfigurācijas parametrs SSLKeyLocation. Šajā laukā var izmantot `HOST.*` makro un lietotāja makro. Tādējādi tiek iestatīta cURL opcija CURLOPT_SSLKEY.
SSL key password	SSL privātās atslēgas faila parole. Šajā laukā var izmantot lietotāja makro. Tādējādi tiek iestatīta cURL opcija CURLOPT_KEYPASSWD.

Attention:
[1] Zabbix atbalsta tikai PEM formāta sertifikātu un privāto atslēgu failus. Ja sertifikāta un privātās atslēgas dati ir PKCS #12 formāta failā (parasti ar paplašinājumu *.p12 vai *.pfx), PEM failu no tā var ģenerēt, 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
```

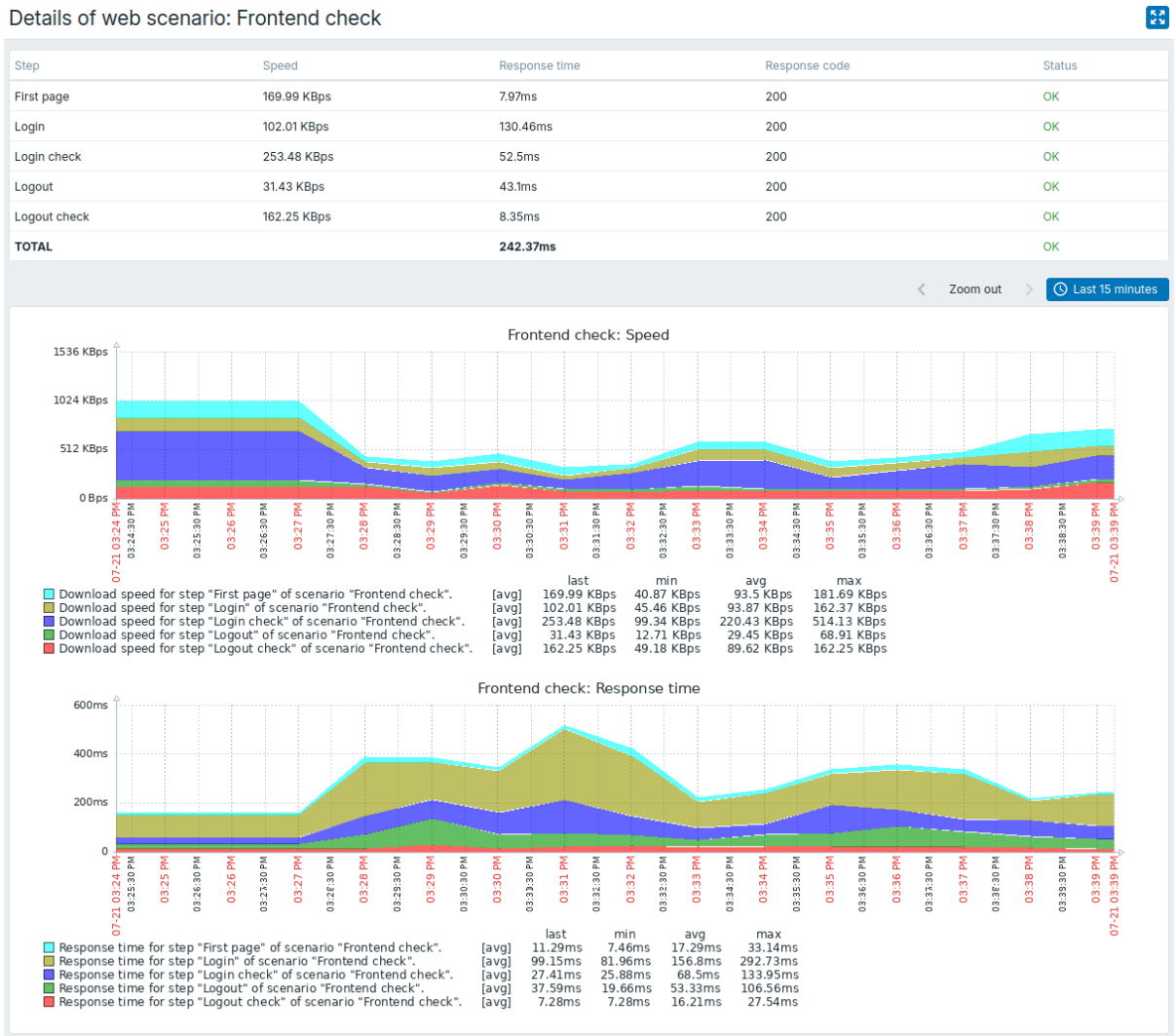
Note:
Zabbix serveris uztver sertifikātu izmaiņas bez restartēšanas.

Note:
Ja klienta sertifikāts un privātā atslēga atrodas vienā failā, vienkārši norādiet to laukā "SSL certificate file" un atstājiet lauku "SSL key file" tukšu. Sertifikātam un atslēgai joprojām jābūt PEM formātā. Sertifikātu un atslēgu var viegli apvienot:

```
cat client.crt client.key > client.pem
```

Attēlošana Lai skatītu hostam konfigurētos tīmekļa scenārijus:

1. Dodieties uz *Monitoring* > *Hosts*.
2. Atrodiet sarakstā hostu un pēdējā kolonnā noklikšķiniet uz hipersaites *Web*.
3. Noklikšķiniet uz scenārija nosaukuma, lai iegūtu detalizētu informāciju.



Tīmekļa scenāriju pārskatu var attēlot arī sadaļā *Dashboards*, izmantojot tīmekļa uzraudzības logrīku.

Jaunākie tīmekļa scenārija izpildes rezultāti ir pieejami sadaļā *Monitoring* > *Latest data*.

Paplašināta uzraudzība Dažkārt ir nepieciešams reģistrēt saņemtās HTML lapas saturu.
Tas ir īpaši noderīgi, ja kāds tīmekļa scenārija solis neizdodas.
Šim nolūkam kalpo atkļūdošanas līmenis 5 (trace).
Šo līmeni var iestatīt serveris un starpniekserveris konfigurācijas failos vai izmantojot izpildlaika vadības opciju (`-R log_level_increase="`

poller,N", kur N ir procesa numurs).
Tālāk redzamie piemēri parāda, kā var sākt paplašinātu uzraudzību, ja jau ir iestatīts atkļūdošanas līmenis 4:

```
# Palielināt visu http poller žurnāla līmeni:
zabbix_server -R log_level_increase="http poller"

# Palielināt otrā http poller žurnāla līmeni:
zabbix_server -R log_level_increase="http poller,2"
```

Ja paplašināta tīmekļa uzraudzība nav nepieciešama, to var apturēt, izmantojot `-R log_level_decrease` opciju.

Tīmekļa monitoringa 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`

Reālās dzīves scenārijs

Pārskats

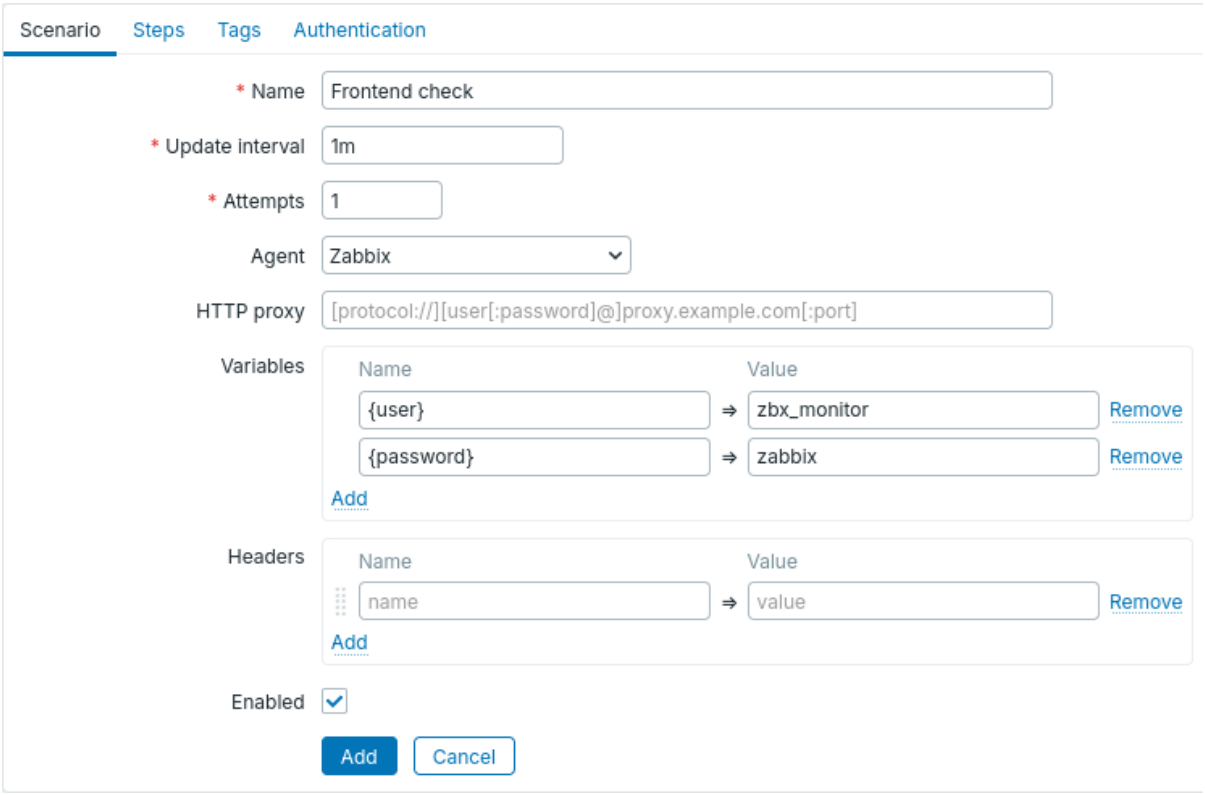
Šajā sadaļā ir sniegts soli pa solim reālas dzīves piemērs, kā var izmantot tīmekļa uzraudzību.

Zabbix tīmekļa uzraudzība tiks izmantota, lai uzraudzītu Zabbix lietotāja saskarni. Mērķis ir noteikt, vai tā ir pieejama, vai tā 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, pieteikšanās veiksmīguma pārbaude, atteikšanās un atteikšanās apstiprināšana.

Scenārijs

Pievienot jaunu tīmekļa scenāriju

1. Dodieties uz *Datu vākšana* > *Hosts*.
2. Izvēlieties hostu un hosta rindā noklikšķiniet uz *Web*.
3. Noklikšķiniet uz *Izveidot tīmekļa scenāriju*.



4. Jaunā scenārija formā cilnē *Scenārijs* aizpildiet šādus laukus:

    - *Nosaukums*: Frontend pārbaude.
    - *Atjaunināšanas intervāls*: 1m.
    - *Mēģinājumi*: 1.
    - *Aģents*: Zabbix.
 Cilnē *Mainīgie* pievienojiet šādus vārda-vērtības pārus:
    - `{user}`: jūsu esošie Zabbix lietotāja akreditācijas dati;
    - `{password}`: jūsu esošie Zabbix lietotāja akreditācijas dati.

Note:
Drošības apsvērumu dēļ ieteicams izveidot atsevišķu lietotāju ar minimālām atļaujām, ko izmantot uzraudzības nolūkiem.

Pēc izvēles pārslēdzieties uz cilni *Tagi* un pievienojiet tīmekļa scenārija tagus. Tīmekļa scenārija tagus var izmantot, lai ātri identificētu saistītos vienumus un trigerus vai meklētu savāktajos datos. Piemēram, šai pamācībai piemēroti tagi ir `component: web-scenario` un/vai `target: lietotāja saskarne`.

Konfigurējiet tīmekļa scenārija soļus

Pārslēdzieties uz cilni *Steps* un definējiet scenārija soļus.
 Pēc katra soļa konfigurēšanas noklikšķiniet uz pogas *Add*.

Attention:
Ņemiet vērā, ka Zabbix lietotāja saskarne, piesakoties sistēmā, izmanto JavaScript pāradresāciju, tāpēc vispirms jānotiek pieteikšanās procesam, un pieteikušās sesijas funkcijas var pārbaudīt tikai turpmākajos soļos. Turklāt pieteikšanās solī jāizmanto pilns URL uz **index.php** failu.

Kopīgie lauki visām darbībām

Papildus darbībai specifiskajiem laukiem aizpildiet tālāk norādītos kopīgos laukus katrai tālāk aprakstītajai darbībai:

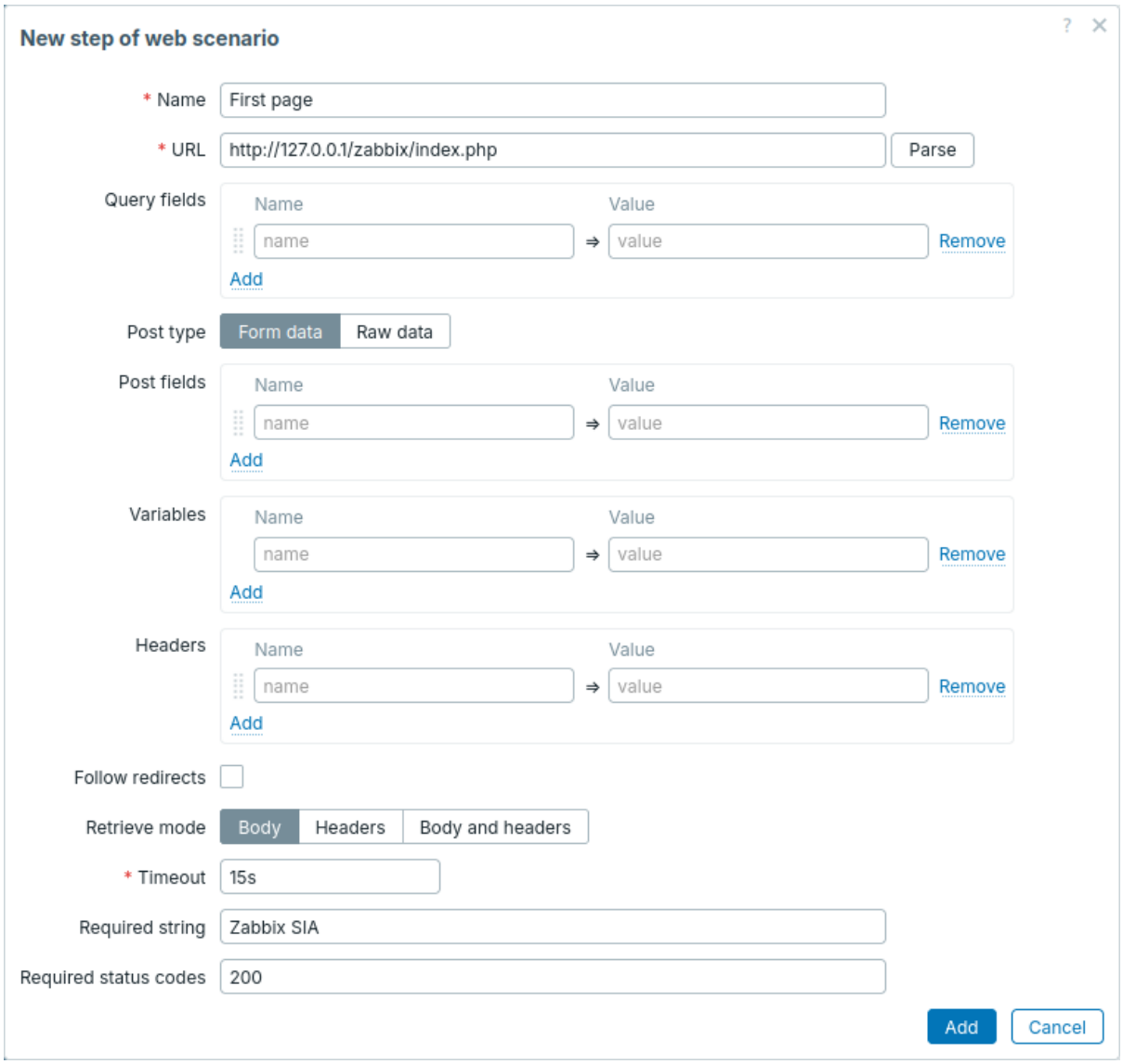
- *URL*: Zabbix lietotāja saskarnes URL (`http://<server_ip_or_name>/zabbix/index.php` (Apache gadījumā) vai `http://<server_ip_or_name>/index.php` (NGINX gadījumā)). Tīmekļa scenārija 3. darbībai izmantojiet citu URL, kā aprakstīts tālāk.
- *Timeout*: 15s.
- *Required status codes*: 200.

Tīmekļa scenārija 1. solis

Pārbaudiet, vai pirmā lapa atbild pareizi, atgriež HTTP statusa kodu 200 un satur tekstu *Zabbix SIA*.
 Aizpildiet šādus laukus:

- *Name*: Pirmā lapa.
- *Timeout*: 15s.

- *Required string*: Zabbix SIA.
- kopīgie lauki visiem soļiem.

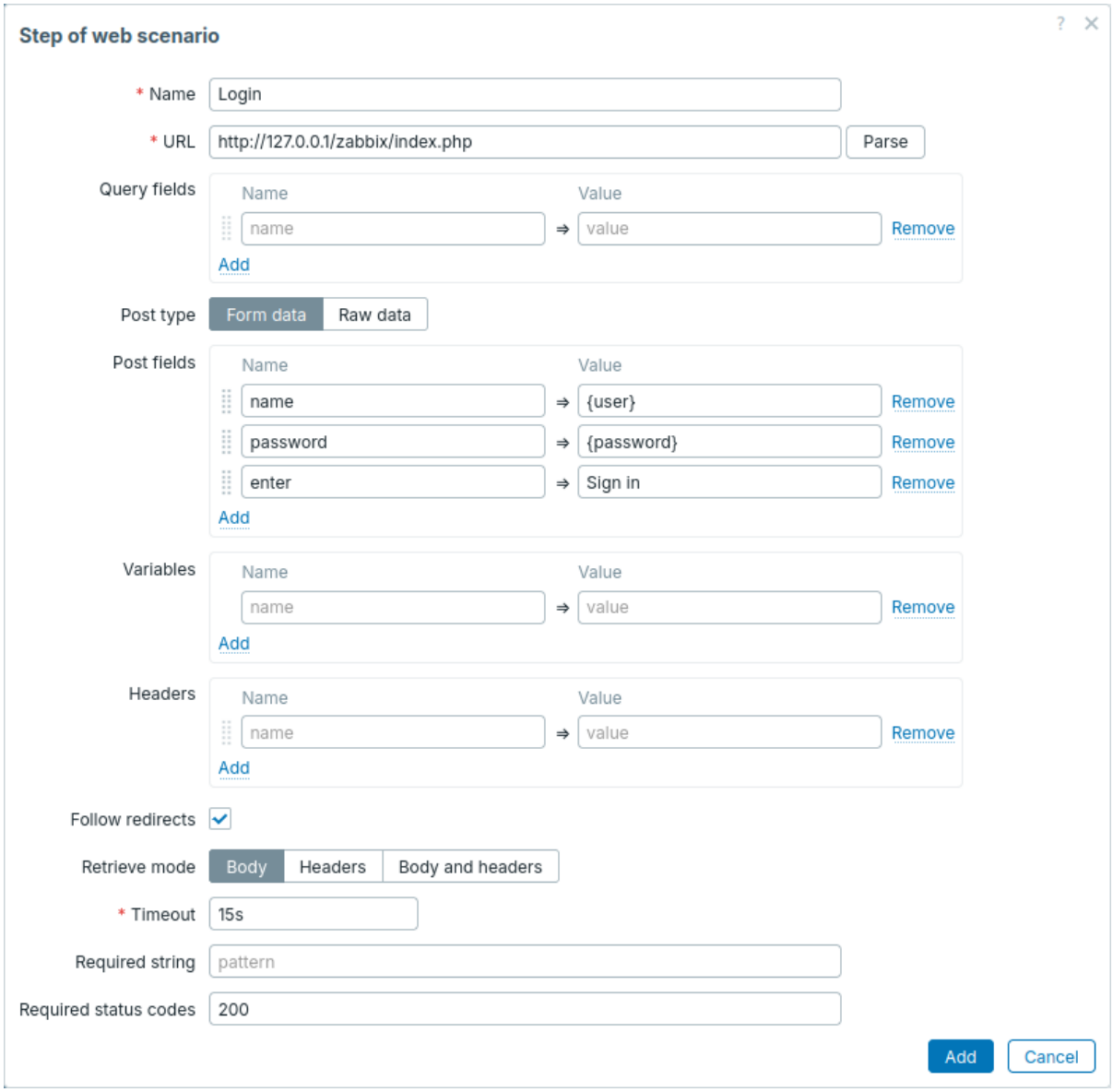


Tīmekļa scenārija 2. solis

Piesakieties Zabbix lietotāja saskarnē, izmantojot makrosus (mainīgos), kas definēti cilnē *Scenario*: *{user}* un *{password}*.
Aizpildiet šādus laukus:

- *Name*: Login.
- Sadaļā *Post fields* pievienojiet šādus nosaukuma un vērtības pārus:
  - name: `{user}`;
  - password: `{password}`;
  - enter: Sign in.
- Visi kopīgie lauki visiem soļiem.

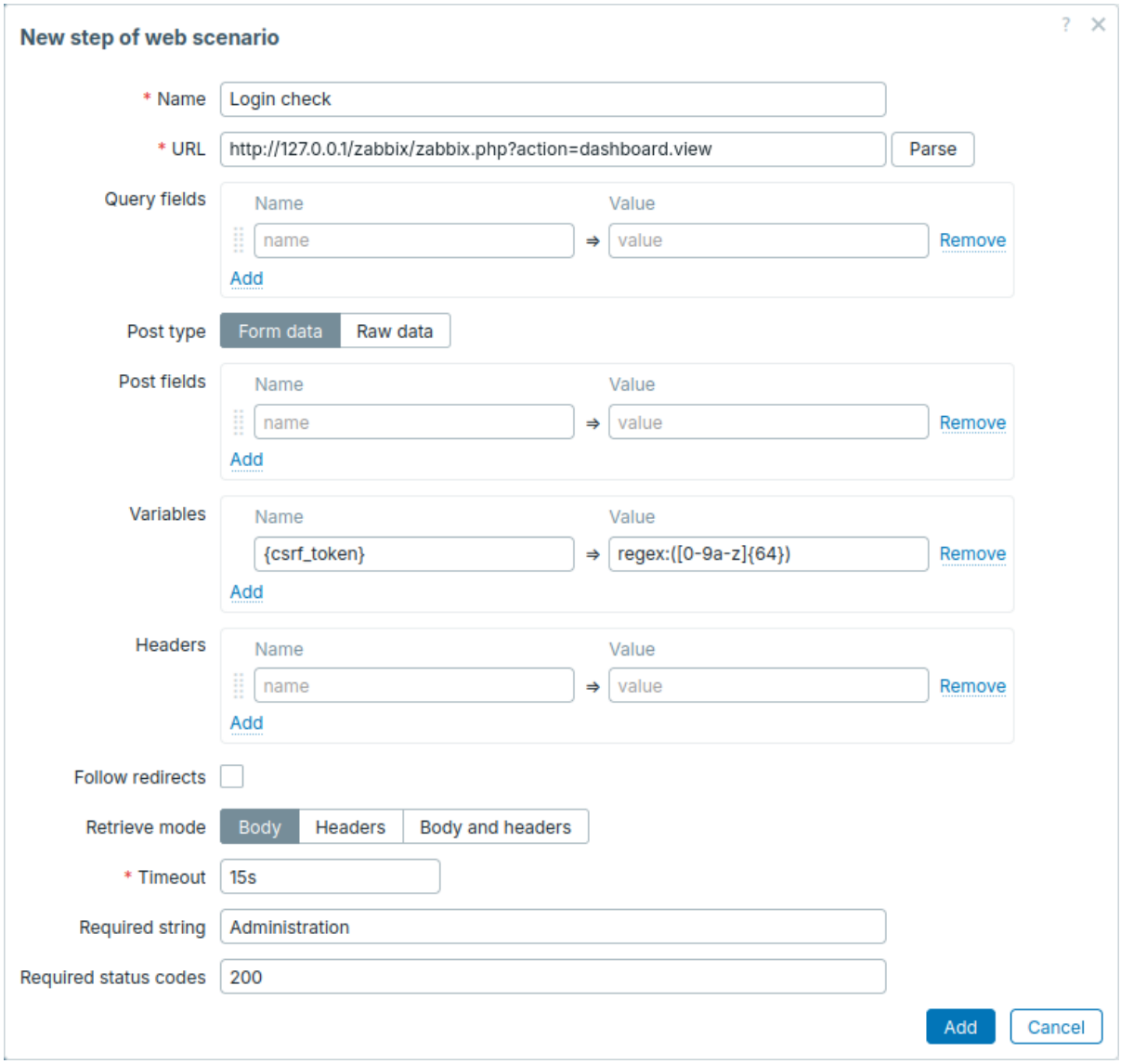
Atzīmējiet izvēles rūtiņu *Follow redirects*.



Tīmekļa scenārija 3. solis

Pēc pieteikšanās pārbaudiet, vai darbība ir veiksmīga, meklējot virkni, kas ir redzama tikai tad, kad esat pieteicies, piemēram, *Administration*.
 Aizpildiet šādus laukus:

- *Name*: Login check.
- *URL*: URL, uz kuru lietotājs tiek novirzīts pēc pieteikšanās (piemēram, http://127.0.0.1/zabbix/zabbix.php?action=dashboard.view).
- Sadaļā *Variables* pievienojiet vārda-vērtības pāri `{csrf_token}`: `regex:([0-9a-z]{64})`. Šis mainīgais uztvers piešķirtā CSRF marķiera vērtību, lai to atkārtoti izmantotu tīmekļa scenārija 4. solī.
- *Required string*: Administration.
- Vispārīgie lauki visiem soļiem.



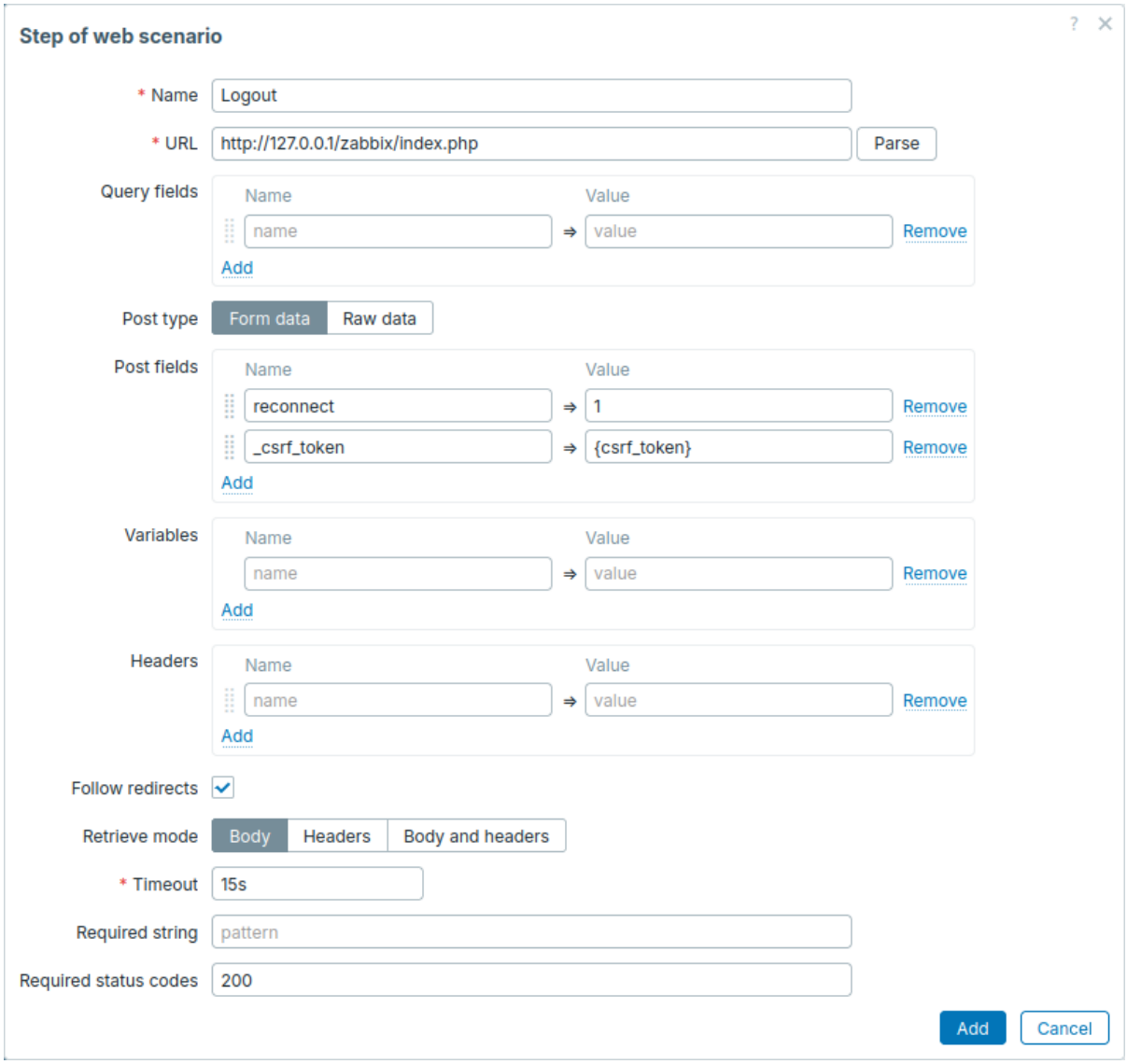
Tīmekļa scenārija 4. solis

Kad lietotāja saskarnes pieejamība un pieteikšanās ir pārbaudīta, pievienojiet atteikšanās soli; pretējā gadījumā Zabbix datubāze tiks piepildīta ar daudziem atvērtiem sesiju ierakstiem.
 Aizpildiet šādus laukus:

- *Name*: Logout.
- Sadaļā *Post fields* pievienojiet šādus nosaukuma un vērtības pārus:
  - reconnect: 1;
  - _csrf_token: `{csrf_token}`.
- Kopīgie lauki visiem soļiem.

Atzīmējiet izvēles rūtiņu *Follow redirects*.

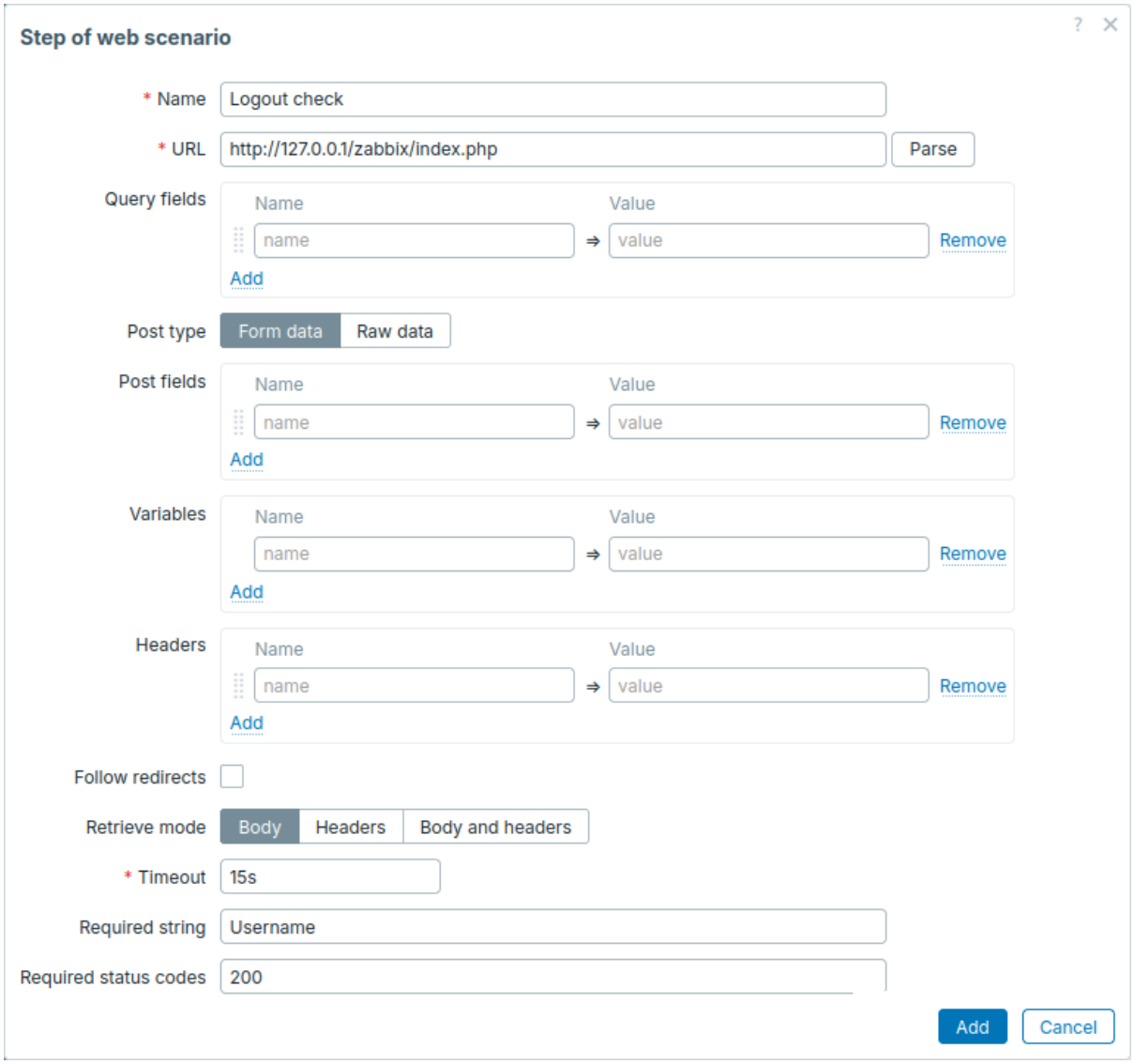
Šajā solī tiek izmantots mainīgais `{csrf_token}`, kas iegūts tīmekļa scenārija 3. solī.



Tīmekļa scenārija 5. solis

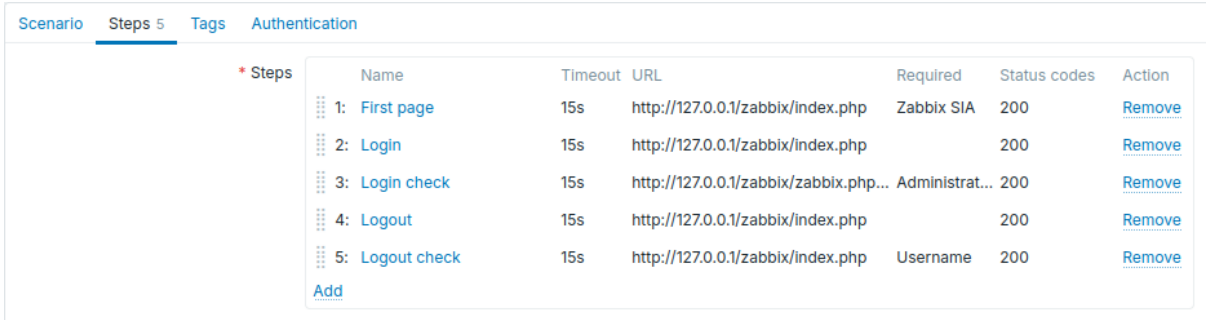
Lai apstiprinātu atteikšanos no sistēmas, pārbaudiet *Username* virkni.
 Aizpildiet šādus laukus:

- *Name*: Logout check.
- *Required string*: Username.
- Kopīgie lauki visiem soļiem.



Pilna soļu konfigurācija

Pilnīgai tīmekļa scenārija soļu konfigurācijai vajadzētu izskatīties šādi:

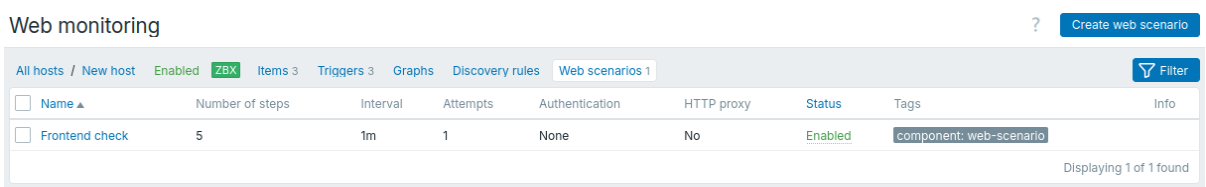


Pārbaudiet rezultātus

Saglabājiet pabeigto tīmekļa uzraudzības scenāriju.

Kad tīmekļa scenārijs ir izveidots, uzraudzībai automātiski tiek pievienoti jauni tīmekļa uzraudzības vienumi.

Scenārijs tiks pievienots hostam. Lai skatītu informāciju par tīmekļa scenāriju, dodieties uz *Monitoring* > *Hosts*, atrodiet hostu sarakstā un noklikšķiniet uz *Web* hipersaites pēdējā kolonnā.



Noklikšķiniet uz scenārija nosaukuma, lai skatītu detalizētu statistiku:



Pakalpojumu uzraudzība

Pārskats Pakalpojumu uzraudzība ir uzņēmuma līmeņa uzraudzība, ko var izmantot, lai iegūtu pārskatu par visu IT infrastruktūras pakalpojumu koku, identificētu vājos infrastruktūras posmus, aprēķinātu dažādu IT pakalpojumu SLA un pārbaudītu citu informāciju augstākā līmenī. Pakalpojumu uzraudzība koncentrējas uz pakalpojuma vispārējo pieejamību, nevis zema līmeņa informāciju, piemēram, diska vietas trūkumu, lielu procesora noslodzi u. c. Pakalpojumu uzraudzība nodrošina arī iespēju atrast problēmas pamatcēloni, ja pakalpojums nedarbojas, kā paredzēts.

Pakalpojumu uzraudzība nodrošina iespēju 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 nodots augstākiem līmeņiem atbilstoši atlasītajam algoritmam. Atsevišķu mezglu statusu ietekmē kartēto problēmu statuss. Problēmu kartēšana tiek veikta, izmantojot tagus.

Zabbix var nosūtīt paziņojumus vai automātiski izpildīt skriptu Zabbix serverī, ja tiek konstatētas pakalpojuma stāvokļa izmaiņas. Ir iespējams definēt elastīgus noteikumus, kas nosaka, vai vecākpakalpojumam jānonāk stāvoklī "Problēma", pamatojoties uz pakār-

toto pakalpojumu statusiem. Pēc tam pakalpojumu problēmu datus var izmantot SLA aprēķināšanai un SLA pārskatu nosūtīšanai, pamatojoties uz elastīgu nosacījumu kopu.

Pakalpojumu uzraudzība tiek konfigurēta izvēlnē Pakalpojumi, kas sastāv no šādām sadaļām:

- Pakalpojumi

Sadaļa Pakalpojumi ļauj izveidot uzraudzītās infrastruktūras hierarhiju, pievienojot vecākpakalpojumus un pēc tam tiem pievienojot pakārtotos pakalpojumus.

Papildus pakalpojumu koka konfigurēšanai šī sadaļa nodrošina visas infrastruktūras pārskatu un ļauj ātri identificēt problēmas, kuru dēļ ir mainījies pakalpojuma stāvoklis.

- SLA

Šajā sadaļā varat definēt pakalpojumu līmeņa vienošanās un iestatīt pakalpojumu līmeņa mērķus konkrētiem pakalpojumiem.

- SLA pārskats

Šajā sadaļā varat skatīt SLA pārskatus.

Pakalpojumu darbības

Varat arī konfigurēt pakalpojumu darbības.

Pakalpojumu darbības nav obligātas un ļauj:

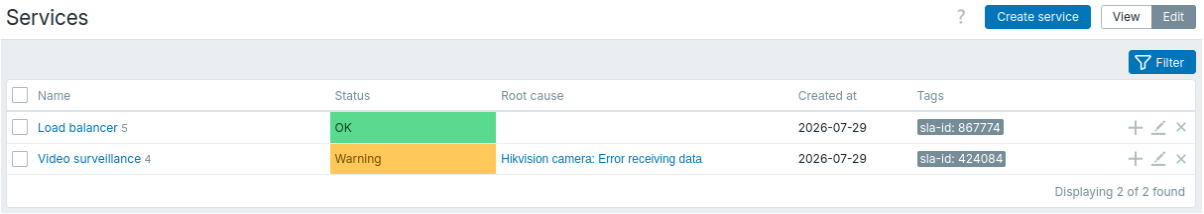
- nosūtīt paziņojumu par pakalpojuma nepieejamību
- izpildīt attālo komandu Zabbix serverī pēc pakalpojuma stāvokļa izmaiņām
- nosūtīt atkopšanas paziņojumu, kad pakalpojums atkal ir pieejams.

Skatiet arī:

- SLA uzraudzības konfigurēšanas piemēru
- Piezīmes par pakalpojumu jauninašanu no Zabbix versijām, kas vecākas par 6.0

Pakalpojumu koks

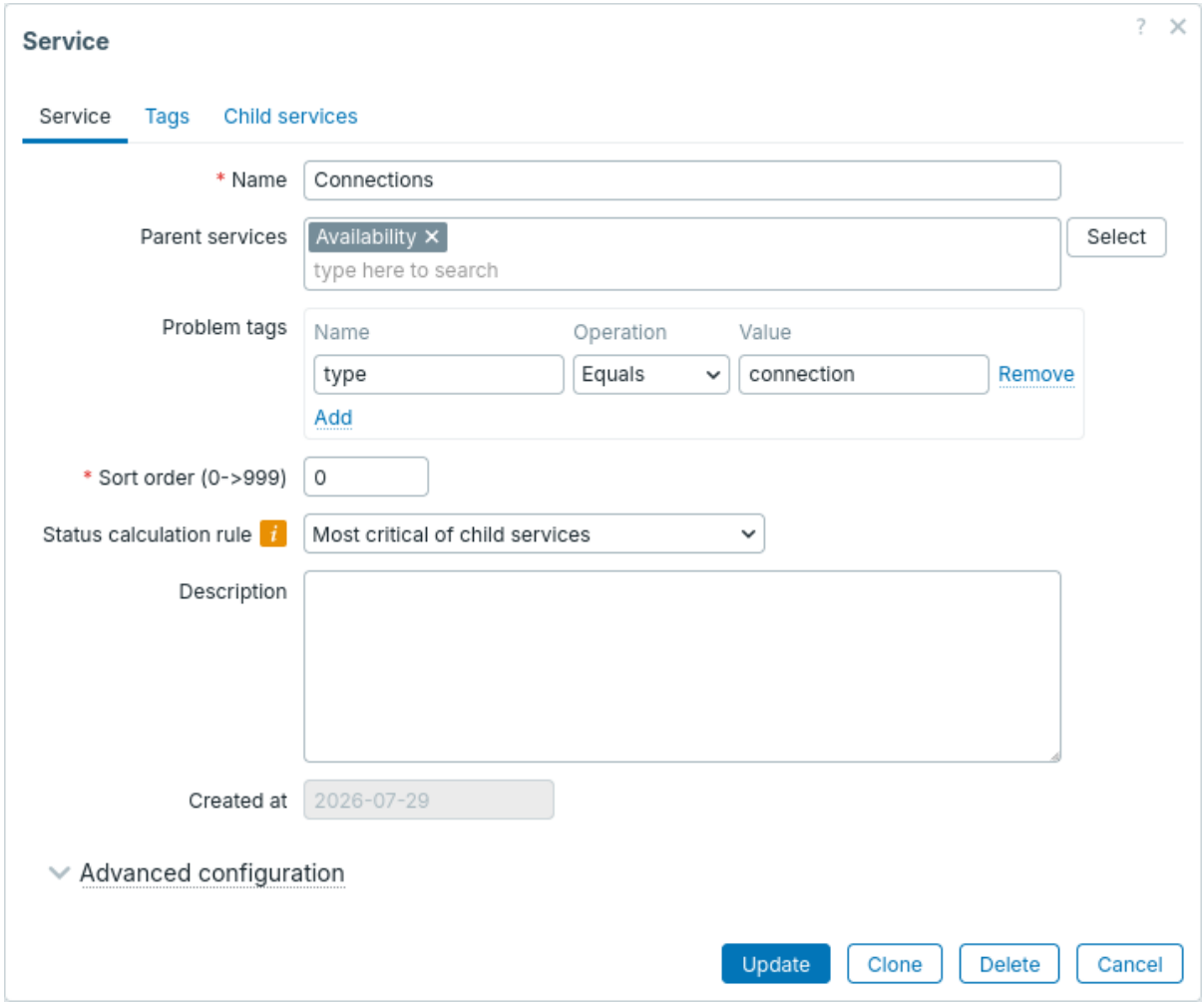
Pakalpojumu koks tiek konfigurēts izvēlnes sadaļā *Services* > *Services*. Augšējā labajā stūrī pārslēdzieties no skatīšanas uz rediģēšanas režīmu.



Lai konfigurētu jaunu pakalpojumu, augšējā labajā stūrī noklikšķiniet uz pogas *Create service*.

Lai ātri pievienotu pakalpojuma apakšpakalpojumu, varat arī noklikšķināt uz plusa ikonas blakus vecākpakalpojumam. Tiks atvērta tā pati pakalpojuma konfigurācijas forma, taču parametrs *Parent services* būs aizpildīts iepriekš.

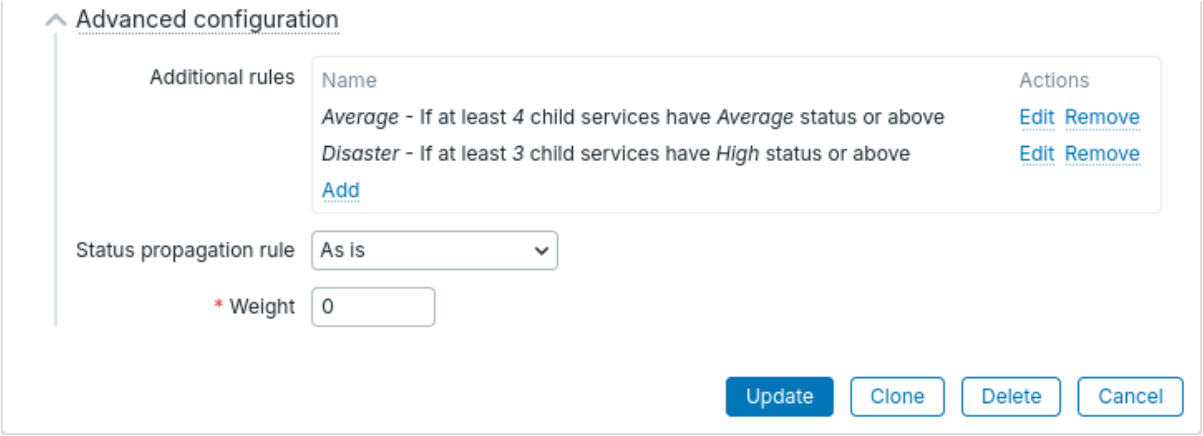
Pakalpojuma konfigurācija Cilnē **Pakalpojums** norādiet nepieciešamos pakalpojuma parametrus:



Visi obligātie ievades lauki ir atzīmēti ar sarkanu zvaigznīti.

Parameter	Description
Name	Pakalpojuma nosaukums.
Parent services	Vecākpakalpojumi, kuriem pakalpojums pieder. Atstājiet šo lauku tukšu, ja pievienojat augstākā līmeņa pakalpojumu. Vienam pakalpojumam var būt vairāki vecākpakalpojumi. Šādā gadījumā tas pakalpojumu kokā tiks parādīts zem katra no vecākpakalpojumiem.
Problem tags	Norādiet tagus, lai problēmu datus piesaistītu pakalpojumam. Var iestatīt vairākus nosacījumus. Tagu nosaukumu salīdzināšana vienmēr ir reģistrjutīga. Katrai nosacījumam ir pieejami divi operatori: **Equals** - iekļaut norādītos tagu nosaukumus un vērtības (reģistrjutīgi) **Contains** - iekļaut norādītos tagu nosaukumus, kuru tagu vērtības satur ievadīto virkni (apakšvirknes atbilstība, reģistrjutīgi)
Sort order	Kārtošanas secība attēlošanai, zemākā tiek rādīta pirmā.
Status calculation rule	Pakalpojuma statusa aprēķināšanas noteikums: **Most critical if all children have problems** - viskritiskākā problēma pakalpojuma apakšpakalpojumos tiek izmantota, lai iekrāsotu pakalpojuma statusu, ja visiem apakšpakalpojumiem ir problēmas **Most critical of child services** - viskritiskākā problēma pakalpojuma apakšpakalpojumos tiek izmantota, lai iekrāsotu pakalpojuma statusu **Set status to OK** - neaprēķināt pakalpojuma statusu Papildu statusa aprēķināšanas noteikumus var konfigurēt papildu konfigurācijas opcijās.
Description	Pakalpojuma apraksts.
Created at	Laiks, kad pakalpojums tika izveidots; tiek parādīts, rediģējot esošu pakalpojumu.
Advanced configuration	Noklikšķiniet uz virsraksta *Advanced configuration*, lai parādītu papildu konfigurācijas opcijas.

Papildu 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* gadījumā, ja nosacījums atbilst.
Nosacījums	Atlasiet nosacījumu tiešajiem pakalpojumiem-pēctečiem: **ja vismaz (N) pakalpojumiem-pēctečiem ir (Status) statuss vai augstāks** **ja vismaz (N%) pakalpojumu-pēcteču ir (Status) statuss vai augstāks** **ja mazāk nekā (N) pakalpojumiem-pēctečiem ir (Status) statuss vai zemāks** **ja mazāk nekā (N%) pakalpojumu-pēcteču ir (Status) statuss vai zemāks** **ja pakalpojumu-pēcteču ar (Status) statusu vai augstāku svars ir vismaz (W)** **ja pakalpojumu-pēcteču ar (Status) statusu vai augstāku svars ir vismaz (N%)** **ja pakalpojumu-pēcteču ar (Status) statusu vai zemāku svars ir mazāks par (W)** **ja pakalpojumu-pēcteču ar (Status) statusu vai zemāku svars ir mazāks par (N%)** Ja ir norādīti vairāki nosacījumi un situācija atbilst vairāk nekā vienam nosacījumam, tiks iestatīta visaugstākā smaguma 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 pārmantošanas noteikums	Noteikums pakalpojuma statusa pārmantošanai uz vecāka pakalpojumu: **Kā ir** - statuss tiek pārmantots bez izmaiņām **Palielināt par** - varat palielināt pārmantoto statusu par 1 līdz 5 smaguma pakāpēm **Samazināt par** - varat samazināt pārmantoto statusu par 1 līdz 5 smaguma pakāpēm **Ignorēt šo pakalpojumu** - statuss vispār netiek pārmantots uz vecāka pakalpojumu **Fiksēts statuss** - statuss tiek pārmantots statiski, t. i., vienmēr ir 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 smaguma pakāpi 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.

Cilnē *Tagi* ir ietverti pakalpojuma līmeņa tagi. Pakalpojuma līmeņa tagi tiek izmantoti pakalpojuma identificēšanai. Šāda veida tagi netiek izmantoti problēmu kartēšanai ar pakalpojumu (šim nolūkam izmantojiet *Problēmu tagus* pirmajā cilnē).

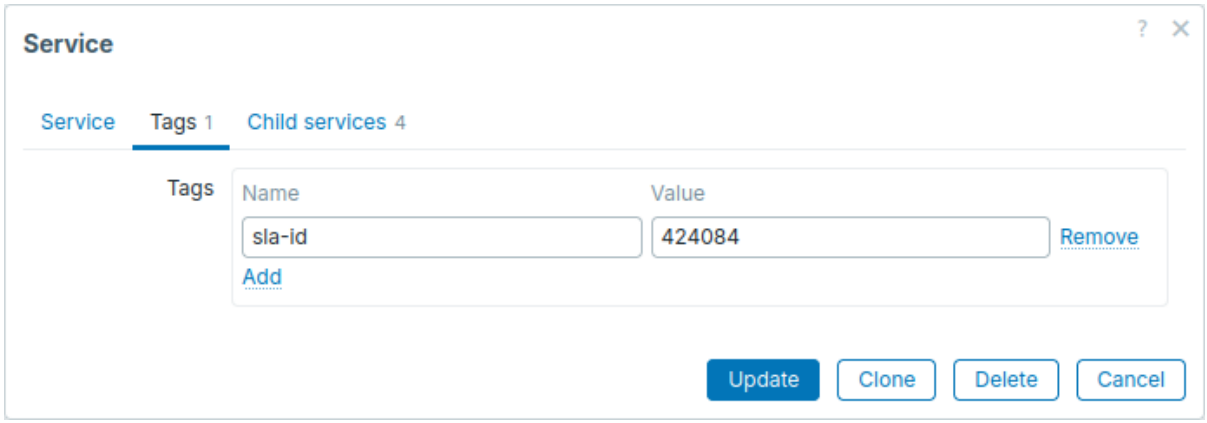
Cilnē *Pakārtotie pakalpojumi* varat norādīt atkarīgos pakalpojumus. Noklikšķiniet uz *Pievienot*, lai pievienotu pakalpojumu no esošo pakalpojumu saraksta. Ja vēlaties pievienot jaunu pakārtoto pakalpojumu, vispirms saglabājiet šo pakalpojumu un pēc tam noklikšķiniet uz pluszīmes ikonas blakus tikko izveidotajam pakalpojumam.

Tagi Servisos ir divu veidu tagi:

- Servisa tagi
- Problēmu tagi

Servisa tagi

Servisa tagi tiek izmantoti, lai saskaņotu servisus ar servisa darbībām un SLA. Šie tagi tiek norādīti servisa konfigurācijas cilnē *Tags*. SLA kartēšanai tiek izmantota *OR* loģika: serviss tiks kartēts uz SLA, ja tam ir vismaz viens atbilstošs tags. Servisa darbības kartēšanas noteikumi ir konfigurējami un var izmantot *AND*, *OR* vai *AND/OR* loģiku.

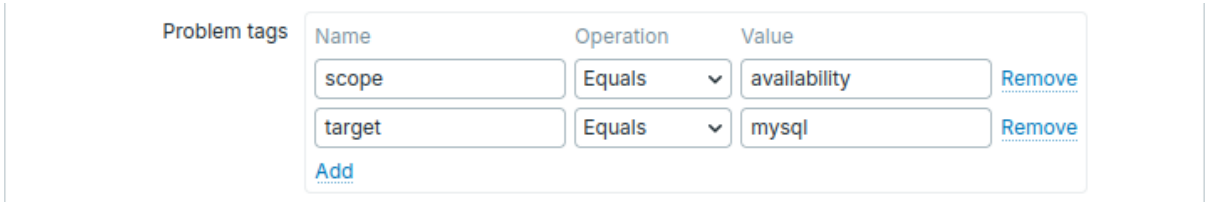


Problēmu tagi

Problēmu tagi tiek izmantoti, lai saskaņotu problēmas un pakalpojumus. Šie tagi tiek norādīti primārās pakalpojuma konfigurācijas cilnē.

Tikai zemākā hierarhijas līmeņa bērnpakalpojumiem var definēt problēmu tagus un tieši tos saistīt ar problēmām. Ja problēmu tagi sakrīt, pakalpojuma statuss mainīsies uz tādu pašu statusu, kāds ir problēmai. Ja ir vairākas problēmas, pakalpojumam būs visnopietnākās problēmas statuss. Pēc tam vecākpakalpojuma statuss tiek aprēķināts, pamatojoties uz bērnpakalpojumu statusiem, saskaņā ar statusa aprēķināšanas noteikumiem.

Ja ir norādīti vairāki tagi, tiek izmantota *AND* loģika: problēmai jābūt ar visiem pakalpojuma konfigurācijā norādītajiem tagiem, lai to varētu piesaistīt pakalpojumam.

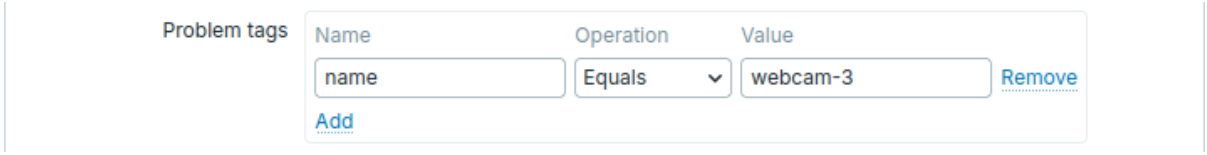


Note:
Problēma Zabbix pārmanto tagus no visas veidņu, hostu, vienumu, web scenāriju un trigeru ķēdes. Jebkuru no šiem tagiem var izmantot, lai saskaņotu problēmas ar pakalpojumiem.

Piemērs:

Problēmai *Web camera 3 is down* ir tagi `type:video-surveillance`, `floor:1` un `name:webcam-3`, kā arī statuss *Warning*

Pakalpojumam **Web camera 3** ir norādīts tikai viens problēmas tags: `name:webcam-3`



Pakalpojuma statuss mainīsies no *OK* uz *Warning*, kad šī problēma tiks konstatēta.

Ja pakalpojumam **Web camera 3** būtu problēmas tagi `name:webcam-3` un `floor:2`, tā statuss netiktu mainīts, kad problēma tiktu konstatēta, jo nosacījumi ir izpildīti tikai daļēji.

Esošo pakalpojumu modificēšana

Note:
Tālāk aprakstītās pogas ir redzamas tikai tad, ja sadaļa *Services* ir rediģēšanas režīmā.

Lai rediģētu esošu pakalpojumu, noklikšķiniet uz zīmuļa ikonas blakus pakalpojumam.

Lai klonētu esošu pakalpojumu, noklikšķiniet uz zīmuļa ikonas, lai atvērtu tā konfigurāciju, un pēc tam noklikšķiniet uz pogas *Clone*. Kad pakalpojums tiek klonēts, tā vecākelementu saites tiek saglabātas, bet bērnelementu saites netiek saglabātas.

Lai dzēstu pakalpojumu, noklikšķiniet uz x ikonas blakus tam. Dzēšot vecāku pakalpojumu, tā bērnu pakalpojumi netiks dzēsti un tiks pārvietoti vienu līmeni augstāk pakalpojumu kokā (1. līmeņa bērni iegūs tādu pašu līmeni kā dzēstais vecāku pakalpojums).

Divas pogas zem pakalpojumu saraksta piedāvā dažas masveida rediģēšanas iespējas:

- *Mass update* - pakalpojumu īpašību masveida atjaunināšana
- *Delete* - pakalpojumu dzēšana

Lai izmantotu šīs opcijas, atzīmējiet izvēles rūtiņas attiecīgo pakalpojumu priekšā, pēc tam noklikšķiniet uz vajadzīgās pogas.

SLA

Pārskats Kad pakalpojumi ir izveidoti, varat sākt pārraudzīt, vai to veiktspēja atbilst pakalpojumu līmeņa līgumam (SLA).

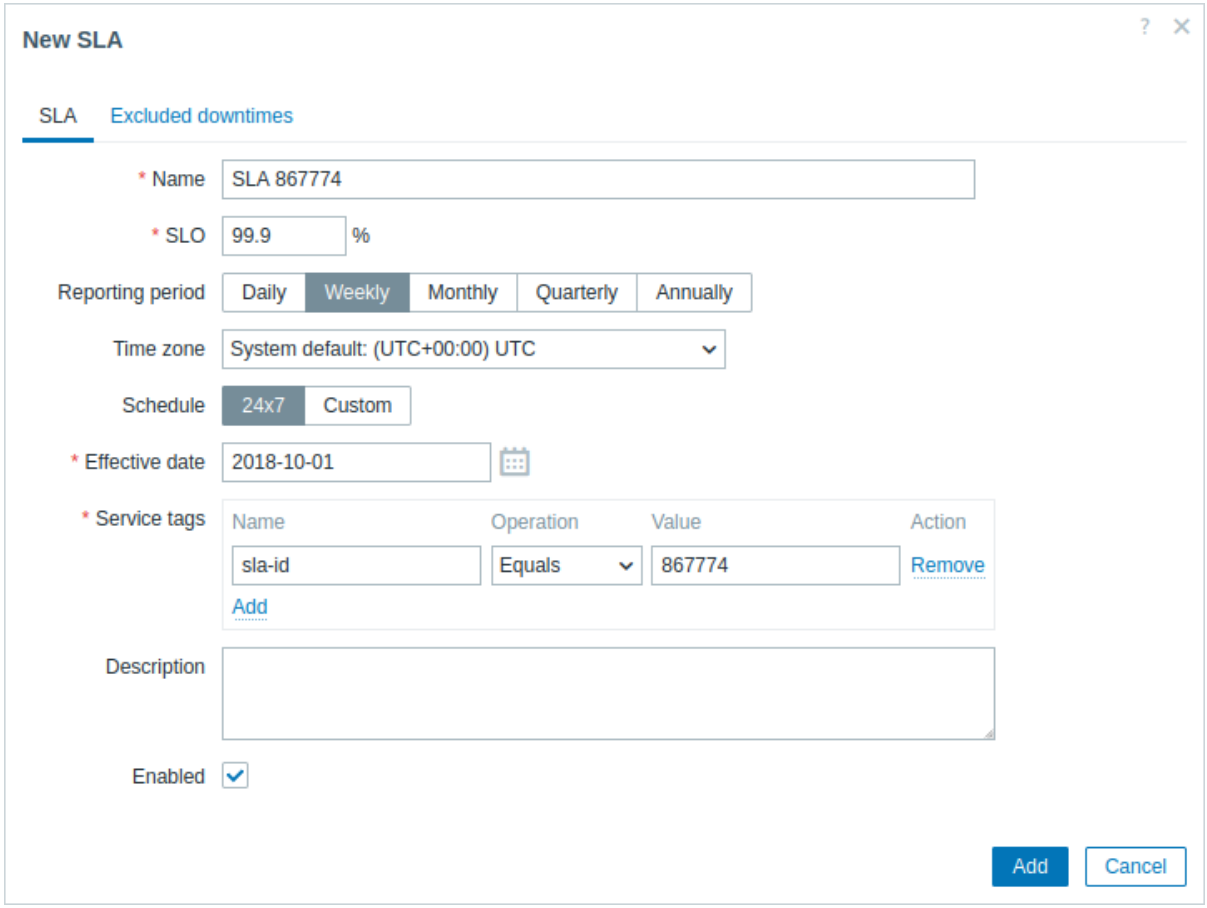
Izvēlnes sadaļa *Pakalpojumi* > *SLA* ļauj konfigurēt SLA dažādiem pakalpojumiem. Zabbix SLA definē pakalpojumu līmeņa mērķi (SLO), paredzēto darbības laika grafiku un plānotās dīkstāves.

SLA un pakalpojumi tiek sasaistīti, izmantojot pakalpojumu tagus. Vienu SLA var piemērot vairākiem pakalpojumiem - veiktspēja tiks mērīta katram atbilstošajam pakalpojumam atsevišķi. Vienam pakalpojumam var piešķirt vairākus SLA - dati par katru SLA tiks parādīti atsevišķi.

SLA pārskatos Zabbix nodrošina pakalpojumu līmeņa indikatora (SLI) datus, kas mēra faktisko pakalpojuma pieejamību. To, vai pakalpojums atbilst SLA mērķiem, nosaka, salīdzinot SLO (paredzēto pieejamību procentos) ar SLI (faktisko pieejamību procentos).

Konfigurācija Lai izveidotu jaunu SLA, noklikšķiniet uz pogas *Create SLA*.

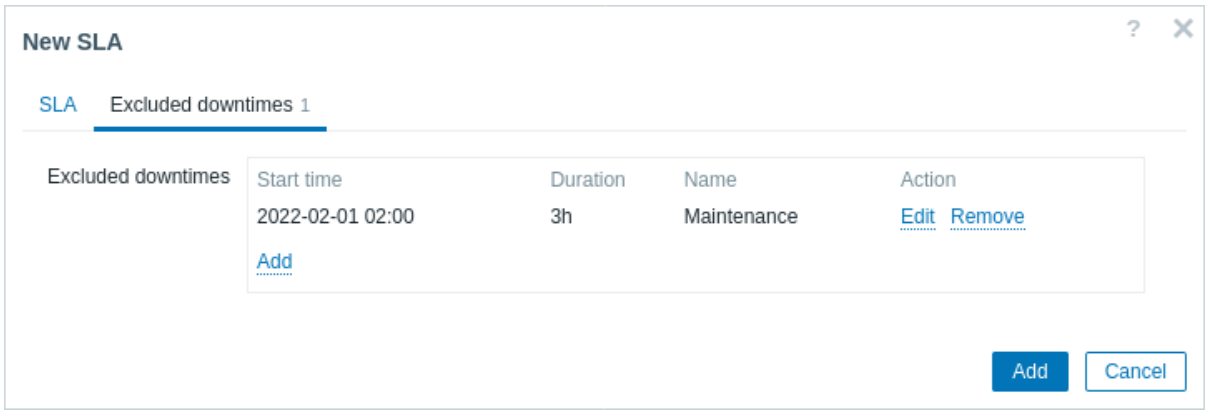
Cilnē *SLA* varat norādīt vispārīgos SLA parametrus.



Parametrs	Apraksts
Name	Ievadiet SLA nosaukumu.
SLO	Ievadiet pakalpojuma līmeņa mērķi (SLO) procentos.

Parametrs	Apraksts
Reporting period	Atlasiet SLI pārskata periodu, kas tiek izmantots SLI datu aprēķināšanai SLA pārskatā: **Daily** - katru dienu no 00:00:00 līdz 23:59:59. **Weekly** - katru nedēļu no svētdienas 00:00:00 līdz sestdienas 23:59:59. **Monthly** - katru mēnesi no pirmās dienas 00:00:00 līdz pēdējās dienas 23:59:59. **Quarterly** - katru kalendāro ceturksni: Q1 (janvāris–marts), Q2 (aprīlis–jūnijs), Q3 (jūlijs–septembris), Q4 (oktobris–decembris). **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 datumu, no kura sākt SLA aprēķināšanu.
Service tags	Pievienojiet pakalpojumu tagus, lai identificētu pakalpojumus, kuriem jāpiemēro šis SLA. **Name** - pakalpojuma taga nosaukums; tam precīzi jāsakrīt, reģistrs tiek ņemts vērā. **Operation** - atlasiet *Equals*, ja taga vērtībai precīzi jāsakrīt (reģistrs tiek ņemts vērā), vai *Contains*, ja jāsakrīt daļai taga vērtības (reģistrs netiek ņemts vērā). **Value** - pakalpojuma taga vērtība, kas jāmeklē atbilstoši atlasītajai darbībai. SLA tiek piemērots pakalpojumam, ja sakrīt vismaz viens pakalpojuma tags.
Description	Pievienojiet SLA aprakstu.
Enabled	Atzīmējiet izvēles rūtiņu, lai iespējotu SLA aprēķināšanu.

Cilnē *Excluded downtimes* varat norādīt dīkstāves periodus, kas tiek izslēgti no SLA aprēķina.



Noklikšķiniet uz *Add*, lai konfigurētu izslēgtos dīkstāves periodus, pēc tam ievadiet perioda nosaukumu, sākuma datumu un ilgumu.

SLA pārskati To, kā pakalpojums darbojas salīdzinājumā ar SLA, var redzēt SLA pārskatā. SLA pārskatus var skatīt:

- no *SLA* sadaļas, noklikšķinot uz SLA pārskata hipersaites;
- no *Services* sadaļas, noklikšķinot uz SLA nosaukuma informācijas cilnē;
- Dashboard logrīkā *SLA report*.

Kad SLA ir konfigurēts, *Info* cilnē pakalpojumu sadaļā tiks parādīta arī noteikta informācija par pakalpojuma veiktspēju.

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:

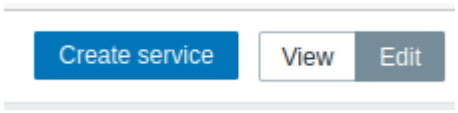
- *HA mezgls 1* ar vismaz vienu trigeri un birku (vēlams iestatītu trigera līmenī) `component:ha-node-1`
- *HA mezgls 2* ar vismaz vienu trigeri un birku (vēlams iestatītu trigera līmenī) `component:ha-node-2`

Pakalpojumu koks Nākamais solis ir izveidot pakalpojumu koku. Šajā piemērā infrastruktūra ir ļoti vienkārša un sastāv no trim pakalpojumiem: *Zabbix klasteris* (vecākais) un diviem pakalpojumiem pakārtotajā līmenī *Zabbix servera mezgls 1* un *Zabbix servera mezgls 2*.

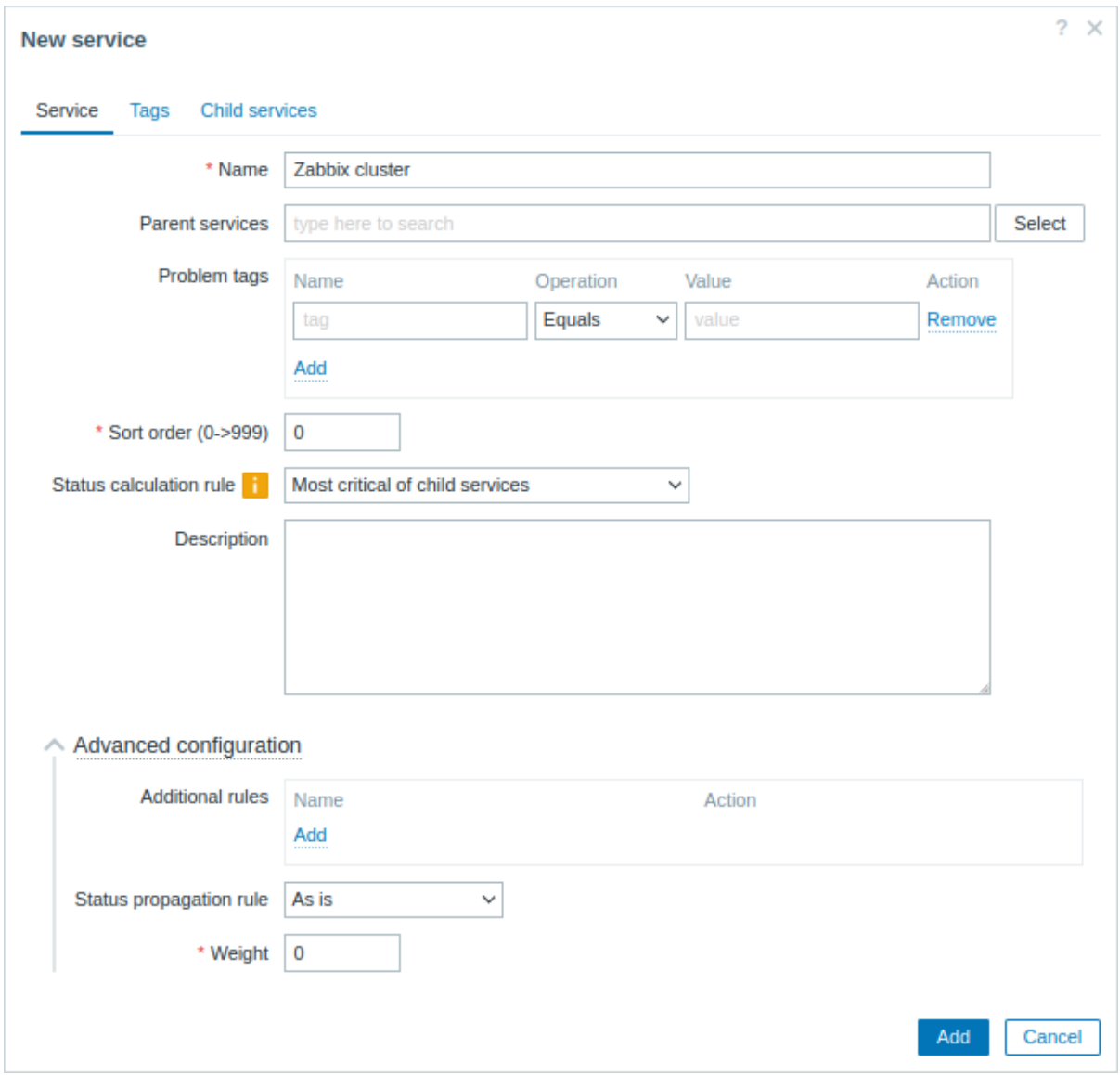
```
Zabbix klasteris
|
```

```
|- Zabbix servera mezgls 1
|- Zabbix servera mezgls 2
```

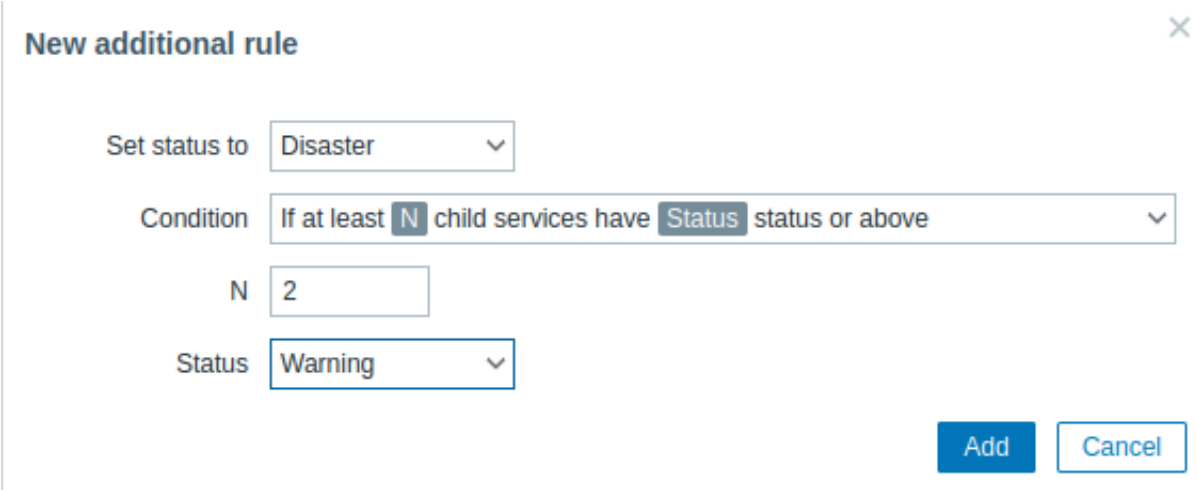
Pakalpojumu lapā ieslēdziet *Rediģēšanas* režīmu un noklikšķiniet uz *Izveidot pakalpojumu*:



Pakalpojuma konfigurācijas logā ievadiet nosaukumu *Zabbix klasteris* un noklikšķiniet uz *Papildu konfigurācija* virsraksta, lai parādītu papildu konfigurācijas opcijas.



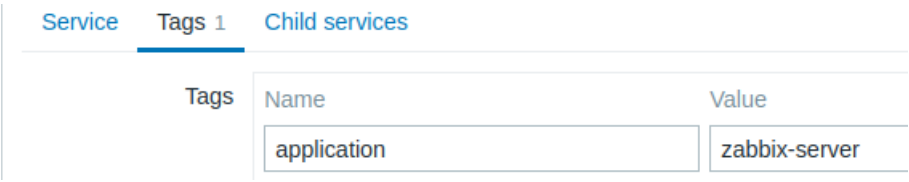
Konfigurējiet papildu noteikumu:



Zabbix klasterim būs divi pakārtotie pakalpojumi - pa vienam katram HA mezglam. Ja abiem HA mezgliem ir problēmas ar vismaz *Warning* statusu, vecākā pakalpojuma statusam jābūt iestatītam uz *Disaster*. Lai to panāktu, papildu noteikums jākonfigurē šādi:

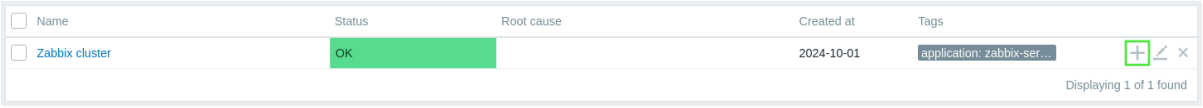
- Iestatīt statusu uz: Disaster
- Nosacījums: Ja vismaz `N` pakārtotajiem pakalpojumiem ir `Status` statuss vai augstāks
- N: 2
- Statuss: Warning

Pārslēdzieties uz cilni *Tagi* un pievienojiet tagu `application:zabbix-server`. Šis tags vēlāk tiks izmantots pakalpojumu darbībām un SLA pārskatiem.



Saglabājiet jauno pakalpojumu.

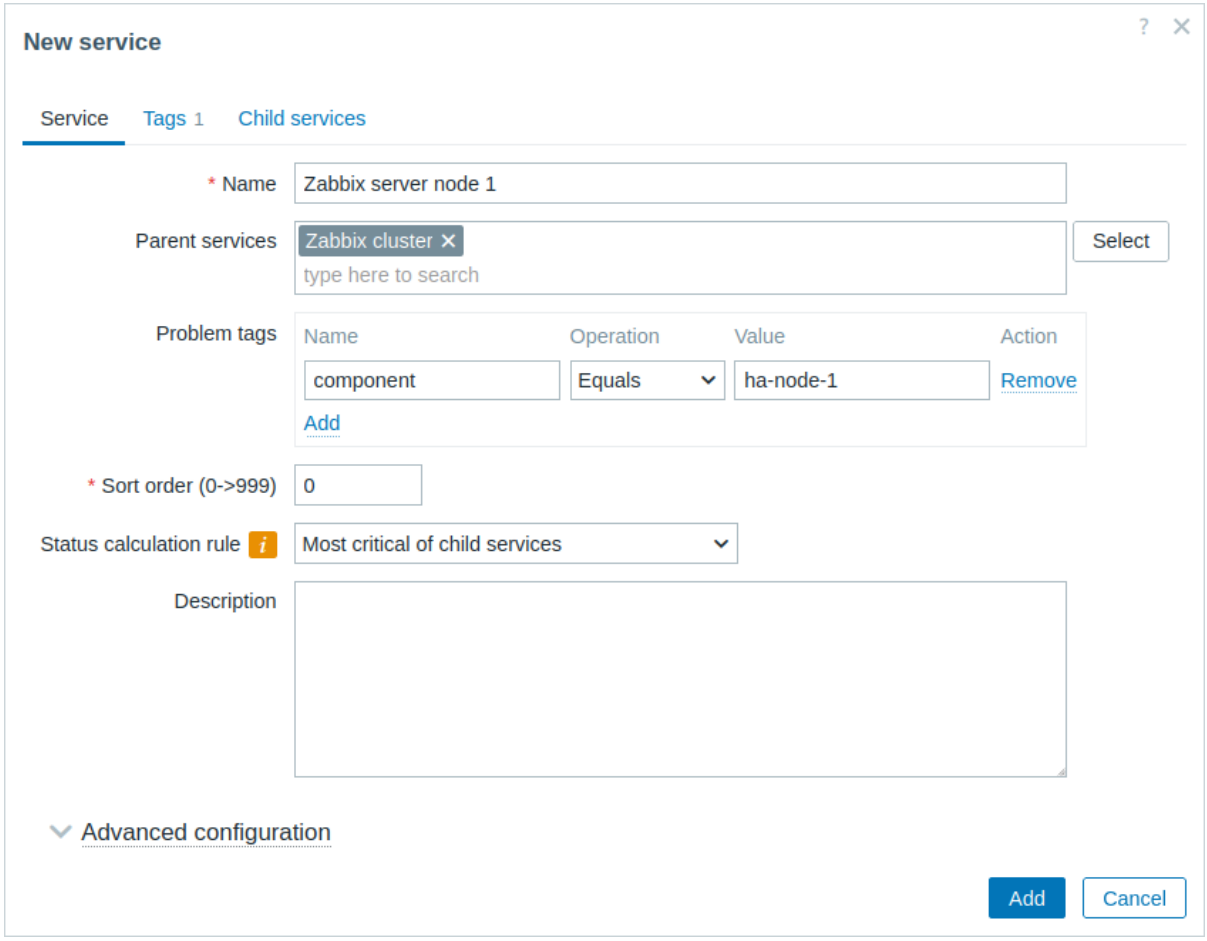
Lai pievienotu pakalpojuma bērnelementu, noklikšķiniet uz plusa ikonas blakus Zabbix klastera pakalpojumam (ikona ir redzama tikai *Rediģēšanas* režīmā).



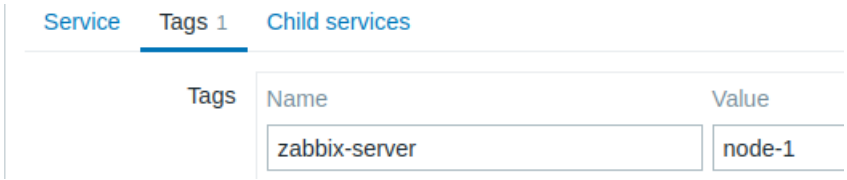
Pakalpojuma konfigurācijas logā ievadiet nosaukumu *Zabbix server node 1*. Ņemiet vērā, ka parametrs *Parent services* jau ir aizpildīts ar *Zabbix cluster*.

Šī pakalpojuma pieejamību ietekmē problēmas uz hosts *HA node 1*, kas atzīmēts ar problēmas tagu `component:ha-node-1`. Parametrā *Problem tags* ievadiet:

- Name: component
- Operation: Equals
- Value: ha-node-1



Pārslēdzieties uz cilni *Tags* un pievienojiet pakalpojuma tagu: `zabbix-server:node-1`. Šis tags vēlāk tiks izmantots pakalpojuma darbībām un SLA pārskatiem.



Saglabājiet jauno pakalpojumu.

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`.

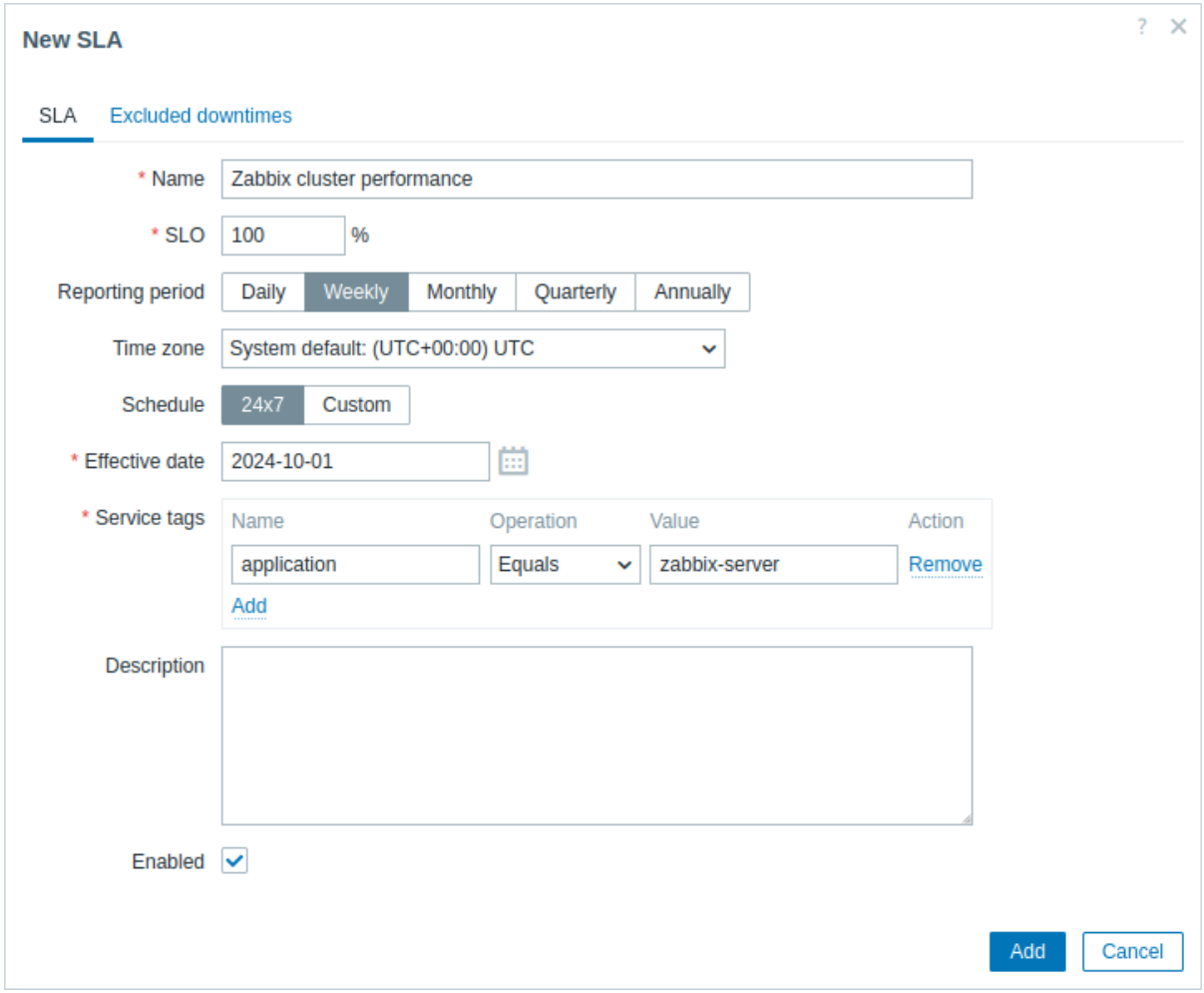
Saglabājiet jauno servisu.

SLA Šajā piemērā paredzētā Zabbix klastera veiktspēja ir 100%, izņemot pusgada apkopes periodu, kas ilgst vienu stundu.

Vispirms jāpievieno jauns pakalpojumu līmeņa līgums:

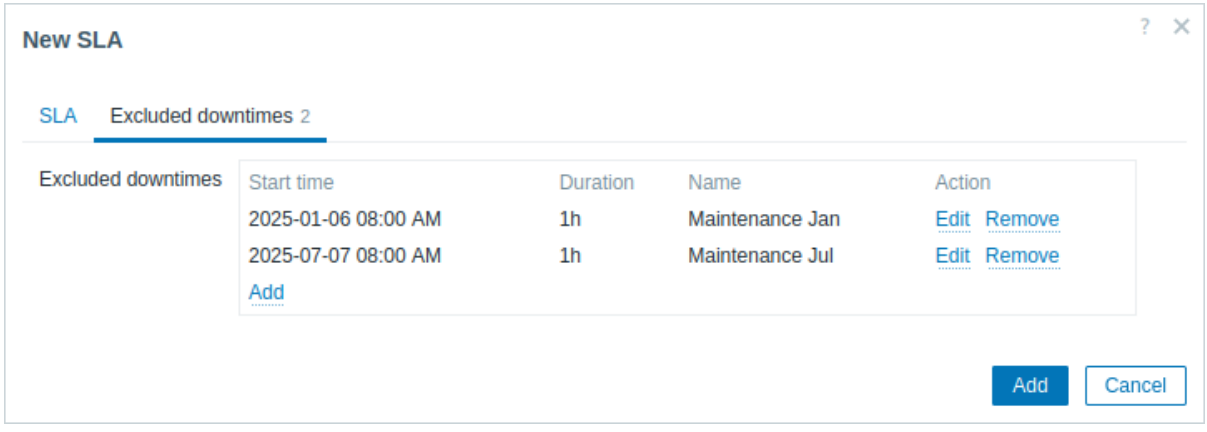
1. Dodieties uz *Services* > *SLA*.
2. Noklikšķiniet uz *Create SLA*.
3. Ievadiet nosaukumu *Zabbix cluster performance* un iestatiet SLO uz 100%. Pakalpojumam Zabbix cluster ir pakalpojuma tags `application:zabbix-server`.
4. Lai izmantotu šo SLA Zabbix klastera veiktspējas mērīšanai, parametrā *Service tags* norādiet:

- Name: application
- Operation: Equals
- Value: zabbix-server



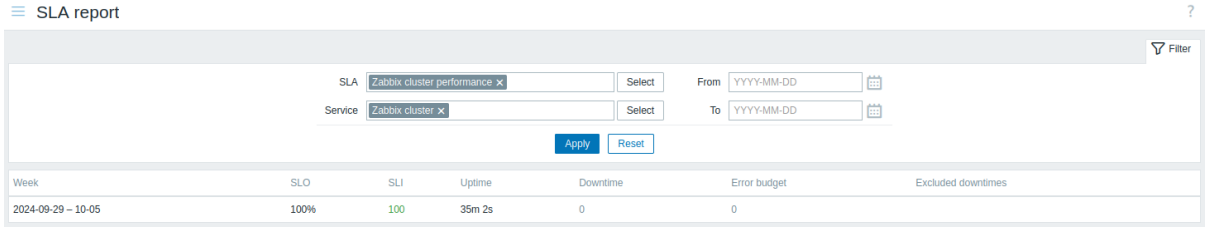
Reālā vidē varat arī atjaunināt vēlamo pārskata periodu, laika joslu un sākuma datumu vai mainīt grafiku no 24/7 uz pielāgotu. Šajā piemērā pietiek ar noklusējuma iestatījumiem.

5. Pārslēdzieties uz cilni *Excluded downtimes* un pievienojiet plānotās apkopes periodu dīkstāves, lai šie periodi netiktu iekļauti SLA aprēķinā.
6. Sadaļā *Excluded downtimes* noklikšķiniet uz saites *Add*, ievadiet dīkstāves nosaukumu, plānoto sākuma laiku un ilgumu.

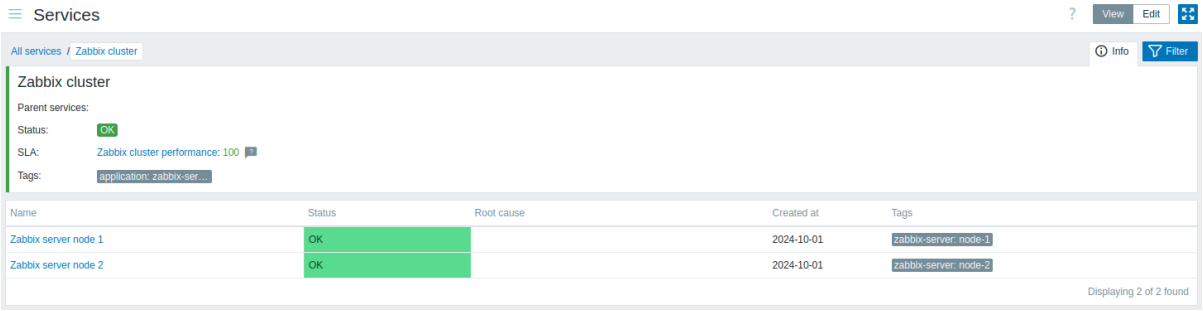


7. Noklikšķiniet uz *Add*, lai saglabātu jauno SLA.

Pārslēdzieties uz SLA pārskatu sadaļu, lai skatītu Zabbix klastera SLA pārskatu.



SLA informāciju var pārbaudīt arī sadaļā *Services*.



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 pakalpojumam "Child service" iepriekš bija saite uz "Parent service 1" ar stingro atkarību un papildus uz "Parent service 2" ar mīksto atkarību, tad pēc jaunināšanas "Child service" būs divi vecākpakalpojumi: "Parent service 1" un "Parent service 2".

Problēmu un pakalpojumu sasaisti, kas balstīta uz trigeriem, ir aizstājusi sasaistīšana, kas balstīta uz tagiem. Zabbix 6.0 un jaunākās versijās pakalpojuma konfigurācijas formai ir jauns parametrs *Problem tags*, kas ļauj norādīt vienu vai vairākus taga nosaukuma un vērtības pārus problēmu atbilstības noteikšanai. Trigeriem, kas bija saistīti ar pakalpojumu, tiks pievienots jauns tags `ServiceLink` : `<trigger ID>:<trigger name>` (taga vērtība tiks saīsināta līdz 32 rakstzīmēm). Saistītajiem pakalpojumiem tiks pievienots `ServiceLink` problēmu tags ar tādu pašu vērtību.

Statusa aprēķināšanas noteikumi

"Statusa aprēķināšanas algoritms" tiks uzlabots, izmantojot šādus noteikumus:

- Neaprēķināt → iestatīt statusu uz OK
- Problēma, ja vismaz vienam bērnelementam ir problēma → viskritiskākais no bērna pakalpojumiem
- Problēma, ja visiem bērnelementiem ir problēmas → viskritiskākais, ja visiem bērnelementiem ir problēmas

SLA Iepriekš SLA mērķi bija jādefinē katram pakalpojumam atsevišķi. Kopš Zabbix 6.0 SLA ir kļuvusi 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 tagus.

Veicot jaunināšanu:

- Identiskas SLA, kas definētas katram pakalpojumam, tiks grupētas, un katrai grupai tiks izveidota viena SLA.
- Katram ietekmētajam pakalpojumam tiks pievienots īpašs tags `SLA:<ID>`, un tas pats tags tiks norādīts atbilstošās SLA parametrā *Service tags*.
- Pakalpojuma izveides laiks, jauna metrika SLA pārskatos, esošajiem pakalpojumiem tiks iestatīts uz 01/01/2000 00:00.

Šifrēšana un drošība

Šifrēšana Zabbix atbalsta šifrētu saziņu starp Zabbix komponentēm, izmantojot Transport Layer Security (TLS) protokolu v.1.2 un 1.3 (atkarībā no kripto bibliotēkas). Tiek atbalstīta uz sertifikātiem balstīta un uz iepriekš koplietotu atslēgu balstīta šifrēšana.

Šifrēšanu var konfigurēt savienojumiem:

- Starp Zabbix serveri, Zabbix starpniekserveri, Zabbix aģentu, Zabbix tīmekļa pakalpojumu, zabbix_sender un zabbix_get utilītprogrammām
- Uz Zabbix datubāzi no Zabbix lietotāja saskarnes un servera/starpniekservera

- Starp Zabbix lietotāja saskarni un Zabbix serveri

Šifrēšana ir izvēles un konfigurējama atsevišķām komponentēm:

- Dažus starpniekserverus un aģentus var konfigurēt tā, lai tie ar serveri izmantotu uz sertifikātiem balstītu šifrēšanu, kamēr citi var izmantot uz iepriekš koplietotu atslēgu balstītu šifrēšanu, bet vēl citi turpina izmantot nešifrētu saziņu (kā iepriekš).
- Serveris (starpniekserveris) var izmantot atšķirīgas šifrēšanas konfigurācijas dažādiem hosts.

Zabbix dēmona programmas izmanto vienu klausīšanās portu šifrētiem un nešifrētiem ienākošajiem savienojumiem. Šifrēšanas pievienošana neprasa atvērt jaunus portus ugunsmūros.

Ierobežojumi

- Privātās atslēgas tiek glabātas vienkāršā tekstā failos, kurus Zabbix komponentes 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 pārlūkprogrammu.
- Pašlaik katrs šifrētais savienojums tiek atvērts ar pilnu TLS rokasspiedienu; sesiju kešošana un biļetes netiek ieviestas.
- Šifrēšanas pievienošana palielina vienums pārbaužu un darbību izpildes laiku atkarībā no tīkla latentuma:
  - Piemēram, ja pakešu aizture ir 100 ms, tad TCP savienojuma atvēršana un nešifrēta pieprasījuma nosūtīšana aizņem apmēram 200 ms. Ar šifrēšanu TLS savienojuma izveidei tiek pievienotas aptuveni 1000 ms.
  - Var būt nepieciešams palielināt noildzes, pretējā gadījumā daži vienumi un darbības, kas uz aģentiem palaiž attālinātus skriptus, var darboties ar nešifrētiem savienojumiem, bet ar šifrētiem savienojumiem neizdoties noildzes dēļ.
- Šifrēšana netiek atbalstīta tīkla atklāšanā. Zabbix aģents pārbaudes, kas tiek veiktas tīkla atklāšanas laikā, būs nešifrētas, un, ja Zabbix aģents ir konfigurēts noraidīt nešifrētus savienojumus, šādas pārbaudes neizdosies.

Zabbix kompilēšana ar šifrēšanas atbalstu Lai atbalstītu šifrēšanu, Zabbix ir jākompilē un jāsaista ar kādu no atbalstītajām kriptogrāfijas bibliotēkām:

- GnuTLS - sākot ar versiju 3.1.18
- OpenSSL - versijas 1.0.1, 1.0.2, 1.1.0, 1.1.1, 3.0.x - 3.5.x
- LibreSSL - pārbaudīts 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; var izmantot tikai sertifikātus.

Note:
Papildinformāciju par SSL iestatīšanu Zabbix lietotāja saskarnei skatiet šajās labākajās praksēs.

Bibliotēku izvēlas, skriptam "configure" norādot attiecīgo opciju:

- `--with-gnutls[=DIR]`
- `--with-openssl[=DIR]` (tiek izmantots arī LibreSSL)

Piemēram, lai konfigurētu servera un aģenta pirmkodu ar *OpenSSL*, varat izmantot aptuveni šādu komandu:

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

Dažādus Zabbix komponentus var kompilēt ar atšķirīgām kriptogrāfijas bibliotēkām (piemēram, serveri ar *OpenSSL*, aģentu ar *GnuTLS*).

Attention:
Ja plānojat izmantot iepriekš koplietotas atslēgas (PSK), apsveriet iespēju Zabbix komponentos, kas izmanto PSK, izmantot *GnuTLS* vai *OpenSSL 1.1.0* (vai jaunāku) bibliotēku. *GnuTLS* un *OpenSSL 1.1.0* bibliotēkas atbalsta PSK šifrēšanas komplektus ar perfektu pārsūtīšanas slepenību. Vecākas *OpenSSL* bibliotēkas versijas (1.0.1, 1.0.2c) arī atbalsta PSK, taču pieejamie PSK šifrēšanas komplekti nenodrošina perfektu pārsūtīšanas slepenību.

Savienojumu šifrēšanas pārvaldība Savienojumos Zabbix var izmantot:

- nešifrētu savienojumu (noklusējums)
- šifrēšanu, izmantojot RSA sertifikātu
- šifrēšanu, izmantojot PSK

Lai norādītu šifrēšanu starp Zabbix komponentēm, tiek izmantoti divi svarīgi parametri:

- TLSConnect - norāda, kādu šifrēšanu izmantot izejošajiem savienojumiem (nešifrētu savienojumu, PSK vai sertifikātu)
- TLSAccept - norāda, kādi savienojumu veidi ir atļauti ienākošajiem savienojumiem (nešifrēts savienojums, PSK vai sertifikāts). Var norādīt vienu vai vairākas vērtības.

`TLSConnect` tiek izmantots Zabbix starpniekservera konfigurācijas failos (aktīvajā režīmā norāda tikai savienojumus ar serveri) un Zabbix aģenta konfigurācijas failos (aktīvajām pārbaudēm). Zabbix lietotāja saskarnē `TLSConnect` ekvivalents ir lauks *Savienojumi ar hostu* sadaļā *Datu apkopošana* > *Hosti* > <some host> > cilnē *Šifrēšana* un lauks *Savienojumi ar starpniekserveri* sadaļā *Administrēšana* > *Starpniekserveri* > <some proxy> > cilnē *Šifrēšana*. Ja konfigurētais savienojuma šifrēšanas veids neizdodas, citi šifrēšanas veidi netiks mēģināti.

`TLSAccept` tiek izmantots Zabbix starpniekservera konfigurācijas failos (pasīvajā režīmā norāda tikai savienojumus no servera) un Zabbix aģenta konfigurācijas failos (pasīvajām pārbaudēm). Zabbix lietotāja saskarnē TLSAccept ekvivalents ir lauks *Savienojumi no hosta* sadaļā *Datu apkopošana* > *Hosti* > <some host> > cilnē *Šifrēšana* un lauks *Savienojumi no starpniekservera* sadaļā *Administrēšana* > *Starpniekserveri* > <some proxy> > cilnē *Šifrēšana*.

Parasti ienākošajiem savienojumiem konfigurē tikai vienu šifrēšanas veidu. Tomēr var būt nepieciešams mainīt šifrēšanas veidu, piemēram, no nešifrēta savienojuma uz šifrēšanu, izmantojot sertifikātu, ar minimālu dīkstāvi un iespēju atjaunot iepriekšējo konfigurāciju. Lai to paveiktu:

- Iestatiet `TLSAccept=unencrypted,cert` aģenta konfigurācijas failā un restartējiet Zabbix aģentu.
- Pārbaudiet savienojumu ar aģentu, izmantojot `zabbix_get` un sertifikātu. Ja tas darbojas, varat pārkonfigurēt šī aģenta šifrēšanu Zabbix lietotāja saskarnē sadaļā *Datu apkopošana* > *Hosti* > <some host> > cilnē *Šifrēšana*, iestatot laukā *Savienojumi ar hostu* vērtību *Sertifikāts*.
- Kad servera konfigurācijas kešatmiņa tiek atjaunināta (un starpniekservera konfigurācija tiek atjaunināta, ja hosts tiek uzraudzīts, izmantojot starpniekserveri), savienojumi ar šo aģentu tiks šifrēti.
- Ja viss darbojas, kā paredzēts, aģenta konfigurācijas failā varat iestatīt `TLSAccept=cert` un restartēt Zabbix aģentu. Tagad aģents pieņems tikai šifrētus savienojumus, kuru pamatā ir sertifikāts. Nešifrēti savienojumi un savienojumi, kuru pamatā ir PSK, tiks noraidīti.

Līdzīgi tas darbojas arī serverī un starpniekserverī. Ja Zabbix lietotāja saskarnē hosta konfigurācijā laukam *Savienojumi no hosta* ir iestatīta vērtība "Sertifikāts", no aģenta (aktīvās pārbaudes) un zabbix_sender (trapper vienumi) tiks pieņemti tikai šifrēti savienojumi, kuru pamatā ir sertifikāts.

Visticamāk, ienākošajiem un izejošajiem savienojumiem konfigurēsiet vienādu šifrēšanas veidu vai arī neizmantosiet šifrēšanu. Tomēr tehniski ir iespējams konfigurēt šifrēšanu asimetriski, piemēram, ienākošajiem savienojumiem izmantot šifrēšanu, kuras pamatā ir sertifikāts, bet izejošajiem savienojumiem - šifrēšanu, kuras pamatā ir PSK.

Katra hosta šifrēšanas konfigurācija tiek parādīta Zabbix lietotāja saskarnē sadaļā *Datu apkopošana* > *Hosti*, kolonnā *Aģenta šifrēšana*. Piemēram:

Piemērs	Savienojumi ar hostu	Atļautie savienojumi no hosta	Noraidītie savienojumi no hosta
NONE	Nešifrēts	Nešifrēts	Šifrēts, ar sertifikātu un PSK šifrēts
CERT NONE PSK CERT	Šifrēts, izmantojot sertifikātu	Šifrēts, izmantojot sertifikātu	Nešifrēts un ar PSK šifrēts
PSK NONE PSK CERT	Šifrēts, izmantojot PSK	Šifrēts, izmantojot PSK	Nešifrēts un ar sertifikātu šifrēts
PSK NONE PSK CERT	Šifrēts, izmantojot PSK	Nešifrēts un ar PSK šifrēts	Šifrēts, izmantojot sertifikātu
CERT NONE PSK CERT	Šifrēts, izmantojot sertifikātu	Nešifrēts, ar PSK vai sertifikātu šifrēts	-

Attention:
Pēc noklusējuma savienojumi ir nešifrēti. Šifrēšana ir jākonfigurē katram hostam un starpniekserverim atsevišķi.

zabbix_get un zabbix_sender ar šifrēšanu Skatiet zabbix_get un zabbix_sender rokasgrāmatas lapas par to lietošanu ar šifrēšanu.

Šifru komplekti Šifru komplekti pēc noklusējuma tiek konfigurēti iekšēji Zabbix startēšanas laikā.

Tiek atbalstīti arī lietotāja konfigurēti šifru komplekti GnuTLS un OpenSSL. Lietotāji var konfigurēt šifru komplektus atbilstoši savām drošības politikām. Šīs funkcijas izmantošana nav obligāta (iebūvētie noklusējuma šifru komplekti joprojām darbojas).

Kriptogrāfijas bibliotēkām, kas kompilētas ar noklusējuma iestatījumiem, Zabbix iebūvētie noteikumi parasti rezultējas šādos šifru komplektos (secībā no augstākas uz zemāku prioritāti):

Library	Certificate ciphersuites	PSK ciphersuites
GnuTLS 3.1.18	TLS_ECDHE_RSA_AES_128_GCM_SHA256 TLS_ECDHE_RSA_AES_128_CBC_SHA256 TLS_ECDHE_RSA_AES_128_CBC_SHA1 TLS_RSA_AES_128_GCM_SHA256 TLS_RSA_AES_128_CBC_SHA256 TLS_RSA_AES_128_CBC_SHA1	TLS_ECDHE_PSK_AES_128_CBC_SHA256 TLS_ECDHE_PSK_AES_128_CBC_SHA1 TLS_PSK_AES_128_GCM_SHA256 TLS_PSK_AES_128_CBC_SHA256 TLS_PSK_AES_128_CBC_SHA1
OpenSSL 1.0.2c	ECDHE-RSA-AES128-GCM-SHA256 ECDHE-RSA-AES128-SHA256 ECDHE-RSA-AES128-SHA AES128-GCM-SHA256 AES128-SHA256 AES128-SHA	PSK-AES128-CBC-SHA
OpenSSL 1.1.0	ECDHE-RSA-AES128-GCM-SHA256 ECDHE-RSA-AES128-SHA256 ECDHE-RSA-AES128-SHA AES128-GCM-SHA256 AES128-CCM8 AES128-CCM AES128-SHA256 AES128-SHA	ECDHE-PSK-AES128-CBC-SHA256 ECDHE-PSK-AES128-CBC-SHA PSK-AES128-GCM-SHA256 PSK-AES128-CCM8 PSK-AES128-CCM PSK-AES128-CBC-SHA256 PSK-AES128-CBC-SHA
OpenSSL 1.1.1d	TLS_AES_256_GCM_SHA384 TLS_CHACHA20_POLY1305_SHA256 TLS_AES_128_GCM_SHA256 ECDHE-RSA-AES128-GCM-SHA256 ECDHE-RSA-AES128-SHA256 ECDHE-RSA-AES128-SHA AES128-GCM-SHA256 AES128-CCM8 AES128-CCM AES128-SHA256 AES128-SHA	TLS_CHACHA20_POLY1305_SHA256 TLS_AES_128_GCM_SHA256 ECDHE-PSK-AES128-CBC-SHA256 ECDHE-PSK-AES128-CBC-SHA PSK-AES128-GCM-SHA256 PSK-AES128-CCM8 PSK-AES128-CCM PSK-AES128-CBC-SHA256 PSK-AES128-CBC-SHA

Lietotāja konfigurēti ciphersuites Iebūvētie ciphersuite atlases kritēriji var tikt pārrakstīti, izmantojot lietotāja konfigurētus ciphersuites.

> **Attention:**
> Lietotāja konfigurēti ciphersuites ir funkcija, kas paredzēta pieredzējušiem lietotājiem, kuri izprot TLS ciphersuites, to drošību un kļūdu sekas, kā arī jūtas pārliecināti TLS problēmu novēršanā.

Iebūvētie ciphersuite atlases kritēriji var tikt pārrakstīti, izmantojot šādus parametrus:

Override scope	Parameter	Value	Description
Ciphersuite selection for certificates	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 ciphersuite 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 ciphersuite atlases kritēriji TLS 1.2/1.3 (GnuTLS), TLS 1.2 (OpenSSL)
Ciphersuite selection for 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 ciphersuite atlases kritēriji TLS 1.3 Tikai OpenSSL 1.1.1 vai jaunāka versija.

Override scope	Parameter	Value	Description
	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 ciphersuite atlases kritēriji TLS 1.2/1.3 (GnuTLS), TLS 1.2 (OpenSSL)
Combined ciphersuite list for certificate and PSK	TLSCipherAll13	Derīgas OpenSSL 1.1.1 cipher strings TLS 1.3 protokolam (to vērtības tiek nodotas OpenSSL funkcijai `SSL_CTX_set_ciphersuites()`).	Ciphersuite atlases kritēriji TLS 1.3 Tikai OpenSSL 1.1.1 vai jaunāka versija.
	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()`.	Ciphersuite atlases kritēriji TLS 1.2/1.3 (GnuTLS), TLS 1.2 (OpenSSL)

Lai pārrakstītu ciphersuite 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 nevar 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 utilītprogrammām var izmantot komandrindas parametrus `--tls-cipher13` vai `--tls-cipher` (šifrēšana ir nepārprotami norādīta ar `--tls-connect` parametru)

Ienākošie savienojumi

Ar ienākošajiem savienojumiem ir nedaudz sarežģītāk, jo noteikumi ir specifiski komponentiem un konfigurācijai.

Zabbix **aģentam**:

Agent connection setup	Cipher configuration
TLSConnect=cert	TLSCipherCert, TLSCipherCert13
TLSConnect=psk	TLSCipherPSK, TLSCipherPSK13
TLSAccept=cert	TLSCipherCert, TLSCipherCert13
TLSAccept=psk	TLSCipherPSK, TLSCipherPSK13
TLSAccept=cert,psk	TLSCipherAll, TLSCipherAll13

Zabbix **serverim** un **starpniekserverim**:

Connection setup	Cipher configuration
Outgoing connections using PSK	TLSCipherPSK, TLSCipherPSK13
Incoming connections using certificates	TLSCipherAll, TLSCipherAll13
Incoming connections using PSK if server has no certificate	TLSCipherPSK, TLSCipherPSK13
Incoming connections using PSK if server has certificate	TLSCipherAll, TLSCipherAll13

Iepriekš redzamajās divās tabulās var saskatīt noteiktu likumsakarību:

- TLSCipherAll un TLSCipherAll13 var norādīt tikai tad, ja tiek izmantots apvienots sertifikātu- **un** PSK balstītu šifru komplektu saraksts. Tas notiek divos gadījumos: serveris (starpniekserveris) ar konfigurētu sertifikātu (PSK šifru komplekti vienmēr ir konfigurēti serverī, starpniekserverī ja kriptogrāfijas bibliotēka atbalsta PSK), aģents ir konfigurēts pieņemt gan sertifikātu-, gan 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.

Parameter	GnuTLS 3.6.12
TLSCipherCert	NONE:+VERS-TLS1.2:+ECDHE-RSA:+RSA:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL:+COMP-NULL:+SIGN-ALL:+CTYPE-X.509
TLSCipherPSK	NONE:+VERS-TLS1.2:+ECDHE-PSK:+PSK:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL:+COMP-NULL:+SIGN-ALL
TLSCipherAll	NONE:+VERS-TLS1.2:+ECDHE-RSA:+RSA:+ECDHE-PSK:+PSK:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL:+COMP-NULL:+SIGN-ALL:+CTYPE-X.509

Parameter	OpenSSL 1.1.1d [1]
TLSCipherCert13	
TLSCipherCert	EECDH+aRSA+AES128:RSA+aRSA+AES128
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 ir atšķirīgas vecākām OpenSSL versijām (1.0.1, 1.0.2, 1.1.0), LibreSSL un gadījumā, ja OpenSSL ir kompilēts bez PSK atbalsta.

Lietotāja konfigurētu šifru komplektu piemēri

Tālāk skatiet šādus lietotāja konfigurētu šifru komplektu piemērus:

- Šifru virkņu testēšana un atļauja tikai PFS šifru komplektiem
- Pārslēgšanās no AES128 uz AES256

Šifrkopu virkņu testēšana un tikai PFS šifrkopu atļaušana

Lai redzētu, kuras šifrkopas ir atlasītas, konfigurācijas failā jāiestata `DebugLevel=4` vai jāizmanto opcija `-vv` programmai zabbix_sender.

Pirms vēlamo šifrkopu iegūšanas var būt nepieciešami vairāki eksperimenti ar `TLSCipher*` parametriem. Ir neērti vairākas reizes restartēt Zabbix serveri, starpniekserveri vai aģentu tikai tādēļ, lai pielāgotu `TLSCipher*` parametrus. Ērtākas iespējas ir izmantot zabbix_sender vai komandu `openssl`. Apskatīsim abas.

1. Izmantojot zabbix_sender.

Izveidosim testa konfigurācijas failu, piemēram, `/home/zabbix/test.conf`, izmantojot `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
```

Šajā piemērā ir nepieciešami derīgi CA un aģenta sertifikāti, kā arī PSK. Pielāgojiet sertifikāta 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
```

Atlasītās šifrkopas var apskatīt, 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 redzamas pēc noklusējuma atlasītās šifrkopas. Šīs noklusējuma vērtības ir izvēlētas, lai nodrošinātu savietojamību ar Zabbix aģentiem, kas darbojas sistēmās ar vecākām OpenSSL versijām (sākot ar 1.0.1).

Jaunākās sistēmās varat uzlabot drošību, atļaujot tikai dažas šifrkopas, piemēram, tikai šifrkopas ar PFS (Perfect Forward Secrecy). Konfigurēsim `TLSCipher*` parametrus, lai atļautu tikai šifrkopas, kas atbalsta PFS.

Attention:
Ja tiek izmantots PSK, rezultāts nebūs savietojams ar sistēmām, kurās tiek izmantots OpenSSL 1.0.1 un 1.0.2. Uz sertifikātiem balstītai šifrēšanai vajadzētu darboties.

Pievienojiet failam `test.conf` 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
```

Saraksti "certificate ciphersuites" un "PSK ciphersuites" ir mainījušies - tie ir īsāki nekā iepriekš un satur tikai TLS 1.3 šifrkopas un TLS 1.2 ECDHE-* šifrkopas, kā paredzēts.

2. TLSCipherAll un TLSCipherAll13 nevar testēt ar zabbix_sender; tie neietekmē iepriekšējā piemērā parādīto vērtību "certificate and PSK ciphersuites". Lai pielāgotu TLSCipherAll un TLSCipherAll13, jāeksperimentē ar aģentu, starpniekserveri vai serveri.

Tādēļ, lai atļautu tikai PFS šifrkopas, iespējams, `zabbix_agentd.conf`, `zabbix_proxy.conf` un `zabbix_server.conf` failiem jāpievieno līdz trim parametriem, ja katram no tiem ir konfigurēts sertifikāts un aģentam ir konfigurēts arī PSK:

```
TLSCipherCert=EECDH+aRSA+AES128
TLSCipherPSK=kECDHEPSK+AES128
TLSCipherAll=EECDH+aRSA+AES128:kECDHEPSK+AES128
```

Ja Zabbix vidē tiek izmantota tikai uz PSK balstīta šifrēšana un nav sertifikātu, nepieciešams tikai viens parametrs:

```
TLSCipherPSK=kECDHEPSK+AES128
```

Tagad, kad saprotat, kā tas darbojas, šifrkopu atlasi varat testēt arī ārpus Zabbix, izmantojot komandu `openssl`. Pārbaudīsim visu 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
```

Varat dot priekšroku `openssl ciphers` ar opciju `-V`, 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 prioritātes 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 izmanto AES128 kā iebūvēto noklusējumu datiem. Pieņemsim, ka jūs izmantojat sertifikātus un vēlaties pārslēgties uz AES256, izmantojot OpenSSL 1.1.1.

To var panākt, pievienojot attiecīgos parametrus failā `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, tiek definēti arī parametri `TLSCipherPSK*`, lai izvairītos no to noklusējuma vērtībām, kas plašākas savietojamības dēļ ietver mazāk drošus šifrus. PSK šifru komplektus serverī/starpniekserverī nevar pilnībā atspējot.

Un failā `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
```

Šifrēšanas problēmu novēršana Vispārīgi ieteikumi:

- Sāciet ar to, ka noskaidrojat, kurš komponents konkrētajā problēmas gadījumā darbojas kā TLS klients un kurš kā TLS serveris.
 Zabbix serveris, starpniekserveri un aģenti atkarībā no mijiedarbības starp tiem visi var darboties gan kā TLS serveri, gan kā klienti.
 Piemēram, Zabbix serveris, kas savienojas ar aģentu pasīvās pārbaudes veikšanai, darbojas kā TLS klients. Aģents darbojas TLS servera lomā.
 Zabbix aģents, kas no starpniekservera pieprasa aktīvo pārbaužu sarakstu, darbojas kā TLS klients. Starpniekserveris darbojas TLS servera lomā.
 `zabbix_get` un `zabbix_sender` utilītas vienmēr darbojas kā TLS klienti.
- Zabbix izmanto savstarpējo autentifikāciju.
 Katra puse pārbauda savu pretpusi un var atteikt savienojumu.
 Piemēram, Zabbix serveris, kas savienojas ar aģentu, var nekavējoties pārtraukt savienojumu, ja aģenta sertifikāts nav derīgs. Un otrādi - Zabbix aģents, kas pieņem savienojumu no servera, var pārtraukt savienojumu, ja serverim aģents neuzticas.
- Pārbaudiet žurnālfailus abās pusēs - gan TLS klientā, gan TLS serverī.
 Puse, kas atsakās no savienojuma, var precīzi norādīt iemeslu, kāpēc savienojums tika atteikts. Otra puse bieži ziņo vispārīgāku kļūdu (piem., "Connection closed by peer", "connection was non-properly terminated").
- Dažkārt nepareizi konfigurēta šifrēšana rada mulsinošus kļūdu ziņojumus, kas nekādi nenorāda uz patieso cēloni.
 Tālākajās apakšnodaļās mēs cenšamies sniegt (tālu no izsmeļoša) ziņojumu un iespējamo cēloņu kopumu, kas var palīdzēt problēmu novēršanā.
 Lūdzu, ņemiet vērā, ka dažādi kriptogrāfijas rīkkopas (OpenSSL, GnuTLS) bieži rada atšķirīgus

kļūdu ziņojumus vienās un tajās pašās problēmsituācijās.
 Dažkārt kļūdu ziņojumi ir atkarīgi pat no konkrētas kriptogrāfijas rīkkopu kombinācijas abās pusēs.

Savienojuma veida vai atļauju problēmu novēršana Serveris ir konfigurēts savienoties 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 pārtrauca attālais mezgls. Pārbaudiet atļautos savienojuma veidus un piekļuves tiesības
```

Viena puse savienojas ar sertifikātu, bet otra puse pieņem tikai PSK vai otrādi

Jebkurā žurnālā (ar *GnuTLS*):

```
neizdevās pieņemt ienākošu savienojumu: no 127.0.0.1: zbx_tls_accept(): gnutls_handshake() neizdevās:\
    -21 Nevarēja vienoties par atbalstītu šifra komplektu.
```

Jebkurā žurnālā (ar *OpenSSL* 1.0.2c):

```
neizdevās pieņemt ienākošu 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 write fatal alert "handshake failure"
```

Mēģinājums izmantot Zabbix sender, kas kompilēts ar TLS atbalstu, lai nosūtītu datus uz Zabbix serveri/starpniekserveri, kas kompilēts bez TLS atbalsta Savienojuma puses žurnālā:

Linux:

```
...In zbx_tls_init_child()
...OpenSSL library (version OpenSSL 1.1.1  11 Sep 2018) initialized
...
...In zbx_tls_connect(): psk_identity:"PSK test sender"
...End of zbx_tls_connect():FAIL error:'connection closed by peer'
...send value error: TCP successful, cannot establish TLS to [[localhost]:10051]: connection closed by pee
```

Windows:

```
...OpenSSL library (version OpenSSL 1.1.1a  20 Nov 2018) initialized
...
...In zbx_tls_connect(): psk_identity:"PSK test sender"
...zbx_psk_client_cb() requested PSK identity "PSK test sender"
...End of zbx_tls_connect():FAIL error:'SSL_connect() I/O error: [0x00000000] The operation completed succ
...send value error: TCP successful, cannot establish TLS to [[192.168.1.2]:10051]: SSL_connect() I/O erro
```

Pie pieņemošās puses žurnāla:

```
...neizdevās pieņemt ienākošo savienojumu: no 127.0.0.1: TLS atbalsts netika iekļauts kompilēšanas laikā
```

Viena puse savienojas ar PSK, bet otra puse izmanto LibreSSL vai ir kompilēta bez šifrēšanas atbalsta LibreSSL neatbalsta PSK.

Savienojošās puses žurnālā:

```
...TCP veiksmīgs, nevar izveidot TLS savienojumu ar [[192.168.1.2]:10050]: SSL_connect() I/O kļūda: [0] Su
```

Pieņemošās puses žurnālā:

```
...neizdevās pieņemt ienākošu savienojumu: no 192.168.1.2: PSK atbalsts netika iekļauts kompilācijā
```

Zabbix lietotāja saskarnē:

```
Neizdevās iegūt vērtību no aģents: TCP veiksmīgs, nevar izveidot TLS savienojumu ar [[192.168.1.2]:10050]:
```

Viena puse savienojas ar PSK, bet otra puse izmanto OpenSSL ar atspējotu PSK atbalstu Savienojošās puses žurnālā:

```
...TCP veiksmīgs, nevar izveidot TLS savienojumu ar [[192.168.1.2]:10050]: SSL_connect() iestatīja rezultā
```

Pieņemošās puses žurnālā:

```
...neizdevās pieņemt ienākošu savienojumu: no 192.168.1.2: TLS handshake iestatīja rezultāta kodu uz 1: fi
```

Šifrēšana, izmantojot sertifikātus

Pārskats Zabbix var izmantot RSA sertifikātus PEM formātā, ko parakstījusi publiska vai iekšēja sertifikātu autoritāte (CA).

Sertifikāta pārbaude 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 darbību, sertifikātu pieprasījumu ģenerēšanu un parakstīšanu, kā arī sertifikātu atsaukšanu skatiet tādās pamācībās kā OpenSSL PKI Tutorial v2.0.

Rūpīgi izvērtējiet un pārbaudiet sertifikāta paplašinājumus. Papildu informāciju skatiet sadaļā Ierobežojumi, izmantojot X.509 v3 sertifikāta paplašinājumus.

Sertifikātu konfigurācijas parametri Tālāk norādītie konfigurācijas parametri tiek atbalstīti sertifikātu iestatīšanai Zabbix komponentos.

Parameter	Mandatory	Description
TLSCAFile	yes	Pilns faila ceļš uz failu, kurā ir augstākā līmeņa CA sertifikāti(-s) vienuma sertifikāta verifikācijai. Ja tiek izmantota sertifikātu ķēde ar vairākiem dalībniekiem, sertifikātus sakārtojiet tā, lai zemāka līmeņa CA sertifikāti būtu pirmie, kam seko augstāka līmeņa CA sertifikāti. Vienā failā var iekļaut vairāku CA sertifikātus.
TLSCRLFile	no	Pilns faila ceļš uz failu, kurā ir sertifikātu atsaukšanas saraksti (CRL).
TLSCertFile	yes	Pilns faila ceļš uz failu, kurā ir sertifikāts. Ja tiek izmantota sertifikātu ķēde ar vairākiem dalībniekiem, sertifikātus sakārtojiet tā, lai vispirms būtu servera, starpniekservera vai aģenta sertifikāts, kam seko zemāka līmeņa CA sertifikāti, un beigās augstāka līmeņa CA sertifikāti.
TLSKeyFile	yes	Pilns faila ceļš uz failu, kurā ir privātā atslēga. Nodrošiniet, ka šo failu var lasīt tikai Zabbix lietotājs, iestatot atbilstošas piekļuves tiesības.
TLSServerCertIssuer	no	Atļautais servera sertifikāta izdevējs.
TLSServerCertSubject	no	Atļautais servera sertifikāta subjekts.

Konfigurācijas piemēri Pēc nepieciešamo sertifikātu iestatīšanas konfigurējiet Zabbix komponentus, lai izmantotu uz sertifikātiem balstītu šifrēšanu.

Tālāk ir sniegtas detalizētas konfigurēšanas darbības:

- Zabbix serveris
- Zabbix starpniekserveris
- Zabbix aģents

Zabbix serveris

1. Sagatavojiet CA sertifikāta failu.

Lai pārbaudītu partneru sertifikātus, Zabbix serverim ir jābūt piekļuvei failam, kurā ir augstākā līmeņa pašparakstīti saknes CA sertifikāti. Piemēram, ja ir nepieciešami divu neatkarīgu saknes CA sertifikāti, 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 starpniekservera 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/wUtcVIvWXdMIFJM0Hw==
-----END CERTIFICATE-----
```

Note:
Izmantojiet tikai iepriekš minētos atribūtus gan klienta, gan servera sertifikātiem, lai neietekmētu sertifikāta pārbaudes procesu. Piemēram, OpenSSL var neizdoties izveidot šifrētu savienojumu, ja tiek izmantoti *X509v3 Subject Alternative Name* vai *Netscape Cert Type* paplašinājumi. Plašāku informāciju skatiet sadaļā X.509 v3 sertifikātu paplašinājumu izmantošanas ierobežojumi.

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 vai 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:

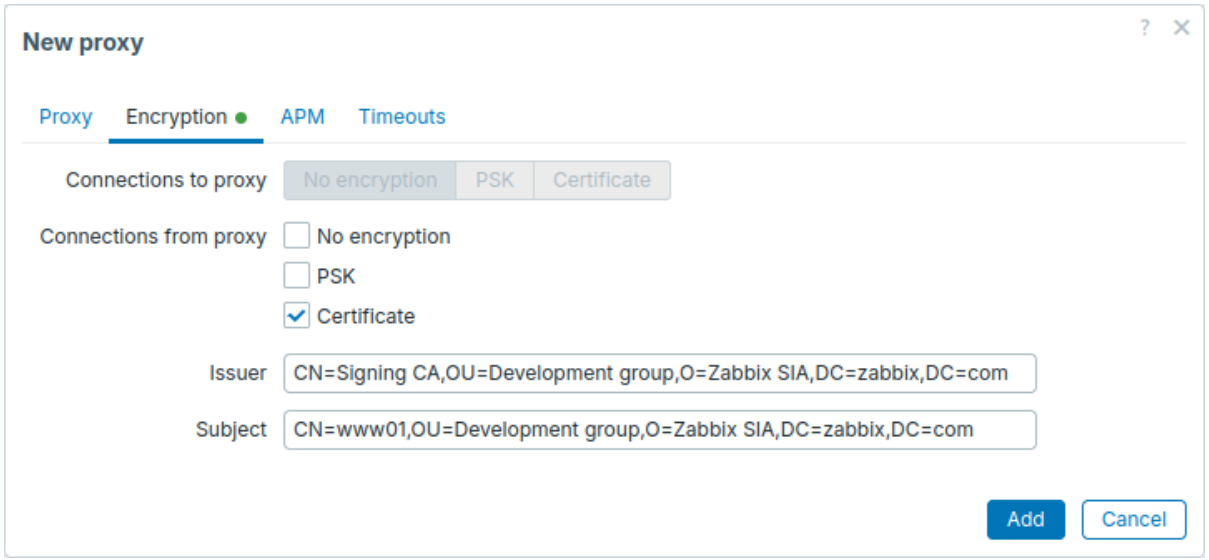
```
TLSConnect=cert
TLSAccept=cert
TLSCAFile=/home/zabbix/zabbix_ca_file.crt
TLSServerCertIssuer=CN=Signing CA,OU=Development group,O=Zabbix SIA,DC=zabbix,DC=com
TLSServerCertSubject=CN=Zabbix server,OU=Development group,O=Zabbix SIA,DC=zabbix,DC=com
TLSCertFile=/home/zabbix/zabbix_proxy.crt
TLSKeyFile=/home/zabbix/zabbix_proxy.key
```

3. Konfigurējiet šī starpniekservera šifrēšanu Zabbix lietotāja saskarnē:

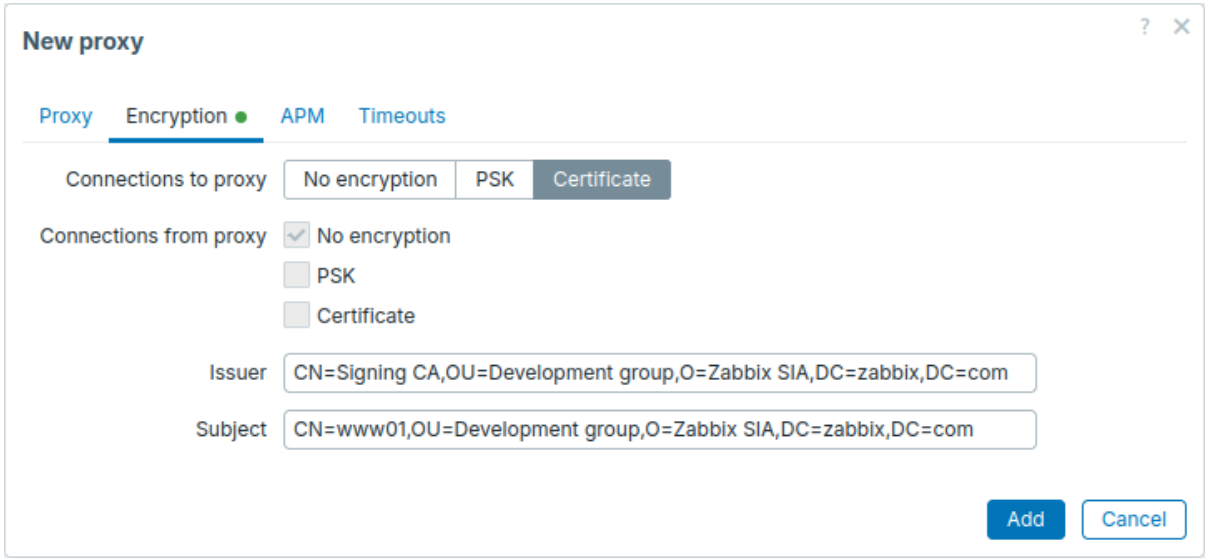
- Dodieties uz: *Administration* > *Proxies*.
- Atlasiet starpniekserveri un noklikšķiniet uz cilnes *Encryption*.

Tālāk sniegtajos piemēros lauki *Issuer* un *Subject* ir aizpildīti. Papildinformāciju par šo lauku izmantošanas iemesliem un veidu 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. Papildinformāciju skatiet sadaļā Atļautā sertifikāta izdevēja un subjekta ierobežošana.

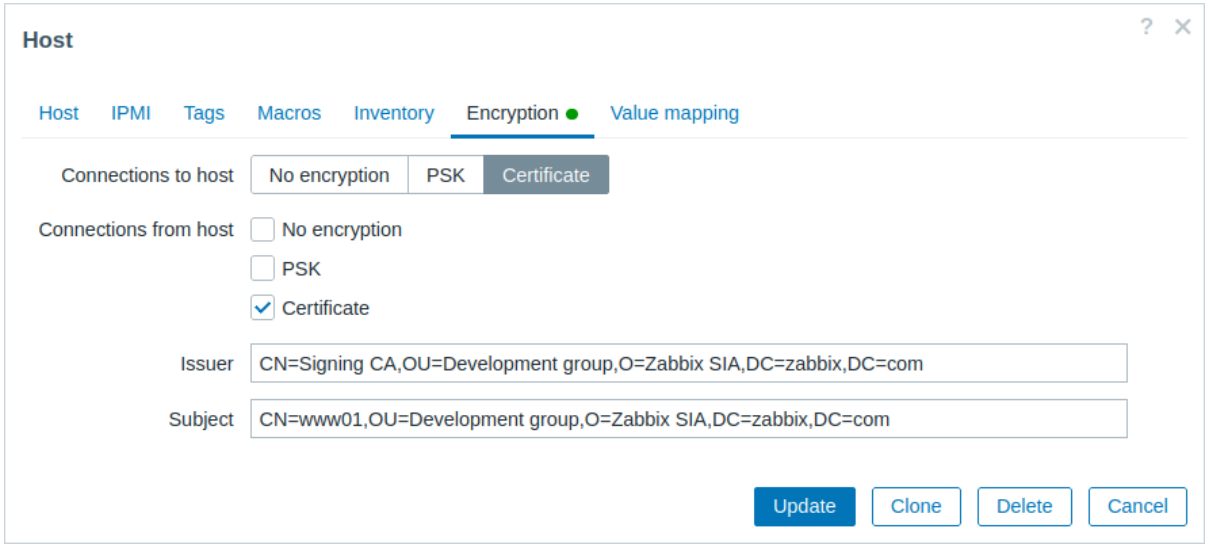
TLS parametri galīgajā aģenta konfigurācijas failā var izskatīties šādi. Ņemiet vērā, ka piemērā tiek pieņemts, ka hosts tiek uzraudzīts, izmantojot starpniekserveri, tāpēc tas ir norādīts kā sertifikāta subjekts:

```
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 apkopošana* > *Hosti*.
- Atlasiet hostu un noklikšķiniet uz cilnes *Šifrēšana*.

Tālāk redzamajā piemērā lauki *Izdevējs* un *Subjekts* ir aizpildīti. Papildinformāciju par šo lauku izmantošanas iemesliem un veidu skatiet sadaļā Atļautā sertifikāta izdevēja un subjekta ierobežošana.



Zabbix tīmekļa pakalpojums

1. Sagatavojiet failus ar augstākā līmeņa CA sertifikātiem, Zabbix tīmekļa pakalpojuma sertifikātu/sertifikātu ķēdi un privāto atslēgu, kā aprakstīts sadaļā Zabbix serveris. Pēc tam attiecīgi rediģējiet parametrus TLSCAFile, TLSCertFile un TLSKeyFile Zabbix tīmekļa pakalpojuma konfigurācijas failā.

2. Rediģējiet papildu TLS parametru Zabbix tīmekļa pakalpojuma konfigurācijas failā: TLSAccept=cert

TLS parametri galīgajā tīmekļa pakalpojuma 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 savienojuma izveidei ar TLS konfigurēto Zabbix tīmekļa pakalpojumu, 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 validē viens otra sertifikātus. Ja partnera sertifikātu ir parakstījusi uzticama CA (ar iepriekš konfigurētu augstākā līmeņa sertifikātu TLSCAFile), tas ir derīgs, nav beidzies tā derīguma termiņš un tas iztur citas pārbaudes, 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, hosts sertifikātu var izmantot, lai uzdotos par serveri). Lai gan tas var būt pieņemami mazās vidēs, kur sertifikātus paraksta īpaša iekšējā CA un uzdošanās par citu

risku ir zems, lielākās vai drošības ziņā jutīgākās vidēs ar to var nepietikt.

Ja jūsu augstākā līmeņa CA izsniedz sertifikātus, kurus Zabbix nevajadzētu pieņemt, vai ja vēlaties samazināt uzdošanās par citu risku, varat ierobežot atļautos sertifikātus, norādot to izdevēju un subjektu.

Piemēram, Zabbix starpniekservera konfigurācijas failā varat 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 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 *tādas, kādas tās ir*, un tām precīzi jāsakrīt.
- Tiek atbalstītas UTF-8 rakstzīmes. Tomēr aizstājējzīmes (*) vai regulārās izteiksmes netiek atbalstītas.
- Tiek 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 restīte ('#', U+0023);
  - virknes beigās: atstarpe (' ', U+0020).
- Nulles rakstzīmes (U+0000) netiek atbalstītas. Ja tiek sastapta nulles rakstzīme, atbilstības pārbaude neizdosies.
- RFC 4517 un RFC 4518 standarti netiek atbalstīti.

Piemēram, ja `Issuer` un `Subject` organizācijas (O) virknēs ir beigu atstarpes 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 nosaka šo lauku "apgriezto" 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
```

Savukārt OpenSSL pēc noklusējuma attēlo `Issuer` un `Subject` virknes no augstākā līmeņa uz zemāko līmeni. Tālākajā piemērā `Issuer` un `Subject` lauki sākas ar augstākā līmeņa (DC) lauku un beidzas ar zemākā līmeņa (CN) lauku. Atstarpju un lauku atdalītāju formatējums arī atšķiras atkarībā no izmantotajām opcijām, tāpēc 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 pareizi noformētu *Issuer* un *Subject* virknes Zabbix vajadzībām, 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, atdalīta ar komatiem un izmantojama Zabbix konfigurācijas failos un lietotāja saskarne:

```
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 sertifikāta paplašinājumu izmantošanas ierobežojumi Ieviešot X.509 v3 sertifikātus Zabbix, daži paplašinājumi var nebūt 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 validēt tikai sertifikāta lauka *Subject* vērtību (skatiet Atļautā sertifikāta izdevēja un Subject ierobežošana). Ja sertifikātos ir iekļauts lauks `subjectAltName`, sertifikāta validācijas rezultāts var atšķirties atkarībā no konkrētajiem kriptogrāfijas rīkkopām, kas izmantotas Zabbix komponentu kompilēšanai. Tāpēc Zabbix var vai nu pieņemt, vai noraidīt sertifikātus, pamatojoties uz šīm kombinācijām.

Extended Key Usage paplašinājums

Zabbix atbalsta *Extended Key Usage* paplašinājumu. Tomēr, ja tas tiek izmantots, parasti ir nepieciešams norādīt gan *clientAuth* (TLS WWW klienta autentifikācijai), gan *serverAuth* (TLS WWW servera autentifikācijai) atribūtus. Piemēram:

- Pasīvajās pārbaudēs, kur Zabbix aģents darbojas kā TLS serveris, aģenta sertifikātā jāiekļauj atribūts *serverAuth*.
- Aktīvajās pārbaudēs, kur aģents darbojas kā TLS klients, aģenta sertifikātā jāiekļauj atribūts *clientAuth*.

Lai gan GnuTLS var izdot brīdinājumu par atslēgas lietojuma pārkāpumiem, tas parasti ļauj saziņai turpināties, neskatoties uz šiem brīdinājumiem.

Name Constraints paplašinājums

Atbalsts *Name Constraints* paplašinājumam atšķiras dažādās kriptogrāfijas rīkkopās. Pārliecinieties, ka jūsu izvēlētā rīkkopa atbalsta šo paplašinājumu. Atkarībā no izmantotās rīkkopas šis paplašinājums var ierobežot Zabbix iespēju ielādēt CA sertifikātus, ja šī sadaļa ir atzīmēta kā kritiska.

Sertifikātu atsaukšanas saraksti (CRL) Ja sertifikāts ir kompromitēts, sertifikātu izsniegšanas iestāde (CA) var to atsaukt, iekļaujot sertifikātu sertifikātu atsaukšanas sarakstā (CRL). CRL tiek pārvaldīti, izmantojot konfigurācijas failus, un tos var norādīt, izmantojot 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 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 tad, kad Zabbix tiek palaists. Lai atjauninātu CRL, restartējiet Zabbix.

> **Attention:**
> Ja Zabbix komponentes ir kompilētas ar OpenSSL un tiek izmantoti CRL, pārliecinieties, ka katram augstākā līmeņa un starpposma CA sertifikātu ķēdēs ir atbilstošs CRL (pat ja tas ir tukšs), kas iekļauts `TLSCRLFile`.

Sertifikātu problēmu novēršana OpenSSL tiek izmantots ar CRL, un kādam CA sertifikātu ķēdē tā CRL nav iekļauta `TLSCRLFile`

TLS servera žurnālā, ja ir *OpenSSL* vienādranga puse:

```
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, ja ir *GnuTLS* vienādranga puse:

```
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 ir beidzies vai 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"
```

Šeit būtība ir tāda, ka ar derīgu CRL atsaukts sertifikāts tiek ziņots kā "certificate revoked". Kad CRL 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, nezināma 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 parakstījusi CA. Issuer un Subject augšējā līmeņa CA sertifikātā ir vienādi, taču servera sertifikātā tie nevar būt vienādi. (Tas pats attiecas uz starpniekservera un aģenta sertifikātiem.)

Lai pārbaudītu, vai sertifikātā ir vienādi Issuer un Subject ieraksti, izpildiet:

```
openssl x509 -in <yourcertificate.crt> -noout -text
```

Saknes (augšējā līmeņa) sertifikātam ir pieļaujamas identiskas Issuer un Subject vērtības.

Šifrēšana, izmantojot iepriekš koplietotas atslēgas

Pārskats Katrs iepriekš koplietotais atslēgas (PSK) ieraksts Zabbix sistēmā faktiski ir divu elementu pāris:

- neslepena PSK identitātes virkne;
- slepena PSK virknes vērtība.

PSK identitātes virkne ir virkne, kas nav tukša un ir kodēta UTF-8 formātā. Piemēram, "PSK ID 001 Zabbix agentd". Tas ir unikāls nosaukums, ar kuru Zabbix komponentes identificē šo konkrēto PSK. Neiekļaujiet sensitīvu informāciju PSK identitātes virknē — tā tiek pārsūtīta tīklā nešifrētā veidā.

PSK vērtība ir grūti uzminama heksadecimālo ciparu virkne, piemēram, e560cb0d918d26d31b4f642181f5f570ad89a390931102e5391d0

Warning:
Nekad neizmantojiet dokumentācijā norādītās PSK vērtības ražošanas vidēs. Tā vietā ģenerējiet unikālu, kriptogrāfiski drošu PSK.

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 heksadecimālie cipari)	2048 biti (256 baitu PSK, ievadīts kā 512 heksadecimālie cipari)

Komponents	PSK identitātes maksimālais izmērs	PSK vērtības minimālais izmērs	PSK vērtības maksimālais izmērs
GnuTLS	128 baiti (var ietvert UTF-8 rakstzīmes)	-	2048 biti (256 baitu PSK, ievadīts kā 512 heksadecimālie cipari)
OpenSSL 1.0.x, 1.1.0	127 baiti (var ietvert UTF-8 rakstzīmes)	-	2048 biti (256 baitu PSK, ievadīts kā 512 heksadecimālie cipari)
OpenSSL 1.1.1	127 baiti (var ietvert UTF-8 rakstzīmes)	-	512 biti (64 baitu PSK, ievadīts kā 128 heksadecimālie cipari)
OpenSSL 1.1.1a un jaunāka versija	127 baiti (var ietvert UTF-8 rakstzīmes)	-	2048 biti (256 baitu PSK, ievadīts kā 512 heksadecimālie cipari)

Attention:
Zabbix lietotāja saskarne ļauj konfigurēt līdz 128 rakstzīmēm garu PSK identitātes virkni un 2048 bitu garu PSK neatkarīgi no izmantotajām kriptogrāfijas bibliotēkām.
 Ja daži Zabbix komponenti atbalsta zemākus ierobežojumus, lietotājs ir atbildīgs par tāda PSK identitātes un vērtības garuma konfigurēšanu, kas ir atļauts šiem komponentiem.
 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 PSK identitāti un PSK vērtību no savas konfigurācijas datnes. Ja abām pusēm ir viena un tā pati PSK identitātes virkne un PSK vērtība, savienojums var tikt izveidots.

Attention:
Katrai PSK identitātei drīkst būt piesaistīta tikai viena vērtība. Lietotājs ir atbildīgs par to, lai nebūtu divu PSK ar vienādu identitātes virkni, bet atšķirīgām vērtībām. Šī nosacījuma 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 tālāk norādītās komandas:

- ar *OpenSSL*:

```
$ openssl rand -hex 32
af8ced32dfe8714e548694e2d29e1a14ba6fa13f216cb35c19d0feb1084b0429
```

- ar *GnuTLS*:

```
$ psktool -u psk_identity -p database.psk -s 32
Generating a random key for user 'psk_identity'
Key stored to database.psk

$ cat database.psk
psk_identity:9b8eafedfaae00cece62e85d5f4792c7d9c9bcc851b23216a1d300311cc4f7cb
```

Ņemiet vērā, ka iepriekš izmantotā komanda "psktool" ģenerē datubāzes failu ar PSK identitāti un tai atbilstošo PSK. Zabbix PSK failā sagaida tikai PSK, tāpēc identitātes virkne un kols (':') no faila ir jānoņem.

PSK konfigurēšana servera un aģenta saziņai (piemērs) Aģenta hostā ierakstiet iepriekš ģenerēto PSK vērtību failā, piemēram, `/home/zabbix/zabbix_agentd.psk`. Faila pirmajā rindā jābūt PSK vērtībai.

Iestatiet piekļuves tiesības PSK failam — to drīkst nolasīt tikai Zabbix lietotājs.

Rediģējiet TLS parametrus Zabbix 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:

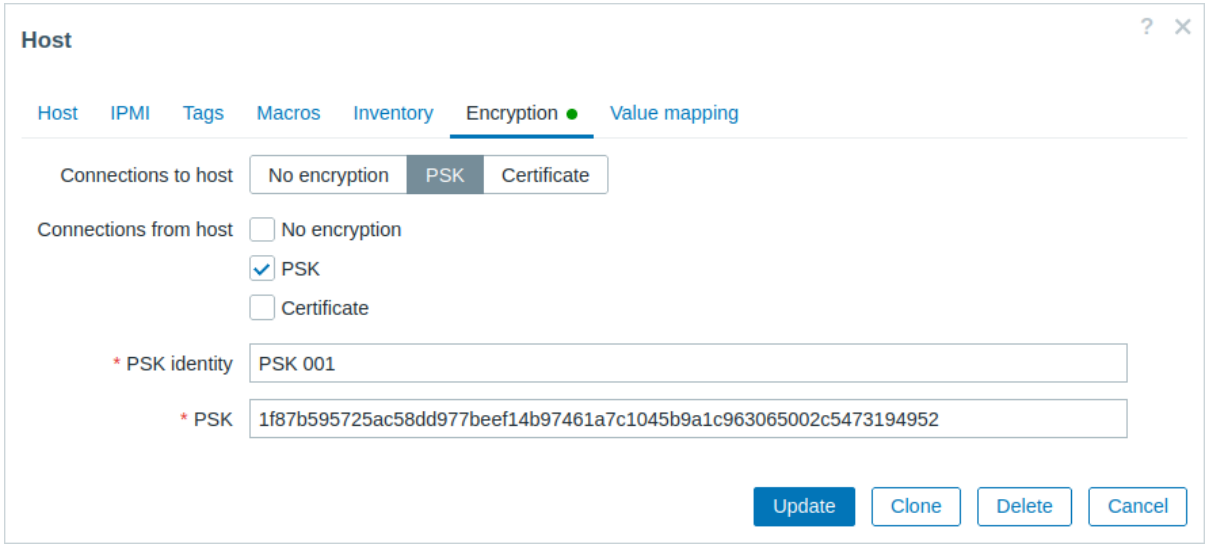
```
zabbix_get -s 127.0.0.1 -k "system.cpu.load[all,avg1]" --tls-connect=psk --tls-psk-identity="PSK 001" --tl
```

(Lai samazinātu dīkstāvi, skatiet, kā mainīt savienojuma tipu sadaļā Savienojuma šifrēšanas pārvaldība.)

Konfigurējiet PSK šifrēšanu šim aģentam Zabbix lietotāja saskarnē:

1. Dodieties uz: *Datu apkopošana* > *Hosti*.
2. Atlasiet hostu un noklikšķiniet uz cilnes *Šifrēšana*.
3. Konfigurējiet PSK šifrēšanas informāciju.

Piemērs:



Visi obligātie ievades lauki ir atzīmēti ar sarkanu zvaigznīti.

Kad konfigurācijas kešatmiņa ir sinhronizēta ar datubāzi, jaunajos savienojumos tiks izmantots 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`. Faila pirmajā teksta rindā jābūt PSK vērtībai, piemēram:

```
e560cb0d918d26d31b4f642181f5f570ad89a390931102e5391d08327ba434e9
```

Iestatiet piekļuves tiesības PSK failam - to drīkst nolasīt tikai Zabbix lietotājs.

Rediģējiet TLS parametrus Zabbix 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 tipu sadaļā Savienojuma šifrēšanas pārvaldība.)

Konfigurējiet PSK šim starpniekserverim Zabbix lietotāja saskarnē. Atveriet *Administration* > *Proxies*, atlasiet starpniekserveri un atveriet cilni *Encryption*. Sadaļā "Connections from proxy" atzīmējiet PSK. Laukā "PSK identity" ielīmējiet "PSK 002", bet laukā "PSK" ielīmējiet "e560cb0d918d26d31b4f642181f5f570ad89a390931102e5391d08327ba434e9". 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 - Zabbix starpniekservera konfigurācijas failā jāiestata `TLSAccept=psk`, bet Zabbix lietotāja saskarnē sadaļai *Connections to proxy* jāiestata PSK.

Problēmu novēršana ar PSK problēmām PSK satur nepāra skaitu heksadecimālo ciparu

Starpniekserveris vai aģents netiek palaists, ziņojums starpniekservera vai aģenta žurnālā:

```
nederīgs PSK failā "/home/zabbix/zabbix_proxy.psk"
```

PSK identitātes virkne, kas ir garāka par 128 baitiem, tiek nodota GnuTLS

TLS klienta puses žurnālā:

```
gnutls_handshake() failed: -110 The TLS connection was non-properly terminated.
```

TLS servera puses žurnālā.

```
gnutls_handshake() failed: -90 The SRP username supplied is illegal.
```

Pārāk gara PSK vērtība, izmantojot OpenSSL 1.1.1

Savienojošās puses žurnālā:

```
...OpenSSL bibliotēka (versija OpenSSL 1.1.1  11 Sep 2018) inicializēta
...
...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:

```
...Ziņojumam no 123.123.123.123 trūkst galvenes. Ziņojums ignorēts.
```

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, ievadīts kā 128 heksadecimāli cipari).

Skatiet arī: Vērtības lieluma ierobežojumi

Drošs savienojums ar datubāzi

Pārskats

Šī sadaļa sniedz Zabbix iestatīšanas soļus un konfigurācijas piemērus drošiem TLS savienojumiem starp:

Database	Zabbix components
MySQL	Zabbix lietotāja saskarne, Zabbix serveris, Zabbix starpniekserveris
PostgreSQL	Zabbix lietotāja saskarne, Zabbix serveris, Zabbix starpniekserveris

Lai iestatītu savienojuma šifrēšanu DBMS ietvaros, skatiet oficiālo piegādātāja dokumentāciju, lai iegūtu detalizētu informāciju:

- MySQL: avota un replikas replikācijas datubāzes serveri.
- MySQL: grupas replikācijas u. c. datubāzes serveri.
- PostgreSQL šifrēšanas opcijas.

Visi piemēri ir balstīti uz MySQL CE (8.0) un PgSQL (13) GA laidieniem, kas pieejami oficiālajās repozitorijās, izmantojot CentOS 8.

Prasības

Lai iestatītu šifrēšanu, ir nepieciešams:

- Izstrādātāja atbalstīta operētājsistēma ar OpenSSL >=1.1.X vai alternatīvu.

Note:
Ieteicams izvairīties no operētājsistēmām, kurām ir beidzies atbalsta termiņš, īpaši jaunu instalāciju gadījumā

- Datu bāzes dzinējs (RDBMS), kas ir instalēts un uzturēts no izstrādātāja nodrošinātās oficiālās repozitorijas. Operētājsistēmas bieži tiek piegādātas ar novecojušām datu bāzes programmatūras versijām, kurām šifrēšanas atbalsts nav ieviests, piemēram, uz RHEL 7 balstītas sistēmas un PostgreSQL 9.2, MariaDB 5.5 bez šifrēšanas atbalsta.

Terminoloģija

Šīs opcijas iestatīšana uzliek par pienākumu 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, ka datubāzes identitāte (CN), kas norādīta ar DBHost, atbilst tās sertifikātam

Zabbix konfigurācija

Lietotāja saskarne un datubāze

Drošu savienojumu ar datubāzi var konfigurēt lietotāja saskarnes instalēšanas laikā:

- Lai iespējotu transporta šifrēšanu, darbībā Configure DB connection atzīmējiet izvēles rūtiņu *Database TLS encryption*.
- Lai iespējotu šifrēšanu ar sertifikātiem, atzīmējiet izvēles rūtiņu *Verify database certificate*, kas tiek parādīta, kad ir atzīmēts lauks *TLS encryption*.

Note:
MySQL gadījumā izvēles rūtiņa *Database TLS encryption* ir atspējota, ja *Database host* ir iestatīts uz localhost, jo savienojumu, kas izmanto ligzdas failu (Unix sistēmā) vai koplietojamo atmiņu (Windows sistēmā), nevar šifrēt.
 PostgreSQL gadījumā izvēles rūtiņa *TLS encryption* ir atspējota, ja lauka *Database host* vērtība sākas ar slīpsvītru vai lauks ir tukšs.

TLS šifrēšanas ar sertifikātiem režīmā kļūst pieejami šādi parametri (ja ir atzīmētas abas izvēles rūtiņas):

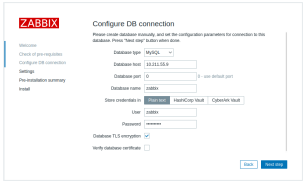
Parameter	Description
Database TLS CA file	Norādiet derīga TLS sertificēšanas centra (CA) faila pilno ceļu.
Database TLS key file	Norādiet derīga TLS atslēgas faila pilno ceļu.
Database TLS certificate file	Norādiet derīga TLS sertifikāta faila pilno ceļu.
Database host verification	Atzīmējiet šo izvēles rūtiņu, lai aktivizētu hosts pārbaudi. MySQL gadījumā atspējots, jo PHP MySQL bibliotēka neļauj izlaist partnera sertifikāta validācijas darbību.
Database TLS cipher list	Norādiet pielāgotu derīgu šifru sarakstu. Šifru saraksta formātam jāatbilst OpenSSL standartam. Pieejams tikai MySQL.

Attention:
TLS parametriem jānorāda derīgi faili. Ja tie norāda uz neesošiem vai nederīgiem failiem, tiks parādīta autorizācijas kļūda.
 Ja sertifikātu faili ir rakstāmi, lietotāja saskarne System information pārskatā ģenerē brīdinājumu: "TLS certificate files must be read-only." (tiek parādīts tikai tad, ja PHP lietotājs ir sertifikāta īpašnieks).
Ar parolēm aizsargāti sertifikāti netiek atbalstīti.

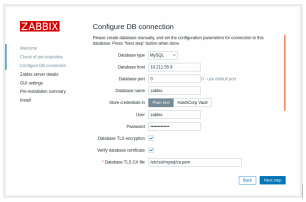
Lietošanas gadījumi

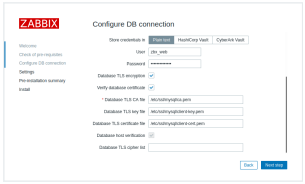
Zabbix lietotāja saskarne izmanto GUI saskarni, lai definētu iespējamās opcijas: `required`, `verify_ca`, `verify_full`. Norādiet nepieciešamās opcijas instalēšanas vedņa darbībā *Configure DB connections*. Šīs opcijas tiek kartētas uz lietotāja saskarnes konfigurācijas failu (`zabbix.conf.php`) šādi:

GUI iestatījumi	Konfigurācijas fails	Apraksts	Rezultāts

	...
	// Used for TLS connection.
	`$DB['ENCRYPTION'] = true;`
	`$DB['KEY_FILE'] = '';`
	`$DB['CERT_FILE'] = '';`
	`$DB['CA_FILE'] = '';`
	`$DB['VERIFY_HOST'] = false;`
	`$DB['CIPHER_LIST'] = '';`
	...
		1. Atzīmējiet *Database TLS encryption*
		2. Neatzīmējiet *Verify database certificate*
			Iespējo `required` režīmu.

GUI iestatījumi	Konfigurācijas fails	Apraksts	Rezultāts

	...
	`$DB['ENCRYPTION'] = true;`
	`$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 *Database TLS CA file* ceļu
			Iespējo `verify_ca` režīmu.

	...
	// Used for TLS connection with strictly defined Cipher list.
	`$DB['ENCRYPTION'] = true;`
	`$DB['KEY_FILE'] = '<key_file_path>';`
	`$DB['CERT_FILE'] = '<key_file_path>';`
	`$DB['CA_FILE'] = '<key_file_path>';`
	`$DB['VERIFY_HOST'] = true;`
	`$DB['CIPHER_LIST'] = '<cipher_list>';`
	...
	
	Vai:
	
	...
	// Used for TLS connection without Cipher list defined - selected by MySQL server
	`$DB['ENCRYPTION'] = true;`
	`$DB['KEY_FILE'] = '<key_file_path>';`
	`$DB['CERT_FILE'] = '<key_file_path>';`
	`$DB['CA_FILE'] = '<key_file_path>';`
	`$DB['VERIFY_HOST'] = true;`
	`$DB['CIPHER_LIST'] = '';`
	...
		1. Atzīmējiet *Database TLS encryption* un *Verify database certificate*
		2. Norādiet *Database TLS key file* ceļu
		3. Norādiet *Database TLS CA file* ceļu
		4. Norādiet *Database TLS certificate file* ceļu
		5. Norādiet *Database TLS cipher list* (neobligāti)
			Iespējo `verify_full` režīmu MySQL.

GUI iestatījumi	Konfigurācijas fails	Apraksts	Rezultāts

	...
	`$DB['ENCRYPTION'] = true;`
	`$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 *Database TLS key file* ceļu
		3. Norādiet *Database TLS CA file* ceļu
		4. Norādiet *Database TLS certificate file* ceļu
		5. Atzīmējiet *Database host verification*
			Iespējo `verify_full` režīmu PostgreSQL.

Skatiet arī: MySQL šifrēšanas konfigurācijas piemēri, PostgreSQL šifrēšanas konfigurācijas piemēri.

Zabbix servera/starpniekservera konfigurācija

Drošus savienojumus ar datubāzi var konfigurēt, izmantojot attiecīgos parametrus Zabbix servera un/vai starpniekservera konfigurācijas failā.

Konfigurācija	Rezultāts
Nav	Savienojums ar datubāzi bez šifrēšanas.
Iestatiet `DBTLSConnect=required`	Serveris/starpniekserveris izveido TLS savienojumu ar datubāzi. Nešifrēts savienojums nav atļauts.
1. Iestatiet `DBTLSConnect=verify_ca` 2. Iestatiet `DBTLSCAFile` - norādiet TLS sertificēšanas iestādes failu	Serveris/starpniekserveris izveido TLS savienojumu ar datubāzi pēc datubāzes sertifikāta pārbaudes.
1. Iestatiet `DBTLSConnect=verify_full` 2. Iestatiet `DBTLSCAFile` - norādiet TLS sertificēšanas iestādes failu	Serveris/starpniekserveris izveido TLS savienojumu ar datubāzi pēc datubāzes sertifikāta un datubāzes resursdatora identitātes pārbaudes.
1. Iestatiet `DBTLSCAFile` - norādiet TLS sertificēšanas 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.
Iestatiet `DBTLSCipher` - šifrēšanas algoritmu sarakstu, ko klients atļauj savienojumiem, izmantojot TLS protokolus līdz TLS 1.2 vai `DBTLSCipher13` - šifrēšanas algoritmu sarakstu, ko klients atļauj savienojumiem, izmantojot TLS 1.3 protokolu	(MySQL) TLS savienojums tiek izveidots, izmantojot algoritmu no norādītā saraksta. (PostgreSQL) Šīs opcijas iestatīšana tiks uzskatīta par kļūdu.

MySQL šifrēšanas konfigurācija

Pārskats

Šajā sadaļā ir sniegti vairāki šifrēšanas konfigurācijas piemēri CentOS 8.2 un MySQL 8.4.0, un to var izmantot kā ātru darba sākšanas rokasgrāmatu savienojuma ar datubāzi šifrēšanai.

Attention:
Ja MySQL hosts ir iestatīts uz localhost, šifrēšanas opcijas nebūs pieejamas. Šādā gadījumā savienojums starp Zabbix lietotāja saskarni un datubāzi izmanto ligzdas failu (Unix sistēmās) vai koplietojamo atmiņu (Windows sistēmās), un to nevar šifrēt.

Note:
Šifrēšanas kombināciju saraksts neaprobežojas tikai ar šajā lapā uzskaitītajām. Ir pieejams daudz vairāk kombināciju.

Priekšnosacījumi

Instalējiet MySQL datubāzi no oficiālā repozitorija.

Skatiet MySQL dokumentāciju, lai iegūtu informāciju par MySQL repozitorija lietošanu.

MySQL serveris ir gatavs pieņemt drošus savienojumus, izmantojot pašparakstītu sertifikātu.

Lai redzētu, kuri lietotāji izmanto šifrētu savienojumu, izpildiet šādu vaicājumu (Performance Schema jābūt ieslēgtam):

```
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

Mūsdienīgas datubāzes versijas ir gatavas lietošanai ar `required` šifrēšanas režīmu. Pēc sākotnējās iestatīšanas un palaišanas tiks izveidots servera puses sertifikāts.

Izveidojiet lietotājus un lomas galvenajiem komponentiem (aizstājiet <strong_password> ar drošu, unikālu paroli):

```
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 X.509 protokols netiek izmantots identitātes pārbaudei, taču lietotājs ir konfigurēts tā, lai izmantotu tikai šifrētus savienojumus. Plašāku informāciju par lietotāju konfigurēšanu skatiet MySQL dokumentācijā.

Palaidiet šo komandu, lai pārbaudītu savienojumu (ligzdas savienojumu nevar izmantot drošu savienojumu pārbaudei):

```
mysql -u zbx_srv -p -h 10.211.55.9 --ssl-mode=REQUIRED
```

Pārbaudiet pašreizējo statusu un pieejamās šifrēšanas komplektācijas:

```
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
```

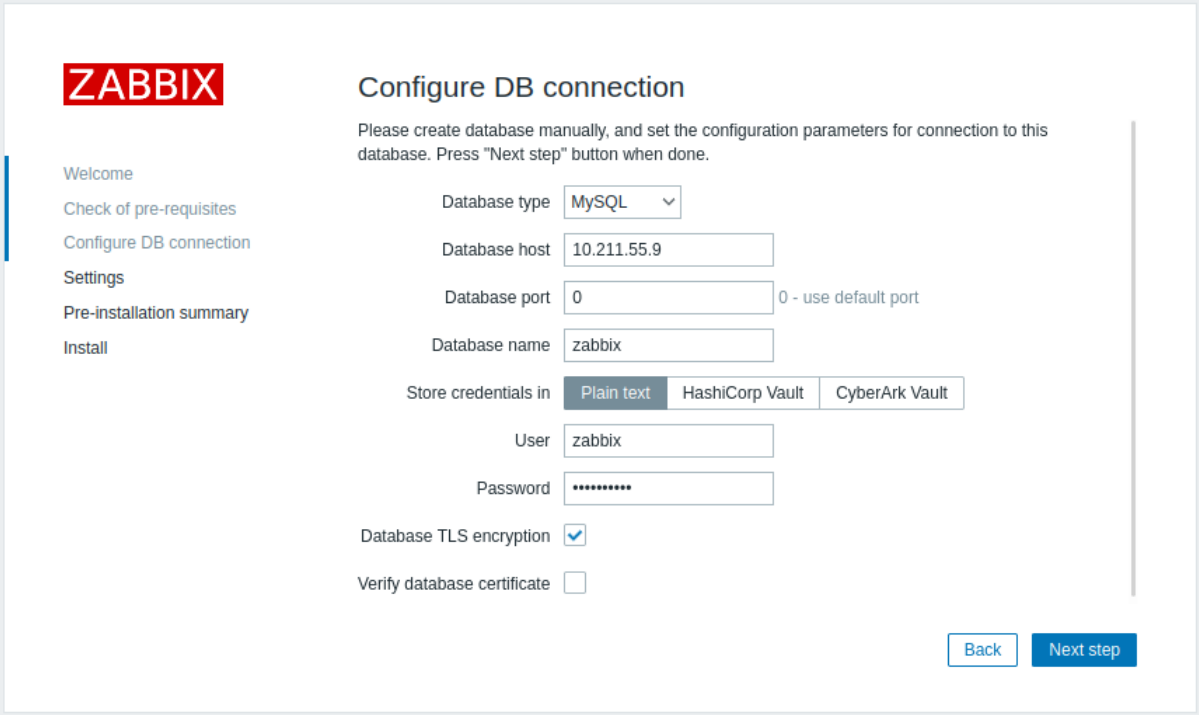
```
mysql> SHOW SESSION STATUS LIKE 'Ssl_cipher_list'\G;
*************************** 1. row ***************************
Variable_name: Ssl_cipher_list
Value: TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256:TLS_AES_128_CCM_SHA256:E
1 row in set (0.00 sec)

ERROR:
No query specified
```

Lietotāja saskarne

Lai iespējotu tikai transporta šifrēšanu savienojumiem starp Zabbix lietotāja saskarni un datubāzi:

- Atzīmējiet *Database TLS encryption*
- Neatzīmējiet *Verify database certificate*



Serveris

Lai iespējotu tikai transporta līmeņa šifrēšanu savienojumiem starp serveri un datubāzi, rediģējiet servera konfigurācijas failu:

```
...
DBHost=10.211.55.9
DBName=zabbix
DBUser=zbx_srv
DBPassword=<strong_password>
DBTLSConnect=required
...
```

Šifrēšana ar sertifikātu iestādes verifikāciju

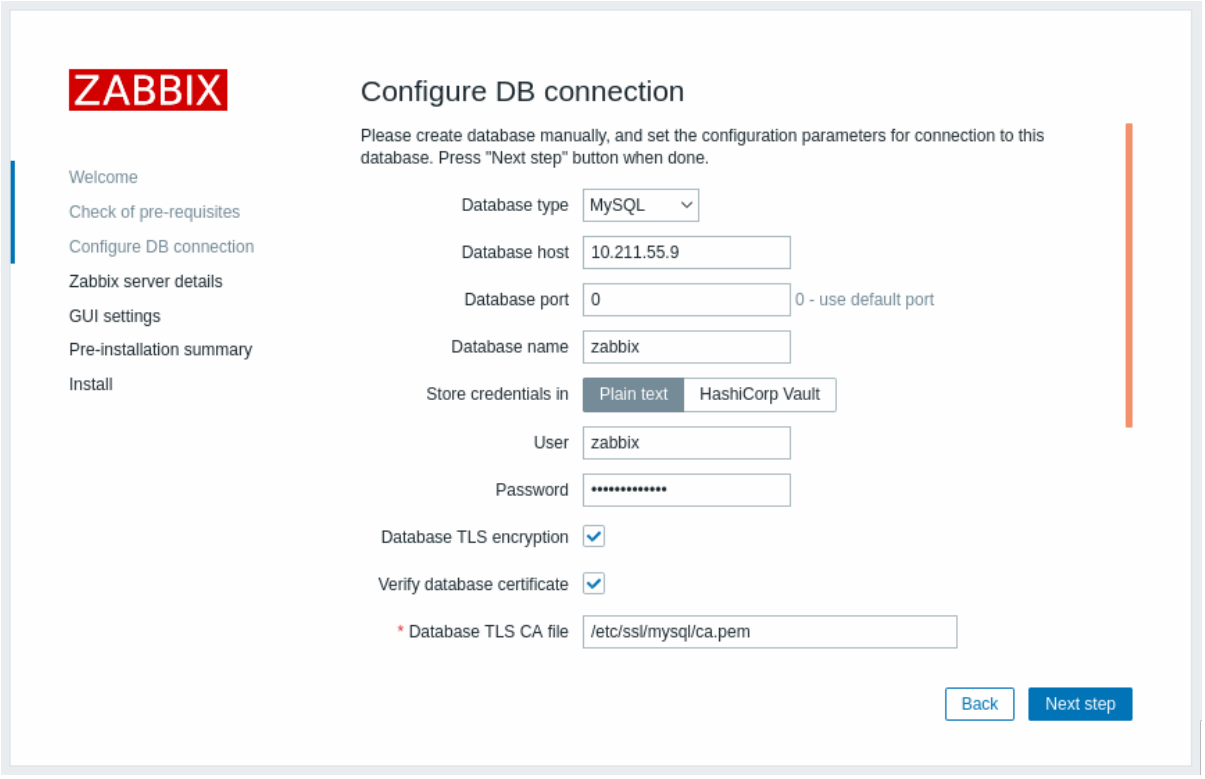
Kopējiet nepieciešamo MySQL CA uz Zabbix lietotāja saskarnes serveri un piešķiriet atbilstošas atļaujas, lai tīmekļa serveris varētu nolasīt šo failu.

Note:
Šis režīms nedarbojas RHEL 7 dēļ vecākām MySQL bibliotēkām.

Lietotāja saskarne

Lai iespējotu šifrēšanu ar sertifikāta pārbaudi savienojumiem starp Zabbix lietotāja saskarni un datubāzi:

- Atzīmējiet *Datubāzes TLS šifrēšana* un *Pārbaudīt datubāzes sertifikātu*
- Norādiet ceļu uz datubāzes TLS CA failu



Varat to iestatīt arī lietotāja saskarnes konfigurācijas failā (`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'] = '';
...
```

Lai novērstu problēmas, izmantojiet komandrindas rīku, lai pārbaudītu, vai nepieciešamajam lietotājam ir iespējams izveidot savienojumu:

```
    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 pārbaudi savienojumiem starp Zabbix serveri un datubāzi, rediģējiet servera konfigurācijas failu:

```
...
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*) uz:

```
[mysqld]
...
#### šajā piemērā 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 ir jāizveido manuāli saskaņā ar MySQL CE dokumentāciju: Creating SSL and RSA certificates and keys using MySQL vai Creating SSL certificates and keys using openssl

Attention:
MySQL servera sertifikātā laukam Common Name jābūt iestatītam uz FQDN nosaukumu, jo Zabbix lietotāja saskarne izmantos DNS nosaukumu, lai sazinātos 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
```

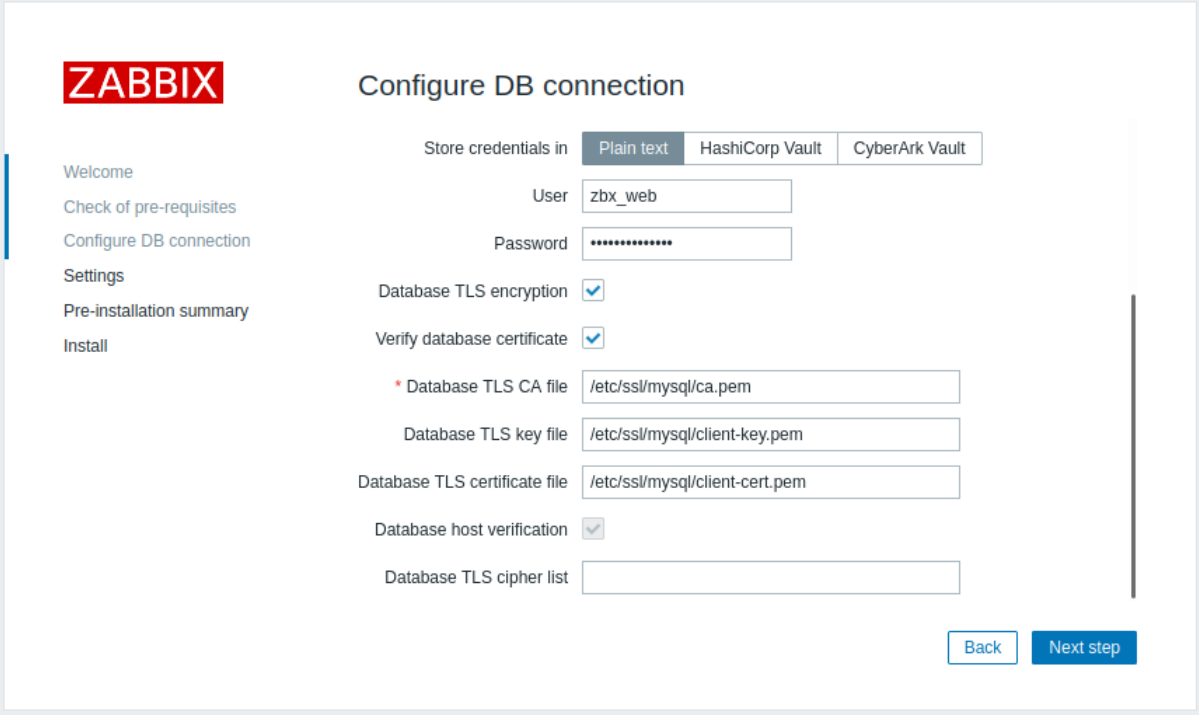
Lietotāja saskarne

Lai iespējotu šifrēšanu ar pilnu pārbaudi savienojumiem starp Zabbix lietotāja saskarni un datubāzi:

- Atzīmējiet *Database TLS encryption* un *Verify database certificate*
- Norādiet *Database TLS key file* ceļu
- Norādiet *Database TLS CA file* ceļu
- Norādiet *Database TLS certificate file* ceļu

Ņemiet vērā, ka *Database host verification* ir atzīmēts un neaktīvs - šo darbību MySQL gadījumā nevar izlaist.

Warning:
Ja lauks *Database TLS cipher list* ir atstāts tukšs, tiks iespējoti kopīgie šifri, kurus atļauj gan lietotāja saskarne (klients), gan serveris. Alternatīvi šifrus var iestatīt tieši, ievērojot šifru konfigurēšanas prasības.



To var iestatīt arī lietotāja saskarnes konfigurācijas failā (`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, rediģējiet servera konfigurācijas failu:

```
...
DBHost=10.211.55.9
DBName=zabbix
DBUser=zbx_srv
DBPassword=<strong_password>
DBTLSConnect=verify_full
DBTLSCAFile=/etc/ssl/mysql/ca.pem
DBTLSCertFile=/etc/ssl/mysql/client-cert.pem
DBTLSKeyFile=/etc/ssl/mysql/client-key.pem
...
```

PostgreSQL šifrēšanas konfigurācija

Pārskats

Šajā sadaļā ir sniegti vairāki šifrēšanas konfigurācijas piemēri CentOS 8.2 un PostgreSQL 13.

> **Note:**
> Savienojumu starp Zabbix lietotāja saskarni 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 dokumentācijā sniegtajām instrukcijām par sertifikāta sagatavošanu ar postgresql.conf, kā arī par lietotāju piekļuves kontroli caur `ph_hba.conf`.

Pēc noklusējuma PostgreSQL ligzda ir piesaistīta localhost, lai tīkla attālinātie savienojumi varētu klausīties uz reālā tīkla interfeisa.

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

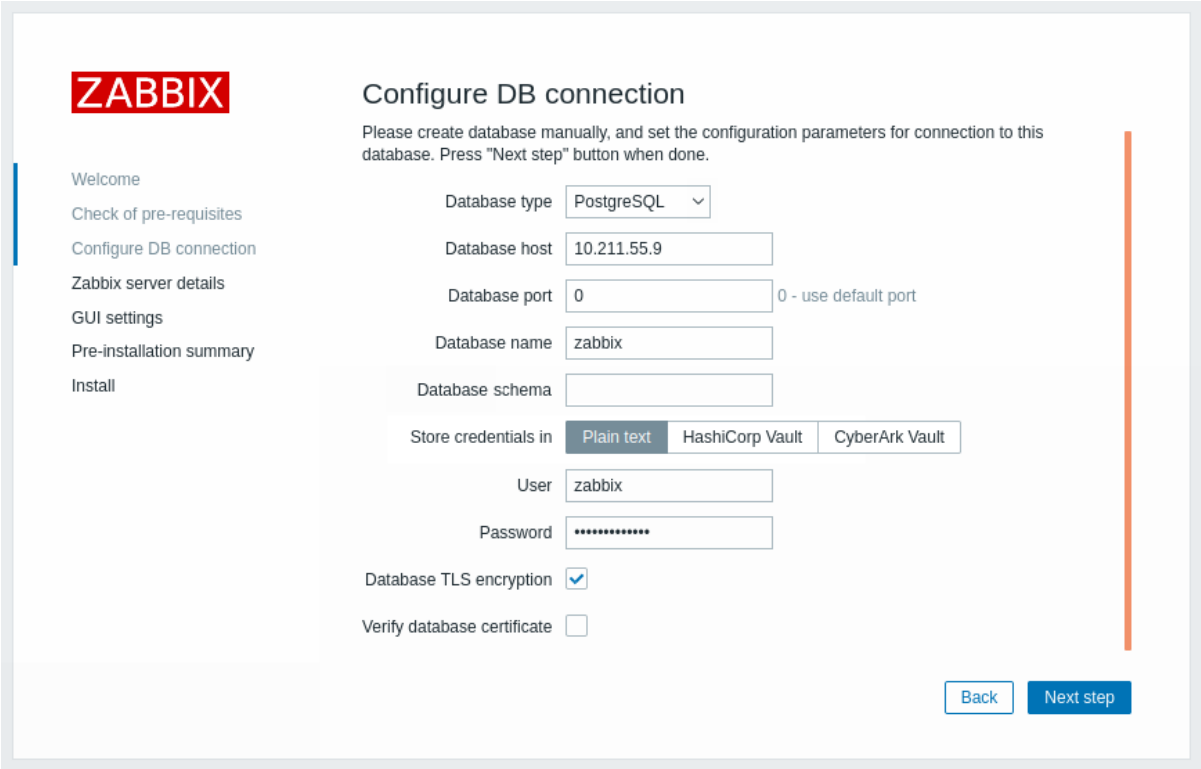
### verify full
hostssl all all 0.0.0.0/0 md5 clientcert=verify-full
...
```

Tikai transporta šifrēšana

Lietotāja saskarne

Lai iespējotu tikai transporta šifrēšanu savienojumiem starp Zabbix lietotāja saskarni un datubāzi:

- Atzīmējiet *Database TLS encryption*
- Neatzīmējiet *Verify database certificate*



Serveris

Lai iespējotu tikai transporta līmeņa šifrēšanu savienojumiem starp serveri un datubāzi, rediģējiet servera konfigurācijas failu:

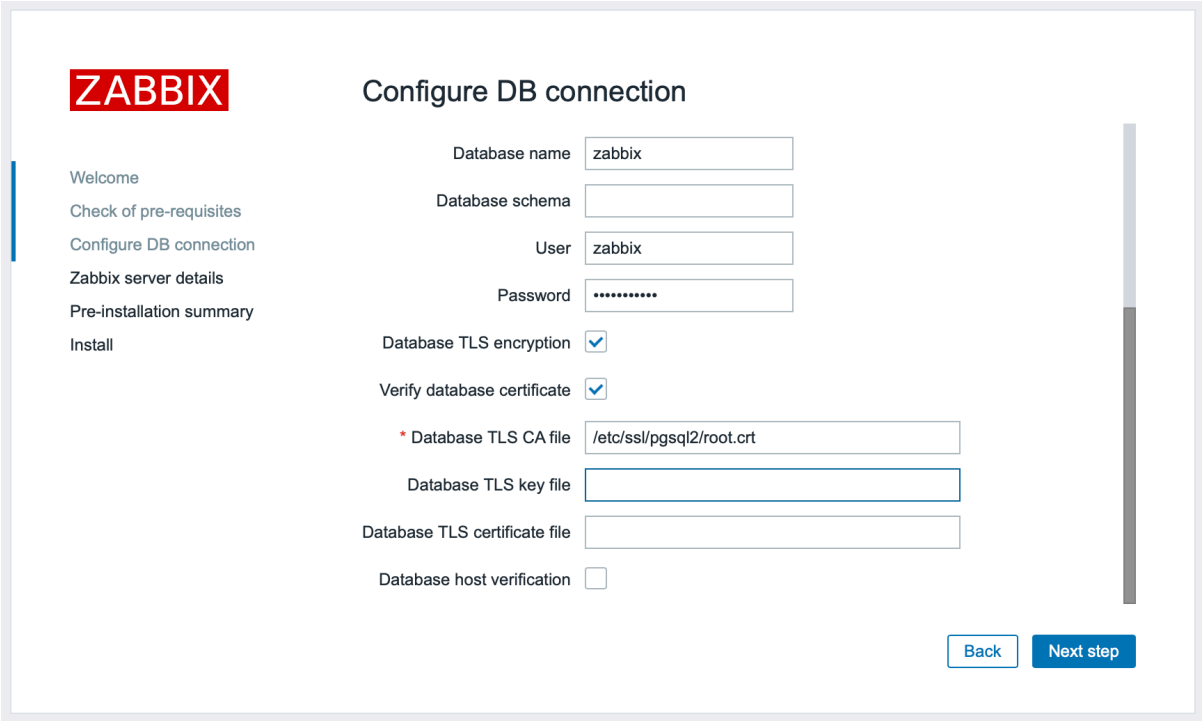
```
...
DBHost=10.211.55.9
DBName=zabbix
DBUser=zbx_srv
DBPassword=<strong_password>
DBTLSConnect=required
...
```

Šifrēšana ar sertifikātu iestādes verifikāciju

Lietotāja saskarne

Lai iespējotu šifrēšanu ar sertificēšanas iestādes pārbaudi savienojumiem starp Zabbix lietotāja saskarni un datubāzi:

- Atzīmējiet *Datubāzes TLS šifrēšana* un *Pārbaudīt datubāzes sertifikātu*
- Norādiet *Datubāzes TLS CA faila* ceļu



To var iestatīt arī lietotāja saskarnes konfigurācijas failā (`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 pārbaudi savienojumiem starp Zabbix serveri un datubāzi, rediģējiet servera konfigurācijas failu (aizstājiet <strong_password> ar spēcīgu, unikālu paroli):

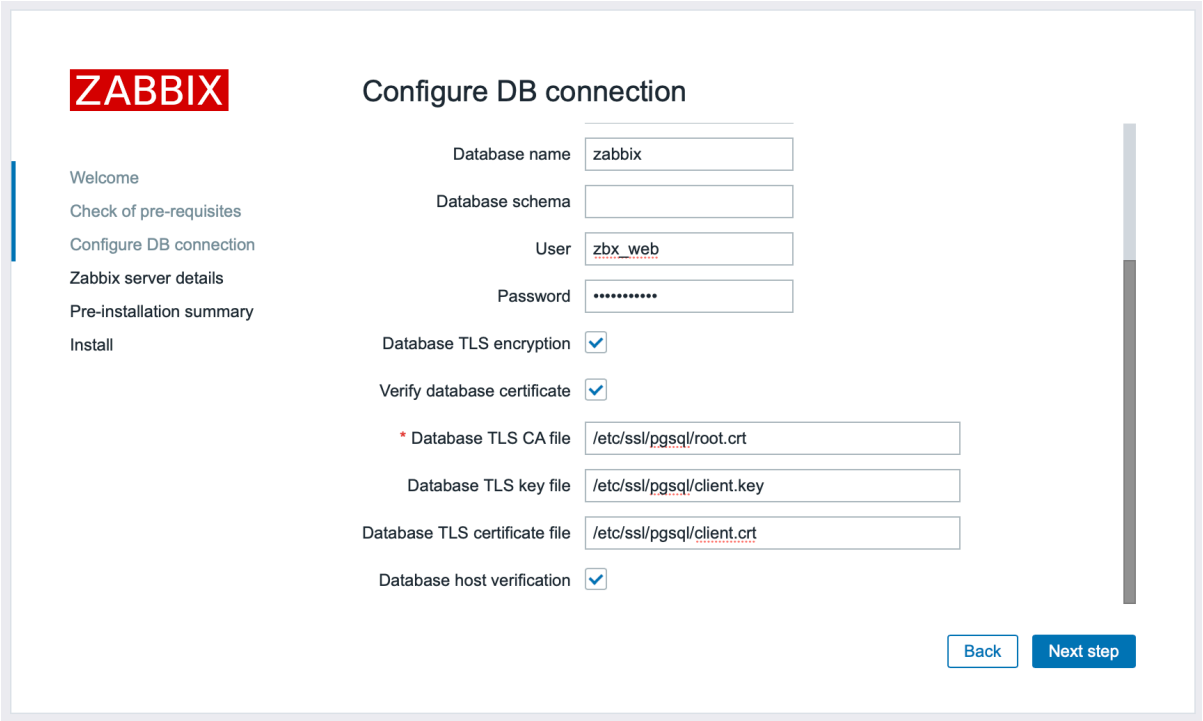
```
...
DBHost=10.211.55.9
DBName=zabbix
DBUser=zbx_srv
DBPassword=<strong_password>
DBTLSConnect=verify_ca
DBTLSCAFile=/etc/ssl/pgsql/root.crt
...
```

Šifrēšana ar pilnu verifikāciju

Lietotāja saskarne

Lai iespējotu šifrēšanu ar sertifikātu un datubāzes hosts identitātes pārbaudi savienojumiem starp Zabbix lietotāja saskarni un datubāzi:

- Atzīmējiet *Database TLS encryption* un *Verify database certificate*
- Norādiet *Database TLS key file* ceļu
- Norādiet *Database TLS CA file* ceļu
- Norādiet *Database TLS certificate file* ceļu
- Atzīmējiet *Database host verification*



Varat to iestatīt arī lietotāja saskarnes konfigurācijas failā (`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 resursdatora identitātes pārbaudi savienojumiem starp Zabbix serveri un datubāzi, rediģējiet servera konfigurācijas failu:

```
...
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
...
```

Drošs savienojums starp lietotāja saskarni un serveri

Pārskats

Šajā sadaļā ir sniegtas Zabbix iestatīšanas darbības un konfigurācijas piemēri drošiem TLS savienojumiem starp Zabbix lietotāja saskarni un Zabbix serveri.

Konfigurēšana

Pēc noklusējuma saziņa starp Zabbix lietotāja saskarni un Zabbix serveri nav šifrēta. Lai uzlabotu drošību, iespējojiet TLS abās pusēs. Tālāk ir sniegts piemērs, kā to izdarīt visvienkāršākajā veidā.

1. Ģenerējiet sertifikātus un atslēgas.

Izveidojiet darba direktoriju:

```
sudo mkdir -p /etc/zabbix/ssl && cd /etc/zabbix/ssl
```

Izveidojiet CA sertifikātu (pielāgojiet `MyZabbixCA` vērtību 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 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 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 izplatījuma 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 izplatījuma 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.

Servera konfigurācijas failā (`zabbix_server.conf`) pievienojiet:

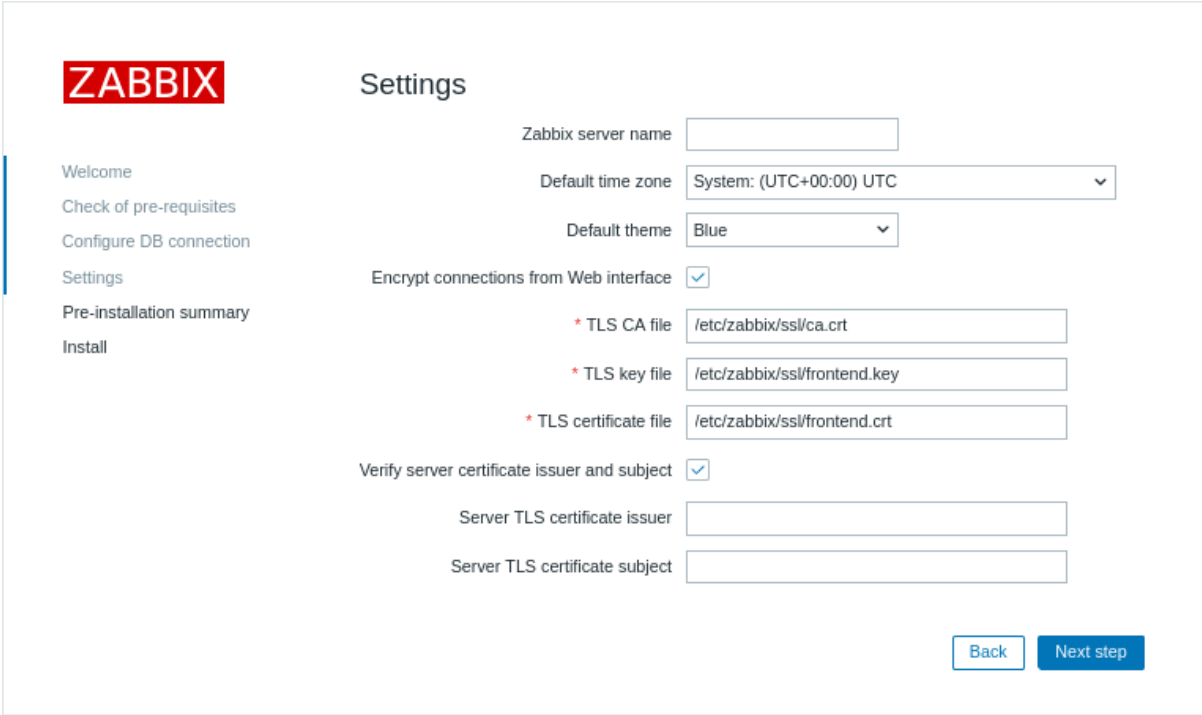
```
TLSFrontendAccept=cert
TLSCertFile=/etc/zabbix/ssl/server.crt
TLSKeyFile=/etc/zabbix/ssl/server.key
TLSCAFile=/etc/zabbix/ssl/ca.crt
### Pēc izvēles:
### TLSFrontendCertIssuer=CN=MyZabbixCA
### TLSFrontendCertSubject=CN=zabbix-frontend-node
```

Pēc tam restartējiet serveri:

```
sudo systemctl restart zabbix-server
```

4. Konfigurējiet Zabbix lietotāja saskarni.

Web saskarnes instalēšanas laikā iespējojiet opciju *Šifrēt savienojumus no web saskarnes* (un, ja nepieciešams, opciju *Pārbaudīt servera sertifikāta izdevēju un subjektu*) un aizpildiet laukus *TLS CA fails*, *TLS atslēgas fails*, *TLS sertifikāta fails* (un, ja nepieciešams, laukus *Servera TLS sertifikāta izdevējs* un *Servera TLS sertifikāta subjekts*):



Parametrs	Apraksts
TLS CA fails	Norādiet pilnu ceļu uz sertificēšanas iestādes (CA) sertifikāta failu, kas tiek izmantots servera sertifikāta pārbaudei.
TLS atslēgas fails	Norādiet pilnu ceļu uz klienta privātās atslēgas failu, kas atbilst klienta sertifikātam.
TLS sertifikāta fails	Norādiet pilnu ceļu uz klienta sertifikāta failu, ja ir nepieciešama savstarpēja TLS autentifikācija.
Servera TLS sertifikāta izdevējs	Norādiet izdevēja atšķirīgo nosaukumu (DN), kas jāsalīdzina ar servera sertifikātu.
Servera TLS sertifikāta subjekts	Norādiet subjekta atšķirīgo nosaukumu (DN), kas jāsalīdzina ar servera sertifikātu.

Esošās instalācijās rediģējiet tālāk norādītos parametrus lietotāja saskarnes konfigurācijas failā:

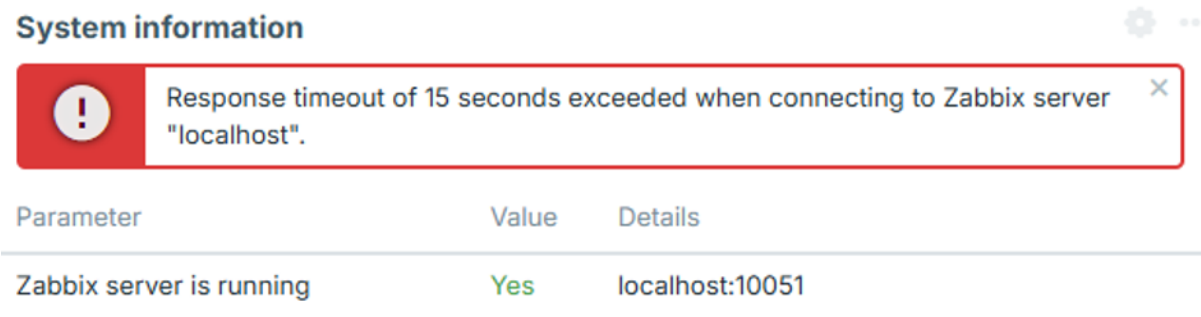
```
$ZBX_SERVER_TLS['ACTIVE'] = 'true';
$ZBX_SERVER_TLS['CA_FILE'] = '/etc/zabbix/ssl/ca.crt';
$ZBX_SERVER_TLS['KEY_FILE'] = '/etc/zabbix/ssl/frontend.key';
$ZBX_SERVER_TLS['CERT_FILE'] = '/etc/zabbix/ssl/frontend.crt';
// Pēc izvēles:
// $ZBX_SERVER_TLS['CERTIFICATE_ISSUER']  = 'CN=MyZabbixCA';
// $ZBX_SERVER_TLS['CERTIFICATE_SUBJECT'] = 'CN=zabbix-server.example.com';
```

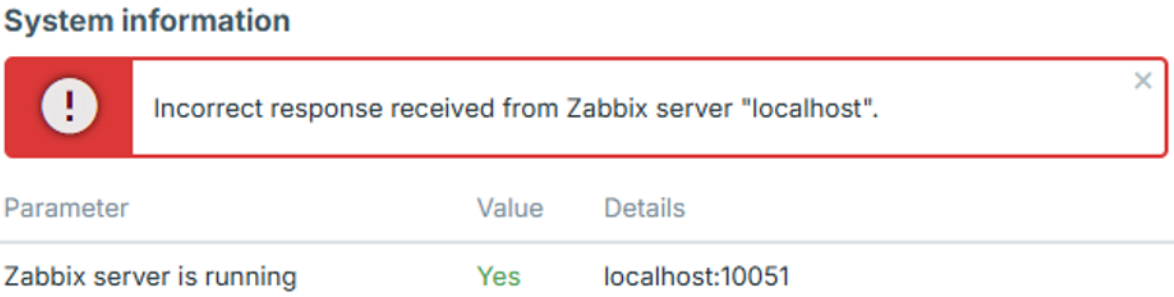
5. Pārbaudiet šifrēšanu, pārliecinoties, ka Zabbix lietotāja saskarnē vai Zabbix servera žurnālfailā nav kļūdu ziņojumu:

```
tail -f /var/log/zabbix/zabbix_server.log
```

Problēmu novēršana

Nepareiza vai nepilnīga TLS konfigurācija var izraisīt kļūdas Zabbix lietotāja saskarnē vai norādīt, ka Zabbix serveris darbojas, taču savienojums nav veiksmīgi izveidots:





Lai identificētu precīzu konfigurācijas problēmu, iestatiet Zabbix servera žurnāla līmeni uz 4, atkārtojiet kļūdu lietotāja saskarnē un pārbaudiet Zabbix servera žurnālu, lai atrastu atbilstošo kļūdas ziņojumu.

Tālāk aprakstīti biežākie kļūdu gadījumi un to iespējamie cēloņi.

Lietotāja saskarnes IP adrese nav atļauta

Zabbix servera žurnālā ir ietverta līdzīga kļūda:

```
54283:20260819:105926.400 frontend connection from "127.0.0.1" is not allowed by FrontendAllowedIP
```

Savienojums tiek pārtraukts pēc TCP rokasspiediena, jo lietotāja saskarnes IP adrese nav autorizēta.

Pievienojiet lietotāja saskarnes IP adresi parametram `FrontendAllowedIP` failā `zabbix_server.conf` un restartējiet Zabbix serveri.

Nešifrēts savienojums nav atļauts

Zabbix servera žurnālā ir ietverts kļūdas ziņojums, kas līdzīgs šim:

```
56884:20260819:111154.285 frontend connection of type "unencrypted" is not allowed by TLSFrontendAccept
```

Zabbix serveris ir konfigurēts pieņemt tikai TLS savienojumus, piemēram, izmantojot `TLSFrontendAccept=cert`, taču lietotāja saskarne mēģina izveidot savienojumu bez šifrēšanas.

Iespējojiet lietotāja saskarnes šifrēšanu un konfigurējiet nepieciešamos TLS failus failā `zabbix.conf.php`, kā aprakstīts sadaļas Configuration 4. darbībā.

Lietotāja saskarne mēģina izmantot TLS, kamēr serveris sagaida nešifrētu savienojumu

Zabbix servera žurnālā ir ietverts kļūdas ziņojums, kas līdzīgs šim:

```
61091:20260819:115916.019 End of zbx_tls_accept():FAIL error:'TLS handshake set result code to 1: file ../
```

Precīzs kļūdas ziņojums ir atkarīgs no OpenSSL versijas.

Zabbix serveris ir konfigurēts pieņemt tikai nešifrētus savienojumus, piemēram, izmantojot `TLSFrontendAccept=unencrypted`, kamēr lietotāja saskarne mēģina izveidot TLS savienojumu.

Konfigurējiet Zabbix serveri TLS savienojumu pieņemšanai, iestatot `TLSFrontendAccept=cert` un definējot `TLSCertFile` un `TLSKeyFile` failā `zabbix_server.conf`. Varat arī atspējot lietotāja saskarnes šifrēšanu failā `zabbix.conf.php`.

Lietotāja saskarnes sertifikāts nav uzticams

Zabbix servera žurnālā ir ietverts kļūdas ziņojums, kas līdzīgs šim:

```
62248:20260819:124203.812 End of zbx_tls_accept():FAIL error:'self-signed certificate: TLS handshake set r
```

Lietotāja saskarne uzrādīja sertifikātu, taču Zabbix serveris neuzticas sertificēšanas iestādei (CA), kas to parakstīja.

Tā pati problēma var tikt parādīta arī lietotāja saskarnes sistēmas informācijā:

```
fread(): SSL operation failed with code 1. OpenSSL Error messages:

error:0A000418:SSL routines::tlsv1 alert unknown ca [zabbix.php:17 → require_once() → ZBase->run() → ZBase
```

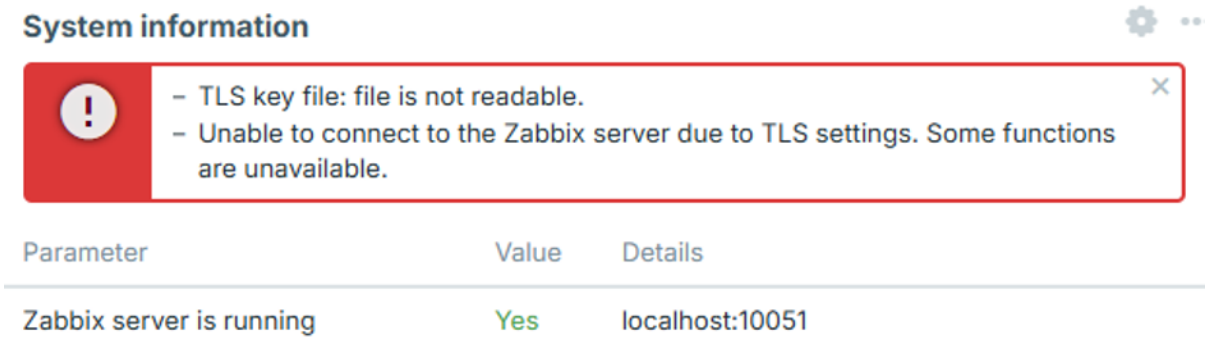
Lietotāja saskarnē var tikt parādīts arī ziņojums “Incorrect response received from Zabbix server ”localhost””.

Pārbaudiet, vai lietotāja saskarnes sertifikātu ir parakstījusi CA, kas norādīta `TLSCAFile` parametrā failā `zabbix_server.conf`.

TLS atslēgas fails nav lasāms

Lietotāja saskarnē var tikt parādīts paziņojums “TLS key file: file is not readable.”

Var tikt parādīts arī paziņojums “Unable to connect to the Zabbix server due to TLS settings. Some functions are unavailable.”.

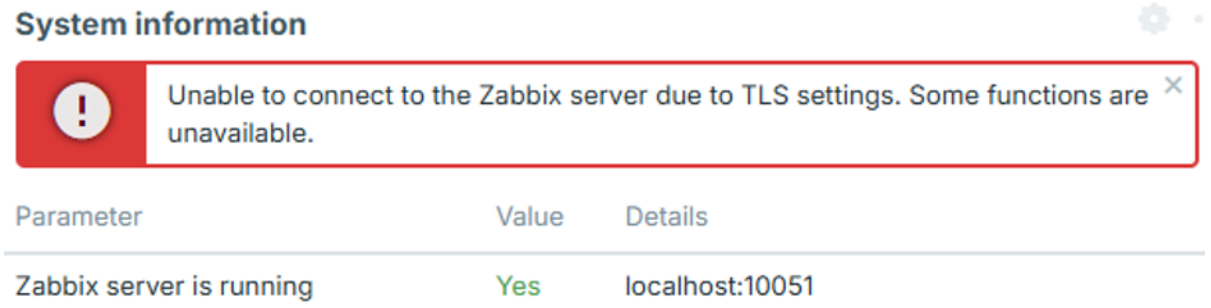


Tīmekļa servera lietotājam nav pietiekamu atļauju lasīt vienu vai vairākus sertifikātu, atslēgu vai CA failus, kas konfigurēti failā `zabbix.conf.php`. Tas var notikt, ja sertifikātu faili tiek glabāti ierobežotas piekļuves direktorijā, piemēram, /root/ vai lietotāja mājas direktorijā.

Pārvietojiet sertifikātu failus uz direktoriju, kurai var piekļūt tīmekļa serveris, un iestatiet atbilstošas atļaujas, kā aprakstīts sadaļas Configuration 2. darbībā.

TLS faili nav derīgi vai ir bojāti

Lietotāja saskarnē var tikt parādīts ziņojums “Nevar izveidot savienojumu ar Zabbix serveri TLS iestatījumu dēļ. Dažas funkcijas nav pieejamas.”.



Tas var notikt, ja konfigurētie sertifikāta vai atslēgas faili pastāv un ir nolasāmi, bet to saturs ir tukšs, bojāts vai citādi nederīgs. Šādā gadījumā lietotāja saskarne neizveido savienojumu ar Zabbix serveri, un Zabbix servera žurnālā netiek reģistrēts atbilstošs savienojuma mēģinājums.

Pārbaudiet, vai sertifikāta un atslēgas faili satur derīgus, pareizi formatētus datus un vai privātā atslēga atbilst sertifikātam.

Piekļuves kontrole

Pārskats

Šajā sadaļā ir ietvertas labākās prakses piekļuves kontroles drošai konfigurēšanai.

Mazāko privilēģiju princips

Lietotāju kontiem vienmēr jādarbojas ar pēc iespējas mazāku privilēģiju skaitu. Tas nozīmē, ka lietotāju kontiem Zabbix lietotāja saskarnē, datubāzes lietotājiem vai lietotājam Zabbix servera/starpniekservera/aģenta procesos jābūt tikai tām privilēģijām, kas ir būtiskas paredzēto funkciju veikšanai.

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 *Admin* lietotāja tips ir ar mazāk privilēģijām nekā *Super Admin* lietotāja tips, tas joprojām var pārvaldīt konfigurāciju un izpildīt pielāgotus skriptus.

Note:
Daļa informācijas ir pieejama arī lietotājiem bez 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 izslēgšana no skriptiem (piemēram, piekļuves akreditācijas dati) var palīdzēt izvairīties no sensitīvas informācijas atklāšanas globālajos skriptos.

Drošs lietotājs Zabbix aģentam

Pēc noklusējuma Zabbix serveris, starpniekserveris un aģents (vai agent 2) procesi izmanto vienu `zabbix` lietotāju. Lai novērstu to, ka Zabbix aģents/agent 2 (kas darbojas 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 User parametrā.
3. Pārstartējiet aģentu, lai tas pārietu uz jaunā lietotāja privilēģijām.

Zabbix agent 2 gadījumā konfigurācija jāpiemēro pakalpojuma līmenī, jo agent 2 konfigurācijas fails neatbalsta `User` parametru. Piemēru skatiet ZBX-26442.

Atsaukt rakstīšanas piekļuvi SSL konfigurācijai (Windows)

Ja esat kompilējis Zabbix aģentu operētājsistēmā Windows un OpenSSL atrodas neaizsargātā direktorijā (piem., `C:\zabbix`, `c:\openssl-64bit`, `C:\OpenSSL-Win64-111-static` vai `C:\dev\openssl`), pārliecinieties, ka lietotājiem, kas nav administratori, ir atsaukta rakstīšanas piekļuve šim direktorijam. Pretējā gadījumā aģents ielādē SSL iestatījumus no ceļa, ko var modificēt lietotāji bez paaugstinātām tiesībām, tādējādi radot iespējamu drošības ievainojamību.

Zabbix komponentu drošības nostiprināšana

Lai nostiprinātu Zabbix komponentu drošību, dažas funkcijas var atspējot:

- Globālo skriptu izpildi Zabbix serverī var atspējot, servera konfigurācijā iestatot `EnableGlobalScripts=0`.
- Globālā skriptu izpilde Zabbix starpniekserverī pēc noklusējuma ir atspējota (to var iespējot, starpniekservera konfigurācijā iestatot `EnableRemoteCommands=1`).
- Globālā skriptu izpilde Zabbix aģentos pēc noklusējuma ir atspējota (to var iespējot, aģenta konfigurācijā katrai atļautajai komandai pievienojot parametru `AllowKey=system.run[<command>,*]` vai `AllowKeyRegexp=^system\.run\[.*\]$`).
- Piekļuve lietotāju HTTP autentifikācijas konfigurācijai pēc noklusējuma ir iespējota; to var atspējot, lietotāja saskarnes konfigurācijā iestatot `$ZBX_FEATURE_FLAGS['http_auth_enabled'] = false`.
- Piekļuve moduļu konfigurācijai pēc noklusējuma ir iespējota; to var atspējot, lietotāja saskarnes konfigurācijā iestatot `$ZBX_FEATURE_FLAGS['modules_config_enabled'] = false`.
- Konkrētu multivides tipu rediģēšanu var ierobežot, lietotāja saskarnē iestatot `$ZBX_FEATURE_FLAGS['media_type_denylist'] = [<value1>, <value2>, ...]`. Iespējamās vērtības: `'email'`, `'script'`, `'sms'`, `'webhook'` un `'push'`. Var norādīt vienu vai vairākas vērtības.

Lietotāja saskarnes atkārtota instalēšana, palaižot `setup.php`, ir ierobežota, ja pastāv konfigurācijas fails `zabbix.conf.php`.

UNC ceļu piekļuve 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 lūgts piekļūt UNC ceļam, tā mēģina autentificēties šajā serverī. Tas nozīmē, ka ļaunprātīgs pieprasījums Zabbix aģentam var atklāt NTLM jaucējvērtību pieprasījuma nosūtītāja serverim. Lietotāji var mazināt šo risku, pēc vajadzības konfigurējot konfigurācijas parametrus AllowKey, AllowKeyRegexp, DenyKey un DenyKeyRegexp.

MySQL/MariaDB drošības nodrošināšana

Pārskats

Šī sadaļa satur labākās prakses 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, veicot Zabbix jaunināšanu.

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 servera un starpniekservera palaišanai:

```
CREATE ROLE 'zbx_srv';
GRANT DELETE, INSERT, SELECT, UPDATE ON zabbix.* TO 'zbx_srv';
FLUSH PRIVILEGES;
```

- **zbx_web** - loma Zabbix lietotāja saskarnes un API palaišanai:

```
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 sadalīšanai pa nodalījumiem; 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';
-- For MariaDB: skip the next line (GRANT session_variables_admin ON *.* TO 'zbx_part';)
GRANT session_variables_admin ON *.* TO 'zbx_part';
GRANT SELECT ON zabbix.dbversion TO 'zbx_part';
GRANT SELECT, DELETE ON zabbix.housekeeper TO 'zbx_part';
FLUSH PRIVILEGES;
```

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> ar atbilstošajām vērtībām. Pārliecinieties, ka <password> ir droša un unikāla.

```
CREATE USER '<user>'@'<host>' IDENTIFIED BY '<password>';
GRANT '<role>' TO '<user>'@'<host>';
SET DEFAULT ROLE '<role>' TO '<user>'@'<host>';
-- MariaDB: SET DEFAULT ROLE '<role>' FOR '<user>'@'<host>'
FLUSH PRIVILEGES;
```

Piemēram, lai izveidotu un piešķirtu lomu Zabbix servera un starpniekservera darbināšanai:

```
CREATE USER 'usr_srv'@'localhost' IDENTIFIED BY 'strong-password';
GRANT 'zbx_srv' TO 'usr_srv'@'localhost';
SET DEFAULT ROLE ALL TO 'usr_srv'@'localhost';
FLUSH PRIVILEGES;
```

PostgreSQL/TimescaleDB drošības nodrošināšana

Pārskats

Šajā sadaļā ir aprakstīta labākā prakse PostgreSQL datubāzes drošības nodrošināšanai.

Note:
Pamata iestatīšanai skatiet noklusējuma PostgreSQL datubāzes izveides norādījumus, kas ietver lietotāja 'zabbix' izveidi ar pilnām tiesībām Zabbix datubāzē. Šis lietotājs ir datubāzes īpašnieks, kuram ir arī nepieciešamās tiesības datubāzes struktūras modificēšanai, veicot Zabbix jaunināšanu.

Lai uzlabotu drošību, ieteicams konfigurēt drošu shēmas lietošanas modeli, kā arī izveidot papildu datubāzes lomas un lietotājus ar minimālām tiesībām. Šīs lomas un lietotāji jākonfigurē, pamatojoties uz mazāko privilēģiju principu, tas ir, tiem jābūt tikai tām tiesībām, kas ir būtiskas paredzēto funkciju veikšanai.

Datubāzes iestatīšana

Izveidojiet lietotāju, kas būs datubāzes īpašnieks, un izveidojiet Zabbix datubāzi; datubāzes īpašnieks ir lietotājs, kas tiek 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 atjaunināšana ir 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:
Tālāk norādītās komandas šajā lapā ir 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 serveris un starpniekserveris palaišanai:

```
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 saskarne un API palaišanai:

```
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> ar atbilstošām vērtībām. Pārliecinieties, ka <password> ir droša un unikāla.

```
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 'strong-password';
GRANT zbx_srv TO usr_srv;
```

Datu bāzes sadalīšana ar TimescaleDB

Datu bāzes sadalīšanu nodrošina TimescaleDB. Lai izmantotu TimescaleDB, Zabbix serverim ir nepieciešamas datu bāzes īpašnieka tiesības.

Ja PostgreSQL zabbix shēma jau ir izveidota datu bāzē zabbix, varat iespējot TimescaleDB ar šādu komandu:

```
echo "CREATE EXTENSION IF NOT EXISTS timescaledb WITH SCHEMA zabbix CASCADE;" | sudo -u postgres psql zabb
```

Kriptogrāfija

Pārskats

Šajā sadaļā ir ietvertas labākās prakses kriptogrāfijas iestatīšanai drošā veidā.

SSL iestatīšana Zabbix lietotāja saskarnei

RHEL bāzētās sistēmās 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 -
```

Aizpildiet uzvednes atbilstoši. Svarīgākā rinda ir tā, kas pieprasa Common Name. Jums jāievada domēna nosaukums, ko 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
```

Pārstartējiet Apache pakalpojumu, lai piemērotu izmaiņas:

```
systemctl restart httpd.service
```

Tīmekļa servera drošības nodrošināšana

Pārskats

Šajā sadaļā ir aprakstīta labākā prakse tīmekļa servera drošai konfigurēšanai.

Saknes URL novirzīšanas uz Zabbix SSL piespiedu iestatīšana

RHEL bāzētās sistēmās pievienojiet virtuālo hostu Apache konfigurācijai (`/etc/httpd/conf/httpd.conf`) un iestatiet pastāvīgu dokumentu saknes novirzīšanu uz Zabbix SSL URL. Ņemiet vērā, ka `example.com` ir jāaizstāj ar faktiskā servera nosaukumu.

```
### Pievienojiet rindas:

<VirtualHost *:*>
    ServerName example.com
    Redirect permanent / https://example.com
</VirtualHost>
```

Pārstartējiet Apache pakalpojumu, lai piemērotu izmaiņas:

```
systemctl restart httpd.service
```

HTTP Strict Transport Security (HSTS) iespējošana tīmekļa serverī

Lai aizsargātu Zabbix lietotāja saskarni pret protokola pazemināšanas uzbrukumiem, mēs iesakām tīmekļa serverī iespējot HSTS politiku.

Lai Apache konfigurācijā iespējotu HSTS politiku savam Zabbix lietotāja saskarnei, veiciet šādas 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 Strict-Transport-Security "max-age=31536000"
</VirtualHost>
```

3. Restartējiet Apache pakalpojumu, lai piemērotu izmaiņas:

```
### RHEL bāzētās sistēmās:
systemctl restart httpd.service

### Debian/Ubuntu
systemctl restart apache2.service
```

Drošu un SameSite sesijas sīkfailu ieviešana Zabbix

Konfigurējot Zabbix, ir svarīgi piespiest sesijas sīkfailiem izmantot drošos un SameSite atribūtus, lai uzlabotu drošību un novērstu starpvietņu pieprasījumu viltošanas (CSRF) uzbrukumus. Tomēr `SameSite=Strict` piespiešana noteiktos scenārijos var radīt problēmas, piemēram:

- Informācijas paneļa URL logrīki rāda "user not logged in", ja tiek iegulti viena domēna iframe.
- Lietotāji, kas piekļūst informācijas panelim, izmantojot HTTP, nevis HTTPS, var saskarties ar pieteikšanās problēmām.
- Nav iespējams kopīgot URL uz konkrētām Zabbix izvēlnes sadaļām vai hostiem.

Lai mazinātu šīs problēmas, lietotājiem vajadzētu būt iespējai pielāgot SameSite politiku.

1. Drošie sīkfaili

Iestatot `secure` karodziņu, tiek nodrošināts, ka sīkfaili tiek pārsūtīti tikai pa HTTPS, novēršot to atklāšanu nešifrētos savienojumos.

Lai Zabbix iespējotu drošos sīkfailus, tīmekļa servera konfigurācijā pievienojiet vai modificējiet šādu iestatījumu:

Apache:

```
Header always edit Set-Cookie ^(.*)$ $1;Secure
```

Nginx:

```
proxy_cookie_path / "/; Secure";
```

Pārliecinieties, ka jūsu Zabbix lietotāja saskarne ir pieejama, izmantojot HTTPS; pretējā gadījumā sīkfaili ar `Secure` karodziņu netiks nosūtīti.

2. SameSite atribūta konfigurēšana

Tīmekļa servera iestatījumi var arī piespiest SameSite atribūtu:

Apache:

```
<IfModule mod_headers.c>
    Header onsuccess edit Set-Cookie (.*) "$1; SameSite=Strict"
</IfModule>
```

Nginx (versija 1.19.3+):

```
proxy_cookie_flags ~ samesite=Strict; # Replace ~ with 'zbx_session' for specificity
```

Satura drošības politikas (CSP) iespējošana tīmekļa serverī

Lai aizsargātu Zabbix lietotāja saskarni pret Cross Site Scripting (XSS), datu ievadīšanu un līdzīgiem uzbrukumiem, iesakām tīmekļa serverī iespējot Content Security Policy.
Lai to izdarītu, konfigurējiet tīmekļa serveri tā, lai tas atgrieztu HTTP galveni.

Attention:
Tālāk norādītā CSP galvenes konfigurācija ir paredzēta tikai noklusējuma Zabbix lietotāja saskarnes instalācijai un gadījumiem, kad viss saturs nāk no vietnes domēna (izņemot apakšdomēnus).
Var būt nepieciešama cita CSP galvenes konfigurācija, ja, piemēram, konfigurējat *URL* logrīku, lai rādītu saturu no vietnes apakšdomēniem vai ārējiem domēniem, pārslēdzaties no *OpenStreetMap* uz citu karšu dzinēju vai pievienojat ārēju CSS vai logrīkus.
Ja izmantojat Duo Universal Prompt vairāku faktoru autentifikācijas metodi, noteikti pievienojiet "duo.com" CSP direktīvai sava virtuālā resursdatora konfigurācijas failā.

Lai iespējotu CSP savā Zabbix lietotāja saskarnē Apache konfigurācijā, veiciet šīs darbības:

1. Atrodiet sava virtuālā resursdatora 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 resursdatora konfigurācijas failam:

```
<VirtualHost *:*>
    Header set Content-Security-Policy: "default-src 'self' *.openstreetmap.org; script-src 'self' 'unsafe
</VirtualHost>
```

3. Restartējiet Apache pakalpojumu, lai piemērotu izmaiņas:

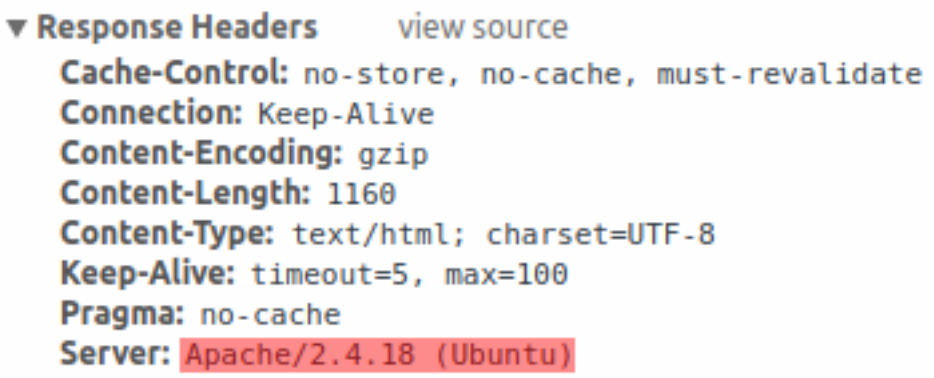
```
### On RHEL-based systems:
systemctl restart httpd.service

### 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, Apache konfigurācijas failam pievienojot 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, nepieciešams restartēt tīmekļa serveri.

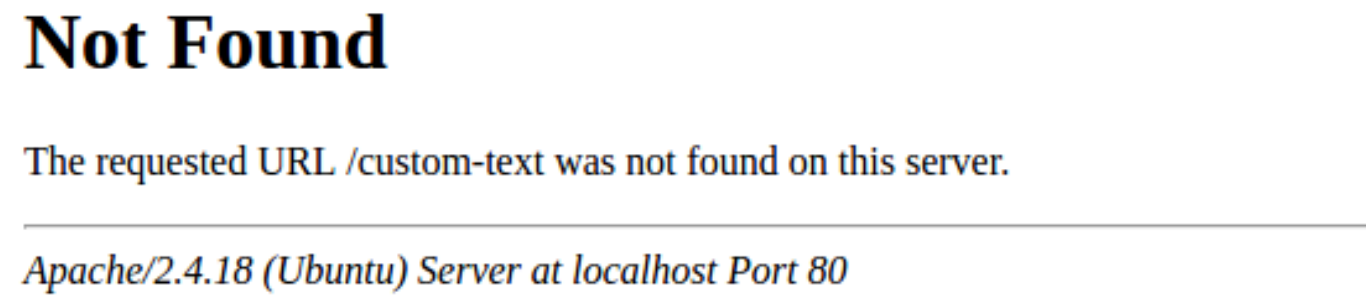
Papildu drošībai kopā ar Apache varat izmantot rīku *mod_security* (pakotni *libapache2-mod-security2*). Šis rīks ļauj noņemt servera parakstu, nevis tikai noņemt versiju no servera paraksta. Servera parakstu var mainīt uz jebkādu vērtību, pēc *mod_security* instalēšanas iestatot parametram "SecServerSignature" vēlamo vērtību.

Lai uzzinātu, kā noņemt vai mainīt programmatūras parakstus, skatiet sava tīmekļa servera dokumentāciju.

Tīmekļa servera noklusējuma kļūdu lapu atspējošana

Lai novērstu informācijas atklāšanu, ieteicams atspējot noklusējuma kļūdu lapas.

Pēc noklusējuma tīmekļa serveris izmanto iebūvētās kļūdu lapas:



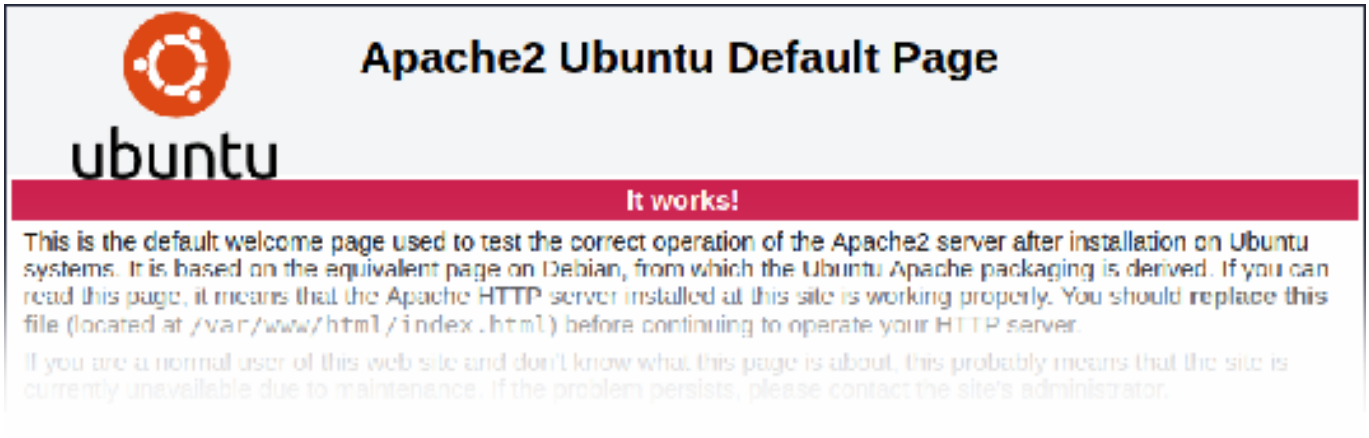
Šīs noklusējuma kļūdu lapas ir jāaizstāj vai jānoņem. Piemēram, Apache tīmekļa serverī direktīvu "ErrorDocument" var izmantot, lai definētu pielāgotu kļūdu lapu vai tekstu.

Lai uzzinātu, kā aizstāt vai noņemt noklusējuma kļūdu lapas, skatiet sava tīmekļa servera dokumentāciju.

Tīmekļa servera testa lapas noņemšana

Lai novērstu informācijas noplūdi, ieteicams noņemt tīmekļa servera testa lapu.

Pēc noklusējuma Apache tīmekļa servera tīmekļa saknes direktorijā atrodas testa lapa `index.html`:



Lai uzzinātu, kā noņemt noklusējuma testa lapas, skatiet sava tīmekļa servera dokumentāciju.

Iestatīt X-Frame-Options HTTP atbildes galveni

Pēc noklusējuma Zabbix ir konfigurēts ar parametru *Use X-Frame-Options HTTP header*, kam ir iestatīta vērtība `SAMEORIGIN`. Tas nozīmē, ka saturs var tikt ielādēts tikai rāmī, kura izcelsme ir tāda pati kā pašai lapai.

Zabbix lietotāja saskarnes elementi, kas iegūst saturu no ārējiem URL (proti, URL dashboard widget), parāda iegūto saturu smilškastē ar iespējotiem visiem smilškastes ierobežojumiem.

Šie iestatījumi uzlabo Zabbix lietotāja saskarnes drošību un nodrošina aizsardzību pret XSS un clickjacking uzbrukumiem. *Super admin* lietotāji var modificēt parametrus *Use iframe sandboxing* un *Use X-Frame-Options HTTP header* pēc vajadzības. Lūdzu, rūpīgi izvērtējiet riskus un ieguvumus pirms noklusējuma iestatījumu maiņas. Nav ieteicams pilnībā atspējot iframe smilškastēšanu vai X-Frame-Options HTTP galveni.

Faila ar biežāk lietoto paroļu sarakstu slēpšana

Lai palielinātu paroļu brutāla spēka uzbrukumu sarežģītību, ieteicams ierobežot piekļuvi failam `ui/data/top_passwords.txt`. Šis fails satur visbiežāk lietoto un kontekstam specifisko paroļu sarakstu un neļauj lietotājiem iestatīt šādas paroles (ja ir iespējots parametrs *Avoid easy-to-guess passwords* paroļu politikā).

Lai ierobežotu piekļuvi failam `top_passwords.txt`, modificējiet sava tīmekļa servera konfigurāciju.

Apache gadījumā piekļuvi failam var ierobežot, izmantojot `.htaccess` failu:

```
<Files "top_passwords.txt">
    Order Allow,Deny
    Deny from all
</Files>
```

NGINX gadījumā piekļuvi failam var ierobežot, izmantojot direktīvu `location`:

```
location = /data/top_passwords.txt {
    deny all;
    return 404;
}
```

Slepeno datu glabāšana

Pārskats Zabbix var konfigurēt tā, lai tas izgūtu sensitīvu informāciju no droša glabātuves. Tiek atbalstīti šādi slepeno datu pārvaldības pakalpojumi: HashiCorp Vault KV Secrets Engine - Version 2, CyberArk Vault CV12.

Slepenie dati var tikt izmantoti, lai izgūtu:

- lietotāja makro vērtības
- datubāzes piekļuves akreditācijas datus

Zabbix nodrošina tikai lasīšanas piekļuvi glabātuvē esošajiem slepenajiem datiem, pieņemot, ka slepenie dati tiek pārvaldīti citur.

Lai iegūtu informāciju par konkrētu vault nodrošinātāja konfigurāciju, skatiet:

- HashiCorp konfigurācija
- CyberArk konfigurācija

Slepeno vērtību kešatmiņa Pēc noklusējuma vault slepeno makro vērtības Zabbix serveris izgūst katrā konfigurācijas datu atsvaidzināšanas reizē un pēc tam saglabā konfigurācijas kešatmiņā. Zabbix starpniekserveris saņem vault slepeno makro vērtības no Zabbix servera katras konfigurācijas sinhronizācijas laikā un saglabā tās savā konfigurācijas kešatmiņā.

> **Attention:**
> Starp Zabbix serveri un starpniekserveri ir jābūt iespējotai šifrēšanai; pretējā gadījumā žurnālā tiek ierakstīts servera brīdinājuma ziņojums.

Ir iespējams arī konfigurēt, lai makro vērtības Zabbix serveris un Zabbix starpniekserveris izgūtu neatkarīgi.

Lai manuāli aktivizētu kešoto slepeno vērtību atsvaidzināšanu no vault, 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` lietotāja saskarnes konfigurācijas failā. Akreditācijas dati tiks saglabāti lokālajā kešatmiņā, izmantojot failu sistēmas pagaidu failu direktoriju. Tīmekļa serverim ir jāatļauj rakstīšana privātā pagaidu mapē (piemēram, Apache gadījumā ir jāiestata konfigurācijas opcija `PrivateTmp=True`). Lai kontrolētu, cik bieži datu kešatmiņa tiek atsvaidzināta/anulēta, izmantojiet `ZBX_DATA_CACHE_TTL` konstanti.

TLS konfigurācija Lai konfigurētu TLS saziņai starp Zabbix komponentiem un seifu, pievienojiet sertifikātu, ko parakstījusi sertifikātu iestāde (CA), sistēmas noklusējuma CA krātuvei.
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 .
```

CyberArk konfigurācija

Šajā sadaļā ir paskaidrots, kā konfigurēt Zabbix, lai izgūtu noslēpumus no CyberArk Vault CV12.

Vault jābūt instalētam un konfigurētam, kā aprakstīts oficiālajā CyberArk dokumentācijā.

Lai uzzinātu par TLS konfigurēšanu Zabbix, skatiet *Noslēpumu glabāšana*.

Datu bāzes akreditācijas dati

Piekļuve noslēpumam ar datu bāzes akreditācijas datiem tiek konfigurēta katram Zabbix komponentam atsevišķi.

Serveris un starpniekserveri

Lai iegūtu datu bāzes akreditācijas datus no glabātavas Zabbix serverim vai starpniekserverim, konfigurācijas failā norādiet šādus konfigurācijas parametrus:

- `Vault` - kurš glabātavas nodrošinātājs jāizmanto;
- `VaultURL` - glabātavas servera HTTP(S) URL;
- `VaultDBPath` - vaicājums glabātavas noslēpumam, kas satur datu bāzes akreditācijas datus, kuri tiks izgūti, izmantojot atslēgas "Content" un "UserName" (šo opciju var izmantot tikai tad, ja `DBUser` un `DBPassword` nav norādīti);
- `VaultTLSCertFile`, `VaultTLSKeyFile` - SSL sertifikāta un atslēgas failu nosaukumi; šo opciju iestatīšana nav obligāta, taču ļoti ieteicama;
- `VaultPrefix` - pielāgots glabātavas ceļa vai vaicājuma prefikss atkarībā no glabātavas; ja tas nav norādīts, tiks izmantota vispiemērotākā noklusējuma vērtība.

Attention:
Konfigurācijas parametri `Vault`, `VaultURL`, `VaultTLSCertFile`, `VaultTLSKeyFile` un `VaultPrefix` tiek izmantoti arī glabātavas autentifikācijai, kad Zabbix serveris apstrādā slepenās glabātavas makro (un Zabbix starpniekserveris, ja tas ir konfigurēts). Zabbix serveris un starpniekserveri neatvērs slepenās glabātavas makro, kas satur DB akreditācijas datus no `VaultDBPath`.

Zabbix serveris un Zabbix starpniekserveris palaišanas laikā nolasa ar glabātavu saistītos konfigurācijas parametrus no failiem `zabbix_server.conf` un `zabbix_proxy.conf`.

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 šādu API pieprasījumu uz seifu:

```
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. Seifa atbildē būs atslēgas "Content" un "UserName":

```
{
    "Content": <password>,
    "UserName": <username>,
    "Address": <address>,
    "Database": <Database>,
    "PasswordChangeInProcess":<PasswordChangeInProcess>
}
```

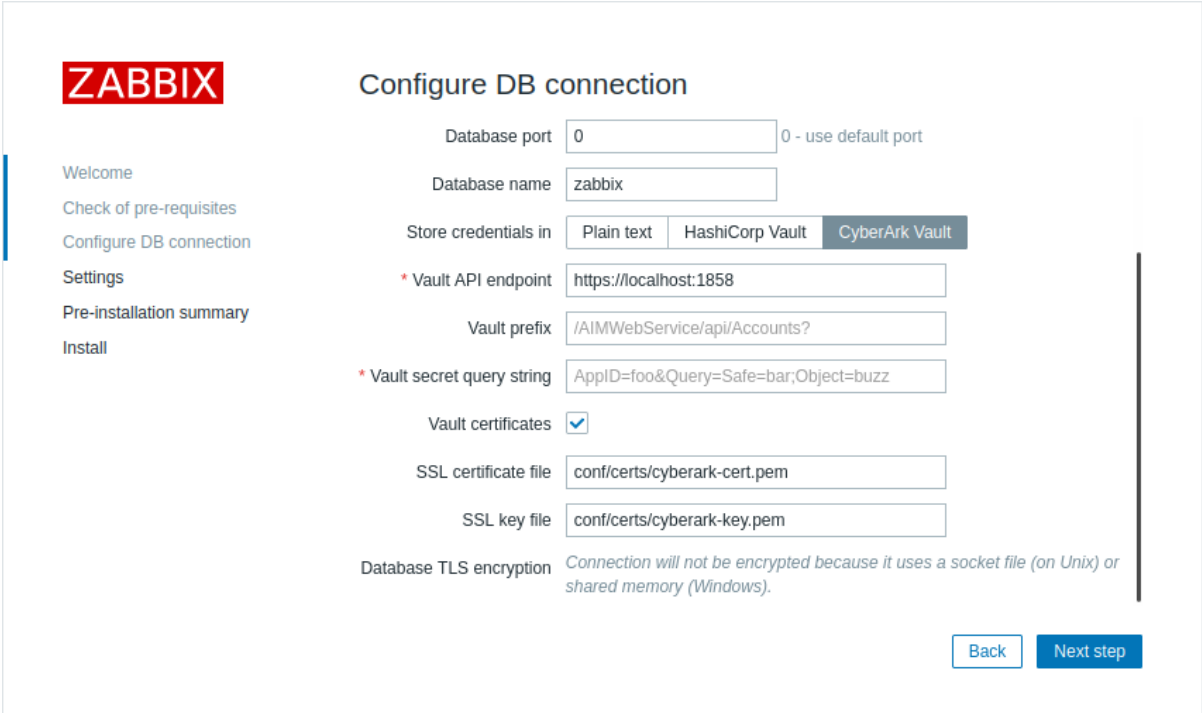
4. Rezultātā Zabbix datubāzes autentifikācijai izmantos šādus akreditācijas datus:

- Lietotājvārds: <username>
- Parole: <password>

Lietotāja saskarne

Lai iegūtu datubāzes akreditācijas datus no glabātavas Zabbix lietotāja saskarnei, lietotāja saskarnes instalēšanas laikā norādiet tālāk minētos parametrus.

1. Darbībā *Configure DB Connection* iestatiet parametram *Store credentials in* vērtību *CyberArk Vault*.



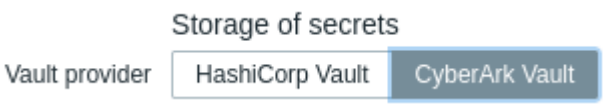
2. Pēc tam aizpildiet papildu parametrus:

Parameter	Mandatory	Default value	Description
Vault API endpoint	yes	https://localhost:1858	Norādiet URL savienojuma izveidei ar glabātavu formātā `scheme://host:port`
Vault prefix	no	/AIMWebService/api/Accounts?	Norādiet pielāgotu glabātavas ceļa vai vaicājuma prefiksu. Ja tas nav norādīts, tiek izmantota noklusējuma vērtība.
Vault secret query string	yes		Vaicājums, kas norāda, no kurienes jāiegūst datubāzes akreditācijas dati. Piemērs: `AppID=foo&Query=Safe=bar;Object=buzz`
Vault certificates	no		Pēc izvēles rūtiņas atzīmēšanas 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	no	conf/certs/cyberark-cert.pem	SSL sertifikāta faila ceļš. Failam jābūt PEM formātā. Ja sertifikāta fails satur arī privāto atslēgu, atstājiet SSL atslēgas faila parametru tukšu.
SSL key file	no	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 *Administrācija > 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 jāietver vaicājums (kā `query:key`).

Skatiet *Vault secret macros*, lai iegūtu detalizētu informāciju par makro vērtību apstrādi Zabbix.

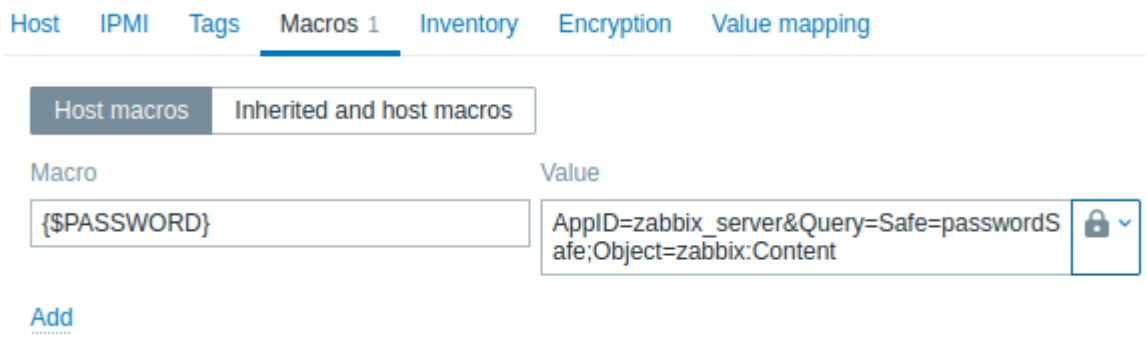
Vaicājuma sintakse

Kols (`:`) ir rezervēts vaicājuma atdalīšanai no atslēgas.

Ja pats vaicājums satur slīpsvītru vai kolu, šie simboli ir jāiekodē URL formātā (`/` tiek kodēts kā `%2F`, `:` tiek kodēts kā `%3A`).

Piemērs

1. Zabbix pievienojiet lietotāja makro `{$PASSWORD}` ar tipu *Vault secret* un vērtību `AppID=zabbix_server&Query=Safe=passwordSa`



2. Zabbix nosūtīs vault 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. Vault atbilde saturēs atslēgu `Content`:

```
{
    "Content": <password>,
    "UserName": <username>,
    "Address": <address>,
    "Database" :<Database>,
    "PasswordChangeInProcess":<PasswordChangeInProcess>
}
```

4. Rezultātā Zabbix atrisinās makro `{$PASSWORD}` līdz vērtībai - `<password>`

Esošās konfigurācijas atjaunināšana

Lai atjauninātu esošu konfigurāciju noslēpumu izgūšanai no CyberArk Vault:

1. Atjauniniet Zabbix servera vai starpniekservera konfigurācijas faila parametrus, kā aprakstīts sadaļā *Datubāzes akreditācijas dati*.

2. Atjauniniet Zabbix lietotāja saskarnes konfigurācijas faila parametrus:

```
$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 makro, kā aprakstīts sadaļā *Lietotāja makro vērtības*.

Lai atjauninātu esošu konfigurāciju noslēpumu izgūšanai no HashiCorp Vault, skatiet sadaļu *HashiCorp konfigurācija*.

HashiCorp konfigurācija

Pārskats

Šajā sadaļā ir paskaidrots, kā konfigurēt Zabbix, lai izgūtu noslēpumus no HashiCorp Vault KV Secrets Engine - Version 2.

Vault ir jāizvieto un jākonfigurē, kā aprakstīts oficiālajā HashiCorp dokumentācijā.

Lai uzzinātu par TLS konfigurēšanu Zabbix, skatiet *Noslēpumu glabāšana*.

Datu bāzes akreditācijas datu izgūšana

Lai veiksmīgi izgūtu noslēpumu ar datu bāzes akreditācijas datiem, ir jākonfigurē abi:

- Zabbix serveris/starpniekserveris
- Zabbix lietotāja saskarne

Serveris/starpniekserveris

Lai konfigurētu Zabbix serveri vai starpniekserveri, konfigurācijas failā norādiet šādus konfigurācijas parametrus:

- `Vault` - kurš vault nodrošinātājs jāizmanto.
- `VaultToken` - vault autentifikācijas tokens (detalizētu informāciju skatiet Zabbix servera/starpniekservera konfigurācijas failā); to nevar izmantot kopā ar `VaultAppRoleID`.
- `VaultAppRoleID` un `VaultAppSecretID` - vault autentifikācijas lomas ID un slepenais ID (detalizētu informāciju skatiet Zabbix servera/starpniekservera konfigurācijas failā); tos nevar izmantot kopā ar `VaultToken`.
- `VaultURL` - vault servera HTTP(S) URL.
- `VaultDBPath` - ceļš uz vault noslēpumu, kurā ir datu bāzes akreditācijas dati (šo opciju var izmantot tikai tad, ja `DBUser` un `DBPassword` nav norādīti). Zabbix serveris vai starpniekserveris izgūs akreditācijas datus, izmantojot atslēgas ”password” un ”username”.
- `VaultPrefix` - pielāgots prefikss vault ceļam vai vaicājumam atkarībā no vault; ja tas nav norādīts, tiks izmantots vispiemērotākais noklusējuma iestatījums. Ņemiet vērā, ka prefiksam jāsākas un jābeidzas ar `/`.

Attention:
Konfigurācijas parametri `Vault`, `VaultToken` (vai `VaultAppRoleID`/`VaultAppSecretID`), `VaultURL` un `VaultPrefix` tiek izmantoti arī vault autentifikācijai, kad Zabbix serveris apstrādā slepenos vault makro (un Zabbix starpniekserveris, ja konfigurēts). Zabbix serveris un starpniekserveri neatvērs slepenos vault makro, kas satur DB akreditācijas datus no VaultDBPath.

Ir stingri ieteicams dažādiem starpniekserveriem izmantot atšķirīgus tokenus.

Zabbix serveris un Zabbix starpniekserveris palaišanas laikā nolasa ar vault saistītos konfigurācijas parametrus no `zabbix_server.conf` un `zabbix_proxy.conf`.

Turklāt Zabbix serveris un Zabbix starpniekserveris palaišanas laikā vienu reizi nolasa vides mainīgo `VAULT_TOKEN` un noņem to, lai tas nebūtu pieejams, izmantojot forkotos skriptus; ja gan `VaultToken`, gan `VAULT_TOKEN` parametri satur vērtību, tā ir kļūda.

Piemērs

1. Failā `zabbix_server.conf` norādiet šādus parametrus.

Izmantojot VaultToken:

```
Vault=HashiCorp
VaultToken=hvs.CAESIIG_PILmULFYOsEyWHxkZ2mF2a8VPKNLE8eHqd4autYGGh4KHGh2cy5aeTYONFNSaUp3ZnpWbDF1RUNjUkNTZEg
VaultURL=https://127.0.0.1:8200
VaultDBPath=database
VaultPrefix=/v1/secret/data/zabbix/
```

Vai arī, izmantojot VaultAppRoleID:

```
Vault=HashiCorp
VaultAppRoleID=6f1567f1-538d-7d70-e8e4-5d3e59471b2e
VaultAppSecretID=89dac8b1-f7d6-96a7-2ecf-f94d2de3d6f1
VaultURL=https://127.0.0.1:8200
VaultDBPath=database
VaultPrefix=/v1/secret/data/zabbix/
```

2. Izpildiet šādas CLI komandas, lai vault izveidotu nepieciešamo slepeno informāciju:

```
#### Iespējojiet "secret/" mount point, ja tas vēl nav iespējots; ņemiet vērā, ka jāizmanto "kv-v2".
vault secrets enable -path=secret/ kv-v2
```

```
#### Ievietojiet jaunus noslēpumus ar atslēgām username un password mount point "secret/" un ceļā "zabbix/
vault kv put -mount=secret zabbix/database username=zabbix password=<password>

#### Pārbaudiet, vai slepenā informācija ir veiksmīgi pievienota.
vault kv get secret/zabbix/database

#### Visbeidzot pārbaudiet ar Curl; ņemiet vērā, ka "data" ir manuāli jāpievieno pēc mount point un "/v1"
curl --header "X-Vault-Token: <VaultToken>" https://127.0.0.1:8200/v1/secret/data/zabbix/database
```

3. Rezultātā Zabbix serveris autentifikācijai datubāzē izgūs šādus akreditācijas datus:

- Lietotājvārds: zabbix
- Parole: <password>

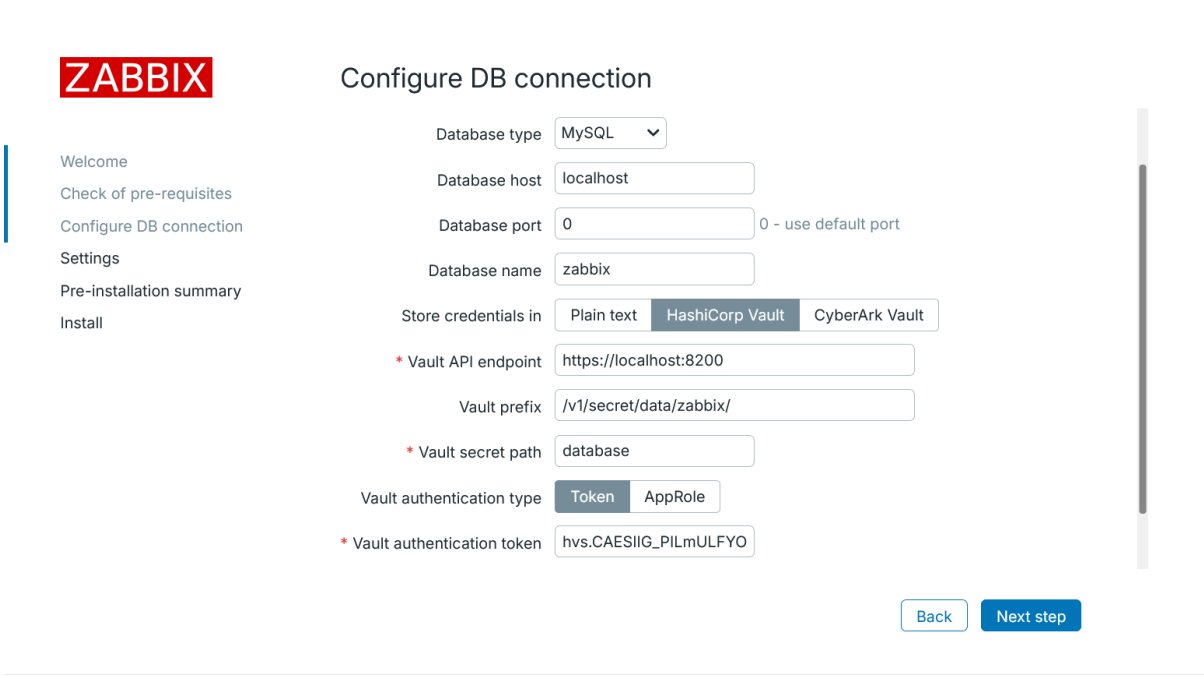
Lietotāja saskarne

Zabbix lietotāja saskarni var konfigurēt, lai tā izgūtu datubāzes akreditācijas datus no glabātuves vai nu lietotāja saskarnes instalēšanas laikā, vai atjauninot lietotāja saskarnes konfigurācijas failu (`zabbix.conf.php`).

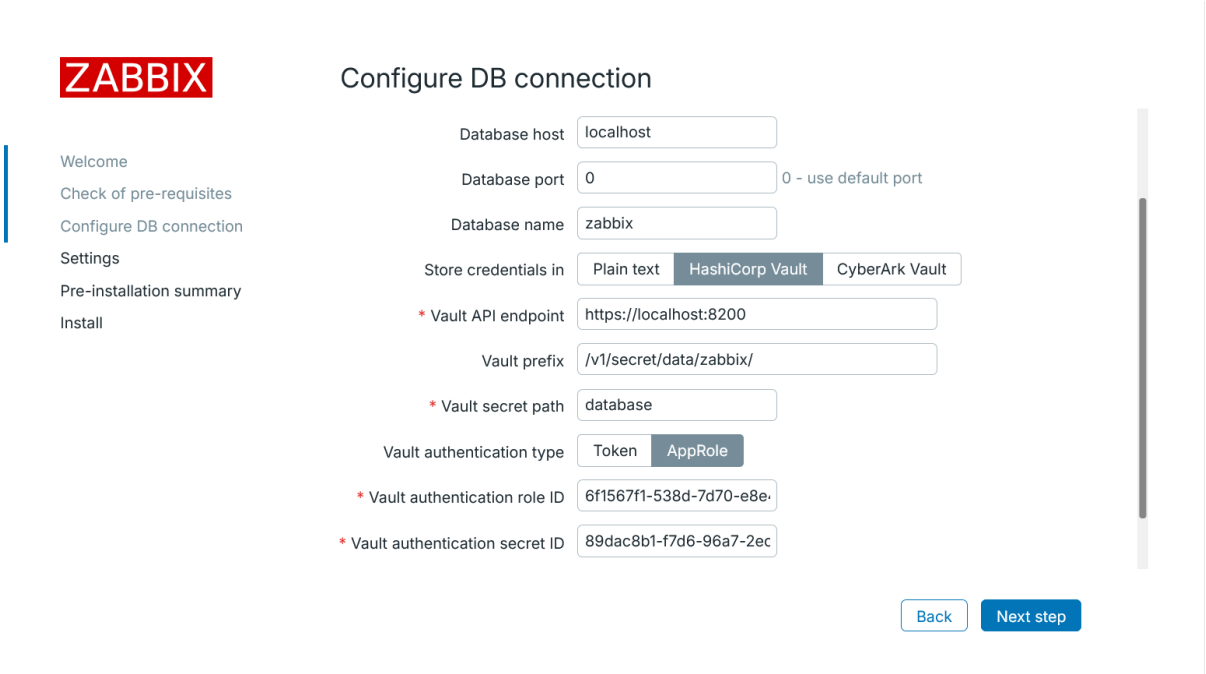
Attention:
Ja glabātuves akreditācijas dati kopš iepriekšējās lietotāja saskarnes instalēšanas ir mainīti, atjauniniet esošo konfigurāciju.

Lietotāja saskarnes instalēšanas laikā savienojuma parametri jānorāda darbībā *Configure DB Connection*.

Konfigurācijas piemērs, izmantojot marķiera autentifikāciju:



Konfigurācijas piemērs, izmantojot AppRole autentifikāciju:

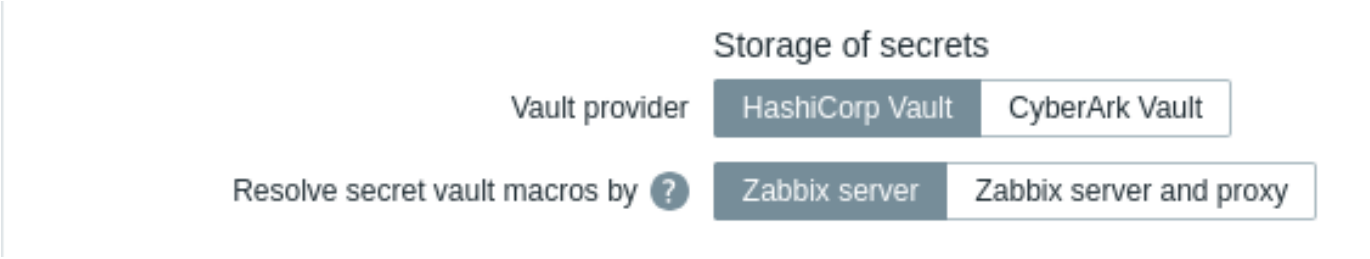


Parameter	Mandatory	Default value	Description
Store credentials in	no		Atlasiet **HashiCorp Vault**.
Vault API endpoint	yes	https://localhost:8200	Norādiet URL savienojuma izveidei ar glabātuvi formātā `scheme://host:port`
Vault prefix	no	/v1/secret/data/	Norādiet pielāgotu prefiksu glabātuves ceļam vai vaicājumam. Ja tas nav norādīts, tiek izmantota noklusējuma vērtība. Piemērs: `/v1/secret/data/zabbix/`
Vault secret path	yes		Ceļš uz noslēpumu, no kura datubāzes akreditācijas dati jāizgūst, izmantojot atslēgas ”password” un ”username”. Piemērs: `database`
Vault authentication type	no		Atlasiet glabātuves autentifikācijas veidu: **Token** - izmantot marķiera autentifikāciju; **AppRole** - izmantot AppRole autentifikācijas metodi.
Vault authentication token	yes		Norādiet autentifikācijas marķieri tikai lasīšanas piekļuvei noslēpuma ceļam. Informāciju par marķieru un glabātuves politiku izveidi skatiet HashiCorp dokumentācijā. Šis lauks ir pieejams tikai tad, ja kā glabātuves autentifikācijas veids ir atlasīts *Token*.
Vault authentication role ID	yes		Norādiet AppRole lomas ID. Informāciju par AppRole skatiet HashiCorp dokumentācijā. Šis lauks ir pieejams tikai tad, ja kā glabātuves autentifikācijas veids ir atlasīts *AppRole*.
Vault authentication secret ID	yes		Norādiet AppRole izsniegto noslēpuma ID. Informāciju par AppRole skatiet HashiCorp dokumentācijā. Šis lauks ir pieejams tikai tad, ja kā glabātuves autentifikācijas veids ir atlasīts *AppRole*.

Lietotāja makro vērtību izgūšana

Lai izmantotu HashiCorp Vault *Vault secret* lietotāja makro vērtību glabāšanai, pārliecinieties, ka:

- Zabbix serveris/starpniekserveris ir konfigurēts darbam ar HashiCorp Vault.
- Parametrs *Vault provider* sadaļā *Administration > General > Other* ir iestatīts uz ”HashiCorp Vault” (noklusējums).



Note:
Zabbix serverim (un Zabbix starpniekserverim, ja tas ir konfigurēts) ir nepieciešama piekļuve *Vault secret* makro vērtībām no seifa. Zabbix lietotāja saskarnei šāda piekļuve nav nepieciešama.

Makro vērtībai jāietver atsauces ceļš (piemēram, `path:key`, piemēram, `macros:password`). Autentifikācijas marķierim, kas norādīts Zabbix servera/starpniekservera konfigurācijas laikā (izmantojot parametru `VaultToken`), ir jānodrošina tikai lasīšanas piekļuve šim ceļam.

Skatiet *Vault secret macros* detalizētai informācijai par makro vērtību apstrādi Zabbix.

Ceļa sintakse

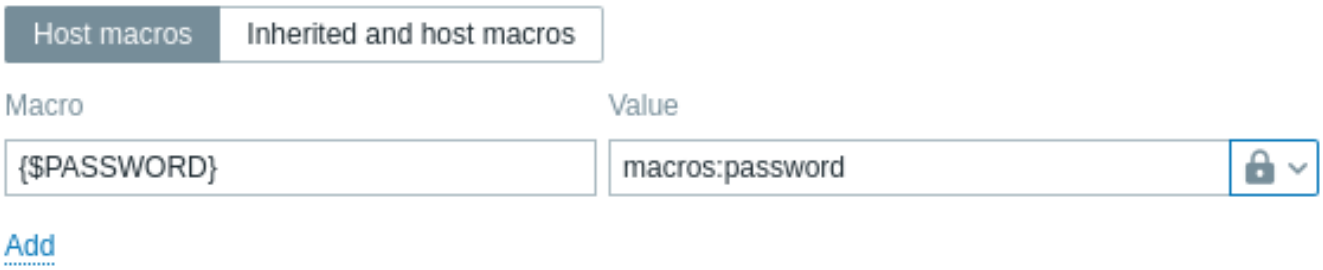
Simboli slīpsvītra (”/”) un kols (”:”) ir rezervēti.

Slīpsvītru var izmantot tikai, lai atdalītu montēšanas punktu no ceļa (piem., *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 un kola simbolus, ja ir nepieciešams izveidot montēšanas punktu ar nosaukumu, kas satur slīpsvītru (piem., *foo/bar/zabbix*, kur montēšanas punkts ir ”foo/bar” un ceļš ir ”zabbix”, var kodēt kā ”foo%2Fbar/zabbix”), kā arī tad, ja montēšanas punkta nosaukumā vai ceļā ir jāiekļauj kols.

Piemērs

1. Zabbix pievienojiet lietotāja makro {$PASSWORD} ar tipu ”Vault secret” un vērtību `macros:password`



2. Palaidiet šādas CLI komandas, lai izveidotu nepieciešamo noslēpumu vault:

```
#### Enable "secret/" mount point if not already enabled; note that "kv-v2" must be used.
vault secrets enable -path=secret/ kv-v2

#### Put new secret with key "password" under mount point "secret/" and path "zabbix/macros".
vault kv put -mount=secret zabbix/macros password=<password>

#### Test that secret is successfully added.
vault kv get secret/zabbix/macros

#### Finally test with Curl; note that "data" need to be manually added after mount point and "/v1" before
curl --header "X-Vault-Token: <VaultToken>" https://127.0.0.1:8200/v1/secret/data/zabbix/macros
```

3. Rezultātā Zabbix atrisinās makro {$PASSWORD} uz vērtību: <password>

Esošās konfigurācijas atjaunināšana

Lai atjauninātu esošu konfigurāciju noslēpumu izgūšanai no HashiCorp Vault:

1. Atjauniniet Zabbix servera vai starpniekservera konfigurācijas faila parametrus, kā aprakstīts sadaļā *Datubāzes akreditācijas dati*.
2. Atjauniniet Zabbix lietotāja saskarnes konfigurācijas faila parametrus:

Izmantojot Vault pilnvaru:

```
$DB['VAULT']                    = 'HashiCorp';
$DB['VAULT_URL']                = 'https://localhost:8200';
$DB['VAULT_DB_PATH']            = 'database';
$DB['VAULT_TOKEN']              = 'hvs.CAESIIG_PILmULFYOsEyWHxkZ2mF2a8VPKNLE8eHqd4autYGGh4KHGh2cy5aeTY0NFN
$DB['VAULT_PREFIX']             = '/v1/secret/data/zabbix/';
```

Vai arī, izmantojot Vault AppRole metodi:

```
$DB['VAULT']                    = 'HashiCorp';
$DB['VAULT_URL']                = 'https://localhost:8200';
$DB['VAULT_DB_PATH']            = 'database';
$DB['VAULT_APP_ROLE_ID']        = '6f1567f1-538d-7d70-e8e4-5d3e59471b2e';
$DB['VAULT_APP_SECRET_ID']      = '89dac8b1-f7d6-96a7-2ecf-f94d2de3d6f1';
$DB['VAULT_PREFIX']             = '/v1/secret/data/zabbix/';
```

3. Ja nepieciešams, konfigurējiet lietotāja makro, kā aprakstīts sadaļā *Lietotāja makro vērtības*.

Lai atjauninātu esošu konfigurāciju noslēpumu izgūšanai no CyberArk Vault, skatiet sadaļu *CyberArk konfigurācija*.

Problēmu novēršana

1. Zabbix servera/starpniekservera žurnālfailā:

```
237839:20260729:130916.425 Vault token renew started to fail: cannot login into HashiCorp vault with AppRo
```

Iespējamais iemesls: HashiCorp Vault ir (atkārtoti) palaists, bet nav atbloķēts (tas ir nepieciešams noklusējuma konfigurācijā, bet nav nepieciešams, ja ir konfigurēts Auto Unseal).

2. Zabbix servera/starpniekservera žurnālfailā:

```
235545:20260729:125617.944 Vault token renew started to fail: cannot login into HashiCorp vault with AppRo
```

vai

```
235545:20260729:125718.537 Vault token renew failed: cannot login into HashiCorp vault with AppRole method
```

Iespējamais iemesls: VaultAppSecretID ir beidzies derīguma termiņš vai sasniegts tā lietojumu limits HashiCorp vault.

Google Chrome TLS sertifikāta uzticamība

Pārskats

Šajā lapā ir sniegti Zabbix iestatīšanas soļi un konfigurācijas piemēri Google Chrome lietošanai ar Zabbix lietotāja saskarni vai zabbix-web-service galapunktiem, kas ir aizsargāti ar pašparakstītu sertifikātu vai privātu sertifikātu izdevēju.

Šie norādījumi pieņem, ka mērķa tīmekļa serveris jau ir konfigurēts HTTPS lietošanai. Lai konfigurētu TLS Zabbix lietotāja saskarnei, skatiet Drošs savienojums ar lietotāja saskarni.

Konfigurācija

Google Chrome operētājsistēmā Linux uzticamiem sertifikātiem izmanto katram lietotājam atsevišķu NSS sertifikātu datubāzi. Lai Chrome uzticētos pašparakstītam sertifikātam vai privātai sertifikātu iestādei, pievienojiet nepieciešamos sertifikātus tās lietotāja konta NSS datubāzei, kas palaiž Chrome.

1. Instalējiet nepieciešamās pakotnes.

Debian/Ubuntu sistēmām:

```
sudo apt install ca-certificates libnss3-tools
```

RHEL bāzētām sistēmām:

```
sudo dnf install ca-certificates nss-tools
```

Vai:

```
sudo yum install ca-certificates nss-tools
```

2. Sagatavojiet sertifikātu failus.

Katram nepieciešamajam sertifikātam: izveidojiet uz PEM balstītu `.crt` failu pašparakstītajam sertifikātam vai izveidojiet atsevišķus uz PEM balstītus `.crt` failus saknes sertifikātu iestādes sertifikātam un, ja nepieciešams, katram starpsertifikātam.

3. Izveidojiet NSS datubāzes direktoriju.

Izmantojiet tās lietotāja mājas direktoriju, kas palaiž Chrome.

Piemēram:

```
sudo mkdir -p /var/lib/zabbix/.pki/nssdb
sudo -u zabbix certutil -N -d sql:/var/lib/zabbix/.pki/nssdb
sudo chown -R zabbix:zabbix /var/lib/zabbix/.pki/nssdb
```

Ja Chrome mērķa sistēmā izmanto citu NSS datubāzes atrašanās vietu, izmantojiet šo direktoriju.

Pārliecinieties, ka sertifikātu faili ir lasāmi lietotājam, kas palaiž Google Chrome. Ja nepieciešams, nokopējiet tos uz vietu, kurai šis lietotājs var piekļūt.

4. Importējiet sertifikātus.

Ja izmantojat pašparakstītu sertifikātu:

```
sudo -u zabbix certutil -d sql:/var/lib/zabbix/.pki/nssdb \
  -A -t "P,," \
  -n "Zabbix self-signed certificate" \
  -i /path/to/self-signed.crt
```

Ja izmantojat privātu sertifikātu iestādi, importējiet saknes sertifikātu iestādes sertifikātu datubāzē:

```
sudo -u zabbix certutil -d sql:/var/lib/zabbix/.pki/nssdb \
  -A -t "C,," \
  -n "Zabbix root certificate authority" \
  -i /path/to/root-ca.crt
```

Ja tiek izmantoti kādi starpsertifikāti, importējiet katru atsevišķi:

```
sudo -u zabbix certutil -d sql:/var/lib/zabbix/.pki/nssdb \
  -A -t ",," \
  -n "Zabbix intermediate certificate" \
  -i /path/to/intermediate-ca.crt
```

5. Pārbaudiet sertifikātu datubāzi.

```
sudo -u zabbix certutil -d sql:/var/lib/zabbix/.pki/nssdb -L
```

6. Pārbaudiet Chrome.

Google Chrome izmanto tās lietotāja konta sertifikātu datubāzi, kas to palaiž. Ja lietotāja kontam nav rakstāma mājas direktorija, pirms testa palaišanas konfigurējiet rakstāmu mājas direktoriju un XDG direktorijus.

Piemērs (aizstājiet `google-chrome` ar `chromium`, ja tiek izmantots Chromium.):

```
sudo -u zabbix env \
  HOME=/var/lib/zabbix-home \
  XDG_CONFIG_HOME=/var/lib/zabbix-home/.config \
  XDG_DATA_HOME=/var/lib/zabbix-home/.local/share \
  XDG_CACHE_HOME=/var/lib/zabbix-home/.cache \
  google-chrome --headless --disable-gpu \
  --user-data-dir=/var/lib/zabbix-home/chrome-profile \
  --dump-dom https://FQDN_OF_ZABBIX_SERVER
```

Ja konfigurācija ir pareiza, komanda atgriež pieprasītās lapas HTML.

Ja Chrome neuzticas sertifikātam, komanda atgriež sertifikāta kļūdas lapu ar tādiem ziņojumiem kā `NET::ERR_CERT_AUTHORITY_INVALID`.

Problēmu novēršana

Kļūda	Iespējamais iemesls
`NET::ERR_CERT_AUTHORITY_INVALID`	Sertifikāts importēts nepareizajā lietotāja profilā, trūkst saknes sertifikātu iestādes sertifikāta vai trūkst starpsertifikāta.
`NET::ERR_CERT_COMMON_NAME_INVALID`	Nederīgs nosaukums. Sertifikāts neatbilst resursdatora nosaukumam, kas izmantots pārlūkprogrammas URL.
`NET::ERR_CERT_DATE_INVALID`	Sertifikāta derīguma termiņš ir beidzies, tas vēl nav derīgs vai sistēmas pulkstenis ir nepareizs.

Kļūda	Iespējamais iemesls
Chrome joprojām rāda sertifikāta brīdinājumu pēc importēšanas	Chrome pašreizējam lietotājam izmanto citu NSS datubāzi vai sertifikāta uzticamības karodziņi ir nepareizi.

Drošības labākā prakse

Pārskats Šajā sadaļā ir sniegta informācija par labāko praksi Zabbix drošai iestatīšanai.

Šajā sadaļā aprakstītās prakses nav nepieciešamas Zabbix darbībai, taču tās ir ieteicamas labākai sistēmas drošībai.

Lai uzzinātu par labāko praksi Zabbix mobilās lietotnes un Zabbix bridge adaptera drošības nodrošināšanai, skatiet Droša mobilā piekļuve.

UTF-8 kodējums UTF-8 ir vienīgais Zabbix atbalstītais kodējums. Ir zināms, ka tas darbojas bez jebkādām drošības nepilnībām. Lietotājiem jāņem vērā, ka, izmantojot dažus citus kodējumus, ir zināmas drošības problēmas.

Windows instalētāja ceļi Izmantojot Windows instalētājus, ieteicams lietot instalētāja nodrošinātos noklusējuma ceļus. Pielāgotu ceļu izmantošana bez atbilstošām atļaujām var apdraudēt instalācijas drošību.

Makrokomandas lietotāja definētajos globālajos skriptos Lai uzlabotu drošību, ieteicams lietot makro funkcijas parasto makrokomandu vietā lietotāja definētajos globālajos skriptos, jo makrokomandas netiek automātiski ekrānētas.

Zabbix drošības ieteikumi un CVE datubāze Skatiet *Zabbix drošības ieteikumi un CVE datubāze*.

HTML e-pasta veidnes Veidojot vai rediģējot ziņojumu veidnes, kas tiek izmantotas HTML e-pastiem, vienmēr aptveriet katru makro ar `htmlencode()` makro 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
```

`htmlencode()` izmantošana nodrošina, ka visi HTML simboli makro vērtībās tiek aizstāti ar atbilstošām entītijām, tādējādi novēršot HTML koda ievietošanu paziņojumos (piemēram, uzbrucējam ievietojot paziņojumā ļaunprātīgu vai pikšķerēšanas saiti).

Piezīme: Zabbix nodrošinātajos noklusējuma HTML e-pasta ziņojumos makro jau tiek apstrādāti ar `htmlencode()`. Šīs vadlīnijas attiecas uz esošu veidņu rediģēšanu vai jaunu veidņu izveidi — pirms veidnes izmantošanas HTML e-pasta nosūtīšanai pārbaudiet, vai makro ir kodēti.

Lietotāja saskarne

Pārskats Lai ērti piekļūtu Zabbix no jebkuras vietas un platformas, ir pieejama tīmekļa lietotāja saskarne.

> **Note:**
> Ja tiek izmantas vairākas lietotāja saskarnes instances, pārliecinieties, ka lokalizācijas un bibliotēkas (LDAP, SAML u. c.) visās lietotāja saskarnēs ir instalētas un konfigurētas identiski.

Lietotāja saskarnes palīdzība Zabbix lietotāja saskarnes sadaļās un veidlapās augšējā labajā stūrī ir pieejama palīdzības saite ?, kas tieši novirza uz atbilstošajām dokumentācijas sadaļām.

Novietojot kursoru virs palīdzības ikonas ● veidlapas laukos, tiek parādīts atbilstošais padoms.

Kioska režīms Lietotāja saskarnes sadaļās, kurās ir poga *Kioska režīms* (atrodas augšējā labajā stūrī), noklikšķiniet uz pogas, lai pārslēgtu sadaļu kioska režīmā. Kioska režīmā tiek rādīts tikai sadaļas saturs.

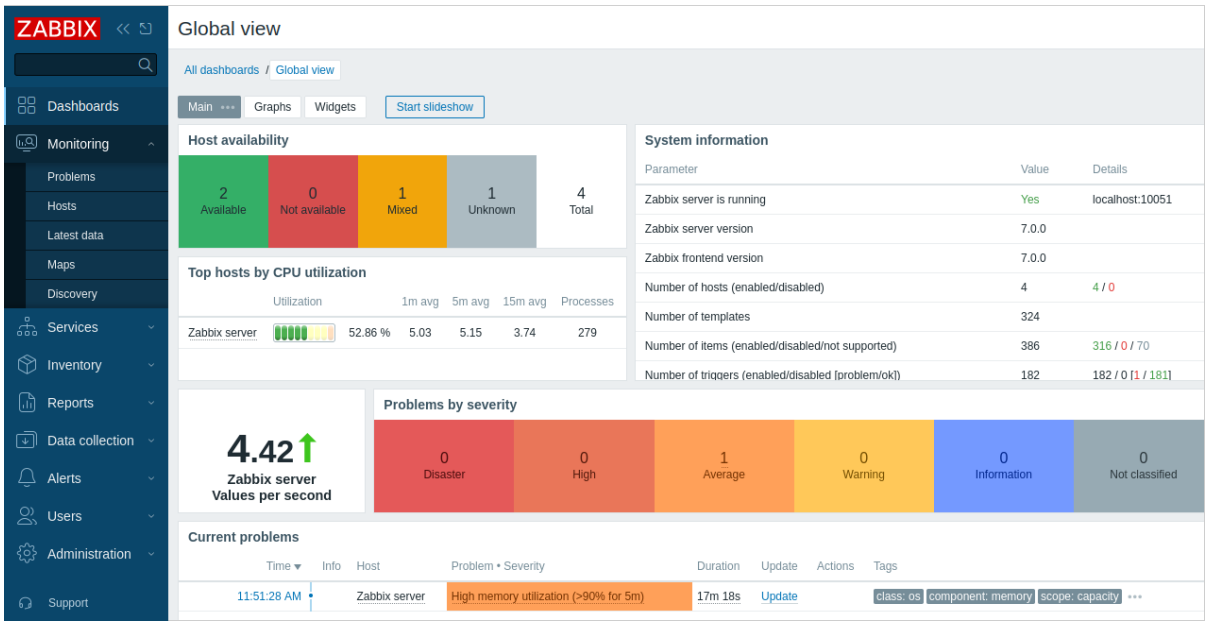
Lai izietu no kioska režīma, novietojiet kursoru virs satura un noklikšķiniet uz pogas *Parastais skats*, kas tiek parādīta augšējā labajā stūrī.

Izvēlne

Pārskats

Vertikāla izvēlne sānjoslā nodrošina piekļuvi dažādām Zabbix lietotāja saskarne sadaļām.

Izvēlne noklusējuma tēmā ir tumši zila.

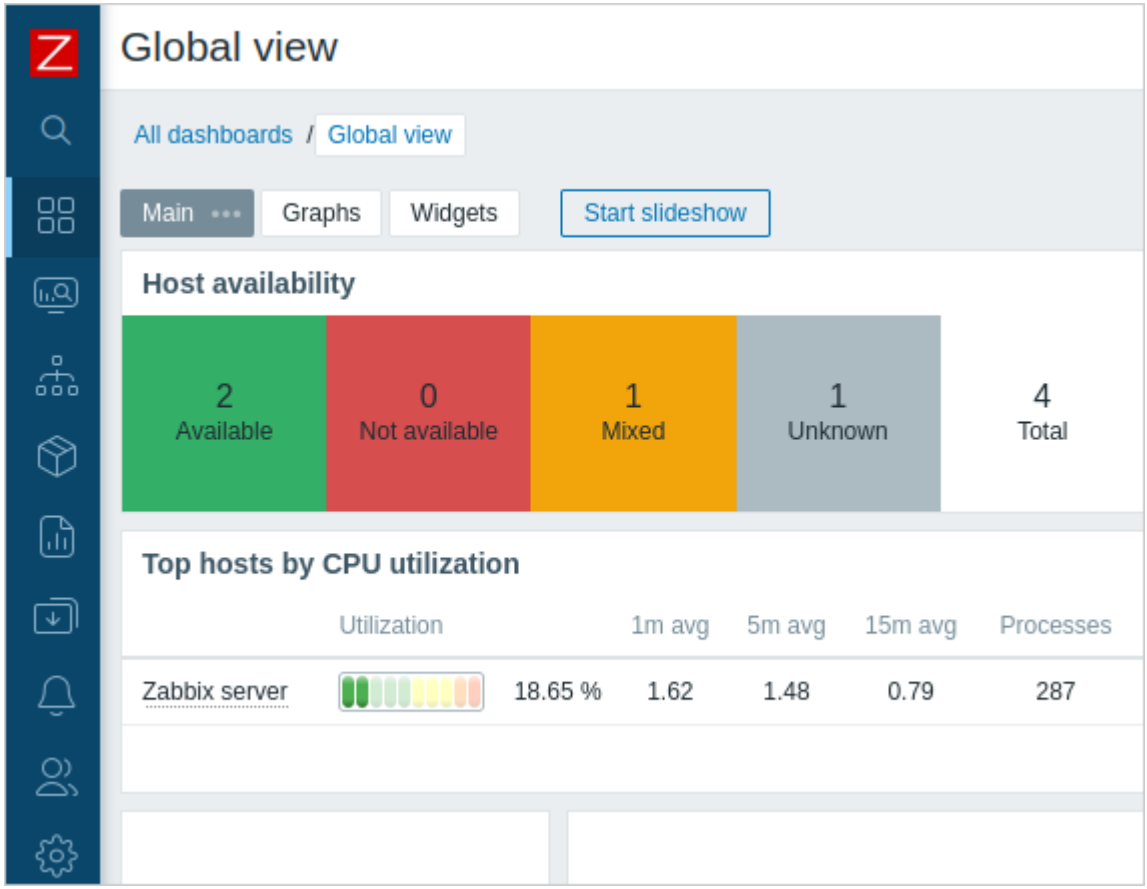


Darbs ar izvēlni

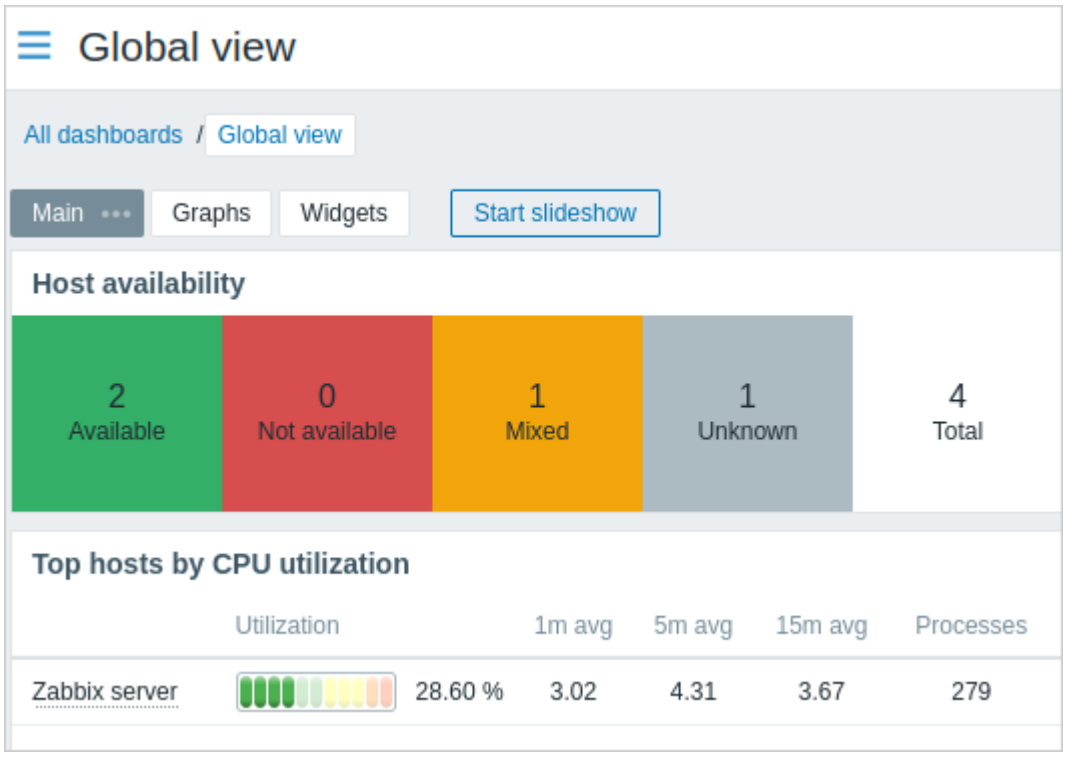
globālās meklēšanas lauks atrodas zem Zabbix logotipa.

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.



Sakļauta izvēlne

Kad izvēlne ir sakļauta, rādot tikai ikonas, pilna izvēlne atkal tiek parādīta, tiklīdz peles kursors tiek novietots uz tās. Ņemiet vērā, ka tā tiek parādīta virs lapas satura; lai pārvietotu lapas saturu pa labi, jānoklikšķina uz izvēršanas pogas. Ja peles kursors atkal tiek novietots ārpus pilnās izvēlnes, izvēlne pēc divām sekundēm atkal tiks sakļauta.

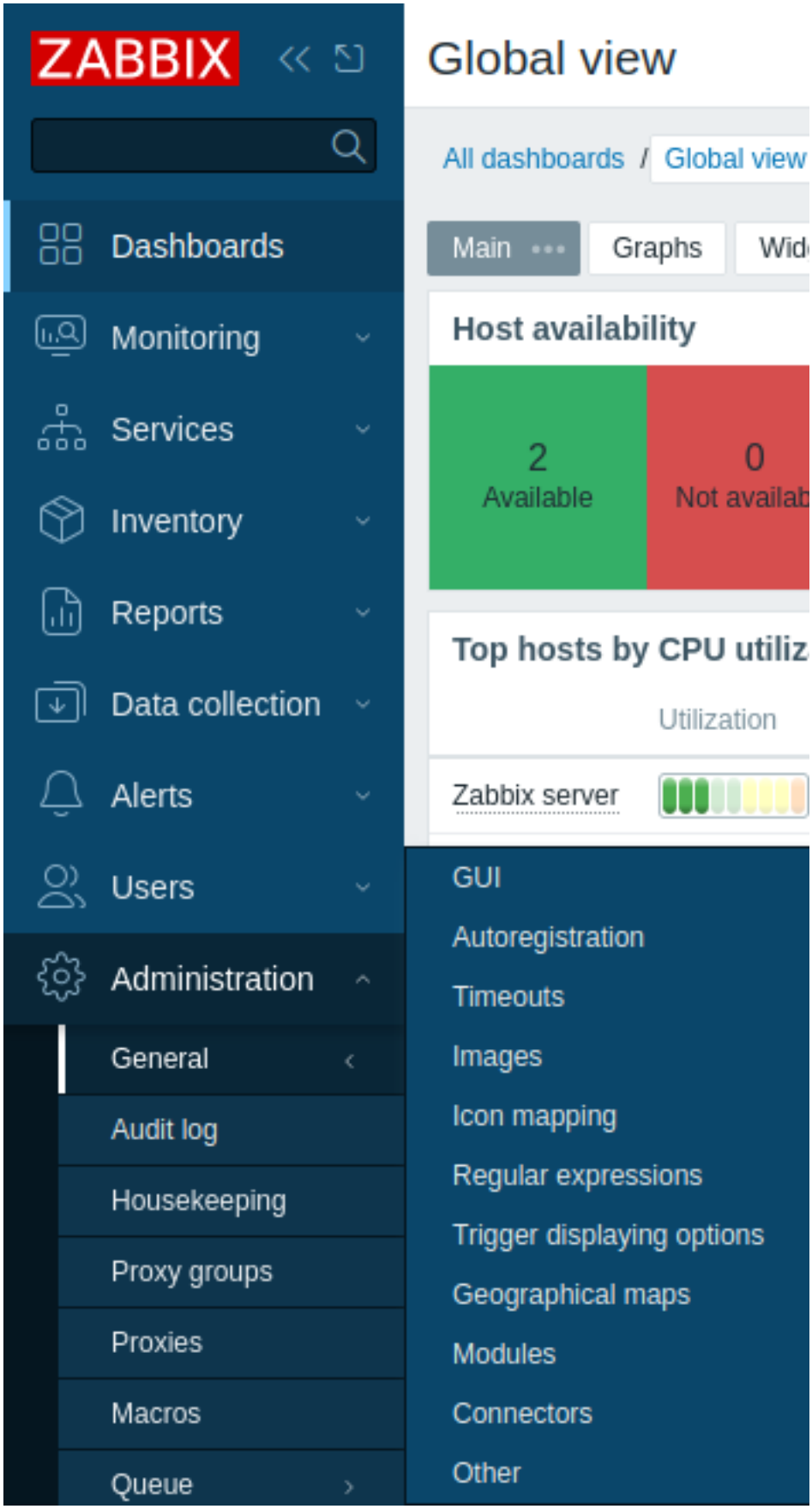
Sakļautu izvēlni var pilnībā parādīt arī, nospiežot taustiņu Tab. Atkārtoti nospiežot taustiņu Tab, varat pārvietot fokusu uz nākamo izvēlnes elementu.

Slēptā 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 jāatceļ izvēlnes slēpšana, noklikšķinot uz pogas ”show sidebar”.

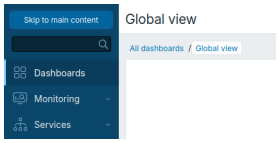
Izvēlnes līmeņi

Izvēlnē var būt līdz trim līmeņiem.



Pieejamība: Pāriet pāri navigācijai

Saite *Pāriet uz galveno saturu* ļauj tastatūras un ekrānlasītāja lietotājiem izlaist atkārtotu sānjoslas navigāciju un pāriet tieši uz galveno saturu.



Parastas lietošanas laikā tā paliek paslēpta un tiek parādīta tikai tad, kad tai tiek piešķirts fokuss, piemēram, kad pēc lapas ielādes tiek nospiests taustiņš Tab vai kad tai piekļūst, izmantojot ekrānlasītāju.

Konteksta izvēlnes

Papildus galvenajai izvēlnei Zabbix nodrošina konteksta izvēlnes host, vienums un event ātrai piekļuvei bieži izmantotām entītijām, piemēram, jaunākajām vērtībām, vienkāršam grafikam, konfigurācijas formai, saistītajiem skriptiem vai ārējām saitēm.

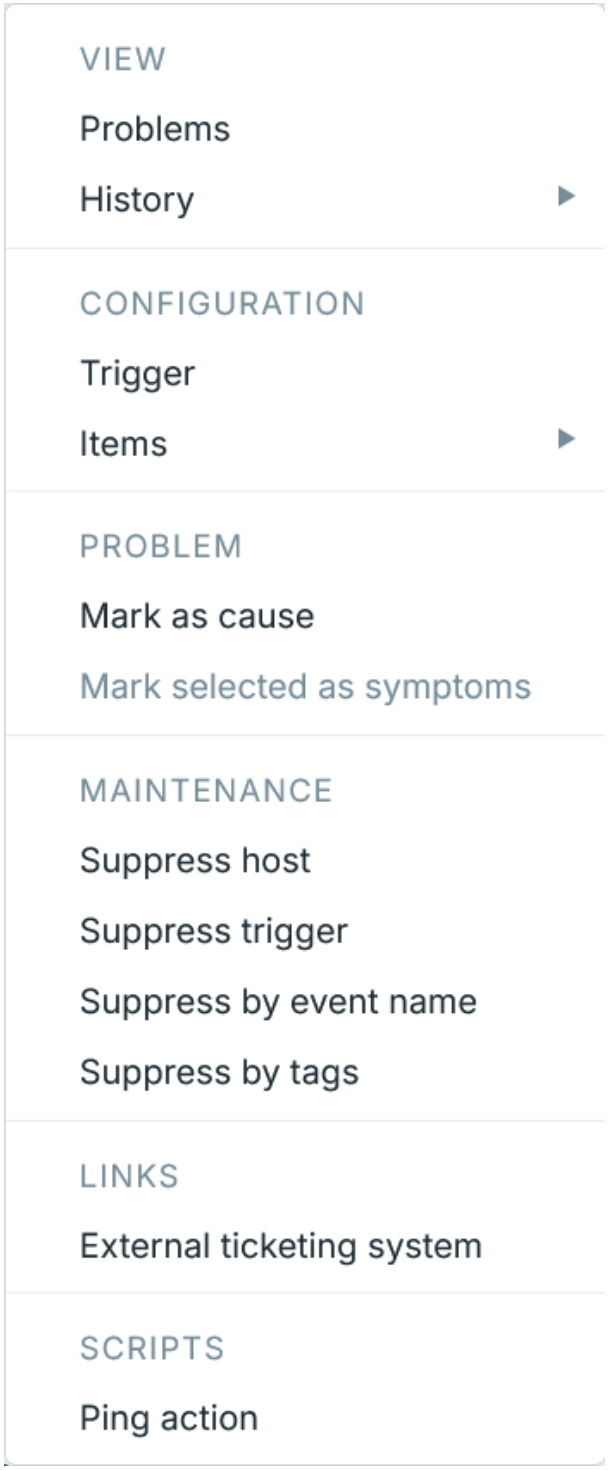
Konteksta izvēlnes ir pieejamas, noklikšķinot uz host, vienums vai problēmas/trigeris nosaukuma atbalstītajās vietās.

Notikuma izvēlne

Pārskats

Notikuma izvēlne satur īsceļus uz darbībām vai lietotāja saskarnes sadaļām, kas bieži nepieciešamas notikumam.

Notikuma izvēlni var atvērt, noklikšķinot uz notikuma nosaukuma.



Saturs

Notikuma konteksta izvēlnei ir vairākas sadaļas: *Skatīt*, *Darbības*, *Konfigurācija*, *Problēma*, *Uzturēšana*, *Saites* un *Skripti*. Entītijām, kas nav konfigurētas, saites ir atspējotas un tiek parādītas pelēkā krāsā. Sadaļas *Skripti* un *Saites* tiek parādītas, ja to entītijas ir konfigurētas.

Sadaļā *Skatīt* ir saites uz:

- **Problēmas** - atver pamatā esošā trigeri neatrisināto problēmu sarakstu;
- **Vēsture** - atver pamatā esošā(-o) vienuma(-u) diagrammu/vienuma vēsturi sadaļā *Jaunākie dati*. Ja trigeris izmanto vairākus vienumus, katram no tiem būs pieejama atsevišķa saite.

Sadaļa *Darbības* ir pieejama tikai *Trigeru pārskata* logrīkos. Tajā ir saite uz:

- **Atjaunināt problēmu** - atver problēmas atjaunināšanas ekrānu.

Sadaļā *Konfigurācija* ir saites uz tālāk norādīto konfigurāciju:

- **Trigeris**, kas izraisīja problēmu;
- **Vienumi**, kas izmantoti trigeru izteiksmē.

Note:
Ņemiet vērā, ka konfigurācijas sadaļa ir pieejama tikai administratora un superadministratora lietotājiem.

Sadaļā *Problēma* ir pieejamas šādas opcijas:

- **Atzīmēt kā cēloni** - atzīmēt problēmu kā cēloni;
- **Atzīmēt atlasītās kā simptomus** - atzīmēt atlasītās problēmas kā šīs problēmas simptomus.

Sadaļā *Uzturēšana* ir pieejamas opcijas ad hoc uzturēšanas sākšanai:

- **Apturēt hostu** - sākt uzturēšanas periodu hostam;
- **Apturēt trigeri** - sākt uzturēšanas periodu trigerim;
- **Apturēt pēc notikuma nosaukuma** - sākt uzturēšanas periodu notikuma nosaukumam;
- **Apturēt pēc tagiem** - sākt uzturēšanas periodu notikumiem ar atbilstošajiem tagiem.

Atlasot attiecīgo opciju, tiks atvērta uzturēšanas perioda konfigurācijas veidlapa.

Pēc noklusējuma tiks atlasīts vienreizējs vienas stundas uzturēšanas periods (konfigurējams lietotāja iestatījumos un profilā), kas sākas no šī brīža. Tiks iepriekš atlasīts arī attiecīgais hosts/trigeris/notikuma nosaukums/tagi. Uzturēšanas nosaukums tiks automātiski aizpildīts formātā `Ad-hoc: <event name>`.

Sadaļā *Saites* ir saites uz:

- konfigurētu trigera URL;
- pielāgotām saitēm, kas konfigurētas sadaļā Globālie skripti (ar tvērumu "Manuāla notikuma darbība" un tipu "URL");
- konfigurētu ārēju biļeti problēmai (skatiet opciju *Iekļaut notikuma izvēlnes ierakstu*, konfigurējot webhook).

Sadaļā *Skripti* ir saites globāla skripta izpildei (ar tvērumu *Manuāla notikuma darbība*). Šī funkcija var būt noderīga, lai palaistu skriptus, ko izmanto problēmu biļešu pārvaldībai ārējās sistēmās.

Atbalstītās atrašanās vietas

Notikuma konteksta izvēlne ir pieejama, noklikšķinot uz problēmas vai notikuma nosaukuma dažādās lietotāja saskarnes sadaļās, piemēram:

- Informācijas paneļi logrīki, piemēram, *Problems* logrīks, *Trigger overview* logrīks u.c.
- *Monitoring* > *Problems*
- *Monitoring* > *Problems* > Notikuma informācija
- *Reports* > *Top 100 triggers* (globālie skripti un piekļuve ārējam biļešu sistēmas ierakstam šajā atrašanās vietā netiek atbalstīti)

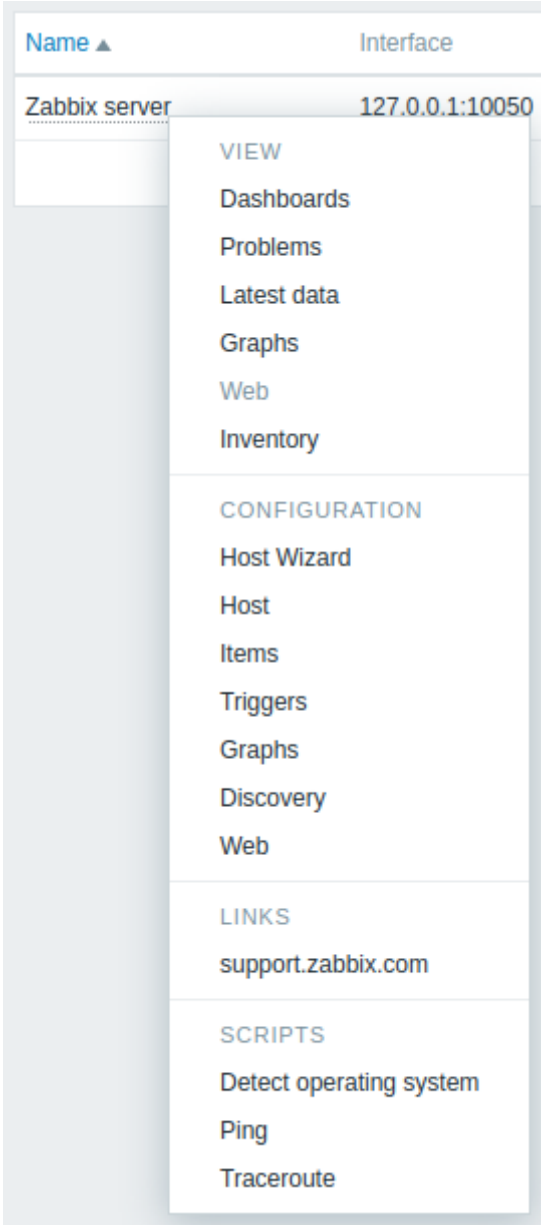
Hostu izvēlne

Pārskats

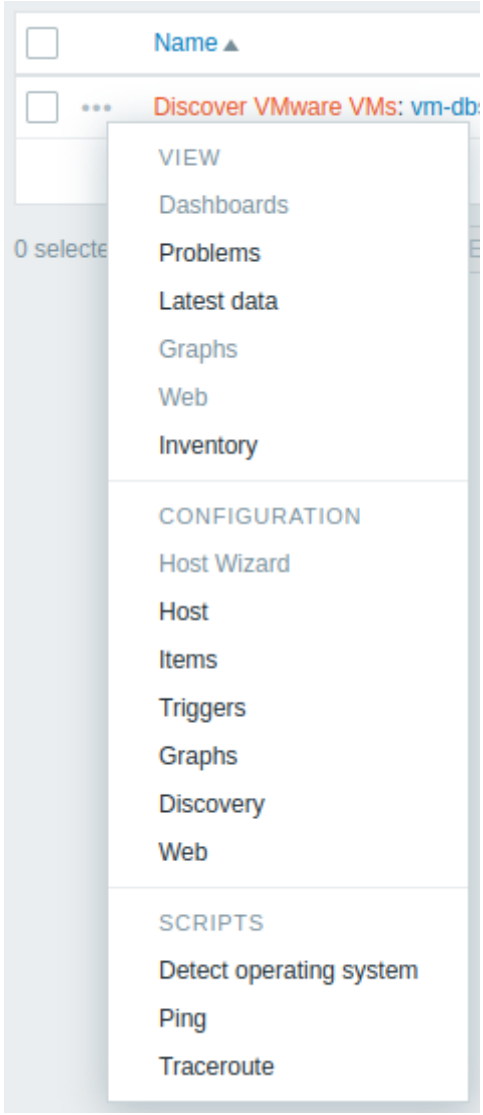
Hosts izvēlne satur saīsnes uz darbībām vai lietotāja saskarnes sadaļām, kas hostam ir bieži nepieciešamas.

Hosts izvēlni var atvērt, noklikšķinot uz hosta nosaukuma vai trīs punktu ikonas atkarībā no lietotāja saskarnes sadaļas, piemēram:

Monitoring > *Latest data*



Data collection > *Hosts*



Saturs

Hosts konteksta izvēlnei ir četras sadaļas: *Skatīt*, *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.

Sadaļā *Skatīt* ir saites uz:

- **Informācijas paneļi** - atver logrīkus un grafikus.
- **Problēmas** - atver sadaļu *Problēmas* ar pamatā esošā trigera neatrisināto problēmu sarakstu.
- **Jaunākie dati** - atver sadaļu *Jaunākie dati* ar visu 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.

Sadaļā *Konfigurācija* ir saites uz:

- **Hosta vednis** - atver Hosta vedni pašreizējam hostam (atspējots automātiski atklātiem hostiem).
- **Hosts** - pašreizējā hosta konfigurācijas veidlapa.
- **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 administratora un superadministratora lietotājiem.

Sadaļā *Saites* ir saites uz:

- konfigurētu trigera URL.
- pielāgotām saitēm, kas konfigurētas sadaļā Globālie skripti (ar tvērumu *Manuāla hosta darbība* un tipu 'URL').

Sadaļa *Skripti* ļauj izpildīt globālos skriptus, kas konfigurēti pašreizējam hostam. Lai tie būtu pieejami hosta izvēlnē, šiem skriptiem tvērumam ir jābūt iestatītam uz *Manuāla hosta darbība*.

Atbalstītās atrašanās vietas

Hosts izvēlne ir pieejama, noklikšķinot uz hosta nosaukuma dažādās lietotāja saskarnes sadaļās, piemēram:

- Paneļi logrīki, piemēram, *Problems*, *Top items*, *Trigger overview* u. c.
- *Monitoring* > *Problems*
- *Monitoring* > *Problems* > Notikuma informācija
- *Monitoring* > *Hosts*
- *Monitoring* > Hosts > *Web Monitoring*
- *Monitoring* > *Latest data*
- *Monitoring* > *Maps*
- *Inventory* > *Hosts*
- *Reports* > *Top 100 triggers*

Hosts izvēlne ir pieejama, noklikšķinot uz trīs punktu ikonas sadaļā Data collection > Hosts.

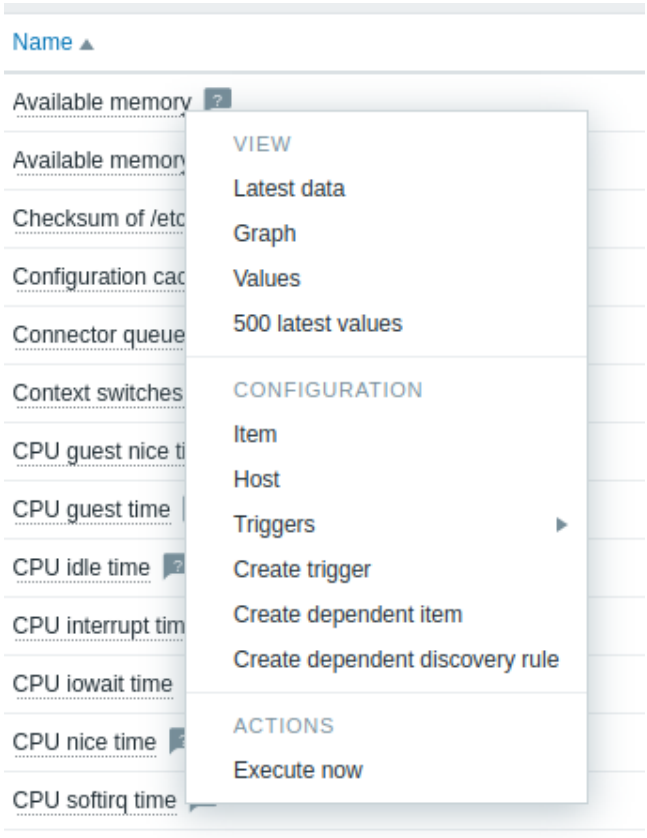
Vienuma izvēlne

Pārskats

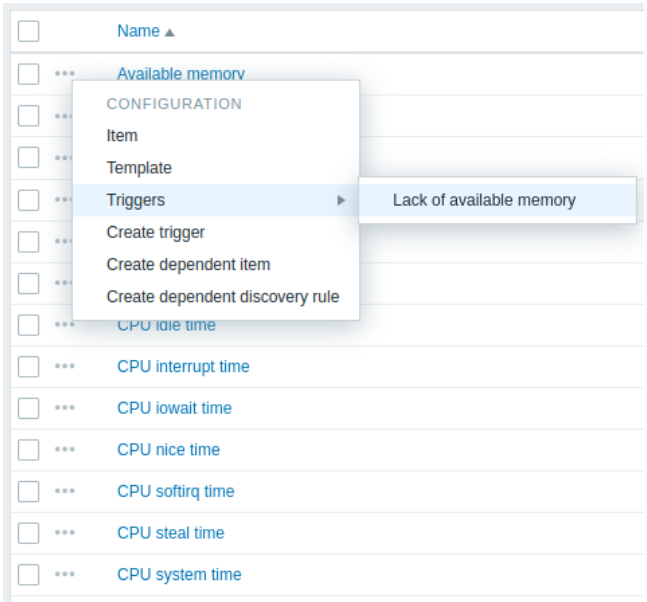
Vienuma izvēlne satur saīsnes uz darbībām vai lietotāja saskarnes sadaļām, kas vienumam bieži ir nepieciešamas.

Vienuma izvēlni var atvērt, noklikšķinot uz vienuma nosaukuma vai trīs punktu ikonas atkarībā no lietotāja saskarnes sadaļas, piemēram:

Monitoring > *Latest data*



Data Collection > *Templates* > *Items*



Saturs

Vienuma izvēlnei ir trīs sadaļas: *Skats*, *Konfigurācija* un *Darbības*.

Sadaļā *Skats* ir pieejamas šādas opcijas:

- **Jaunākie dati** - atver sadaļu *Jaunākie dati*, filtrējot pēc pašreizējā hosts un vienuma;
- **Grafiks** - atver pašreizējā vienuma vienkāršo grafiku;
- **Vērtības** - atver visu pašreizējā vienuma pēdējo 60 minūšu laikā saņemto vērtību sarakstu;
- **500 jaunākās vērtības** - atver pašreizējā vienuma 500 jaunāko vērtību sarakstu.

Sadaļā *Konfigurācija* (pieejama tikai *Administratora* un *Super administratora* tipa lietotājiem) ir pieejamas šādas opcijas:

- **Vienums** - atver pašreizējā vienuma konfigurācijas veidlapu;
- **Veidne** - atver veidnes, kurai pieder vienums, konfigurācijas veidlapu (pieejama tikai tad, ja vienuma izvēlnei piekļūst no *Datu apkopošana* > *Veidnes* > *Vienumi*);
- **Hosts** - atver hosts, kuram pieder vienums, konfigurācijas veidlapu;
- **Trigeri** - novietojot peles rādītāju virs opcijas, atver vienuma trigeru sarakstu, ja tādi ir; noklikšķinot uz trigera, tiek atvērta tā konfigurācijas veidlapa;
- **Izveidot trigeri** - atver trigera konfigurācijas veidlapu, lai izveidotu trigeri šim vienumam;
- **Izveidot atkarīgo vienumu** - atver vienuma konfigurācijas veidlapu, lai izveidotu atkarīgo vienumu, par galveno vienumu izmantojot pašreizējo vienumu;
- **Izveidot atkarīgo atklāšanas noteikumu** - atver atklāšanas noteikuma konfigurācijas veidlapu, lai izveidotu atkarīgo atklāšanas noteikumu, par galveno vienumu izmantojot pašreizējo vienumu.

Sadaļā *Darbības* ir pieejama šāda opcija:

- **Izpildīt tūlīt** - nekavējoties izpilda pārbaudi, lai iegūtu jaunu vienuma vērtību.

Atbalstītās vietas

Vienuma izvēlne ir pieejama, noklikšķinot uz vienuma nosaukuma dažādās lietotāja saskarnes sadaļās, piemēram:

- *Monitoring* > *Latest data*
- *Data collection* > *Hosts* > *Items*
- *Data collection* > *Hosts* > *Discovery rules* > *Item prototypes*

Vienuma izvēlne ir pieejama, noklikšķinot uz vienuma vērtības informācijas paneļa logrīkā *Top items*.

Lietotāja saskarnes sadaļas

Izvēļņu struktūra Zabbix lietotāja saskarnes izvēlnei ir šāda struktūra:

- Informācijas paneļi
- Uzraudzība
  - Problēmas
  - Hosti
  - Jaunākie dati
  - Kartes
  - Atklāšana
- APM
  - Trases
  - Metrikas
  - Žurnāli
- Pakalpojumi
  - Pakalpojumi
  - SLA
  - SLA pārskats
- Inventārs
  - Pārskats
  - Hosti
- Pārskati
  - Sistēmas informācija
  - Ieplānotie pārskati
  - Pieejamības pārskats
  - 100 populārākie trigeri

    - Audita žurnāls
    - Darbību žurnāls
    - Paziņojumi
- Datu apkopošana
    - Veidņu grupas
    - Hostu grupas
    - Veidnes
    - Hosti
    - Uzturēšana
    - Notikumu korelācija
    - Atklāšana
- Brīdinājumi
    - Darbības
        * Trigeru darbības
        * Pakalpojumu darbības
        * Atklāšanas darbības
        * Automātiskās reģistrācijas darbības
        * Iekšējās darbības
    - Multivides tipi
    - Skripti
- Lietotāji
    - Lietotāju grupas
    - Lietotāju lomas
    - Lietotāji
    - API marķieri
    - Autentifikācija
    - Ierīces
- Administrēšana
    - Vispārīgi
        * Lietotāja saskarne
        * Automātiskā reģistrācija
        * Taimauti
        * Attēli
        * Ikonu kartēšana
        * Regulārās izteiksmes
        * Trigeru attēlošanas opcijas
        * Ģeogrāfiskās kartes
        * Moduļi
        * Savienotāji
        * Citi
    - Audita žurnāls
    - Mājturība
    - Starpniekserveru grupas
    - Starpniekserveri
    - Makrosi
    - Rinda
        * Rindas pārskats
        * Rindas pārskats pēc starpniekservera
        * Rindas detalizēta informācija

Informācijas paneļi

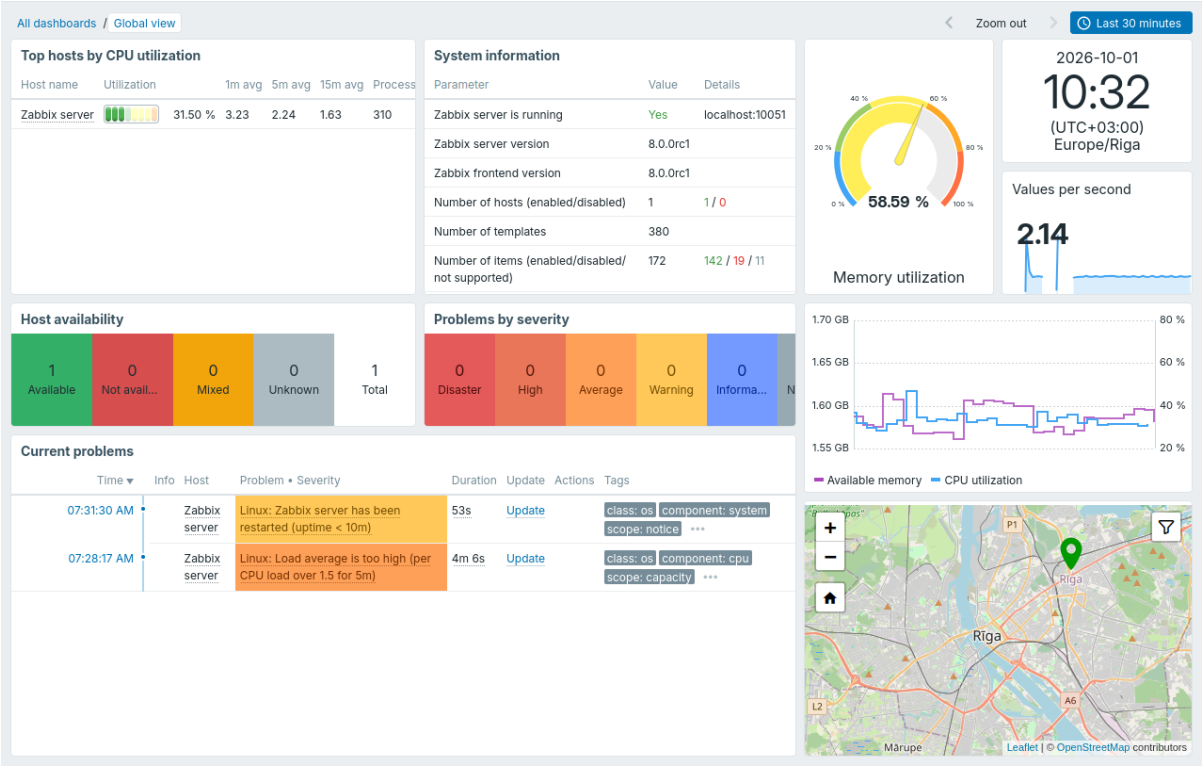
Informācijas paneļi sniedz pārskatu par jūsu uzraudzīto vidi. Varat tos izveidot un pielāgot atbilstoši savām vajadzībām.

To pamatelementi ir logrīki: grafiks, karte, jaunāko problēmu saraksts un citi, un katrs no tiem ir paredzēts noteikta veida informācijas attēlošanai.

Logrīkus var sakārtot vairākās informācijas paneļa lapās. Šādas lapas var skatīt kā slaidrādi. Varat arī pāriet uz jebkuru lapu, noklikšķinot uz tās cilnes.

Varat izveidot vairākus informācijas paneļus, taču vienlaikus tiek rādīts tikai viens.

Zabbix ietver vairākus noklusējuma informācijas paneļus. Viens no tiem ir informācijas panelis *Globālais skats*, kurā tiek rādīta jūsu uzraudzīto hostu, tostarp Zabbix servera, darbspēja un statuss:

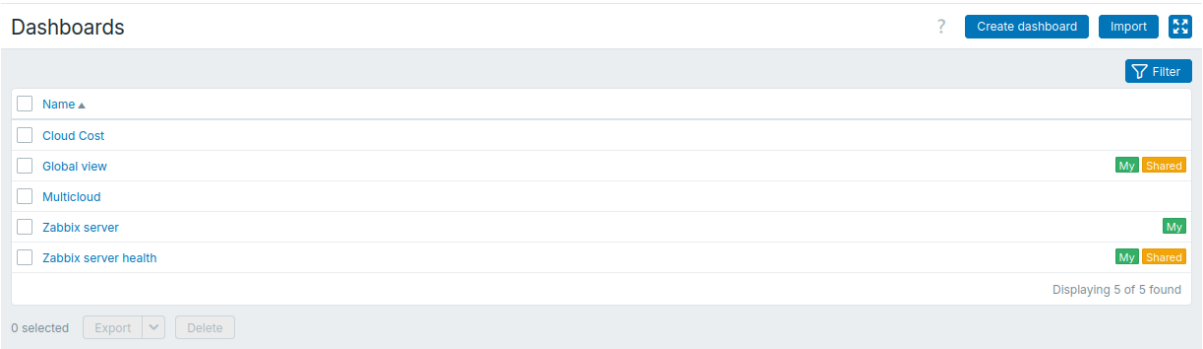


Vairāk informācijas paneļu varat atrast informācijas paneļu repozitorijā. Šie informācijas paneļi ir paredzēti darbam ar esošajām Zabbix veidnēm, un tos var lejupielādēt un importēt savā Zabbix vidē. Detalizēts katra informācijas paneļa apraksts, tostarp priekšnosacījumi, ir pieejams informācijas paneļa failā `README`.

Informācijas paneļu saraksts

Lai atvērtu informācijas paneļu sarakstu:

- Izvēlnē noklikšķiniet uz *Dashboards*.
- Skatot informācijas paneli, noklikšķiniet uz *All dashboards*, lai atvērtu informācijas paneļu sarakstu.



Piekļuve izvēlnei *Dashboards* ir atkarīga no lietotāja lomas un atļaujas *Dashboard* sadaļā *Access to UI elements*.

Šeit varat: - Izveidot informācijas paneļus. - Dzēst informācijas paneļus. Informācijas paneli varat dzēst, atlasot tā izvēles rūtiņu un noklikšķinot uz pogas *Delete*. - Importēt informācijas paneļus. - Eksportēt informācijas paneļus.

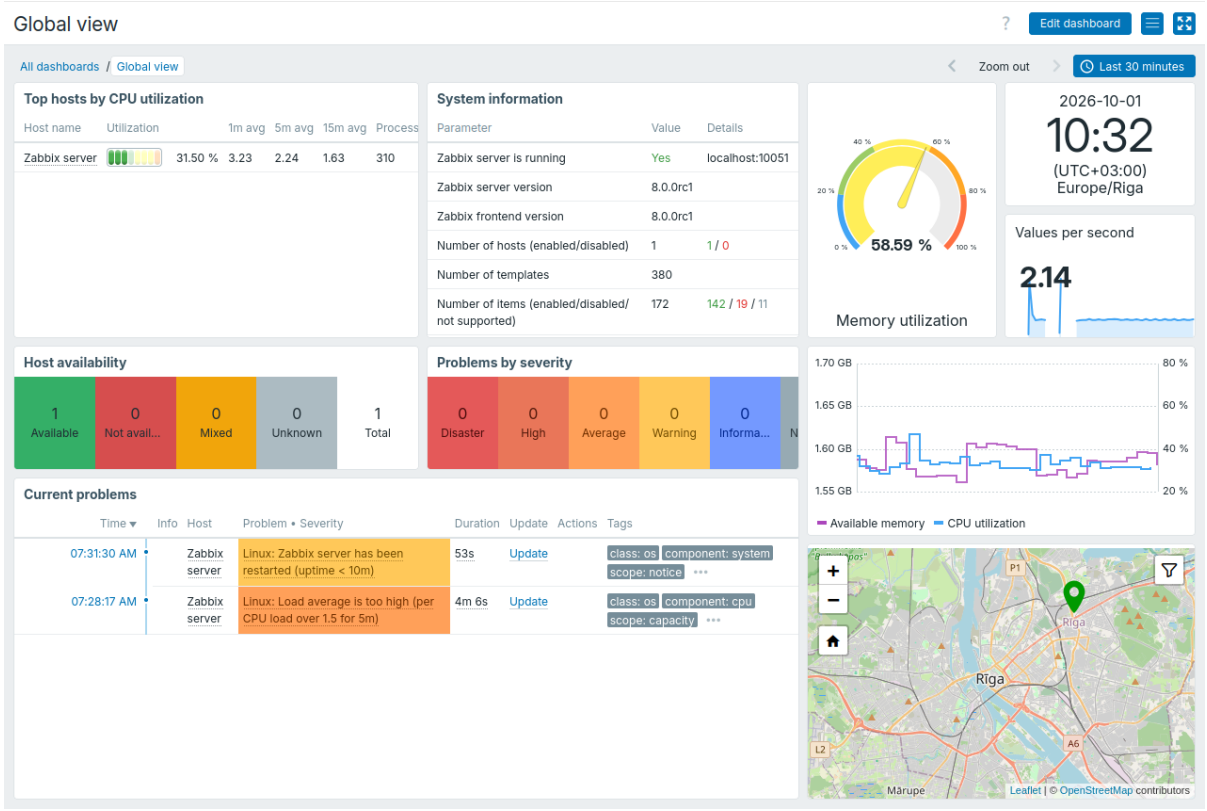
Informācijas paneļiem ir tagi, kas norāda to īpašumtiesības un kopīgošanas iestatījumus:

- *My* - informācijas panelis pieder jums un ir redzams tikai jums.
- *My + Shared* - informācijas panelis pieder jums, taču ir publisks vai kopīgots ar konkrētiem lietotājiem vai lietotāju grupām.
- Informācijas paneļi bez tagiem pieder citiem lietotājiem, taču ir publiski vai kopīgoti ar jums.

Sarakstu varat filtrēt pēc informācijas paneļa nosaukuma un īpašnieka.

Informācijas paneļa skatīšana

Noklikšķiniet uz informācijas paneļa nosaukuma informācijas paneļu sarakstā, lai skatītu informācijas paneli:



Zabbix lietotāja saskarne nākamajai sesijai saglabā pēdējo atvērto informācijas paneli.

Note:
Informācija, ko varat skatīt informācijas panelī, ir atkarīga no jūsu piekļuves hostiem.

Logrīkiem, kuros tiek rādītas diagrammas, laika perioda atlasītājs virs logrīkiem nosaka laika diapazonu. Tas parāda pašreizējo laika diapazonu kā etiķeti. Noklikšķiniet uz etiķetes, lai atvērtu vai aizvērtu atlasītāju.

Šeit varat pārvaldīt informācijas paneļus šādi:

- Rediģēt informācijas paneli.
- Informācijas paneļa izvēlnē *Actions* veikt šādas darbības:
  - *Sharing* - rediģēt informācijas paneļa kopīgošanas iestatījumus.
  - *Create new* - izveidot jaunu informācijas paneli.
  - *Clone* - izveidot jaunu informācijas paneli, klonējot pašreizējo. Pirms saglabāšanas varat rediģēt informācijas paneļa rekvizītus.
  - *Delete* - dzēst informācijas paneli.
  - *Export* - eksportēt informācijas paneli.
  - *Create new report* - izveidot ieplānotu pārskatu. Šī opcija ir atspējota, ja jums nav atļaujas pārvaldīt ieplānotus pārskatus.
  - *View related reports* - skatīt pašreizējam informācijas panelim saistīto esošo pārskatu sarakstu. Šī opcija ir atspējota, ja nav saistītu pārskatu vai jums nav atļaujas pārvaldīt ieplānotus pārskatus.
- Pārslēgties uz kioska režīmu.
- Sākt vai apturēt informācijas paneļa slaidrādi.

Informācijas paneļa skatīšanas režīmā logrīkiem ir pieejamas šādas opcijas:

- Rediģēt logrīku.
- Logrīka izvēlnē *Actions* ⋯ veikt šādas darbības:
  - *Copy* - kopēt logrīku.
  - *Download image* - lejupielādēt logrīku PNG attēla formātā. Šī opcija ir pieejama logrīkiem, kas atbalsta šo funkciju (skatiet atsevišķo logrīku lapas).
  - *Refresh interval* - atlasīt logrīka satura atsvaidzināšanas biežumu.

Informācijas paneļiem ar vairākām lapām informācijas paneļa skatīšanas režīmā lapas izvēlnē *Actions*  ir pieejamas šādas darbības:

- *Copy* - kopēt lapu.
- *Properties* - pielāgot lapas rekvizītus, tostarp nosaukumu un slaidrādes intervālu.

Slaidrāde

Ja paneļa rekvizītos ir atlasīta izvēles rūtiņa *Sākt slaidrādi*, slaidrāde tiek sākta automātiski, kad panelim ir vairāk nekā viena lapa. Pretējā gadījumā slaidrādi var sākt, noklikšķinot uz pogas *Sākt slaidrādi*.



Lapas mainās atbilstoši paneļa un atsevišķu lapu rekvizītos norādītajiem intervāliem. Ja atsevišķai lapai ir definēts rekvizīts *Slaidrādes intervāls*, tiek lietots šai lapai norādītais intervāls. Pretējā gadījumā tiek lietots paneļa *Noklusējuma slaidrādes intervāls*.

Lai apturētu slaidrādi, noklikšķiniet uz *Apturēt slaidrādi*.

Kioska režīmā izmantojiet šādas pogas:

-  , lai sāktu slaidrādi;
-  , lai apturētu slaidrādi;
-  , lai pārietu uz iepriekšējo lapu;
-  , lai pārietu uz nākamo lapu.

Informācijas paneļa kopīgošana

Informācijas paneļi var būt publiski vai privāti.

Publiskie informācijas paneļi ir redzami visiem lietotājiem.
Privātie informācijas paneļi ir redzami tikai to īpašniekiem un lietotājiem vai lietotāju grupām, ar kurām tie ir kopīgoti.

Lai mainītu informācijas paneļa kopīgošanas iestatījumus, informācijas paneļa izvēlnē *Actions* noklikšķiniet uz *Sharing*, kamēr skatāt informācijas paneli:

Parameter	Description
Type	Atlasiet informācijas paneļa tipu: **Private** - informācijas panelis ir redzams tikai atlasītajām lietotāju grupām un lietotājiem. **Public** - informācijas panelis ir redzams visiem lietotājiem.
List of user group shares	Atlasiet lietotāju grupas, kurām ir piekļuve informācijas panelim. Varat piešķirt tikai lasīšanas vai lasīšanas un rakstīšanas piekļuvi.
List of user shares	Atlasiet lietotājus, kuriem ir piekļuve informācijas panelim. Varat piešķirt tikai lasīšanas vai lasīšanas un rakstīšanas piekļuvi.

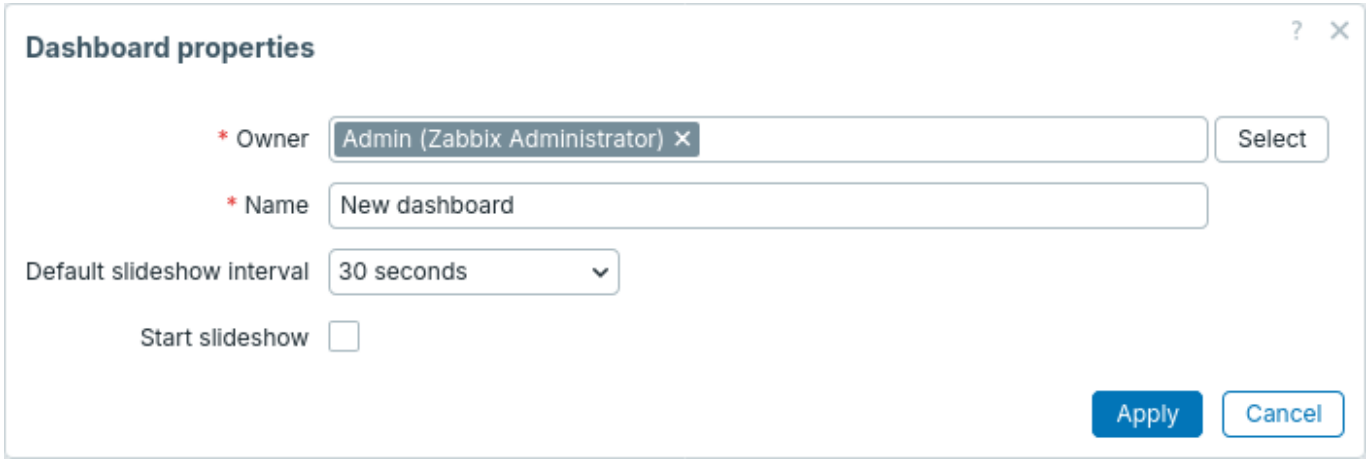
Skatiet sadaļu Permissions to dashboards, lai uzzinātu, kā kopīgošanas opcijas ierobežo lietotājiem pieejamās darbības.

Izveidot informācijas paneli

Informācijas paneli var izveidot divos veidos:

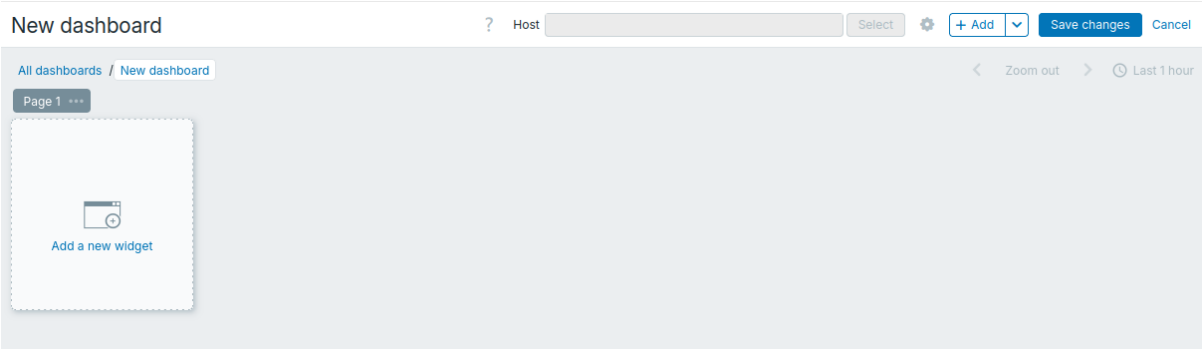
- Informācijas paneļu sarakstā noklikšķiniet uz pogas *Izveidot informācijas paneli*.
- Informācijas paneļa skatīšanas režīmā informācijas paneļa izvēlnē *Darbības* noklikšķiniet uz *Izveidot jaunu*.

Pēc tam norādiet informācijas paneļa rekvizītus:



Parametrs	Apraksts
Īpašnieks	Atlasiet lietotāju, kuram piederēs informācijas panelis.
Nosaukums	Ievadiet informācijas paneļa nosaukumu.
Noklusējuma slaidrādes intervāls	Atlasiet, cik ilgi tiek rādīta katra informācijas paneļa lapa, pirms slaidrāde pāriet uz nākamo lapu.
Sākt slaidrādi	Atzīmējiet šo izvēles rūtiņu, lai slaidrāde tiktu automātiski sākta, tiklīdz informācijas panelim ir vairāk nekā viena lapa.

Pēc tam noklikšķiniet uz *Lietot*. Jaunais informācijas panelis tiek atvērts rediģēšanas režīmā:



Šeit varat pievienot informācijas paneļa logrīkus un lapas.

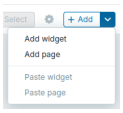
Ņemiet vērā, ka informācijas paneļa minimālais platums ir 1200 pikseļi; šaurākos logos tiek parādīta horizontāla ritjosla. Platāks informācijas panelis vienmēr aizpilda pārlūkprogrammas logu; tā logrīki nevar pārsniegt loga malas.

Noklikšķiniet uz pogas *Saglabāt izmaiņas*, lai saglabātu informācijas paneli, vai uz *Atcelt*, lai atgrieztos iepriekšējā ekrānā.

Logrīku pievienošana

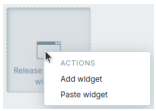
Informācijas paneļa rediģēšanas režīmā logrīkus var pievienot divos veidos:

- Noklikšķiniet uz pogas *Add*. Varat arī noklikšķināt uz nolaižamās bultiņas pie pogas *Add* un atlasīt *Add widget*.



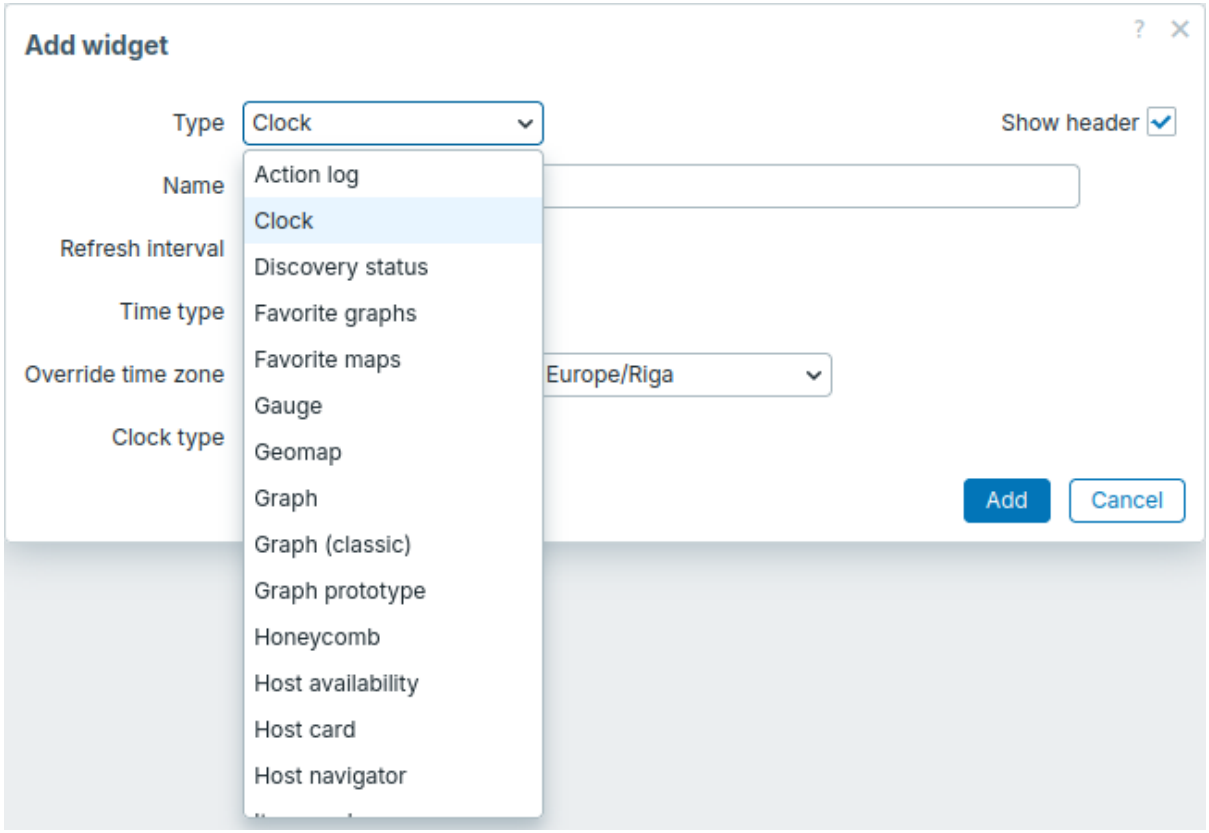
  Logrīka vietturis tiek pievienots noklusējuma izmērā pirmajā pieejamajā brīvajā vietā informācijas panelī, un tiek atvērta logrīka konfigurācijas veidlapa.

- Novietojiet peles rādītāju virs tukša informācijas paneļa apgabala. Pēc tam noklikšķiniet uz logrīka viettura, lai atvērtu logrīka konfigurācijas veidlapu. Varat arī noklikšķināt uz viettura un vilkt to, lai mainītu tā izmēru, pēc tam atlaist peles pogu, lai atvērtu logrīka konfigurācijas veidlapu; ja logrīks ir nokopēts starpliktuvē, vispirms tiks piedāvāts izvēlēties *Add widget* vai *Paste widget*.



Pēc tam veiciet tālāk norādītās darbības:

1. Logrīka konfigurācijas veidlapā atlasiet logrīka tipu.
2. Konfigurējiet logrīka parametrus.
3. Noklikšķiniet uz *Add*, lai pievienotu logrīku informācijas panelim.



Logrīks tiek pievienots noklusējuma izmērā vai mainīts, lai ietilptu pieejamajā vietā. Logrīku izmēru un novietojumu var mainīt, kā aprakstīts sadaļā *Logrīku rediģēšana*.

Varat arī kopēt un ielīmēt logrīkus, lai izveidotu jaunus logrīkus ar tādu pašu konfigurāciju — gan tajā pašā informācijas panelī, gan starp informācijas paneļiem, kas atvērti dažādās pārlūkprogrammas cilnēs.

Logrīku kopēšana un ielīmēšana

Lai kopētu un ielīmētu logrīku:

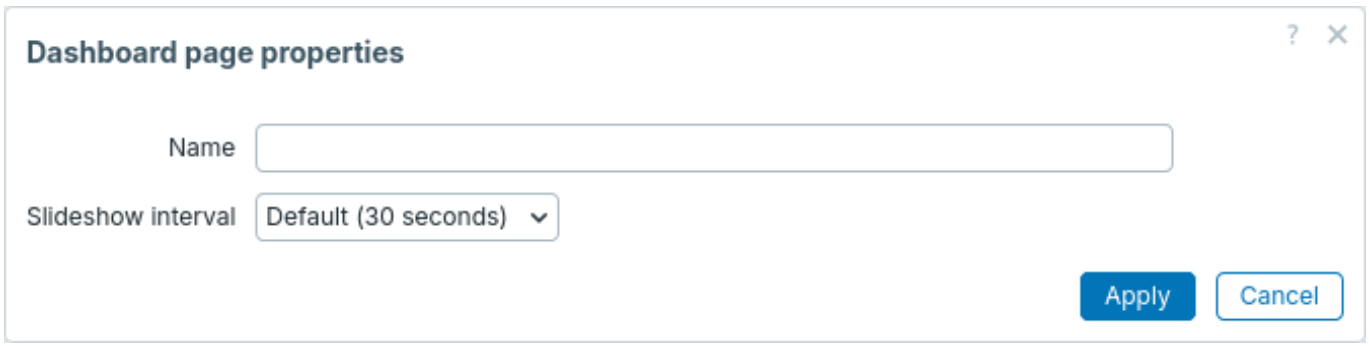
1. Logrīka izvēlnē *Darbības* ⋯ atlasiet *Kopēt*.
2. Informācijas paneļa rediģēšanas režīmā noklikšķiniet uz nolaižamās bultiņas pie pogas *Pievienot* un atlasiet *Ielīmēt logrīku*. Varat arī noklikšķināt uz tukša apgabala informācijas panelī un atlasīt *Ielīmēt logrīku*.

Varat arī ielīmēt kopētu logrīku virs esoša logrīka, tā izvēlnē *Darbības* atlasot opciju *Ielīmēt*.

Lapu pievienošana

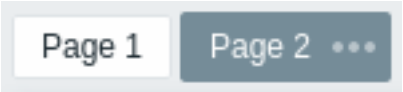
Lai informācijas panelim pievienotu jaunu lapu:

1. Informācijas paneļa rediģēšanas režīmā noklikšķiniet uz nolaižamās bultiņas pie pogas *Add* un atlasiet *Add page*.
2. Aizpildiet lapas rekvizītus:
   - *Name* - ievadiet lapas nosaukumu. Ja nosaukumu atstāsiet tukšu, lapa tiks pievienota ar nosaukumu *Page N*, kur "N" ir kārtas lapas numurs.
   - *Slideshow interval* - atlasiet, cik ilgi lapa tiek rādīta, pirms slideshow pāriet uz nākamo lapu.



3. Noklikšķiniet uz *Apply*.

Jaunajai lapai tiek pievienota jauna cilne (piemēram, *Page 2*).



Varat arī kopēt un ielīmēt lapas, lai izveidotu jaunas lapas ar tādiem pašiem rekvizītiem, gan vienā informācijas panelī, gan starp informācijas paneļiem, kas atvērti dažādās cilnēs.

Lapu kopēšana/ielīmēšana

Lai kopētu un ielīmētu lapu:

1. Lapas izvēlnē *Darbības* atlasiet *Kopēt*.
2. Informācijas paneļa rediģēšanas režīmā noklikšķiniet uz nolaižamās izvēlnes bultiņas pie pogas *Pievienot* un atlasiet *Ielīmēt lapu*.

Informācijas paneļa rediģēšana

Lai rediģētu informācijas paneli, augšējā labajā stūrī noklikšķiniet uz pogas *Edit dashboard*. Varat arī noklikšķināt uz logrīka pogas *Edit*. Informācijas panelis tiek atvērts rediģēšanas režīmā.

Šeit informācijas paneļus var pārvaldīt šādi:

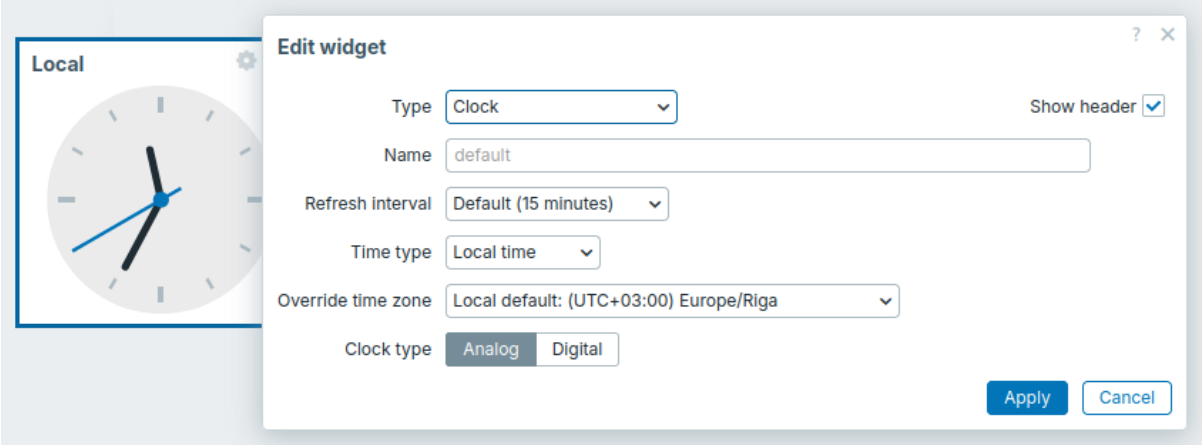
- Rediģējiet informācijas paneļa rekvizītus, noklikšķinot uz zobrata pogas.
- Rediģējiet informācijas paneļa logrīkus.
- Piekļūstiet tālāk norādītajām informācijas paneļa darbībām, noklikšķinot uz nolaižamās bultiņas pie pogas *Add*:
  - *Add widget* - pievienojiet logrīku informācijas panelim.
  - *Add page* - pievienojiet lapu informācijas panelim.
  - *Paste widget* - ielīmējiet nokopētu logrīku. Šī opcija ir neaktīva, ja neviens logrīks nav nokopēts. Vienlaikus var nokopēt tikai vienu logrīku.
  - *Paste page* - ielīmējiet nokopētu lapu. Šī opcija ir neaktīva, ja neviena lapa nav nokopēta. Vienlaikus var nokopēt tikai vienu lapu.

Pēc informācijas paneļa rediģēšanas noklikšķiniet uz pogas *Save changes*, lai saglabātu informācijas paneli, vai uz *Cancel*, lai atgrieztos iepriekšējā ekrānā.

Logrīku rediģēšana

Logrīku var rediģēt informācijas paneļa rediģēšanas režīmā, noklikšķinot uz tā pogas *Rediģēt*. Varat arī skatot informācijas paneli, novietot kursoru virs logrīka un noklikšķināt uz pogas *Rediģēt*.

Logrīka rediģēšanas laikā tas tiek izcelts, un izmaiņas tiek priekšskatītas reāllaikā.



Logrīku un tā konfigurācijas veidlapu var pārvietot, velkot to virsraksta joslas, bet logrīka izmērus var mainīt, velkot tā izmēru maiņas turi.



Rediģējot logrīku, tā izvēlnē *Darbības* ir pieejamas šādas darbības:

- *Kopēt* - kopēt logrīku.
- *Ielīmēt* - ielīmēt nokopētu logrīku virs šī logrīka. Šī opcija ir pelēkota, ja neviens logrīks nav nokopēts.
- *Dzēst* - dzēst logrīku.

Ja obligātie logrīka parametri nav konfigurēti, logrīks pāriet stāvoklī *Logrīks nav pilnībā konfigurēts*.

Pēc rediģēšanas logrīka konfigurācijas veidlapā noklikšķiniet uz *Lietot*, pēc tam informācijas paneļa augšējā labajā stūrī noklikšķiniet uz *Saglabāt izmaiņas*, lai lietotu izmaiņas.

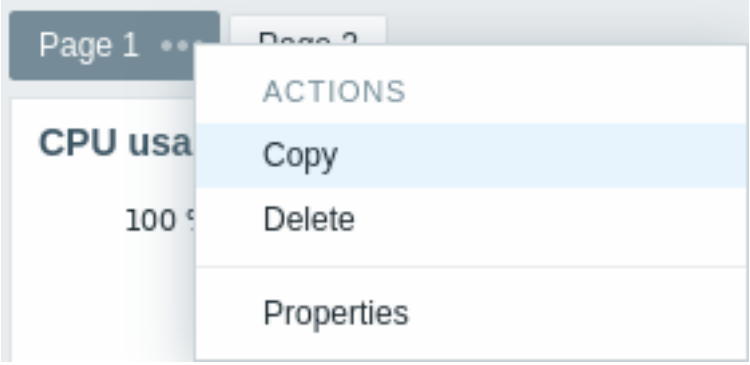
Lapu rediģēšana

Lapas var rediģēt informācijas paneļa rediģēšanas režīmā.

Lapu secību var mainīt, velkot un nometot lapu cilnes. Secības maiņa nepārdēvē lapas, izņemot lapas ar automātiski piešķirtiem nosaukumiem (*Lapa N*), kurām tiek mainīti numuri.

Rediģējot lapu, lapas izvēlnē *Darbības* ir pieejamas šādas darbības:

- *Kopēt* - kopēt lapu.
- *Dzēst* - dzēst lapu.
- *Rekvizīti* - pielāgot lapas rekvizītus.



Piekļuves tiesības informācijas paneļiem

Parasto lietotāju un *Admin* tipa lietotāju piekļuves tiesības informācijas paneļiem ir ierobežotas, kā aprakstīts tālāk, ja vien viņiem piešķirtajā lietotāja lomā ir iespējota piekļuve informācijas paneļiem un opcija *Izveidot un rediģēt informācijas paneļus*:

- Viņi var skatīt un klonēt publisku informācijas paneli arī tad, ja informācijas paneļa kopīgošanas iestatījumiem nav pievienots neviens lietotājs vai lietotāju grupa.
- Viņi var skatīt un klonēt privātu informācijas paneli, ja viņiem ir vismaz *Lasīšanas* tiesības uz to, kā norādīts kopīgošanas iestatījumos.
- Viņi var rediģēt un dzēst informācijas paneli tikai tad, ja viņiem ir *Lasīšanas un rakstīšanas* tiesības uz to, kā norādīts kopīgošanas iestatījumos.
- Viņi nevar mainīt informācijas paneļa īpašnieku.

Informācijas paneļa sīkrīki

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.

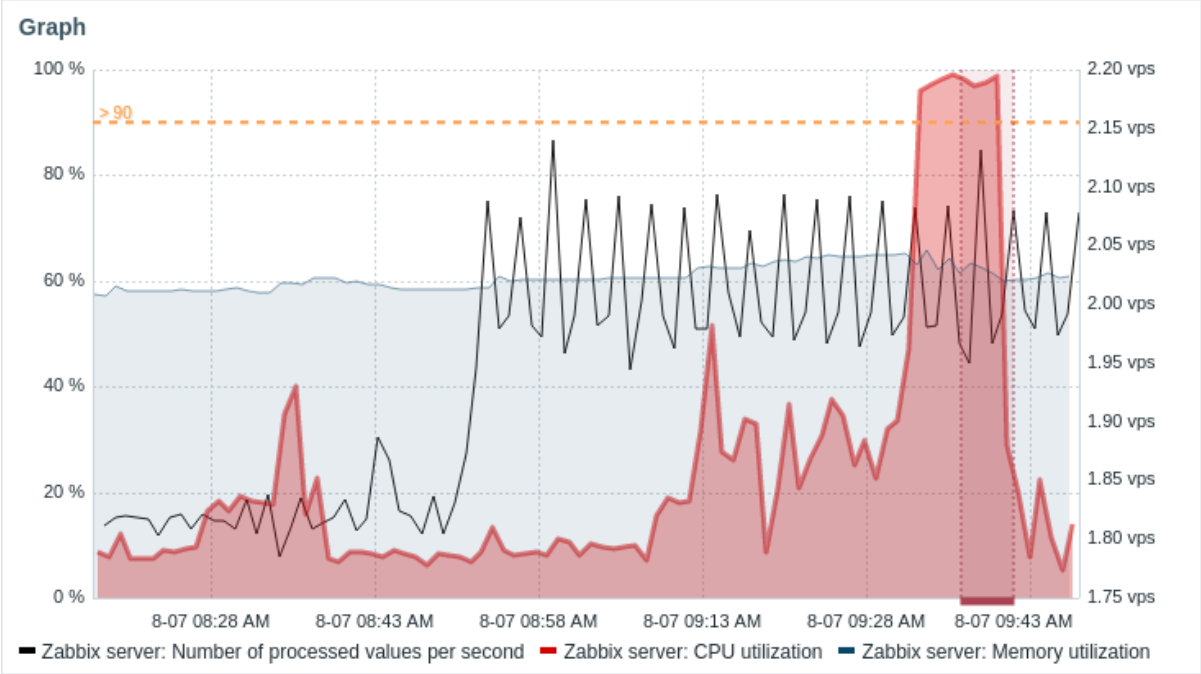
Lai iegūtu detalizētu informāciju par logrīku konfigurāciju, skatiet Paneļa logrīku parametrus.

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.

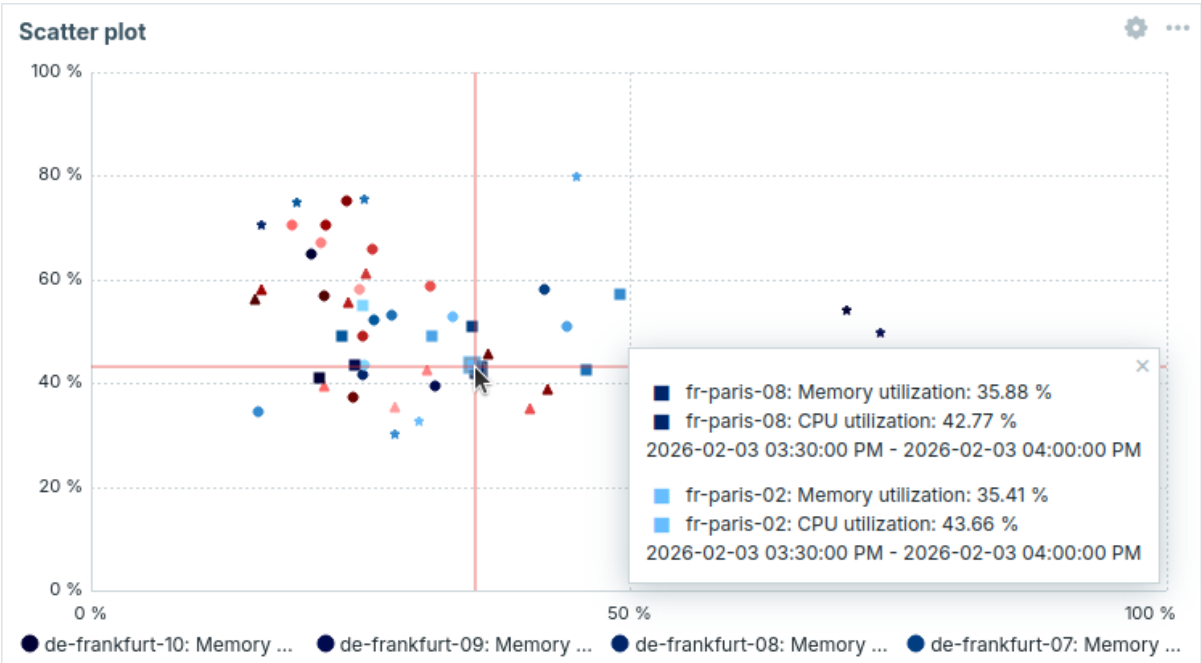


Skatiet arī:

- 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.

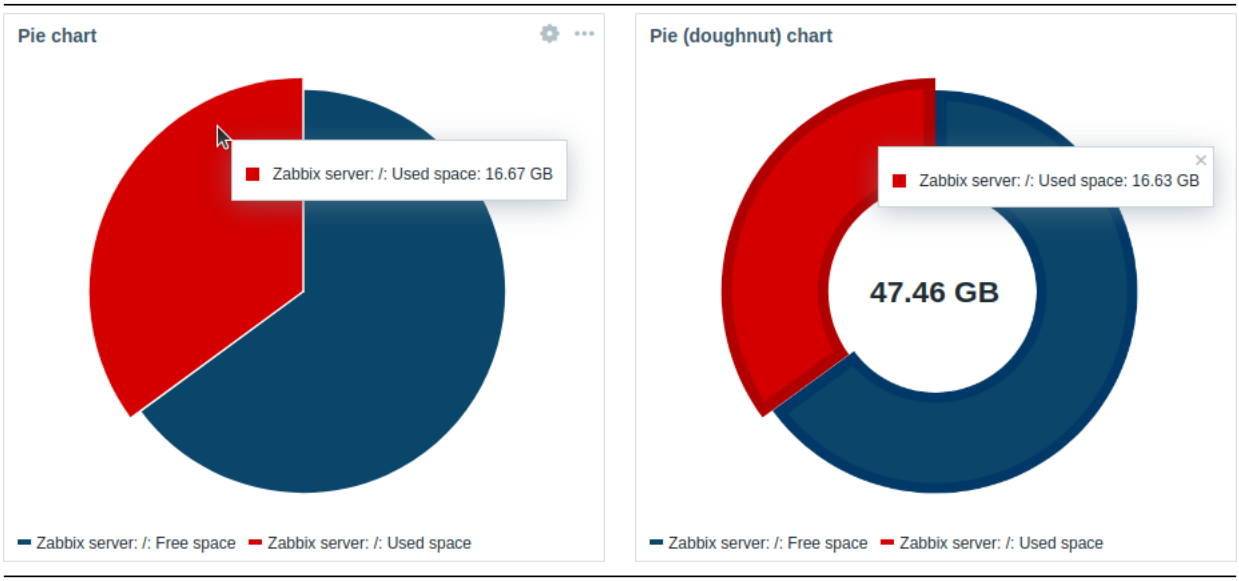
Izkliedes diagramma

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

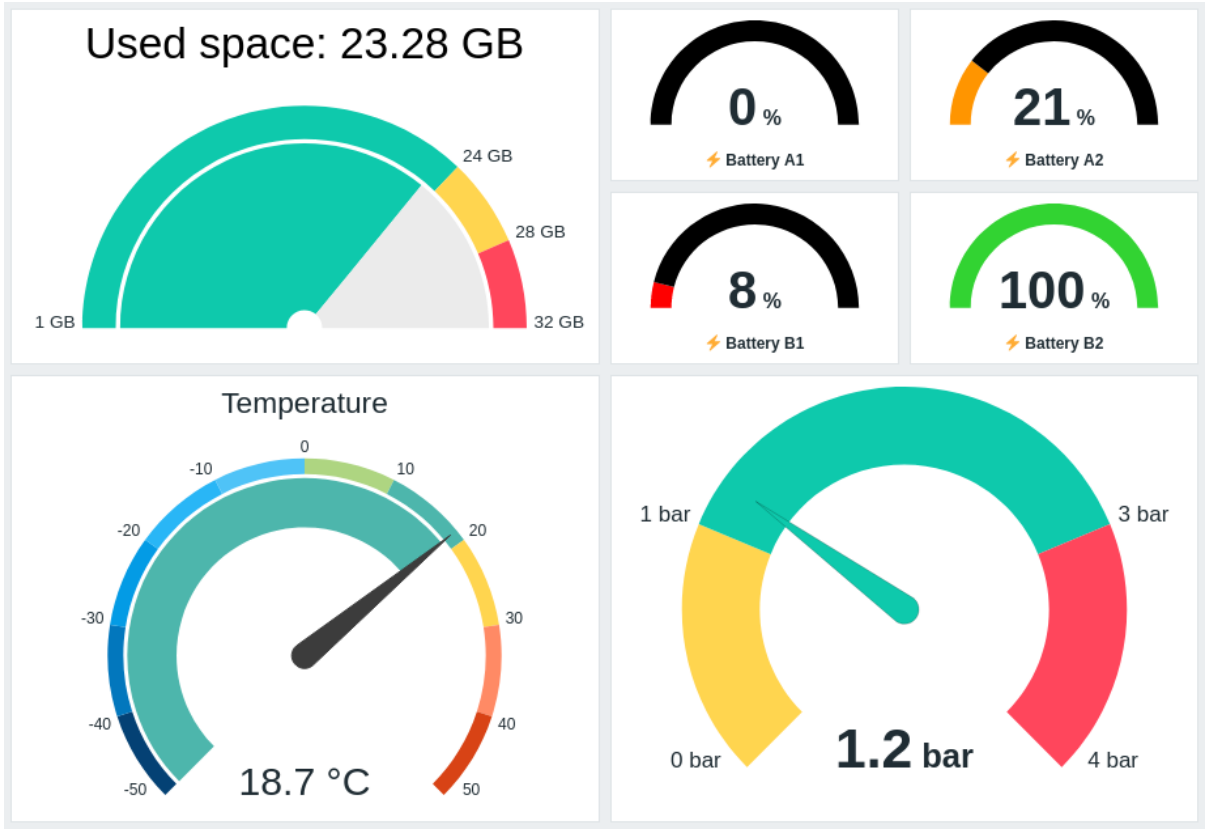
Sektoru diagrammas logrīks attēlo skaitliskus vienumu datus kā vektorgrafikas sektoru vai gredzenveida diagrammu. Šis logrīks var palīdzēt vizualizēt, kā vienumi vai hosti veido kopējo datu kopu.



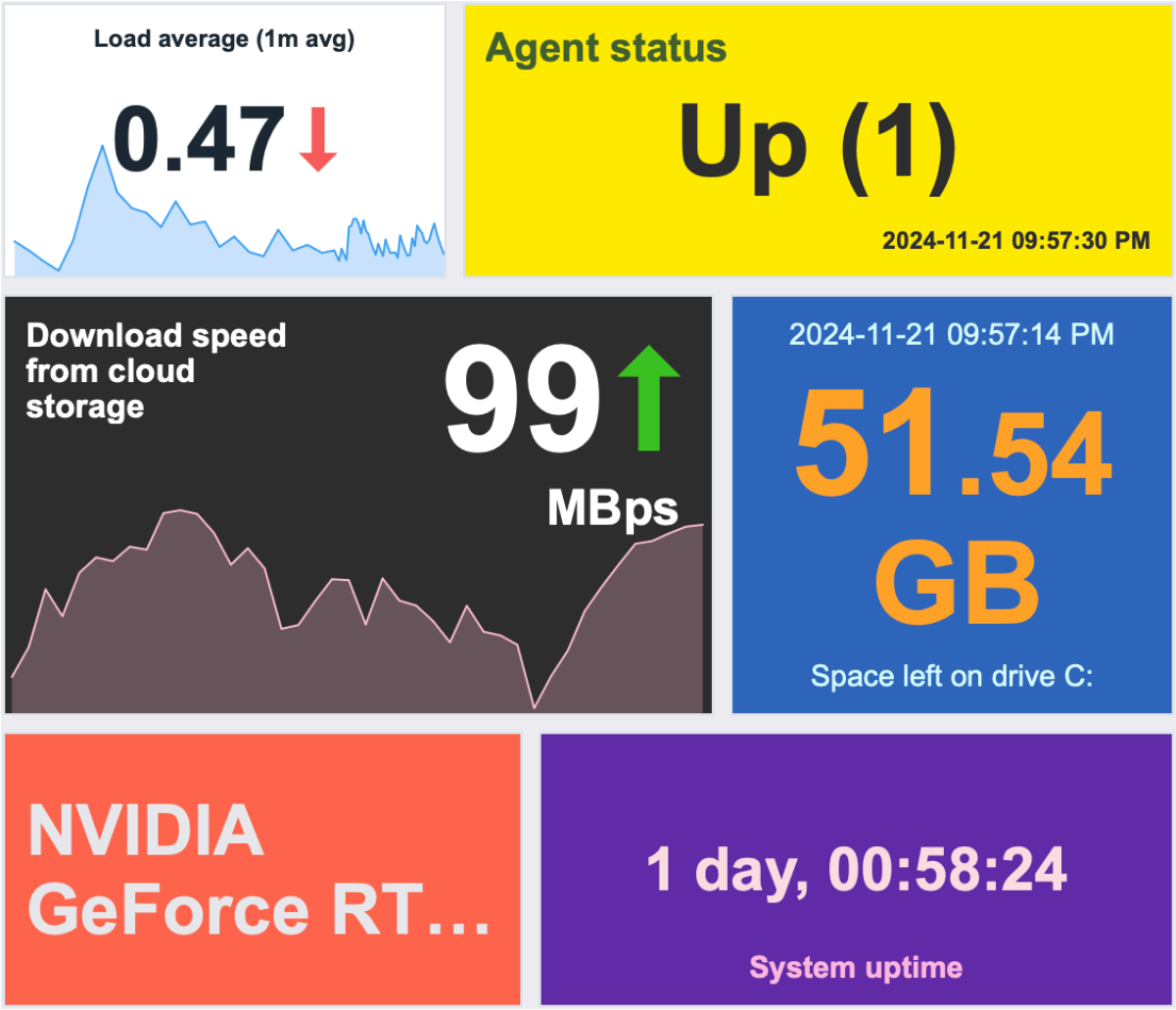
Mērinstrumenti/vienuma vērtība

Mērinstruments/Vienuma vērtība logrīki attēlo viena vienuma vērtību, kas ir noderīgi, lai sekotu svarīgiem rādītājiem, vizualizētu sliekšņus un noteiktu pēkšņas datu izmaiņas.

Mērinstrumenta logrīks:

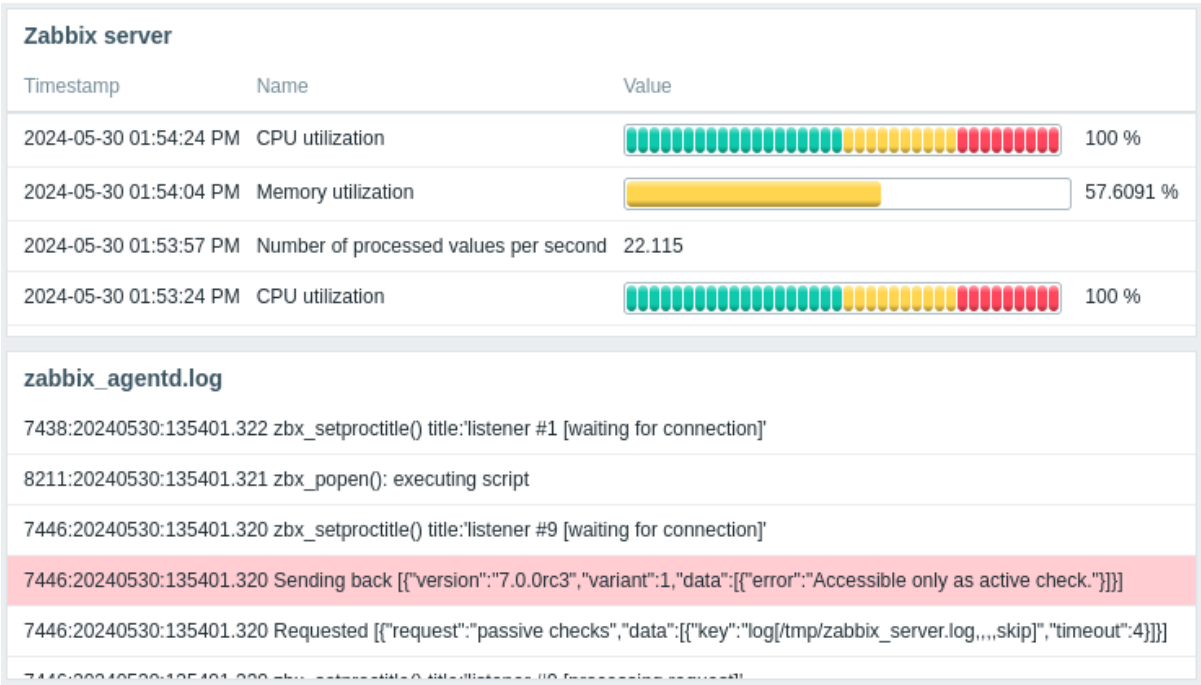


Vienuma vērtības logrīks:



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).



Zabbix server

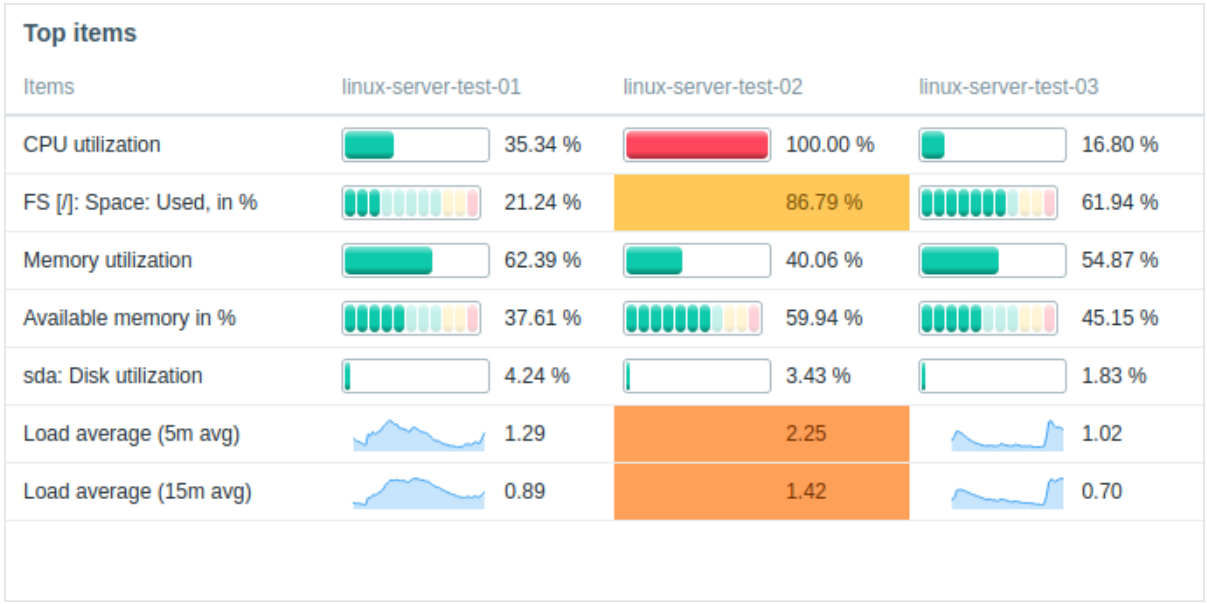
Timestamp	Name	Value
2024-05-30 01:54:24 PM	CPU utilization	100 %
2024-05-30 01:54:04 PM	Memory utilization	57.6091 %
2024-05-30 01:53:57 PM	Number of processed values per second	22.115
2024-05-30 01:53:24 PM	CPU utilization	100 %

zabbix_agentd.log

```
7438:20240530:135401.322 zbx_setproctitle() title:'listener #1 [waiting for connection]'
8211:20240530:135401.321 zbx_popen(): executing script
7446:20240530:135401.320 zbx_setproctitle() title:'listener #9 [waiting for connection]'
7446:20240530:135401.320 Sending back [{"version":"7.0.0rc3","variant":1,"data":[{"error":"Accessible only as active check."}]}]
7446:20240530:135401.320 Requested [{"request":"passive checks","data":[{"key":"log[/tmp/zabbix_server.log,,,,skip]","timeout":4}]}]
```

Populārākie vienumi

Logrīks Top items attēlo atlasīto vienumu augstākās/zemākās vērtības, nodrošinot ātru pārskatu par to veiktspēju.

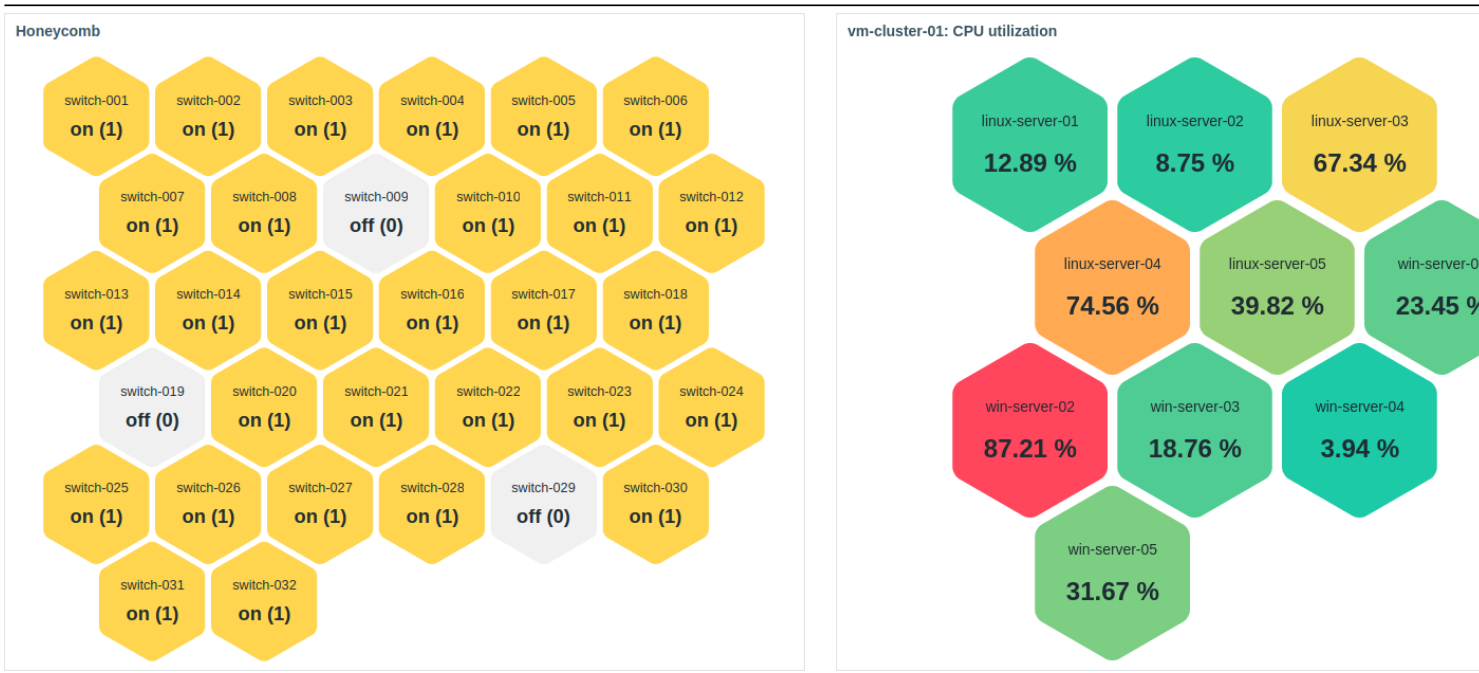


Skatiet arī:

- 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.

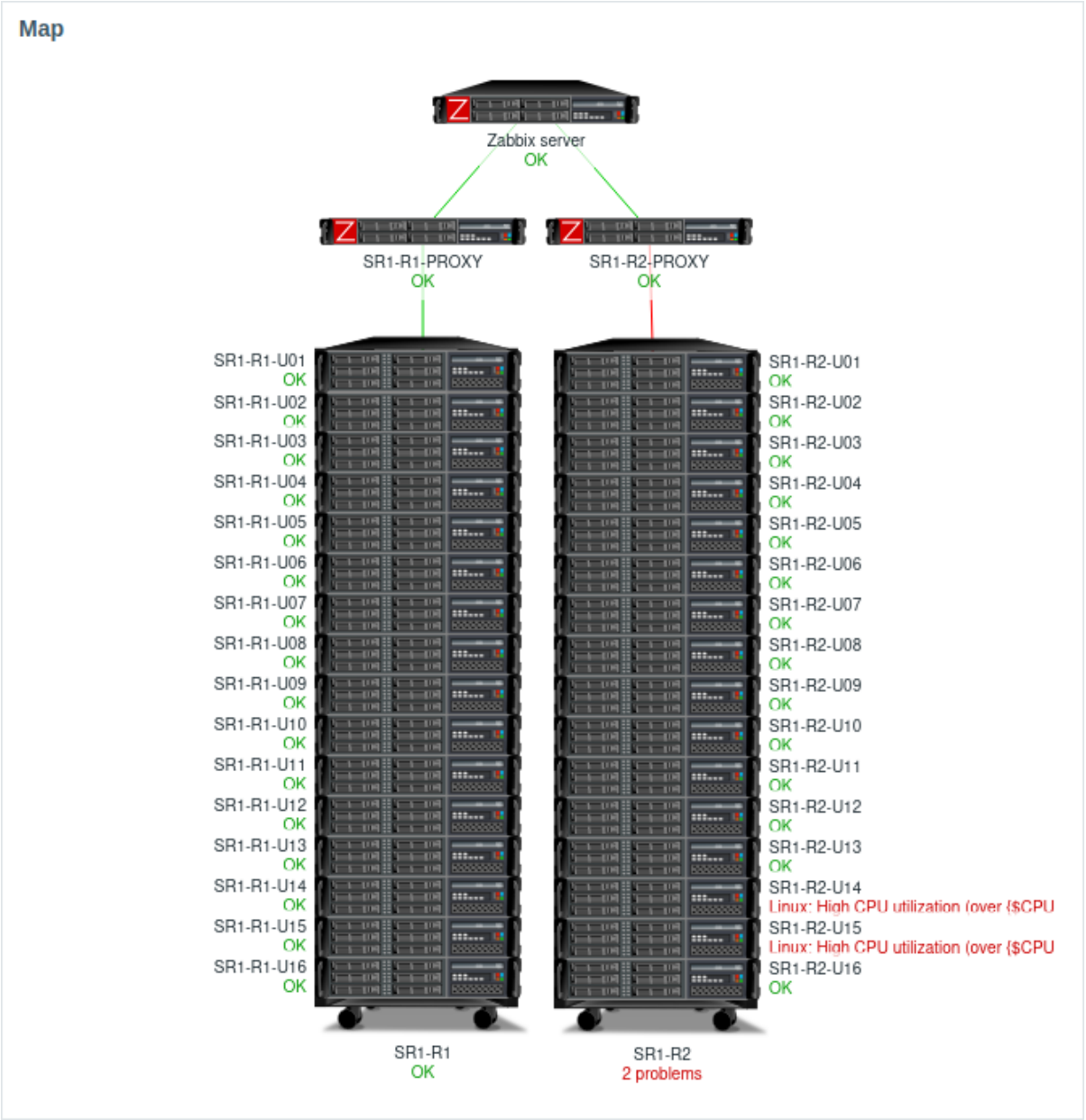


Kartes

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.

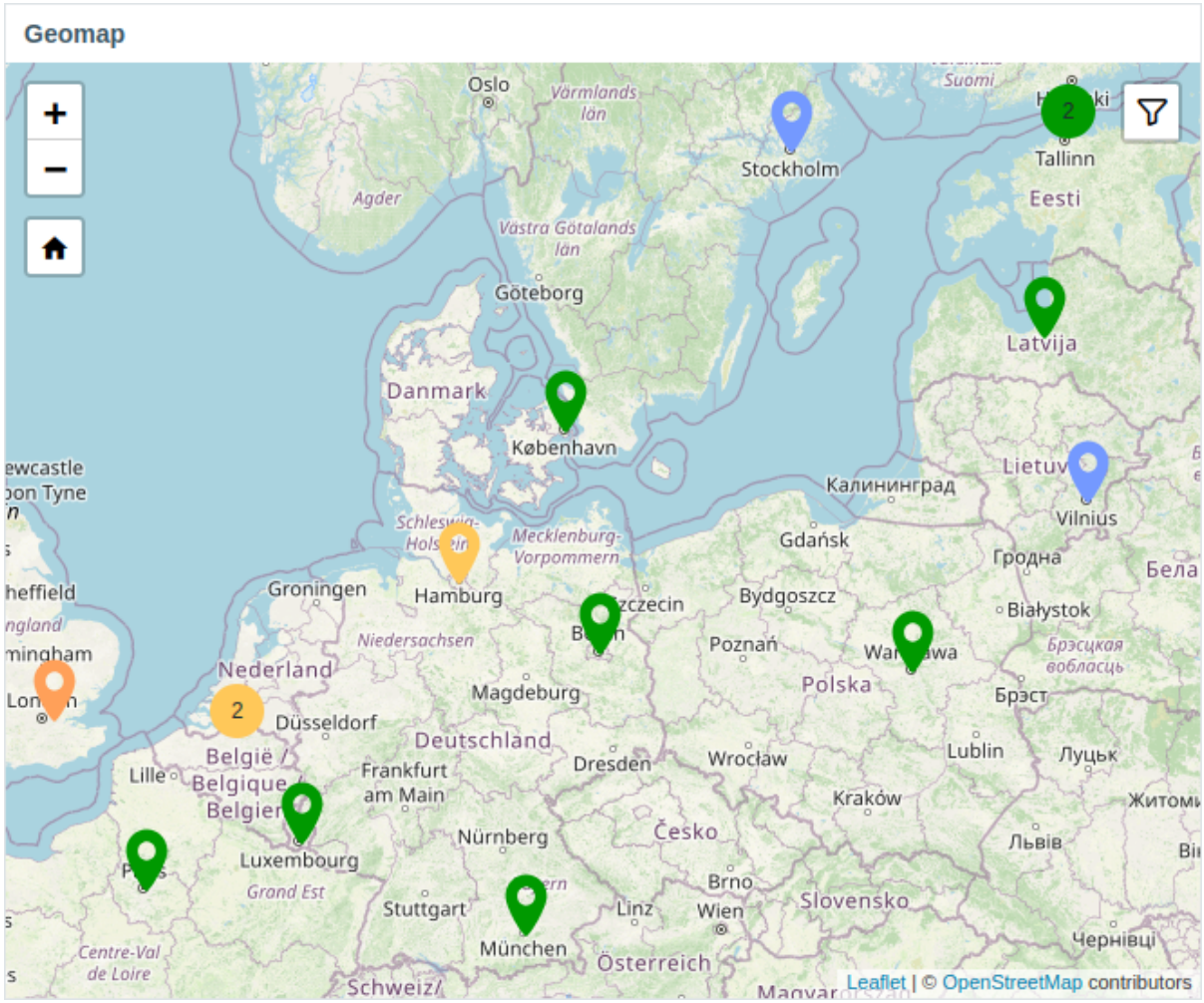


Skatiet arī:

- 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.

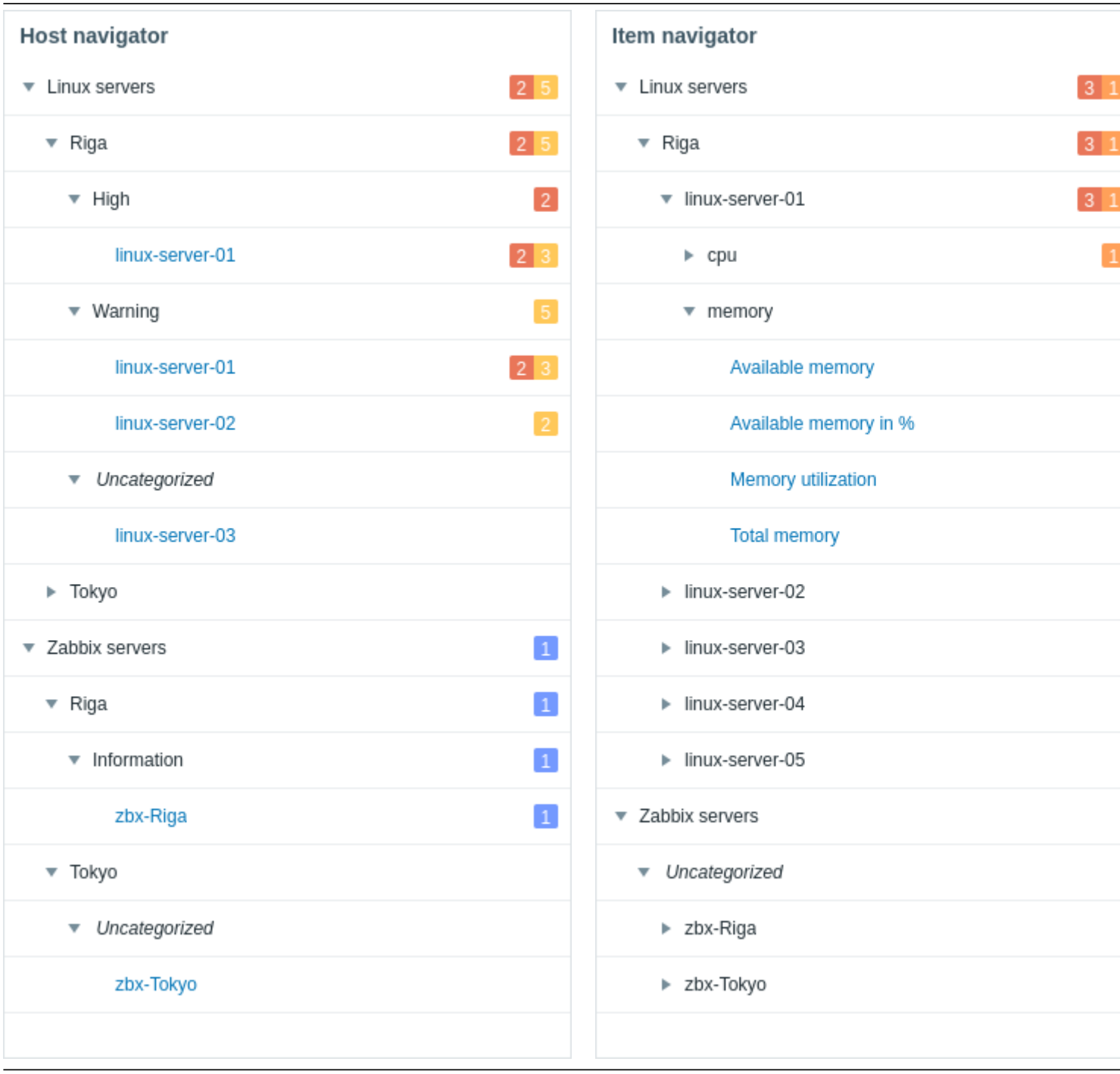


Hosta/vienuma informācija

Šie logrīki nodrošina hostu vai vienumu sarakstus un detalizētus skatus.

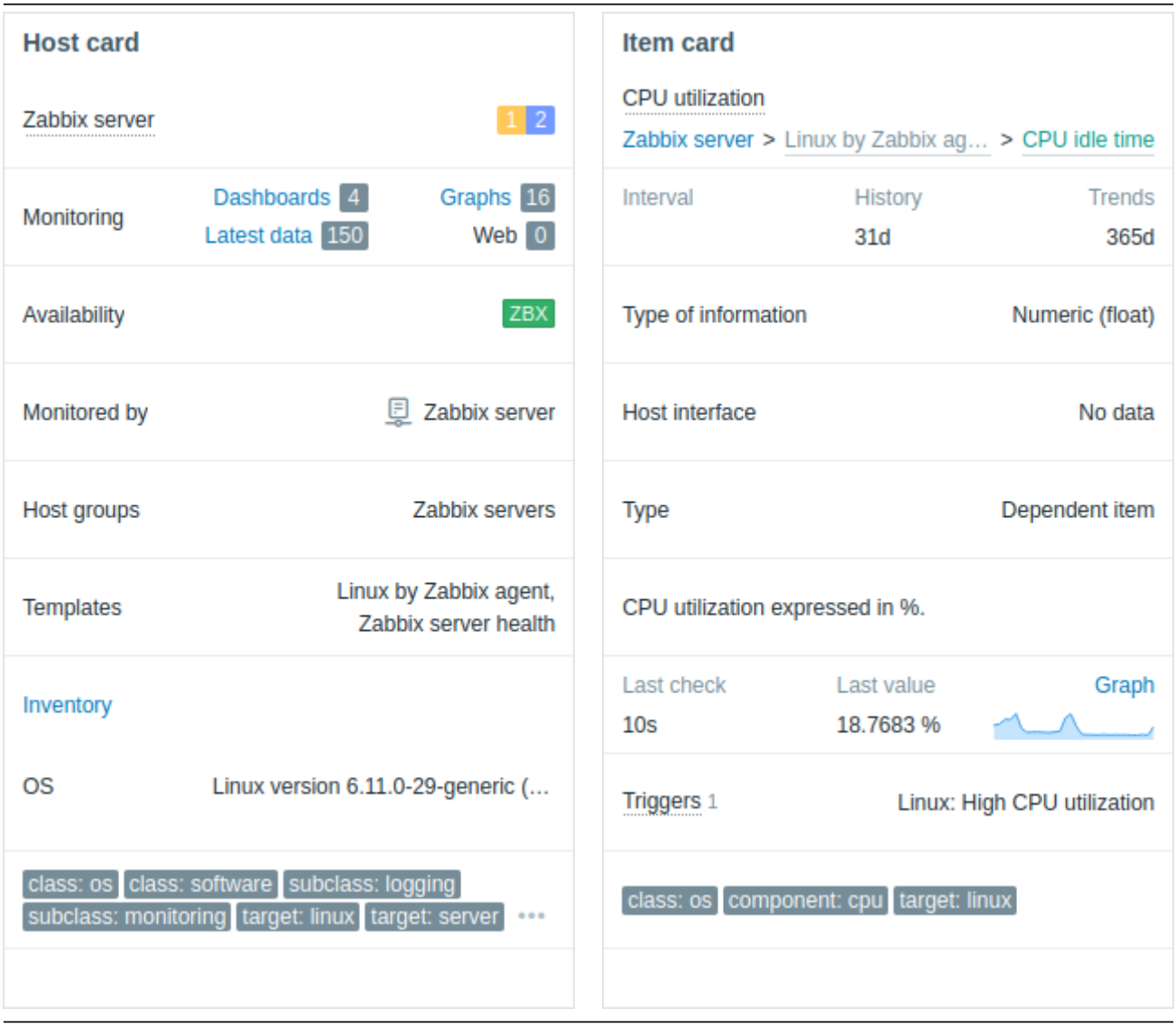
Hostu/vienumu navigators

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

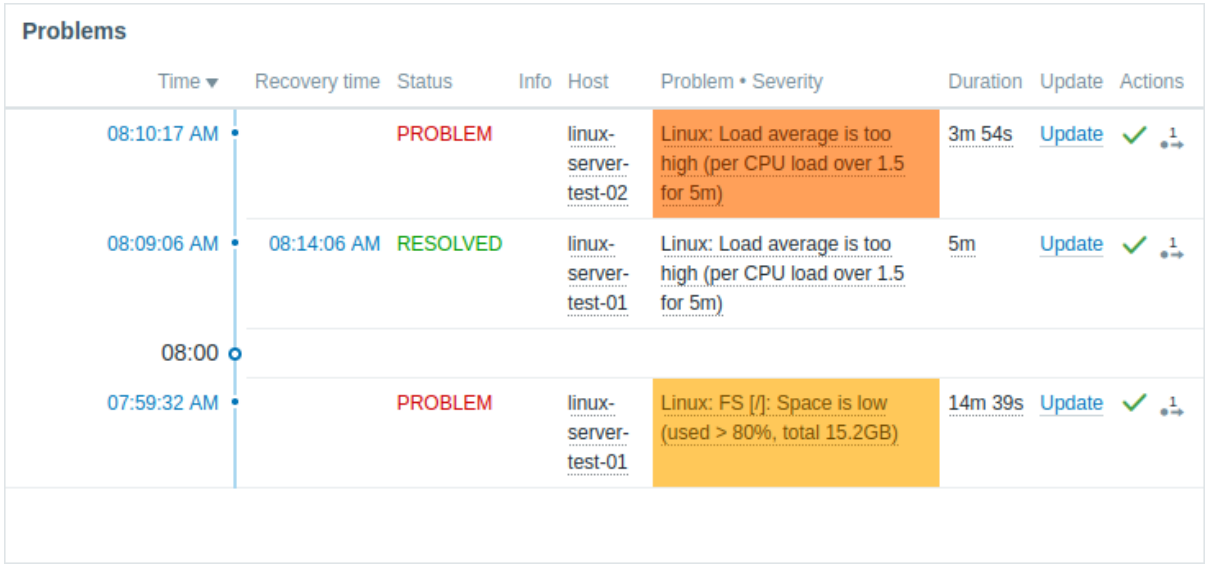
Logrīks Host availability attēlo hostu pieejamību atlasītajās hostu grupās, ļaujot jums uzraudzīt ieslēgto vai izslēgto hostu skaitu.

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

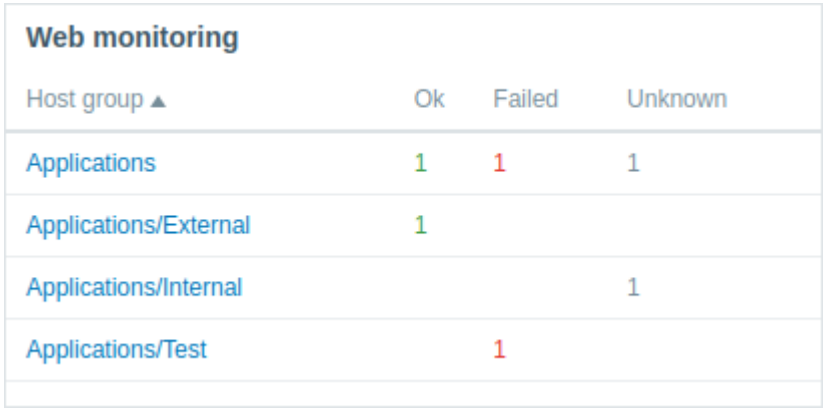
SLA atskaite logrīks attēlo SLA atskaites, palīdzot jums uzraudzīt pakalpojumu veiktspēju attiecībā pret noteiktajiem mērķiem.

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.

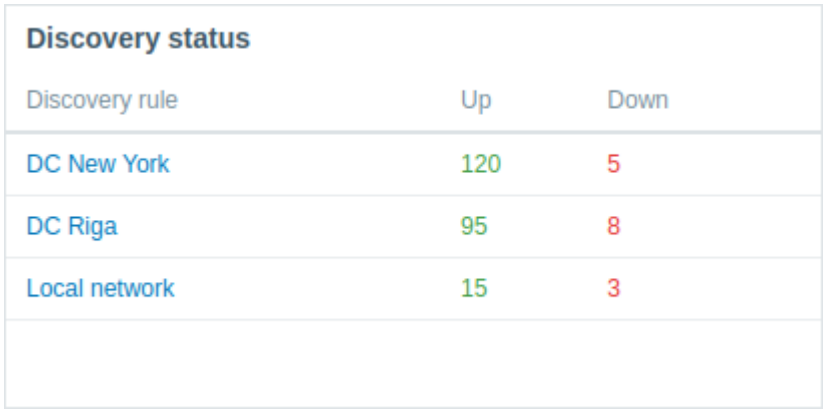


Web monitoring

Host group ▲	Ok	Failed	Unknown
Applications	1	1	1
Applications/External	1		
Applications/Internal			1
Applications/Test		1	

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.



Discovery status

Discovery rule	Up	Down
DC New York	120	5
DC Riga	95	8
Local network	15	3

Paneļa utilītas un sistēmas informācija

Šie logrīki nodrošina papildu informāciju un rīkus, lai uzlabotu jūsu paneļus.

Sistēmas informācija

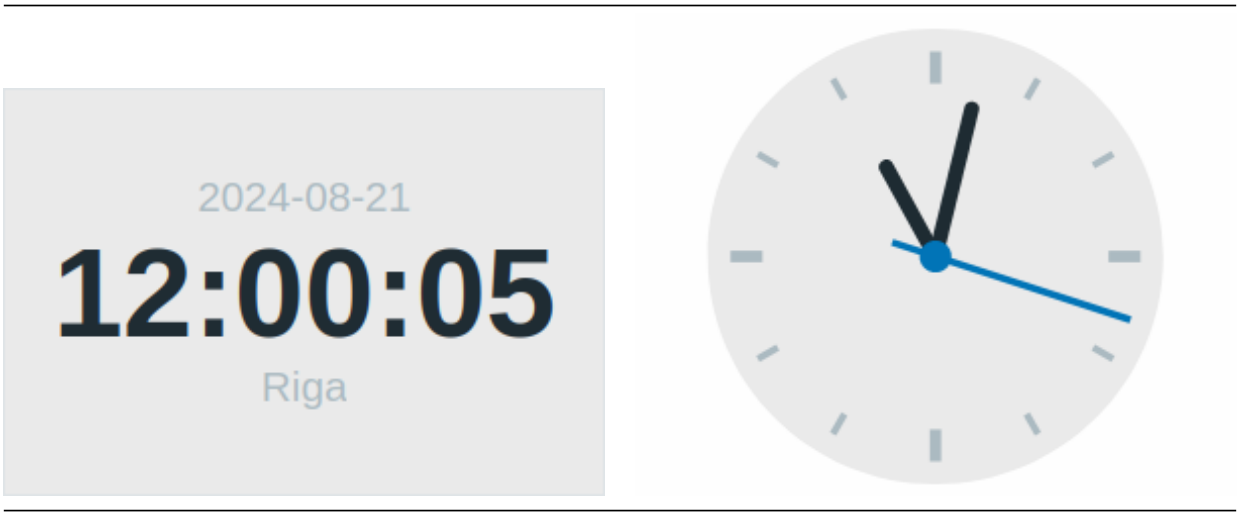
Logrīks Sistēmas informācija parāda galveno Zabbix serveris un sistēmas datu kopsavilkumu vai detalizētu informāciju par augstas pieejamības mezgliem.

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	

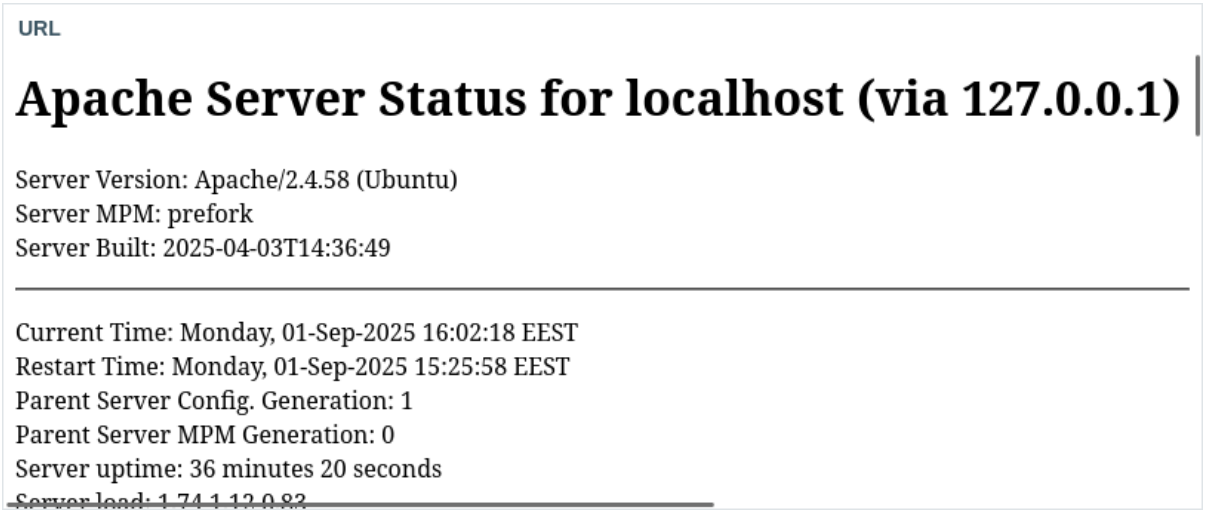
Pulkstenis

Clock logrīks attēlo lokālo, servera vai hosta laiku analogā vai digitālā formātā.



URL

URL logrīks attēlo saturu, kas iegūts no URL.



Informācijas paneļa sīkrīka 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.

Bieži lietotie parametri

Tālāk norādītie parametri ir kopīgi visiem logrīkiem:

Nosaukums	Ievadiet logrīka nosaukumu.
Atsvaidzināšanas intervāls	Konfigurējiet noklusējuma atsvaidzināšanas intervālu. Logrīku noklusējuma atsvaidzināšanas intervāli atkarībā no logrīka veida ir no *Bez atsvaidzināšanas* līdz *15 minūtēm*. Piemēram: - *Bez atsvaidzināšanas* logrīkam *URL*; - *1 minūte* logrīkam *Darbību žurnāls*; - *15 minūtes* logrīkam *Pulkstenis*. Atsvaidzināšanas intervālus var iestatīt kā noklusējuma vērtību visiem lietotājiem. Pārslēdziet informācijas paneli uz rediģēšanas režīmu, noklikšķiniet uz pogas rediģēt logrīku un nolaižamajā sarakstā atlasiet vēlamo atsvaidzināšanas intervālu. Katrs lietotājs var iestatīt arī savu logrīka atsvaidzināšanas intervālu. Informācijas paneļa skatīšanas režīmā noklikšķiniet uz trīs punktu pogas logrīkā un nolaižamajā sarakstā atlasiet vēlamo atsvaidzināšanas intervālu. Ņemiet vērā, ka lietotāja unikālajam atsvaidzināšanas intervālam ir prioritāte pār logrīka iestatījumu, un tas tiek saglabāts arī 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 taupītu vietu, un kļūst redzama tikai, novietojot peles kursoru virs logrīka (gan skatīšanas, gan rediģēšanas režīmā). Galvene ir daļēji redzama arī tad, kad logrīks tiek vilkts 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*.

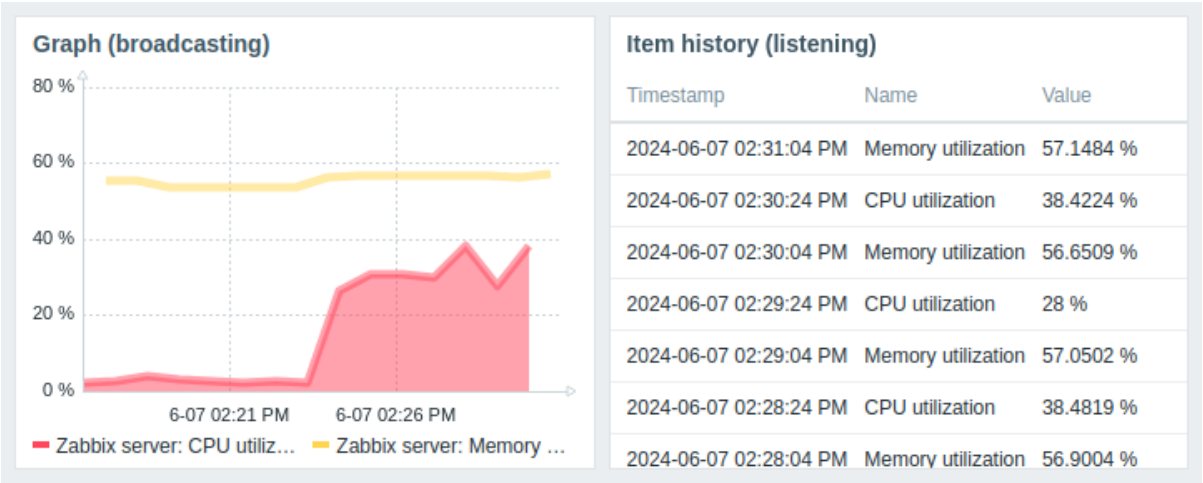
- Logrīks *Geomap* var pārraidīt hostu datus logrīkiem, kas tos saņem (*Honeycomb*, *Top items* u.c.), un var arī saņemt hostu grupu un hostu datus no logrīkiem, kas tos pārraida (*Honeycomb*, *Problem hosts* u.c.).
- Logrīks *Clock* nevar ne pārraidīt, ne saņemt nekādus datus.

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

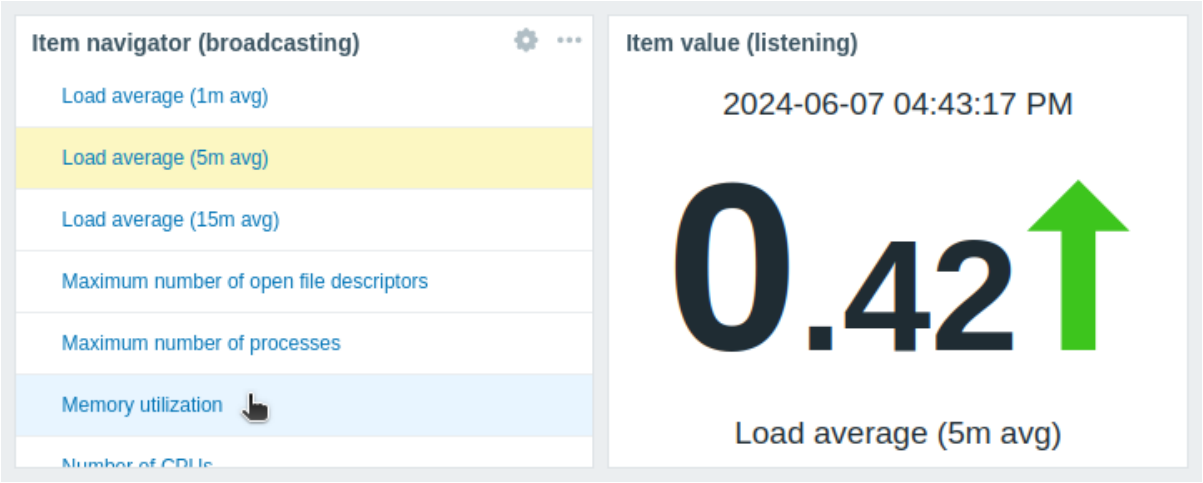
Logrīki atšķiras pēc tā, kā tie **pārraida** datus citiem logrīkiem.

Logrīki, kas atbalsta datu pārraidīšanu, sāk to darīt automātiski pēc izveides. Piemēram, logrīks *Graph* nekavējoties pārraida laika perioda datus klausīšanās režīmā esošajiem logrīkiem.

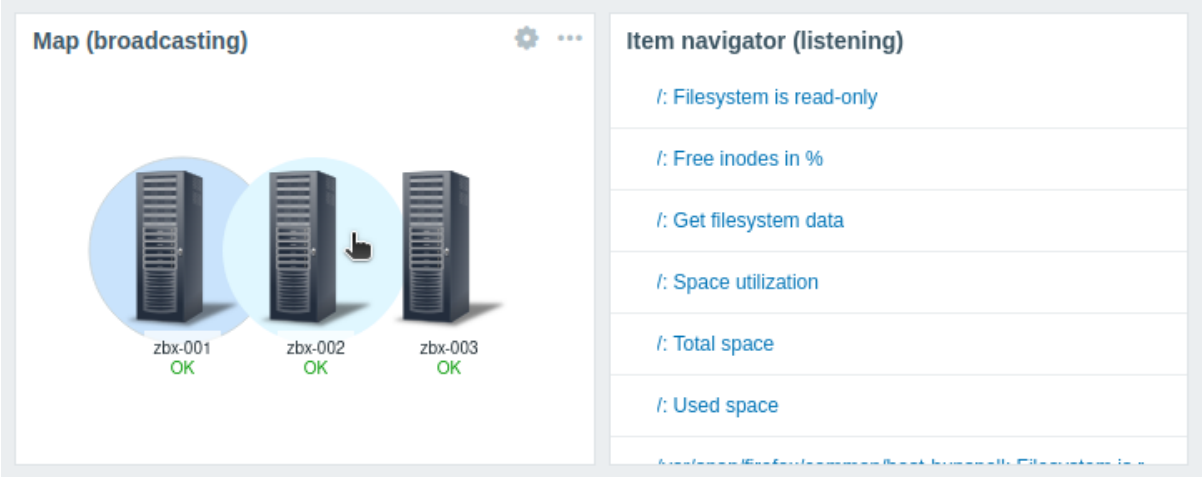


Logrīki, kas atbalsta entītiju atlasi, automātiski pārraida datus par pirmo pieejamo entītiju. Piemēram, logrīks *Item navigator*

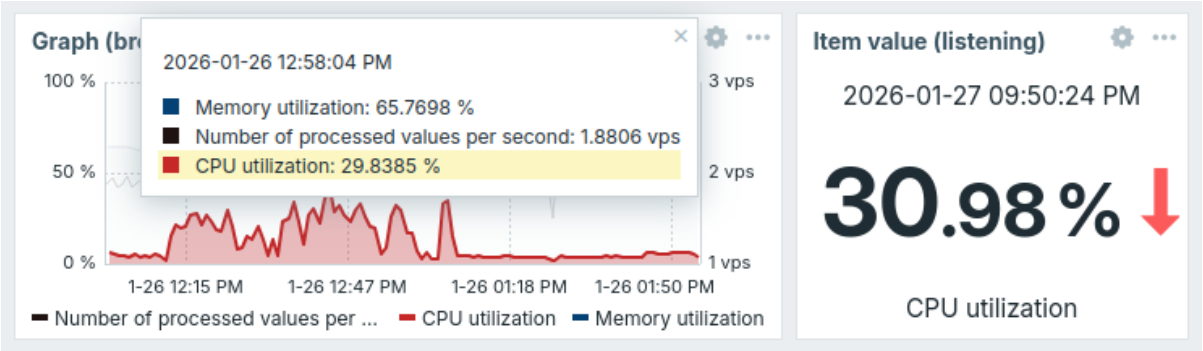
pārraida datus par pirmo vienumu savā vienumu sarakstā. Kad tiek atlasīts cits vienums, tas pārraida datus par šo vienumu. Novietojot peles kursoru virs vienuma, tas tiek izcelts gaiši zilā krāsā; atlasot to, tas tiek izcelts dzeltenā krāsā.



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.

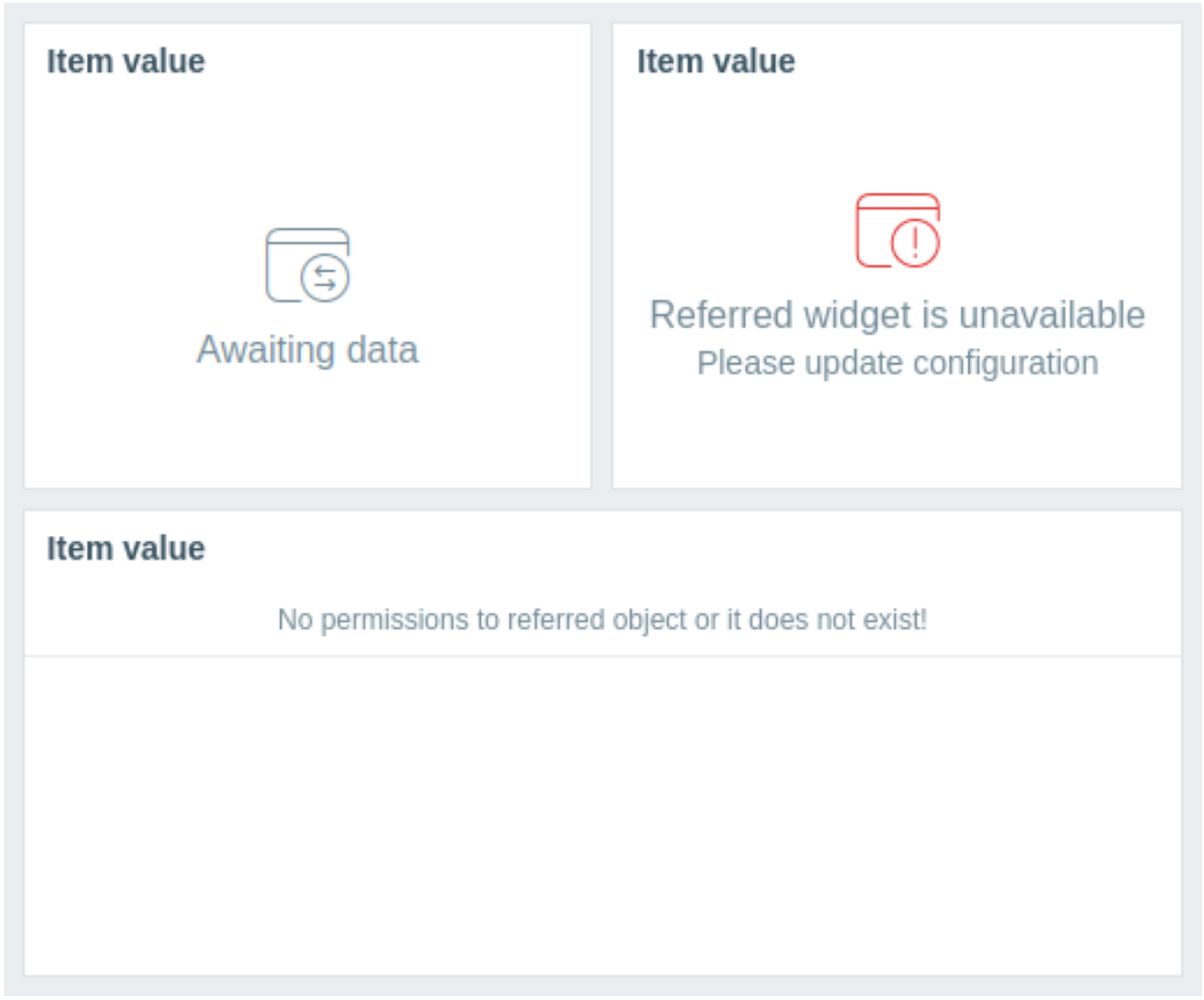


Tādi logrīki kā *Graph* un *Scatter plot* var pārraidīt vienumu datus, kad vienums ir atlasīts to uznirstošajos paskaidrojumos:



Logrīkiem ir arī noteikta darbība, kad tie **saņem** datus no citiem logrīkiem:

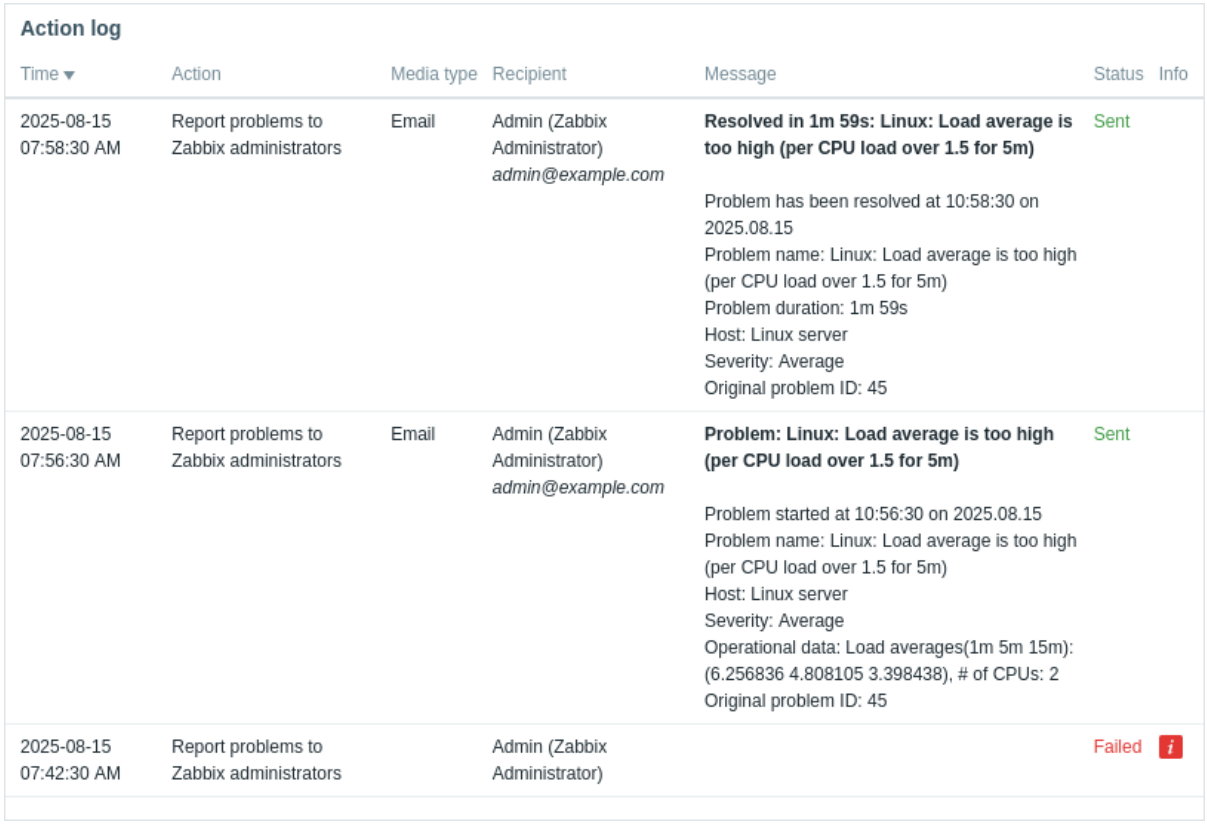
- Ja datu avota logrīks nepārraida datus, klausīšanās režīmā esošais logrīks pāriet stāvoklī *Awaiting data*.
- Ja datu avota logrīks ir dzēsts, aizstāts ar nesaderīgu logrīku vai pārvietots uz citu paneļa lapu, klausīšanās režīmā esošais logrīks pāriet stāvoklī *Referred widget is unavailable*.
- Ja 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!"*



Darbību žurnāls

Pārskats

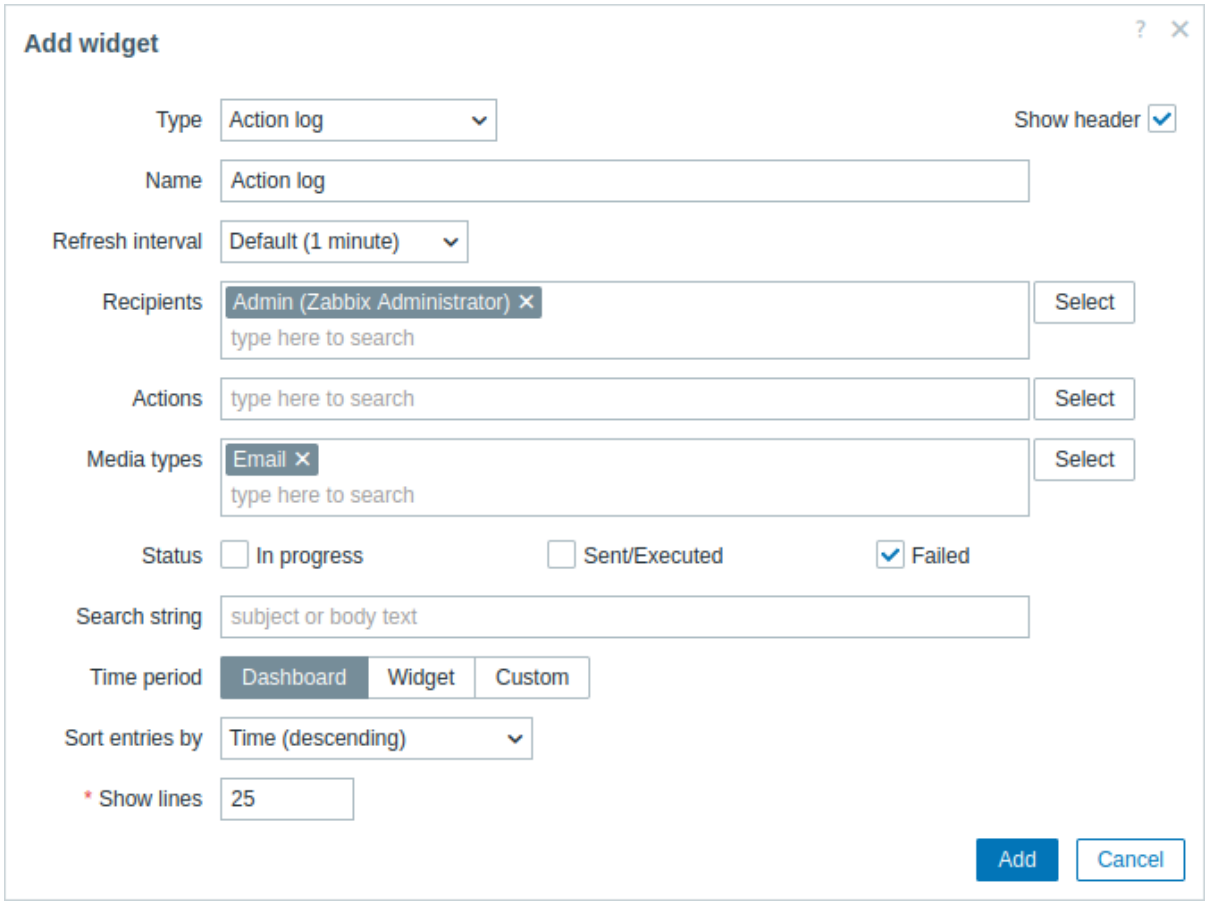
Logrīks *Darbību žurnāls* parāda detalizētu informāciju par operācijām (paziņojumiem, attālinātajām komandām), kas izpildītas darbībā. Detalizētais žurnāls palīdz auditēšanā, problēmu novēršanā un katras darbības izpildes uzraudzībā.



Tas parāda tos pašus datus kā *Atskaites* > *Darbību žurnāls* un var attēlot līdz 1000 ierakstiem.

Konfigurācija

Lai konfigurētu, atlasiet *Action log* kā tipu:



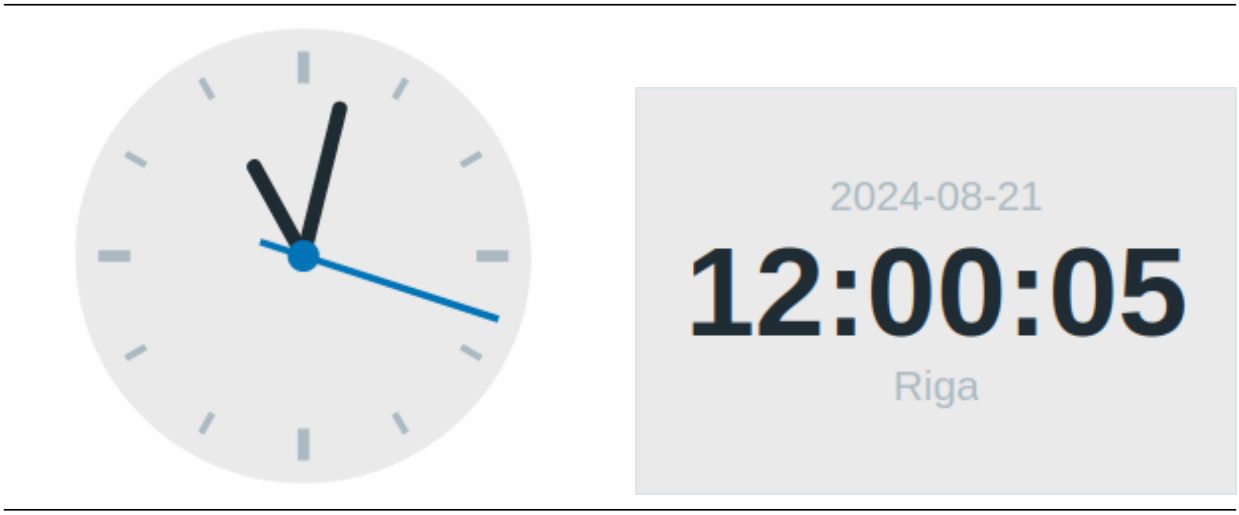
Papildus parametriem, kas ir kopīgi visiem logrīkiem, varat iestatīt šādas specifiskas opcijas:

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.

Pulkstenis

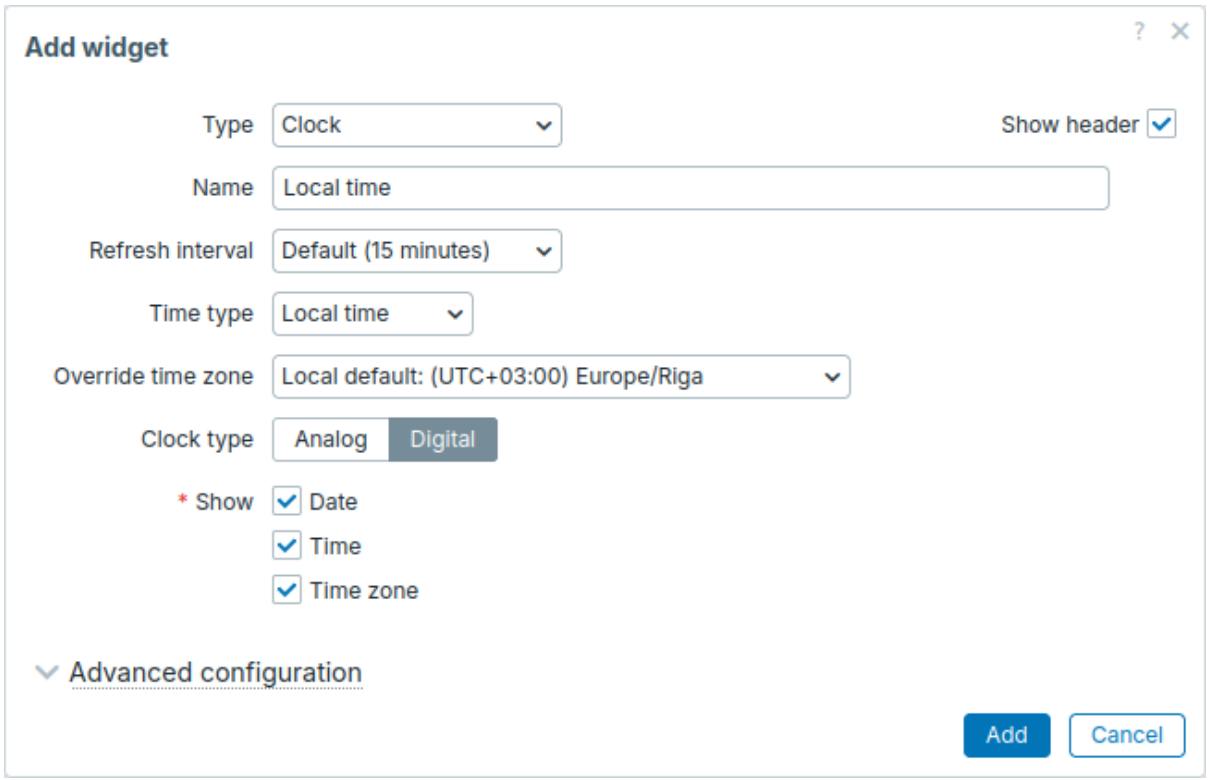
Pārskats

Pulksteņa logrīks attēlo lokālo, servera vai hosta laiku analogā vai digitālā formātā.



Konfigurēšana

Lai konfigurētu, kā tipu atlasiet *Pulkstenis*:



Papildus parametriem, kas ir kopīgi visiem logrīkiem, varat iestatīt tālāk norādītās specifiskās opcijas:

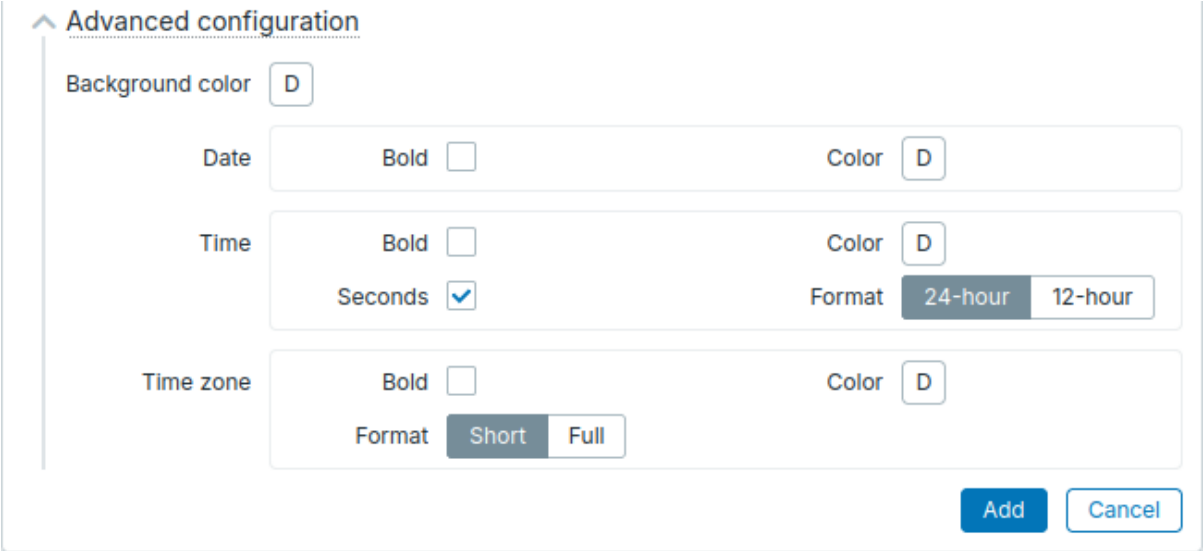
Laika tips	Atlasiet laika avotu: **Lokālais laiks** - ierīces, kurā darbojas pārlūkprogramma, sistēmas laiks; **Servera laiks** - laika josla, kas iestatīta globāli vai Zabbix lietotājam; **Hosts laiks** - laiks, kas iegūts no hosts vienuma.
Vienums	Atlasiet vienumu ar `system.localtime[local]` vienuma atslēgu, lai parādītu hosts laiku. Šis parametrs ir pieejams, ja *Laika tips* ir iestatīts uz "Hosts laiks".
Pulksteņa tips	Atlasiet pulksteņa tipu (analogais vai digitālais).
Ignorēt laika joslu	Atlasiet laika joslu. Šis parametrs ir pieejams laika tipiem *Lokālais laiks* un *Servera laiks*.
Rādīt	Atlasiet vienu vai vairākas informācijas vienības (datumu, laiku, laika joslu), ko rādīt digitālajā pulkstenī. Vienības izmērs tiek automātiski pielāgots parādītajām vienībām, logrīka izmēram un displeja mērogam. Šis parametrs ir pieejams, ja *Pulksteņa tips* ir iestatīts uz "Digitālais".

Papildu konfigurēšana	Noklikšķiniet uz galvenes *Papildu konfigurēšana*, lai parādītu digitālā pulksteņa papildu konfigurēšanas opcijas. Šī sadaļa ir pieejama, ja *Pulksteņa tips* ir iestatīts uz “Digitālais”.

Papildu konfigurācija

Papildu konfigurācijas opcijas ir pieejamas izvēršamajā sadaļā *Papildu konfigurācija* un tikai tiem elementiem, kas ir atlasīti laukā *Rādīt* (skatīt iepriekš).

Turklāt papildu konfigurācija ļauj mainīt visa logrīka fona krāsu.

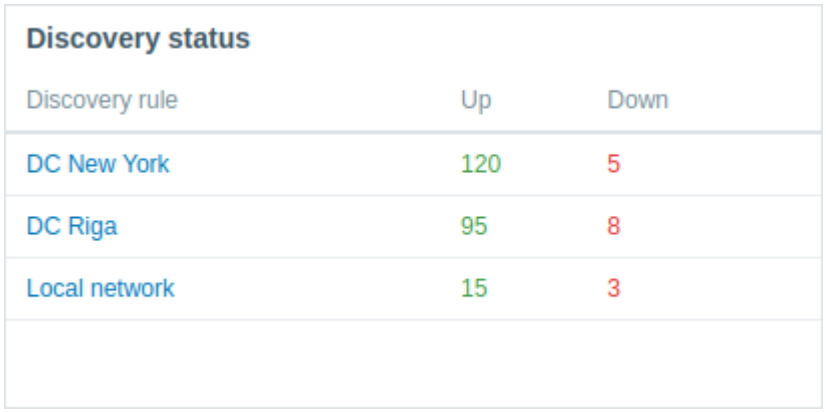


Fona krāsa		Atlasiet fona krāsu krāsu atlasītājā. D apzīmē noklusējuma krāsu (tā ir atkarīga no lietotāja saskarnes tēmas). Lai atgrieztos pie noklusējuma vērtības, krāsu atlasītājā noklikšķiniet uz pogas *Izmantot noklusējumu*.
Datums		Šī sadaļa ir pieejama, ja parametram *Rādīt* ir atlasīta izvēles rūtiņa *Datums*.
	Treknraksts	Atzīmējiet izvēles rūtiņu, lai datums tiktu rādīts treknrakstā.
	Krāsa	Atlasiet datuma krāsu krāsu atlasītājā. D apzīmē noklusējuma krāsu (tā ir atkarīga no lietotāja saskarnes tēmas). Lai atgrieztos pie noklusējuma vērtības, krāsu atlasītājā noklikšķiniet uz pogas *Izmantot noklusējumu*.
Laiks		Šī sadaļa ir pieejama, ja parametram *Rādīt* ir atlasīta izvēles rūtiņa *Laiks*.
	Treknraksts	Atzīmējiet izvēles rūtiņu, lai laiks tiktu rādīts treknrakstā.
	Krāsa	Atlasiet laika krāsu krāsu atlasītājā. D apzīmē noklusējuma krāsu (tā ir atkarīga no lietotāja saskarnes tēmas). Lai atgrieztos pie noklusējuma vērtības, krāsu atlasītājā noklikšķiniet uz pogas *Izmantot noklusējumu*.
	Sekundes	Atzīmējiet izvēles rūtiņu, lai tiktu rādītas sekundes. Pretējā gadījumā tiks rādītas tikai stundas un minūtes.
	Formāts	Atlasiet, vai laiks jāparāda 24 stundu vai 12 stundu formātā.
Laika josla		Šī sadaļa ir pieejama, ja parametram *Rādīt* ir atlasīta izvēles rūtiņa *Laika josla*.
	Treknraksts	Atzīmējiet izvēles rūtiņu, lai laika josla tiktu rādīta treknrakstā.
	Krāsa	Atlasiet laika joslas krāsu krāsu atlasītājā. D apzīmē noklusējuma krāsu (tā ir atkarīga no lietotāja saskarnes tēmas). Lai atgrieztos pie noklusējuma vērtības, krāsu atlasītājā noklikšķiniet uz pogas *Izmantot noklusējumu*.
	Formāts	Atlasiet, vai laika josla jāparāda īsajā formātā (piemēram, `New York`) vai pilnajā formātā (piemēram, `(UTC-04:00) America/New York`).

Atklāšanas statuss

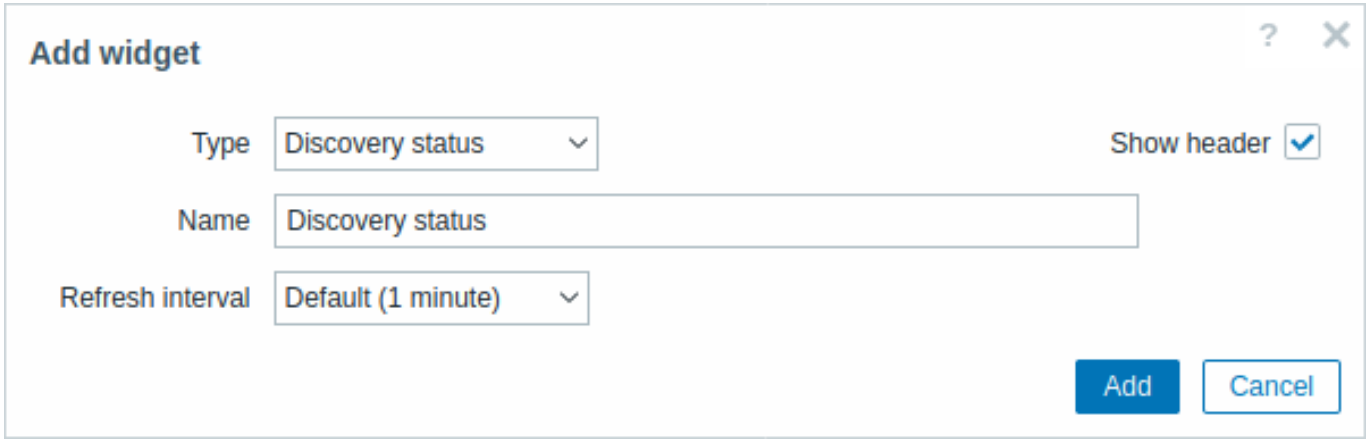
Pārskats

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

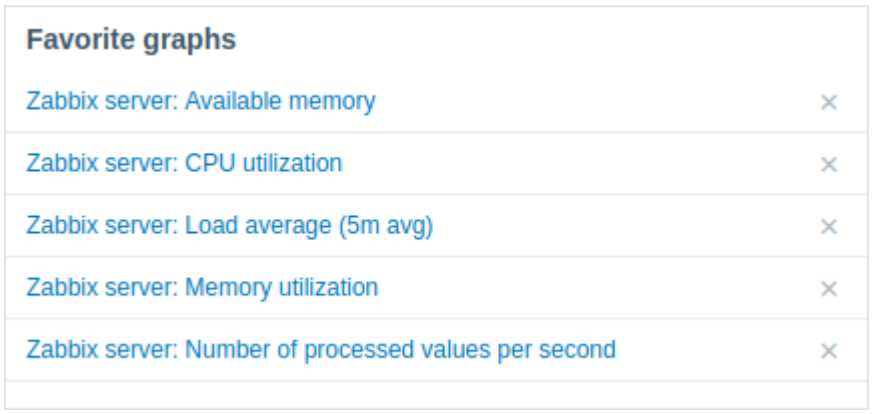
Visi šī logrīka konfigurācijas parametri ir kopīgi visiem logrīkiem.



Izlases grafiki

Pārskats

Logrīks *Iecienītākie grafiki* parāda pašreizējā lietotāja kā iecienītus atzīmēto grafiku saīšņu sarakstu alfabētiskā secībā.



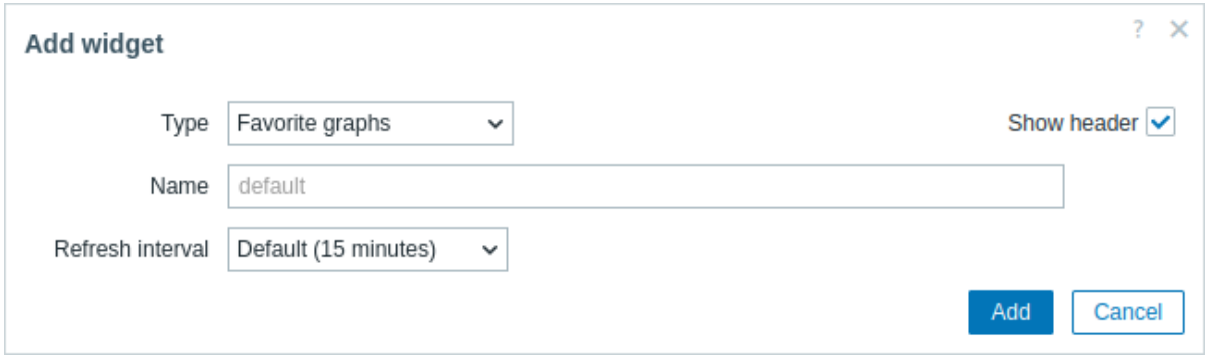
Noklikšķinot uz x, grafiks tiek noņemts no saraksta un no iecienītajiem.

Grafiki tiek atzīmēti kā iecienīti, noklikšķinot uz pogas  *Pievienot iecienītajiem*, skatot vienuma grafikus sadaļā *Monitoring* > *Latest data*.



Konfigurācija

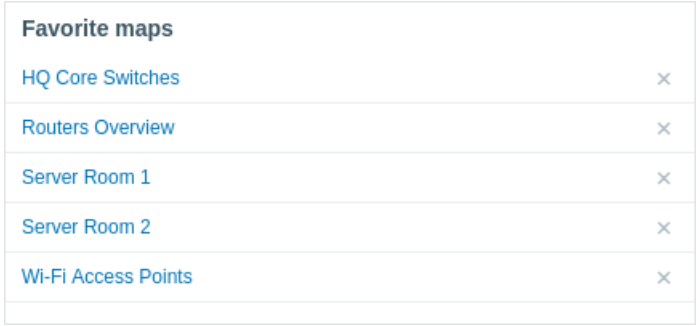
Visi šī logrīka konfigurācijas parametri ir kopīgi visiem logrīkiem.



Izlases kartes

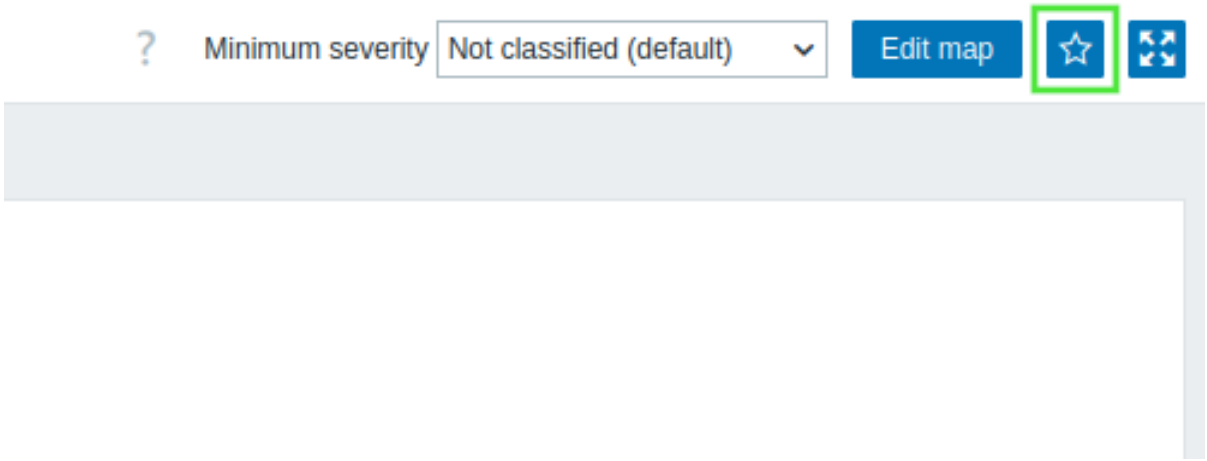
Pārskats

Logrīks *Iecienītās kartes* parāda pašreizējā lietotāja kā iecienītas atzīmēto karšu saīšņu sarakstu, kas sakārtots alfabētiskā secībā.



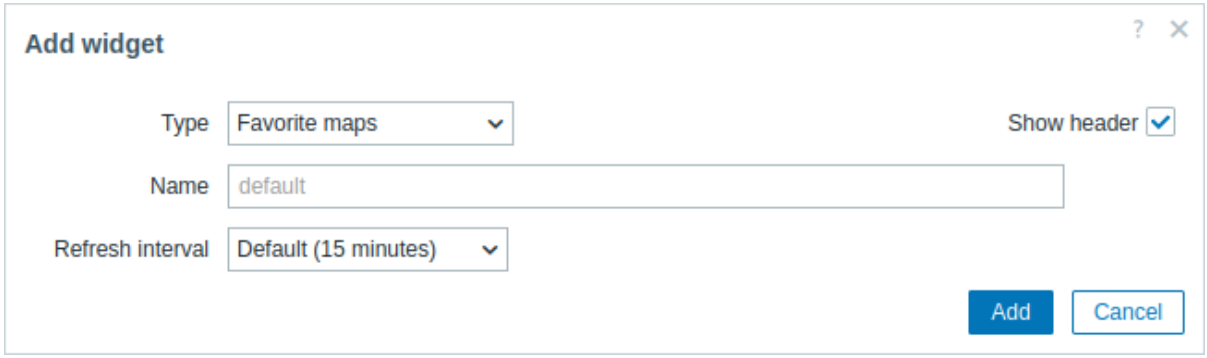
Noklikšķinot uz x, karte tiek noņemta no iecienītajām.

Kartes tiek atzīmētas kā iecienītas, noklikšķinot uz pogas  *Pievienot iecienītajām*, skatot kartes sadaļā *Uzraudzība* > *Kartes*.



Konfigurācija

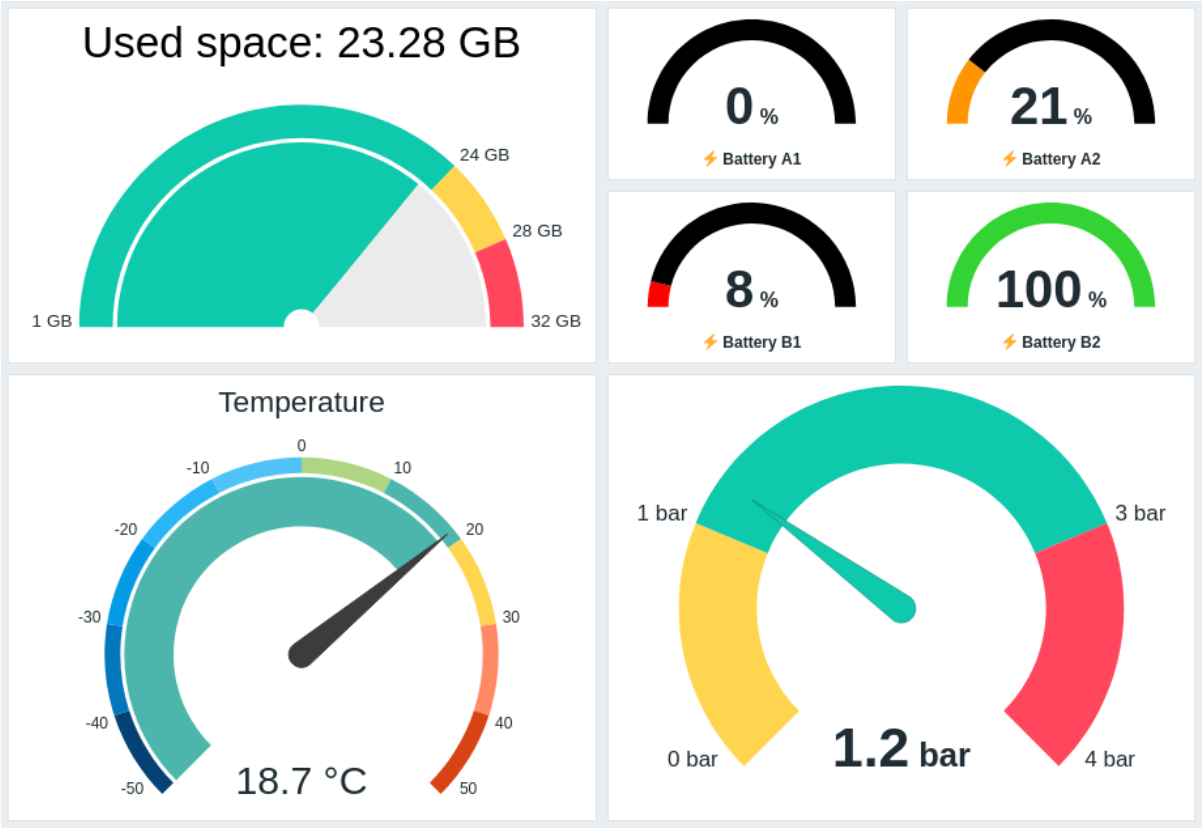
Visi šī logrīka konfigurācijas parametri ir kopīgi visiem logrīkiem.



Mērītājs

Pārskats

Gauge logrīks attēlo viena skaitliska vienuma vērtību kā mērinstrumentu. Tas ir noderīgs, lai sekotu svarīgiem rādītājiem, vizualizētu sliekšņus un noteiktu pēkšņas izmaiņas.



Varat konfigurēt logrīku, lai tas attēlotu tālāk norādīto:

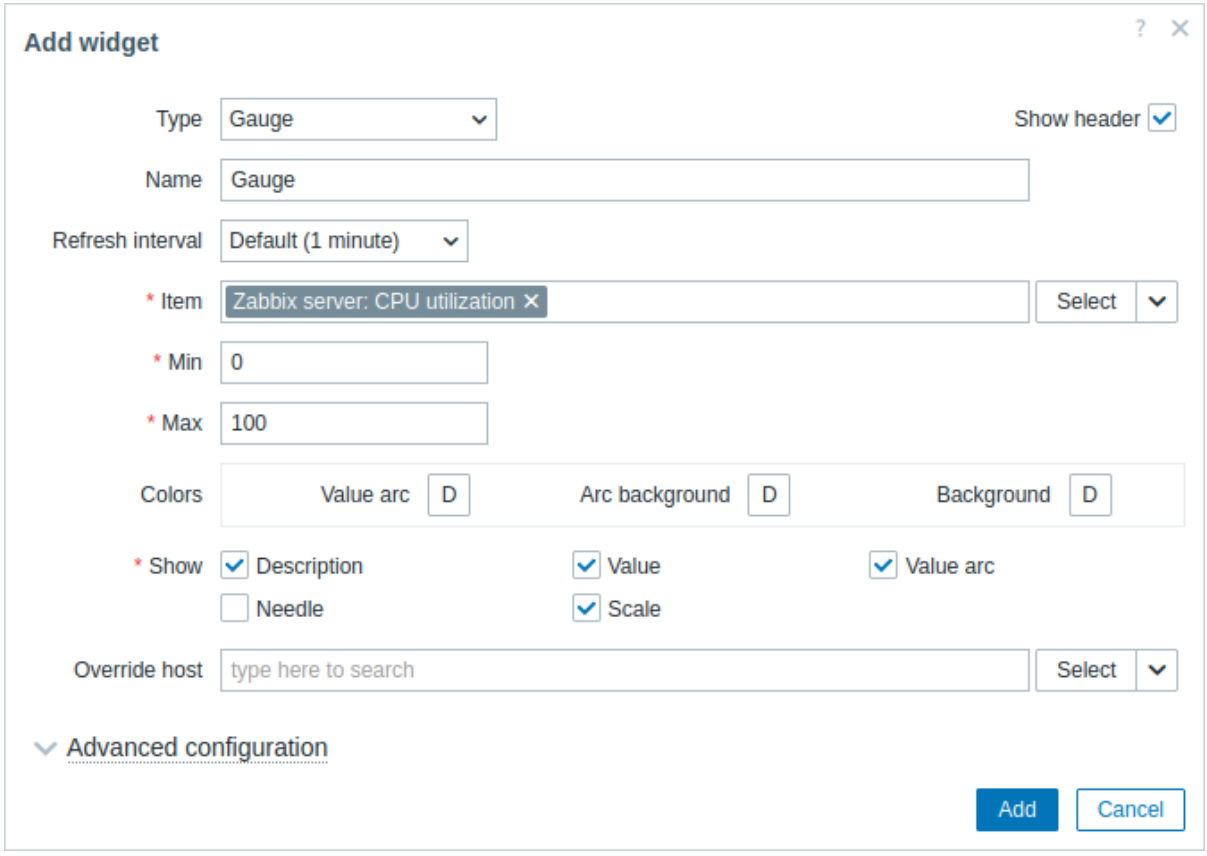
- Vienuma apraksts (Used space: {ITEM.LASTVALUE}, Temperature)
- Vienuma vērtība un mērvienības (21 %, 18.7 °C, 1.2 bar)
- Mērinstrumenta skala (1GB/24GB/28GB/32GB, -50/-40/-30/etc.)
- Mērinstrumenta loks (vērtības loks un sliekšņu loks)
- Mērinstrumenta adata

Noklikšķinot uz logrīka, vienumam tiek atvērts vienkāršs grafiks.

Gauge logrīkā attēloto informāciju var lejupielādēt kā PNG attēlu, logrīka izvēlnē *Actions* atlasot opciju *Download image* , kas tiek parādīta, novietojot kursoru virs logrīka. Atkarībā no pārlūkprogrammas iestatījumiem attēls tiek automātiski saglabāts mapē `Downloads` vai tiek atvērts dialoglodziņš *Save*, kurā varat izvēlēties saglabāšanas vietu.

Konfigurācija

Lai konfigurētu, atlasiet *Gauge* kā tipu:

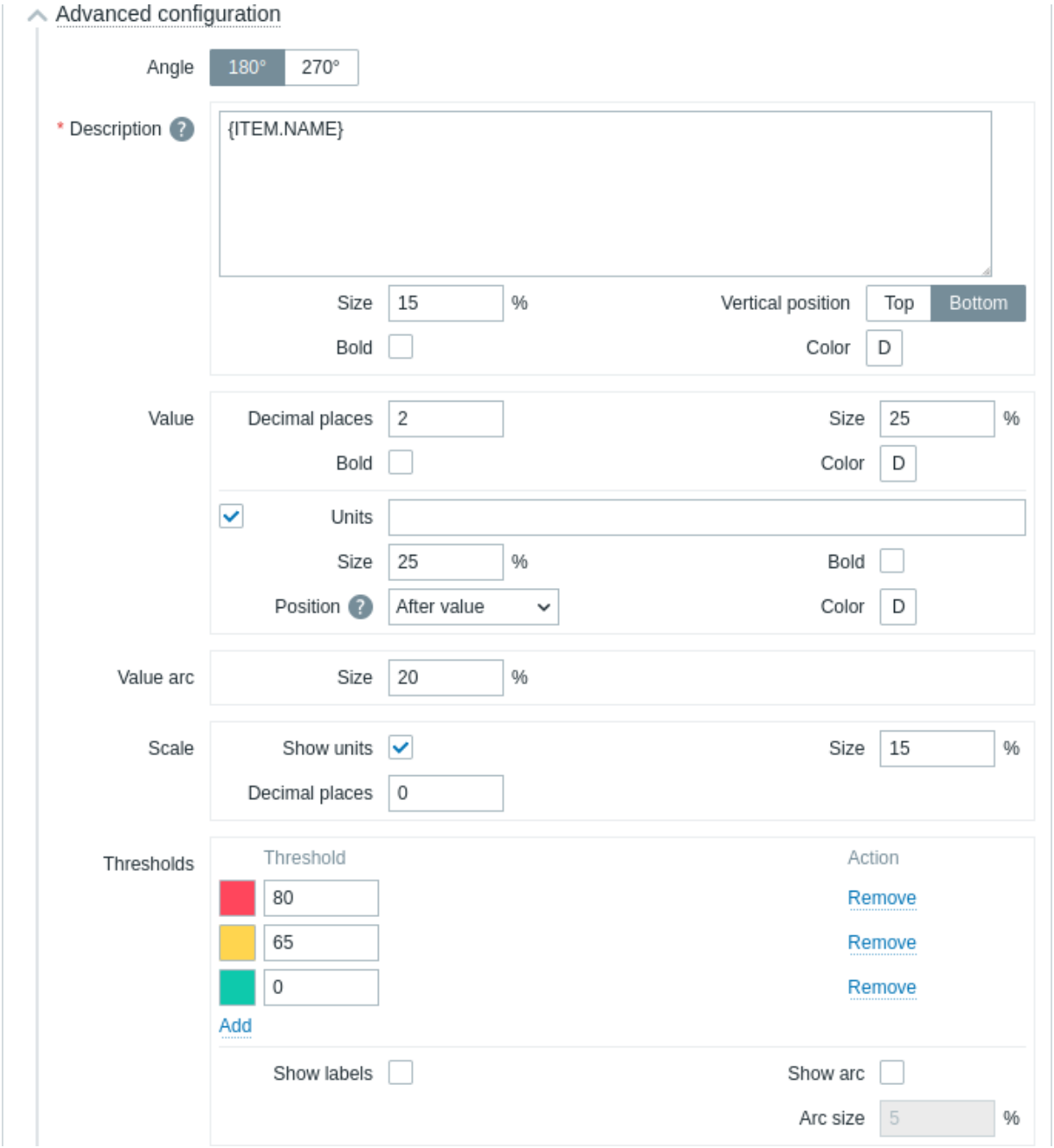


Papildus parametriem, kas ir kopīgi visiem logrīkiem, varat iestatīt šādas specifiskās opcijas:

Item	Atlasiet vienumu. Alternatīvi atlasiet 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*, **Value arc** noklusējuma krāsa 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 tiek parādīts mērinstrumenta vērtības loks vai mērinstrumenta sliekšņu loks (skatiet paplašinātās konfigurācijas opcijas). Ņemiet vērā arī to, ka, ja tiek parādīts mērinstrumenta rādītājs, vērtība tiek novietota zem rādītāja; ja rādītājs ir paslēpts, vērtība tiek līdzināta ar mērinstrumenta loka apakšdaļu.
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 virsraksta *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*:



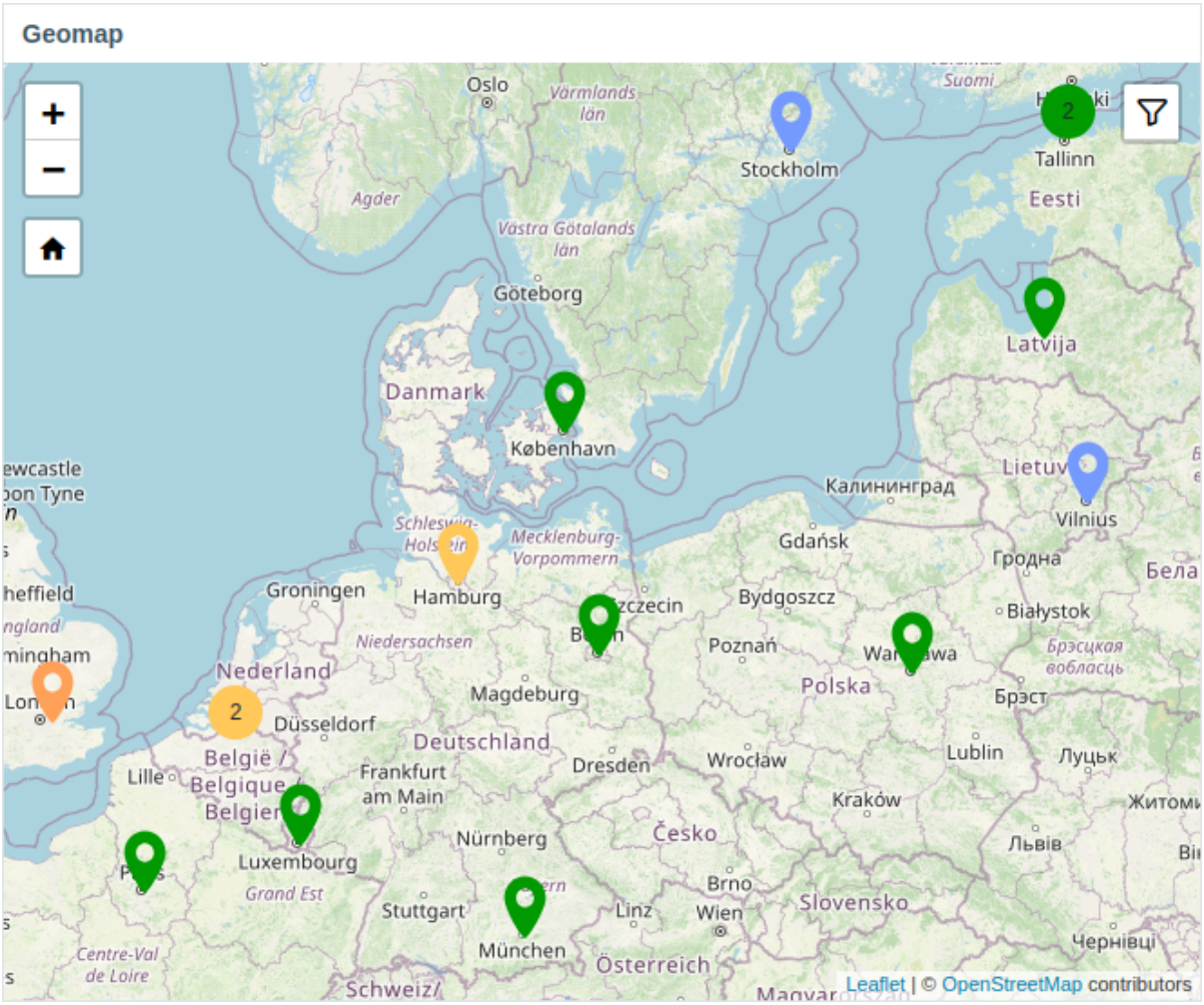
Leņķis	Izvēlieties mērierīces leņķi (180° vai 270°).
Apraksts	
Apraksts	Ievadiet vienuma aprakstu. Šis apraksts var aizstāt noklusējuma vienuma nosaukumu. Tiek atbalstīti vairākrindu apraksti. Ir iespējama teksta un atbalstīto makrosu kombinācija. Tiek atbalstīti {HOST.*}, {ITEM.*}, {INVENTORY.*} un 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).

Geomap

Pārskats

Geomap logrīks attēlo hostus kā marķierus interaktīvā uz Leaflet balstītā kartē, kur marķieru krāsas norāda katra hosta vissmagāko problēmu.

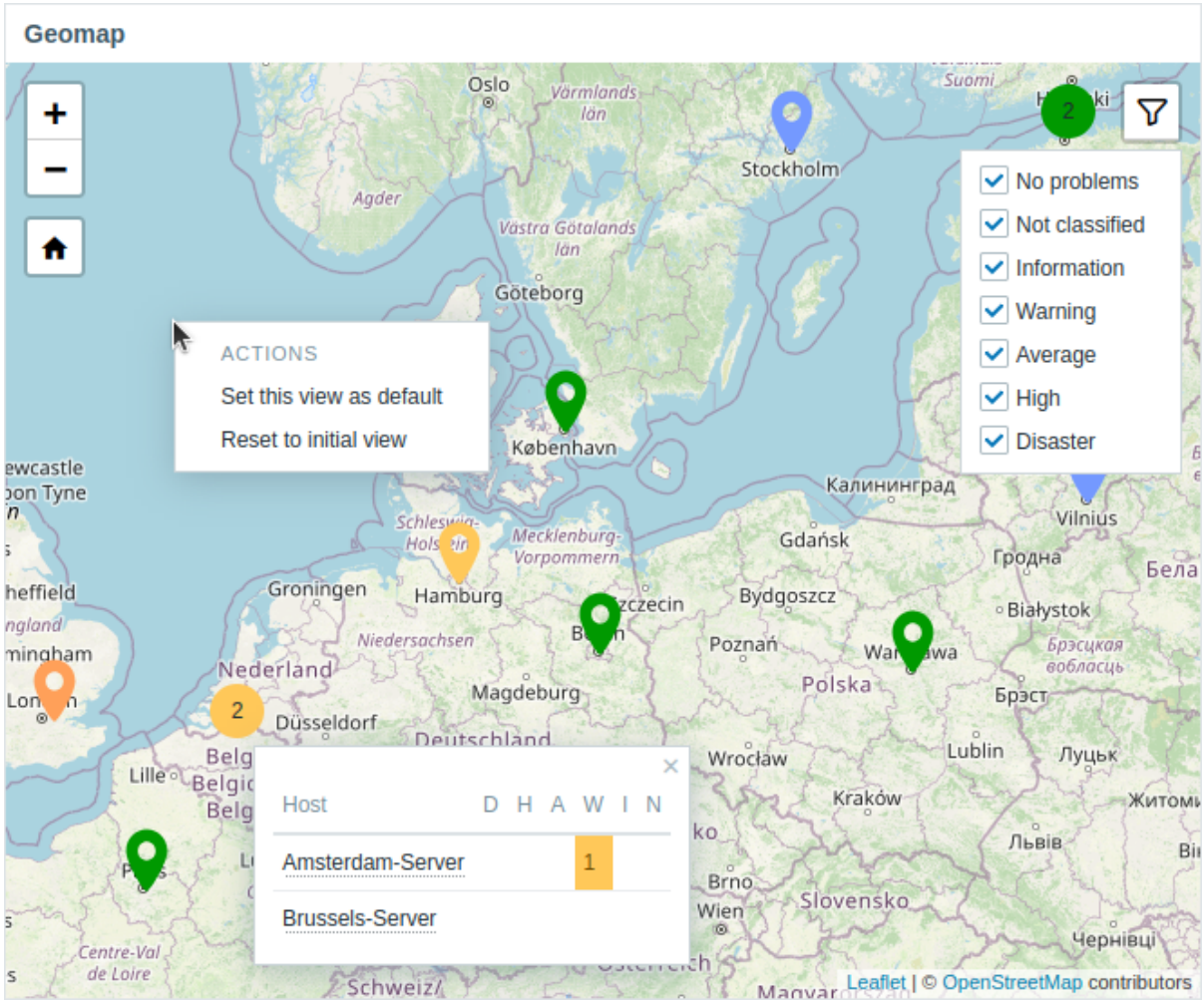


Note:
Zabbix piedāvā vairākus iepriekš definētus karšu flīžu nodrošinātājus un ļauj pievienot pielāgotus nodrošinātājus sadaļā *Administration* > *General* > *Geographical maps*.

Pēc noklusējuma logrīks attēlo visus iespējotos hostus, kuriem to inventarizācijā ir derīgas koordinātas (platums no -90 līdz 90, garums no -180 līdz 180).

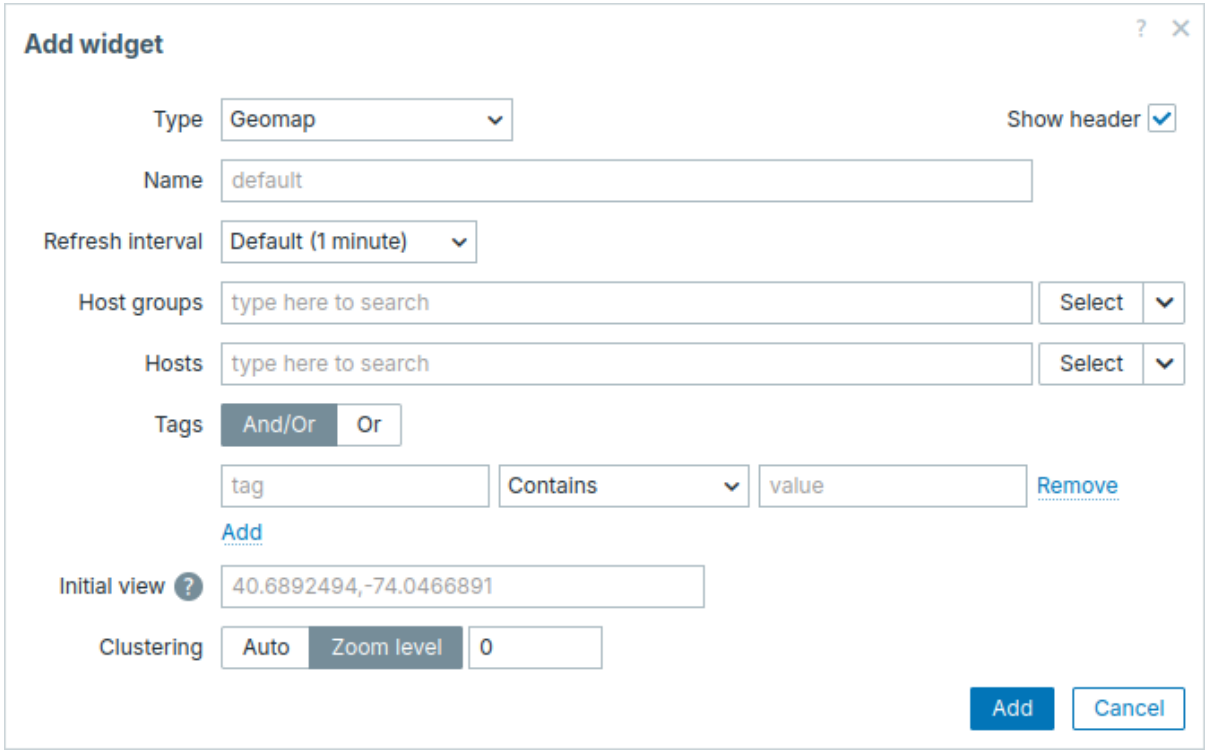
Izmantojiet tālāk norādītās opcijas, lai mijiedarbotos ar logrīku:

- 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.



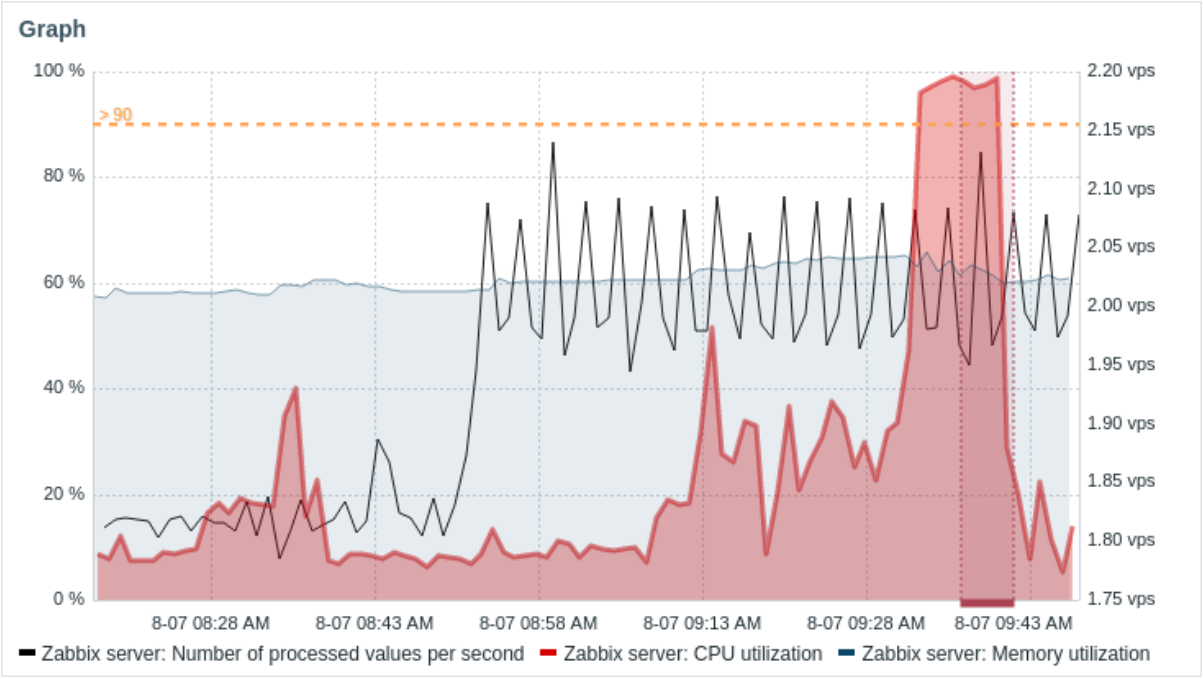
Papildus parametriem, kas ir kopīgi visiem logrīkiem, varat iestatīt šādas specifiskas opcijas:

Hostu grupas	Atlasiet hostu grupas, kas tiks attēlotas kartē. 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. 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. 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 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 tipi: **Un/Vai** - visiem nosacījumiem jābūt izpildītiem, nosacījumi ar vienādu birkas nosaukumu tiks grupēti pēc *Vai* nosacījuma; **Vai** - pietiek, ja ir izpildīts viens nosacījums. Šis parametrs nav pieejams, konfigurējot logrīku veidnes panelī.
Sākotnējais skats	Ar komatiem atdalītas centra koordinātas un neobligāts tālummaiņas līmenis, kas jāattēlo, kad logrīks tiek sākotnēji ielādēts, formātā `<latitude>,<longitude>,<zoom>` Ja ir norādīta sākotnējā tālummaiņa, Geomap logrīks tiek ielādēts ar norādīto tālummaiņas līmeni. Pretējā gadījumā sākotnējā tālummaiņa tiek aprēķināta 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.

Grafiks

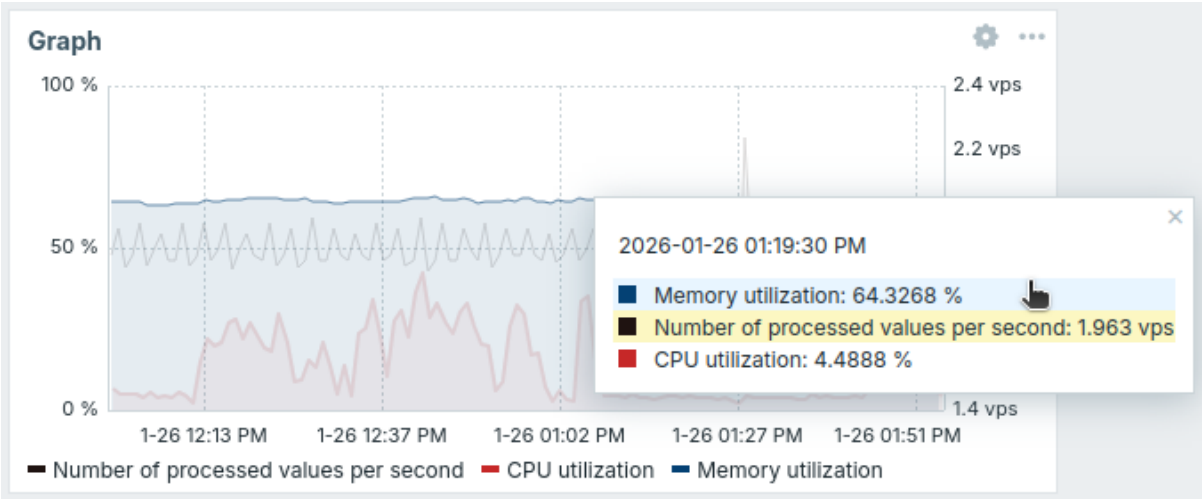
Pārskats

Graph logrīks attēlo skaitliskus vienuma datus vektorgrafikā. Tas palīdz sekot tendencēm, pamanīt problēmas un salīdzināt vērtības laika gaitā un dažādos hosts.

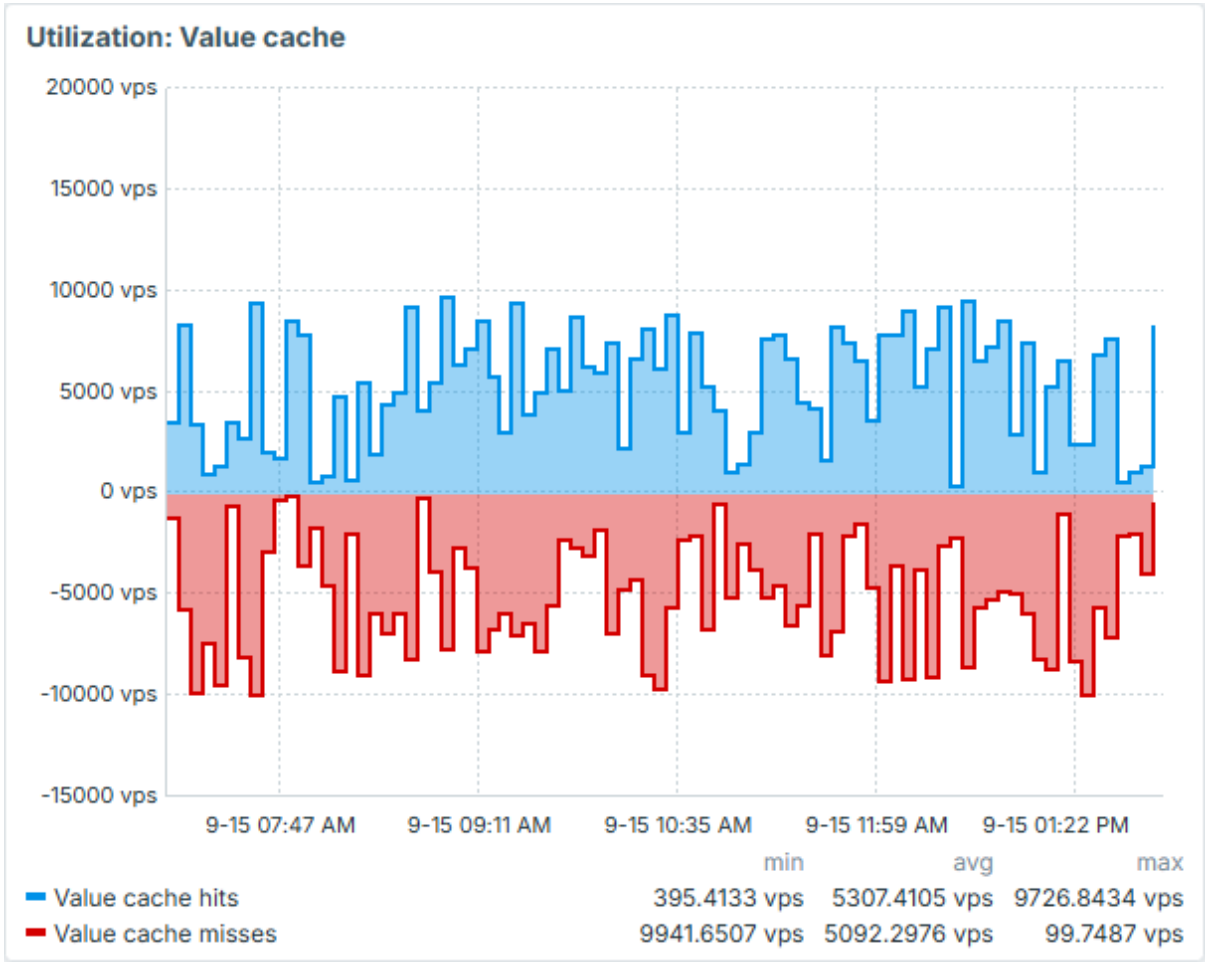


Novietojot kursoru virs grafika, tiek parādīts rīka padoms ar vienumu sarakstu, kas sakārtots dilstošā vērtību secībā.

Novietojot kursoru virs vienuma rīka padomā, tiek izcelts tā grafiks (pārējie tiek aptumšoti); atlasot vienumu, tā dati tiek pārraidīti citiem logrīkiem.



Grafiku var attēlot ar apgrieztām Y ass vērtībām; tas ietekmē tikai vizuālo attēlojumu un nemaina sākotnējās vērtības.

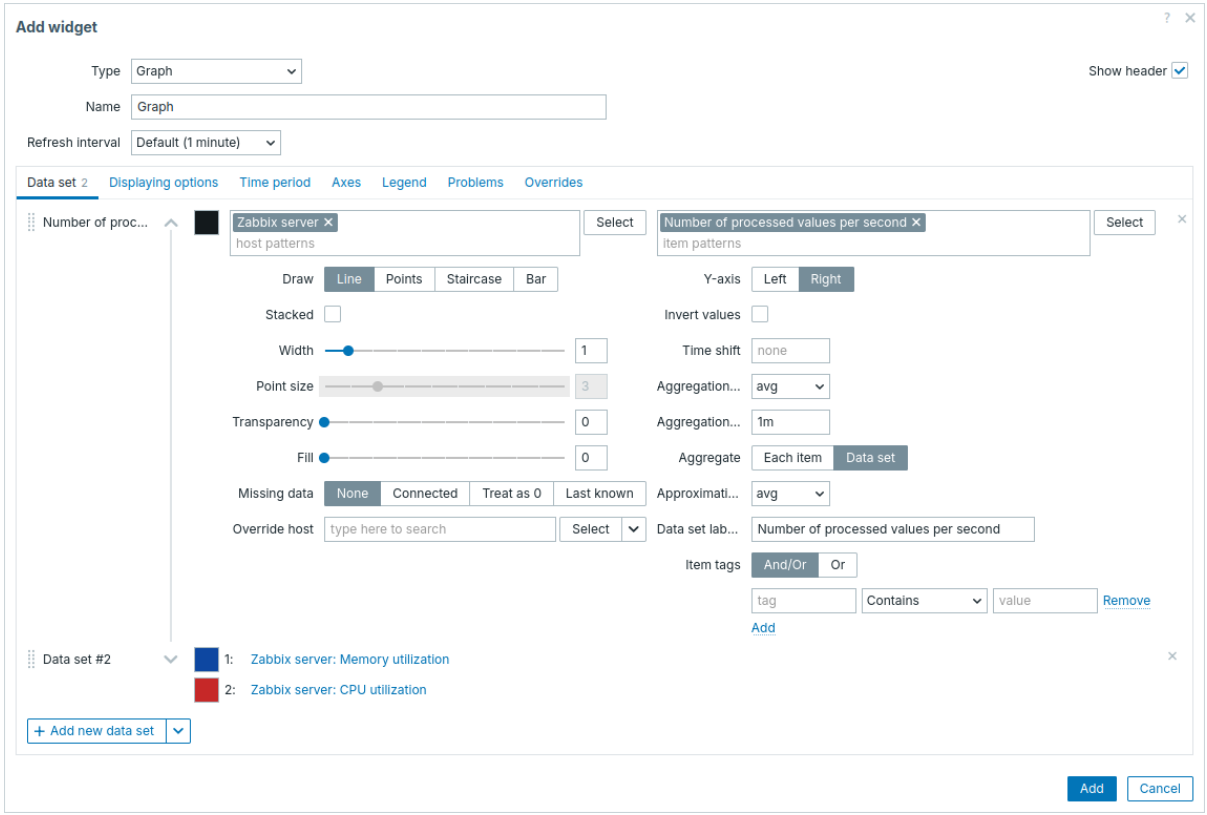


Ir pieejams arī mantotais Graph (classic) logrīks.

Graph logrīkā attēloto informāciju var lejupielādēt kā PNG attēlu, logrīka izvēlnē *Actions* ..., kas tiek parādīta, novietojot kursoru virs logrīka, atlasot opciju *Download image*. Atkarībā no pārlūkprogrammas iestatījumiem attēls tiek automātiski saglabāts mapē `Downloads` vai tiek atvērts dialoglodziņš *Save*, kurā var izvēlēties saglabāšanas vietu.

Konfigurācija

Lai konfigurētu, atlasiet *Graph* kā tipu:



Datu kopa

Cilne **Datu kopa** ļauj atlasīt diagrammas datus, pievienojot datu kopas. Var pievienot divu veidu datu kopas:

- *Vienumu šabloni* - tiek parādīti dati no vienumiem, kas atbilst šabloniem. Var izvēlēties vienu pamatkrāsu vai paletes rindu, lai katram atbilstošajam vienumam piešķirtu atšķirīgu krāsu.
- *Vienumu saraksts* - tiek parādīti dati no atlasītajiem vienumiem. Katra vienuma krāsu var izvēlēties atsevišķi krāsu atlasītājā.

Pēc noklusējuma tiek pievienota *Vienumu šabloni* datu kopa.

Datu kopa	**Vienumu šabloni** datu kopai: Atlasiet vai ievadiet hostu un vienumu šablonus; diagrammā tiks parādīti to vienumu dati, kas atbilst šiem šabloniem; var tikt parādīti ne vairāk kā 50 vienumi. Atlasīšanai 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 pastāv citi atbilstoši vienumi (piemēram, *item2*, *item3*). Hosta un vienumu šablonu norādīšana ir obligāta ”Vienumu šabloni” datu kopām. Skatiet arī: Datu kopas konfigurācijas informācija. **Vienumu saraksts** datu kopai: Atlasiet diagrammas vienumus, 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 saraksts” datu kopām. Skatiet arī: Datu kopas konfigurācijas informācija. Ņemiet vērā, ka ir atļauti tikai skaitliska tipa vienumi. Konfigurējot logrīku veidnes informācijas panelī, parametru hostu šablonu norādīšanai nevar izmantot, bet parametrs vienumu saraksta norādīšanai ļauj atlasīt tikai veidnē konfigurētos vienumus.

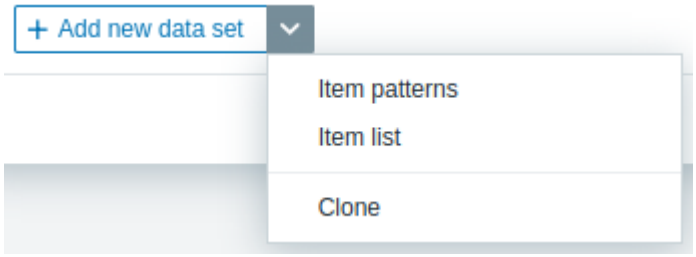
Zīmēšana	Izvēlieties metrikas zīmēšanas veidu. Iespējamie zīmēšanas veidi: *Līnija* (iestatīts pēc noklusējuma), *Punkti*, *Pakāpieni* un *Josla*. Ņemiet vērā, ka, ja līnijas/pakāpienu diagrammā ir tikai viens datu punkts, tas tiek attēlots kā punkts neatkarīgi no zīmēšanas veida. Punkta izmērs tiek aprēķināts no līnijas platuma, taču tas nevar būt mazāks par 3 pikseļiem, pat ja līnijas platums ir mazāks.
Sakrauts	Atzīmējiet izvēles rūtiņu, lai datus parādītu sakrautā veidā (tiek parādīti aizpildīti apgabali). Šī opcija ir atspējota, ja ir atlasīts zīmēšanas veids *Punkti*.
Platums	Iestatiet līnijas platumu. Šī opcija ir pieejama, ja ir atlasīts zīmēšanas veids *Līnija* vai *Pakāpieni*.
Punkta izmērs	Iestatiet punkta izmēru. Šī opcija ir pieejama, ja ir atlasīts zīmēšanas veids *Punkti*.
Caurspīdīgums	Iestatiet caurspīdīguma līmeni.
Aizpildījums	Iestatiet aizpildījuma līmeni. Šī opcija ir pieejama, ja ir atlasīts zīmēšanas veids *Līnija* vai *Pakāpieni*.
Trūkstošie dati	Atlasiet trūkstošo datu attēlošanas opciju: **Nav** - atstarpe paliek tukša; **Savienoti** - tiek savienotas divas robežvērtības; **Uzskatīt par 0** - trūkstošie dati tiek parādīti kā 0 vērtības; **Pēdējā zināmā vērtība** - trūkstošie dati tiek parādīti ar tādu pašu vērtību kā pēdējā zināmā vērtība; nav piemērojams zīmēšanas veidiem *Punkti* un *Josla*.
Ignorēt hostu	Atlasiet saderīgu logrīku vai informācijas paneļa hostu atlasītāju kā hostu datu avotu. Šis parametrs nav pieejams, konfigurējot logrīku veidnes informācijas panelī.
Y ass	Atlasiet diagrammas pusi, kurā tiks parādīta Y ass.
Apgriezt vērtības	Atzīmējiet izvēles rūtiņu, lai pirms parādīšanas diagrammā visas Y ass vērtības reizinātu ar -1. Tas ietekmē tikai vizuālo attēlojumu; sākotnējie dati paliek nemainīti. Šī opcija neietekmē rīka padomu, leģendu vai procentiles līniju. Ja asij ir atlasīta opcija *Vienkāršie trigeri*, trigeru sliekšņi tiks parādīti apgriezti.
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 vai visai datu kopai atlasītajā *Agregācijas intervālā*: **min** - parādīt mazāko vērtību; **max** - parādīt lielāko vērtību; **avg** - parādīt vidējo vērtību; **sum** - parādīt vērtību summu; **count** - parādīt vērtību skaitu; **first** - parādīt pirmo vērtību; **last** - parādīt pēdējo vērtību. Ja ir atlasīta opcija **netiek izmantota**, tiek parādītas visas vērtības (bez agregācijas). Opcijām *min*, *max*, *avg* un *sum* var parādīt tikai skaitliskus datus. Opcijai *count* neskaitliskie dati tiks pārveidoti par skaitliskiem. Skatiet arī: Agregācija diagrammās.
Agregācijas intervāls	Norādiet vērtību agregācijas intervālu. Šajā laukā var izmantot laika sufiksus. Skaitliska vērtība bez sufiksa tiek uzskatīta par sekundēm. Ņemiet vērā: ja logrīks ir konfigurēts vēsturisko datu attēlošanai, pamatojoties uz tendencēm (*Vēstures datu atlase* ir iestatīta uz *Tendences* vai *Automātiski*), ieteicams izmantot agregācijas intervālu, kas ir 1 stundas daudzkārtnis (piemēram, 3600, 60m, 1h, 3h u. c.). Tendencēs tiek glabātas stundas laikā agregētas vērtības, tāpēc agregācijas intervāla izmantošana, kas nav 1 stundas daudzkārtnis (piemēram, 100s, 7min, 15min, 90min u. c.), var radīt grūti interpretējamus rezultātus.
Agregēt	Norādiet, vai veikt agregāciju: **Katrs vienums** - katrs datu kopas vienums tiks agregēts un parādīts atsevišķi; **Datu kopa** - visi datu kopas vienumi tiks agregēti un parādīti kā viena vērtība.

Aproksimācija	Norādiet, kuru vērtību parādīt, ja uz vienu diagrammas vertikālo pikseli attiecas vairāk nekā viena vērtība: **all** - parādīt mazāko, lielāko un vidējo vērtību; **min** - parādīt mazāko vērtību; **max** - parādīt lielāko vērtību; **avg** - parādīt vidējo vērtību. Šis iestatījums ir noderīgs, attēlojot diagrammu par ilgu laika periodu ar biežu atjaunināšanas intervālu (piemēram, viena gada vērtības, kas apkopotas ik pēc 10 minūtēm).
Datu kopas etiķete	Norādiet datu kopas etiķeti, kas tiek parādīta diagrammas *Datu kopas* konfigurācijā un diagrammas *Leģendā* (agregētām datu kopām). Visas datu kopas tiek numurētas, tostarp tās, kurām ir norādīta *Datu kopas etiķete*. Ja etiķete nav norādīta, datu kopa tiek automātiski apzīmēta atbilstoši tās numuram (piemēram, ”Datu kopa Nr. 2”, ”Datu kopa Nr. 3” u. c.). Datu kopu numerācija tiek pārrēķināta pēc datu kopu pārkārtošanas/pārvilkšanas. Pārāk garas datu kopu etiķetes tiks saīsinātas, lai ietilptu attēlošanas vietā (piemēram, ”Procesu sk...”).
Vienumu tagi	Norādiet tagus, pēc kuriem filtrēt logrīkā parādītos vienumus. Var iekļaut vai izslēgt konkrētus tagus un tagu vērtības. Var iestatīt vairākus nosacījumus. Tagu nosaukumu atbilstība vienmēr ir reģistrjutīga. Katram nosacījumam ir pieejami vairāki operatori: **Pastāv** - iekļaut norādītos tagu nosaukumus; **Vienāds ar** - iekļaut norādītos tagu nosaukumus un vērtības (reģistrjutīgi); **Satur** - iekļaut norādītos tagu nosaukumus, kuru tagu vērtības satur ievadīto virkni (apakšvirknes atbilstība, nav reģistrjutīga); **Nepastāv** - izslēgt norādītos tagu nosaukumus; **Nav vienāds ar** - izslēgt norādītos tagu nosaukumus un vērtības (reģistrjutīgi); **Nesatur** - izslēgt norādītos tagu nosaukumus, kuru tagu vērtības satur ievadīto virkni (apakšvirknes atbilstība, nav reģistrjutīga). Nosacījumiem ir pieejami divi aprēķina veidi: **Un/Vai** - jāizpilda visi nosacījumi; nosacījumi ar vienādu taga nosaukumu tiks grupēti, izmantojot nosacījumu Vai; **Vai** - pietiek, ja ir izpildīts viens nosacījums.

Datu kopas konfigurācijas informācija

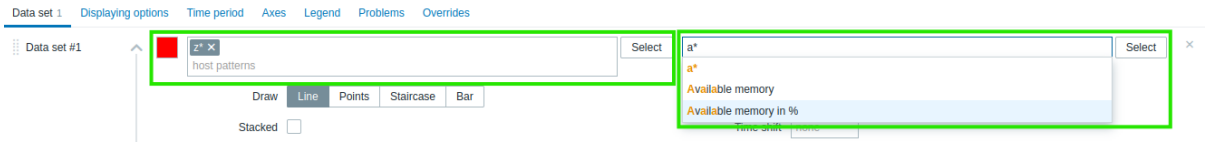
Esošās datu kopas tiek parādītas sarakstā. Varat:

- Noklikšķināt uz pārvietošanas ikonas un pārvilkt 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 krāsu atlasītāju. Varat ievadīt heksadecimālo kodu, izvēlēties vienkrāsainu paraugu vai pārslēgties uz cilni *Palete* un atlasīt iepriekš definētu krāsu rindu. Izvēlētā krāsa tiek tieši lietota *Vienumu saraksta* datu kopām vai izmantota kā pamats ģenerētajiem toņiem *Vienumu rakstu* datu kopās. Izmantojiet Tab, lai pārvietotos starp dialoglodziņa vadīklām, bulttaustiņus, lai pārvietotos pa krāsu paraugiem vai paletes rindām, Enter, lai atlasītu, un Esc, lai atceltu.
- Noklikšķiniet uz pogas *Pievienot jaunu datu kopu*, lai pievienotu tukšu datu kopu, kurā varat atlasīt hostu un vienumu rakstus. Ja noklikšķināt uz lejupvērstās bultiņas ikonas blakus pogai *Pievienot jaunu datu kopu*, tiek parādīta nolaižamā izvēlne, kurā varat pievienot jaunu *Vienumu rakstu* vai *Vienumu saraksta* datu kopu vai *Klonēt* pašlaik atvērto datu kopu. Ja visas datu kopas ir sakļautas, opcija *Klonēt* nav pieejama.

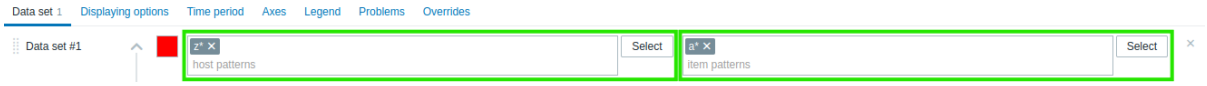


Datu kopā **Vienumu šabloni** ir lauki *Hostu šabloni* un *Vienumu šabloni*, kas abi atpazīst pilnus nosaukumus vai šablonus, kuros ir aizstājējzīme (*). Šī funkcionalitāte ļauj atlasīt visus hostu nosaukumus un vienumu nosaukumus, kas satur izvēlēto šablonu. Rakstot vienuma nosaukumu vai vienuma šablonu laukā *Vienumu šabloni*, nolaižamajā sarakstā tiek parādīti tikai tie vienumi, kas pieder atlasītā(-o) hosta(-u) nosaukumam(-iem).

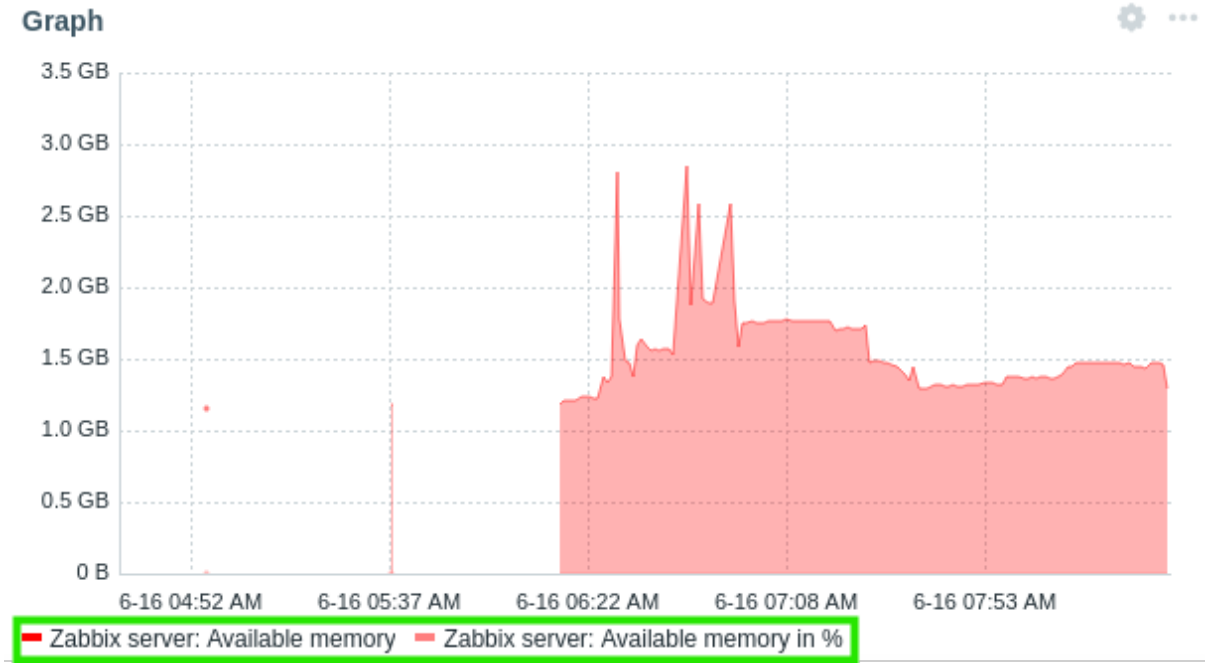
Piemēram, ievadot šablonu **z*** laukā *Hostu šabloni*, nolaižamajā sarakstā tiek parādīti visi hostu nosaukumi, kas satur šo šablonu: **z*, Zabbix serveris, Zabbix starpniekserveris**. Pēc *Enter* nospiešanas šis šablons tiek pieņemts un parādīts kā **z***. Līdzīgi, ievadot šablonu **a*** laukā *Vienumu šabloni*, nolaižamajā sarakstā tiek parādīti visi vienumu nosaukumi, kas satur šo šablonu: **a*, Pieejamā atmiņa, Pieejamā atmiņa %**.



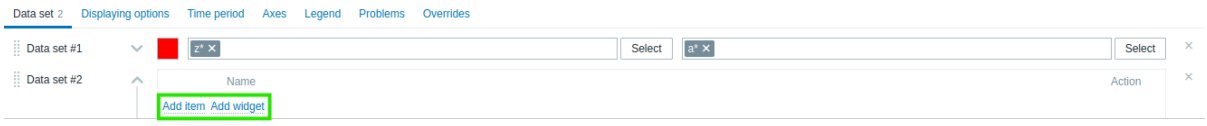
Pēc *Enter* nospiešanas šablons tiek pieņemts un parādīts kā **a***.



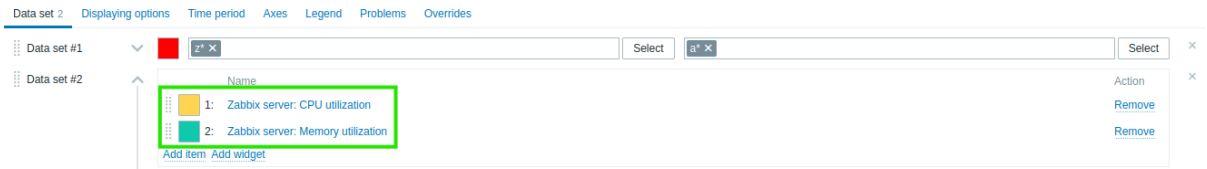
Pēc tam grafikā tiek parādīti visi vienumi, kas pieder atlasītā(-o) hosta(-u) nosaukumam(-iem).



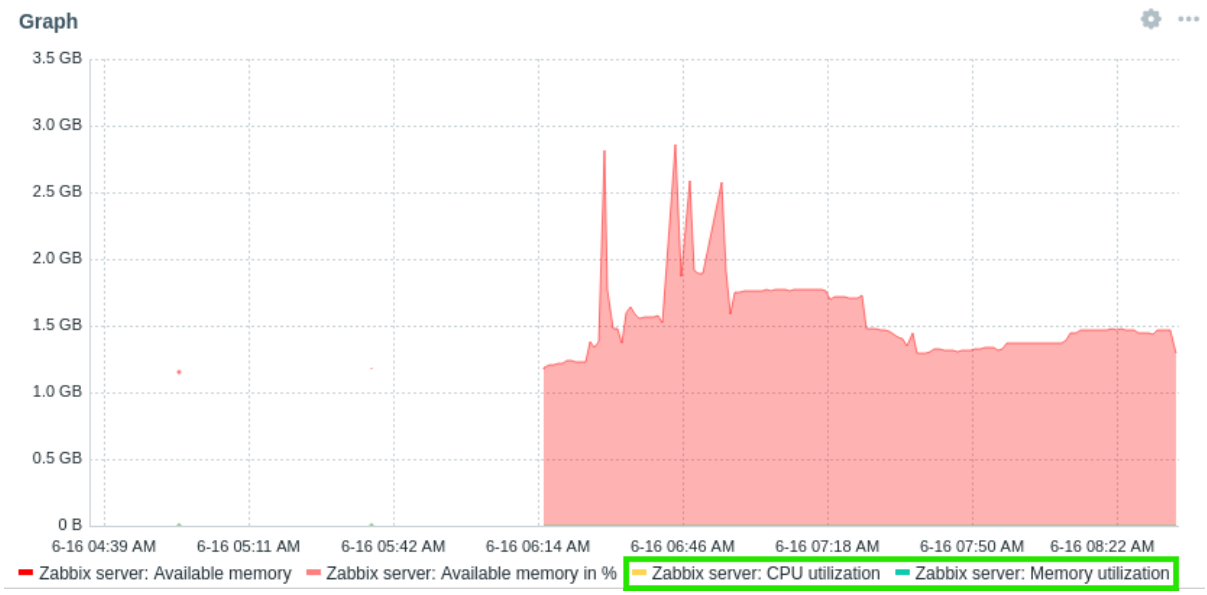
Datu kopa **Vienumu saraksts** satur pogu *Pievienot vienumu*, kas ļauj pievienot vienumus attēlošanai grafikā. Varat arī pievienot saderīgus logrīkus kā vienumu datu avotu, noklikšķinot uz pogas *Pievienot logrīku*.



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.

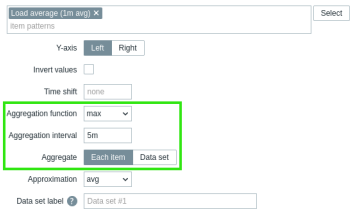


Agregācija grafikos

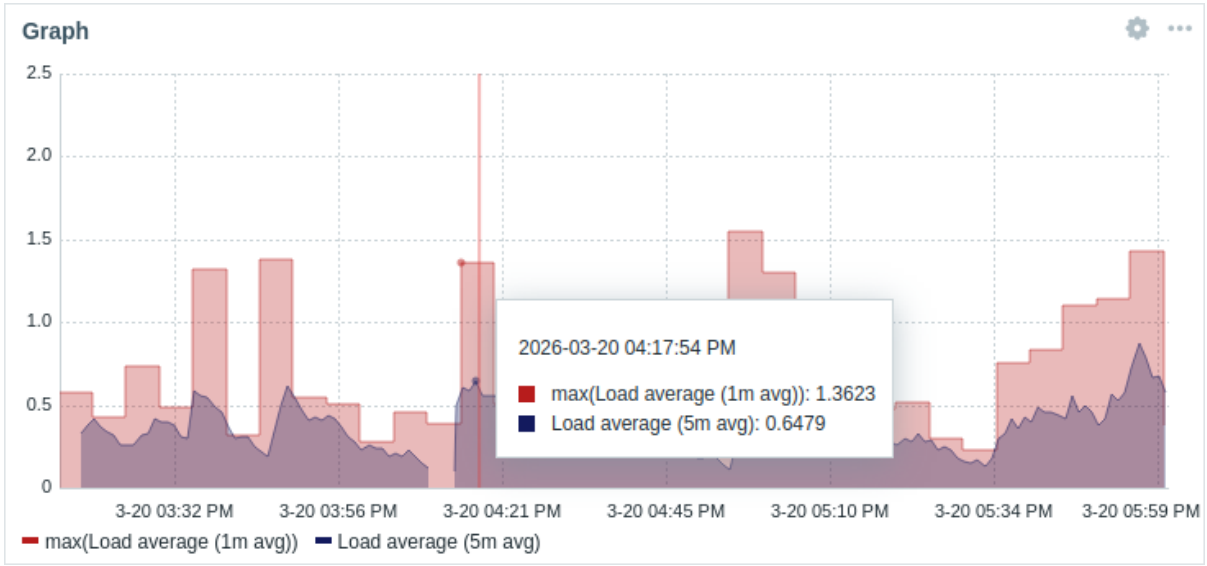
Agregācija ļauj norādītajam intervālam (5 minūtēm, stundai, dienai utt.) visu atsevišķo vērtību vietā parādīt apkopotu vērtību (minimumu, maksimumu, vidējo vērtību utt.). Datu kopā agregāciju var lietot katram vienumam atsevišķi vai visai datu kopai.

Daži agregācijas funkcijas izmantošanas piemēri: - Vidējais pieprasījumu skaits NGINX serverim dienā. - Mazākā diska vieta starp klasteriem nedēļā.

Nākamajā piemērā vienumam tiek lietota agregācijas funkcija ”max” ar piecu minūšu agregācijas intervālu:



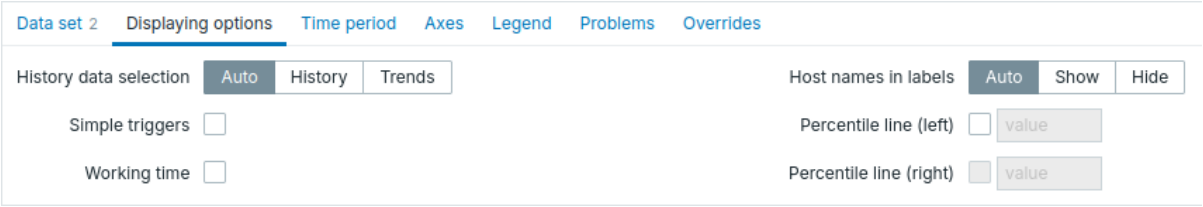
Piemērā ir parādīta maksimālā CPU noslodze piecu minūšu laikā (apkopota no vienas minūtes vidējām vērtībām), salīdzinot ar vidējo CPU noslodzi piecu minūšu laikā.



Ja leģendai ir atlasīta opcija *Rādīt agregācijas funkciju*, vienumi tiek parādīti iekavās, bet leģendā un rīka padomā pirms tiem tiek norādīta agregācijas funkcija. Ja grafikam ir norādīta *Datu kopas etiķete*, tā tiek parādīta iekavās, bet pirms tās tiek norādīta agregācijas funkcija.

Attēlošanas opcijas

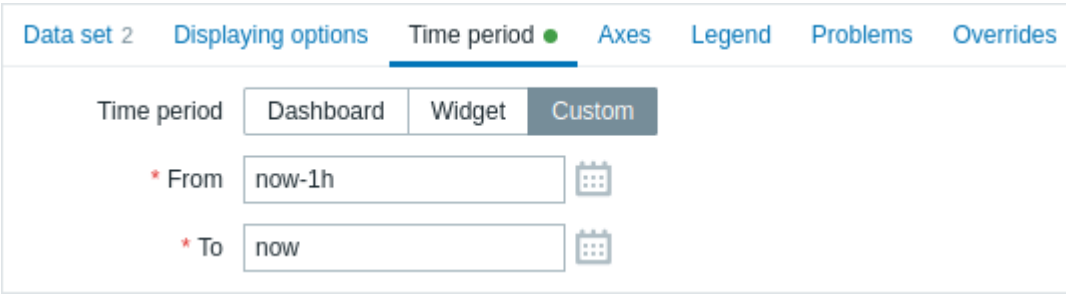
Cilne **Attēlošanas opcijas** ļauj definēt vēsturisko datu atlasi:



Vēsturisko datu atlase	Iestatiet grafika datu avotu: **Automātiski** - dati tiek iegūti atbilstoši klasiskā grafika algoritmam (noklusējums); **Vēsture** - dati no vēstures; **Tendences** - dati no tendencēm.
Vienkāršie trigeri	Atzīmējiet izvēles rūtiņu, lai parādītu vienkāršo trigeru sliekšņvērtības. Sliekšņvērtības tiks attēlotas kā pārtrauktas līnijas, izmantojot trigera prioritātes krāsu. Vienkāršs trigeris ir trigeris ar vienu funkciju (tikai `last`, `max`, `min`, `avg`) vienam vienumam izteiksmē. Var tikt attēloti ne vairāk kā trīs trigeri. 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 nestrādājošais laiks tiek attēlots pelēkā krāsā (izmantojot noklusējuma lietotāja saskarnes tēmu *Original blue*).
Hostu nosaukumi apzīmējumos	Izvēlieties, vai grafika leģendā un rīka padomā, kas tiek parādīts, novietojot kursoru virs grafika datiem, rādīt hostu nosaukumus: **Automātiski** - hostu nosaukumi tiek rādīti tikai tad, ja datu kopās ir vairāk nekā viens hosts (noklusējums). **Rādīt** - hostu nosaukumi tiek rādīti. **Paslēpt** - hostu nosaukumi tiek paslēpti. Šis parametrs nav pieejams, konfigurējot logrīku veidnes informācijas panelī.
Procentiles līnija (kreisā)	Atzīmējiet izvēles rūtiņu un ievadiet procentiles vērtību, lai norādīto procentili grafikā parādītu kā līniju uz kreisās Y ass. Piemēram, ja ir iestatīta 95% procentile, procentiles līnija atradīsies līmenī, zem kura atrodas 95 procenti vērtību.
Procentiles līnija (labā)	Atzīmējiet izvēles rūtiņu un ievadiet procentiles vērtību, lai norādīto procentili grafikā parādītu kā līniju uz labās Y ass. Piemēram, ja ir iestatīta 95% procentile, procentiles līnija atradīsies līmenī, zem kura atrodas 95 procenti vērtību.

Laika periods

Cilne **Laika periods** ļauj iestatīt laika periodu, par kuru grafikā rādīt datus:

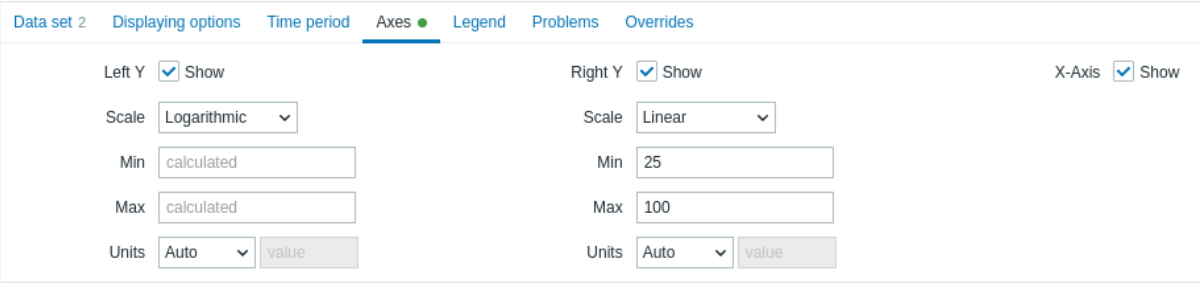


Laika periods	Atlasiet datu avotu laika periodam: **Informācijas panelis** - izmantojiet informācijas paneļa laika perioda atlasītāju; **Logrīks** - izmantojiet saderīgu logrīku (iestatīts parametrā *Logrīks*); **Pielāgots** - izmantojiet pielāgotu laika periodu, kas iestatīts parametros *No* un *Līdz*; ja tas ir iestatīts, logrīka augšējā labajā stūrī tiks parādīta pulksteņa ikona, kas, novietojot uz tās peles rādītāju, norādīs iestatīto laiku. Ņemiet vērā, ka neatkarīgi no logrīka *Laika periods* konfigurācijas saderīgi logrīki to joprojām var 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` 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

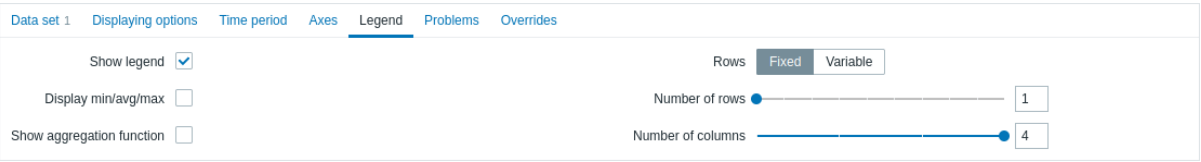
Cilne **Asis** ļauj pielāgot asu attēlošanu:



Kreisā Y	Atzīmējiet šo izvēles rūtiņu, lai padarītu kreiso Y asi redzamu. Izvēles rūtiņa var būt atspējota, ja tā nav atlasīta cilnē *Datu kopa* vai *Ignorējumi*.
Labā Y	Atzīmējiet šo izvēles rūtiņu, lai padarītu labo Y asi redzamu. Izvēles rūtiņa var būt atspējota, ja tā nav atlasīta cilnē *Datu kopa* vai *Ignorējumi*.
X ass	Noņemiet atzīmi no šīs izvēles rūtiņas, lai paslēptu X asi (pēc noklusējuma tā ir atzīmēta).
Mērogs	Nolaižamajā sarakstā izvēlieties diagrammas asu vērtību mērogu: **Lineārs** - asu vērtības palielinās par fiksētu lielumu (piemēram, 10, 20, 30), un tas ir piemērots datiem, kas mainās vienmērīgi vai aptver nelielu līdz vidēju diapazonu; **Logaritmisks** - asu vērtības palielinās eksponenciāli (piemēram, 10, 100, 1000), un tas ir piemērots datiem, kas strauji mainās vai aptver lielu diapazonu.
Min.	Iestatiet atbilstošās ass minimālo vērtību. Tiek norādīta Y ass redzamā diapazona minimālā vērtība.
Maks.	Iestatiet atbilstošās ass maksimālo vērtību. Tiek norādīta Y ass redzamā diapazona maksimālā vērtība.
Mērvienības	Nolaižamajā sarakstā izvēlieties diagrammas asu vērtību mērvienību: **Automātiski** - asu vērtības tiek attēlotas, izmantojot datu kopas pirmā vienuma mērvienību; **Statiski** - asu vērtības tiek attēlotas, izmantojot laukā *value* norādīto mērvienību; ja lauks ir atstāts tukšs, tiek attēlotas tikai skaitliskās vērtības.

Leģenda

Cilne **Leģenda** ļauj pielāgot grafika leģendu:

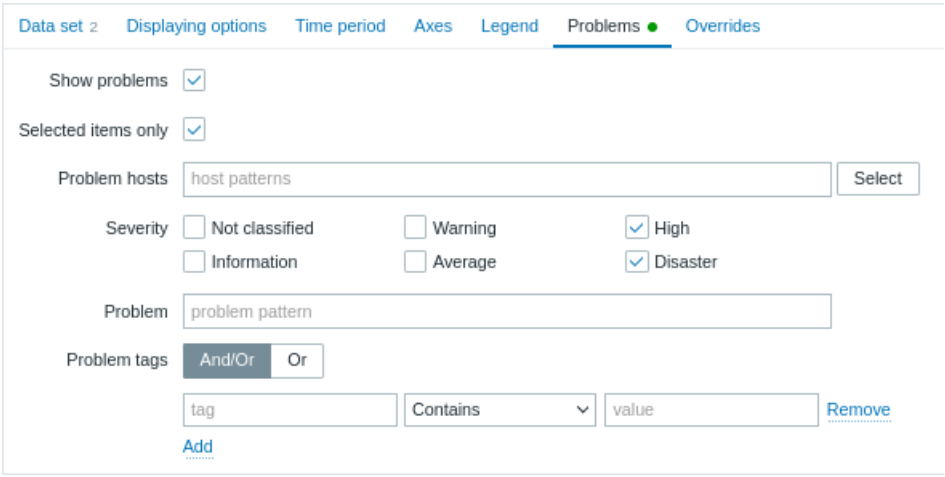


Rādīt leģendu	Noņemiet atzīmi no šīs izvēles rūtiņas, lai grafikā paslēptu leģendu (pēc noklusējuma tā ir atzīmēta).
Rādīt mini-mumu/vidējo/maksimumu	Atzīmējiet šo izvēles rūtiņu, lai leģendā rādītu vienuma minimālās, vidējās un maksimālās vērtības.
Rādīt agregācijas funkciju	Atzīmējiet šo izvēles rūtiņu, lai leģendā rādītu agregācijas funkciju.
Rindas	Atlasiet leģendas rindu attēlošanas režīmu: **Fiksēts** - attēlojamo rindu skaitu nosaka parametra *Rindu skaits* vērtība; **Mainīgs** - attēlojamo rindu skaitu nosaka konfigurēto vienumu skaits, nepārsniedzot parametra *Maksimālais rindu skaits* vērtību.

Rindu skaits/ Maksimālais rindu skaits	Ja laukam *Rindas* ir iestatīta vērtība "Fiksēts", iestatiet attēlojamo leģendas rindu skaitu (1-10). Ja laukam *Rindas* ir iestatīta vērtība "Mainīgs", iestatiet maksimālo attēlojamo leģendas rindu skaitu (1-10).
Kolonnu skaits	Iestatiet attēlojamo leģendas kolonnu skaitu (1-4). Šis parametrs ir pieejams, ja izvēles rūtiņa *Rādīt minimumu/vidējo/maksimumu* nav atzīmēta.

Problēmas

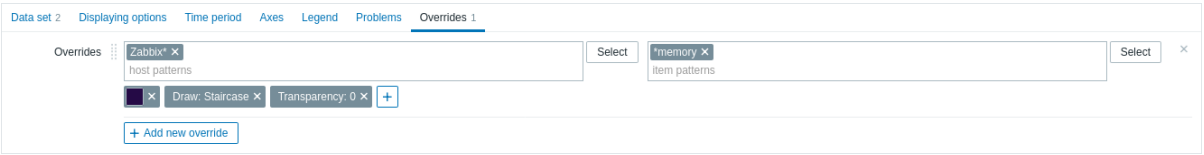
Cilne **Problēmas** ļauj pielāgot problēmu attēlojumu:



Rādīt problēmas	Atzīmējiet šo izvēles rūtiņu, lai iespējotu problēmu attēlošanu grafikā (pēc noklusējuma tā nav atzīmēta, t. i., funkcija ir atspējota).
Tikai atlasītie vienumi	Atzīmējiet šo izvēles rūtiņu, lai grafikā tiktu attēlotas tikai atlasīto vienumu problēmas.
Problēmu hosti	Atlasiet problēmu hostus, kas jāattēlo grafikā. Var izmantot aizstājējzīmju šablonus (piemēram, ∗ atgriezīs rezultātus, kas atbilst nullei vai vairāk rakstzīmēm). Lai norādītu aizstājējzīmes šablonu, manuāli ievadiet virkni un nospiediet *Enter*. Ievadīšanas laikā nolaižamajā sarakstā tiek parādīti visi atbilstošie hosti. Šis parametrs nav pieejams, konfigurējot logrīku veidnes informācijas panelī.
Nopietnība	Atzīmējiet problēmu nopietnības līmeņus, lai filtrētu grafikā attēlojamās problēmas. Ja nav atzīmēts neviens nopietnības līmenis, tiks attēlotas visas problēmas.
Problēma	Norādiet problēmas nosaukumu, kas jāattēlo grafikā.
Problēmu tagi	Norādiet problēmu tagus, lai ierobežotu logrīkā attēlojamo problēmu skaitu. Var iekļaut, kā arī izslēgt konkrētus tagus un tagu vērtības. Var iestatīt vairākus nosacījumus. Taga nosaukuma atbilstība vienmēr ir reģistrjutīga. Katram nosacījumam ir pieejami vairāki operatori: **Pastāv** - iekļaut norādītos tagu nosaukumus; **Ir vienāds ar** - iekļaut norādītos tagu nosaukumus un vērtības (reģistrjutīgi); **Satur** - iekļaut norādītos tagu nosaukumus, kuru tagu vērtības satur ievadīto virkni (apakšvirknes atbilstība, nav reģistrjutīga); **Nepastāv** - izslēgt norādītos tagu nosaukumus; **Nav vienāds ar** - izslēgt norādītos tagu nosaukumus un vērtības (reģistrjutīgi); **Nesatur** - izslēgt norādītos tagu nosaukumus, kuru tagu vērtības satur ievadīto virkni (apakšvirknes atbilstība, nav reģistrjutīga). Nosacījumiem ir pieejami divi aprēķina veidi: **Un/Vai** - jāizpilda visi nosacījumi; nosacījumi ar vienādu taga nosaukumu tiks grupēti, izmantojot nosacījumu *Vai*; **Vai** - pietiek ar viena nosacījuma izpildi.

Pārraksti

Cilne **Pārraksti** ļauj pievienot pielāgotus datu kopu pārrakstus:



Pārraksti ir noderīgi, ja datu kopai, izmantojot aizstājējzīmi ∗, ir atlasīti vairāki vienumi un vēlaties mainīt to noklusējuma attēlošanas veidu (piemēram, noklusējuma pamatkrāsu vai kādu citu rekvizītu).

Esošie pārraksti (ja tādi ir) tiek parādīti sarakstā. Lai pievienotu jaunu pārrakstu:

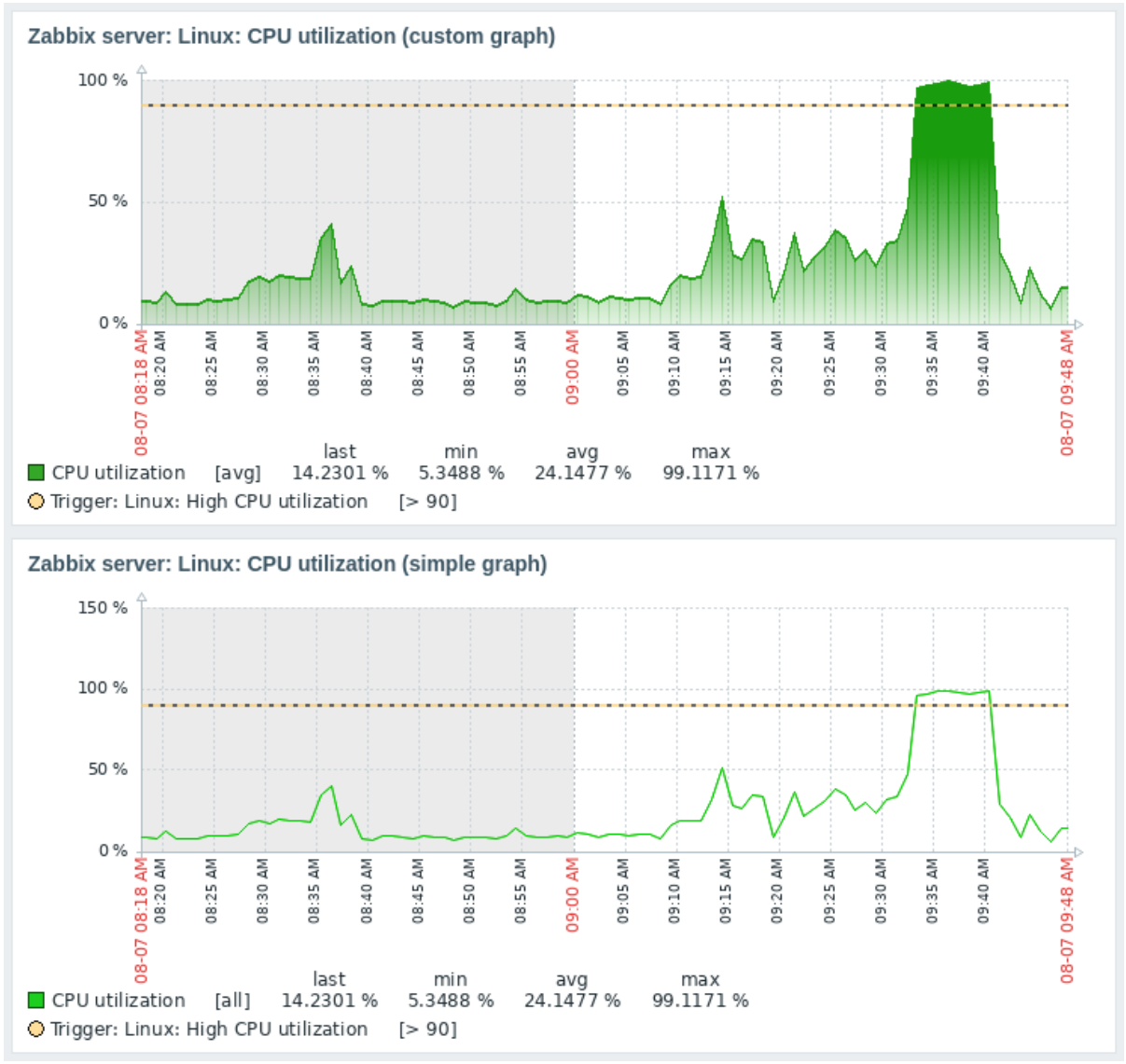
- Noklikšķiniet uz pogas 

- Atlasiet pārrakstam paredzētos hostus un vienumus. Varat arī 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 ņemiet vērā, ka nolaižamajā sarakstā tiek parādīti visi atbilstošie hosti. Aizstājējzīmes simbols vienmēr tiek interpretēts, tāpēc nav iespējams atsevišķi pievienot, piemēram, vienumu ar nosaukumu ”item*”, ja ir arī citi atbilstoši vienumi (piemēram, item2, item3). Hosta šablons un vienuma šablons ir obligāti parametri. Parametrs hostu šablonu norādīšanai nav pieejams, konfigurējot logrīku veidnes informācijas panelī. Parametrs vienumu saraksta norādīšanai ļauj atlasīt tikai veidnē konfigurētos vienumus, konfigurējot logrīku veidnes informācijas panelī.
- Noklikšķiniet uz  , lai atlasītu pārraksta parametrus. Jāatlasa vismaz viens pārraksta parametrs. Parametru aprakstus skatiet iepriekš cilnē *Datu kopa*.

Grafiks (klasiskais)

Pārskats

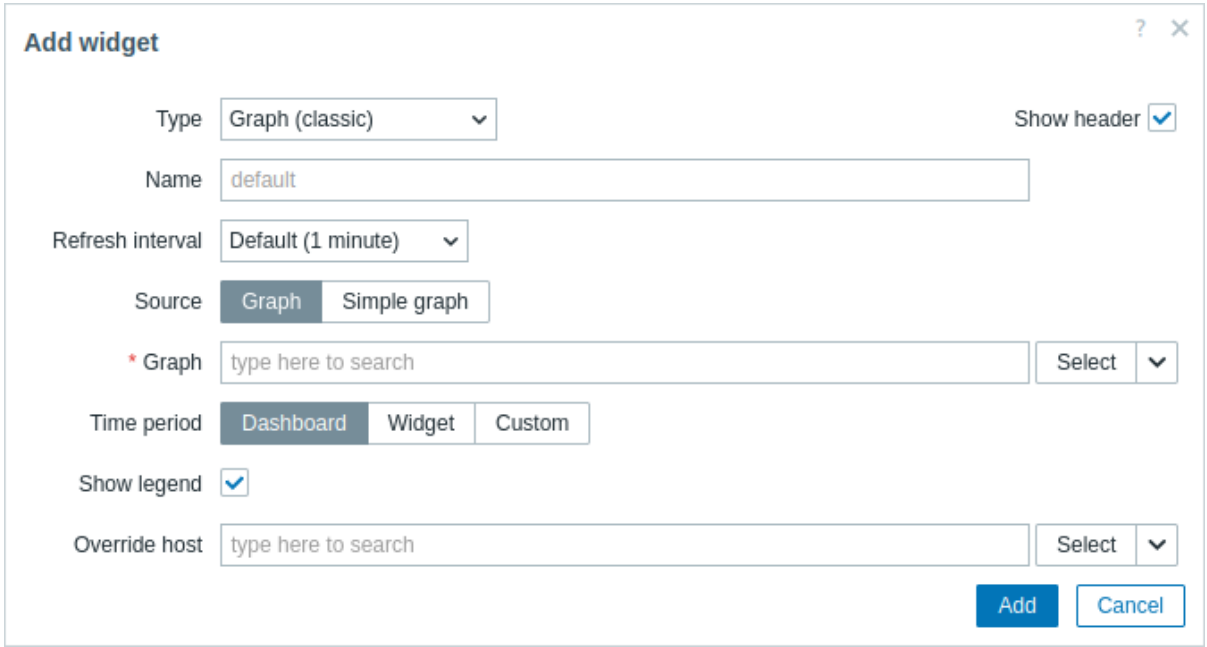
Graph (classic) logrīks attēlo vienuma skaitliskos datus kā attēlu, izmantojot pielāgotu grafiku vai vienkāršu grafiku. Tas atbalsta laika perioda atlasi, trigeru līnijas un neobligātu leģendas rādīšanu.



Informāciju, kas tiek rādīta *Graph (classic)* logrīkā, var lejupielādēt kā PNG attēlu, logrīka izvēlnē *Actions* atlasot opciju *Download image* …, kas tiek parādīta, novietojot kursoru virs logrīka. Atkarībā no pārlūkprogrammas iestatījumiem attēls tiek automātiski saglabāts mapē `Downloads` vai tiek atvērts dialoglodziņš *Save*, kurā varat izvēlēties saglabāšanas vietu.

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)

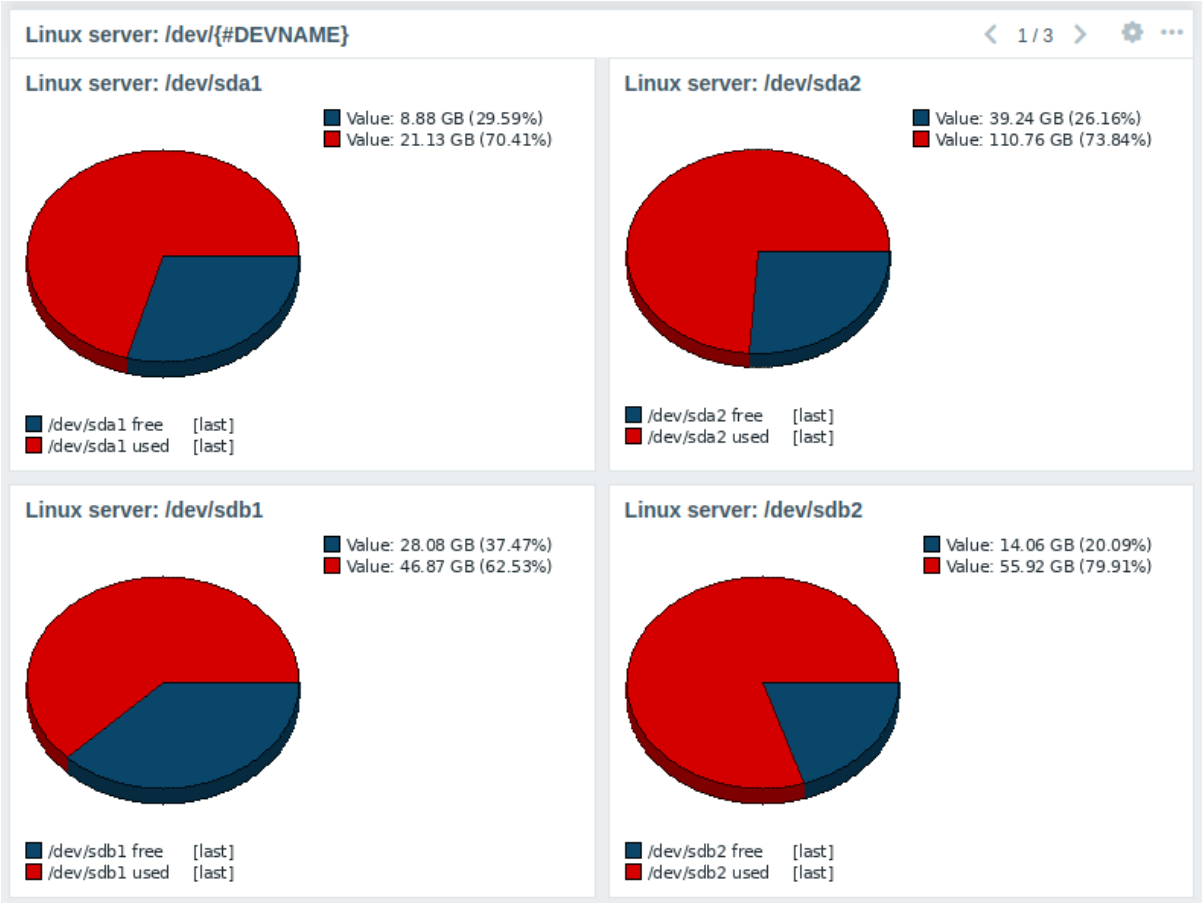
Ja grafika augstums logrīkā ir nepietiekams, leģenda var netikt attēlota vai tikt attēlota tikai daļēji. Vispirms tiek paslēpti trigeri un procentiles, pēc tam vienumu leģenda. Lai attēlotu pilnu leģendu, palieliniet logrīka vertikālo izmēru.

Note:
Var attēlot ne vairāk kā 3 trigeru līnijas. Ja trigeru ir vairāk, attēlošanai prioritāri tiek izvēlēti trigeri ar zemāku svarīgumu.

Grafika prototips

Pārskats

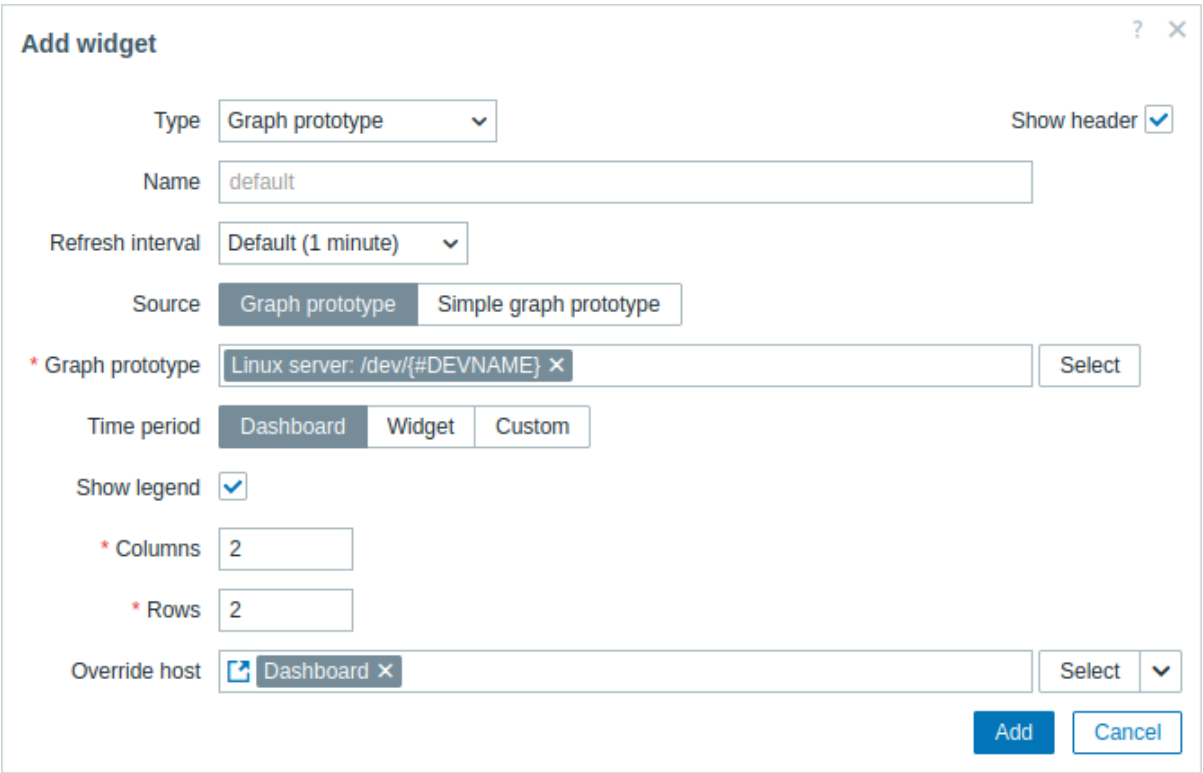
Grafika prototipa logrīks attēlo režģi ar automātiski izveidotiem pielāgotiem grafikiem, kas balstīti uz zema līmeņa atklāšanas kārtulas grafiku prototipiem vai vienumu prototipiem.



Grafiki ir sakārtoti konfigurējamā režģa izkārtojumā; ja grafiku skaits pārsniedz izkārtojuma ietilpību, logrīkā tiek iespējota lapošana.

Konfigurēšana

Lai konfigurētu, kā logrīka tipu atlasiet *Graph prototype*:



Papildus parametriem, kas ir kopīgi visiem logrīkiem, varat iestatīt tālāk norādītās specifiskās 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 grafikos rādīt datus. Atlasiet datu avotu laika periodam: **Dashboard** - izmantojiet informācijas paneļa laika perioda atlasītāju; **Widget** - izmantojiet saderīgu logrīku (iestatīts parametrā *Widget*); **Custom** - izmantojiet pielāgotu laika periodu, kas iestatīts parametros *From* un *To*; ja tas ir iestatīts, logrīka augšējā labajā stūrī tiks parādīta pulksteņa ikona, kas, novietojot uz tās peles rādītāju, 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` u. c.). 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` u. c.). 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 tā ir atzīmēta).
Override host	Atlasiet saderīgu logrīku vai informācijas paneļa hostu atlasītāju kā datu avotu hostiem. Šis parametrs nav pieejams, konfigurējot logrīku veidnes informācijas panelī.
Columns	Ievadiet grafiku kolonnu skaitu, kas jāparāda grafika prototipa logrīkā.
Rows	Ievadiet grafiku rindu skaitu, kas jāparāda grafika prototipa logrīkā.

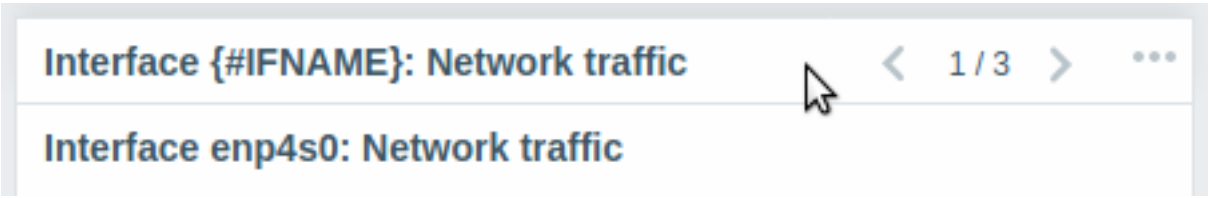
Grafika leģenda sastāv no trim sadaļām:

- Vienumi un to apkopotās vērtības
- Procentiles (ja konfigurētas)
- Trigeri (ja kādi ir saistīti ar parādītajiem vienumiem)

Ja grafika augstums logrīkā ir nepietiekams, leģenda var netikt parādīta vai var tikt parādīta tikai daļēji. Vispirms tiek paslēpti trigeri un procentiles, pēc tam - vienumu leģenda. Lai parādītu pilnu leģendu, palieliniet logrīka vertikālo izmēru.

Note:
Var tikt parādītas ne vairāk kā 3 trigeru līnijas. Ja trigeru ir vairāk, attēlošanai vispirms tiek atlasīti trigeri ar zemāku nopietnības pakāpi.

Lai gan parametri *Columns* un *Rows* ļauj logrīkā ietilpināt vairāk nekā vienu grafiku, atklāto grafiku joprojām var būt vairāk nekā logrīkā ir kolonnu/rindu. Šādā gadījumā logrīkā kļūst pieejama lapošana, un slīdošā galvene ļauj pārslēgties starp lapām, izmantojot kreiso un labo bultiņu:

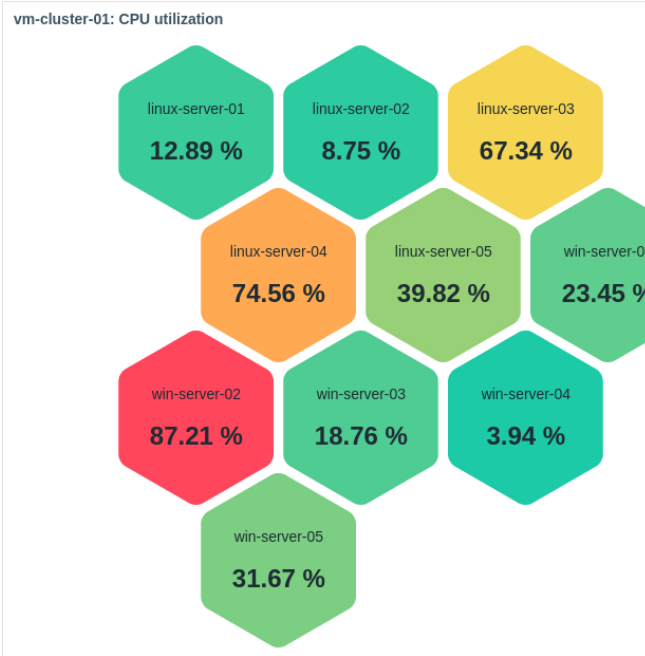


Honeycomb

Pārskats

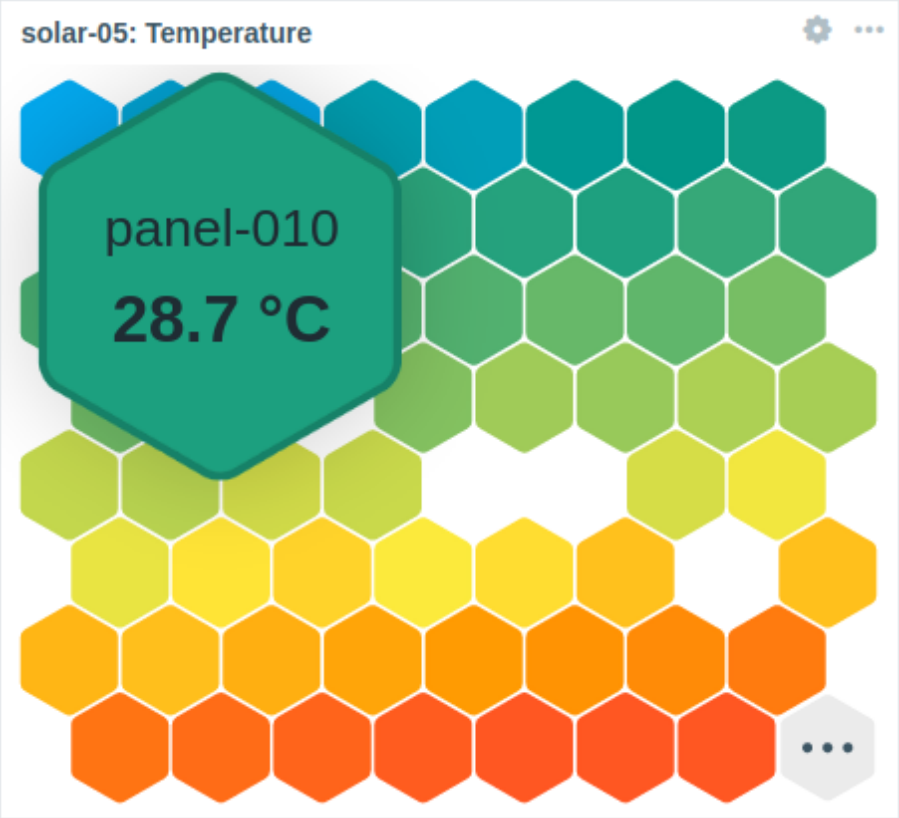
Honeycomb logrīks režģī attēlo hostus vai vienumus kā sešstūra šūnas. Šāds izkārtojums ļauj viegli noteikt problemātiskos hostus/vienumus, pamanīt problēmu kopas, ātri salīdzināt grupas un veikt citas darbības.





Attēloto šūnu skaitu ierobežo logrīka izmērs un minimālais šūnas izmērs (32px). Ja visas šūnas neietilpst, pēdējā šūna attēlo daudzpunkti. Logrīka izmēru var mainīt, un šūnu skaits, izmērs un novietojums tiek dinamiski pielāgots.

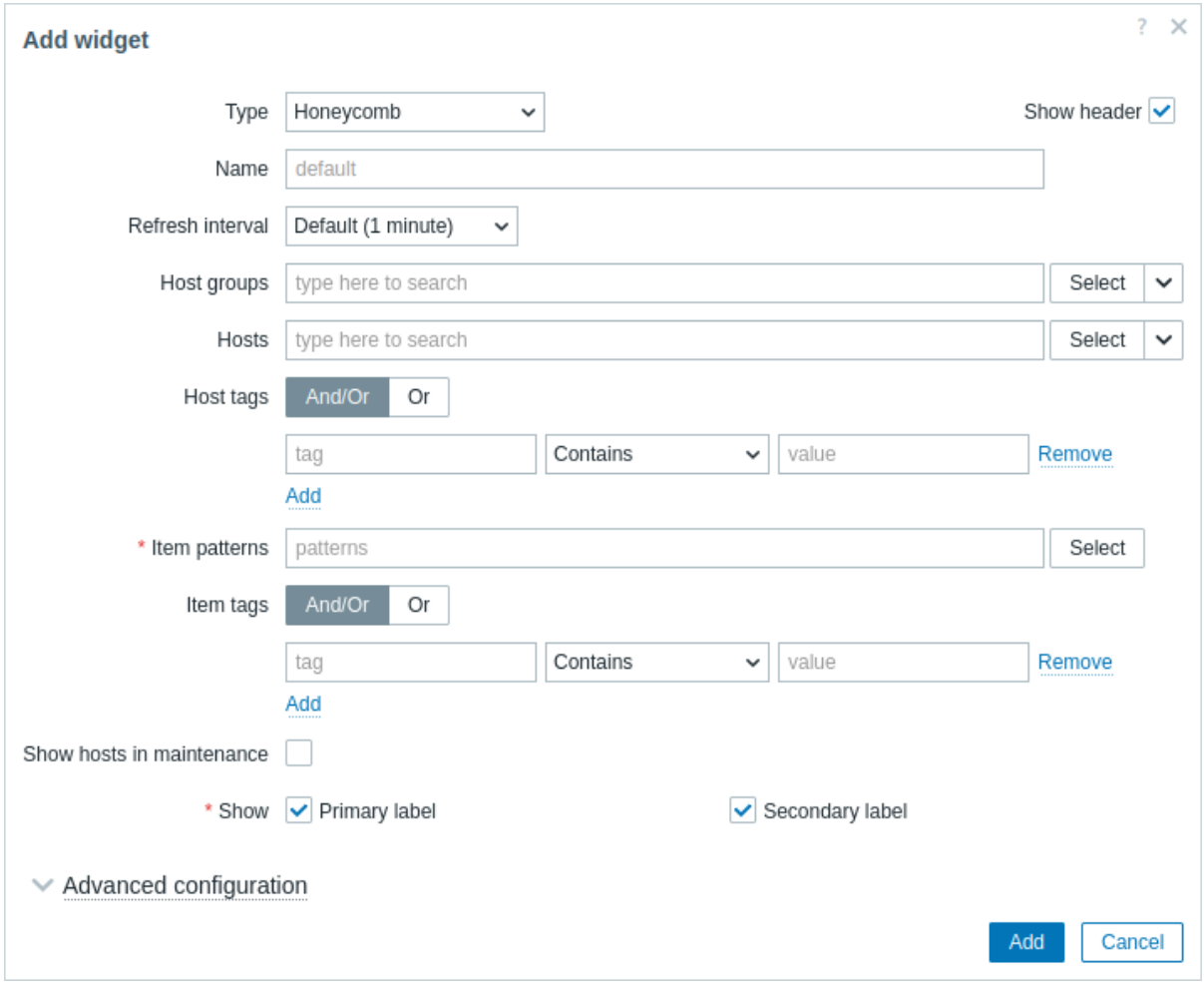
Novietojot peles rādītāju virs šūnas, fokusētā šūna tiek palielināta, lai uzlabotu redzamību. Noklikšķinot uz šūnas, tās apmale tiek izcelta, līdz tiek atlasīta cita šūna.



Honeycomb logrīkā attēloto informāciju var lejupielādēt kā PNG attēlu, logrīka izvēlnē *Actions* ..., kas tiek parādīta, novietojot peles rādītāju virs logrīka, atlasot opciju *Download image*. Atkarībā no pārlūkprogrammas iestatījumiem attēls tiek automātiski saglabāts mapē `Downloads` vai tiek atvērts dialoglodziņš *Save*, kurā var izvēlēties saglabāšanas vietu.

Konfigurācija

Lai konfigurētu, atlasiet *Honeycomb* kā tipu:



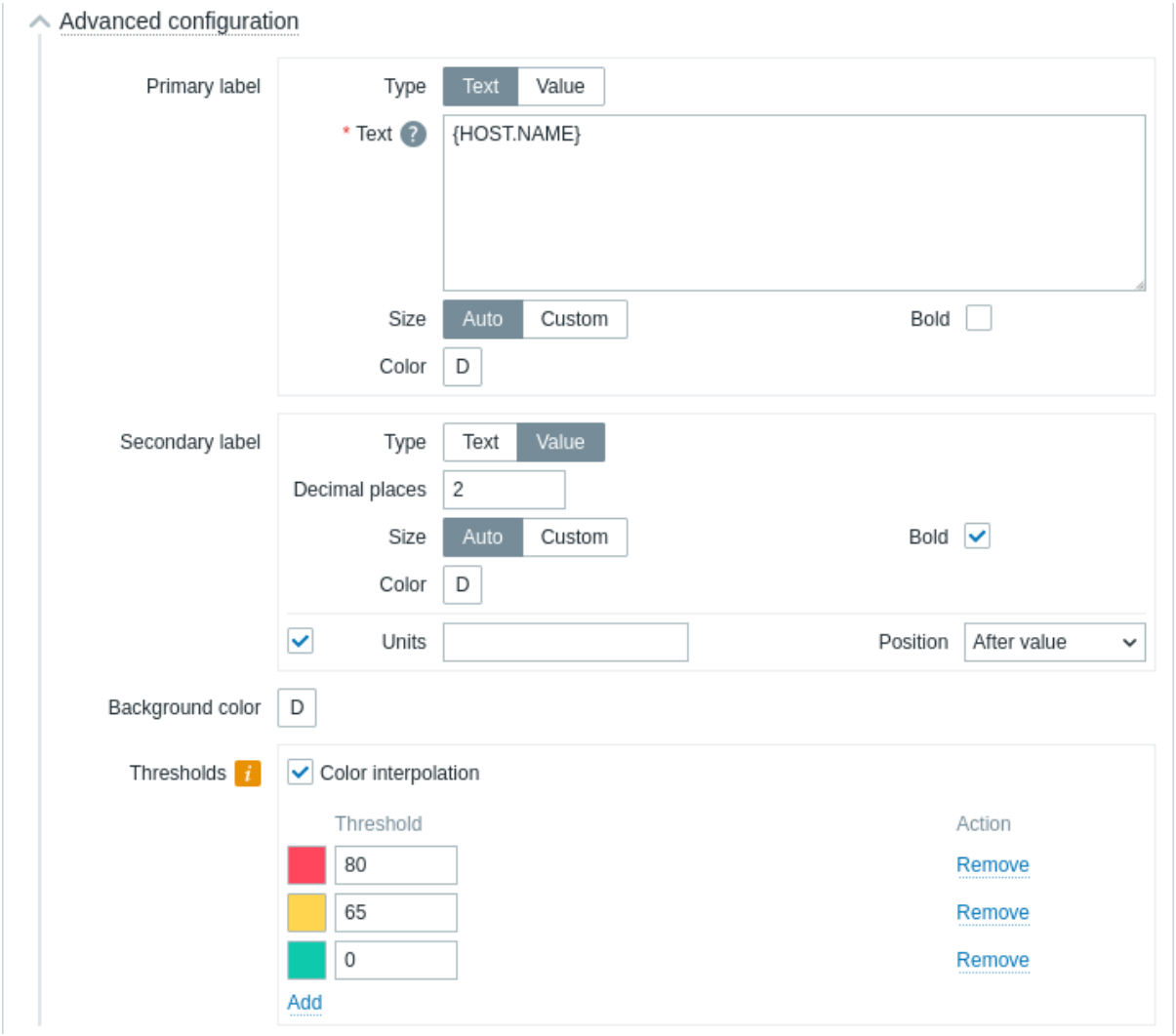
Papildus parametriem, kas ir kopīgi visiem logrīkiem, varat iestatīt šādas specifiskas opcijas:

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 parādīs 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 parādīs 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 parādīto 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: **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. Šis parametrs nav pieejams, konfigurējot logrīku veidnes panelī.
Vienumu šabloni	Ievadiet vienumu šablonus vai atlasiet esošos vienumus kā vienumu šablonus. Honeycomb tiks parādīti 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 sākat 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 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 ierobežotu logrīkā parādīto vienumu skaitu. Papildinformāciju skatiet iepriekš sadaļā *Hostu birkas*.
Rādīt hostus uzturēšanā	Atzīmējiet šo izvēles rūtiņu, lai parādītu hostus uzturēšanā (šajā gadījumā blakus hosta nosaukumam tiks parādīta uzturēšanas ikona). Šis parametrs ir apzīmēts kā *Rādīt datus uzturēšanā*, konfigurējot logrīku veidnes panelī.
Rādīt	Atzīmējiet šo izvēles rūtiņu, lai parādītu attiecīgo honeycomb šūnas elementu — primāro etiķeti, sekundāro etiķeti. Jābūt atlasītam vismaz vienam elementam.
Paplašinātā konfigurācija	Noklikšķiniet uz virsraksta *Paplašinātā konfigurācija*, lai parādītu paplašinātās konfigurācijas opcijas.

Paplašinātā konfigurācija

Paplašinātās konfigurācijas opcijas ir pieejamas salokāmajā sadaļā *Paplašinātā konfigurācija* un tikai tiem elementiem, kas atlasīti laukā *Rādīt* (skatīt iepriekš), kā arī šūnveida šūnu fona krāsai vai sliekšņiem.



Primārā/sekundārā etiķete

Tips — Atlasiet etiķetes tipu:
Teksts — etiķetē tiks parādīts teksts, kas norādīts parametram *Teksts*;
Vērtība — etiķetē tiks parādīta vienuma vērtība ar decimālzīmju skaitu, kas norādīts parametram *Decimālzīmes*.

Teksts — Ievadiet etiķetes tekstu. Šis teksts var aizstāt noklusējuma vienuma nosaukumu.
Tiek atbalstīts vairākrindu teksts. Ir iespējama teksta un atbalstīto makrosu kombinācija.
Tiek atbalstīti `{HOST.\*}`, `{ITEM.\*}`, `{INVENTORY.\*}` un lietotāja makrosi.
Šūnveida šūnas tiek kārtotas alfabētiskā secībā pēc hosta nosaukuma un katra hosta ietvaros — pēc vienuma nosaukuma.
Šis parametrs ir pieejams, ja *Tips* ir iestatīts uz *Teksts*.

Decimālzīmes — Ievadiet decimālzīmju skaitu, kas jāparāda kopā ar vērtību.
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 šūnas izmēru).
Ņemiet vērā, ka etiķetes, kas neietilpst šūnveida šūnas izmērā, tiek saīsinātas.

Treknraksts — Atzīmējiet izvēles rūtiņu, lai etiķetes tiktu parādītas treknrakstā.

Krāsa — Atlasiet etiķetes krāsu krāsu atlasītājā.
"D" apzīmē noklusējuma krāsu, kas ir atkarīga no lietotāja saskarnes tēmas. Lai atgrieztos pie noklusējuma krāsas, krāsu atlasītājā noklikšķiniet uz pogas *Izmantot noklusējumu*.

Mērvienības

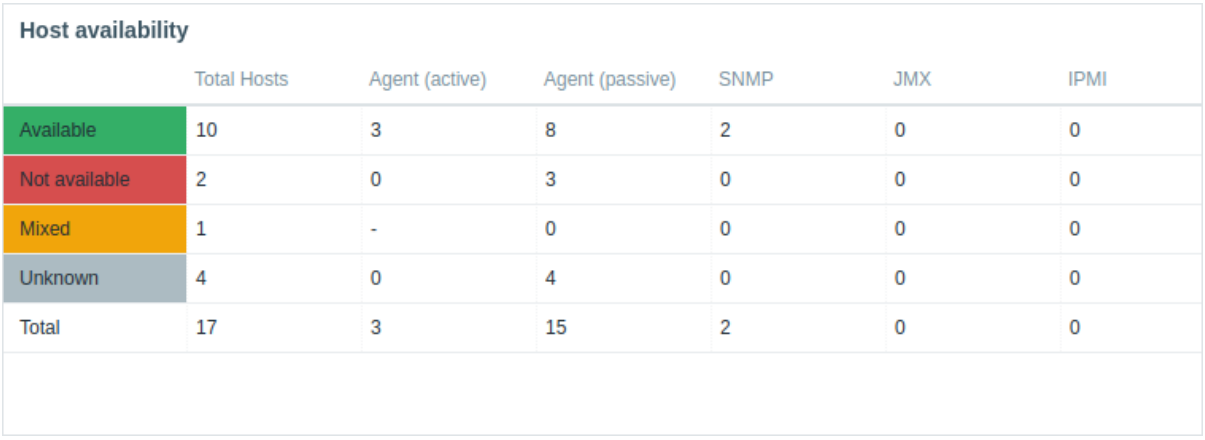
Mērvienības — Atzīmējiet izvēles rūtiņu, lai kopā ar vienuma vērtību parādītu mērvienības.
Ja ievadīsiet mērvienības nosaukumu, tas aizstās mērvienības, kas iestatītas vienuma konfigurācijā.
Šis parametrs ir pieejams, ja *Tips* ir iestatīts uz *Vērtība*.

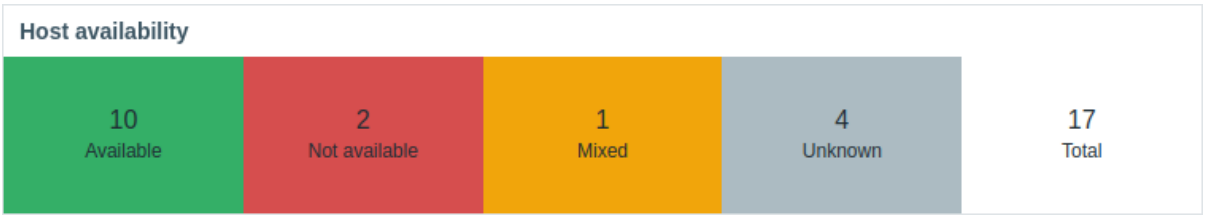
Pozīcija	Atlasiet vienuma mērvienību novietojumu (pirms vai pēc vienuma vērtības). Šis parametrs tiek ignorēts šādām ar laiku saistītām mērvienībām: unixtime, uptime, s. Šis parametrs ir pieejams, ja *Tips* ir iestatīts uz *Vērtība* un ir atlasīta opcija *Mērvienības*.
Fona krāsa	
Fona krāsa	Atlasiet šūnveida šūnu fona krāsu krāsu atlasītājā. "D" apzīmē noklusējuma krāsu, kas ir atkarīga no lietotāja saskarnes tēmas. Lai atgrieztos pie noklusējuma krāsas, krāsu atlasītājā noklikšķiniet uz pogas *Izmantot noklusējumu*.
Sliekšņi	
Krāsu interpolācija	Atzīmējiet izvēles rūtiņu, lai iespējotu vienmērīgu pāreju starp šūnveida šūnu sliekšņu krāsām. 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.

Hosta pieejamība

Pārskats

Logrīks *Host availability* attēlo pieejamību atlasītajās hostu grupās, ļaujot jums uzraudzīt, cik hostu ir pieejami vai nav pieejami.



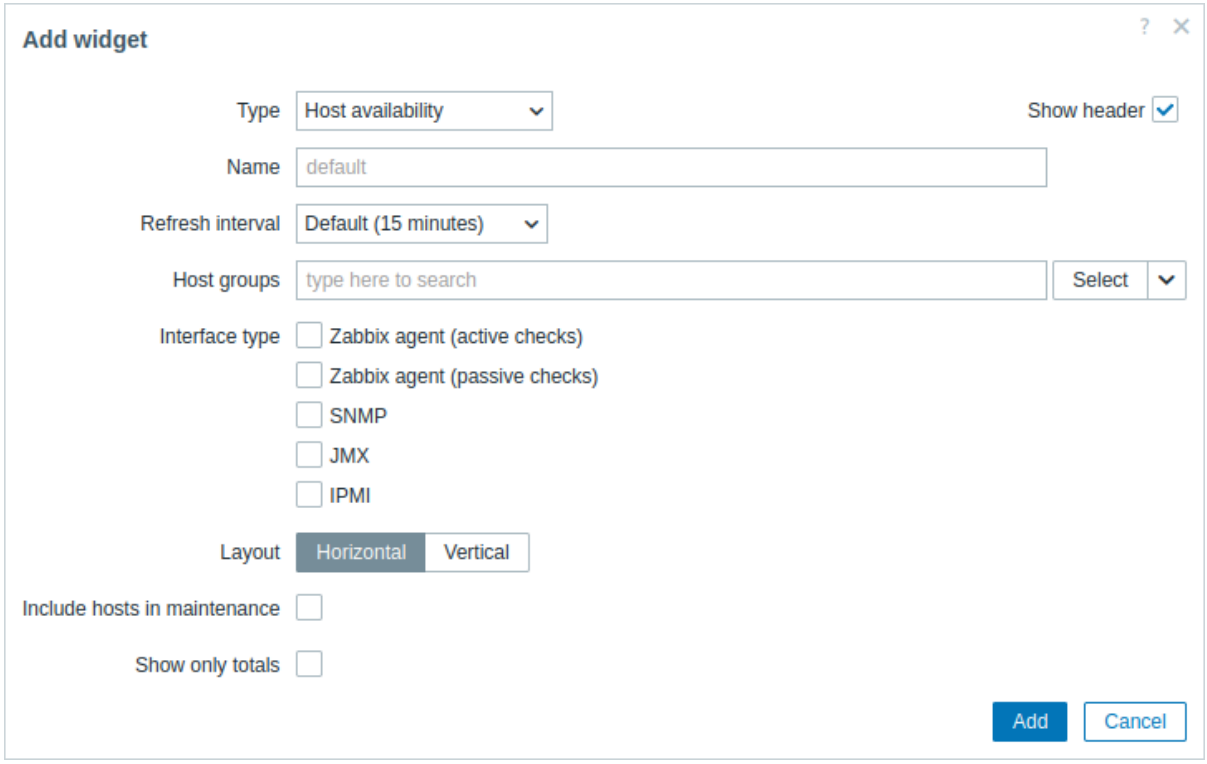


Hostu pieejamība tiek skaitīta šādi:

- *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:



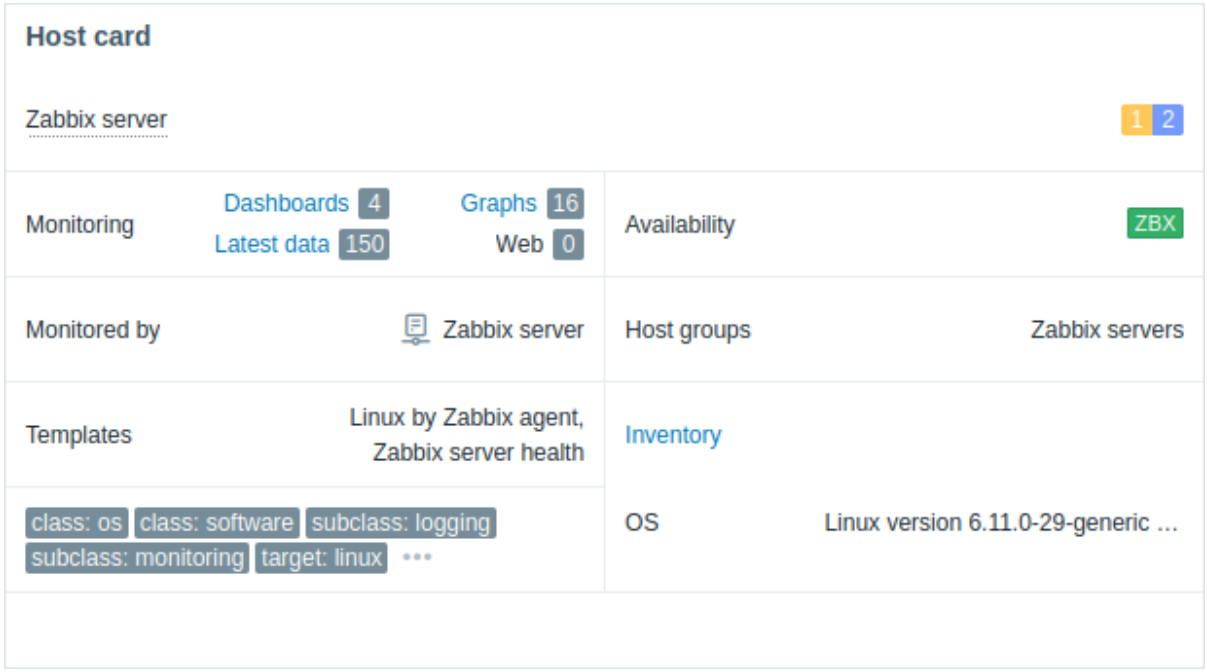
Papildus parametriem, kas ir kopīgi visiem logrīkiem, varat iestatīt šādas specifiskas opcijas:

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.

Host karte

Pārskats

Logrīks *Host card* attēlo informāciju par vienu hostu, ļaujot ātri novērtēt hosta statusu un konfigurāciju (pieejamību, problēmas, inventāru u.c.).

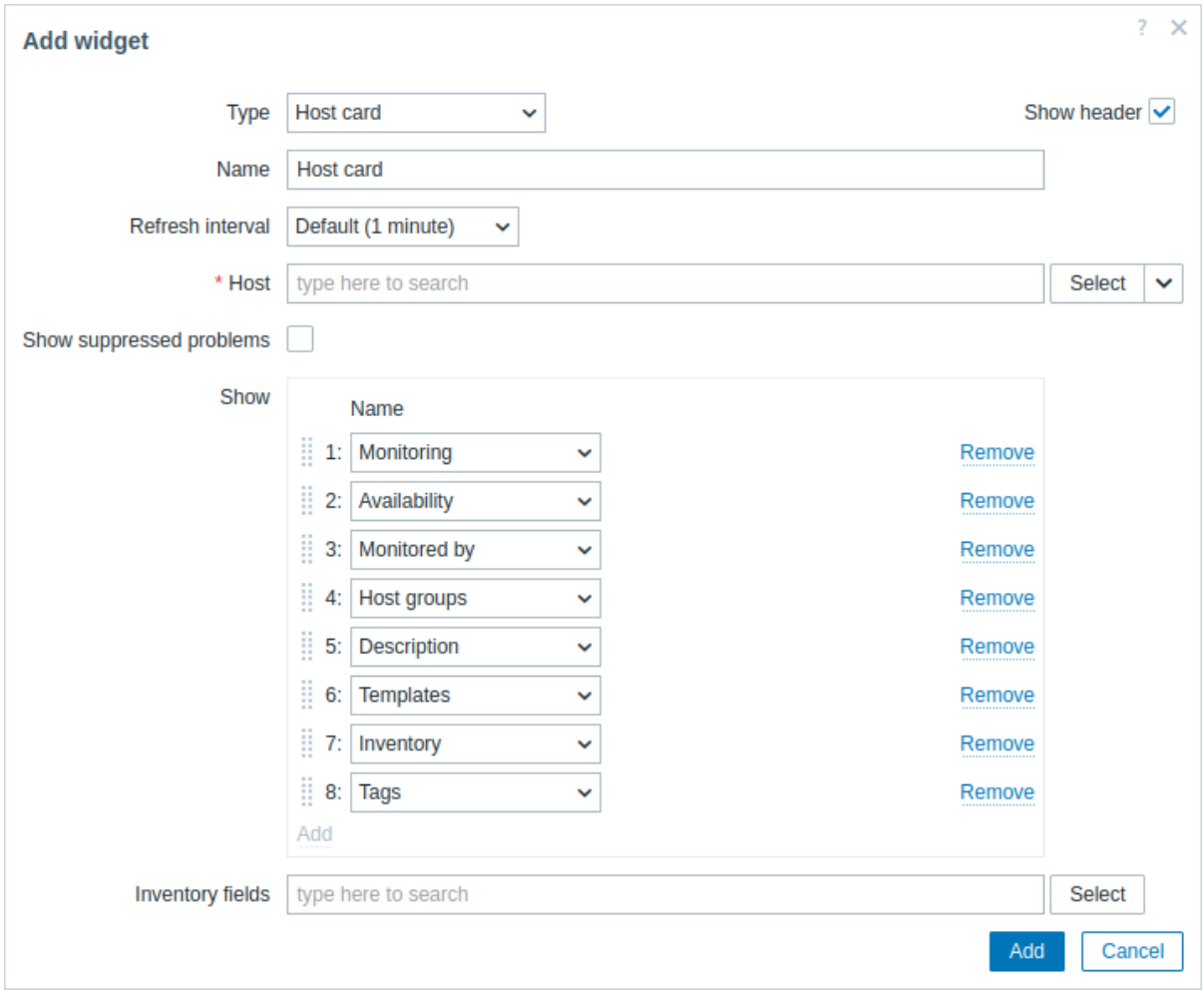


Hostu var iestatīt tieši logrīka konfigurācijā vai atlasīt no saderīga logrīka vai hosta atlasītāja informācijas panelī.

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:



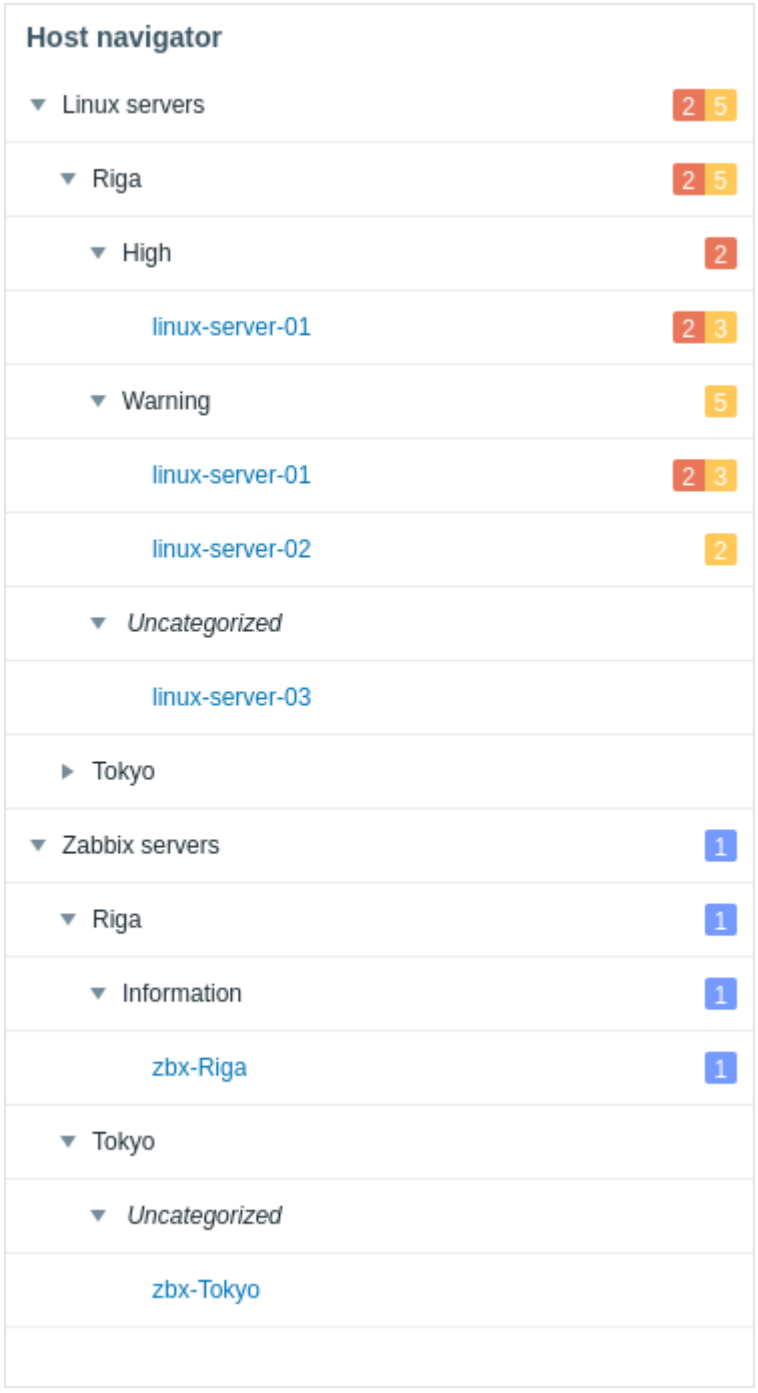
Papildus parametriem, kas ir kopīgi visiem logrīkiem, varat iestatīt šādus specifiskos parametrus:

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*.

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.

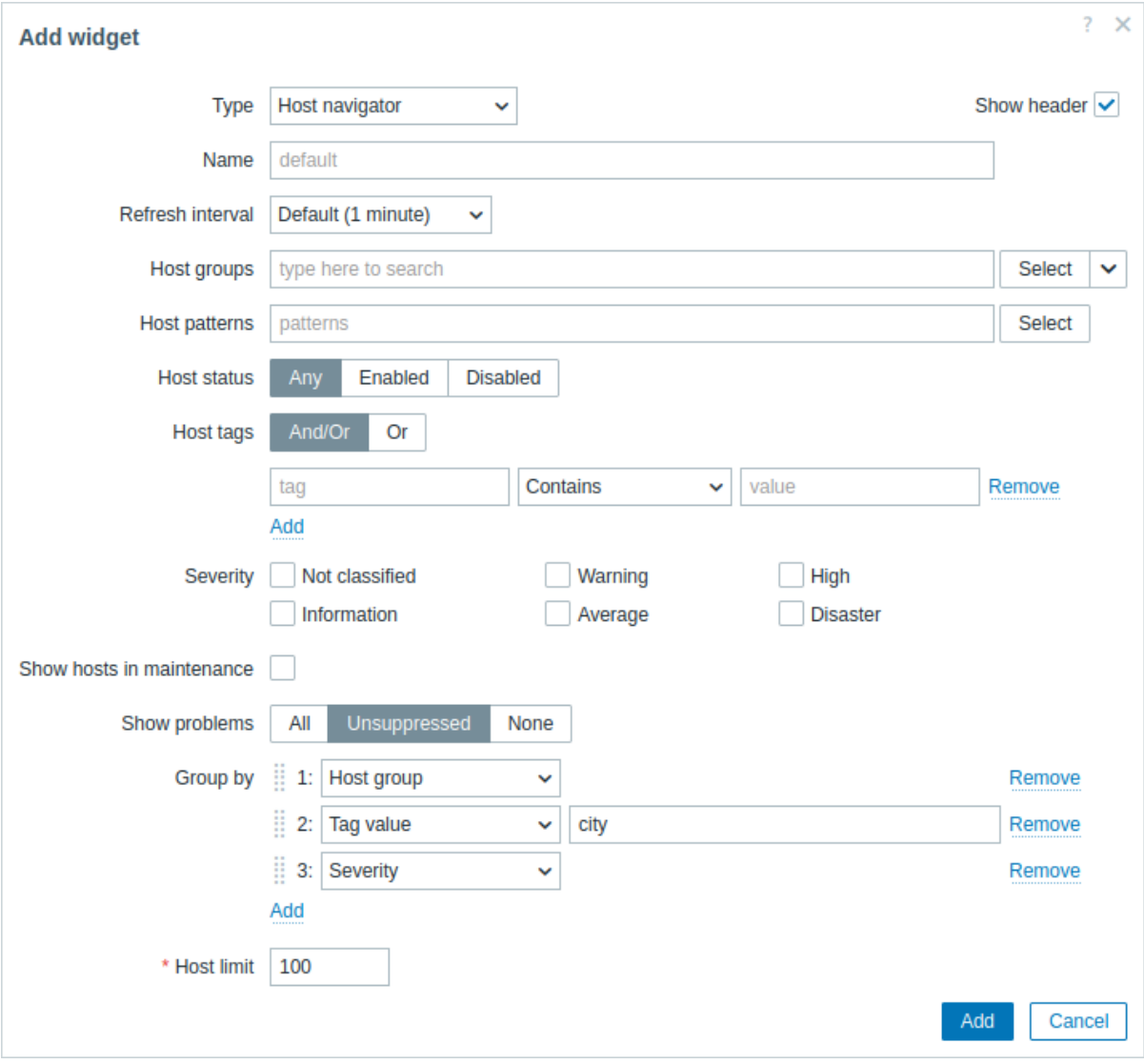
Grupām, problēmām un uzturēšanā esošajiem hostiem papildu informācija ir pieejama, novietojot peles kursoru virs attiecīgā elementa.

Logrīks ir visnoderīgākais, lai kontrolētu, ko citi logrīki attēlo, pamatojoties uz atlasīto hostu.



Konfigurācija

Lai konfigurētu, kā tipu atlasiet *Host navigator*:



Papildus parametriem, kas ir kopīgi visiem logrīkiem, varat iestatīt tālāk norādītās specifiskās opcijas:

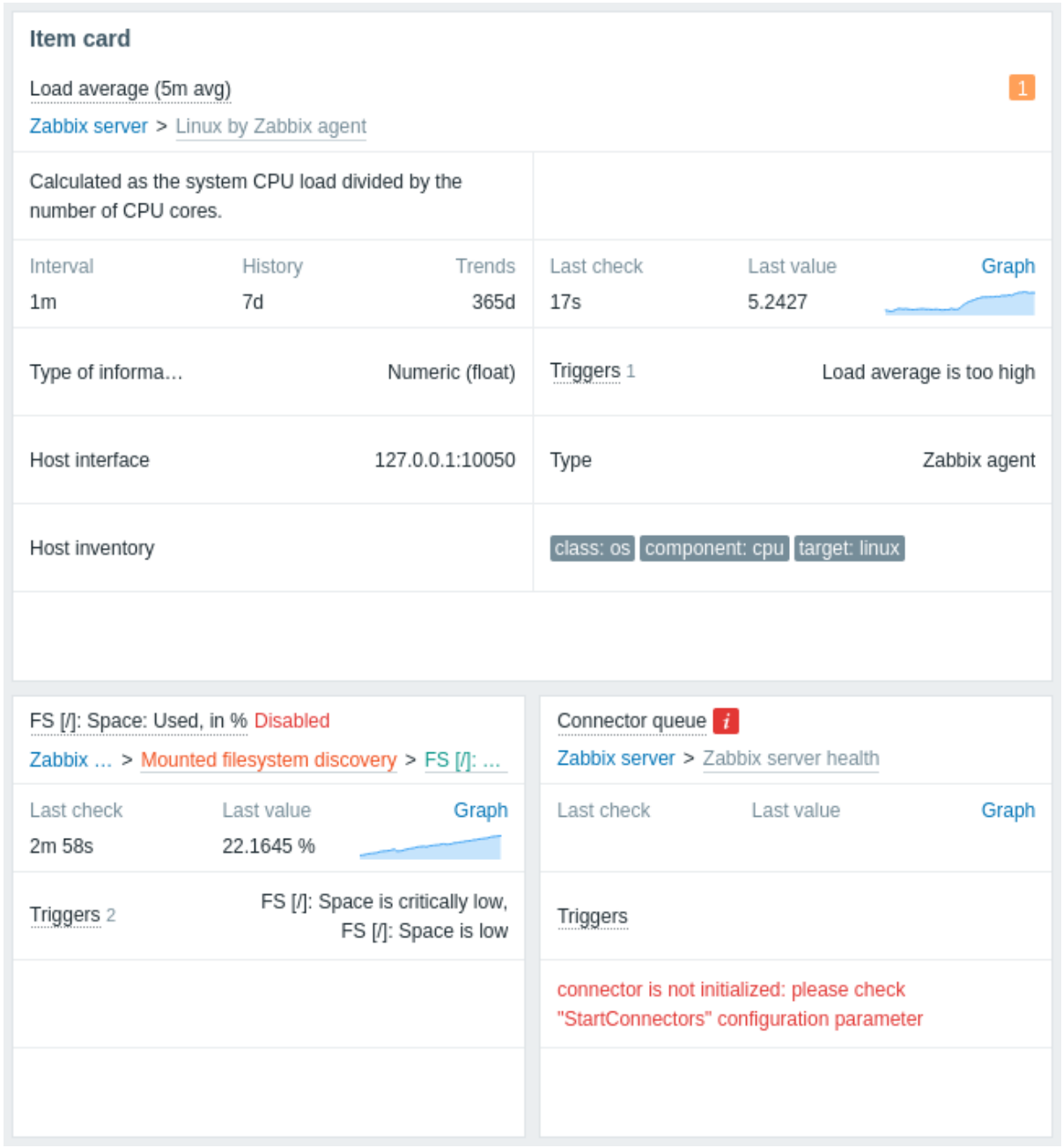
Hosts grupas	Atlasiet hosts grupas. Varat arī atlasīt saderīgu logrīku kā hosts grupu datu avotu. Šis lauks nodrošina automātisko pabeigšanu, tāpēc, sākot rakstīt grupas nosaukumu, tiks parādīts atbilstošo grupu nolaižamais saraksts. Atlasot vecākgrupu, tiek netieši atlasītas visas ligzdotās hosts grupas; ja nav atlasīta neviena hosts grupa, logrīks parādīs visus hosts no visām hosts grupām. Šis parametrs nav pieejams, konfigurējot logrīku veidnes informācijas panelī.
Hosts raksti	Ievadiet hosts rakstus vai atlasiet esošos hosts kā hosts rakstus. Hosts, kas atbilst norādītajiem rakstiem, tiks parādīti hosts navigatorā. Šis lauks nodrošina automātisko pabeigšanu, tāpēc, sākot rakstīt hosts nosaukumu, tiks parādīts atbilstošo hosts nolaižamais saraksts. Ja nav atlasīts neviens hosts, logrīks parādīs visus hosts. Atlasīšanai var izmantot aizstājējzīmju rakstus (piemēram, `*` atgriezīs hosts, kas atbilst nullei vai vairāk rakstzīmēm; `Zabbix*` atgriezīs hosts, kuru nosaukums sākas ar "Zabbix"). Lai norādītu aizstājējzīmes rakstu, ievadiet virkni manuāli un nospiediet *Enter*. Sākot rakstīt, nolaižamajā sarakstā tiks parādīti atbilstošie hosts, ierobežojot tos līdz hosts, kas pieder atlasītajās *Hosts grupās*, ja tādas ir. Šis parametrs nav pieejams, konfigurējot logrīku veidnes informācijas panelī.
Hosts statuss	Filtrējiet, kurus hosts rādīt, pamatojoties uz to statusu (jebkurš, iespējots, atspējots). Šis parametrs nav pieejams, konfigurējot logrīku veidnes informācijas panelī.
Hosts tagi	Norādiet tagus, lai filtrētu logrīkā parādītos hosts. Varat iekļaut vai izslēgt konkrētus tagus un tagu vērtības. Var iestatīt vairākus nosacījumus. Tagu nosaukumu 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, nav reģistrjutīga); **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, nav reģistrjutīga). Nosacījumiem ir pieejami divi aprēķina veidi: **And/Or** - jāizpilda visi nosacījumi; nosacījumi ar vienādu tagu nosaukumu tiks grupēti, izmantojot nosacījumu Or; **Or** - pietiek, ja ir izpildīts viens nosacījums. Šis parametrs nav pieejams, konfigurējot logrīku veidnes informācijas panelī.
Severity	Atzīmējiet problēmu nopietnības pakāpes, lai filtrētu hosts ar problēmām, kas jāparāda logrīkā. Ja nav atzīmēta neviena nopietnības pakāpe, tiks parādīti visi hosts ar visām problēmām.
Show hosts in maintenance	Atzīmējiet šo izvēles rūtiņu, lai parādītu hosts uzturēšanas režīmā (šādā gadījumā blakus hosts nosaukumam tiks parādīta uzturēšanas ikona). Konfigurējot logrīku veidnes informācijas panelī, šis parametrs tiek saukts *Show data in maintenance*.
Show problems	Filtrējiet, kuras problēmas rādīt, pamatojoties uz to statusu (visas, neapslāpētas, neviena). Problēmas tiek norādītas ar krāsainiem blokiem hosts rindas beigās. Krāsas ir balstītas uz problēmas nopietnības pakāpi, ko var pielāgot sadaļā *Administration > General > Trigger displaying options*. Problēmu nopietnības pakāpes var mainīt, atjauninot problēmas.

Group by	Pievienojiet grupēšanas atribūtu, pēc kura grupēt atlasītos hosts: **Host group** - grupēt hosts pēc to hosts grupas; **Tag value** - ievadiet taga nosaukumu, lai grupētu hosts pēc šī taga vērtībām (piemēram, ievadiet ”city”, lai grupētu hosts pēc vērtībām ”Riga”, ”Tokyo” u. c.); **Severity** - grupēt hosts pēc to problēmu nopietnības pakāpēm. Ja *Show problems* ir konfigurēts problēmu rādīšanai, tās tiek parādītas šādi: - katrai nopietnības pakāpes grupai tiek parādīts tikai atbilstošais problēmu skaits; - katram hosts tiek parādīts visu tā problēmu skaits. Ņemiet vērā, ka hosts tiks grupēti tikai pēc tām nopietnības pakāpēm, kas atzīmētas parametrā *Severity*; ja nav atzīmēta neviena nopietnības pakāpe, visi hosts tiks grupēti pēc visām nopietnības pakāpēm. Grupēšanas atribūtus var pārkārtot, velkot tos uz augšu vai uz leju aiz roktura pirms grupas nosaukuma. Ņemiet vērā, ka grupēšanas atribūtu secība nosaka grupu ligzdošanas secību. Piemēram, norādot vairākus tagu nosaukumus (1: color, 2: city), hosts tiks grupēti pēc krāsas (red, blue u. c.) un pēc tam pēc pilsētas (Riga, Tokyo u. c.). Hosts var tikt parādīts vairākās grupās atkarībā no konfigurētajiem grupēšanas atribūtiem (piemēram, grupējot pēc hosts grupas, ja hosts pieder vairākām hosts grupām). Noklikšķinot uz šādiem hosts, tie tiek atlasīti un izcelti visās grupās. Hosts, kas neatbilst konfigurētajiem grupēšanas atribūtiem, tiek parādīti grupā *Uncategorized*. Var norādīt līdz 10 grupēšanas atribūtiem, un tiem visiem jābūt unikāliem. Ja grupēšanas atribūti nav norādīti, hosts netiks grupēti.
Host limit	Ievadiet maksimālo parādāmo hosts skaitu. Iespējamās vērtības ir no 1 līdz 9999. Ja parādīšanai ir pieejams vairāk hosts nekā iestatītais ierobežojums, zem parādītajiem hosts tiek rādīts atbilstošs ziņojums (piemēram, ”100 of 100+ hosts are shown”). Ņemiet vērā, ka konfigurētais hosts ierobežojums ietekmē arī konfigurēto grupu attēlošanu; piemēram, ja hosts ierobežojums ir iestatīts uz 100 un hosts tiek grupēti pēc tagu vērtībām (vairāk nekā 200), logrīkā tiks parādītas tikai pirmās 100 tagu vērtības ar atbilstošajiem hosts. Šo parametru neietekmē parametrs *Limit for search and filter results* sadaļā *Administration > General > GUI*. Šis parametrs nav pieejams, konfigurējot logrīku veidnes informācijas panelī.

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.).

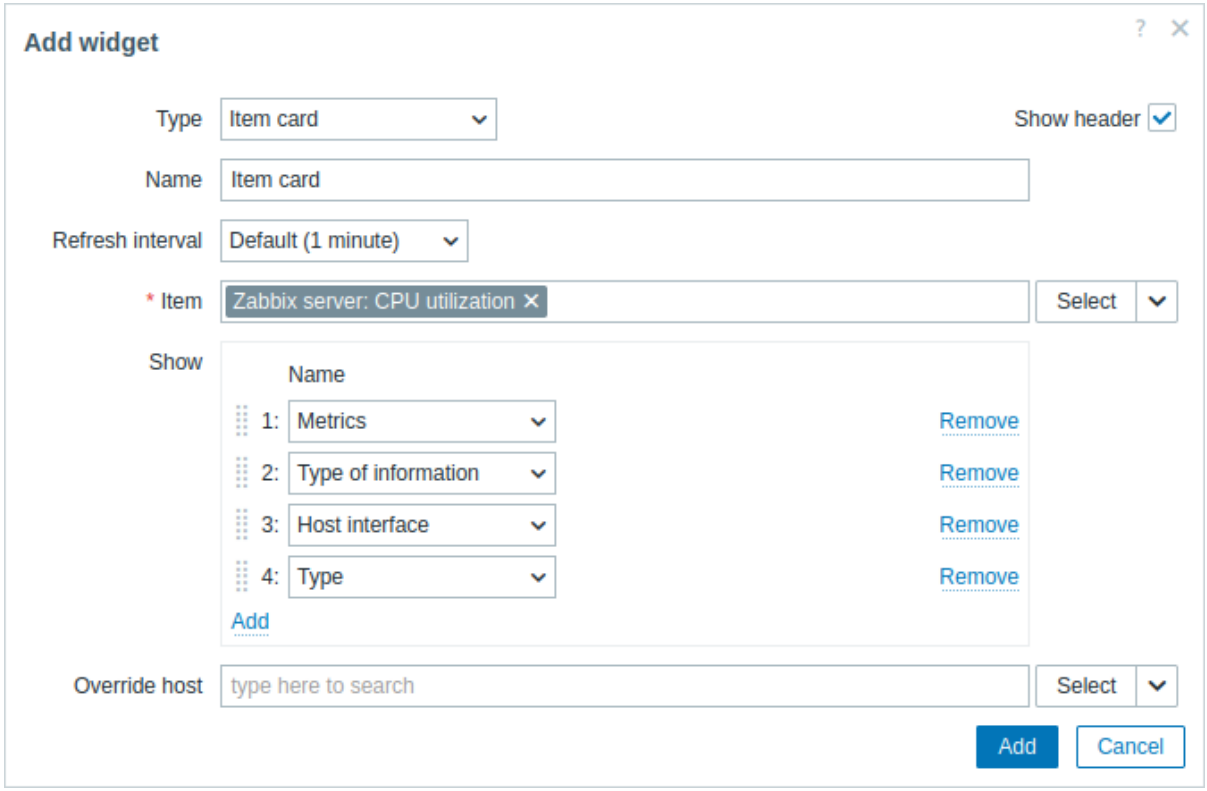


Vienumu var iestatīt tieši logrīka konfigurācijā vai atlasīt no saderīga logrīka informācijas panelī.

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:



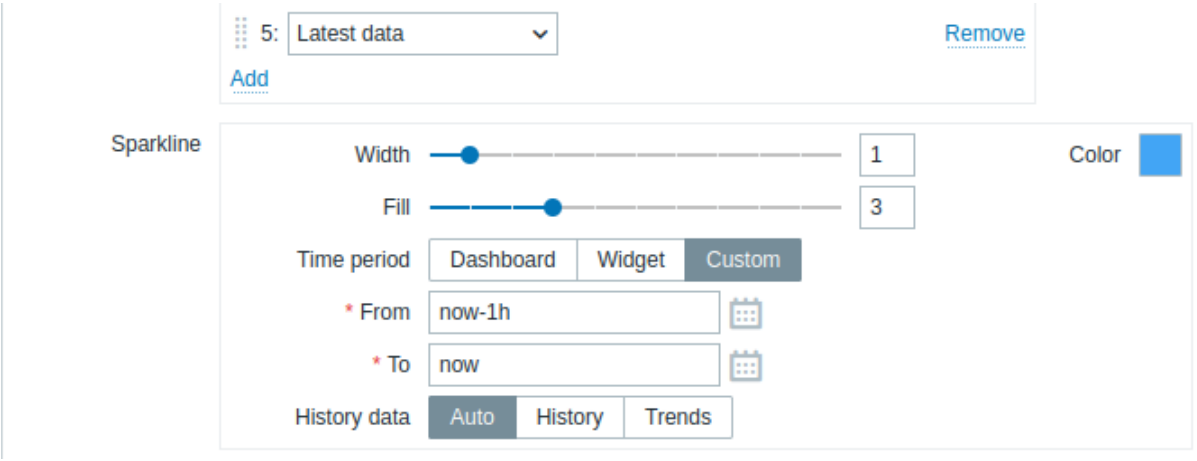
Papildus parametriem, kas ir kopīgi visiem logrīkiem, varat iestatīt šādas specifiskas opcijas:

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). Bināra 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”.

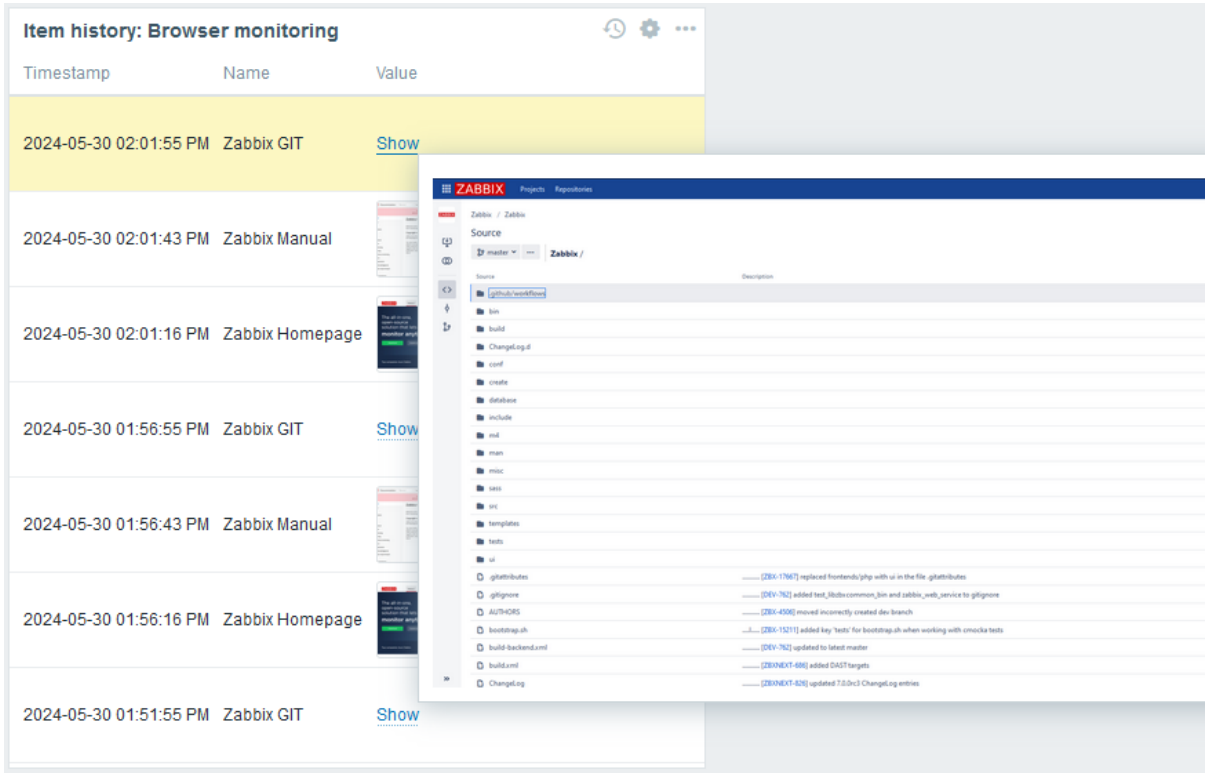
No	Ievadiet vai atlasiet laika perioda sākumu. Tiek atbalstīta relatīvā laika sintakse (`now`, `now/d`, `now/w-1w` utt.). Šis parametrs ir pieejams, ja *Laika periods* ir iestatīts uz "Pielāgots".
Līdz	Ievadiet vai atlasiet laika perioda beigas. Tiek atbalstīta relatīvā laika sintakse (`now`, `now/d`, `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.

Vienuma vēsture

Pārskats

Vienumu vēstures logrīks tabulas formātā parāda jaunākos datus visiem vienumu tipiem (skaitliskiem, teksta u. c.). Tas var rādīt progresa joslas un attēlus (noderīgi pārlūkprogrammas vienumiem) un var izcelt vērtības (noderīgi žurnālfailu uzraudzībai).

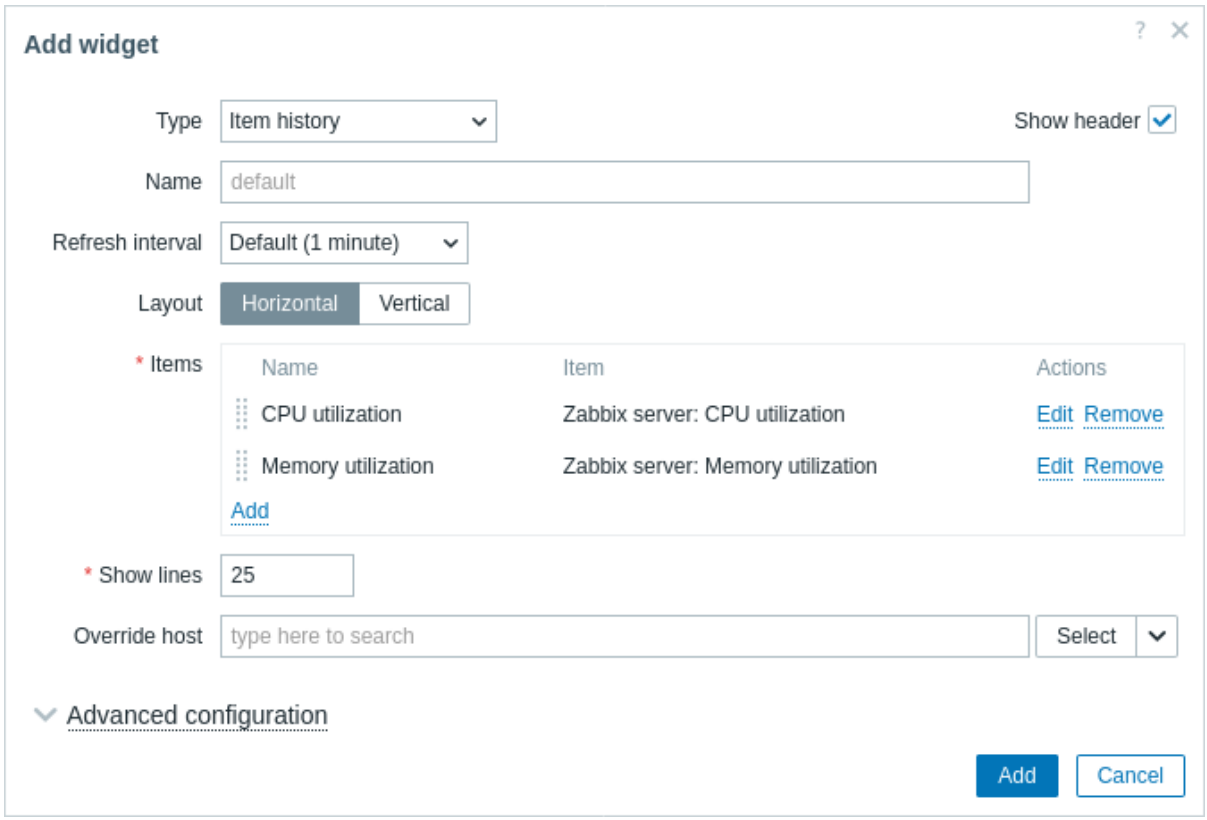




Var tikt parādīti līdz 1000 ierakstiem.

Konfigurācija

Lai konfigurētu, atlasiet *Vienuma vēsture* kā tipu:



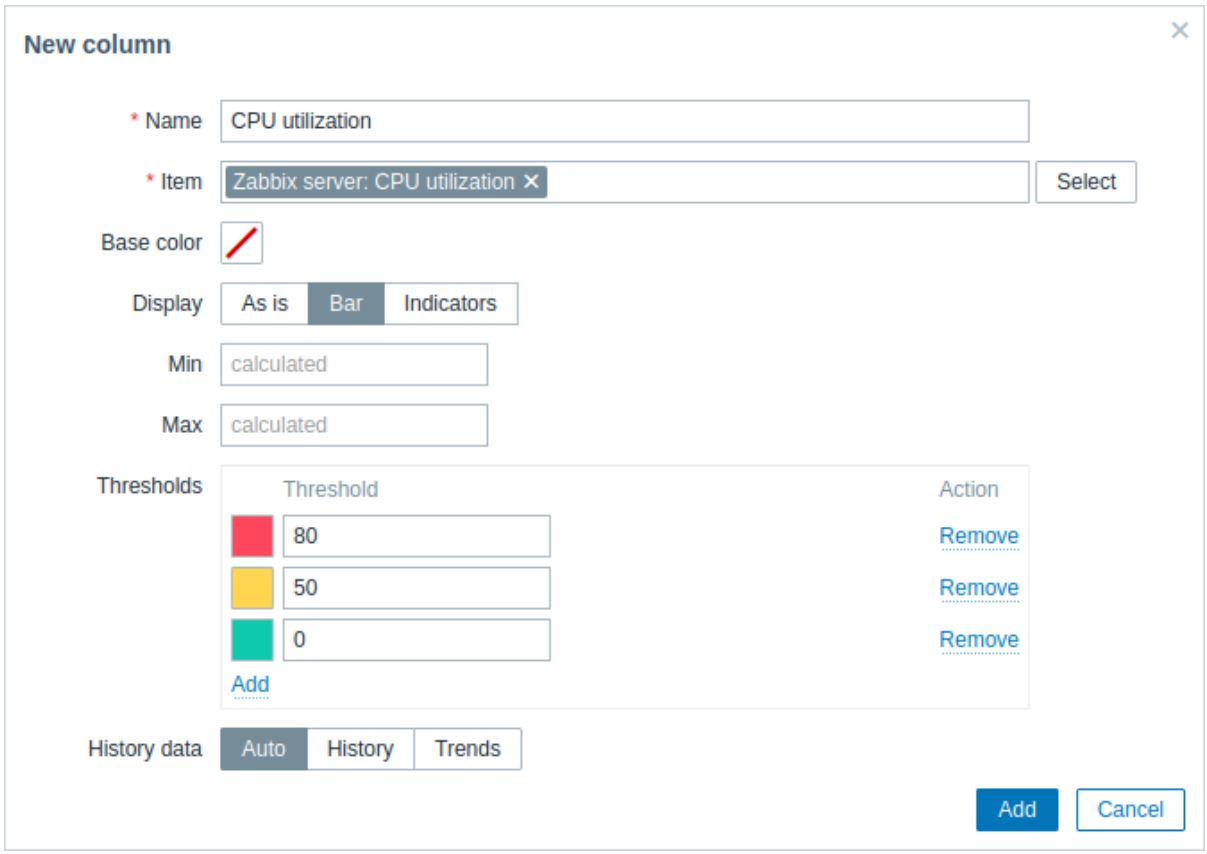
Papildus parametriem, kas ir kopīgi visiem logrīkiem, varat iestatīt šādas specifiskas opcijas:

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.

Rakstzīmju, teksta, žurnāla un JSON tipa vienumu specifiskie kolonnu parametri:

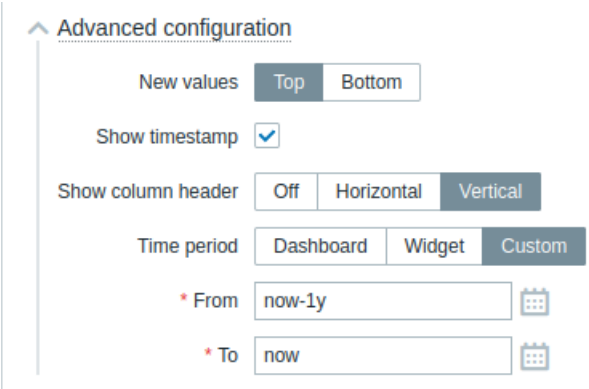
Izcēlumi	Noklikšķiniet uz *Add*, lai pievienotu izcēlumu, atlasiet izcēluma krāsu krāsu atlasītājā un norādiet regulāro izteiksmi. Atlasītā krāsa tiks izmantota kā fona krāsa vienumu vērtībām, kur norādītā regulārā izteiksme atbilst tekstam.
Attēlošana	Atlasiet, kā vienuma vērtība jāattēlo: **As is** - tieši tā, kā saņemta. Rindu pārneses tiek saglabātas, ja vērtība satur rindu beigas. Vārdu aplaušana tiek lietota, ja *Layout* ir iestatīts uz ”Vertical” un ir konfigurēta vairāk nekā viena kolonna; **HTML** - kā HTML formatēts teksts; **Single line** - kā viena rinda, saīsināta līdz norādītam garumam (1-500 rakstzīmes). Novietojot kursoru virs saīsinātās vērtības vai noklikšķinot uz tās, tiek atvērts uznirstošais logs ar pilnu vērtību.
Use monospace font	Atzīmējiet šo izvēles rūtiņu, lai attēlotu vienuma vērtību monospace fontā (pēc noklusējuma nav atzīmēta).
Display local time	Atzīmējiet šo izvēles rūtiņu, lai timestamp kolonnā attēlotu vietējo laiku, nevis timestamp. Ņemiet vērā, ka arī izvēles rūtiņai *Show timestamp* advanced configuration sadaļā jābūt atzīmētai. Šis parametrs ir pieejams tikai log 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*:

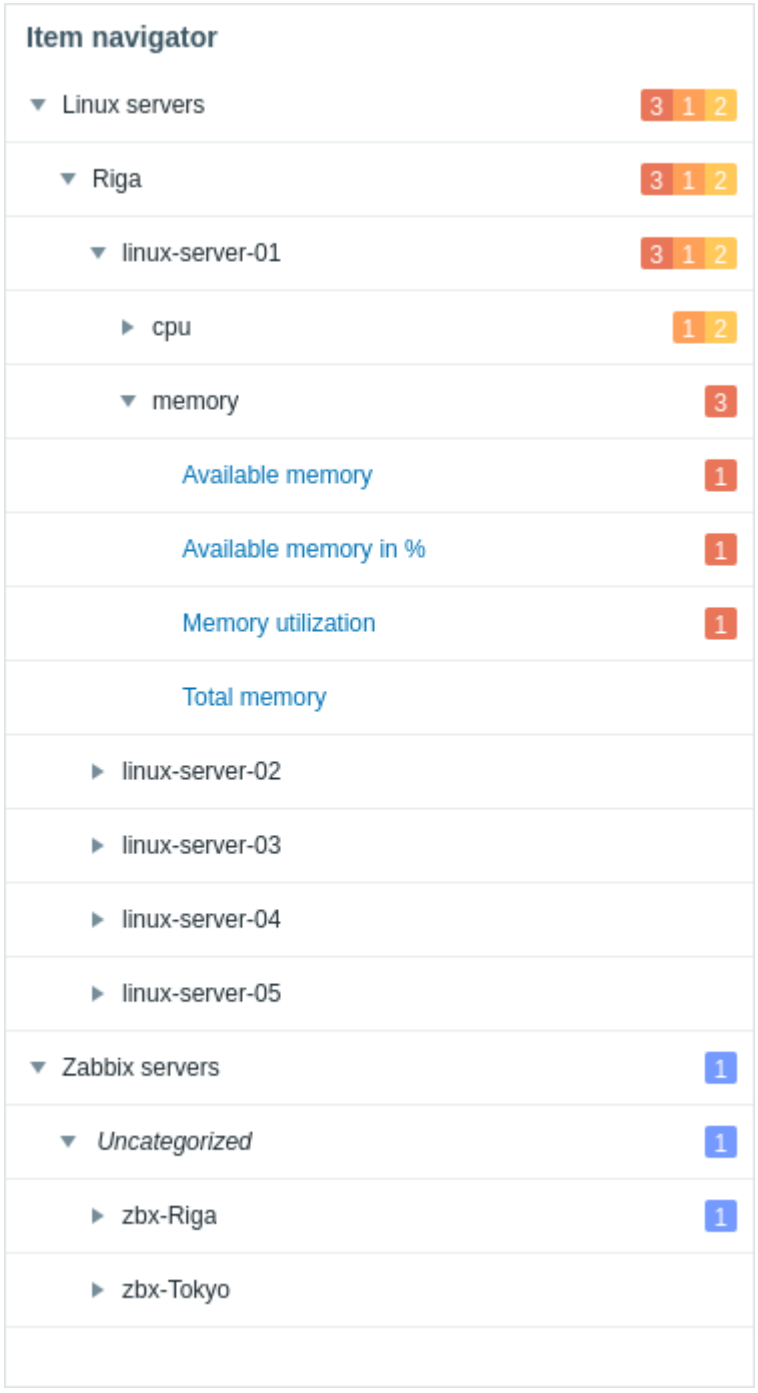


Jaunās vērtības	Izvēlieties, kur jāpievieno jaunās vienumu vērtības: **Augšā** — kolonnu augšdaļā; **Apakšā** — kolonnu apakšdaļā.
Rādīt laikspiedolu	Atzīmējiet šo izvēles rūtiņu, lai parādītu laikspiedola kolonnu (pēc noklusējuma nav atzīmēta).
Rādīt kolonnas galveni	Izvēlieties kolonnas galvenes orientāciju: **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”.

Vienumu navigators

Pārskats

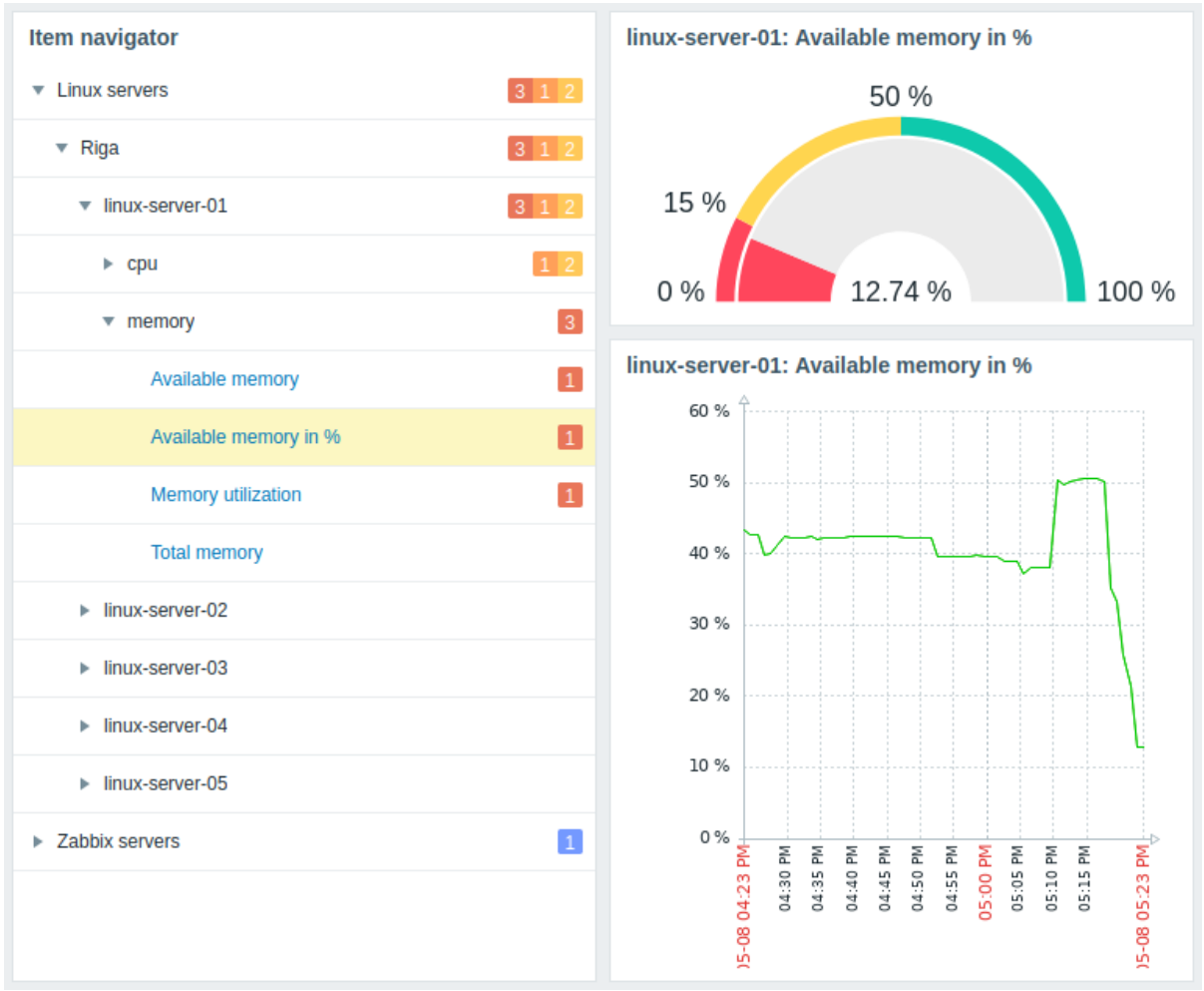
Logrīks *Vienumu navigators* attēlo vienumu sarakstu, pamatojoties uz dažādām filtrēšanas un grupēšanas opcijām.



Grupas, pēc kurām vienumi ir organizēti, var izvērst vai sakļaut.

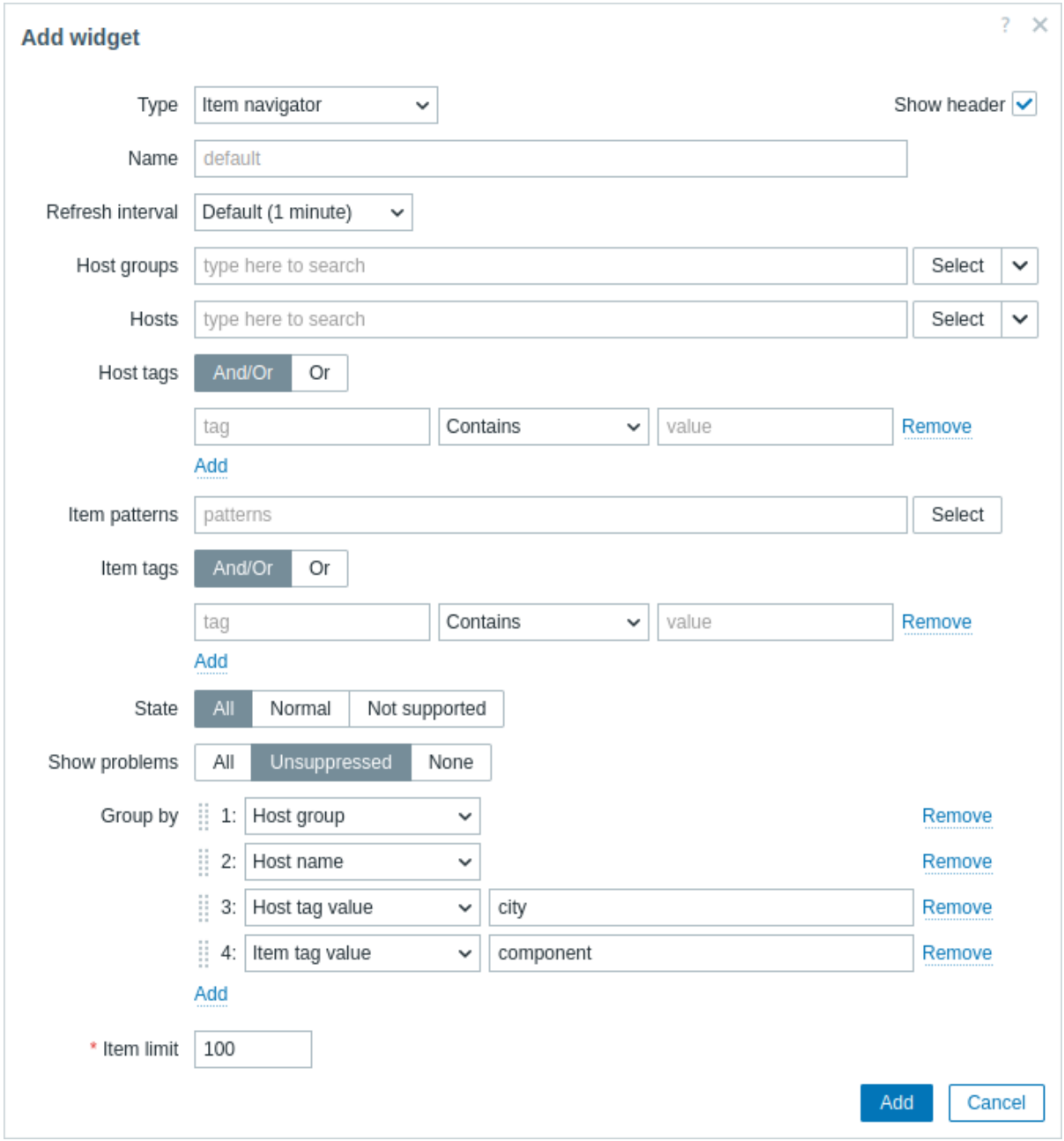
Grupām un problēmām papildu informācija ir pieejama, novietojot peles kursoru virs tām.

Logrīks ir visnoderīgākais, lai kontrolētu, ko attēlo citi logrīki, pamatojoties uz atlasīto vienumu.



Konfigurācija

Lai konfigurētu, kā tipu atlasiet *Vienumu navigatoru*:



Papildus parametriem, kas ir kopīgi visiem logrīkiem, varat iestatīt tālāk norādītās specifiskās opcijas:

Hosts grupas	Atlasiet hosts grupas. Varat arī atlasīt saderīgu logrīku kā hosts grupu datu avotu. Šajā laukā tiek izmantota automātiskā pabeigšana, tāpēc, sākot rakstīt grupas nosaukumu, tiks parādīts atbilstošo grupu nolaižamais saraksts. Atlasot vecākelementa hosts grupu, tiek netieši atlasītas visas ligzdotās hosts grupas; ja nav atlasīta neviena hosts grupa, logrīks parādīs vienumus, kas pieder visiem visu hosts grupu hostiem. Šis parametrs nav pieejams, konfigurējot logrīku veidnes informācijas panelī.
Hosti	Atlasiet hostus. Varat arī atlasīt saderīgu logrīku vai informācijas paneli kā hostu datu avotu. Šajā laukā tiek izmantota automātiskā pabeigšana, tāpēc, sākot rakstīt hosts nosaukumu, tiks parādīts atbilstošo hostu nolaižamais saraksts. Ja nav atlasīts neviens hosts, logrīks parādīs vienumus, kas pieder visiem hostiem. Šis parametrs nav pieejams, konfigurējot logrīku veidnes informācijas panelī.

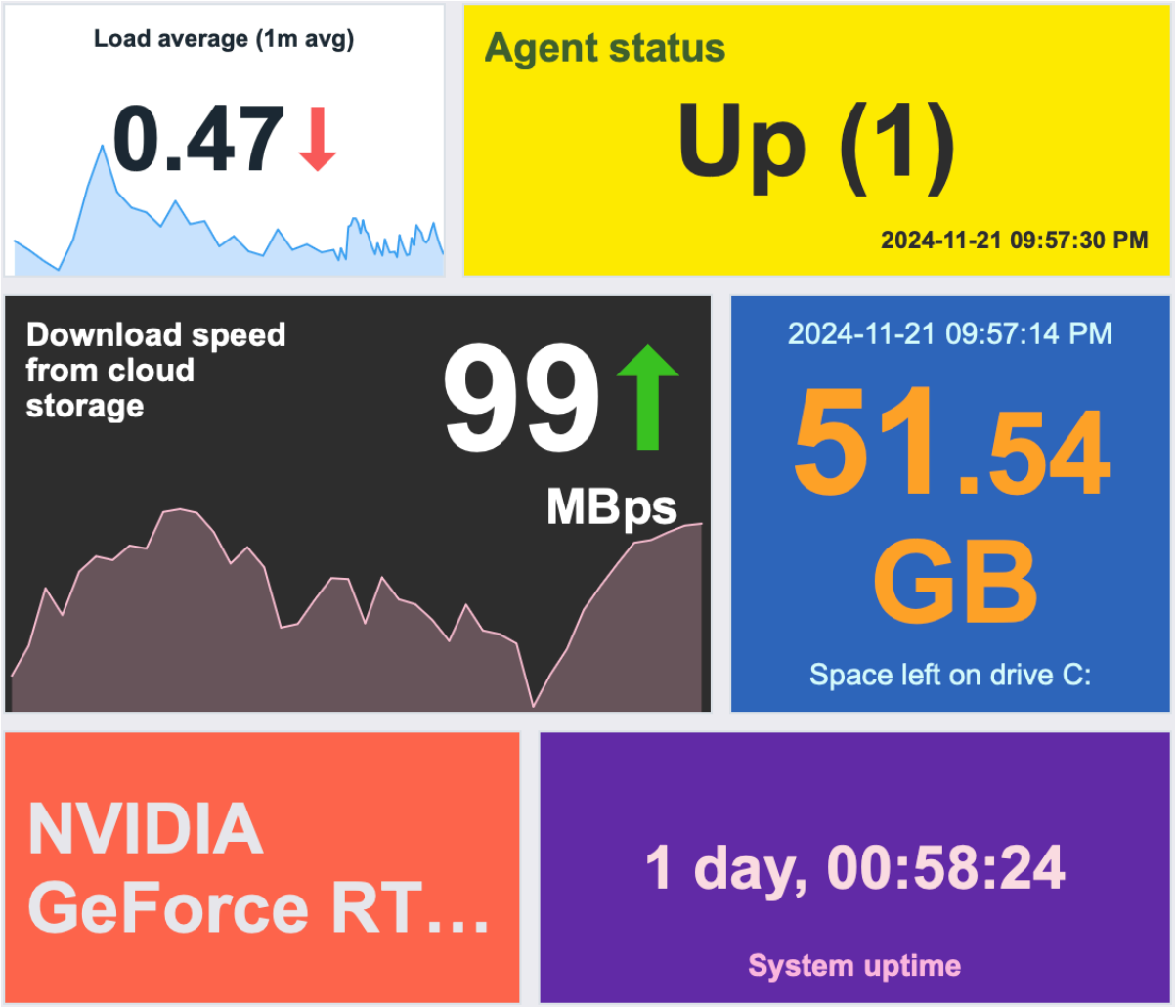
Hosts birkas	Norādiet hosts birkas, lai filtrētu logrīkā parādītos vienumus. Varat iekļaut vai izslēgt konkrētas birkas un birku vērtības. Var iestatīt vairākus nosacījumus. Birkas nosaukuma atbilstība vienmēr ir reģistrjutīga. Katram nosacījumam ir pieejami vairāki operatori: **Exists** - iekļaut norādītos birku nosaukumus; **Equals** - iekļaut norādītos birku nosaukumus un vērtības (reģistrjutīgi); **Contains** - iekļaut norādītos birku nosaukumus, kuru birku vērtības satur ievadīto virkni (apakšvirknes atbilstība, nav reģistrjutīga); **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, nav reģistrjutīga). Nosacījumiem ir divi aprēķina veidi: **And/Or** - jāizpilda visi nosacījumi; nosacījumi ar vienādu birkas nosaukumu tiks grupēti, izmantojot nosacījumu Or; **Or** - pietiek ar viena nosacījuma izpildi. Šis parametrs nav pieejams, konfigurējot logrīku veidnes informācijas panelī.
Vienumu šabloni	Ievadiet vienumu šablonus vai atlasiet esošos vienumus kā vienumu šablonus. Vienumi, kas atbilst norādītajiem šabloniem, tiks parādīti vienumu navigatorā. Atlasīšanai 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*. Sākot rakstīt, nolaižamajā sarakstā tiks parādīti atbilstošie vienumi, kas ierobežoti ar atlasītajiem *Hostiem* vai hostiem atlasītajās *Hosts grupās*, ja tādas ir. Aizstājējzīmes simbols vienmēr tiek interpretēts, tāpēc, piemēram, nav iespējams atsevišķi pievienot vienumu ar nosaukumu *item*∗, ja ir arī citi atbilstoši vienumi (piemēram, item2, item3). Konfigurējot logrīku veidnes informācijas panelī, šis parametrs ļauj atlasīt tikai veidnē konfigurētos vienumus.
Vienumu birkas	Norādiet vienumu birkas, lai filtrētu logrīkā parādītos vienumus. Varat iekļaut vai izslēgt konkrētas 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, nav reģistrjutīga); **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, nav reģistrjutīga). Nosacījumiem ir divi aprēķina veidi: **And/Or** - jāizpilda visi nosacījumi; nosacījumi ar vienādu birkas nosaukumu tiks grupēti, izmantojot nosacījumu Or; **Or** - pietiek ar viena nosacījuma izpildi.
Stāvoklis	Filtrējiet parādāmos vienumus pēc to stāvokļa (visi, normāli, neatbalstīti).
Rādīt problēmas	Filtrējiet parādāmās problēmas pēc to statusa (visas, neapslāpētas, 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 nopietnību var mainīt, atjauninot problēmas.

Grupēt pēc	Pievienojiet grupēšanas atribūtu, pēc kura grupēt vienumus: **Hosts grupa** - grupēt vienumus pēc to hostu hosts grupām; **Hosts nosaukums** - grupēt vienumus pēc to hostiem; **Hosts birkas vērtība** - ievadiet birkas nosaukumu, lai grupētu vienumus pēc šīs hosts birkas vērtībām (piemēram, ievadiet ”city”, lai grupētu vienumus pēc vērtībām ”Riga”, ”Tokyo” u. c.); **Vienuma birkas vērtība** - ievadiet birkas nosaukumu, lai grupētu vienumus pēc šīs vienuma birkas vērtībām (piemēram, ievadiet ”component”, lai grupētu vienumus pēc vērtībām ”cpu”, ”memory” u. c.). Grupēšanas atribūtus var pārkārtot, velkot tos uz augšu vai uz leju aiz roktura pirms grupas nosaukuma. Ņemiet vērā, ka grupēšanas atribūtu secība nosaka grupu ligzdošanas secību. Piemēram, norādot vairākus hosts birku nosaukumus (1: color, 2: city), vienumi tiks grupēti pēc krāsas (red, blue u. c.) un pēc tam pēc pilsētas (Riga, Tokyo u. c.). Vienums var tikt parādīts vairākās grupās atkarībā no konfigurētajiem grupēšanas atribūtiem (piemēram, grupējot pēc hosts grupas, ja vienuma hosts pieder vairākām hosts grupām). Noklikšķinot uz šādiem vienumiem, tie tiek atlasīti un izcelti visās grupās. Vienumi, kas neatbilst konfigurētajiem grupēšanas atribūtiem, tiek parādīti grupā *Nekategorizēti*. Ja *Rādīt problēmas* ir konfigurēts problēmu parādīšanai, tās tiek parādītas šādi: - katrai grupai tiek parādīts visu pakārtoto vienumu problēmu skaits; - katram vienumam tiek parādīts tikai tā problēmu skaits. Var norādīt līdz 10 grupēšanas atribūtiem, un tiem visiem jābūt unikāliem. Ja grupēšanas atribūti nav norādīti, vienumi netiks grupēti.
Vienumu ierobežojums	Ievadiet maksimālo parādāmo vienumu skaitu. Iespējamās vērtības ir no 1 līdz 9999. Ja parādīšanai ir pieejams vairāk vienumu nekā iestatītais ierobežojums, zem parādītajiem vienumiem tiek rādīts atbilstošs ziņojums (piemēram, ”Tiek parādīti 100 no vairāk nekā 100 vienumiem”). Ņemiet vērā, ka konfigurētais vienumu ierobežojums ietekmē arī konfigurēto grupu attēlošanu; piemēram, ja vienumu ierobežojums ir iestatīts uz 100 un vienumi tiek grupēti pēc to hostiem (katrā ir 200 vienumu), logrīkā tiks parādīts tikai pirmais hosts ar tā 100 vienumiem. Šo parametru neietekmē parametrs *Meklēšanas un filtra rezultātu ierobežojums* sadaļā *Administrēšana > Vispārīgi > GUI*.

Vienuma vērtība

Pārskats

Vienuma vērtība sīkrīks parāda viena skaitliska vai teksta vienuma vērtību. Tas ir noderīgs, lai uzraudzītu svarīgus rādītājus, vizualizētu sliekšņus un noteiktu pēkšņas izmaiņas datos.



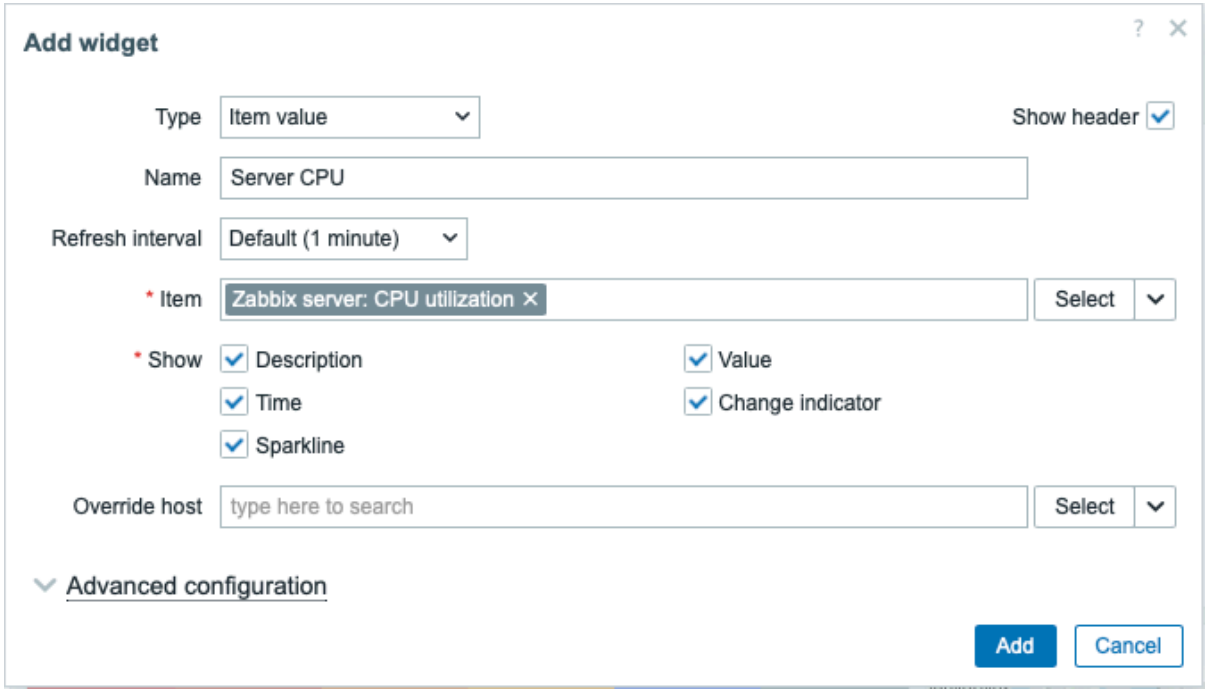
Varat konfigurēt sīkrīku, lai tas parādītu šādu informāciju:

- Metrikas laikspiedols (2024-11-21 09:57:30)
- Vienuma apraksts (Load average (1m), aģenta statuss)
- Vienuma vērtība, izmaiņu indikators un vienības (0.47 ↓, 99 ↑ Mbps)
- Fona krāsa
- Sparkline diagramma vērtībām no norādītā laika perioda

Noklikšķinot uz sīkrīka, tiek atvērts ad hoc grafiks skaitliskajiem vienumiem vai jaunākie dati teksta vienumiem.

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 virsraksta *Paplašinātā konfigurācija*, lai parādītu paplašinātās konfigurācijas opcijas.

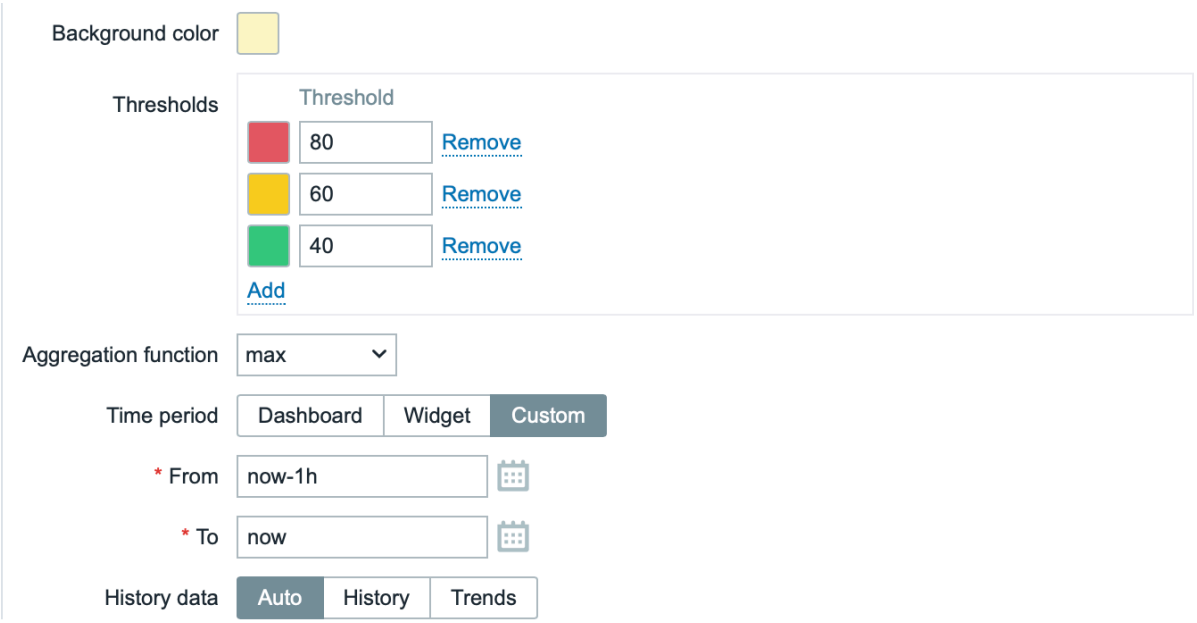
Papildu konfigurācija

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 visa logrīka fona krāsu (statisku vai dinamisku) un vērtību attēlošanas apkopošanas funkciju.

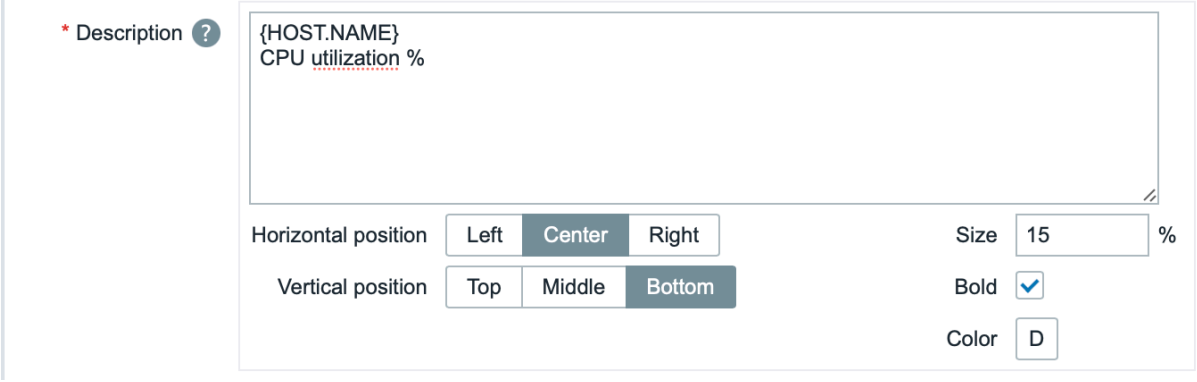


Fona krāsa	Atlasiet visa logrīka fona krāsu krāsu atlasītājā. D apzīmē noklusējuma krāsu (tā ir atkarīga no lietotāja saskarnes motīva). Lai atjaunotu noklusējuma vērtību, krāsu atlasītājā noklikšķiniet uz pogas *Izmantot noklusējumu*.
Sliekšņi	Konfigurējiet visa logrīka dinamisko fona krāsu. Noklikšķiniet uz *Pievienot*, lai pievienotu slieksni, atlasiet fona krāsu krāsu atlasītājā un norādiet skaitlisku vērtību. Kad vienuma vērtība ir vienāda ar sliekšņa vērtību vai lielāka par to, fona krāsa mainīsies. Saglabājot, saraksts tiks sakārtots augošā secībā. Ņemiet vērā, ka dinamiskā fona krāsa tiks pareizi attēlota tikai skaitliskiem vienumiem.
Apkopošanas funkcija	Norādiet, kuru apkopošanas 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 **neizmantot**, tiek attēlota jaunākā vērtība (bez apkopošanas). Apkopošana ļauj attēlot apkopotu vērtību norādītajā intervālā (5 minūtēs, stundā, dienā), nevis jaunāko vērtību. Funkcijām *min*, *max*, *avg* un *sum* var attēlot tikai skaitliskus datus. Funkcijai *count* neskaitliskie dati tiks pārveidoti par skaitliskiem.
Laika periods	Laika periods, kas jāizmanto vērtību apkopošanai. Atlasiet datu avotu laika periodam: **Informācijas panelis** - izmantot informācijas 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 rādītāju virs tās, norāda iestatīto laiku. Ņemiet vērā, ka neatkarīgi no logrīka *Laika perioda* konfigurācijas saderīgi logrīki joprojām var to izmantot kā laika perioda datu avotu. Šis parametrs ir paslēpts, ja *Apkopošanas funkcijai* ir iestatīta vērtība ”neizmantot”.
Logrīks	Ievadiet vai atlasiet saderīgu logrīku kā laika perioda datu avotu. Šis parametrs ir pieejams, ja *Laika periodam* ir iestatīta vērtība ”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 periodam* ir iestatīta vērtība ”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 periodam* ir iestatīta vērtība ”Pielāgots”.

Vēsturiskie dati	Iegūt datus no vēstures vai tendencēm: **Automātiski** - automātiska atlase; **Vēsture** - iegūt vēsturiskos datus; **Tendences** - iegūt tendenču datus. Šis iestatījums attiecas tikai uz skaitliskiem datiem. Neskaitliskie dati vienmēr tiks iegūti no vēstures.

Apraksts

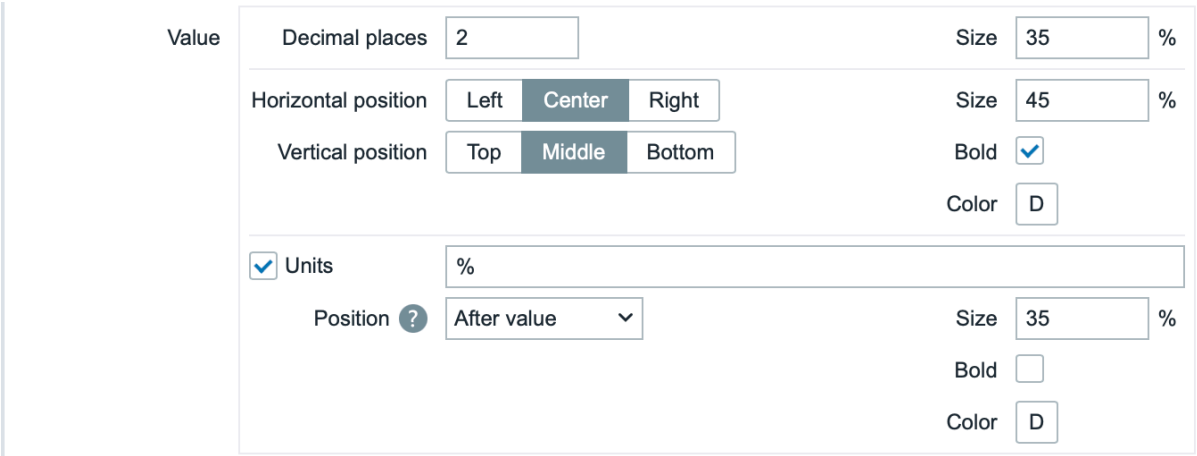
Šie parametri nosaka, kā jāattēlo vienuma apraksts.



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.

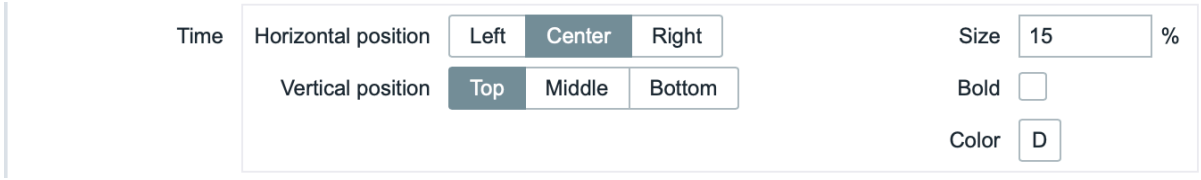


Decimālvietas	Izvēlieties, cik daudz decimālvietu tiks attēlots kopā ar vērtību. Šī vērtība ietekmēs tikai peldošā komata vienumus. Vienumiem, kuriem mērvienības ir iestatītas uz ”s”, ja šī vērtība ir lielāka par 0 (noklusējuma vērtība ir 2), logrīks noapaļo nozīmīgāko laika vienību un attēlo skaitlisku 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 varat atlasīt izmaiņu indikatoru krāsu, izmantojot krāsu atlasītāju.



Izmaiņu indikatori ir šādi:

- ↑ - vienuma vērtība ir palielinājusies (skaitliskiem vienumiem)
- ↓ - vienuma vērtība ir samazinājusies (skaitliskiem vienumiem)
- ↕ - vienuma vērtība ir mainījusies (virkņu vienumiem un vienumiem ar vērtību kartējumu)

Piezīme:

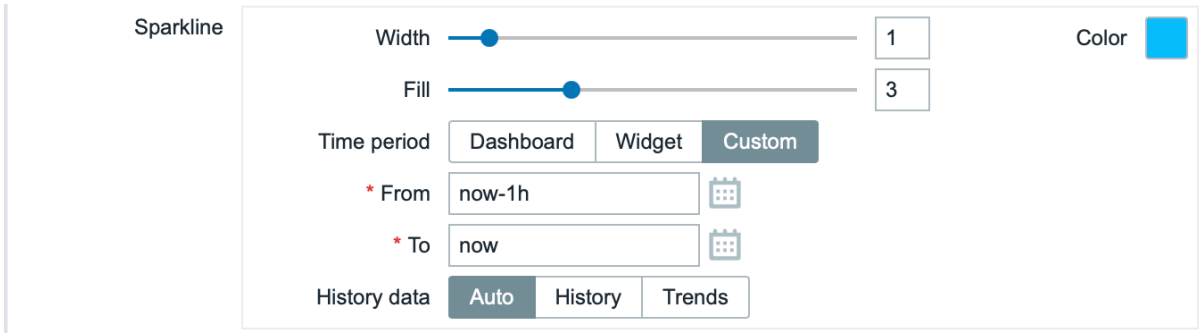
- Augšup (↑) un lejup (↓) indikatori netiek rādīti, ja ir tikai viena vērtība.
- Vērtības vienmēr tiek salīdzinātas vienā un tajā pašā tvērumā; piemēram:
  - Jaunākā vērtība tiek salīdzināta ar iepriekšējo vērtību.
  - Mēneša vērtība tiek salīdzināta ar iepriekšējo mēnesi.
  - Apkopojumiem iepriekšējam periodam ir tāds pats ilgums kā atlasītajam periodam, un tas beidzas tieši pirms atlasītā perioda sākuma.

D apzīmē noklusējuma krāsu (tā ir atkarīga no lietotāja saskarnes motīva). Lai atjaunotu noklusējuma vērtību, krāsu atlasītājā noklikšķiniet uz pogas *Izmantot noklusējumu*.

Izmaiņu indikatora vertikālais izmērs ir vienāds ar vērtības izmēru (skaitliskām vērtībām - vērtības veselo daļu).

Sparkline

Šie parametri nosaka, kā jāattēlo sparkline diagramma.

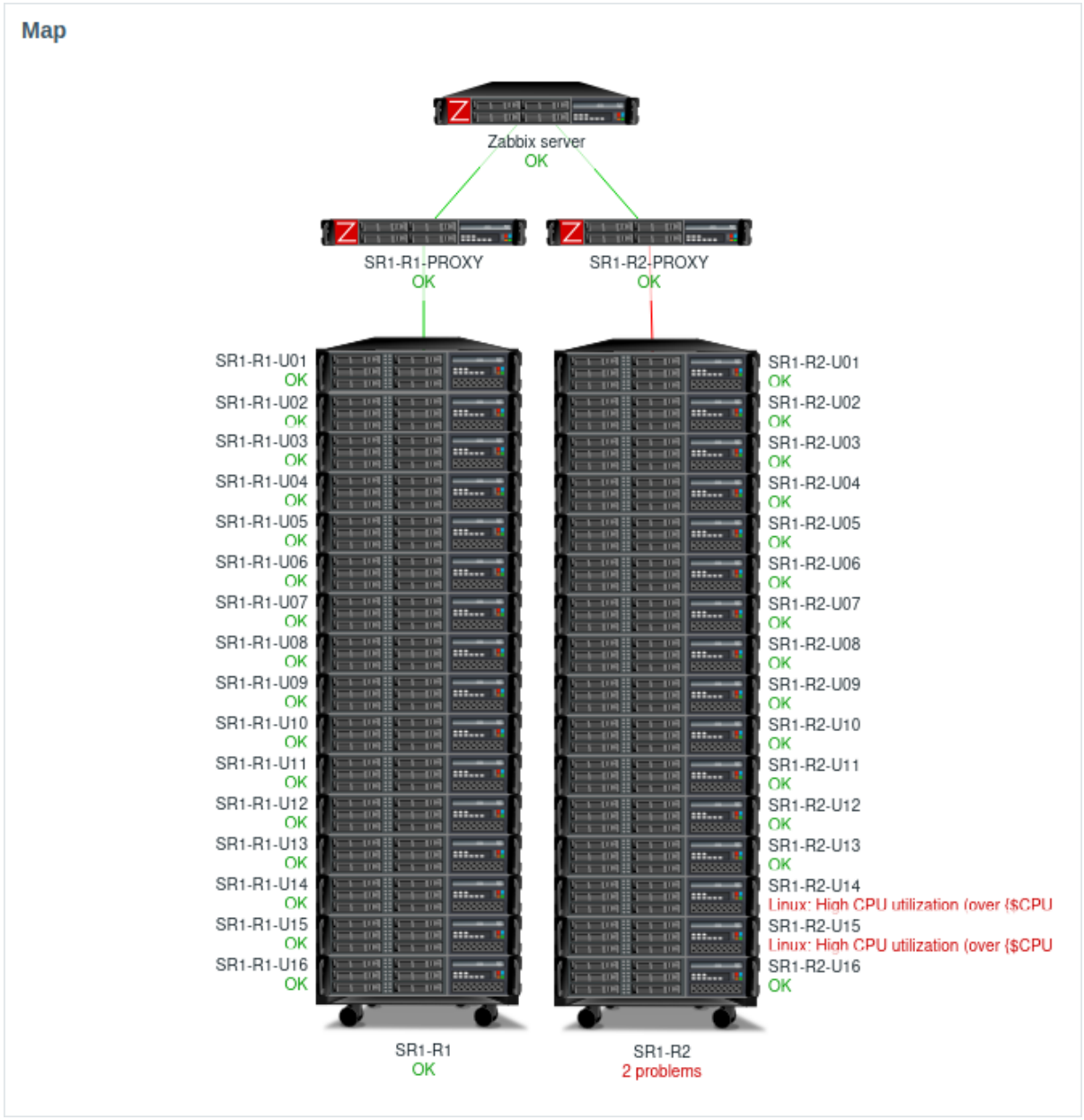


Width	Iestatiet diagrammas līnijas biezumu, izmantojot slīdni vai manuāli ievadot vērtību diapazonā no 0 līdz 10.
Color	Atlasiet līnijas un aizpildījuma krāsu.
Fill	Iestatiet aizpildījuma krāsas caurspīdīguma līmeni, izmantojot slīdni vai manuāli ievadot vērtību diapazonā no 0 līdz 10.
Time period	Norādiet 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.

Karte

Pārskats

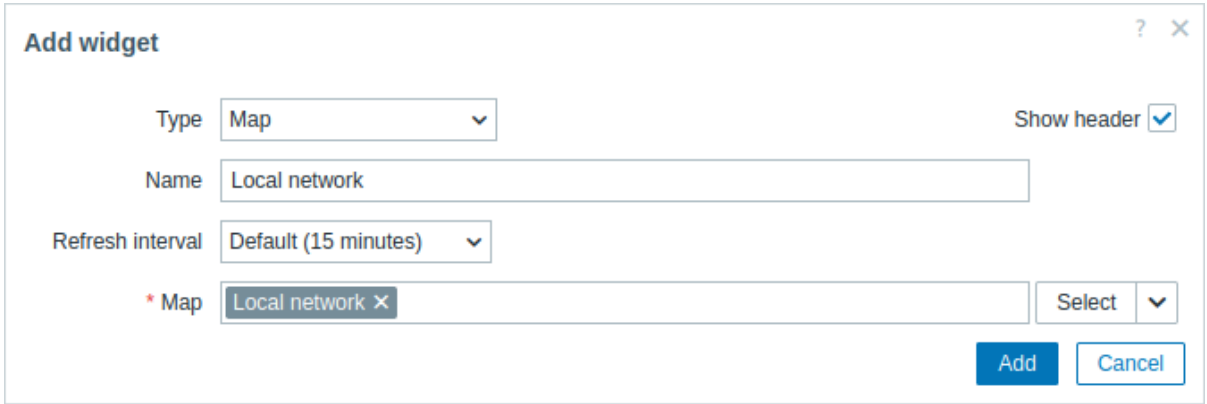
Logrīks *Karte* var attēlot tīkla karti vai līdzīgu vizualizāciju, sniedzot jums dinamisku jūsu tīkla pārskatu.



Varat to savienot pārī ar logrīku Kartes navigācijas koks, lai dinamiski attēlotu kokā atlasīto karti.

Konfigurācija

Lai konfigurētu, atlasiet *Map* kā tipu:



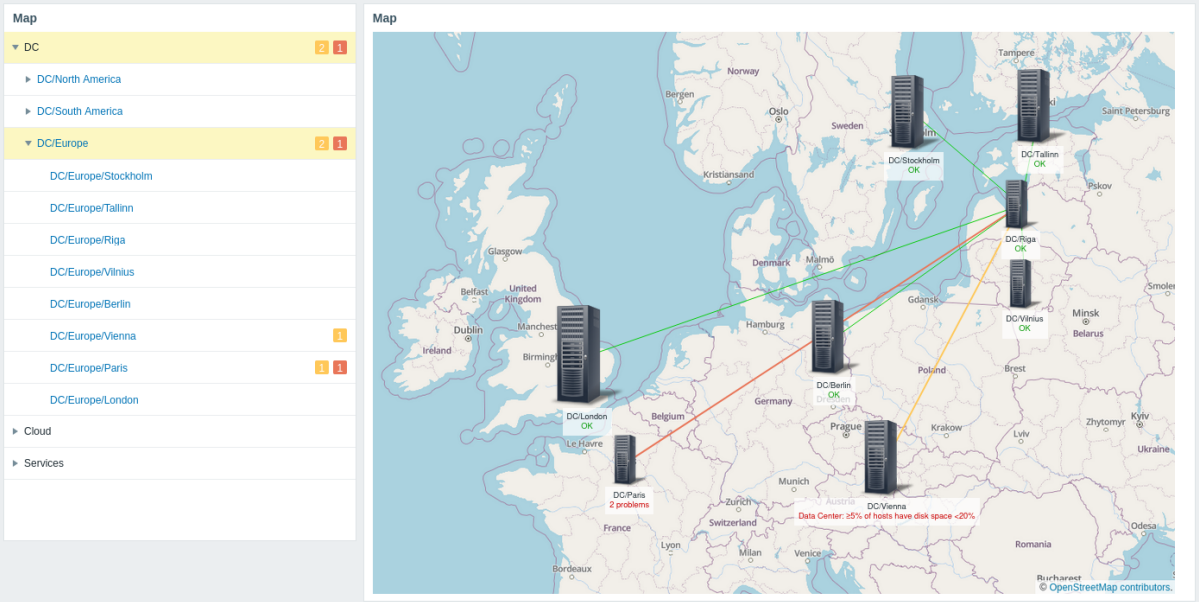
Papildus parametriem, kas ir kopīgi visiem logrīkiem, varat iestatīt šādas specifiskas opcijas:

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.

Kartes navigācijas koks

Pārskats

Logrīks *Kartes navigācijas koks* attēlo esošo karšu hierarhiju un problēmu skaitu katrai kartei un karšu grupai.

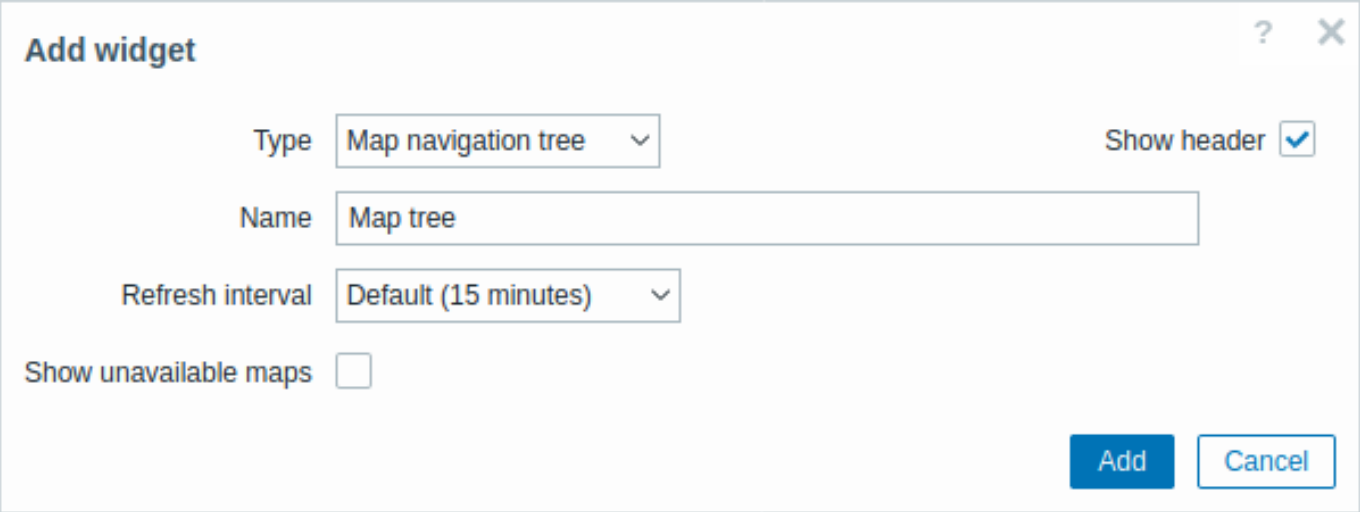


To var savienot pārī ar logrīku Karte, lai dinamiski attēlotu kokā atlasīto karti.

Hierarhijas augstākā līmeņa kartes problēmu skaits ietver visas apakškaršu problēmas, kā arī pašas kartes problēmas.

Konfigurācija

Lai konfigurētu, atlasiet *Kartes navigācijas koks* kā tipu:

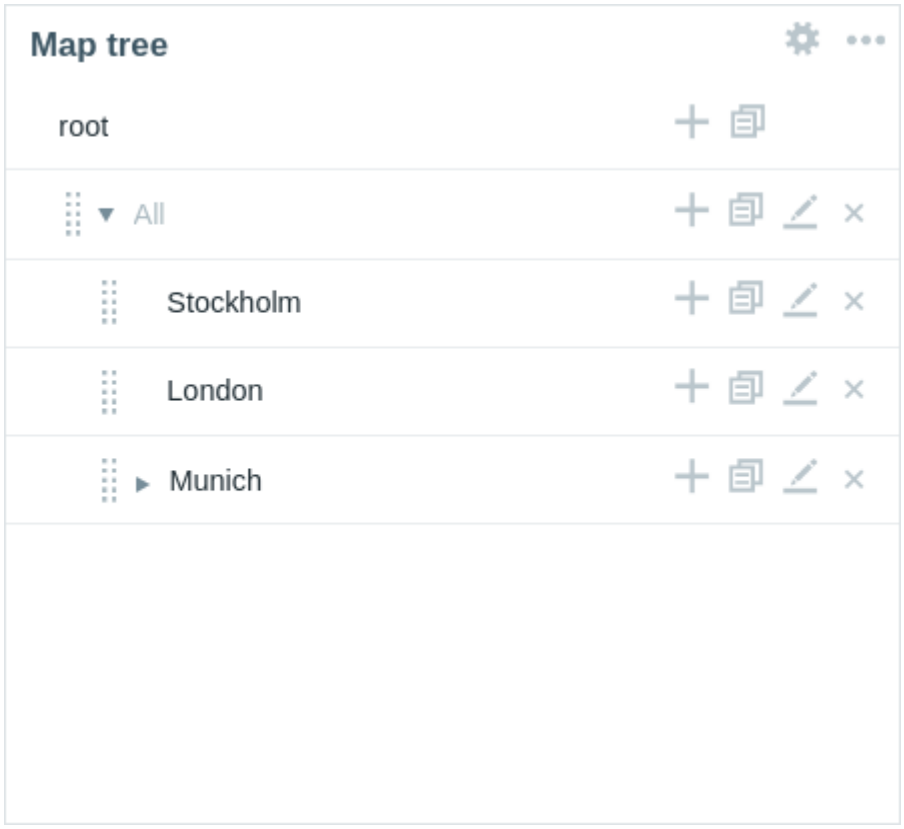


Papildus parametriem, kas ir kopīgi visiem logrīkiem, varat iestatīt šādas specifiskas opcijas:

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.

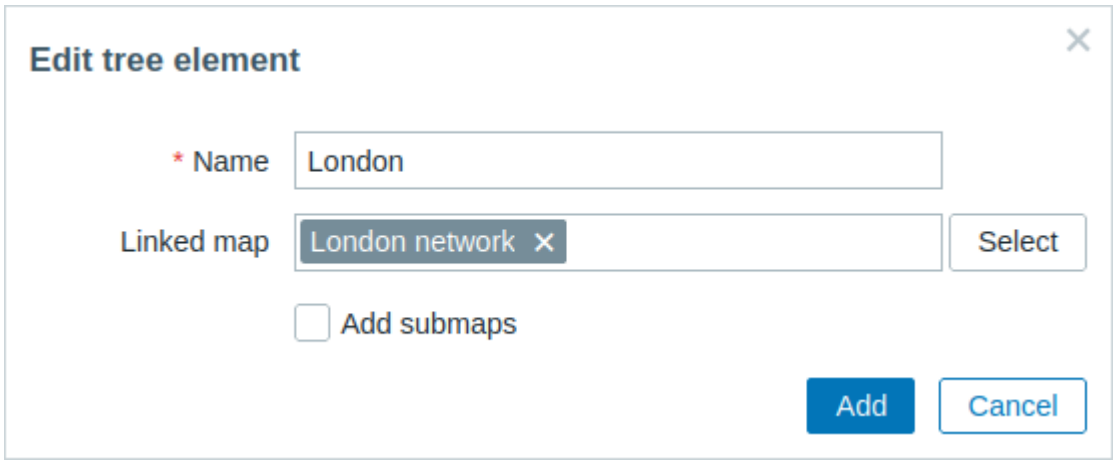
Navigācijas koka elementi tiek parādīti sarakstā. Jūs varat:

- 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.



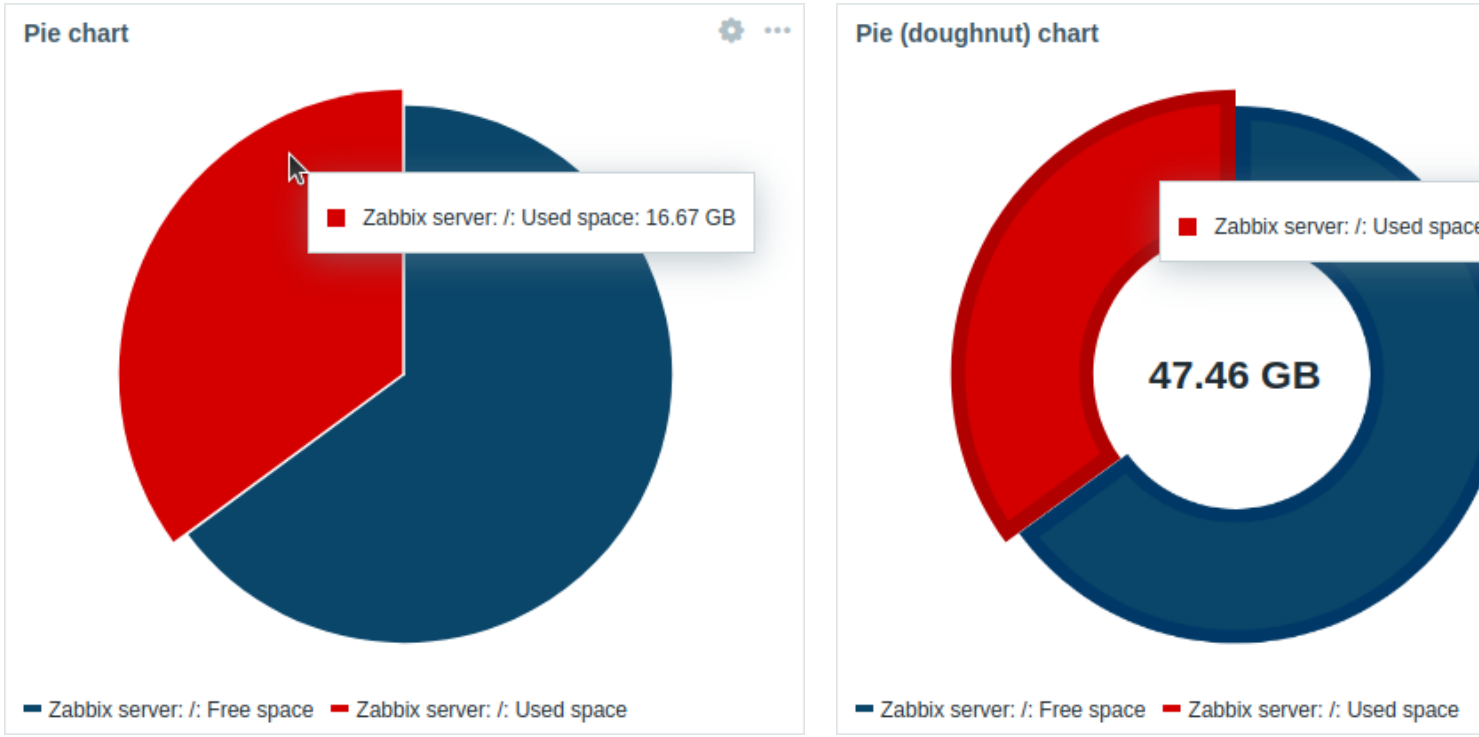
Ir pieejami šādi navigācijas koka elementa konfigurācijas parametri:

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.

Sektoru diagramma

Pārskats

Sektoru diagrammas logrīks attēlo vienumu skaitliskos datus kā vektoru sektoru vai gredzenveida diagrammu. Šis logrīks var palīdzēt vizualizēt, kā vienumi vai hosti veido kopējo datu kopu.

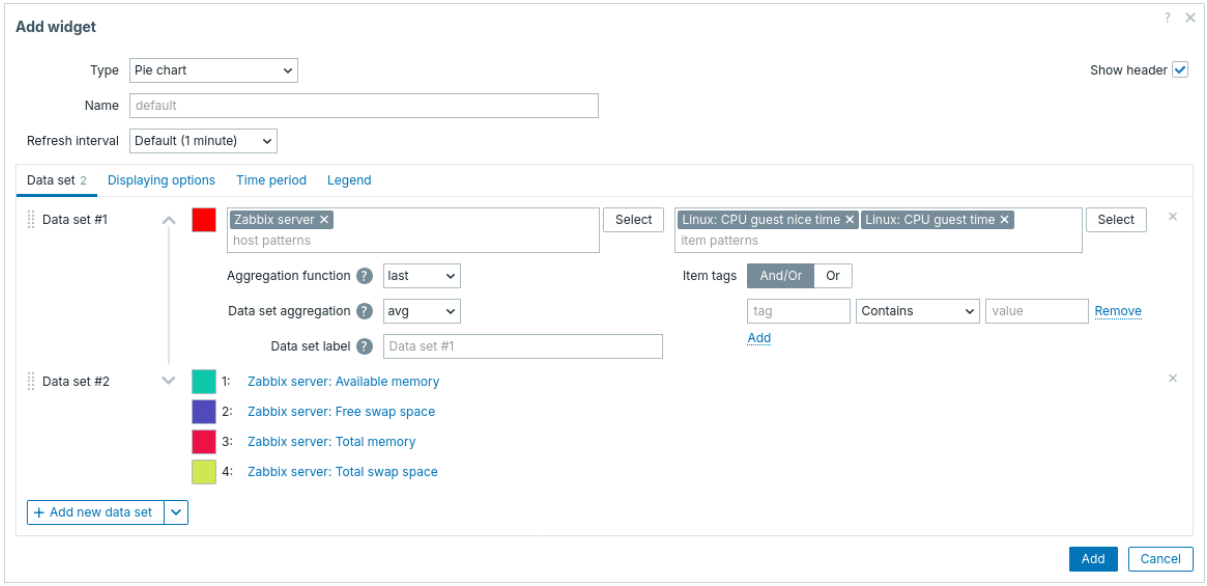


Novietojot peles rādītāju virs sektora, tas tiek izvērsts un tiek parādīta tā leģenda; noklikšķinot uz sektora, tas paliek izvērsts un tā leģenda ir redzama, līdz tā tiek aizvērta.

Sektoru diagrammas logrīkā redzamo informāciju var lejupielādēt kā PNG attēlu, logrīka izvēlnē *Darbības* ... atlasot opciju *Lejupielādēt attēlu*. Šī izvēlne tiek parādīta, kad novietojat peles rādītāju virs logrīka. Atkarībā no pārlūkprogrammas iestatījumiem attēls tiek automātiski saglabāts mapē `Downloads` vai arī tiek atvērts dialoglodziņš *Saglabāt*, kurā varat izvēlēties atrašanās vietu.

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 parādīti dati no atbilstošajiem vienumiem. Var izvēlēties vienu pamatkrāsu vai paletes rindu, lai katram atbilstošajam vienumam piešķirtu atšķirīgu krāsu.
- *Vienumu saraksts* - tiek parādīti dati no atlasītajiem vienumiem. Katram vienumam krāsu var izvēlēties atsevišķi krāsu atlasītājā.

Pēc noklusējuma tiek pievienota *Vienumu šabloni* datu kopa.

Datu kopa	**Vienumu šabloni** datu kopai: Atlasiet vai ievadiet hostu un vienumu šablonus; sektoru diagrammā tiks parādīti to vienumu dati, kas atbilst ievadītajiem šabloniem; var tikt parādīti ne vairāk kā 50 vienumi. Atlasīšanai var izmantot aizstājējzīmju šablonus (piemēram, ∗ atgriezīs rezultātus, kas atbilst nullei vai vairāk rakstzīmēm). Lai norādītu aizstājējzīmes šablonu, ievadiet virkni manuāli un nospiediet *Enter*. Aizstājējzīme vienmēr tiek interpretēta, tāpēc nav iespējams atsevišķi pievienot, piemēram, vienumu ar nosaukumu *item**, ja pastāv citi atbilstoši vienumi (piemēram, *item2*, *item3*). **Vienumu šabloni** datu kopām hosta un vienumu šablonu norādīšana ir obligāta. Skatiet arī: Datu kopas konfigurācijas informācija. **Vienumu saraksts** datu kopai: Atlasiet vienumus sektoru diagrammai, noklikšķinot uz pogas *Pievienot vienumu*. Nolaižamais saraksts *Tips* pēc vienuma nosaukuma ļauj atlasīt katra vienuma attēlošanas tipu: **Parasts** - vienuma vērtība sektoru diagrammā tiek attēlota proporcionāli (noklusējums); **Kopējais** - vienuma vērtība aizņem visu sektoru diagrammu. Ņemiet vērā, ka vienā sektoru diagrammā var būt tikai viens vienums ar tipu ”Kopējais”, un tas sektoru diagrammas leģendā tiks novietots pirmais. Ja vienumam ir iestatīts tips ”Kopējais”, parametrs *Datu kopas apkopošana* (skatiet tālāk) tiks atspējots un iestatīts uz ”netiek izmantots”. Varat arī atlasīt saderīgus logrīkus kā vienumu datu avotu, noklikšķinot uz pogas *Pievienot logrīku*. Uz logrīkiem attiecas tās pašas opcijas kā uz atsevišķiem vienumiem. **Vienumu saraksts** datu kopām vienumu vai logrīku norādīšana ir obligāta. Skatiet arī: Datu kopas konfigurācijas informācija. Ņemiet vērā, ka ir atļauti tikai skaitliski vienumu tipi. Konfigurējot logrīku veidnes informācijas panelī, parametrs hostu šablonu norādīšanai nav pieejams, bet parametrs vienumu saraksta norādīšanai ļauj atlasīt tikai veidnē konfigurētos vienumus.

Apkopošanas funkcija	Norādiet, kuru apkopošanas funkciju izmantot katram datu kopas vienumam atlasītajā *Laika periodā*: **min** - parādīt mazāko vērtību; **max** - parādīt lielāko vērtību; **avg** - parādīt vidējo vērtību; **sum** - parādīt vērtību summu; **count** - parādīt vērtību skaitu; **first** - parādīt pirmo vērtību; **last** - parādīt pēdējo vērtību (noklusējums). Apkopošana ļauj parādīt apkopotu vērtību norādītajā intervālā (5 minūtes, stunda, diena). Funkcijām *min*, *max*, *avg* un *sum* var parādīt tikai skaitliskus datus. Funkcijai *count* neskaitliskie dati tiks pārveidoti par skaitliskiem.
Datu kopas apkopošana	Norādiet, kuru apkopošanas funkciju izmantot visai datu kopai: **netiek izmantots** - apkopošana netiek veikta, vienumi tiek parādīti atsevišķi (noklusējums); **min** - parādīt mazāko vērtību; **max** - parādīt lielāko vērtību; **avg** - parādīt vidējo vērtību; **sum** - parādīt vērtību summu; **count** - parādīt vērtību skaitu. Apkopošana ļauj parādīt apkopotu vērtību norādītajā intervālā (5 minūtes, stunda, diena), kas atlasīts cilnē *Laika periods* vai izmantots visam informācijas panelim.
Datu kopas etiķete	Norādiet pielāgotu datu kopas etiķeti. Etiķete tiek parādīta datu kopas konfigurācijā un sektoru diagrammas leģendā (apkopotām datu kopām). Visas datu kopas tiek numurētas, tostarp tās, kurām ir norādīta *Datu kopas etiķete*. Ja etiķete nav norādīta, datu kopai automātiski tiek piešķirta etiķete atbilstoši tās numuram (piemēram, ”Datu kopa Nr. 2”, ”Datu kopa Nr. 3” u. c.). Datu kopu numerācija tiek pārrēķināta pēc datu kopu pārkārtošanas vai vilkšanas. Pārāk garas datu kopu etiķetes tiks saīsinātas, lai ietilptu attēlošanas vietā (piemēram, ”Procesu sk...”).
Vienumu tagi	Norādiet tagus, pēc kuriem filtrēt logrīkā parādītos vienumus. Var iekļaut vai izslēgt konkrētus tagus un tagu vērtības. Var iestatīt vairākus nosacījumus. Tagu nosaukumu atbilstība vienmēr ir reģistrjutīga. Katram nosacījumam ir pieejami vairāki operatori: **Pastāv** - iekļaut norādītos tagu nosaukumus; **Vienāds ar** - iekļaut norādītos tagu nosaukumus un vērtības (reģistrjutīgi); **Satur** - iekļaut norādītos tagu nosaukumus, kuru tagu vērtības satur ievadīto virkni (apakšvirknes atbilstība, nav reģistrjutīga); **Nepastāv** - izslēgt norādītos tagu nosaukumus; **Nav vienāds ar** - izslēgt norādītos tagu nosaukumus un vērtības (reģistrjutīgi); **Nesatur** - izslēgt norādītos tagu nosaukumus, kuru tagu vērtības satur ievadīto virkni (apakšvirknes atbilstība, nav reģistrjutīga). Nosacījumiem ir divi aprēķina veidi: **Un/Vai** - jāizpilda visi nosacījumi; nosacījumi ar vienādu taga nosaukumu tiks grupēti, izmantojot nosacījumu Vai; **Vai** - pietiek, ja tiek izpildīts viens nosacījums.

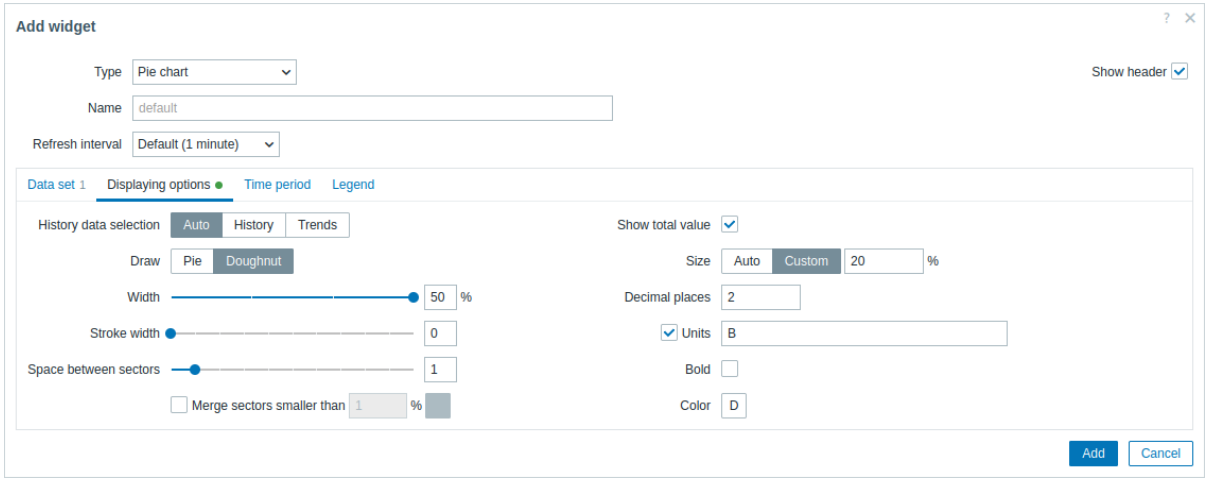
Datu kopas konfigurācijas informācija

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

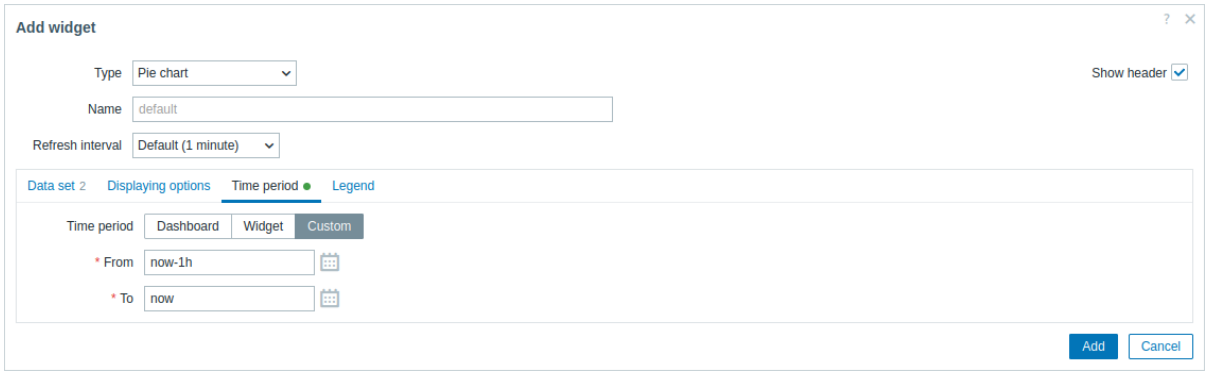
Cilne **Attēlošanas opcijas** ļauj definēt vēsturisko datu atlases un vizualizācijas opcijas sektoru diagrammai:



Vēsturisko datu atlase	Atlasiet datu avotu: **Automātiski** - dati tiek iegūti atbilstoši klasiskajam algoritmam (noklusējums); **Vēsture** - dati no vēstures; **Tendences** - dati no tendencēm.
Zīmēt	Atlasiet sektoru diagrammas vizualizācijas stilu: **Sektoru diagramma** - pilna sektoru diagramma (sektori aizņem 100% rādiusa); **Gredzenveida diagramma** - sektoru diagramma ar tukšu apli vidū (sektori aizņem līdz 50% rādiusa).
Atstarpe starp sektoriem	Atlasiet atstarpes lielumu (vienībās no 0 līdz 10) starp sektoriem (noklusējuma vērtība ir "1").
Apvienot sektorus, kas ir mazāki par N%	Atzīmējiet izvēles rūtiņu, lai apvienotu sektorus, kas ir mazāki par N%. Ja šī opcija ir iespējota, 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.
Līnijas platums	Atlasiet gredzenveida sektora apmales platumu (0-10).
Rādīt kopējo vērtību	Atzīmējiet izvēles rūtiņu, lai gredzenveida diagrammas vidū parādītu kopējo vērtību.
Izmērs	Atlasiet kopējās vērtības izmēra opciju: **Automātiski** - teksta izmērs tiek automātiski pielāgots, lai tas būtu salasāms gredzenveida diagrammas vidū; **Pielāgots** - norādiet teksta izmēru kā procentuālo augstumu no visa logrīka augstuma.
Decimāldaļas	Norādiet kopējās vērtības decimāldaļu skaitu (0-6).
Vienības	Norādiet kopējās vērtības mērvienības.
Treknraksts	Atzīmējiet izvēles rūtiņu, lai kopējo vērtību parādītu treknrakstā.
Krāsa	Atlasiet kopējās vērtības krāsu.

Laika periods

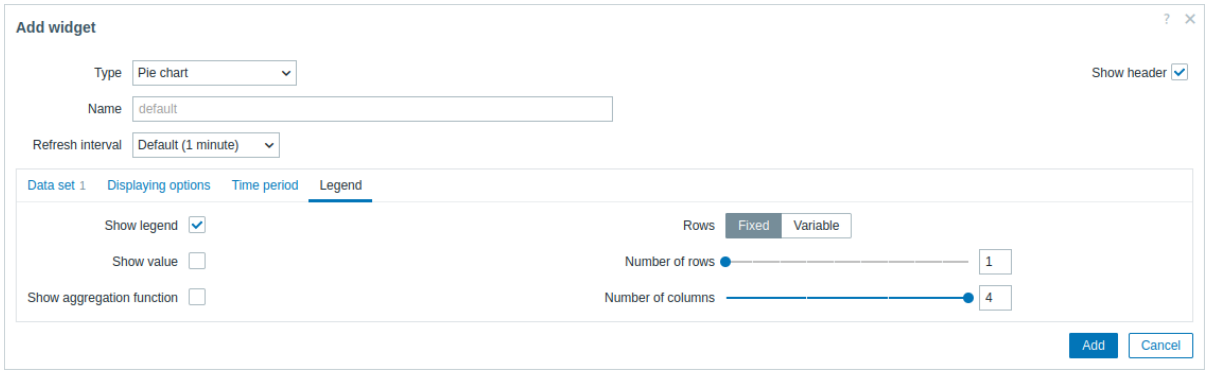
Cilne **Laika periods** ļauj iestatīt pielāgotu laika periodu sektoru diagrammas apkopošanas iestatījumiem:



Laika periods	Atlasiet datu avotu laika periodam: **Informācijas panelis** - izmantojiet informācijas paneļa laika perioda atlasītāju; **Logrīks** - izmantojiet saderīgu logrīku (iestatīts parametrā *Logrīks*); **Pielāgots** - izmantojiet pielāgotu laika periodu, kas norādīts parametros *No* un *Līdz*; ja tas ir iestatīts, logrīka augšējā labajā stūrī tiks parādīta pulksteņa ikona, kas, novietojot peles rādītāju 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 (*Grafiks*, *Grafiks (klasiskais)*, *Grafika prototips*) 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` 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:

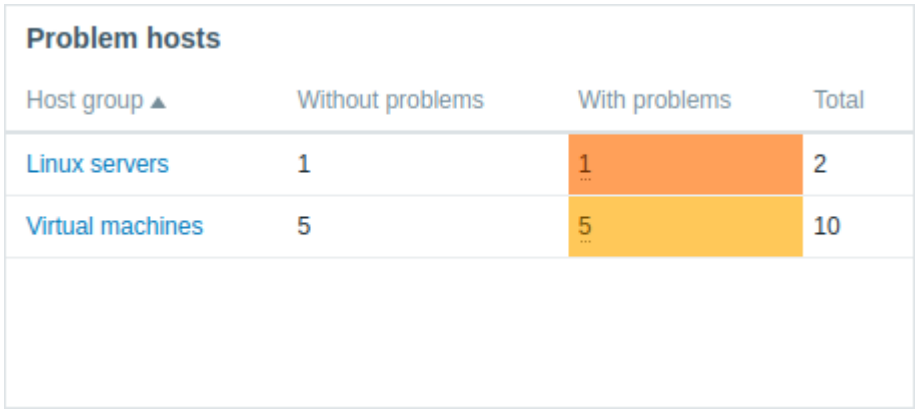


Rādīt leģendu	Noņemiet atzīmi no šīs izvēles rūtiņas, lai sektoru diagrammā paslēptu leģendu (pēc noklusējuma tā ir atzīmēta).
Rādīt vērtību	Atzīmējiet šo izvēles rūtiņu, lai leģendā rādītu vienuma vērtību.
Rādīt apkopošanas funkciju	Atzīmējiet šo izvēles rūtiņu, lai leģendā rādītu apkopošanas funkciju.
Rindas	Atlasiet leģendas rindu attēlošanas režīmu: **Fiksēts** - attēlojamo rindu skaitu nosaka parametra *Rindu skaits* vērtība; **Mainīgs** - attēlojamo rindu skaitu nosaka konfigurēto vienumu skaits, nepārsniedzot parametra *Maksimālais rindu skaits* vērtību.
Rindu skaits/ Maksimālais rindu skaits	Ja laukam *Rindas* ir iestatīta vērtība "Fiksēts", iestatiet attēlojamo leģendas rindu skaitu (1-10). Ja laukam *Rindas* ir iestatīta vērtība "Mainīgs", iestatiet maksimālo attēlojamo leģendas rindu skaitu (1-10).
Kolonnu skaits	Iestatiet attēlojamo leģendas kolonnu skaitu (1-4). Šis parametrs ir pieejams, ja izvēles rūtiņa *Rādīt vērtību* nav atzīmēta.

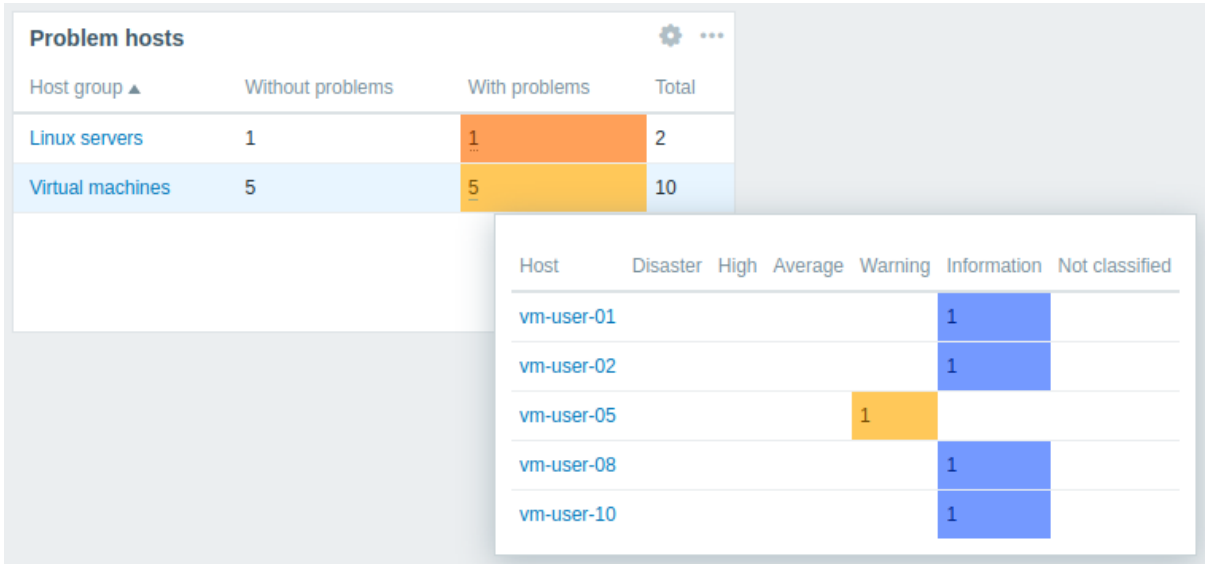
Problēmu hosts

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.).

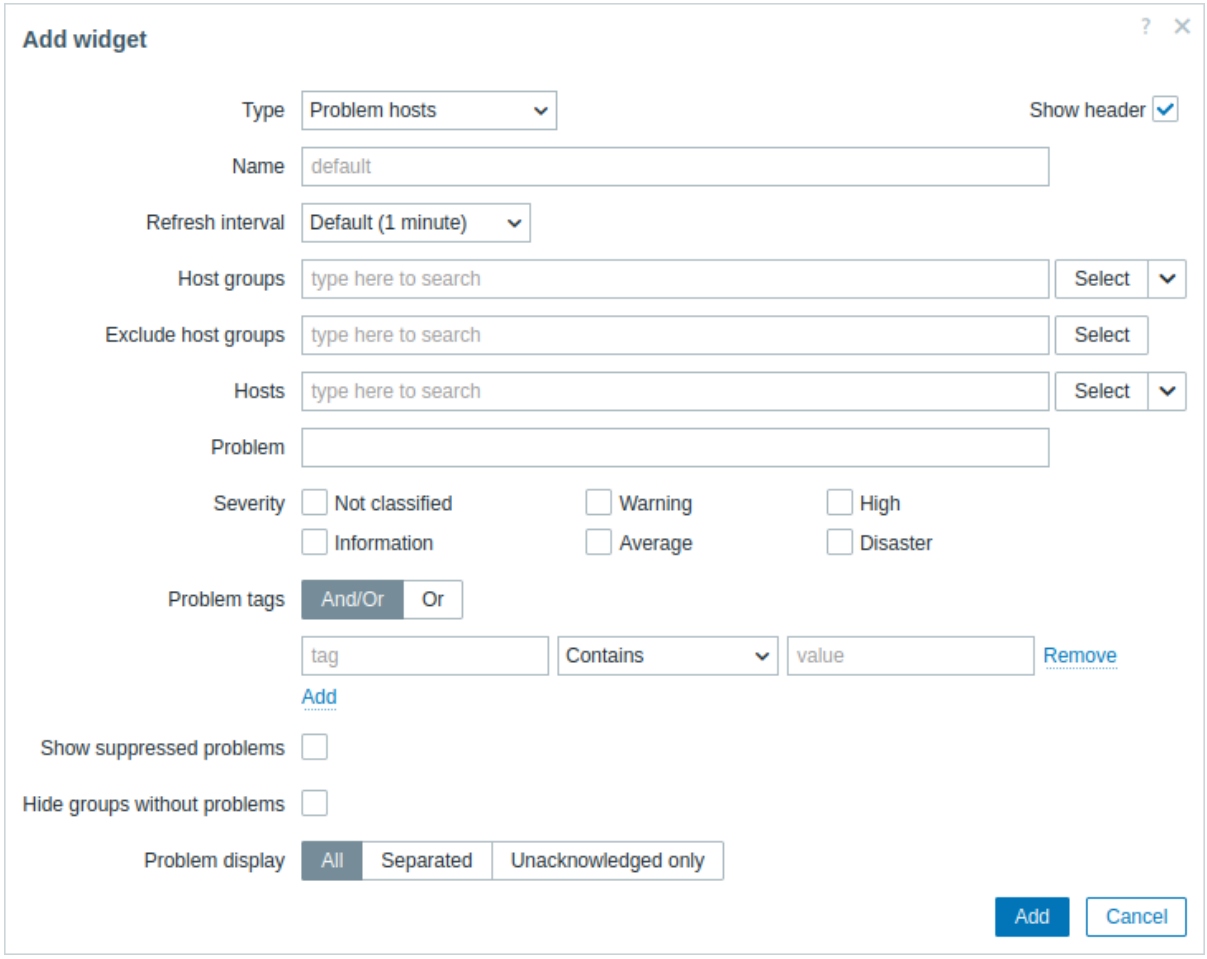


Novietojot kursoru virs problēmu skaita, tiek parādīts hostu ar problēmām saraksts:



Konfigurācija

Lai konfigurētu, kā tipu atlasiet *Problem hosts*:



Papildus parametriem, kas ir kopīgi visiem logrīkiem, varat iestatīt šādas specifiskas opcijas:

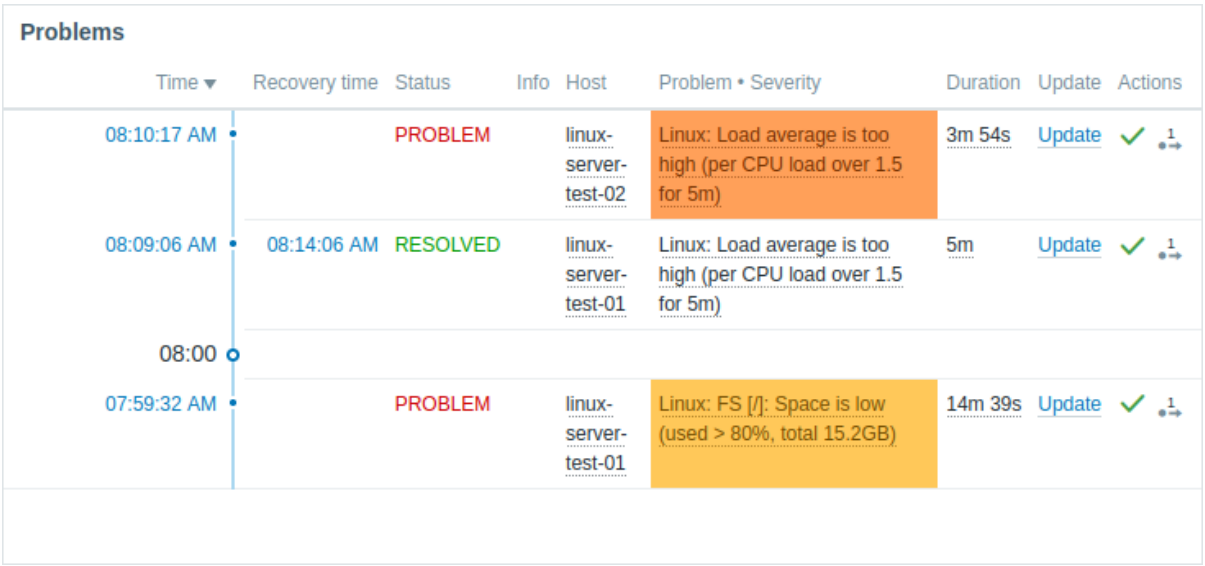
Hostu grupas	Atlasiet hostu grupas, ko attēlot logrīkā. Alternatīvi atlasiet saderīgu logrīku kā hostu grupu datu avotu. Šis lauks atbalsta automātisko pabeigšanu, tāpēc, sākot rakstīt grupas nosaukumu, tiks piedāvāts atbilstošo grupu nolaižamais saraksts. Norādot vecākgrupas hostu grupu, tiek netieši 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.

Problēmas

Pārskats

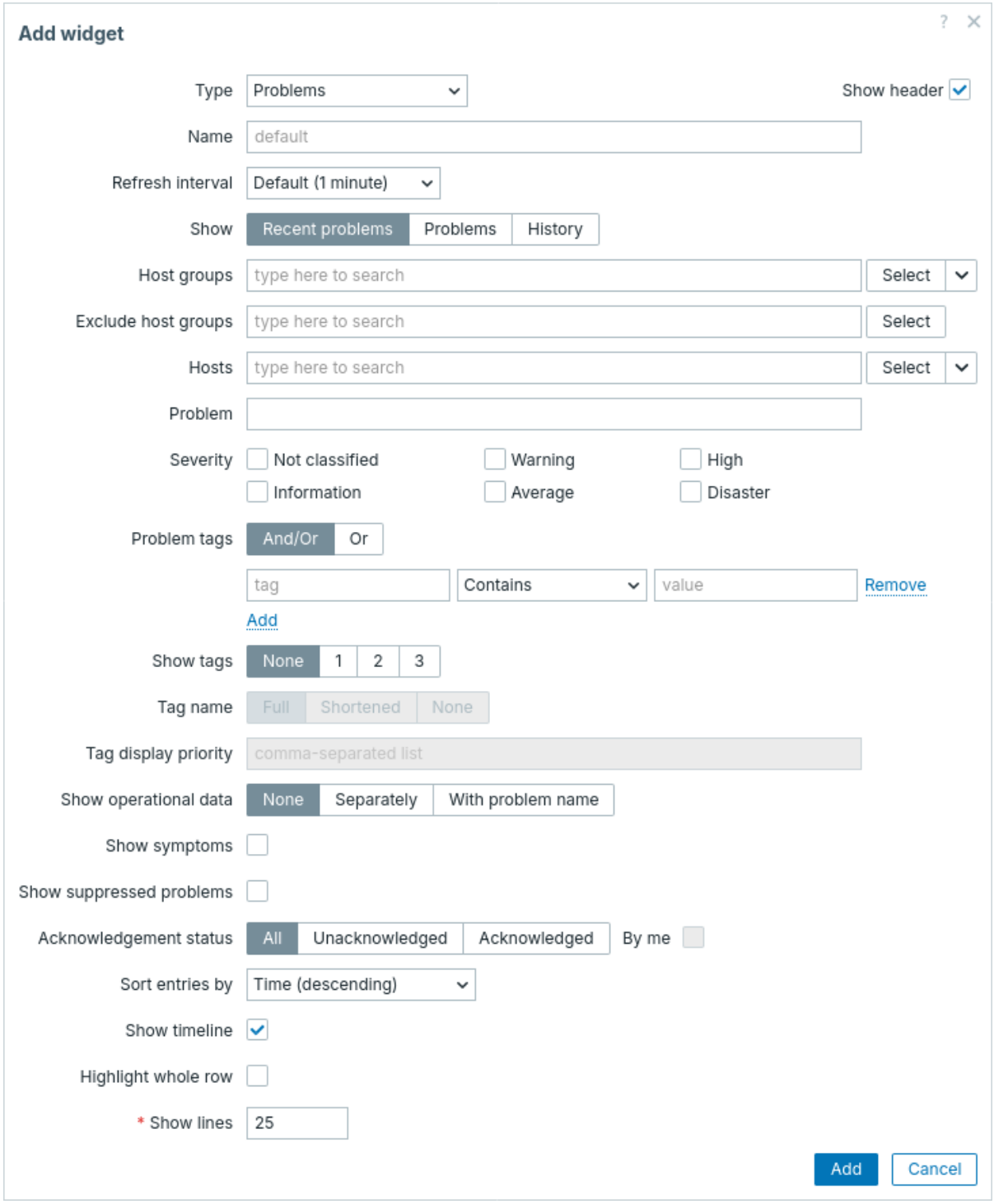
Logrīks *Problēmas* attēlo pašreizējās problēmas, kas filtrētas pēc dažādiem parametriem (hostu grupas, hosti, problēmu nosaukumi u.c.), sniedzot jums skaidru pārskatu par to, kam jāpievērš uzmanība.



Tas attēlo tos pašus datus kā *Uzraudzība* > *Problēmas* un var attēlot līdz 1000 ierakstiem.

Konfigurācija

Lai konfigurētu, izvēlieties *Problems* kā tipu:



Varat ierobežot, cik daudz problēmu tiek parādīts logrīkā, dažādos veidos - pēc problēmas statusa, problēmas nosaukuma, smaguma pakāpes, hostu grupas, hosta, notikuma taga, apstiprinājuma statusa utt.

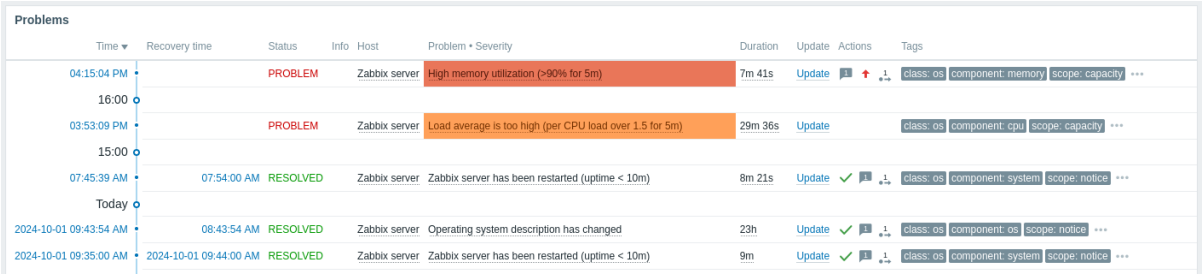
Papildus parametriem, kas ir kopīgi visiem logrīkiem, varat iestatīt šādas specifiskas opcijas:

Show	Filtrēt pēc problēmas statusa: **Recent problems** - tiek parādītas neatrisinātas un nesen atrisinātas problēmas (noklusējums); **Problems** - tiek parādītas neatrisinātas problēmas; **History** - tiek parādīta visu notikumu vēsture.
Host groups	Atlasiet hostu grupas, kuru problēmas rādīt logrīkā. Alternatīvi atlasiet saderīgu logrīku kā datu avotu hostu grupām. Šis lauks atbalsta automātisko pabeigšanu, tāpēc, sākot rakstīt grupas nosaukumu, tiks piedāvāts nolaižamais saraksts ar atbilstošajām grupām. Norādot vecāko hostu grupu, netieši tiek atlasītas visas ligzdotās hostu grupas. Šo hostu grupu problēmas tiks parādītas logrīkā; ja hostu grupas nav norādītas, tiks parādītas visu hostu grupu problēmas. Šis parametrs nav pieejams, konfigurējot logrīku veidnes informācijas panelī.

Exclude host groups	Atlasiet hostu grupas, kuru problēmas paslēpt no logrīka. Šis lauks atbalsta automātisko pabeigšanu, tāpēc, sākot rakstīt grupas nosaukumu, tiks piedāvāts nolaižamais saraksts ar atbilstošajām grupām. Norādot vecāko hostu grupu, netieši tiek atlasītas visas ligzdotās hostu grupas. Šo hostu grupu problēmas netiks parādītas. Piemēram, hosti 001, 002, 003 var būt grupā A, bet hosti 002, 003 - arī grupā B. Ja vienlaikus izvēlamies *show* grupu A un *exclude* grupu B, logrīkā tiks parādītas tikai hosta 001 problēmas. Šis parametrs nav pieejams, konfigurējot logrīku veidnes informācijas panelī.
Hosts	Atlasiet hostus, kuru problēmas rādīt logrīkā. Alternatīvi atlasiet saderīgu logrīku vai informācijas paneli kā datu avotu hostiem. Šis lauks atbalsta automātisko pabeigšanu, tāpēc, sākot rakstīt hosta nosaukumu, tiks piedāvāts nolaižamais saraksts ar atbilstošajiem hostiem. Ja hosti nav norādīti, tiks parādītas visu hostu problēmas. Šis parametrs nav pieejams, konfigurējot logrīku veidnes informācijas panelī.
Problem	Varat ierobežot parādīto problēmu skaitu pēc to nosaukuma. Ja šeit ievadāt virkni, tiks parādītas tikai tās problēmas, kuru nosaukumā ir ietverta ievadītā virkne. Makro netiek izvērsti.
Severity	Atzīmējiet problēmu smaguma pakāpes, lai filtrētu logrīkā parādāmās problēmas. Ja neviena smaguma pakāpe nav atzīmēta, tiks parādītas visas problēmas.
Problem tags	Norādiet problēmu tagus, lai ierobežotu logrīkā parādīto problēmu skaitu. Ir iespējams iekļaut, kā arī izslēgt konkrētus tagus un tagu vērtības. Var iestatīt vairākus nosacījumus. Tagu nosaukumu atbilstība vienmēr ir reģistrjutīga. Katrai nosacījuma rindai 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, nav reģistrjutīga); **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, nav reģistrjutīga). 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. Filtrēšanas gadījumā šeit norādītie tagi tiks parādīti pirmie kopā ar problēmu, ja vien tos nepārraksta saraksts *Tag display priority* (skatiet tālāk).
Show tags	Atlasiet parādāmo tagu skaitu: **None** - 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 parādītas pilnībā; **Shortened** - tagu nosaukumi tiek saīsināti līdz 3 simboliem, bet tagu vērtības tiek parādītas pilnībā; **None** - tiek parādītas tikai tagu vērtības; nosaukumi netiek rādīti.
Tag display priority	Ievadiet problēmas tagu attēlošanas prioritāti kā ar komatiem atdalītu tagu sarakstu. Jāizmanto tikai tagu nosaukumi, nevis vērtības. Piemērs: `Services,Applications,Application` Šī saraksta tagi vienmēr tiks parādīti pirmie, pārrakstot dabisko kārtojumu alfabēta secībā.
Show operational data	Atlasiet operatīvo datu attēlošanas režīmu: **None** - operatīvie dati netiek rādīti; **Separately** - operatīvie dati tiek rādīti atsevišķā kolonnā; **With problem name** - operatīvie dati tiek pievienoti problēmas nosaukumam, izmantojot iekavas operatīvajiem datiem. Sīkāku informāciju skatiet lauka *Operational data* aprakstā trigeru konfigurācijā.

Show symptoms	Atzīmējiet izvēles rūtiņu, lai savā rindā parādītu problēmas, kas klasificētas kā simptomi.
Show suppressed problems	Atzīmējiet izvēles rūtiņu, lai parādītu problēmas, kas citādi tiktu slēptas (nerādītas) hosta uzturēšanas vai atsevišķas problēmu slāpēšanas dēļ.
Acknowledgement status	Filtrējiet, lai parādītu visas problēmas, tikai neapstiprinātās problēmas vai tikai apstiprinātās problēmas. Atzīmējiet papildu izvēles rūtiņu, lai filtrētu tās problēmas, kuras jebkad esat apstiprinājis jūs.
Sort entries by	Kārtot ierakstus pēc: **Time** (augošā vai dilstošā secībā); **Severity** (augošā vai dilstošā secībā); **Problem name** (augošā vai dilstošā secībā); **Host** (augošā vai dilstošā secībā). Ierakstu kārtošana pēc **Host** (augošā vai dilstošā secībā) nav pieejama, konfigurējot logrīku veidnes informācijas panelī.
Show timeline	Atzīmējiet izvēles rūtiņu, lai parādītu vizuālu laika skalu.
Highlight whole row	Atzīmējiet izvēles rūtiņu, lai izceltu visu rindu neatrisinātām problēmām. Izcelšanai tiek izmantota problēmas smaguma pakāpes krāsa. *Highlight whole row* nav pieejams augsta kontrasta tēmās.
Show lines	Norādiet parādāmo problēmu rindu skaitu.

Darbs ar logrīku

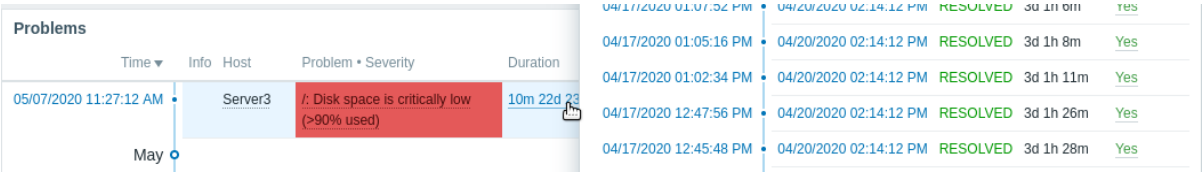


Logrīks Problems nodrošina ātru piekļuvi papildu informācijai:

- Noklikšķiniet uz problēmas datuma (kolonnā Time) vai atjaunošanas datuma (kolonnā Recovery Time), lai skatītu notikuma informāciju.
- Ja Info lauks nav tukšs, novietojiet kursoru virs parādītās ikonas, lai skatītu papildu informāciju.
- Noklikšķiniet uz hosts nosaukuma, lai atvērtu hosts izvēlni.
- Noklikšķiniet uz problēmas nosaukuma, lai atvērtu notikuma izvēlni.
- Novietojiet kursoru virs problēmas ilguma vai noklikšķiniet uz tā, lai skatītu problēmas notikuma uznirstošo logu.
- Noklikšķinot uz *Update*, tiek atvērts Problēmas atjaunināšana logs.
- Novietojiet kursoru virs pelēkās bultiņas ikonas kolonnā Actions vai noklikšķiniet uz tās, lai skatītu izpildīto darbību sarakstu.

Problēmas notikuma uznirstošais logs

Problēmas notikuma uznirstošais logs ietver šī trigeris problēmu notikumu sarakstu un, ja tas ir definēts, arī trigeris aprakstu un noklikšķināmu URL.



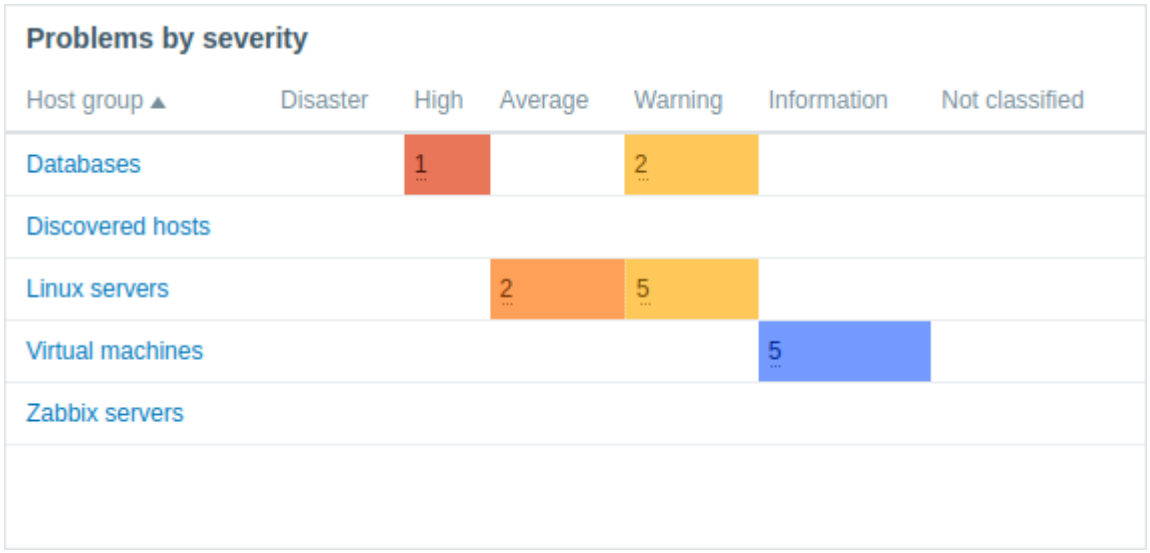
Lai atvērtu problēmas notikuma uznirstošo logu:

- Novietojiet peli virs problēmas ilguma *Problems* logrīka kolonnā *Duration*. Uznirstošais logs pazūd, tiklīdz noņemat peli no ilguma.
- Noklikšķiniet uz ilguma *Problems* logrīka kolonnā *Duration*. Uznirstošais logs pazūd tikai tad, ja vēlreiz noklikšķināt uz ilguma.

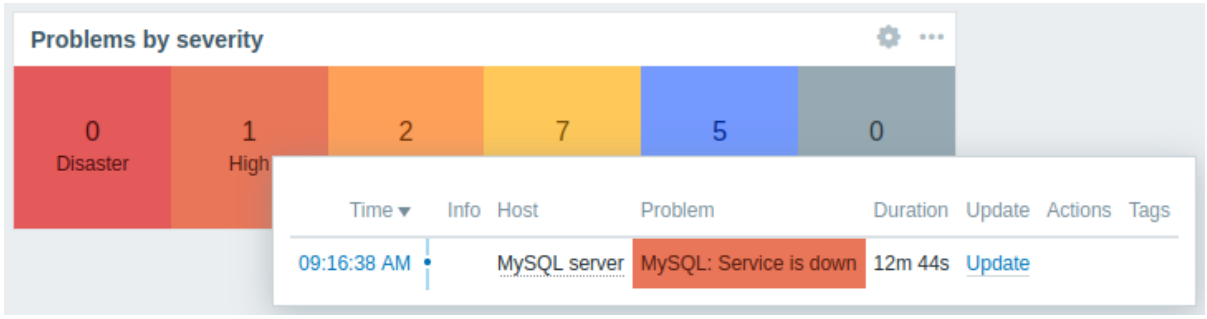
Problēmas pēc nopietnības

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.).

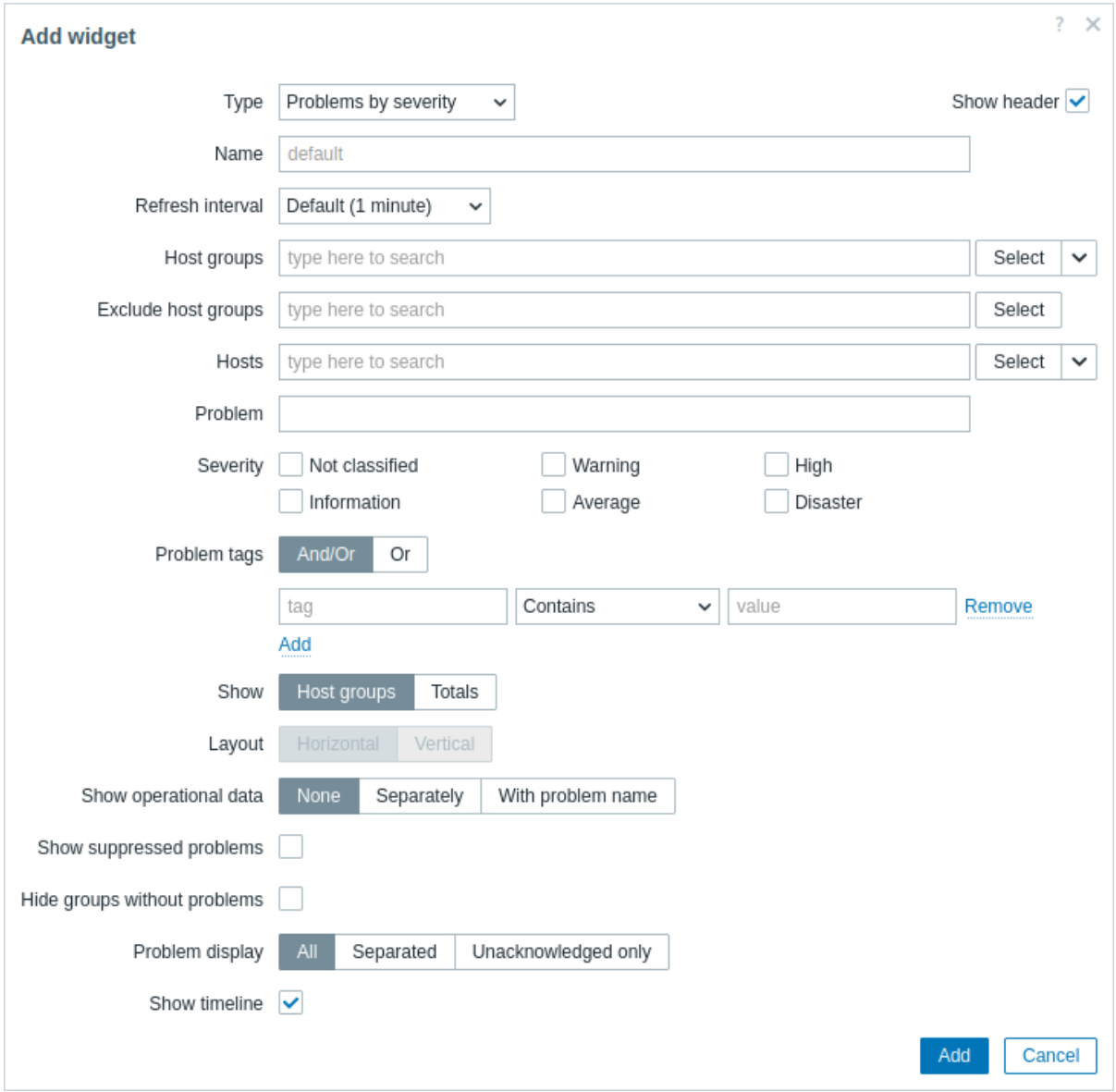


Novietojot kursoru virs problēmu skaita, tiek parādīts saistīto problēmu saraksts:



Konfigurācija

Lai konfigurētu, atlasiet *Problems by severity* kā tipu:



Papildus parametriem, kas ir kopīgi visiem logrīkiem, varat iestatīt šādas specifiskās opcijas:

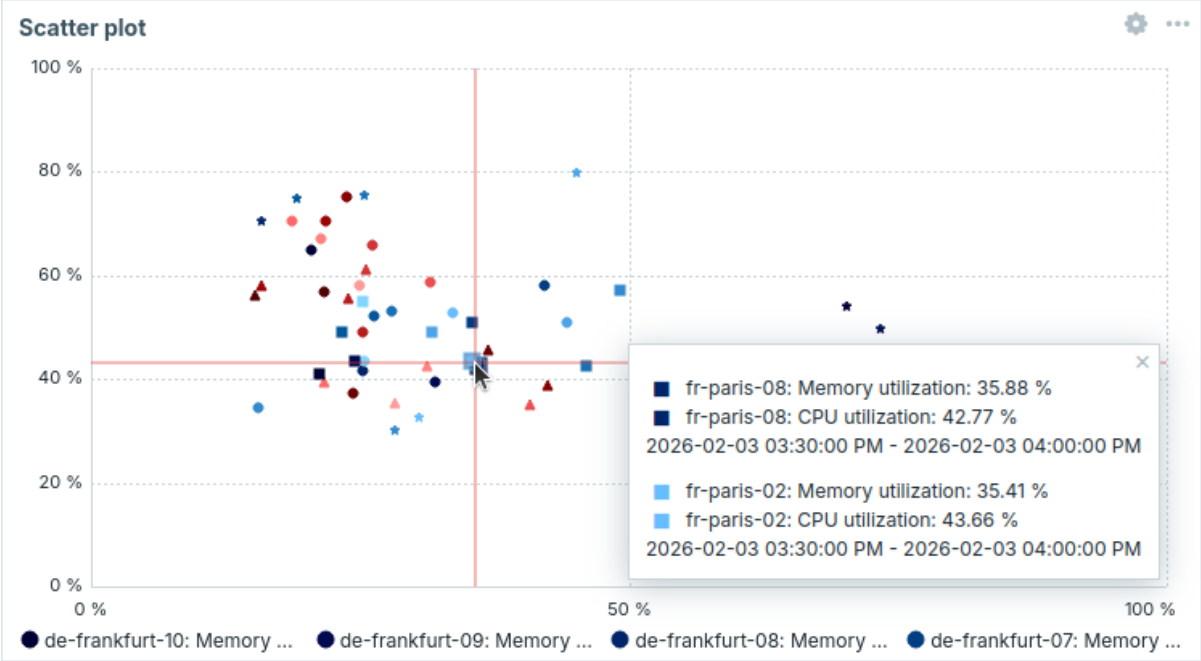
Host groups	Atlasiet hostu grupas, ko rādīt logrīkā. Alternatīvi atlasiet saderīgu logrīku kā datu avotu hostu grupām. Šis lauks atbalsta automātisko pabeigšanu, tāpēc, sākot rakstīt grupas nosaukumu, tiks piedāvāts nolaižamais saraksts ar atbilstošajām grupām. Norādot vecāko hostu grupu, automātiski tiek atlasītas arī visas ligzdotās hostu grupas. Šo hostu grupu hostu dati tiks rādīti logrīkā; ja hostu grupas netiks norādītas, tiks rādītas visas hostu grupas. Šis parametrs nav pieejams, konfigurējot logrīku veidnes informācijas panelī.
Exclude host groups	Atlasiet hostu grupas, ko paslēpt no logrīka. Šis lauks atbalsta automātisko pabeigšanu, tāpēc, sākot rakstīt grupas nosaukumu, tiks piedāvāts nolaižamais saraksts ar atbilstošajām grupām. Norādot vecāko hostu grupu, automātiski tiek atlasītas arī visas ligzdotās hostu grupas. Šo hostu grupu hostu dati netiks rādīti logrīkā. Piemēram, hosti 001, 002, 003 var būt grupā A, bet hosti 002, 003 arī grupā B. Ja vienlaikus izvēlamies *rādīt* grupu A un *izslēgt* grupu B, informācijas panelī tiks rādīti tikai hosta 001 dati. Šis parametrs nav pieejams, konfigurējot logrīku veidnes informācijas panelī.
Hosts	Atlasiet hostus, ko rādīt logrīkā. Alternatīvi atlasiet saderīgu logrīku vai informācijas paneli kā datu avotu hostiem. Šis lauks atbalsta automātisko pabeigšanu, tāpēc, sākot rakstīt hosta nosaukumu, tiks piedāvāts nolaižamais saraksts ar atbilstošajiem hostiem. Ja hosti netiek norādīti, tiks rādīti visi hosti. Šis parametrs nav pieejams, konfigurējot logrīku veidnes informācijas panelī.

Problem	Varat ierobežot rādīto problemātisko hostu skaitu pēc problēmas nosaukuma. Ja šeit ievadāt virkni, tiks rādīti tikai tie hosti, kuru problēmu nosaukumos ir ietverta ievadītā virkne. Makro netiek izvērsti.
Severity	Atzīmējiet problēmu smaguma pakāpes, lai filtrētu logrīkā rādāmās problēmas. Ja neviena smaguma pakāpe nav atzīmēta, tiks rādītas visas problēmas.
Problem tags	Norādiet problēmu tagus, lai ierobežotu logrīkā rādīto problēmu skaitu. Ir iespējams iekļaut, kā arī izslēgt konkrētus tagus un tagu vērtības. Var iestatīt vairākus nosacījumus. Tagu nosaukumu salīdzināšana 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, nereģistrjutīgi); **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, nereģistrjutīgi). 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.
Show	Atlasiet rādīšanas opciju: **Host groups** - rādīt problēmas pa hostu grupām; **Totals** - rādīt visu atlasīto hostu grupu problēmu kopskaitu krāsainos blokos, kas atbilst problēmas smaguma pakāpei. Šis parametrs nav pieejams, konfigurējot logrīku veidnes informācijas panelī, un tiks rādīts tikai problēmu kopskaits.
Layout	Atlasiet izkārtojuma opciju: **Horizontal** - krāsainie kopskaita bloki tiks rādīti horizontāli; **Vertical** - krāsainie kopskaita bloki tiks rādīti vertikāli. Šis parametrs ir pieejams, ja *Show* ir iestatīts uz ”Totals”.
Show operational data	Rādīt operatīvos datus kā: **None** - operatīvie dati netiks rādīti **Separately** - operatīvie dati tiks rādīti kā atsevišķa rinda **With problem name** - operatīvie dati tiks rādīti kopā ar problēmas nosaukumu. Sīkāku informāciju skatiet lauka *Operational data* aprakstā trigeru konfigurācijā.
Show suppressed problems	Atzīmējiet izvēles rūtiņu, lai rādītu problēmas, kas citādi tiktu slāpētas (nerādītas) hosta uzturēšanas dēļ.
Hide groups without problems	Atzīmējiet opciju *Hide groups without problems*, lai logrīkā paslēptu datus no hostu grupām bez problēmām. Šis parametrs nav pieejams, konfigurējot logrīku veidnes informācijas panelī.
Problem display	Rādīt problēmu skaitu kā: **All** - tiks rādīts pilns problēmu skaits; **Separated** - tiks rādīts neapstiprināto problēmu skaits, atsevišķi norādot to kā daļu no kopējā problēmu skaita; **Unacknowledged only** - tiks rādīts tikai neapstiprināto problēmu skaits.
Show timeline	Atzīmējiet izvēles rūtiņu, lai rādītu vizuālu laika skalu.

Izkliedes diagramma

Pārskats

Izkliedes diagrammas logrīks attēlo saistību starp diviem rādītājiem, uz X un Y ass attēlojot atsevišķus datu punktus. Tas palīdz atklāt datu kopas modeļus, grupējumus, korelācijas un novirzes.



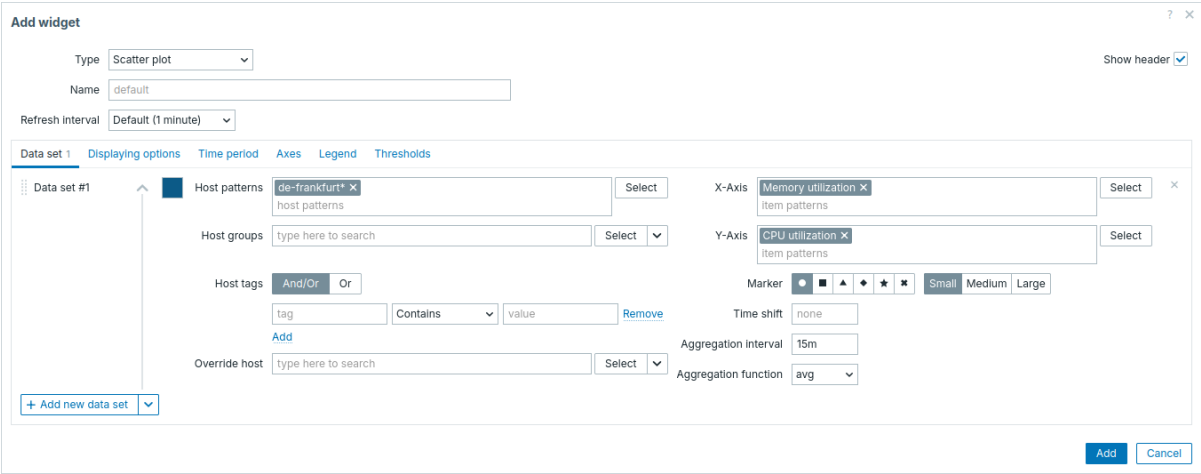
Novietojot kursoru virs datu punkta, tiek parādīts rīka padoms ar šī datu punkta vienumu sarakstu un citiem punktiem 6 pikseļu rādiusā; vispirms tie tiek kārtoti pēc izkliedes diagrammas X ass vērtībām augošā secībā un pēc tam pēc Y ass vērtībām dilstošā secībā.

Atlasot vienumu rīka padomā, tā dati tiek pārraidīti citiem logrīkiem.

Izkliedes diagrammas logrīkā attēloto informāciju var lejupielādēt kā PNG attēlu, logrīka izvēlnē *Darbības* atlasot opciju *Lejupielādēt attēlu* ..., kas tiek parādīta, novietojot kursoru virs logrīka. Atkarībā no pārlūkprogrammas iestatījumiem attēls tiek automātiski saglabāts mapē `Downloads` vai arī tiek atvērts dialoglodziņš *Saglabāt*, kurā var izvēlēties saglabāšanas vietu.

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 raksti* - hosts ar atbilstošiem vienumiem tiek parādīti kā datu punkti. Varat izvēlēties vienu pamatkrāsu vai paletes rindu, lai katram hostam ar atbilstošajiem vienumiem piešķirtu atšķirīgu krāsu.
- *Vienumu saraksts* - atlasītie vienumi tiek parādīti kā datu punkti, kur katrs punkts attēlo vienumu pāri (X un Y). Katras datu kopas krāsu varat izvēlēties atsevišķi krāsu atlasītājā.

Datu punkti tiek izveidoti tikai tiem hostu/vienumu pāriem, kuriem abās asīs ir vismaz viena vērtība.

Pēc noklusējuma tiek pievienota *Vienumu raksti* datu kopa.

Datu kopa	**Vienumu raksti** datu kopai: Atlasiet vai ievadiet hostu un vienumu rakstus; to vienumu dati, kuri atbilst šiem rakstiem, tiks parādīti izkliedes diagrammā. Ņemiet vērā: ja datu punktu skaits pārsniedz 8 tūkstošus, `php.ini` konfigurācijas failā jāpalielina `memory_limit`. Ieteicamais atmiņas ierobežojums 20 tūkstošiem datu punktu ir 512 MB. Atlasīšanai var izmantot aizstājējzīmju rakstus (piemēram, * atgriezīs rezultātus, kas atbilst nullei vai vairākiem simboliem). Lai norādītu aizstājējzīmes rakstu, ievadiet virkni manuāli un nospiediet *Enter*. Aizstājējzīmes simbols vienmēr tiek interpretēts, tāpēc nav iespējams atsevišķi pievienot, piemēram, vienumu ar nosaukumu *item**, ja ir arī citi atbilstoši vienumi (piemēram, *item2*, *item3*). Hostu un vienumu rakstu norādīšana ir obligāta "Vienumu raksti" datu kopām. Skatiet arī: Datu kopas konfigurācijas informācija. **Vienumu saraksts** datu kopai: Atlasiet vienumus izkliedes diagrammai, noklikšķinot uz pogas *Pievienot vienumu*. Varat arī atlasīt saderīgus logrīkus kā datu avotu vienumiem, noklikšķinot uz pogas *Pievienot logrīku*. Vienumu vai logrīku norādīšana ir obligāta "Vienumu saraksts" datu kopām. Skatiet arī: Datu kopas konfigurācijas informācija. Ņemiet vērā, ka ir atļauti tikai skaitlisku tipu vienumi. Konfigurējot logrīku veidnes informācijas panelī, parametrs hostu rakstu norādīšanai nav pieejams, bet parametrs vienumu saraksta norādīšanai ļauj atlasīt tikai veidnē konfigurētos vienumus.
Hostu grupas	Atlasiet hostu grupas. Varat arī atlasīt saderīgu logrīku kā datu avotu hostu grupām. Šis lauks atbalsta automātisko pabeigšanu, tāpēc, sākot rakstīt grupas nosaukumu, tiks piedāvāts atbilstošo grupu nolaižamais saraksts. Norādot vecāka hostu grupu, netieši tiek atlasītas visas ligzdotās hostu grupas. Hostu dati no šīm hostu grupām tiks parādīti logrīkā; ja hostu grupas nav norādītas, tiks parādītas visas hostu grupas. Šis parametrs ir pieejams tikai *Vienumu raksti* datu kopām un nav pieejams, konfigurējot logrīku veidnes informācijas panelī.
Hostu tagi	Norādiet tagus, lai filtrētu logrīkā parādītos hostus. Varat iekļaut vai izslēgt konkrētus tagus un tagu vērtības. Var iestatīt vairākus nosacījumus. Tagu nosaukumu atbilstība vienmēr ir reģistrjutīga. Katram nosacījumam ir pieejami vairāki operatori: **Pastāv** - iekļaut norādītos tagu nosaukumus; **Vienāds ar** - iekļaut norādītos tagu nosaukumus un vērtības (reģistrjutīgi); **Satur** - iekļaut norādītos tagu nosaukumus, kuru tagu vērtības satur ievadīto virkni (apakšvirknes atbilstība, nav reģistrjutīga); **Nepastāv** - izslēgt norādītos tagu nosaukumus; **Nav vienāds ar** - izslēgt norādītos tagu nosaukumus un vērtības (reģistrjutīgi); **Nesatur** - izslēgt norādītos tagu nosaukumus, kuru tagu vērtības satur ievadīto virkni (apakšvirknes atbilstība, nav reģistrjutīga). Nosacījumiem ir divi aprēķina veidi: **Un/Vai** - jāizpilda visi nosacījumi; nosacījumi ar vienādu taga nosaukumu tiks grupēti, izmantojot nosacījumu Vai; **Vai** - pietiek, ja tiek izpildīts viens nosacījums. Šis parametrs ir pieejams tikai *Vienumu raksti* datu kopām un nav pieejams, konfigurējot logrīku veidnes informācijas panelī.
Ignorēt hostu	Atlasiet saderīgu logrīku vai informācijas paneļa hostu atlasītāju kā datu avotu hostiem. Šis parametrs nav pieejams, konfigurējot logrīku veidnes informācijas 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.

Apkopošanas funkcija	Norādiet, kuru apkopošanas funkciju izmantot katram datu kopas vienumam atlasītajā *Apkopošanas intervālā*: **min** - parādīt mazāko vērtību; **max** - parādīt lielāko vērtību; **avg** - parādīt vidējo vērtību; **sum** - parādīt vērtību summu; **count** - parādīt vērtību skaitu (ja abās asīs pastāv vismaz viena vērtība); **first** - parādīt pirmo vērtību; **last** - parādīt pēdējo vērtību. Apkopošana ļauj norādītajā intervālā (5 minūtēs, stundā, dienā) parādīt apkopotu vērtību visu vērtību vietā. *min*, *max*, *avg* un *sum* var parādīt tikai skaitliskus datus. Izmantojot *count*, neskaitliskie dati tiks pārveidoti par skaitliskiem.
Apkopošanas intervāls	Norādiet intervālu vērtību apkopošanai. Šajā laukā varat izmantot laika sufiksus. Skaitliska vērtība bez sufiksa tiks uzskatīta par sekundēm. Ņemiet vērā: ja logrīks ir konfigurēts vēsturisko datu attēlošanai, pamatojoties uz tendencēm (*Vēstures datu atlase* ir iestatīta uz *Tendences* vai *Automātiski*), ieteicams izmantot apkopošanas intervālu, kas ir 1 stundas daudzkārtnis (piemēram, 3600, 60m, 1h, 3h u. c.). Tendencēs tiek glabātas stundas apkopotās vērtības, tāpēc apkopošanas intervāls, kas nav 1 stundas daudzkārtnis (piemēram, 100s, 7min, 15min, 90min u. c.), var radīt grūti interpretējamus rezultātus. Apkopošanas intervāla norādīšana ir obligāta. Apkopošanas intervāls nosaka datu punkta laika intervālu un koordinātas; piemēram, ja X ass vienumiem ir 4 vērtības, bet Y ass vienumiem ir 1 vērtība, koordinātas atspoguļo 4 vērtību apkopošanu uz X ass un 1 vērtības apkopošanu uz Y ass norādītajā intervālā.

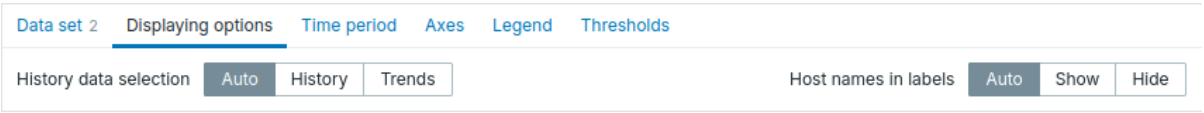
Datu kopas konfigurācijas informācija

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

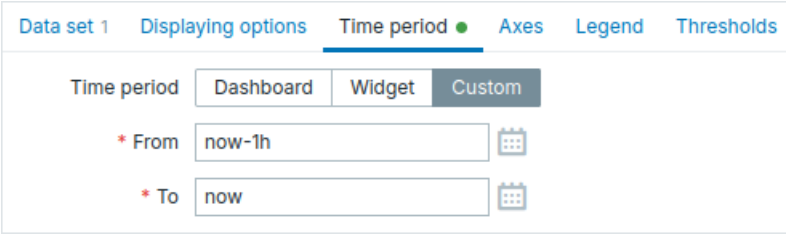
Cilne **Attēlošanas opcijas** ļauj definēt vēsturisko datu atlasi:



Vēsturisko datu atlase	Iestatiet izkliedes diagrammas datu avotu: **Automātiski** - dati tiek iegūti saskaņā ar klasiskās diagrammas algoritmu (noklusējums). **Vēsture** - dati no vēstures. **Tendences** - dati no tendencēm.
Hostu nosaukumi etiķetēs	Izvēlieties, vai diagrammas leģendā un rīka padomā, kas tiek parādīts, novietojot kursoru virs diagrammas datiem, rādīt hostu nosaukumus: **Automātiski** - hostu nosaukumi tiek rādīti tikai tad, ja datu kopās ir vairāk nekā viens hosts (noklusējums). **Rādīt** - hostu nosaukumi tiek rādīti. **Paslēpt** - hostu nosaukumi tiek paslēpti. Šis parametrs nav pieejams, konfigurējot logrīku veidnes informācijas panelī.

Laika periods

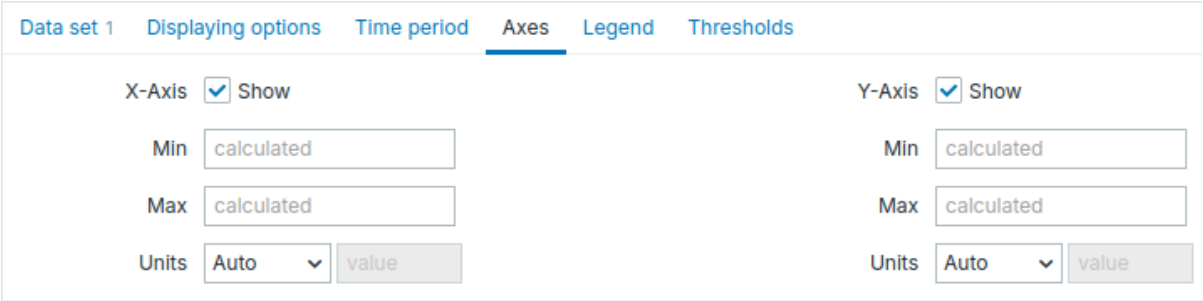
Cilne **Laika periods** ļauj iestatīt laika periodu, par kuru rādīt datus izkliedes diagrammā:



Laika periods	Atlasiet datu avotu laika periodam: **Informācijas panelis** - izmantojiet informācijas paneļa laika perioda atlasītāju; **Logrīks** - izmantojiet saderīgu logrīku (iestatīts parametrā *Logrīks*); **Pielāgots** - izmantojiet pielāgotu laika periodu, kas iestatīts parametros *No* un *Līdz*; ja tas ir iestatīts, logrīka augšējā labajā stūrī tiks parādīta pulksteņa ikona, kas, novietojot peles rādītāju virs tās, norāda 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` 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

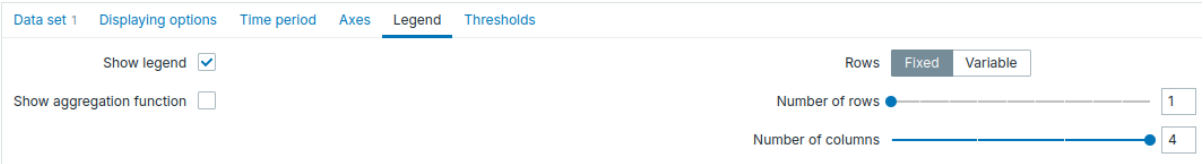
Cilne **Asis** ļauj pielāgot asu attēlošanu:



X ass/Y ass	Noņemiet atzīmi no šīs izvēles rūtiņas, lai paslēptu X asi/Y asi (pēc noklusējuma atzīmēta).
Min	Iestatiet atbilstošās ass minimālo vērtību.
Max	Iestatiet atbilstošās ass maksimālo vērtību.
Vienības	Nolaižamajā sarakstā izvēlieties izkliedes diagrammas asu vērtību vienību: **Automātiski** - asu vērtības tiek attēlotas, izmantojot datu kopas pirmā vienuma vienību; **Statiski** - asu vērtības tiek attēlotas, izmantojot vienību, kas norādīta ievades laukā *value*; ja lauks ir atstāts tukšs, tiek attēlotas tikai skaitliskās vērtības.

Leģenda

Cilne **Leģenda** ļauj pielāgot izkliedes diagrammas leģendu:

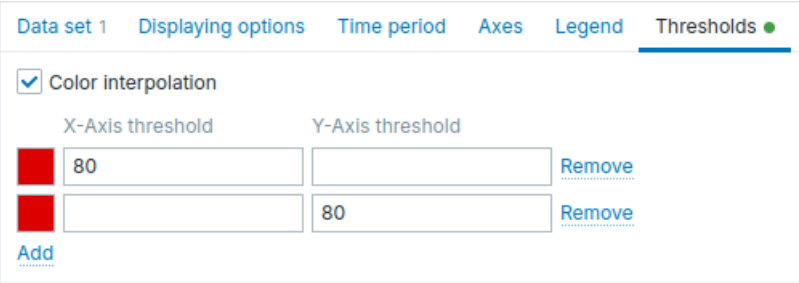


Rādīt leģendu	Noņemiet atzīmi no šīs izvēles rūtiņas, lai paslēptu izkliedes diagrammas leģendu (pēc noklusējuma tā ir atzīmēta).
Rādīt agregācijas funkciju	Atzīmējiet šo izvēles rūtiņu, lai leģendā rādītu agregācijas funkciju.

Rindas	Atlasiet leģendas rindu attēlošanas režīmu: **Fiksēts** - attēlojamo rindu skaitu nosaka parametra *Rindu skaits* vērtība; **Mainīgs** - attēlojamo rindu skaitu nosaka konfigurēto vienumu skaits, nepārsniedzot parametra *Maksimālais rindu skaits* vērtību.
Rindu skaits/ Maksimālais rindu skaits	Ja parametram *Rindas* ir iestatīta vērtība ”Fiksēts”, iestatiet attēlojamo leģendas rindu skaitu (1-10). Ja parametram *Rindas* ir iestatīta vērtība ”Mainīgs”, iestatiet maksimālo attēlojamo leģendas rindu skaitu (1-10).
Kolonnu skaits	Iestatiet attēlojamo leģendas kolonnu skaitu (1-4). Šis parametrs ir pieejams, ja opcija *Rādīt min./vid./maks.* nav atzīmēta.

Sliekšņi

Cilne **Sliekšņi** ļauj konfigurēt sliekšņus, kas nosaka hostu marķieru krāsas:

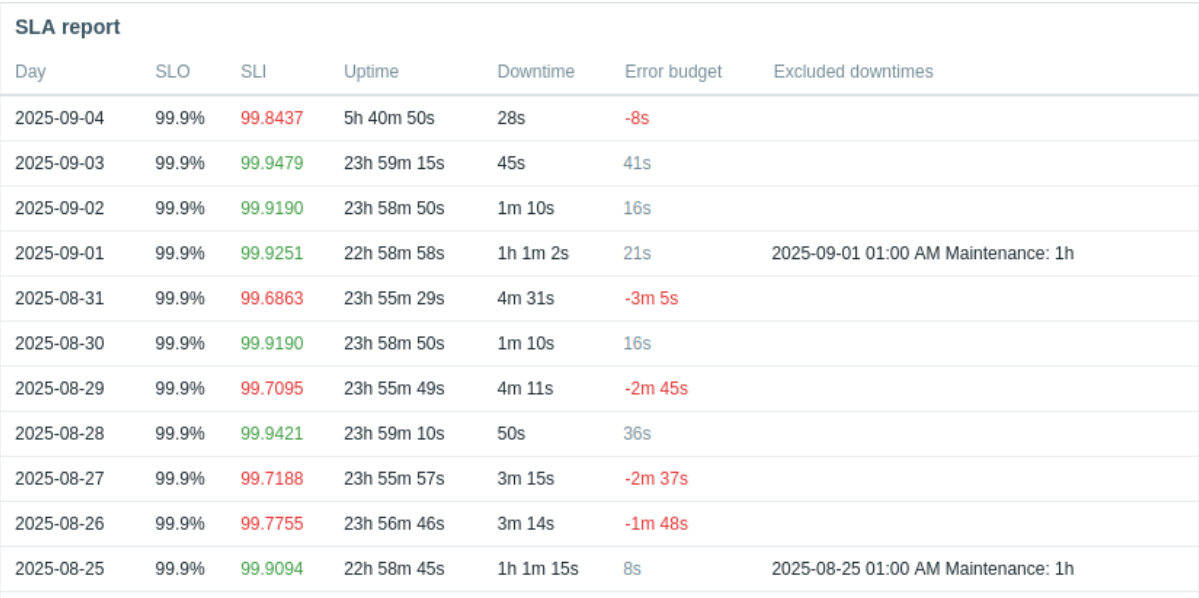


Krāsu interpolācija	Atzīmējiet izvēles rūtiņu, lai iespējotu vienmērīgu pāreju starp sliekšņu krāsām hostu marķieriem. Šis parametrs tiek lietots, ja X ass, Y ass vai abām asīm vienlaikus ir iestatīti divi vai vairāk sliekšņi.
X ass/Y ass slieksnis	Noklikšķiniet uz *Pievienot*, lai pievienotu slieksni, krāsu atlasītājā izvēlieties sliekšņa krāsu un norādiet skaitlisku vērtību. Sliekšņus var lietot 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 vērtību kartēšanas.

SLA pārskats

Pārskats

Logrīks *SLA report* attēlo SLA atskaites, palīdzot jums uzraudzīt pakalpojumu veiktspēju atbilstoši noteiktajiem mērķiem.

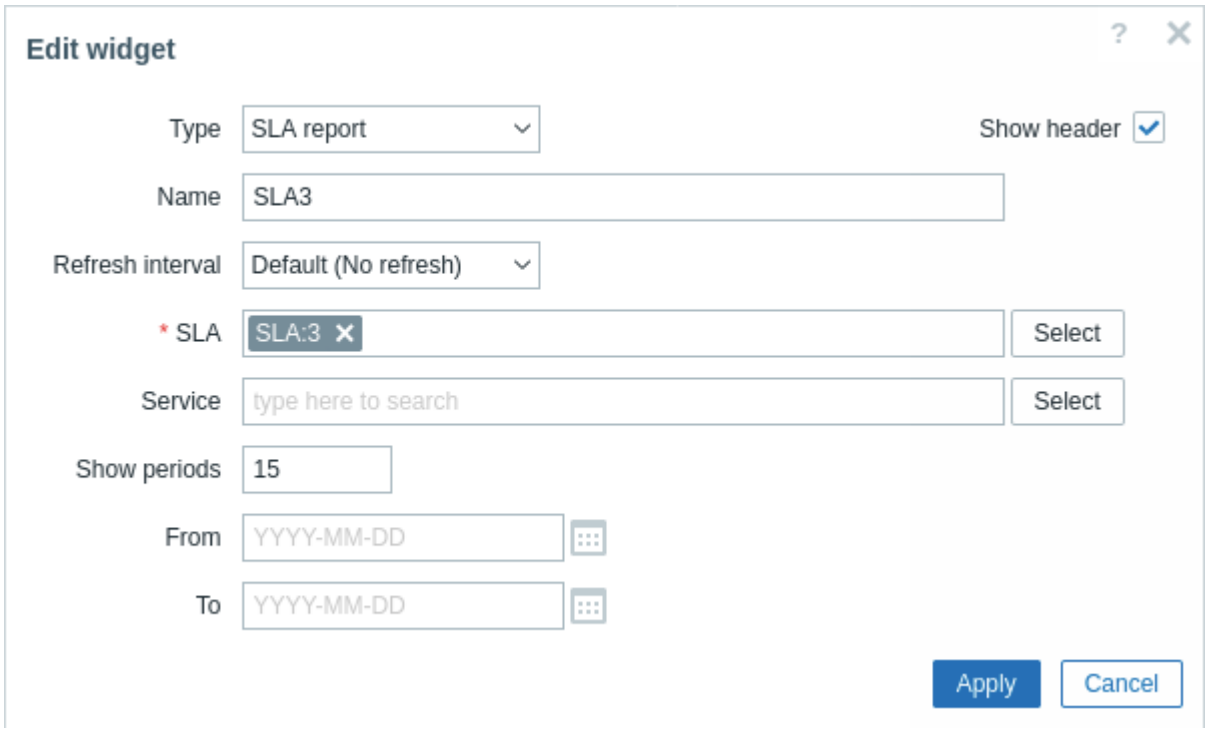


Day	SLO	SLI	Uptime	Downtime	Error budget	Excluded downtimes
2025-09-04	99.9%	99.8437	5h 40m 50s	28s	-8s	
2025-09-03	99.9%	99.9479	23h 59m 15s	45s	41s	
2025-09-02	99.9%	99.9190	23h 58m 50s	1m 10s	16s	
2025-09-01	99.9%	99.9251	22h 58m 58s	1h 1m 2s	21s	2025-09-01 01:00 AM Maintenance: 1h
2025-08-31	99.9%	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:



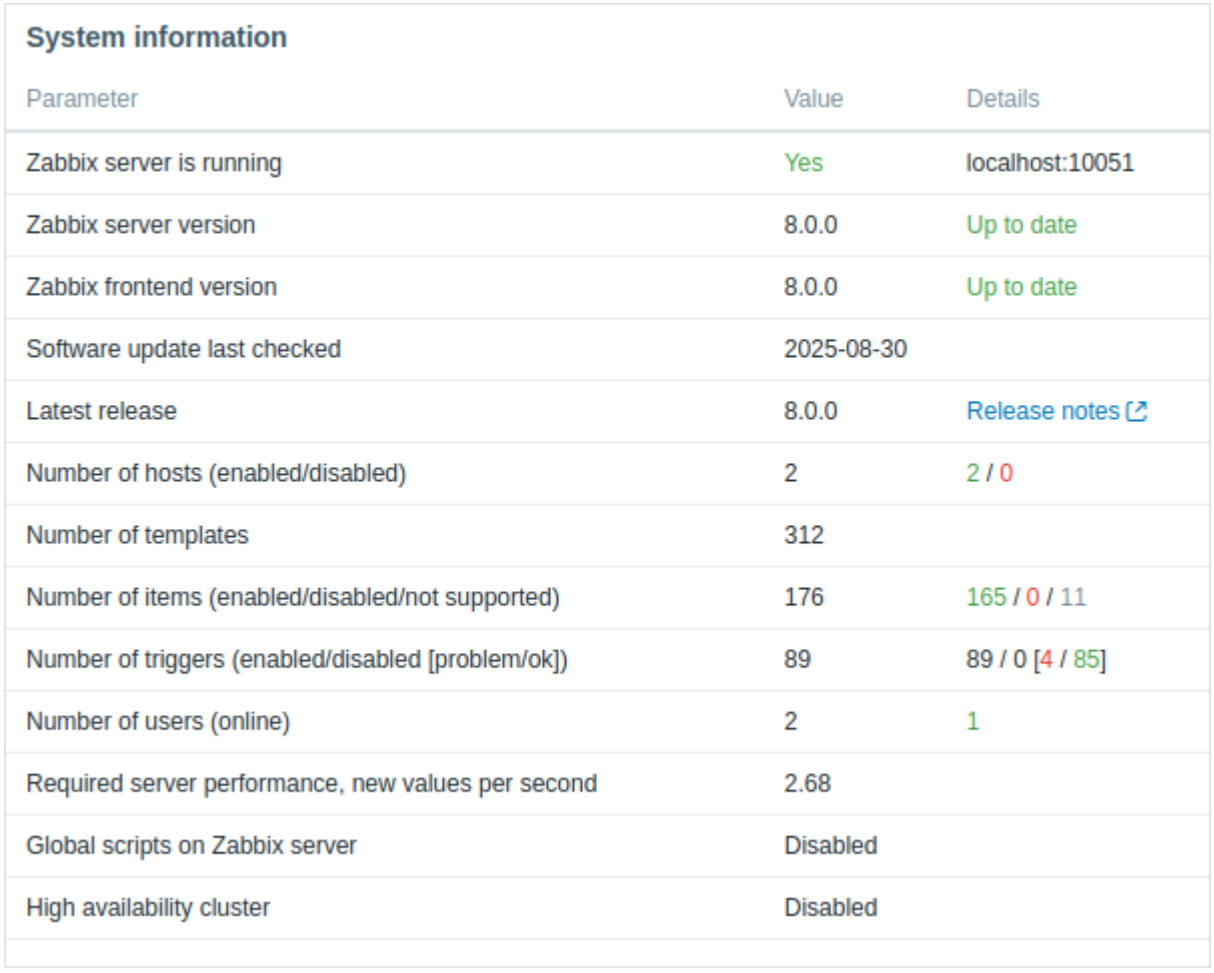
Papildus parametriem, kas ir kopīgi visiem logrīkiem, varat iestatīt šādas specifiskas opcijas:

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`.

Sistēmas informācija

Pārskats

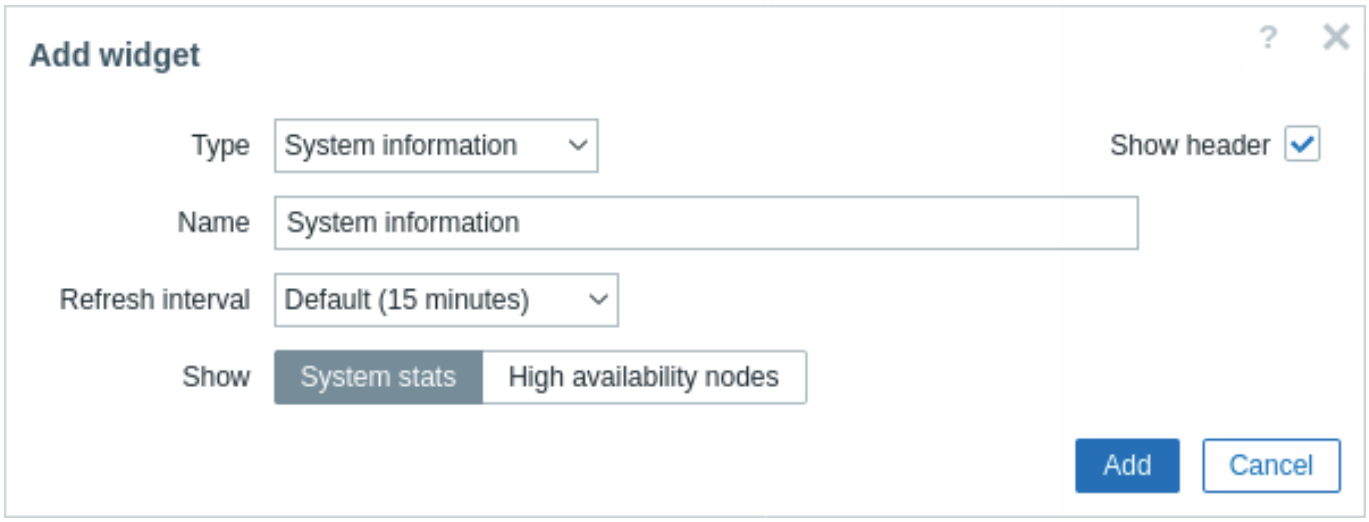
System information logrīks parāda galveno Zabbix servera un sistēmas datu kopsavilkumu, kā arī, ja ir iespējota augsta pieejamība, augstās pieejamības mezglu informāciju.



Tas parāda tos pašus datus kā *Reports* > *System information*, izņemot *Zabbix server ID*.

Konfigurēšana

Lai konfigurētu, kā tipu atlasiet *System information*:



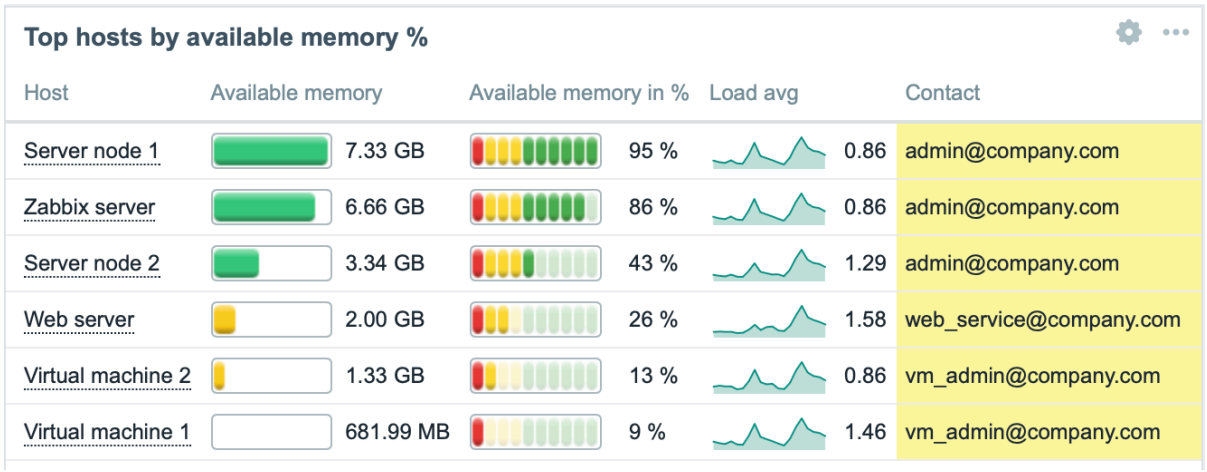
Papildus parametriem, kas ir kopīgi visiem logrīkiem, varat iestatīt tālāk norādītās specifiskās opcijas:

Show	Atlasiet, ko rādīt: **System stats** - rādīt Zabbix servera un sistēmas galveno datu kopsavilkumu; **High availability nodes** - rādīt augstas pieejamības mezglu statusu (ja ir iespējots augstas pieejamības klasteris).
Show software update check details	Atzīmējiet izvēles rūtiņu, lai rādītu detalizētu informāciju par Zabbix programmatūras atjauninājumu pārbaudi. Šī opcija ir pieejama tikai tad, ja Zabbix servera konfigurācijā ir iespējota programmatūras atjauninājumu pārbaude un laukā *Show* ir atlasīts "System stats".

Top hosts

Pārskats

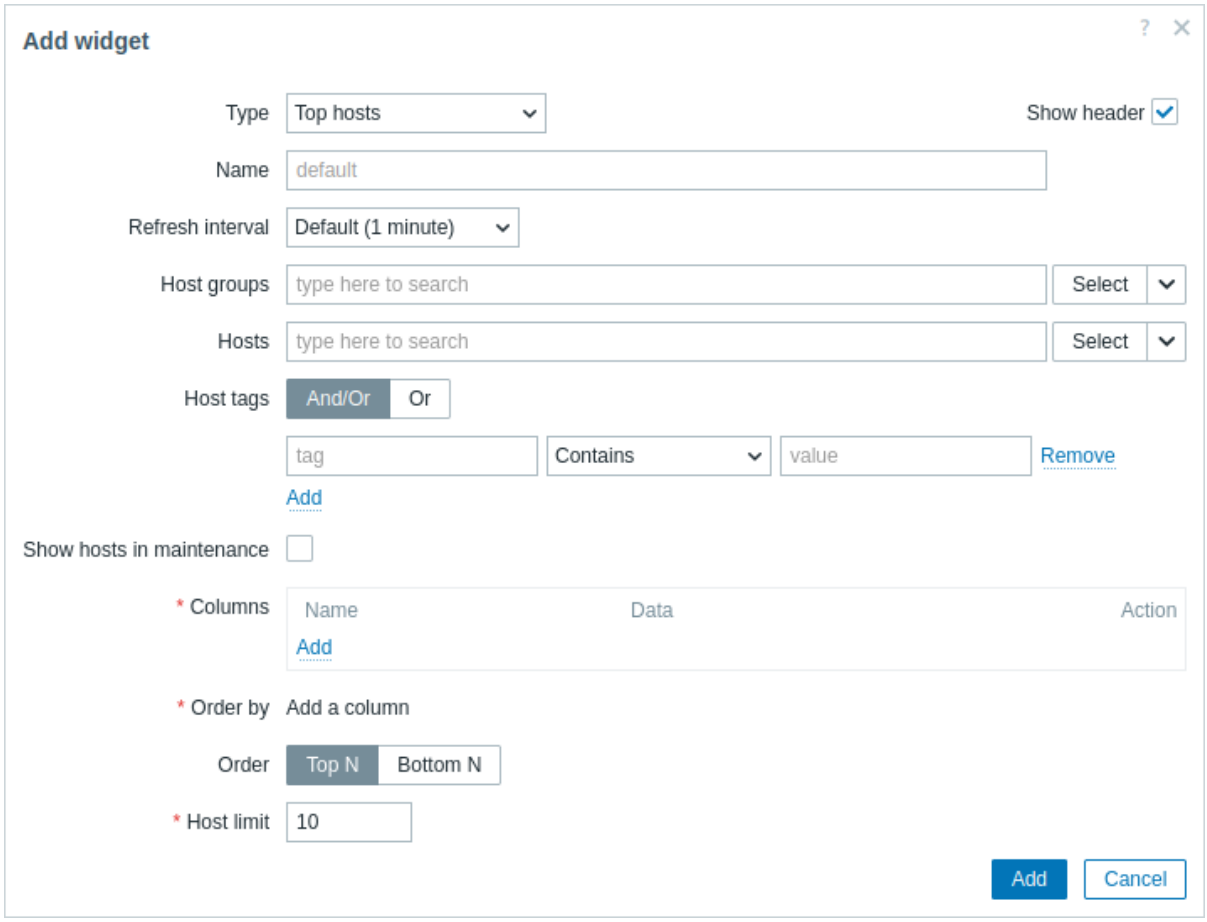
Logrīks *Top hosts* attēlo atlasīto vienumu augšējās/apakšējās vērtības no vairākiem hostiem.



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:

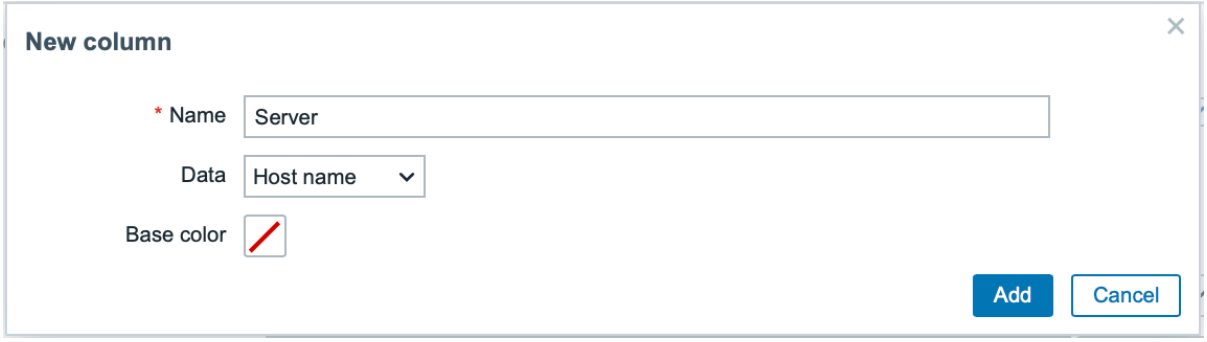
Hostu grupas	Atlasiet hostu grupas, ko attēlot logrīkā. Alternatīvi atlasiet saderīgu logrīku kā datu avotu hostu grupām. Šis lauks atbalsta automātisko pabeigšanu, tāpēc, sākot rakstīt grupas nosaukumu, tiks piedāvāts atbilstošu grupu nolaižamais saraksts. Šis parametrs nav pieejams, konfigurējot logrīku 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.

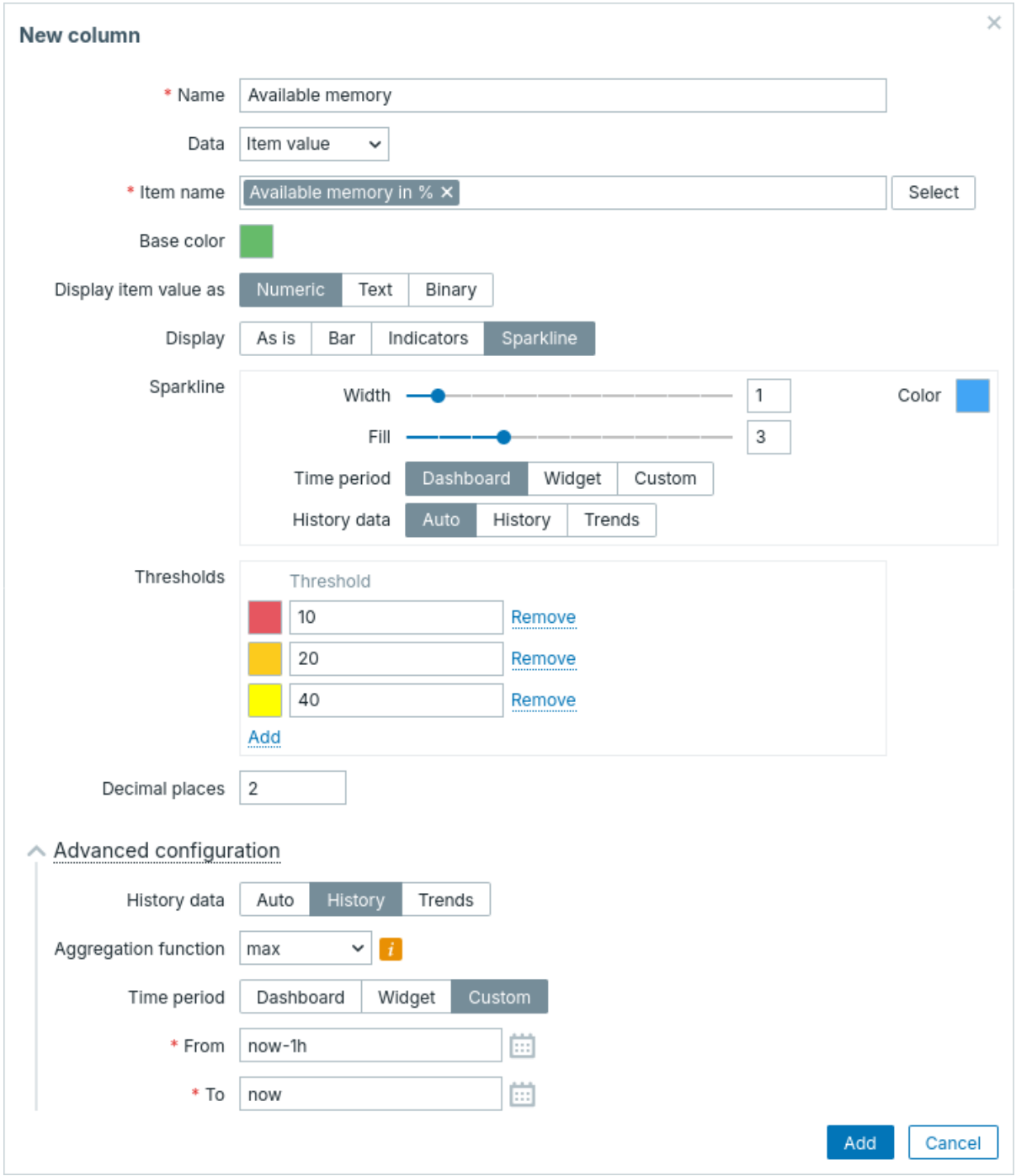


Atbalstītie parametri:

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 parādītu 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 visos atlasītajos hosts atrastu un parādītu vienumus ar tādu pašu nosaukumu. Ja host ir vairāki šādi vienumi, logrīks parāda vienumu, kura atslēga alfabētiski ir pirmā (piemēram, *CPU utilization* ar atslēgu `proc.cpu.util`, nevis `system.cpu.util`). Konfigurējot logrīku veidnes informācijas panelī, var atlasīt tikai veidnē konfigurētos vienumus.
Base color	Kolonnas fona krāsa; aizpildījuma krāsa, ja *Item value* dati tiek attēloti kā josla vai indikatori. Noklusējuma krāsu var aizstāt ar pielāgotu krāsu, ja vienuma vērtība pārsniedz kādu no norādītajām robežvērtībām.
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 tālāk.
Advanced configuration	

History data	Iegūt datus no vēstures vai tendencēm: **Auto** - automātiska atlase; **History** - iegūt vēstures datus; **Trends** - iegūt tendenču datus. Šis parametrs ir pieejams tikai skaitliskām vienumu vērtībām. Teksta un binārajām vērtībām dati vienmēr tiks iegūti no vēstures.
Aggregation function	Norādiet, kuru apkopošanas funkciju izmantot atlasītajā *Time period*: **min** - parādīt mazāko vērtību; **max** - parādīt lielāko vērtību; **avg** - parādīt vidējo vērtību; **count** - parādīt vērtību skaitu; **sum** - parādīt vērtību summu; **first** - parādīt pirmo vērtību; **last** - parādīt pēdējo vērtību. Ja atlasīts **not used**, tiek parādīta jaunākā vērtība (bez apkopošanas). Apkopošana ļauj norādītajā intervālā (5 minūtēs, stundā, dienā) parādīt apkopotu vērtību, nevis jaunāko vērtību. Parametriem *min*, *max*, *avg* un *sum* var parādīt tikai skaitliskus datus. Parametram *count* neskaitliskie dati tiks pārveidoti par skaitliskiem.
Time period	Norādiet laika periodu, kas jāizmanto vērtību apkopošanai. Atlasiet datu avotu laika periodam: **Dashboard** - izmantot informācijas 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 uz tās peles rādītāju, norādīs 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` u.c.). 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` u.c.). Šis parametrs ir pieejams, ja *Time period* ir iestatīts uz "Custom".

Skaitliskas vienuma vērtības parametri:

Display	Norādiet, kā vērtība jāattēlo: **As is** - kā parasts teksts; **Bar** - kā vienkrāsaina, ar krāsu aizpildīta josla; **Indicators** - kā segmentēta, ar krāsu aizpildīta josla; **Sparkline** - miniatūra līniju diagramma.
Thresholds	Norādiet robežvērtības, pie kurām jāmaina fona vai aizpildījuma krāsa. Saglabājot saraksts tiks sakārtots augošā secībā. Dzirksteļdiagrammām robežvērtības tiek piemērotas tikai vienuma pēdējai vērtībai.
Decimal places	Norādiet, cik zīmes aiz komata tiks parādītas kopā ar vērtību.
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 dzirksteļdiagrammā. Atlasiet datu avotu laika periodam: **Dashboard** - izmantot informācijas 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 uz tās peles rādītāju, 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` u.c.). 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` u.c.). Šis parametrs ir pieejams, ja *Time period* ir iestatīts uz "Custom".
History data	Iegūt datus no vēstures vai tendencēm: **Auto** - automātiska atlase; **History** - iegūt vēstures datus; **Trends** - iegūt tendenču datus.

Teksta vienuma vērtības parametri:

Highlights	Norādiet regulārās izteiksmes, kuru atbilstības gadījumā jāmaina fona vai aizpildījuma krāsa.

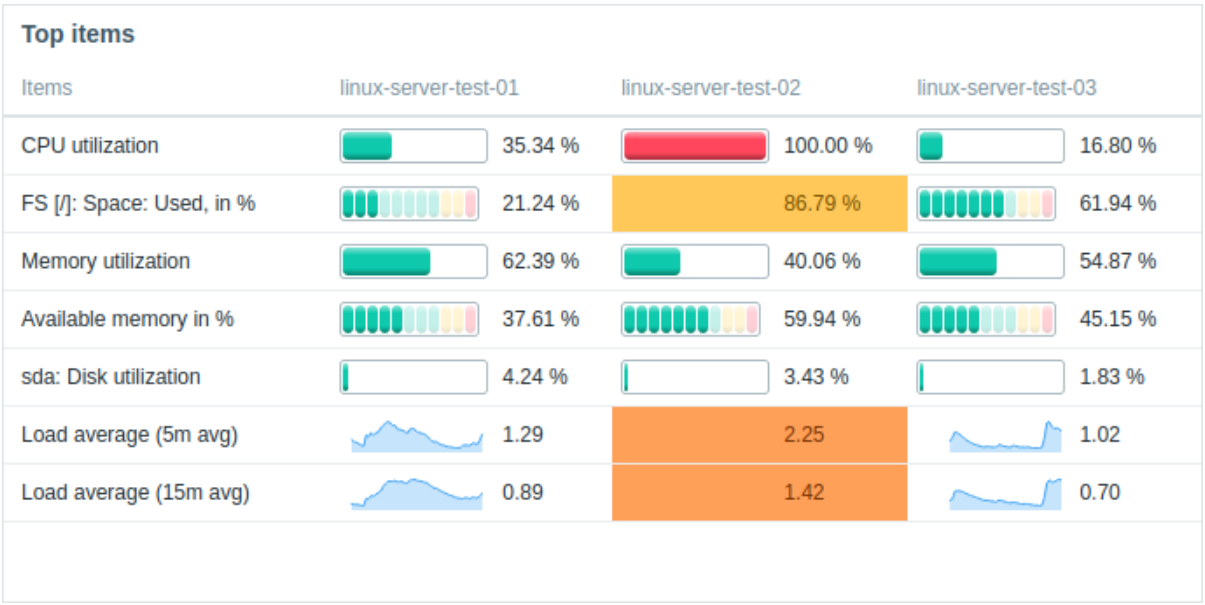
Bināras vienuma vērtības parametri:

Show thumbnail	Norādiet, vai izveidot un parādīt sīktēlu attēlam, kas satur bināros datus, vai vērtību kolonnā parādīt hipersaiti *Show*, kas ved uz pilna izmēra attēlu.

Populārākie vienumi

Pārskats

Logrīks *Top vienumi* parāda atlasīto vienumu augšējās/apakšējās vērtības, nodrošinot ātru to veiktspējas pārskatu.

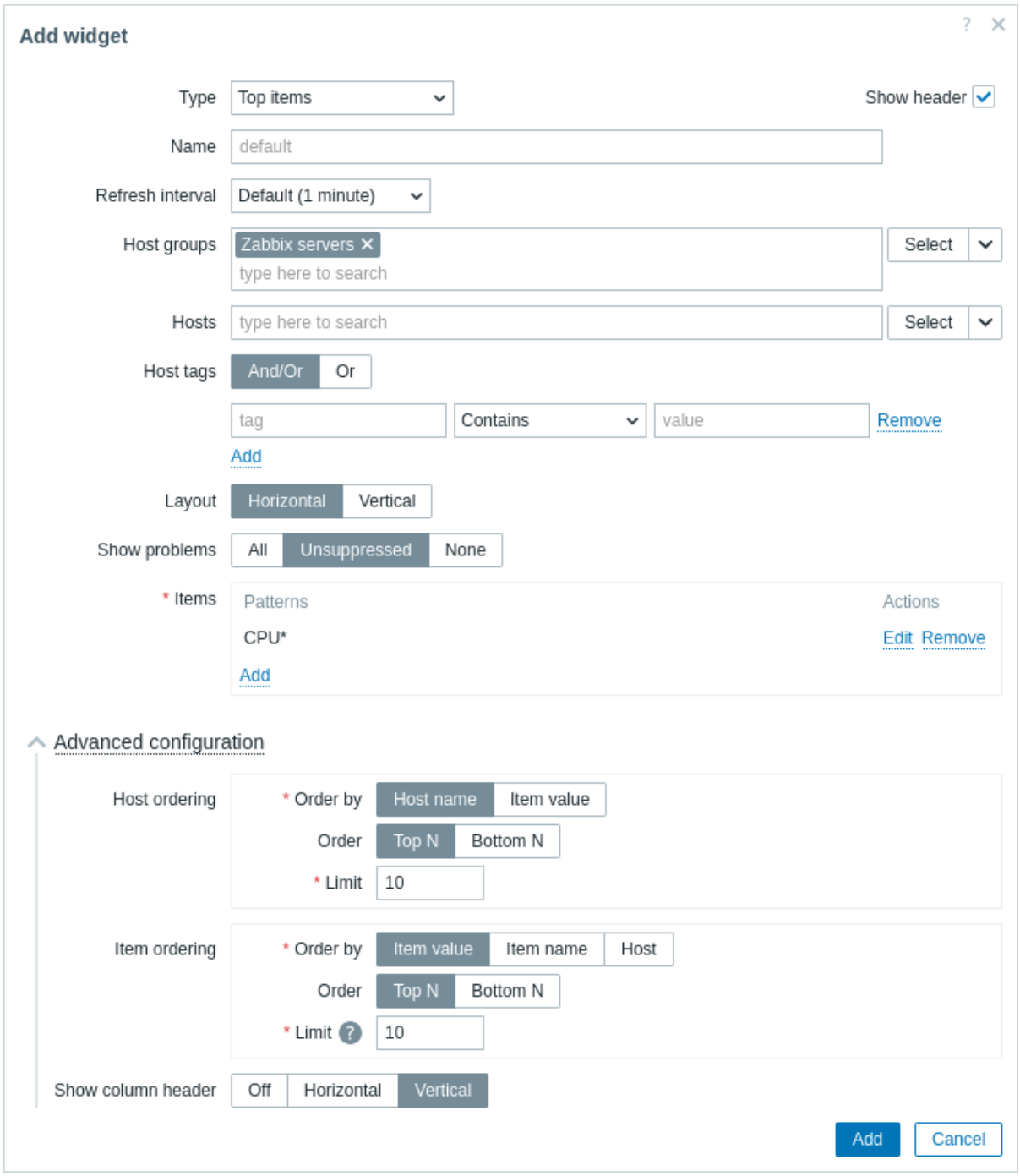


Pēc noklusējuma logrīks parāda vērtības par pēdējām 24 stundām. Šo periodu var mainīt sadaļā *Administrēšana* > *Vispārīgi* > *GUI* (*Maksimālais vēstures attēlošanas periods*).

Noklikšķinot uz vienuma vērtības, tiek atvērta vienuma izvēlne.

Konfigurācija

Lai konfigurētu, kā tipu atlasiet *Top items*:



Papildus parametriem, kas ir kopīgi visiem logrīkiem, varat iestatīt šādas specifiskas opcijas:

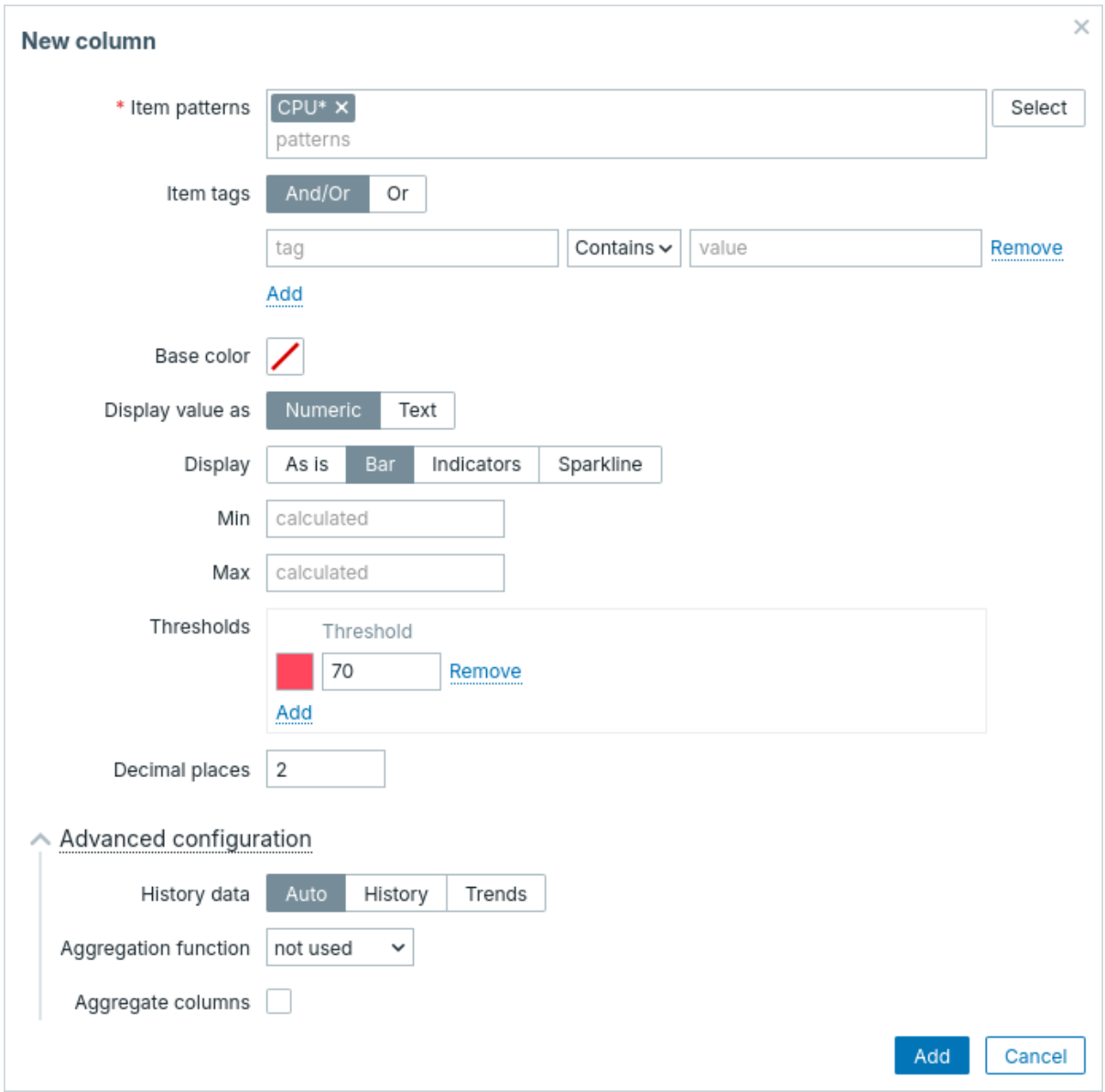
Host gru-pas	Atlasiet host grupas. Vai arī atlasiet saderīgu logrīku kā datu avotu host grupām. Šis lauks atbalsta automātisko pabeigšanu, tāpēc, sākot rakstīt grupas nosaukumu, tiks piedāvāts nolaižamais saraksts ar atbilstošajām grupām. Šis parametrs nav pieejams, konfigurējot logrīku veidnes informācijas panelī.
Hosts	Atlasiet hosts. Vai arī atlasiet saderīgu logrīku vai informācijas paneli kā datu avotu hosts. Šis lauks atbalsta automātisko pabeigšanu, tāpēc, sākot rakstīt hosta nosaukumu, tiks piedāvāts nolaižamais saraksts ar atbilstošajiem hosts. Šis parametrs nav pieejams, konfigurējot logrīku veidnes informācijas panelī.

Host tagi	Norādiet tagus, lai ierobežotu logrīkā attēloto host datu skaitu. Ir iespējams iekļaut, kā arī izslēgt konkrētus tagus un tagu vērtības. Var iestatīt vairākus nosacījumus. Tagu nosaukumu 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, nereģistrjutīgi); **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, nereģistrjutīgi). Nosacījumiem ir pieejami divi aprēķina tipi: **And/Or** - jāizpildās visiem nosacījumiem, nosacījumi ar vienādu taga 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 rādīt, pamatojoties uz to statusu (all, unsuppressed, none). Problēmas tiek norādītas ar iekrāsotām vienumu vērtībām, pārrakstot vienuma kolonnas attēlošanas iestatījumus. Krāsas ir balstītas uz problēmas smagumu, ko var pielāgot sadaļā *Administration* > *General* > *Trigger displaying options*. Problēmu smaguma pakāpes var mainīt, atjauninot problēmas.
Vienumi	Pievienojiet vienumu veidnes vai konkrētus vienumus attēlošanai (skatiet Kolonnas konfigurācija).
Papildu konfigurācija	
Host kārtošana	Atlasiet hostu kolonnas/rindas kārtošanas opcijas.
Kārtot pēc	Kārtojiet hosts pēc: **Host name** - hosts tiks kārtoti pēc hosta nosaukuma; **Item value** - hosts tiks kārtoti pēc atlasīto vienumu vērtības.
Kārtība	Atlasiet, vai rādīt augstākās vai zemākās vērtības: **TopN** - augstākās N vērtības; **BottomN** - zemākās N vērtības. N vērtība tiek atlasīta laukā *Limit*.
Limits	Ievadiet attēlojamo hosts limitu (diapazons 1-1000; noklusējums 10). Šī vērtība kļūs par N vērtību laukā *Kārtība*.
Vienumu kārtošana	Atlasiet vienumu kolonnas/rindas kārtošanas opcijas.
Kārtot pēc	Kārtojiet 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 veidnes.
Kārtība	Atlasiet, vai rādīt augstākās vai zemākās vērtības: **TopN** - augstākās N vērtības; **BottomN** - zemākās N vērtības. N vērtība tiek atlasīta laukā *Limit*.
Limits	Ievadiet attēlojamo vienumu limitu (diapazons 1-1000; noklusējums 10). Šī vērtība kļūs par N vērtību laukā *Kārtība*.

Rādīt kolonnas galveni	Atlasiet kolonnas galvenes attēlošanas opcijas: **Off** - nerādīt kolonnas galveni; **Horizontal** - attēlot tekstu horizontāli galvenē; **Vertical** - attēlot tekstu vertikāli galvenē.

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.

Kolonnas parametri skaitliskām vērtībām:

Attēlojums	Norādiet, kā vērtība jāattēlo: **Kā ir** - kā parasts teksts; **Josla** - kā vienmērīga, ar krāsu aizpildīta josla; **Indikatori** - kā segmentēta, ar krāsu aizpildīta josla; **Sparkline** - miniatūra līniju diagramma.
Min.	Minimālā vērtība joslas/indikatoru attēlošanai.
Maks.	Maksimālā vērtība joslas/indikatoru attēlošanai.
Platums	Iestatiet diagrammas līnijas biezumu, izmantojot slīdni vai manuāli ievadot vērtību diapazonā no 0 līdz 10. Šis parametrs attiecas tikai uz Sparkline attēlojumu.
Aizpildījums	Iestatiet aizpildījuma krāsas caurspīdīguma līmeni, izmantojot slīdni vai manuāli ievadot vērtību diapazonā no 0 līdz 10. Šis parametrs attiecas tikai uz Sparkline attēlojumu.
Krāsa	Atlasiet līnijas un aizpildījuma krāsu. Šis parametrs attiecas tikai uz Sparkline attēlojumu.
Lauku, kas saistīti ar laika perioda un vēstures datu atlasi, aprakstu skatiet sadaļā *Papildu konfigurācija*.	
Sliekšņi	Norādiet sliekšņa vērtības, pie kurām jāmainās fona/aizpildījuma krāsai. Saglabājot, saraksts tiks sakārtots augošā secībā. Sparkline diagrammām sliekšņi tiek piemēroti tikai vienuma pēdējai vērtībai.
Decimāldaļas	Norādiet, cik decimāldaļu tiks attēlots kopā ar vērtību.
Papildu konfigurācija	
Vēstures dati	Iegūt datus no vēstures vai tendencēm: **Automātiski** - automātiska atlase; **Vēsture** - izmantot vēstures datus; **Tendences** - izmantot tendenču datus. Šis parametrs ir pieejams tikai skaitlisku vienumu vērtībām. Teksta vērtībām dati vienmēr tiks iegūti no vēstures.

Apkopošanas funkcija	Norādiet, kuru apkopošanas 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 atlasīts **neizmantot**, tiek attēlota jaunākā vērtība (bez apkopošanas). Apkopošana ļauj norādītajā intervālā (5 minūtēs, stundā, dienā) attēlot apkopotu vērtību, nevis jaunāko vērtību. Funkcijām *min*, *max*, *avg* un *sum* var attēlot tikai skaitliskus datus. Funkcijai *count* neskaitliskie dati tiks pārveidoti par skaitliskiem.
Laika periods	Norādiet laika periodu, kas jāizmanto vērtību apkopošanai. Atlasiet datu avotu laika periodam: **Informācijas panelis** - izmantot informācijas 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 uz tās peles rādītāju, norāda iestatīto laiku. Ņemiet vērā, ka neatkarīgi no logrīka *Laika perioda* konfigurācijas saderīgi logrīki joprojām var tikt izmantoti kā laika perioda datu avots. Šis parametrs nav pieejams, ja *Apkopošanas funkcijai* iestatīta vērtība ”neizmantot”.
Logrīks	Ievadiet vai atlasiet saderīgu logrīku kā laika perioda datu avotu. Šis parametrs ir pieejams, ja *Laika periodam* iestatīta vērtība ”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 periodam* iestatīta vērtība ”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 periodam* iestatīta vērtība ”Pielāgots”.
Apkopot kolonnas	Ja atlasīts, visi vienumi, kas atbilst kolonnas vienumu modeļiem, tiek grupēti vienā apvienotā kolonnā/rindā. Atbilstošo vienumu vērtības tiks apkopotas, izmantojot atlasīto *Kolonnu apkopošanas funkciju*, un attēlotas zem *Apvienotās kolonnas nosaukuma*. Noklusējums: izslēgts (nav atlasīts).Šis parametrs tiek parādīts tikai tad, ja *Attēlot vērtību kā* ir **Skaitliski**.
Kolonnu apkopošanas funkcija	Izvēlieties, kā apvienot vērtības no grupētajiem vienumu modeļiem. Pieejamās opcijas: **min**, **max**, **avg**, **count**, **sum** (noklusējums). Šis parametrs tiek parādīts tikai tad, ja ir atlasīts *Apkopot kolonnas*.
Apvienotās kolonnas nosaukums	Ievadiet apkopotās kolonnas attēlojamo nosaukumu. Šis lauks ir obligāts, ja ir atlasīts *Apkopot kolonnas*, un tas kļūst par apvienoto vērtību kolonnas galveni.

Kolonnas parametri teksta vērtībām:

Izcēlumi	Norādiet regulārās izteiksmes, kuru atbilstības gadījumā jāmainās fona/aizpildījuma krāsai.

Populārākie trigeri

Pārskats

Logrīks *Top trigeri* attēlo trigerus ar lielāko problēmu skaitu.

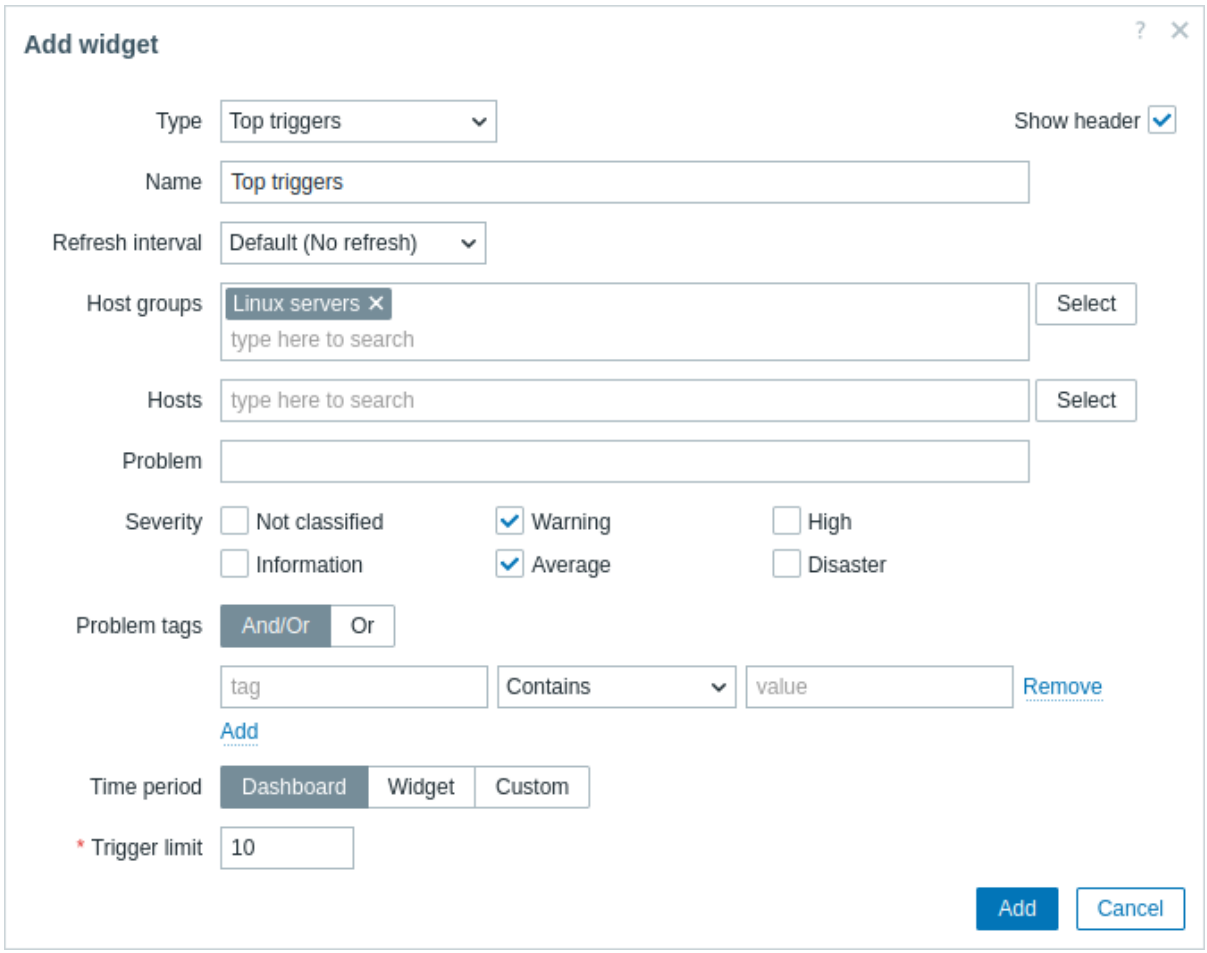
Top triggers

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.

Trigeru pārskats

Pārskats

Trigger overview sīkrīks parāda hostu trigeru pašreizējos stāvokļus kā krāsainu bloku tabulu. Tas nodrošina ātru vizuālu pārskatu par trigeru stāvokli un aktivitāti dažādos hostos.



Sīkrīks izmanto vairākus vizuālos elementus, lai attēlotu trigeru informāciju:

- Bloku krāsas PROBLEM trigeriem atbilst problēmu smaguma pakāpēm.
- Mirgojoši bloki norāda uz nesenām trigeru stāvokļa izmaiņām (pēdējo 2 minūšu laikā).
- Pelēkas bultiņas norāda trigerus ar atkarībām (uzvediet kursoru virs tām, lai skatītu informāciju).
- Atzīmju ikonas norāda apstiprinātas problēmas; lai šī ikona tiktu parādīta, visām aktīvajām vai atrisinātajām problēmām šim trigerim jābūt apstiprinātām.

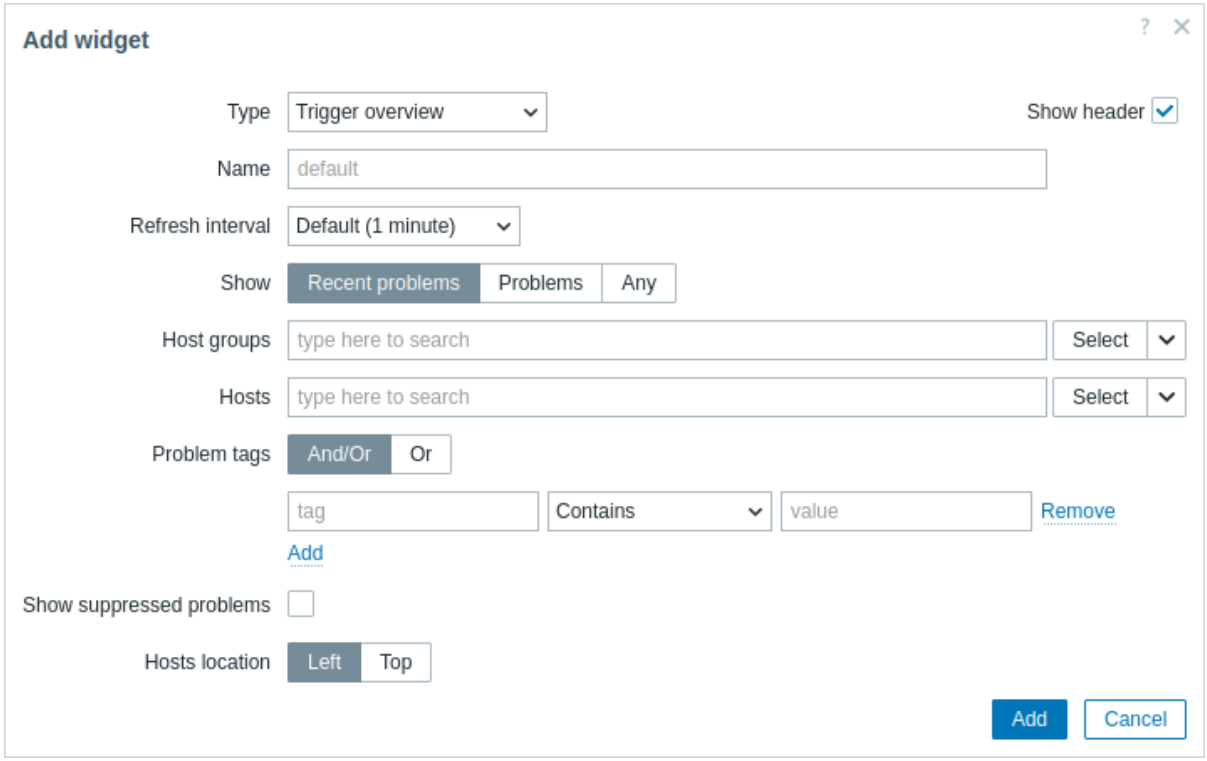
Note:
Trigeru smaguma pakāpju krāsas var pielāgot sadaļā *Administration* > *General*. Trigeru smaguma pakāpes var mainīt, atjauninot problēmas.

Noklikšķinot uz bloka, tiek parādītas saites uz ar trigeri saistītu informāciju (problēmas, trigeru konfigurācija u. c.).

Pēc noklusējuma tiek parādīti līdz 50 ierakstiem (to var konfigurēt sadaļā *Administration* > *General* > *GUI*, laukā *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:

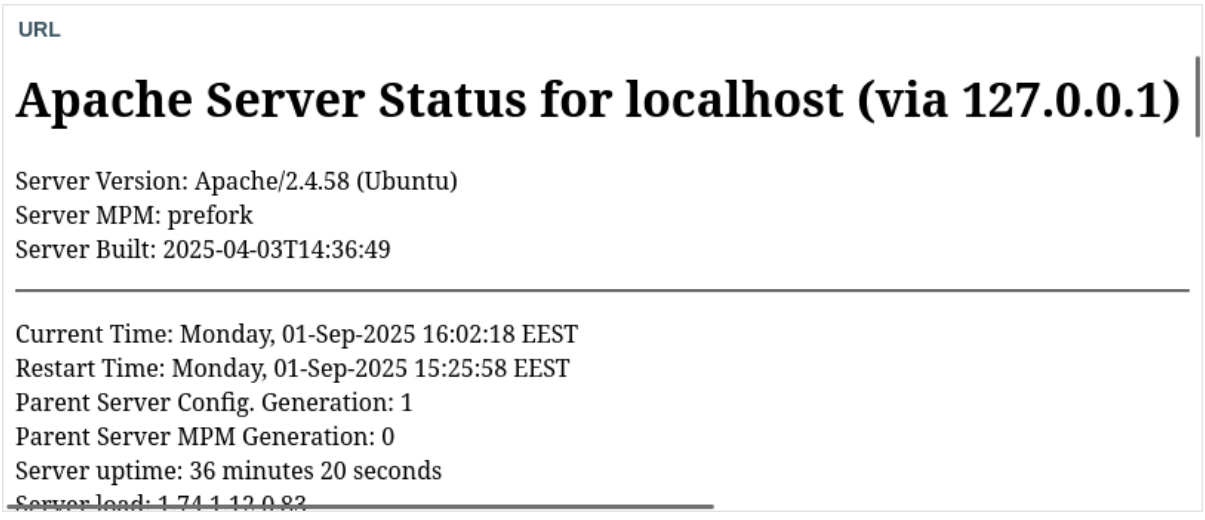
Rādīt	Filtrēt trigerus pēc trigeru stāvokļa: **Nesenās problēmas** - *(noklusējums)* rādīt trigerus, kas nesen bija vai joprojām ir PROBLEM stāvoklī (atrisinātas un neatrisinātas); **Problēmas** - rādīt trigerus, kas ir PROBLEM stāvoklī (neatrisinātas); **Jebkuri** - rādīt visus trigerus.
Hostu grupas	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.

URL

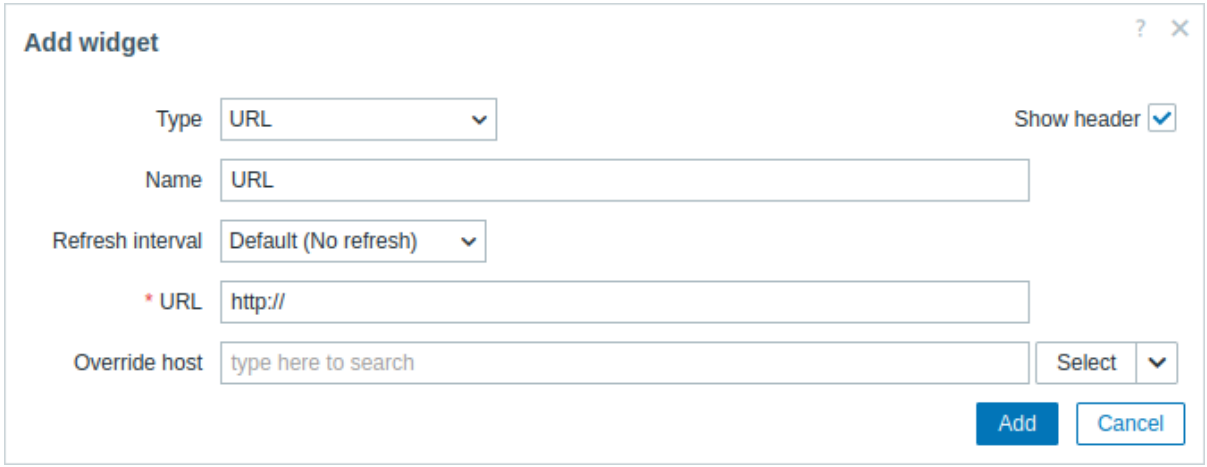
Pārskats

URL logdaļa attēlo saturu, kas iegūts no URL.



Konfigurācija

Lai konfigurētu, kā tipu atlasiet *URL*:



Papildus parametriem, kas ir kopīgi visiem logrīkiem, varat iestatīt šādas specifiskās opcijas:

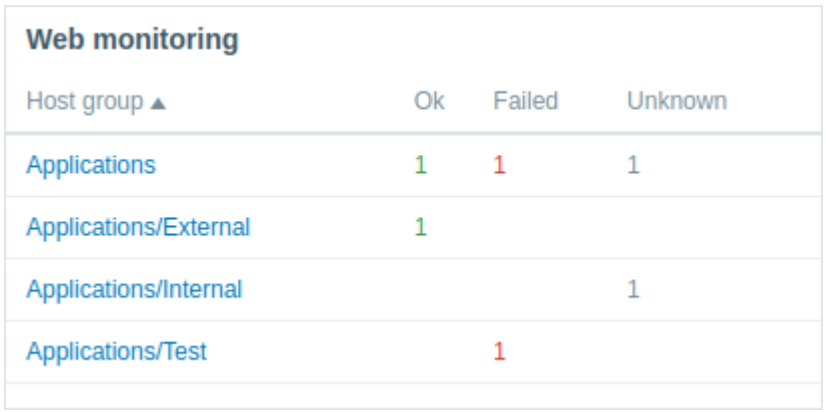
URL	Ievadiet attēlojamo URL (līdz 2048 rakstzīmēm). Ārējiem URL jāsākas ar `http://` vai `https://`. Iekšējie URL atbalsta relatīvos ceļus (piemēram, `zabbix.php?action=report.status`). Tiek atbalstītas `{HOST.*}` makrokomandas.
Override host	Atlasiet saderīgu logrīku vai paneļa host selector kā datu avotu hostiem. Šis parametrs nav pieejams, konfigurējot logrīku veidnes panelī.

Attention:
Pārlūkprogrammas var neielādēt HTTP lapu, kas konfigurēta logrīkā, ja Zabbix lietotāja saskarne tiek atvērta, izmantojot HTTPS. Turklāt, lai parādītu saturu ārpus Zabbix lietotāja saskarnes, iespējams, būs jāmaina X-Frame-Options HTTP atbildes galvene.

Tīmekļa uzraudzība

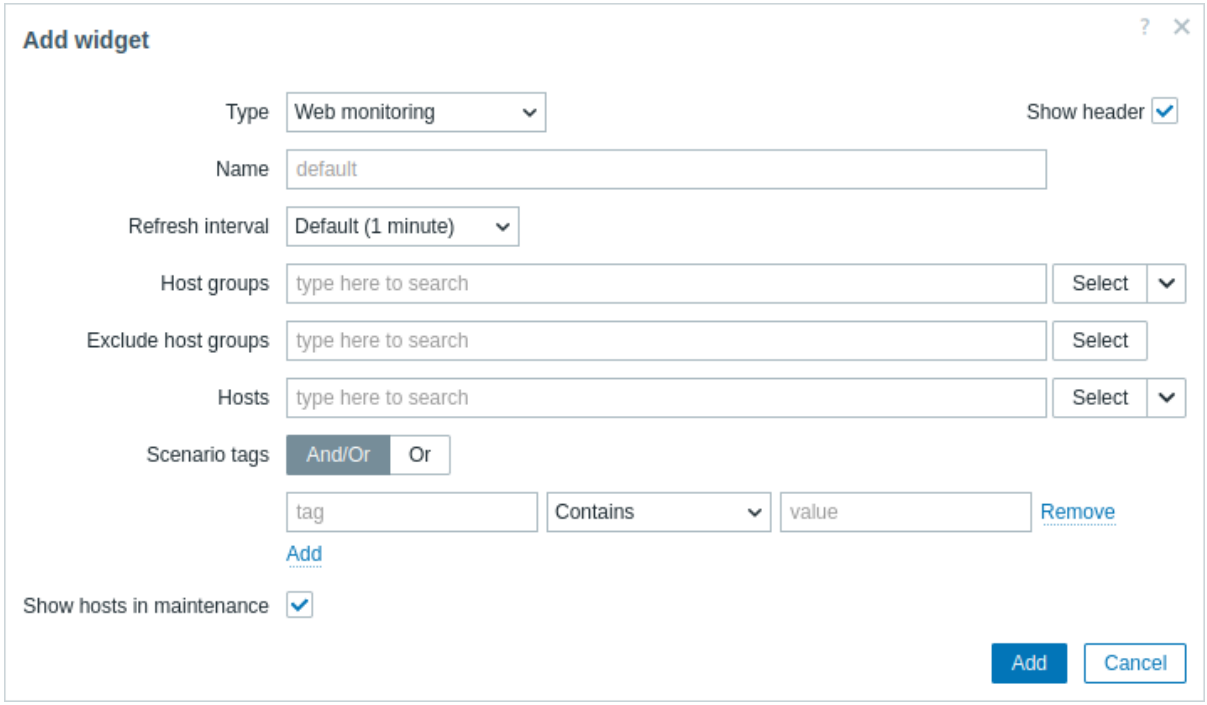
Pārskats

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.

Uzraudzība

Pārskats

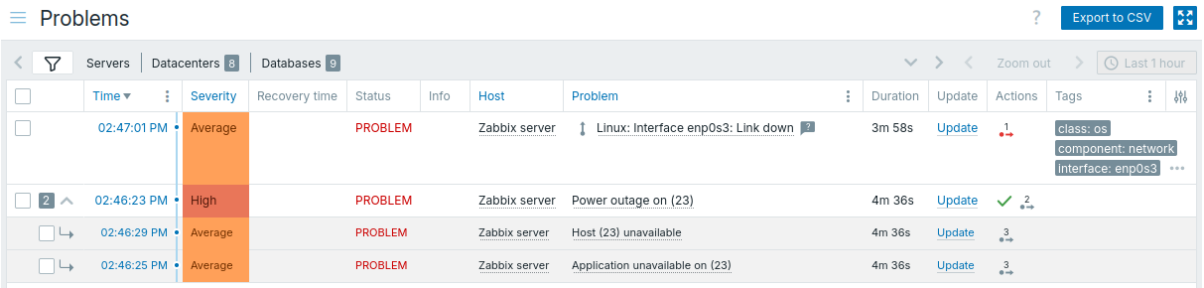
Izvēlne *Monitoring* ir paredzēta datu attēlošanai. Visa informācija, kuras apkopošanai, vizualizēšanai un apstrādei ir konfigurēts Zabbix, tiks attēlota dažādās izvēlnes *Monitoring* sadaļās.

Problēmas

Pārskats

Sadaļā *Monitoring* > *Problems* varat redzēt, kādas problēmas jums pašlaik ir. Par problēmām tiek uzskatīti tie trigeri, kas ir stāvoklī "Problem".

Pēc noklusējuma visas jaunās problēmas tiek klasificētas kā cēloņa problēmas. Dažas problēmas ir iespējams manuāli pārklasificēt kā cēloņa problēmas simptomu problēmas. Plašāku informāciju skatiet sadaļā cause and symptom events.

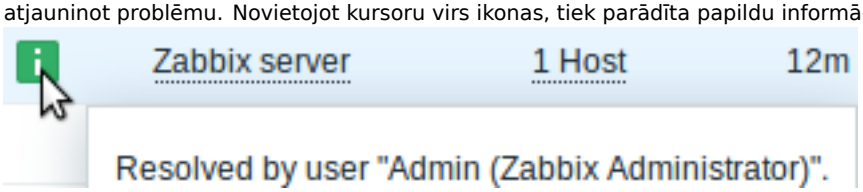
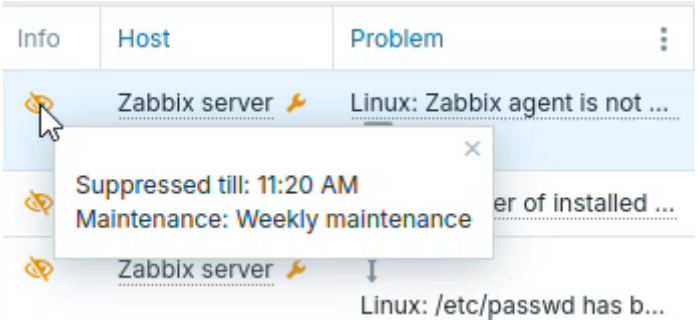


Problēmu informācija tiek parādīta pielāgojamā tabulā. Poga *Customize table* tabulas galvenē ļauj parādīt, paslēpt vai pārkārtot kolonnas, kā arī atiestatīt izkārtojumu uz noklusējuma. Pieejamas arī šādas opcijas:

- *Compact view* - samazina rindas augstumu, lai ekrānā ietilptu vairāk problēmu.
- *Highlight full row* - izmanto problēmas smaguma pakāpes krāsu, lai izceltu neatrisinātas problēmas (nav pieejams augsta kontrasta tēmās).

Varat arī mainīt katras kolonnas platumu, velkot tās labo apmali; veiciet dubultklikšķi uz apmales, lai atiestatītu. Visas pielāgošanas tiek piemērotas nekavējoties un vienmēr tiek saglabātas pašreizējam lietotājam aktīvajā filter cilnē (noklusējuma vai saglabātajā).

Pieejamas šādas kolonnas:

Column	Description
Checkbox	Tiek parādītas izvēles rūtiņas problēmu atlasei. Ikonām blakus izvēles rūtiņām ir šāda nozīme:  - simptomu notikumu skaits cēloņa problēmai;  - izvērst, lai parādītu simptomu notikumus;  - sakļaut, lai paslēptu simptomu notikumus;  - tas ir simptomu notikums.
Time	Tiek parādīts problēmas sākuma laiks. Noklikšķinot uz pogas *Options* kolonnas galvenē, varat piekļūt opcijai *Show timeline*, kas pārslēdz vizuālo laika skalu un problēmu grupēšanu (nav pieejams, ja ir iespējots *Compact view*).
Severity	Tiek parādīta problēmas smaguma pakāpe. Problēmas smaguma pakāpe sākotnēji ir balstīta uz pamatā esošā problēmas trigeris smaguma pakāpi, tomēr pēc notikuma rašanās to var atjaunināt, izmantojot *Update problem* screen. Problēmas smaguma pakāpes krāsa problēmas laikā tiek izmantota kā šūnas fons.
Recovery time	Tiek parādīts problēmas atrisināšanas laiks Šī kolonna nav pieejama, ja filter sadaļā *Show* ir iestatīts uz *Problems*.
Status	Tiek parādīts problēmas statuss: **Problem** - neatrisināta problēma **Resolved** - nesen atrisināta problēma. Nesen atrisinātas problēmas var paslēpt, izmantojot filtru. Jaunas un nesen atrisinātas problēmas mirgo 2 minūtes. Atrisinātas problēmas kopumā tiek rādītas 5 minūtes. Abas šīs vērtības ir konfigurējamas sadaļā *Administration* > *General* > *Trigger displaying options*. Šī kolonna nav pieejama, ja filter sadaļā *Show* ir iestatīts uz *Problems*.
Info	Tiek parādīta zaļa informācijas ikona, ja problēma ir slēgta ar globālu korelāciju vai manuāli, atjauninot problēmu. Novietojot kursoru virs ikonas, tiek parādīta papildu informācija:  Ja tiek rādīta apspiesta problēma, tiek parādīta šāda ikona:  . Ja notikums ir apspiests hosta uzturēšanas laikā, ikona tiek rādīta oranžā krāsā:  . Ja attiecas gan uzturēšana, gan manuāla apspiešana, ikonas krāsā prioritāte ir manuālai apspiešanai. Novietojot kursoru virs ikonas, tiek parādīta papildu informācija: 
Host	Tiek parādīts problēmas hosts. Noklikšķinot uz hosta nosaukuma, tiek atvērta host menu.

Column	Description
Problem	Tiek parādīts problēmas nosaukums. Problēmas nosaukums ir balstīts uz pamatā esošā problēmas trigeris nosaukumu. Trigeris nosaukumā esošie makrosi tiek atrisināti problēmas rašanās brīdī, un atrisinātās vērtības vairs netiek atjauninātas. Noklikšķinot uz problēmas nosaukuma, tiek atvērta event menu. Novietojot kursoru virs ikonas pēc problēmas nosaukuma, tiek parādīts trigeris apraksts (tām problēmām, kurām tas ir pieejams). Noklikšķinot uz pogas *Options* kolonnas galvenē, varat piekļūt divām opcijām: - *Show operational data* parāda operatīvos datus aiz problēmas nosaukuma iekavās (tikai tad, ja lauks *Operational data* trigger configuration nav tukšs). - *Show trigger expression* parāda problēmas trigeris izteiksmi zem problēmas nosaukuma (nav pieejams, ja ir iespējots *Compact view*).
Operational data	Tiek parādīti dinamiskie, reāllaika dati par problēmu. Saturs ir atkarīgs no lauka *Operational data* trigger configuration: ja ir definēta virkne ar makro, tā tiek parādīta šeit; ja lauks ir atstāts tukšs, tā vietā tiek parādītas visu vienums jaunākās vērtības trigeris izteiksmē.
Duration	Tiek parādīts problēmas ilgums. Skatiet arī: Negative problem duration
Update	Noklikšķiniet uz saites *Update*, lai pārietu uz problem update ekrānu, kur problēmai var veikt dažādas darbības, tostarp pievienot komentāru un apstiprināt problēmu.
Actions	Problēmas darbību vēsture tiek parādīta, izmantojot simboliskas ikonas: - problēma ir apstiprināta. Šī ikona vienmēr tiek rādīta pirmā. - ir pievienoti komentāri. Tiek parādīts arī komentāru skaits. - problēmas smaguma pakāpe ir palielināta (piem., Information > Warning). - problēmas smaguma pakāpe ir samazināta (piem., Warning > Information). - problēmas smaguma pakāpe ir mainīta, bet atgriezta sākotnējā līmenī (piem., Warning > Information > Warning). - problēma ir apspiesta. - problēmas apspiešana ir noņemta. - problēma ir apspiesta hosta uzturēšanas dēļ. - problēmas apspiešana ir noņemta pēc hosta uzturēšanas. - ir veiktas darbības. Tiek parādīts arī darbību skaits. - ir veiktas darbības, vismaz viena ir procesā. Tiek parādīts arī darbību skaits. - ir veiktas darbības, vismaz viena ir neizdevusies. Tiek parādīts arī darbību skaits. Pārvietojot peli virs ikonām, tiek parādīti uznirstošie logi ar informāciju par darbību. Skatiet sadaļu viewing details, lai uzzinātu vairāk par ikonām, kas izmantotas uznirstošajā logā veiktajām darbībām.
Tags	Tiek parādīti problēmas tags. Turklāt var tikt parādīti arī ārējās biļešu sistēmas tagi (skatiet opciju *Process tags*, konfigurējot webhooks). Novietojot kursoru virs vai noklikšķinot uz trīs punktu ikonas aiz parādītajiem tagiem, tiek atvērts uznirstošais logs ar visiem tagiem. Noklikšķinot uz pogas *Options* kolonnas galvenē, varat iestatīt parādāmo tagu skaitu (1-3), izvēlēties, kā tiek rādīti tagu nosaukumi (pilni, saīsināti vai netiek rādīti), iestatīt tagu attēlošanas prioritāti vai dublēt kolonnu (dublētās kolonnas var pārdēvēt vai dzēst).
Tag value	Tiek parādītas problēmas tag vērtības. Noklikšķinot uz pogas *Options* kolonnas galvenē, varat pārdēvēt kolonnu, izvēlēties, kuras tagu vērtības rādīt, vai dublēt kolonnu (dublētās kolonnas var pārdēvēt vai dzēst).

Column	Description
Custom text	Tiek parādīts pielāgots teksts, makro vērtības vai abu kombinācija. Noklikšķinot uz pogas ⋮ *Options* kolonnas galvenē, varat pārdēvēt kolonnu, ievadīt pielāgoto tekstu vai makro, vai dublēt kolonnu (dublētās kolonnas var pārdēvēt vai dzēst). Supported macros: {HOST.*} (izņemot {HOST.METADATA}), {INVENTORY.*}, {ITEM.*} (izņemot {ITEM.STATE.ERROR}), {TRIGGER.ID}, {EVENT.ID} un user macros.

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 un pēc tam noklikšķiniet uz pogas *Masveida atjaunināšana*.

Pogas

Poga labajā pusē piedāvā šādu opciju:

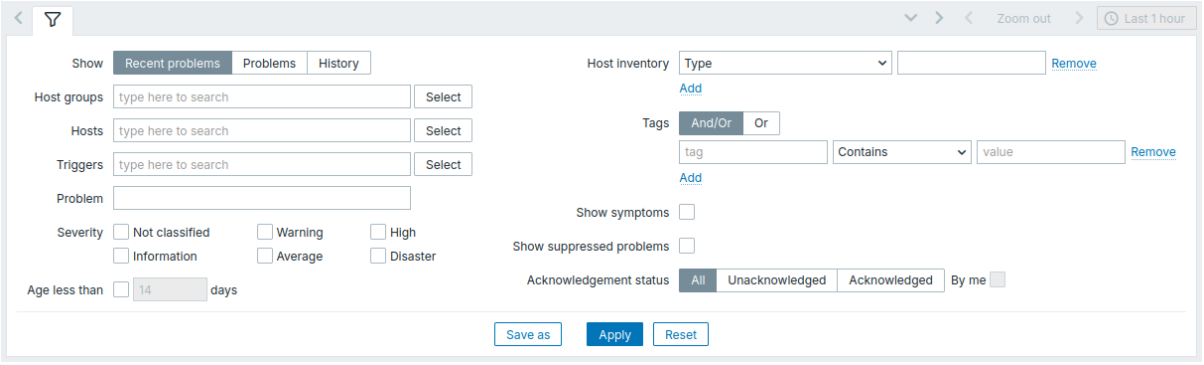
Export to CSV	Eksportēt saturu no visām lapām CSV failā.

Skata režīma pogas, kas ir kopīgas visām sadaļām, ir aprakstītas lapā Monitoring.

Filtra izmantošana

Varat izmantot filtru, lai parādītu tikai tos trigerus, kas jūs interesē. Lai uzlabotu meklēšanas veiktspēju, dati tiek meklēti ar neatrisinātiem makro.

Filtrs atrodas virs tabulas. Iecienītākos filtra iestatījumus var saglabāt kā cilnes un pēc tam ātri atvērt, noklikšķinot uz cilnēm virs filtra.

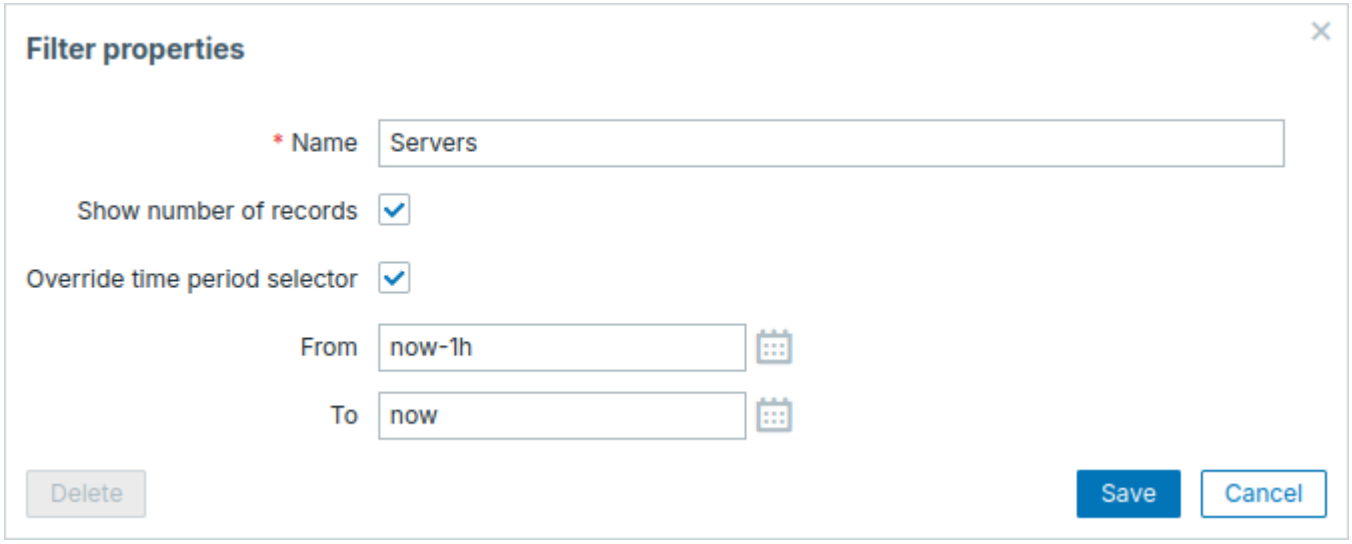


Parameter	Description
Show	Filtrēt pēc problēmas statusa: **Nesenās problēmas** - tiek parādītas neatrisinātas un nesen atrisinātas problēmas (noklusējums) **Problēmas** - tiek parādītas neatrisinātas problēmas **Vēsture** - tiek parādīta visu notikumu vēsture
Host groups	Filtrēt pēc vienas vai vairākām hostu grupām. Norādot vecāko hostu grupu, netieši tiek atlasītas visas ligzdotās hostu grupas.
Hosts	Filtrēt pēc viena vai vairākiem hostiem.
Triggers	Filtrēt pēc viena vai vairākiem trigeriem.
Problem	Filtrēt pēc problēmas nosaukuma.
Severity	Filtrēt pēc trigera (problēmas) nopietnības.
Age less than	Filtrēt pēc tā, cik sena ir problēma.
Host inventory	Filtrēt pēc inventāra tipa un vērtības.
Tags	Filtrēt pēc notikuma taga nosaukuma un vērtības. Ir iespējams iekļaut, kā arī izslēgt konkrētus 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, nav reģistrjutīga) **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, nav reģistrjutīga) Nosacījumiem ir divi aprēķina tipi: **And/Or** - jāizpildās visiem nosacījumiem, nosacījumi ar vienādu taga nosaukumu tiks grupēti pēc Or nosacījuma **Or** - pietiek, ja ir izpildīts viens nosacījums Filtrējot šeit norādītie tagi vispirms tiks parādīti kopā ar problēmu, ja vien tos nepārraksta zemāk esošais saraksts *Tag display priority*.
Show symptoms	Atzīmējiet izvēles rūtiņu, lai savā rindā parādītu problēmas, kas klasificētas kā simptomi.
Show suppressed problems	Atzīmējiet izvēles rūtiņu, lai parādītu problēmas, kas citādi tiktu slāpētas un paslēptas hosta apkopes dēļ vai ko lietotājs manuāli ir slāpējis.
Acknowledgement status	Filtrēt, lai parādītu visas problēmas, tikai neapstiprinātās problēmas vai tikai apstiprinātās problēmas. Atzīmējiet papildu izvēles rūtiņu, lai filtrētu tās problēmas, kuras jebkad esat apstiprinājis jūs.

Izlases filtru cilnes

Bieži izmantotus filtra parametru kopumus var saglabāt cilnēs.

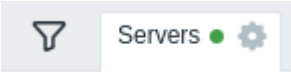
Lai saglabātu jaunu filtra parametru kopumu, atveriet galveno cilni, konfigurējiet filtra iestatījumus un pēc tam noklikšķiniet uz pogas *Save as*. Jaunā uznirstošajā logā definējiet *Filter properties*.



Parameter	Description
Name	Filtra nosaukums, kas tiks rādīts cilņu sarakstā.
Show number of records	Atzīmējiet, ja vēlaties, lai problēmu skaits tiktu rādīts blakus cilnes nosaukumam.
Override time period selector	Atzīmējiet, lai šim filtra kopumam iestatītu noteiktu noklusējuma laika periodu. Ja šī opcija ir iestatīta, šīs cilnes laika periodu varēsiet mainīt tikai, atjauninot filtra iestatījumus. Cilnēm bez pielāgota laika perioda laika diapazonu var mainīt, noklikšķinot uz laika atlasītāja pogas 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 atzīmēts *Set custom time period*.

Pēc saglabāšanas filtrs tiek izveidots kā nosaukta filtra cilne un nekavējoties aktivizēts.

Lai rediģētu esoša filtra īpašības, noklikšķiniet uz zobrata simbola blakus aktīvās cilnes nosaukumam.



Piezīmes:

- Lai paslēptu filtra apgabalu, noklikšķiniet uz pašreizējās cilnes nosaukuma. Lai filtra apgabalu atvērtu vēlreiz, vēlreiz noklikšķiniet uz aktīvās cilnes nosaukuma.
- Tiek atbalstīta navigācija ar tastatūru: izmantojiet bulttaustiņus, lai pārslēgtos starp cilnēm, un nospiediet *Enter*, lai atvērtu.
- Lai pārslēgtos starp saglabātajiem filtriem, var izmantot pogas pa kreisi/pa labi virs filtra. Alternatīvi, lejupvērstā poga atver nolaižamo izvēlni ar visiem saglabātajiem filtriem, un jūs varat noklikšķināt uz vajadzīgā.
- Ja saglabāta filtra iestatījumi ir mainīti (bet nav saglabāti), aiz filtra nosaukuma tiek rādīts zaļš punkts. Lai atjauninātu filtru atbilstoši jaunajiem iestatījumiem, noklikšķiniet uz pogas *Update*, kas tiek rādīta pogas *Save as* vietā.
- Tabulas kolonnu izkārtojums tiek saglabāts katrai filtra cilnei atsevišķi. Lai pielāgotu saglabāta filtra izkārtojumu, vispirms atveriet attiecīgo filtra cilni un pēc tam pielāgojiet kolonnas.
- Filtra cilnes, filtra iestatījumi un tabulas kolonnu izkārtojums katrai cilnei tiek saglabāti lietotāja profilā un atjaunoti, kad lietotājs atkal atver lapu.

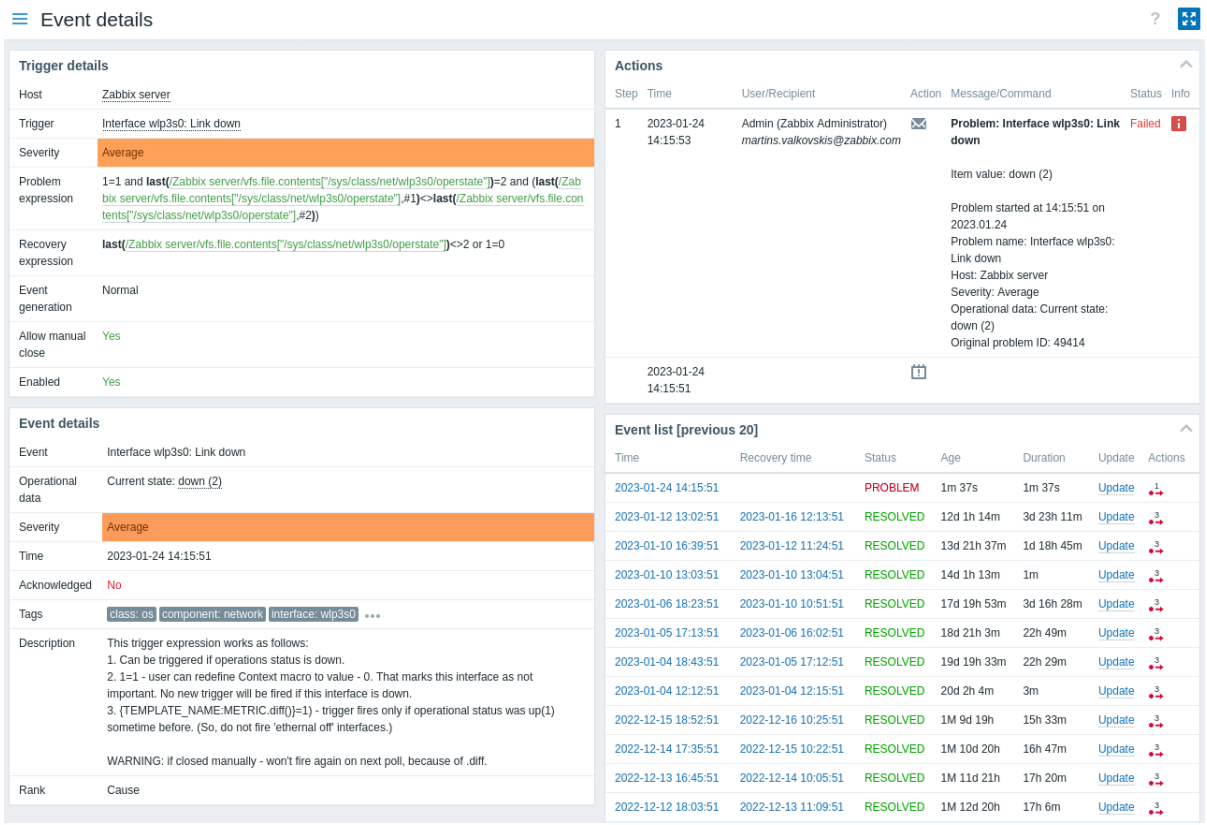
Note:
Lai kopīgotu filtrus, nokopējiet un nosūtiet citiem aktīvā filtra URL. Pēc šī URL atvēršanas citi lietotāji varēs saglabāt šo parametru kopumu kā pastāvīgu filtru savā Zabbix kontā.
 Skatiet arī: Page parameters.

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.

Detaļu skatīšana

Laiki, kad problēma sākās un tika atjaunota sadaļā *Monitoring* > *Problems*, ir saites. Noklikšķinot uz tām, tiek atvērta notikuma papildu informācija.



Notikuma detaļās var redzēt, vai notikums tika slāpēts apkopes dēļ vai arī to manuāli slāpēja lietotājs.

Ņemiet vērā, ka problēmas smagums var atšķirties trigerim un problēmas notikumam - ja tas ir atjaunināts problēmas notikumam, izmantojot *Update problem* ekrānu.

Darbību sarakstā aktivitātes tipa apzīmēšanai tiek izmantotas šādas ikonas:

-  - problēmas notikums ģenerēts.
-  - ziņojums ir nosūtīts.
-  - problēmas notikums apstiprināts.
-  - problēmas notikums nav apstiprināts.
-  - pievienots komentārs.
-  - problēmas smagums ir palielināts (piem., Information > Warning).
-  - problēmas smagums ir samazināts (piem., Warning > Information).
-  - problēmas smagums ir mainīts, bet atgriezies sākotnējā līmenī (piem., Warning > Information > Warning).
-  - izpildīta attālināta komanda.
-  - problēmas notikums ir atjaunojies.
-  - problēma ir manuāli slēgta.
-  - problēma ir slāpēta.
-  - problēmas slāpēšana ir noņemta.
-  - problēma ir slāpēta hosta apkopes dēļ.
-  - problēmas slāpēšana ir noņemta pēc hosta apkopes.
-  - problēma ir pārveidota par simptomu problēmu.
-  - problēma ir pārveidota par cēloņa problēmu.

Cēloņa un simptoma 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.

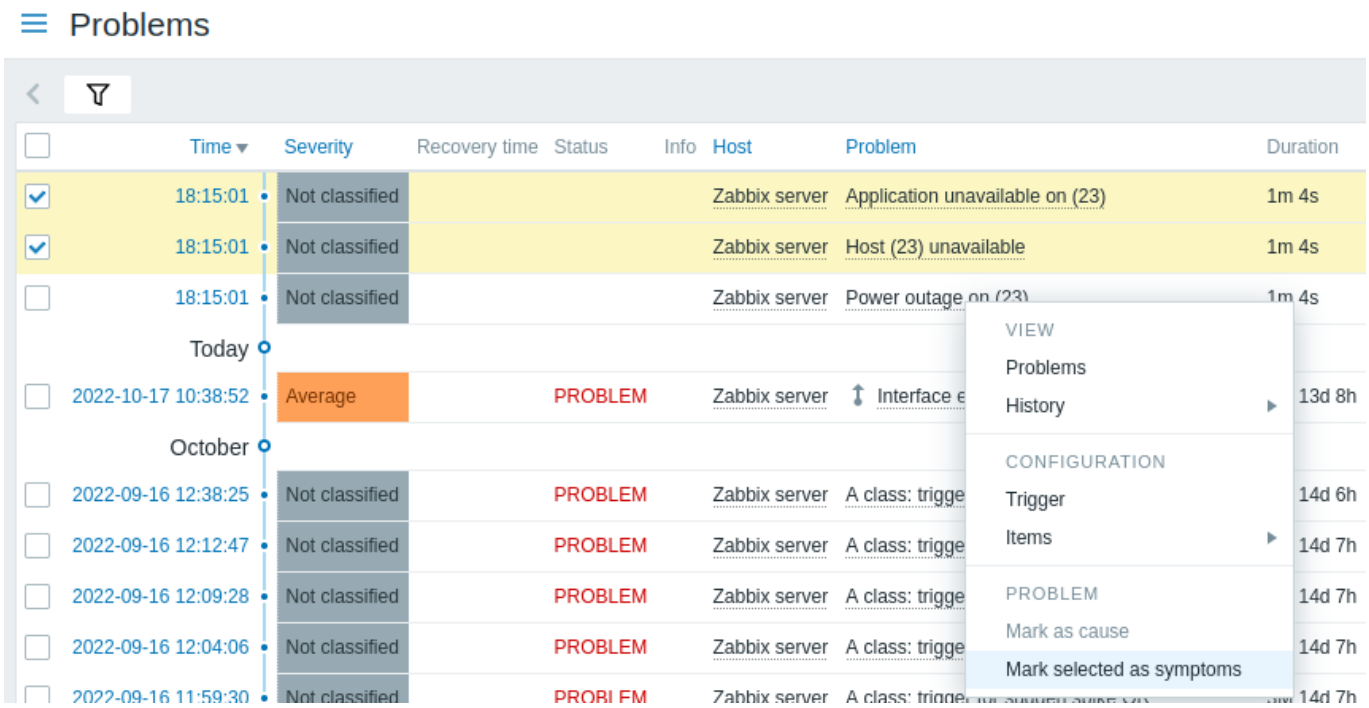
Tikai cēloņa problēmas tiek ieskaitītas kopējā problēmu skaitā kartēs, paneļa logrīkos, piemēram, *Problēmas pēc svarīguma* vai *Problēmu hosti*, utt. Tomēr problēmu ranžēšana neietekmē pakalpojumus.

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 dodieties uz cēloņa problēmu un tās konteksta izvēlnē noklikšķiniet uz opcijas *Atzīmēt atlasītās kā simptomus*.



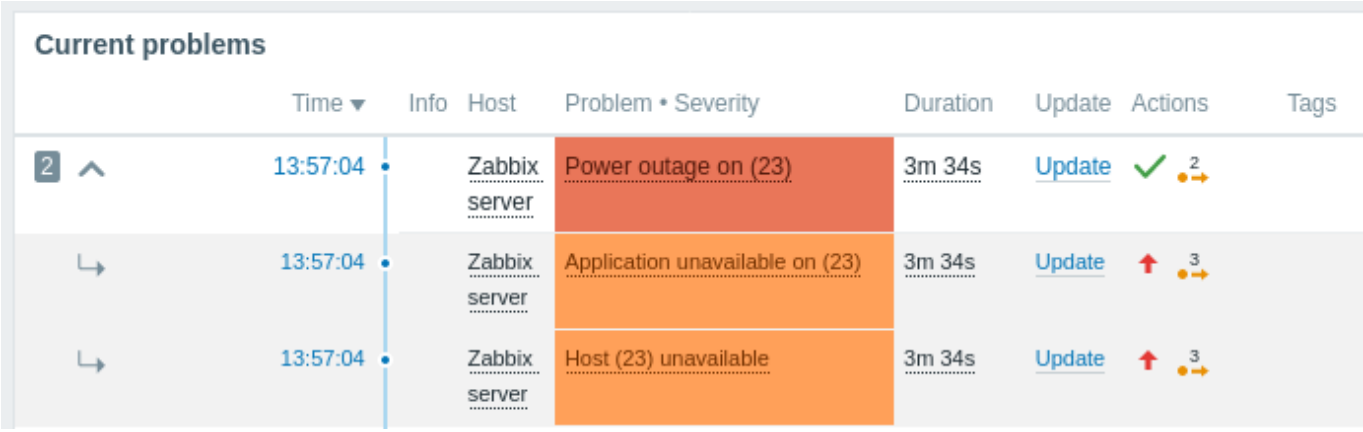
Pēc tam serveris atjaunināsatlasītās problēmas, pārveidojot tās par cēloņa problēmas simptomu problēmām.

Kamēr problēmas statuss tiek atjaunināts, tas tiek attēlots vienā no diviem veidiem:

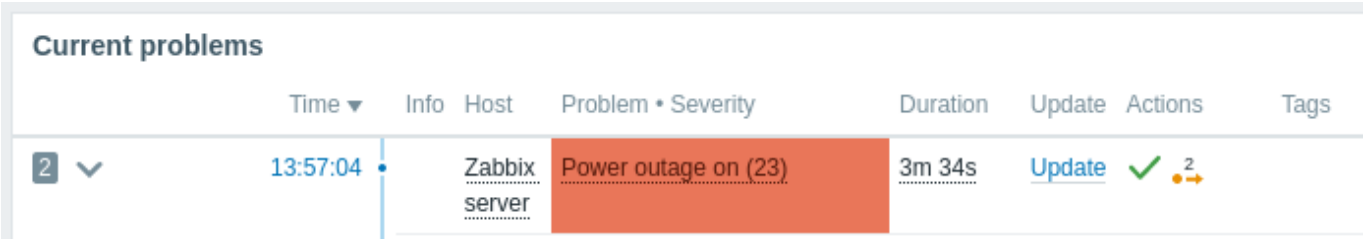
- Statusa kolonnā tiek rādīts mirgojošs statuss "UPDATING";
- Informācijas kolonnā tiek rādīta mirgojoša vai ikona (tas ir spēkā, ja filtrā ir atlasītas tikai *Problēmas*, un tādēļ Statusa kolonna netiek rādīta).

Attēlošana

Simptoma problēmas tiek attēlotas zem cēloņa problēmas un attiecīgi atzīmētas *Monitoring* > *Problems* (un *Problems* paneļa logrīkā) - ar ikonu, mazāku fontu un atšķirīgu fonu.



Sakļautā skatā redzama tikai cēloņa problēma; simptoma problēmu esamību norāda skaitlis rindas sākumā un ikona skata izvēršanai.



Ir iespējams arī papildus attēlot simptoma problēmas parastā fontā un atsevišķā rindā. Lai to izdarītu, filtra iestatījumos vai logrīka konfigurācijā atlasiet *Show symptoms*.

Atgriešana uz cēloņa problēmu

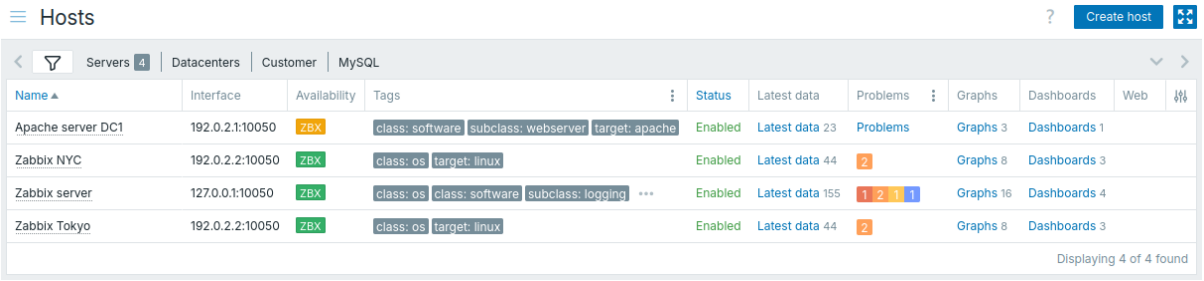
Simptoma problēmu var atgriezt atpakaļ uz cēloņa problēmu. Lai to izdarītu, vai nu:

- konteksta izvēlnē simptomu problēmai noklikšķiniet uz opcijas *Atzīmēt kā cēloni*;
- problēmas atjaunināšanas ekrānā atzīmējiet opciju *Pārveidot par cēloni* un noklikšķiniet uz *Atjaunināt* (šī opcija darbosies arī tad, ja ir atlasītas vairākas problēmas).

Hosts

Pārskats

Sadaļa *Monitoring* > *Hosts* parāda pilnu uzraudzīto hostu sarakstu ar detalizētu informāciju par hosta interfeisu, pieejamību, tagiem, pašreizējām problēmām, statusu (iespējots/atspējots) un saitēm, kas ļauj ērti pāriet uz hosta jaunākajiem datiem, problēmu vēsturi, grafikiem, paneļiem un tīmekļa scenārijiem.



Hosta informācija tiek rādīta pielāgojamā tabulā. Poga *Customize table* tabulas galvenē ļauj parādīt, paslēpt vai pārkārtot kolonnas, kā arī atiestatīt izkārtojumu uz noklusējumu. Varat arī mainīt katras kolonnas platumu, velkot tās labo malu; veiciet dubultklikšķi uz malas, lai atiestatītu. Visas pielāgošanas tiek piemērotas nekavējoties un vienmēr tiek saglabātas pašreizējam lietotājam aktīvajā filtra cilnē (noklusējuma vai saglabātajā).

Pieejamas šādas kolonnas:

Column	Description
Name	Redzamais hosta nosaukums. Noklikšķinot uz nosaukuma, tiek atvērta hosta izvēlne. Oranža uzgriežņu atslēgas ikona aiz nosaukuma norāda, ka šis hosts ir apkopē. Noklikšķiniet uz kolonnas galvenes, lai kārtotu hostus pēc nosaukuma augošā vai dilstošā secībā.

Column	Description
Interface	Tiek parādīts hosta galvenais interfeiss.
Availability	Tiek parādīta hosta pieejamība katram konfigurētajam interfeisam. Pieejamības ikonas attēlo hosta interfeisa pašreizējo statusu Zabbix serverī. Tāpēc, ja jūs atspējojat hostu lietotāja saskarnē, tā pieejamība tiks atjaunināta pēc tam, kad Zabbix serveris būs sinhronizējis konfigurācijas izmaiņas. Līdzīgi, ja jūs iespējojat hostu, tā pieejamība tiks atjaunināta pēc tam, kad Zabbix serveris būs sinhronizējis konfigurācijas izmaiņas un aptaujājis hostu. Pieejamības ikonas attēlo tikai tos interfeisu tipus (Agent, SNMP, IPMI, JMX), kas ir konfigurēti. Novietojot kursoru virs ikonas, tiek parādīts uznirstošais logs ar visu tā paša tipa interfeisu sarakstu, detalizētu informāciju, statusu un kļūdām. Agent interfeisam uznirstošajā logā tiek parādīti interfeisi (pasīvie) un aktīvās pārbaudes. Ja hostam ir tikai aktīvās pārbaudes, tiek parādīta Agent interfeisa ikona pat tad, ja hostam nav konfigurēts Agent interfeiss. Kolonna ir tukša hostiem bez interfeisiem. Atsevišķa hosta interfeisa statuss tiek noteikts pēc savienojuma starp iespējotu vienumu, kas izmanto šo interfeisu, un hostu. Statuss var būt: **Available** - savienojums ar hostu bija veiksmīgs; **Not available** - savienojums ar hostu nebija veiksmīgs (noildze, ugunsmūra problēmas utt.); **Unknown** - savienojums ar hostu nav mēģināts vai rezultāts nav zināms. Papildu informāciju par to, kā Zabbix serveris nosaka interfeisa statusu, skatiet Unknown interface status un Unreachable/unavailable host interface settings. Visu viena tipa hosta interfeisu (Agent, SNMP, IPMI, JMX) statuss tiek noteikts pēc tiem interfeisiem, kurus izmanto vismaz viens iespējots vienums. Statusu norāda ikonas krāsa: **Green** - visi interfeisi ir pieejami; **Yellow** - vismaz viens interfeiss nav pieejams un vismaz viens ir pieejams vai nezināms; **Red** - visi interfeisi nav pieejami; **Gray** - vismaz viens interfeiss ir nezināms, bet neviens nav nepieejams. **Aktīvo pārbaužu pieejamība.** Ja hostā ir iespējota vismaz viena aktīvā pārbaude, aktīvās pārbaudes arī ietekmē kopējo Agent interfeisa pieejamību, kā aprakstīts iepriekš. Lai noteiktu aktīvās pārbaudes pieejamību, aģenta aktīvo pārbaužu pavedienā tiek sūtīti sirdspukstu ziņojumi. Sirdspukstu ziņojumu biežumu kontrolē parametrs `HeartbeatFrequency` Zabbix aģenta vai agent 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.
Tags	Tiek parādīti hosta tagi. Novietojot kursoru virs vai noklikšķinot uz ••• trīs punktu ikonas aiz parādītajiem tagiem, tiek atvērts uznirstošais logs ar visiem tagiem. Noklikšķinot uz pogas ⋮ *Options* kolonnas galvenē, varat iestatīt parādāmo tagu skaitu (1–3), izvēlēties, kā tiek attēloti tagu nosaukumi (pilni, saīsināti vai netiek rādīti), iestatīt tagu attēlošanas prioritāti vai dublēt kolonnu (dublētās kolonnas var pārdēvēt vai dzēst).
Tag value	Tiek parādītas hosta tagu vērtības. Noklikšķinot uz pogas ⋮ *Options* kolonnas galvenē, varat pārdēvēt kolonnu, izvēlēties, kuras tagu vērtības rādīt, vai dublēt kolonnu (dublētās kolonnas var pārdēvēt vai dzēst).
Status	Hosta statuss - *Enabled* vai *Disabled*. Noklikšķiniet uz kolonnas galvenes, lai kārtotu hostus pēc statusa augošā vai dilstošā secībā.
Latest data	Noklikšķinot uz saites, tiks atvērta sadaļa *Monitoring* > *Latest data* ar visiem jaunākajiem datiem, kas savākti no hosta. Pelēkā krāsā tiek parādīts vienumu skaits ar jaunākajiem datiem.
Problems	Tiek parādīts atvērto problēmu skaits hostam, sakārtots pēc smaguma pakāpes. Katra problēmas ikonas krāsa norāda smaguma pakāpi, bet skaitlis norāda problēmu skaitu attiecīgajā smaguma pakāpē. Noklikšķinot uz ikonas, tiek atvērta *Monitoring* > *Problems* šim hostam; ja hostam nav problēmu, tā vietā tiek parādīta saite *Problems*. Noklikšķinot uz pogas ⋮ *Options* kolonnas galvenē, varat piekļūt opcijai *Show suppressed problems*, kas ietver arī problēmas, kuras citādi būtu slēptas hosta apkopes vai atsevišķas problēmu slāpēšanas dēļ.

Column	Description
Graphs	Noklikšķinot uz saites, tiks parādīti hostam konfigurētie grafiki. Pelēkā krāsā tiek parādīts grafiku skaits. Ja hostam nav grafiku, saite netiek rādīta.
Dashboards	Noklikšķinot uz saites, tiks parādīti hostam konfigurētie paneļi. Pelēkā krāsā tiek parādīts paneļu skaits. Ja hostam nav paneļu, saite netiek rādīta.
Web	Noklikšķinot uz saites, tiks parādīti hostam konfigurētie tīmekļa scenāriji. Pelēkā krāsā tiek parādīts tīmekļa scenāriju skaits. Ja hostam nav tīmekļa scenāriju, saite netiek rādīta.
Custom text	Tiek parādīts pielāgots teksts, makro vērtības vai abu kombinācija. Noklikšķinot uz pogas ⋮ *Options* kolonnas galvenē, varat pārdēvēt kolonnu, ievadīt pielāgoto tekstu vai makro, vai dublēt kolonnu (dublētās kolonnas var pārdēvēt vai dzēst). Atbalstītie makro: `{HOST.*}` (izņemot `{HOST.METADATA}`), `{INVENTORY.*}` un lietotāja makro.

Pogas

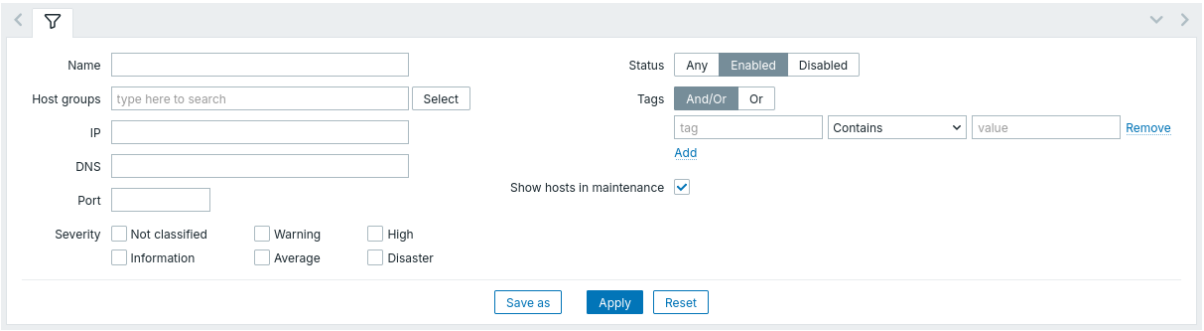
Izveidot hostu ļauj izveidot jaunu hostu. Šī poga ir pieejama tikai administratoriem un superadministratoriem.

Skata režīma pogas, kas ir kopīgas visām sadaļām, ir aprakstītas lapā Uzraudzība.

Filtra izmantošana

Varat izmantot filtru, lai parādītu tikai tos hosts, kas jūs interesē. Lai uzlabotu meklēšanas veiktspēju, dati tiek meklēti ar neatrisinātiem makro.

Filtrs atrodas virs tabulas. Ir iespējams filtrēt hosts pēc nosaukuma, hostu grupas, IP vai DNS, interfeisa porta, tagiem, problēmas smaguma pakāpes, statusa (iespējots/atspējots/jebkurš); varat arī izvēlēties, vai rādīt hosts, kas pašlaik ir apkopē.



Parameter	Description
Name	Filtrēt pēc redzamā hosta nosaukuma.
Host groups	Filtrēt pēc vienas vai vairākām hostu grupām. Norādot vecāko hostu grupu, netieši tiek atlasītas visas ligzdotās 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 rādītas visu smaguma pakāpju problēmas. Problēmas tiek rādītas, ja tās nav apspiestas.
Status	Filtrēt pēc hosta statusa.

Parameter	Description
Tags	Filtrēt pēc hosta taga nosaukuma un vērtības. Hosts var filtrēt gan pēc hosta līmeņa tagiem, gan pēc tagiem no visām saistītajām veidnēm, tostarp ligzdotajām veidnēm. Iespējams iekļaut, kā arī izslēgt konkrētus tagus un tagu vērtības. Var iestatīt vairākus nosacījumus. Taga nosaukuma atbilstība vienmēr ir reģistrjutīga. Katrai 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, nav reģistrjutīga) **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, nav reģistrjutīga) Nosacījumiem ir pieejami divi aprēķina tipi: **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 parādītu hosts, kas ir apkopē (pēc noklusējuma tiek rādīti).

Filtra saglabāšana

Izlases filtra iestatījumus var saglabāt kā cilnes un pēc tam ātri piekļūt tiem, noklikšķinot uz attiecīgās cilnes virs filtra.

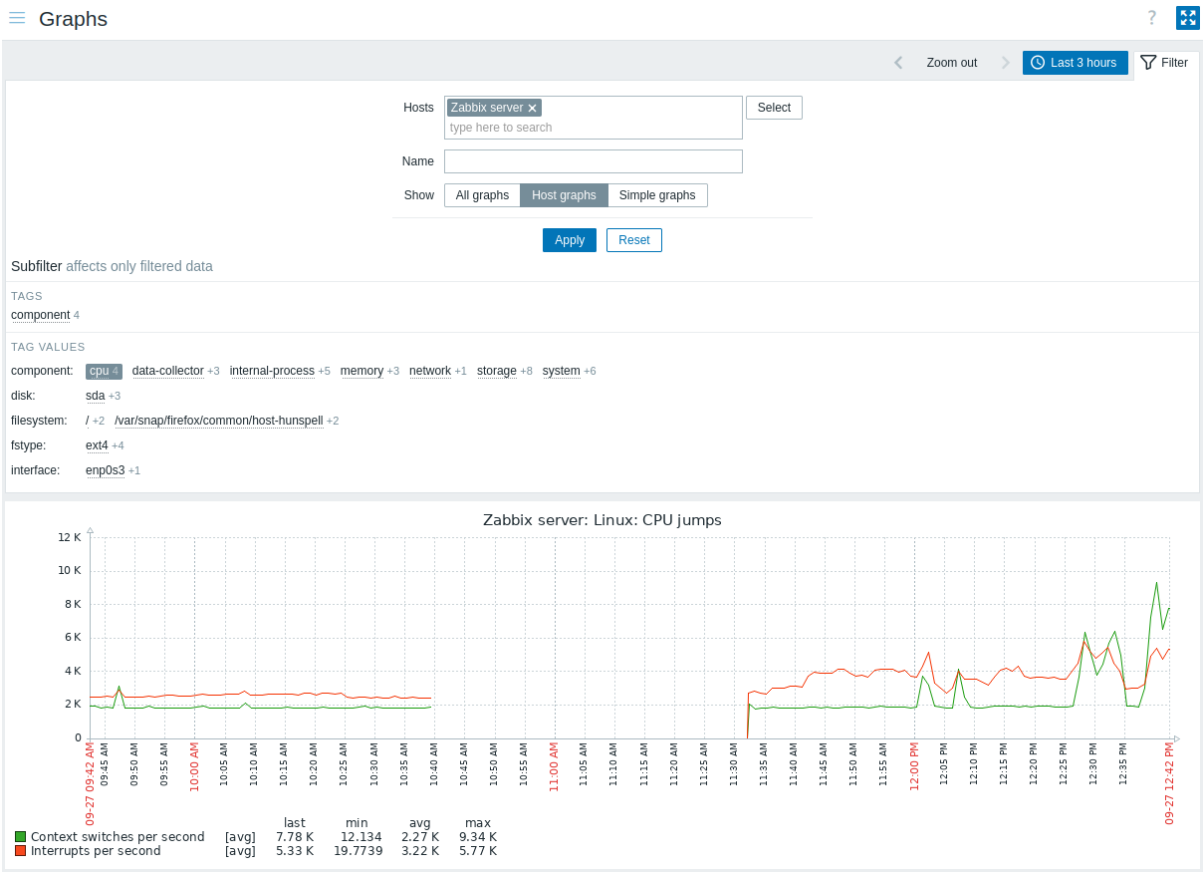
Skatiet vairāk informācijas par filtru saglabāšanu.

Grafiki

Pārskats

Hostu grafikus var atvērt no *Monitoring* > *Hosts*, noklikšķinot uz Graphs attiecīgajam hostam.

Var tikt parādīts jebkurš pielāgotais grafiks, kas ir konfigurēts hostam, kā arī jebkurš vienkāršais grafiks.



Grafiki tiek kārtoti pēc:

- grafika nosaukuma (pielāgotie grafiki)
- vienuma nosaukuma (vienkāršie grafiki)

Grafikiem atspējotiem hostiem arī var piekļūt.

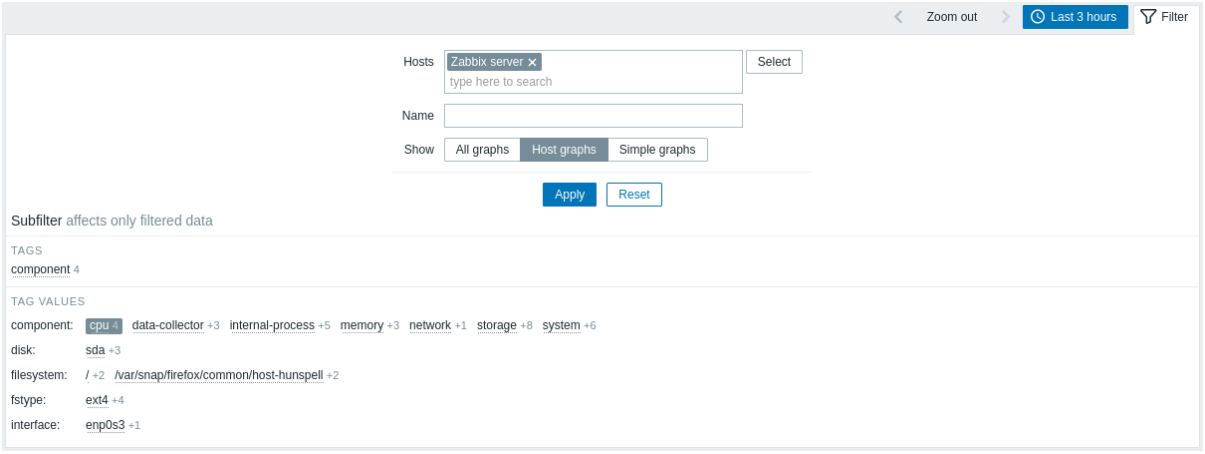
Laika perioda atlasītājs

Pievērsiet uzmanību laika perioda atlasītājam virs diagrammas. Tas ļauj atlasīt bieži nepieciešamos periodus ar vienu peles klikšķi.

Skatiet arī: *Laika perioda atlasītājs*

Filtra izmantošana

Lai skatītu konkrētu grafiku, atlasiet to filtrā. Filtrs ļauj norādīt hosts, grafika nosaukumu un opciju *Rādīt* (visus grafikus/hostu grafikus/vienkāršos grafikus).



Ja filtrā nav atlasīts neviens hosts, grafiki netiek rādīti.

Apakšfiltra izmantošana

Apakšfiltrs ļauj papildus modificēt galvenā filtra lietoto filtrēšanu.

Tas satur noklikšķināmas saites ātrai piekļuvei saistītajām diagrammām. Diagrammas ir saistītas ar kopīgu entītiju - taga nosaukumu vai vērtību. Noklikšķinot uz taga nosaukuma/vērtības, tas tiek izcelts ar pelēku fonu, un diagrammas tiek nekavējoties filtrētas (galvenajā filtrā nav jānoklikšķina uz *Lietot*). Noklikšķinot uz cita taga nosaukuma/vērtības, tas tiek pievienots filtrēšanas rezultātiem. Vēlreiz noklikšķinot uz taga nosaukuma/vērtības, filtrēšana tiek noņemta.

Apakšfiltri tiek ģenerēti, pamatojoties uz filtrētajiem datiem, kas ir ierobežoti līdz 1000 ierakstiem. Ja apakšfiltrā vēlaties redzēt vairāk ierakstu, jāpalielina parametra *Meklēšanas un filtrēšanas rezultātu ierobežojums* vērtība (*Administrēšana* > *Vispārīgi* > *Lietotāja saskarne*).

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 parādīto entītiju skaits ir ierobežots līdz 100. Ja entītiju ir vairāk, beigās tiek parādīta trīs punktu ikona; uz tās nevar noklikšķināt. Vertikālie saraksti (piemēram, tagi ar to vērtībām) ir ierobežoti līdz 20 ierakstiem. Ja ierakstu ir vairāk, tiek parādīta trīs punktu ikona; uz tās nevar noklikšķināt.

Skaitlis blakus katrai noklikšķināmajai entītijai norāda diagrammu skaitu, kas tai ir galvenā filtra rezultātos.

Kad ir atlasīta viena entītija, skaitļi pie citām pieejamajām entītijām tiek parādīti ar pluszīmi, kas norāda, cik diagrammu var pievienot pašreizējai atlasei.

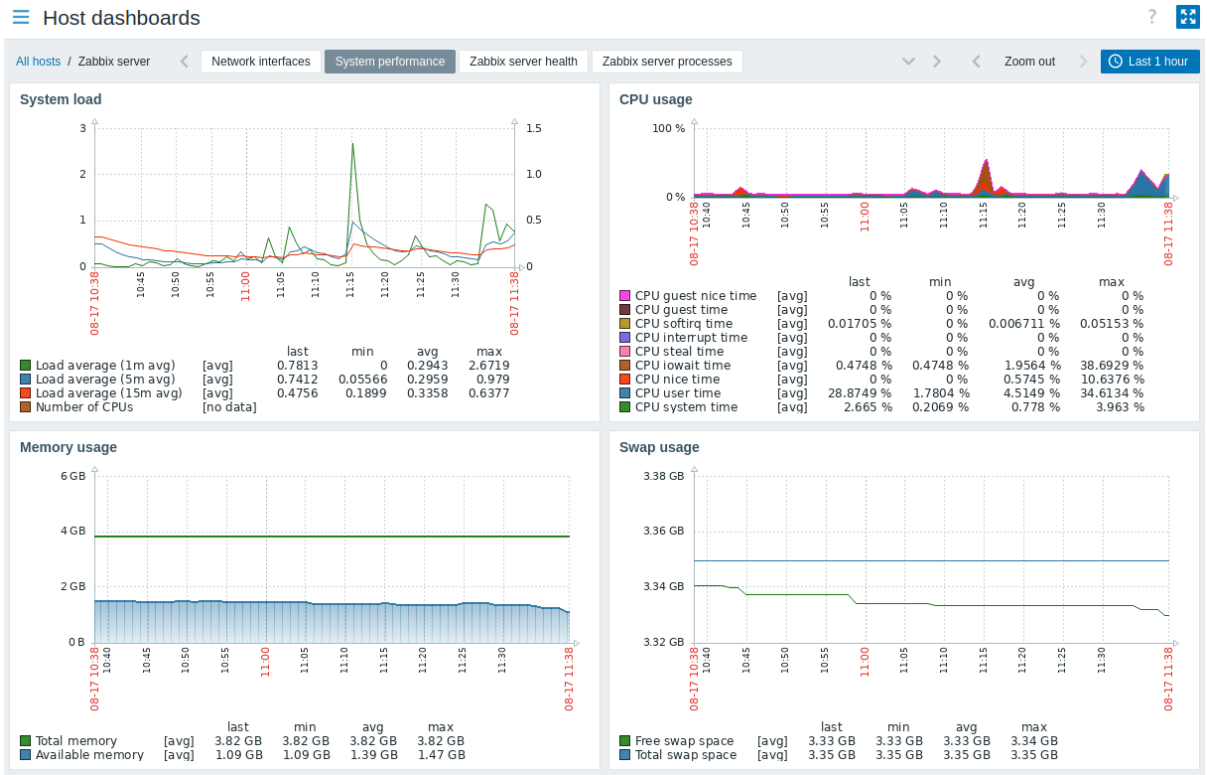
Pogas

Skata režīma pogas, kas ir kopīgas visām sadaļām, ir aprakstītas lapā Monitoring.

Hostu informācijas paneļi

Pārskats

Hosts informācijas paneļi izskatās līdzīgi globālajiem informācijas paneļiem; tomēr hosts informācijas paneļiem nav īpašnieka, un tajos tiek rādīti dati tikai par atlasīto hostu.



Skatot hosts informācijas paneļus, varat pārslēgties starp konfigurētajiem informācijas paneļiem, noklikšķinot uz:

- informācijas paneļu cilnēm;
-  bultiņu pogām zem galvenes;
-  bultiņas pogas zem galvenes, kas parādīs pilnu pieejamo hosts informācijas paneļu sarakstu.

Lai pārslēgtos uz sadaļu *Monitoring* > *Hosts*, augšējā kreisajā stūrī zem galvenes noklikšķiniet uz navigācijas saites *All hosts*.

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.

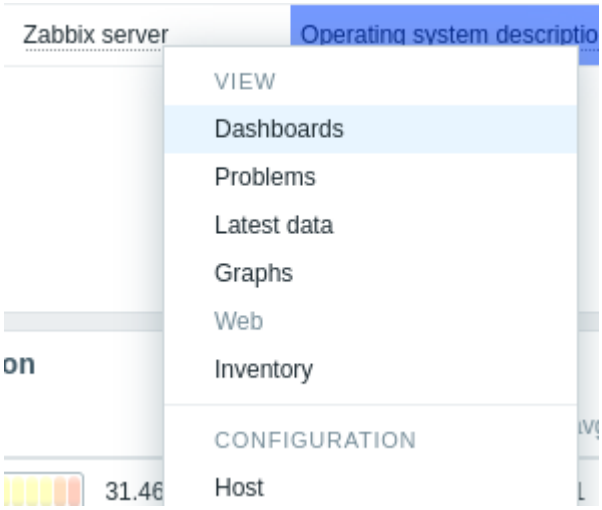
Hostu paneļu logrīkus arī var konfigurēt tikai veidnes līmenī, izņemot atsvaidzināšanas intervāla maiņu. Turklāt hostu paneļu logrīkus var kopēt tikai uz citiem hostu paneļiem tās pašas veidnes ietvaros. Ņemiet vērā, ka logrīkus no globālajiem paneļiem *nevar* kopēt uz hostu paneļiem.

Note:
Pirms Zabbix 5.2 hostu paneļi bija hostu ekrāni. Importējot vecāku veidni, kas satur ekrānus, ekrānu imports tiks ignorēts.

Piekļuve

Hostu informācijas paneļiem var piekļūt:

- pēc hosta nosaukuma meklēšanas globālajā meklēšanā (meklēšanas rezultātos noklikšķiniet uz saites *Dashboards*);
- pēc hosta nosaukuma noklikšķināšanas sadaļā *Inventory* > *Hosts* (hosta pārskatā noklikšķiniet uz saites *Dashboards*);
- no hosta izvēlnes, noklikšķinot uz *Dashboards*.

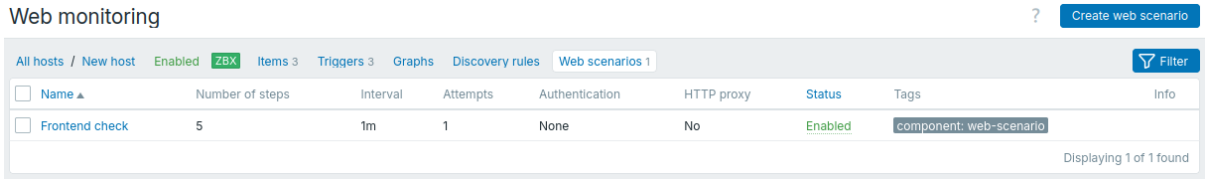


Ņemiet vērā, ka hostu informācijas paneļiem *nevar* tieši piekļūt sadaļā *Dashboards*, kas ir paredzēta globālajiem informācijas paneļiem.

Tīmekļa scenāriji

Pārskats

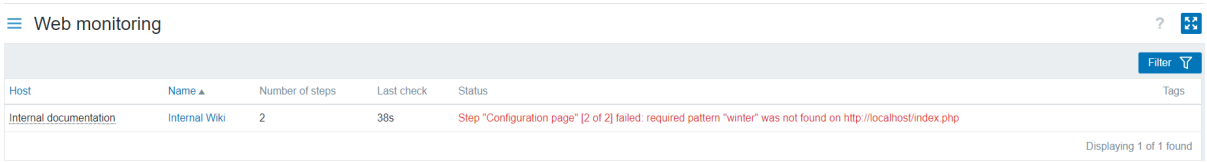
Hosts web scenārija informācijai var piekļūt no *Monitoring* > *Hosts*, noklikšķinot uz Web attiecīgajam hostam.



Noklikšķinot uz hosta nosaukuma, tiek atvērta hosta izvēlne. Ir pieejami arī atspējotu hostu dati. Atspējota hosta nosaukums ir attēlots sarkanā krāsā.

Web scenārija statuss var būt:

- OK (zaļš) - visas darbības ir izdevušās, un Zabbix ir savācis scenārija datus.
- Kļūda (sarkans) - viena vai vairākas darbības ir neveiksmīgas (skatiet kļūdas ziņojumu, lai noteiktu, kuri parametri, iespējams, ir jālabo).
- Nezināms (pelēks) - vēl nav pieejams nevienas darbības statuss vai arī dati vēl nav savākti.



Maksimālais vienā lapā attēlojamo scenāriju skaits ir atkarīgs no lietotāja profila *Rows per page* iestatījuma [/manual/web_interface/user_profile profile).

Pēc noklusējuma tiek attēlotas tikai vērtības, kas ietilpst pēdējo 24 stundu laikā. Šis ierobežojums ir ieviests, lai uzlabotu sākotnējo ielādes laiku lielām jaunāko datu lapām. Šo laika periodu var pagarināt, mainot *Max history display period* parametra vērtību izvēlnes sadaļā *Administration > General > GUI*.

Scenārija nosaukums ir saite uz detalizētāku statistiku par to:



Filtra izmantošana

Lapa parāda visu atlasītā hosta tīmekļa scenāriju sarakstu. Lai skatītu tīmekļa scenārijus citam hostam vai hostu grupai, neatgriežoties uz *Monitoring* > *Hosts* lapu, atlasiet šo hostu vai grupu filtrā. Varat arī filtrēt scenārijus pēc tagiem.

Pogas

Skata režīma pogas, kas ir kopīgas visām sadaļām, ir aprakstītas lapā Monitoring.

Jaunākie dati

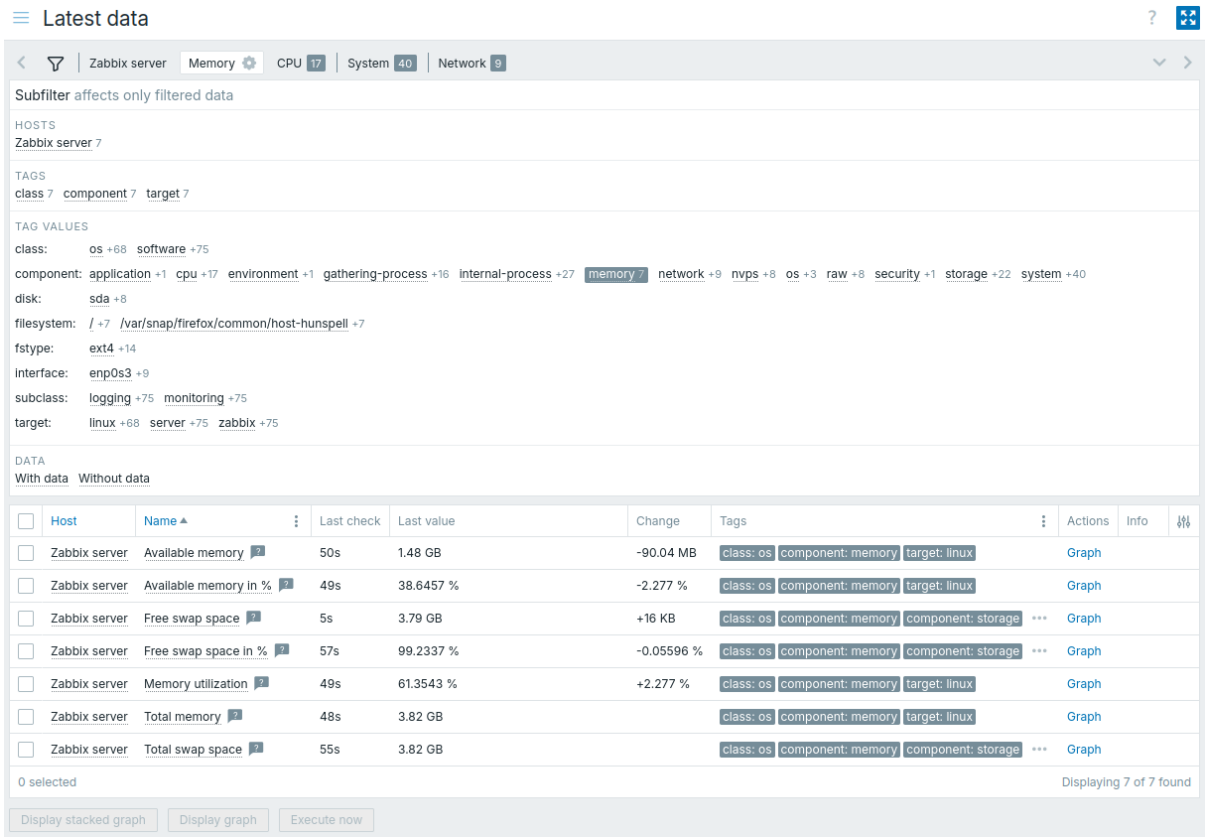
Pārskats

Sadaļa *Monitoring* > *Latest data* parāda jaunākās vērtības, kas iegūtas ar vienumiem.

Šī sadaļa satur šādus elementus:

- Filtrs
- Apakšfiltrs
- Vienumu saraksts

Note:
Apakšfiltrs un vienumu saraksts tiek parādīti tikai tad, ja filtrs ir iestatīts un ir rezultāti, ko parādīt.



Jaunākās vērtības detalizēti tiek parādītas pielāgojamā tabulā. Poga *Pielāgot tabulu* tabulas galvenē ļauj parādīt, paslēpt vai pārkārtot kolonnas, kā arī atiestatīt izkārtojumu uz noklusējuma iestatījumiem. Varat arī mainīt katras kolonnas platumu, velkot tās labo malu; veiciet dubultklikšķi uz malas, lai atiestatītu. Visas pielāgošanas tiek piemērotas nekavējoties un vienmēr tiek saglabātas pašreizējam lietotājam aktīvajā filtra cilnē (noklusējuma vai saglabātajā).

Pieejamas šādas kolonnas:

Column	Description
Host	Hosts, kuram pieder vienums, nosaukums. Noklikšķinot uz nosaukuma, tiek atvērta hosts konteksta izvēlne. Ja hosts ir uzturēšanas režīmā, aiz hosts nosaukuma tiek parādīta oranža uzgriežņu atslēgas ikona. Ja hosts ir atspējots, hosts nosaukums tiek parādīts sarkanā krāsā. Ņemiet vērā, ka atspējotu hosts dati (tostarp grafiki un vienumu vērtību saraksti) ir pieejami sadaļā *Latest data*.
Name	Vienuma nosaukums. Noklikšķinot uz nosaukuma, tiek atvērta vienuma izvēlne. Blakus vienuma nosaukumam visiem vienumiem, kuriem ir apraksts, tiek parādīta jautājuma zīmes ikona. Novietojiet kursoru virs ikonas, lai parādītu paskaidrojuma rīka padomu ar vienuma aprakstu. Noklikšķinot uz pogas *Options* kolonnas galvenē, tiek nodrošināta piekļuve opcijai *Show item key*, kas zem katra vienuma nosaukuma parāda vienuma atslēgu.
Interval	Vienuma atjaunināšanas intervāls.
History	Vienuma vēstures glabāšanas periods.
Trends	Vienuma tendenču glabāšanas periods.
Type	Vienuma tips.
Last check	Laiks kopš pēdējās vienuma pārbaudes.
Last value	Jaunākā vienuma vērtība. Vērtības tiek parādītas, piemērojot vienību konvertēšanu un vērtību kartēšanu. Novietojiet kursoru virs vērtības, lai parādītu neapstrādātus datus (saīsinātus līdz 8192 rakstzīmēm). Binarās vērtības faktiskās vērtības vietā tiek parādītas kā vietturis (*binary value*). Pēc noklusējuma tiek parādītas tikai vērtības, kas saņemtas pēdējo 24 stundu laikā. Šis ierobežojums uzlabo sākotnējo ielādes laiku lielām jaunāko datu lapām; lai to paplašinātu, atjauniniet parametra *Max history display period* vērtību sadaļā *Administration* > *General* > *GUI*.

Column	Description
Change	Starpība starp iepriekšējo vērtību un jaunāko vērtību. Vienumiem ar atjauninašanas biežumu 1 diena vai vairāk izmaiņu apjoms nekad netiks parādīts (ar noklusējuma iestatījumu). Šādā gadījumā jaunākā vērtība vispār netiks parādīta, ja tā saņemta vairāk nekā pirms 24 stundām.
Tags	Ar vienumu saistītie tagi. Noklikšķinot uz taga, tas tiek iespējots apakšfiltrā, tādējādi filtrējot vienumu sarakstu pēc šī taga (un jebkuriem citiem apakšfiltrā atlasītajiem tagiem). Pēc noklikšķināšanas šādi tagi vairs nav klikšķināmi; lai pielāgotu filtrēšanu pēc tagiem, jāizmanto apakšfiltrs. Noklikšķinot uz pogas ⋮ *Options* kolonnas galvenē, varat iestatīt parādāmo tagu skaitu (1-3), izvēlēties, kā tiek parādīti tagu nosaukumi (pilni, saīsināti vai netiek rādīti), iestatīt tagu attēlošanas prioritāti vai dublēt kolonnu (dublētās kolonnas var pārdēvēt vai dzēst).
Tag value	Tiek parādītas vienuma tag vērtības. Noklikšķinot uz pogas ⋮ *Options* kolonnas galvenē, varat pārdēvēt kolonnu, izvēlēties, kuras tagu vērtības rādīt, vai dublēt kolonnu (dublētās kolonnas var pārdēvēt vai dzēst).
Actions	Saite uz vienuma vērtību vienkāršo grafiku/vēsturi.
Info	Papildu informācija par vienumu. Ja vienumam ir kļūdas (piemēram, tas kļuvis neatbalstīts), tiek parādīta informācijas ikona. Novietojiet kursoru virs ikonas, lai skatītu detalizētu informāciju.
Custom text	Tiek parādīts pielāgots teksts, makro vērtības vai abu kombinācija. Noklikšķinot uz pogas ⋮ *Options* kolonnas galvenē, varat pārdēvēt kolonnu, ievadīt pielāgoto tekstu vai makro, vai dublēt kolonnu (dublētās kolonnas var pārdēvēt vai dzēst). Atbalstītie makro: `{HOST.*}` (izņemot `{HOST.METADATA}`), `{INVENTORY.*}`, `{ITEM.*}` (izņemot `{ITEM.STATE.ERROR}`), un lietotāja makro.

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:

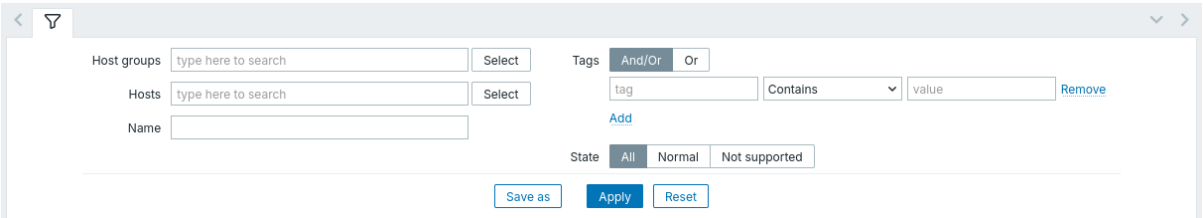
- *Rādīt sakrauto grafiku* - parādīt sakrautu ad hoc grafiku.
- *Rādīt grafiku* - parādīt vienkāršu ad hoc grafiku.
- *Izpildīt tagad* - nekavējoties izpildīt pārbaudi, lai iegūtu jaunas vienuma vērtības. Atbalstīts tikai **pasīvajām** pārbaudēm (skatiet papildu informāciju). Šī opcija ir pieejama tikai hosts ar lasīšanas un rakstīšanas piekļuvi. Piekļuve šai opcijai hostiem ar tikai lasīšanas atļaujām ir atkarīga no lietotāja lomas opcijas *Izsaukt "Izpildīt tagad" tikai lasīšanas hostos*.

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 uzlabotu meklēšanas veiktspēju, dati tiek meklēti ar neatrisinātiem makro.

Filtra ikona atrodas virs vienumu saraksta un apakšfiltra. Noklikšķiniet uz tās, lai izvērstu filtru.



Filtrs ļauj sašaurināt vienumu sarakstu pēc hostu grupas, hosts, vienuma nosaukuma, tagiem un stāvokļa. Filtrā norādot vecāku hostu grupu, tiek netieši atlasītas visas ligzdotās hostu grupas. Detalizētu informāciju par filtrēšanu pēc tagiem skatiet sadaļā *Monitoring* > *Problems*.

Filtra opcija *Show details* ļauj izvērst 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

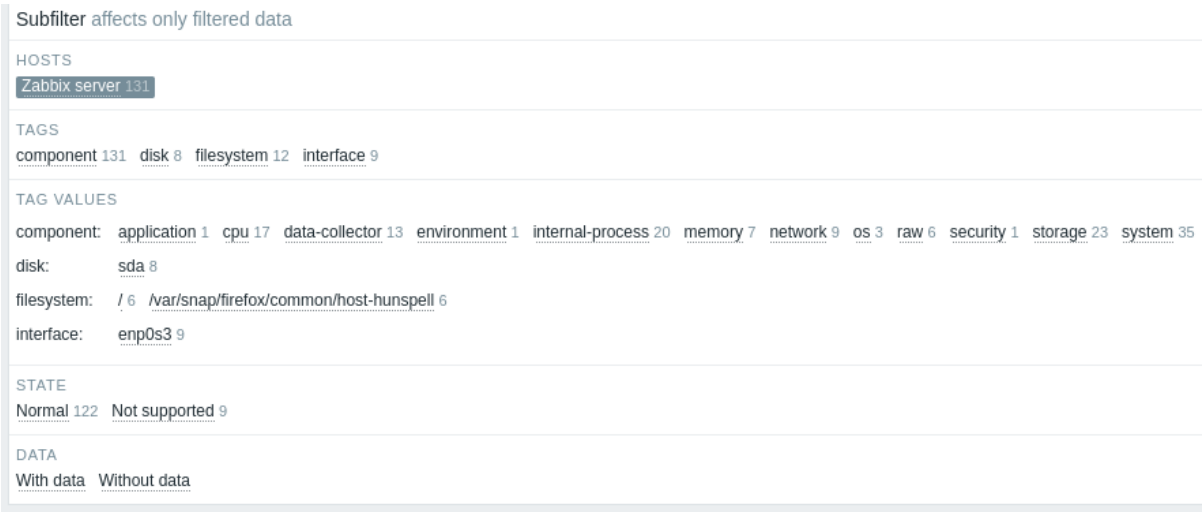
Izlases filtra iestatījumus var saglabāt kā cilnes un pēc tam ātri atvērt, noklikšķinot uz attiecīgās cilnes virs filtra.

Skatiet vairāk informācijas par filtru saglabāšanu.

Apakšfiltra izmantošana

Apakšfiltrs ļauj papildus modificēt galvenajā filtrā iestatīto filtrēšanu.

Tas satur noklikšķināmas saites ātrai piekļuvei saistītajiem vienumiem. Vienumi tiek saistīti pēc kopīgas entītijas - hosts, taga nosaukuma vai vērtības, vienuma stāvokļa vai datu statusa. Noklikšķinot uz entītijas, tā tiek izcelta ar pelēku fonu, un vienumi tiek nekavējoties filtrēti (galvenajā filtrā nav jānoklikšķina uz *Apply*). Noklikšķinot uz citas entītijas, tā tiek pievienota filtrētajiem rezultātiem. Vēlreiz noklikšķinot uz entītijas, filtrēšana tiek noņemta.



Apakšfiltri tiek ģenerēti, pamatojoties uz filtrētajiem datiem, kas ir ierobežoti līdz 1000 ierakstiem. Ja ir 20 hosti, kuriem katram ir 100 vienumu (kopā 2000 ierakstu), apakšfiltrā būs redzama tikai puse hostu. Ja vēlaties apakšfiltrā redzēt vairāk ierakstu, 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 pieejamās filtrēšanas opcijas un to skaitītāji vienmēr būtu aktuāli.

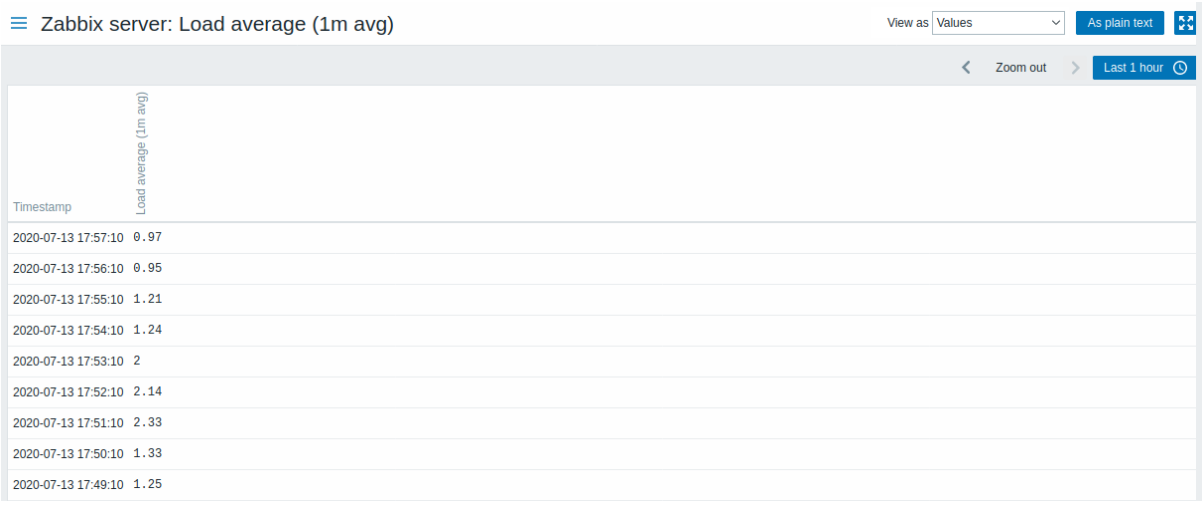
Katrai entītiju grupai (hostiem, tagiem, tagu vērtībām u. c.) tiek parādītas ne vairāk kā 10 entītiju rindas. Ja entītiju ir vairāk, šo sarakstu var izvērst, lai parādītu ne vairāk kā 1000 ierakstu (SUBFILTER_VALUES_PER_GROUP vērtība sadaļā frontend definitions), saraksta beigās noklikšķinot uz trīs punktu ikonas . Sadaļai *Tag values* sarakstu var izvērst, lai parādītu ne vairāk kā 200 tagu nosaukumu ar tiem atbilstošajām vērtībām. Ņemiet vērā, ka pēc pilnīgas izvēršanas sarakstu vairs nevar sakļaut.

Skaitlis blakus katrai noklikšķināmajai entītijai norāda tajā sagrupēto vienumu skaitu (pamatojoties uz galvenā filtra rezultātiem). Kad entītija tiek atlasīta, pie citām pieejamajām entītijām tiek parādīti skaitļi ar pluszīmi, kas norāda, cik vienumu var pievienot pašreizējai atlasei. Entītijas, kurām nav vienumu, netiek parādītas, 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 rādīta iepriekšējo vienumu vērtību vēsture.
- **Graph** - visiem skaitliskajiem vienumiem, ved uz vienkāršu grafiku. Ņemiet vērā, ka, kad grafiks tiek parādīts, augšējā labajā stūrī esošajā nolaižamajā izvēlnē ir pieejama arī iespēja pārslēgties uz *Values/500 latest values*.



Šajā sarakstā parādītās vērtības ir neapstrādātas, tas ir, tām netiek piemērota pēcapstrāde.

Note:
Kopējais parādīto vērtību skaits ir noteikts ar parametra *Limit for search and filter results* vērtību, kas iestatīta sadaļā *Administration* > *General* > *GUI*.

Kartes

Pārskats

Sadaļā *Monitoring* > *Maps* varat konfigurēt, pārvaldīt un skatīt tīkla kartes.

Atverot šo sadaļu, jūs redzēsiet vai nu pēdējo karti, ko skatījāt, vai visu karšu sarakstu, kurām jums ir piekļuve.

Visas kartes var būt vai nu publiskas, vai privātas. Publiskās kartes ir pieejamas visiem lietotājiem, savukārt privātās kartes ir pieejamas tikai to īpašniekam un lietotājiem, ar kuriem karte ir kopīgota.

Kartes saraksts



Attēlotie dati:

Column	Description
Name	Kartes nosaukums. Noklikšķiniet uz nosaukuma, lai skatītu karti.
Width	Tiek parādīts kartes platums.
Height	Tiek parādīts kartes augstums.
Actions	Pieejamas divas darbības: **Properties** - iestatīt kartes vispārīgās īpašības **Edit** - piekļūt režģim, lai pievienotu kartes elementus

Lai konfigurētu jaunu karti, noklikšķiniet uz pogas *Create map* augšējā labajā stūrī. Lai importētu karti no YAML, XML vai JSON faila, noklikšķiniet uz pogas *Import* augšējā labajā stūrī. Lietotājs, kurš importē karti, tiks iestatīts kā tās īpašnieks.

Divas pogas zem saraksta piedāvā vairākas masveida rediģēšanas iespējas:

- *Export* - eksportēt kartes uz YAML, XML vai JSON failu
- *Delete* - dzēst kartes

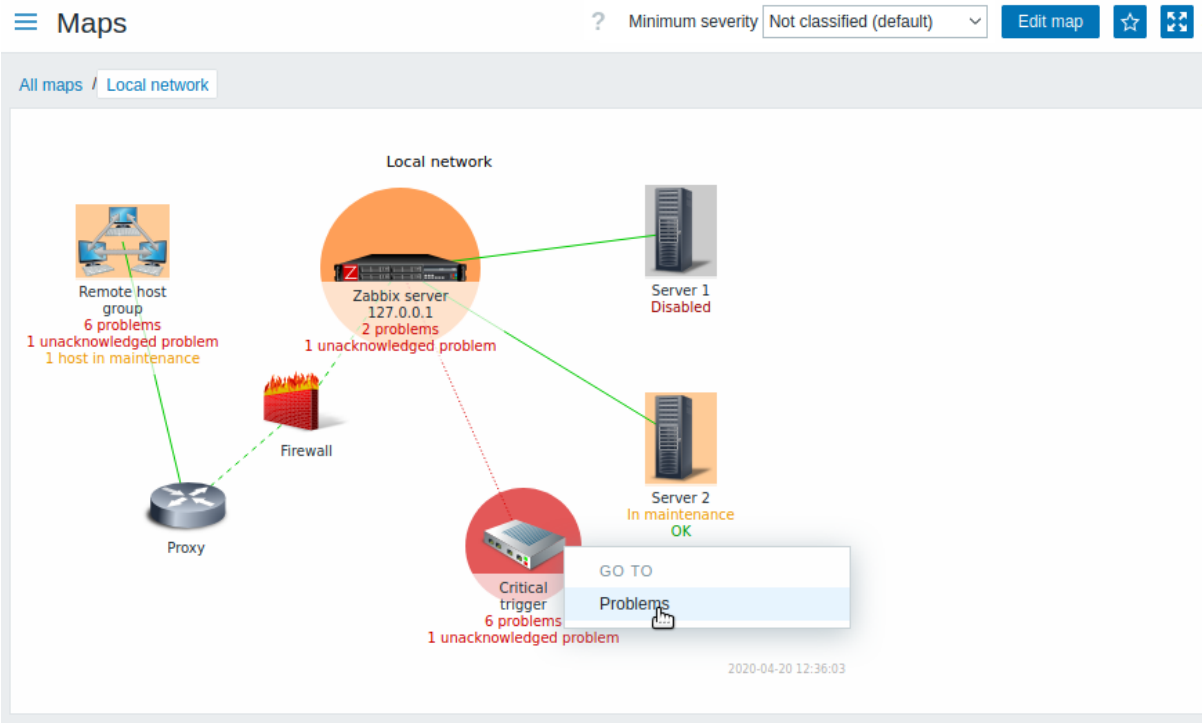
Lai izmantotu šīs opcijas, atzīmējiet izvēles rūtiņas attiecīgo karšu priekšā un pēc tam noklikšķiniet uz vajadzīgās pogas.

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 problēmu trigeru nopietnības līmeni, ko rādīt. Nopietnība, kas atzīmēta kā *default*, ir līmenis, kas iestatīts kartes konfigurācijā. Ja karte satur apakškarti, pāreja uz apakškarti saglabās augstāka līmeņa kartes nopietnību (izņemot gadījumu, ja tā ir *Not classified*; šajā gadījumā tā netiks nodota apakškartē).

Ikonu izcelšana

Ja kartes elements ir problēmas stāvoklī, tas tiek izcelts ar apaļu apli. Apļa aizpildījuma krāsa atbilst problēmas nopietnības krāsai. Kopā ar elementu tiks parādītas tikai tās problēmas, kuru nopietnības līmenis ir vienāds ar atlasīto līmeni vai augstāks. Ja visas problēmas ir apstiprinātas, ap apli tiek parādīta bieza zaļa apmale.

Papildus:

- uzturēšanā esošs hosts tiek izcelts ar oranžu, aizpildītu kvadrātu. Ņemiet vērā, ka uzturēšanas izcelšanai ir prioritāte pār problēmas nopietnības izcelšanu, ja kartes elements ir hosts.
- Atspējots (neuzraudzīts) hosts tiek izcelts ar pelēku, aizpildītu kvadrātu.

Izcelšana tiek parādīta, ja kartes konfigurācijā ir atzīmēta izvēles rūtiņa *Ikonu izcelšana*.

Nesenās izmaiņas marķieri

Uz iekšu vērsti sarkani trīsstūri ap elementu norāda uz nesenām trigerisa stāvokļa izmaiņām — tādām, kas notikušas pēdējo 30 minūšu laikā. Šie trīsstūri 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 pieejamajām 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 Uzraudzība.

Kartēs pieejams kopsavilkums

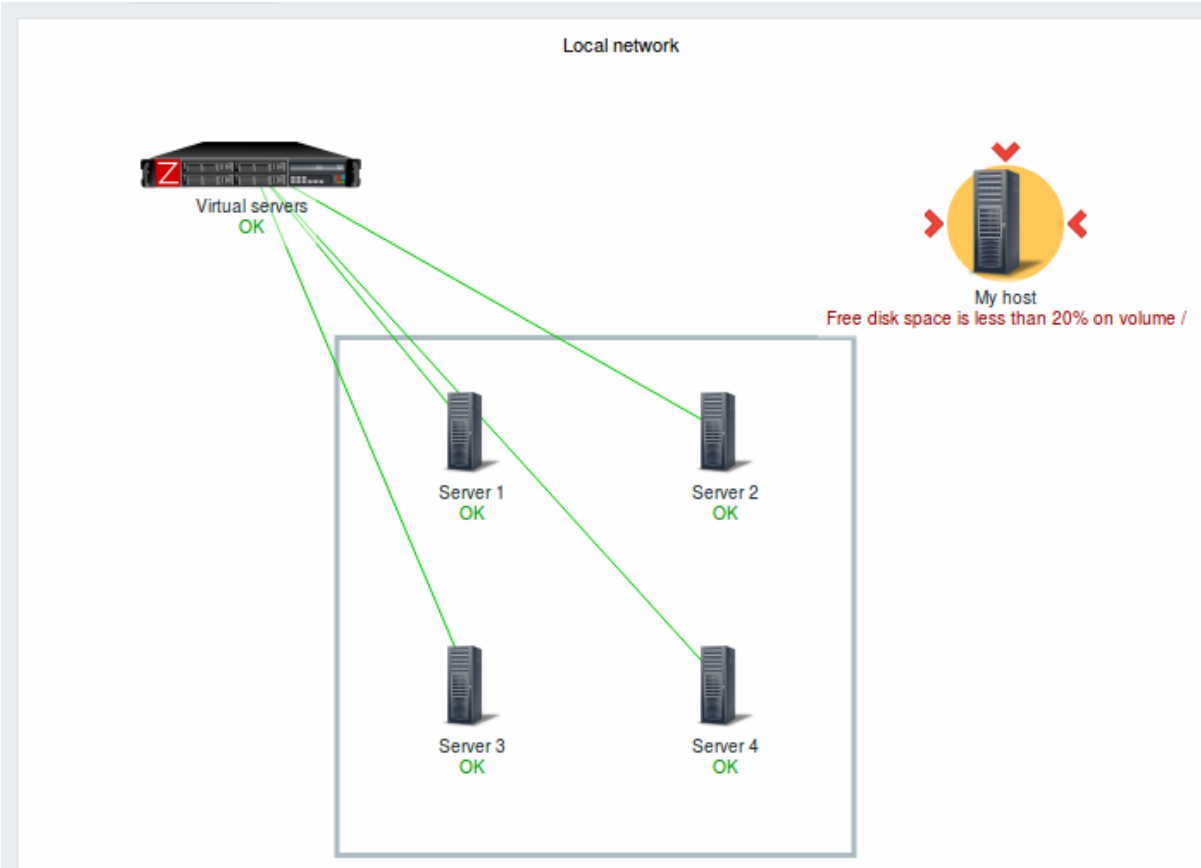
Ir pieejams slēpts rekvizīts "aria-label", kas ļauj kartes informāciju nolasīt ar ekrāna lasītāju.
Ir pieejams gan vispārīgs kartes apraksts, gan atsevišķu elementu apraksts šādā formātā:

- 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, kopā 1 problēma. 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`

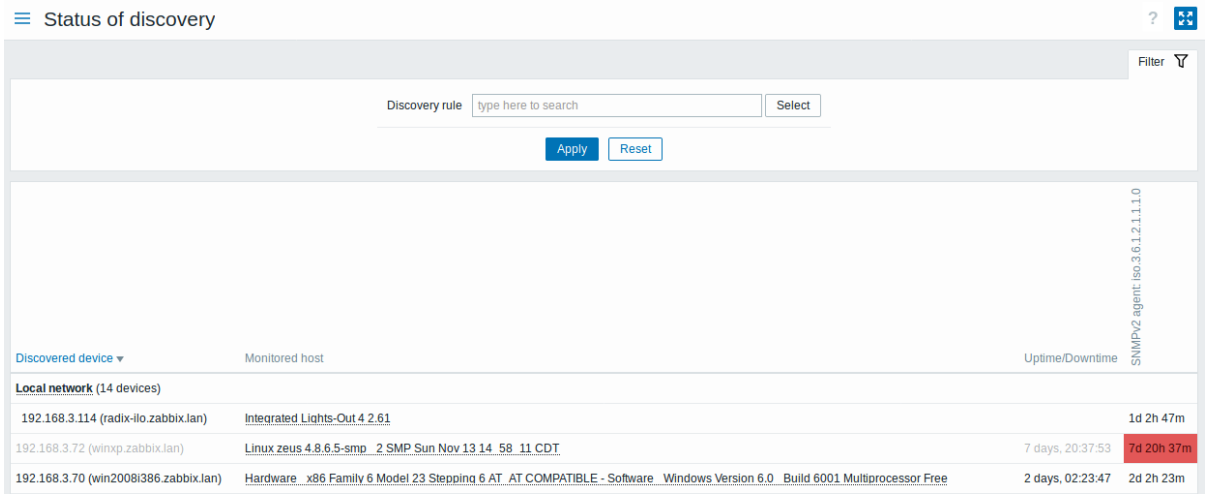
atvērs karti ar šādu nosaukumu (Local network).

Ja ir norādīti gan `sysmapid` (kartes ID), gan `mapname` (kartes nosaukums), parametram `mapname` ir augstāka prioritāte.

Atklāšana

Pārskats

Sadaļā *Monitoring* > *Discovery* tiek parādīti tīkla atklāšanas rezultāti. Atklātās ierīces tiek kārtotas pēc atklāšanas noteikuma.



Parādītie dati:

Kolonna	Apraksts
Atklātā ierīce	Atklātās ierīces tiek uzskaitītas un grupētas pēc atklāšanas noteikuma. Noklikšķinot uz atklāšanas noteikuma, tiek atvērta noteikuma izvēlne ar saiti uz atklāšanas noteikuma konfigurācijas veidlapu. Ja ierīcei ir vairākas IP adreses (kā noteikts atklāšanas noteikuma *Ierīces unikālo identifikācijas kritēriju* iestatījumā), tās tiek uzskaitītas kopā: vispirms ierīces primārā IP adrese, pēc tam pārējās IP adreses (ar atkāpi). Primārā IP adrese ir zemākā IP adrese, kurai kolonnā *Uzraudzītais hosts* ir norādīts hosts, vai, ja šādas adreses nav, ierīces zemākā IP adrese. Ierīces var kārtot pēc to primārās IP adreses augošā vai dilstošā secībā; pārējās ierīces IP adreses vienmēr tiek uzskaitītas zem tās primārās IP adreses augošā secībā.
Uzraudzītais hosts	Ja ierīce jau tiek uzraudzīta, šajā kolonnā tiek norādīts hosts nosaukums. Noklikšķinot uz hosts nosaukuma, tiek atvērta hosts izvēlne.
Darbspējas laiks/ dīkstāves laiks	Šajā kolonnā tiek parādīts ierīces atklāšanas vai tās zaudēšanas ilgums pēc iepriekšējās atklāšanas. Ierīcei ar vairākām IP adresēm ilgums tiek parādīts tikai tās primārajai IP adresei.
Atklāšanas pārbaude	Tiek parādīts katra atklātā ierīcē atrastā atsevišķā pakalpojuma (atklāšanas pārbaudes) stāvoklis. Sarkana šūna norāda, ka pakalpojums nedarbojas. Šūnā tiek iekļauts pakalpojuma darbspējas vai dīkstāves laiks. Šī kolonna tiek parādīta tikai tad, ja pakalpojums ir atrasts vismaz vienā atklātajā ierīcē.

Note:
Šajā sadaļā parādīto atklāšanas noteikumu skaitu un kopējo atklāto ierīču skaitu ierobežo parametrs *Meklēšanas un filtra rezultātu ierobežojums* sadaļā *Administration > General > GUI*. Noteikumi vai ierīces, kas pārsniedz šo ierobežojumu, netiek parādīti, un par to netiek rādīts nekāds paziņojums.

Pogas

Skata režīma pogas, kas ir kopīgas visām sadaļām, ir aprakstītas lapā Monitoring.

Filtra izmantošana

Varat izmantot filtru, lai parādītu tikai tās atklāšanas noteikumus, kas jūs interesē. Lai uzlabotu meklēšanas veiktspēju, dati tiek meklēti ar neatrisinātiem makro.

Ja filtrā nekas nav atlasīts, tiek parādīti visi iespējotie atklāšanas noteikumi. Lai parādītu konkrētu atklāšanas noteikumu, sāciet tā nosaukuma ievadi filtrā. Tiks uzskaitīti visi atbilstošie iespējotie atklāšanas noteikumi, kurus varēs atlasīt. Var atlasīt vairāk nekā vienu atklāšanas noteikumu.

APM

Pārskats

Izvēlne *APM* ir paredzēta Zabbix lietojumprogrammu veiktspējas uzraudzības (APM) funkcijām. Tajā tiek rādīti telemetrijas dati, ko jūsu lietojumprogrammas un pakalpojumi nosūta, izmantojot OpenTelemetry, atvērtu standartu telemetrijas datu apkopošanai. Šie

dati palīdz noskaidrot, kā pieprasījumi plūst caur jūsu lietojumprogrammām un pakalpojumiem, novērtēt to veiktspēju un atrast lēnu vai neveiksmīgu darbību cēloni.

Izvēlnē ir šādas sadaļas, kas atbilst OpenTelemetry signālu tipiem:

- Traces
- Metrics
- Logs

Atšķirībā no *Monitoring* > *Latest data*, kur tiek rādītas Zabbix vienumu apkopotās vērtības, *APM* sadaļās telemetrijas dati tiek rādīti tieši no datu avota, tāpēc nav jāizveido neviens vienums. Lai rādītu datus *APM* sadaļās:

1. Norādiet telemetrijas datu avotu sadaļā *Administration* > *Data source*.
2. Pārliecinieties, ka telemetrijas datus datu avotā ieraksta Zabbix starpniekserveris ar iespējotu APM vai OpenTelemetry kolektors. Ņemiet vērā, ka Zabbix serveris nevar saņemt telemetrijas datus; tos var saņemt tikai Zabbix starpniekserveris.

Lai izmantotu telemetrijas datus trigeros vai grafikos, izmantojiet telemetrijas vaicājuma vienumu.

Traces

Pārskats

Sadaļā *APM* > *Traces* tiek parādīti no jūsu lietojumprogrammām un pakalpojumiem apkopotie izsekojumi.

Izsekojums attēlo ceļu, ko pieprasījums veic jūsu lietojumprogrammās un pakalpojumos. Piemēram, kad lietotājs tiešsaistes veikalā noklikšķina uz pogas *Place order*, pieprasījums var pāriet no lietotāja saskarnes pakalpojuma caur pasūtījuma noformēšanas un maksājumu pakalpojumiem uz datubāzi.

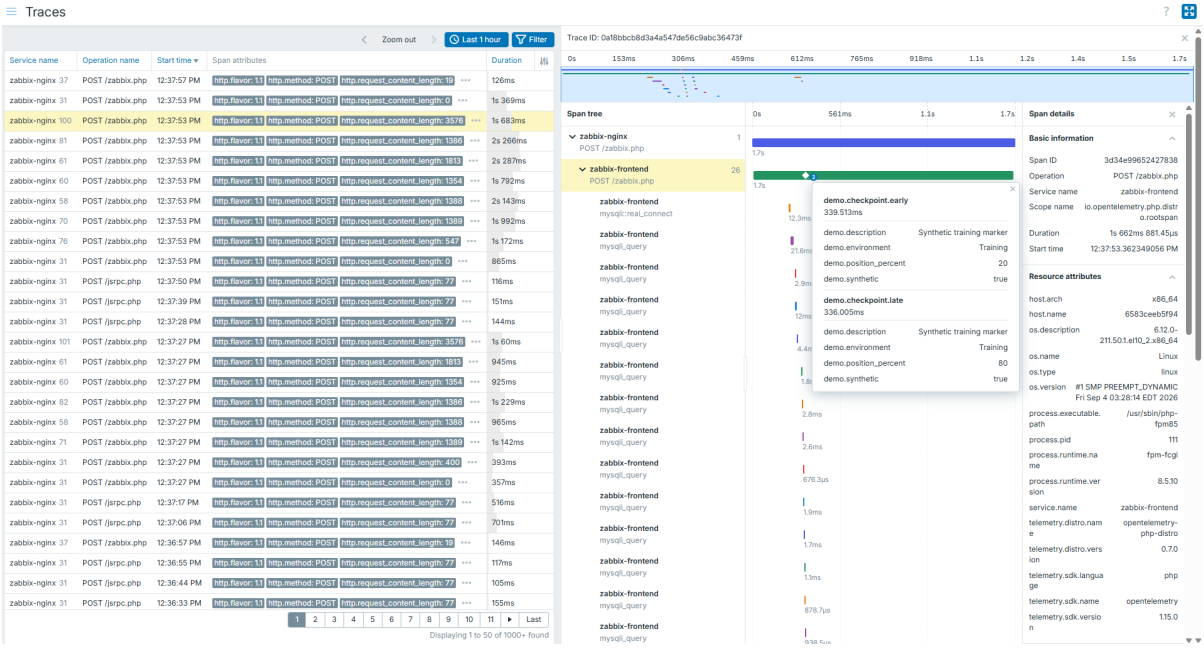
Izsekojums sastāv no **diapazoniem**. Katrs diapazons attēlo vienu darbību, piemēram, HTTP pieprasījumu, ko lietotāja saskarnes pakalpojums nosūta pasūtījuma noformēšanas pakalpojumam, datubāzes vaicājumu, kas saglabā pasūtījumu, vai funkcijas izsaukumu, kas aprēķina piegādes izmaksas.

Diapazoni veido hierarhiju. **Saknes diapazons** attēlo pieprasījuma pirmo darbību. Katrs pārējais diapazons ir tā diapazona **bērnelements**, kura darbības laikā tas tika sākts. Piemēram, lietotāja saskarnes pakalpojums, kas saņem *Place order* pieprasījumu, ir saknes diapazons, bet pasūtījumu apstrādājošais pasūtījuma noformēšanas pakalpojums ir tā bērnelements.

Katru diapazonu izveido **pakalpojums** (piemēram, `zabbix-frontend`; to nosaka OpenTelemetry resursa atribūts `service.name`) un instrumentācijas tvērums, proti, koda daļa, kas to izveidoja (piemēram, bibliotēka, kas izseko HTTP pieprasījumus).

Lai nodrošinātu kontekstu par katru darbību, diapazoniem ir arī atslēgas un vērtības **atribūti**: diapazona atribūti apraksta pašu darbību (piemēram, `http.method: POST`), bet resursa atribūti apraksta pakalpojumu un vidi, kurā tas darbojas (piemēram, `host.name: eabf8c2a5ddc`).

Plašāku informāciju par izsekojumiem skatiet OpenTelemetry dokumentācijā.



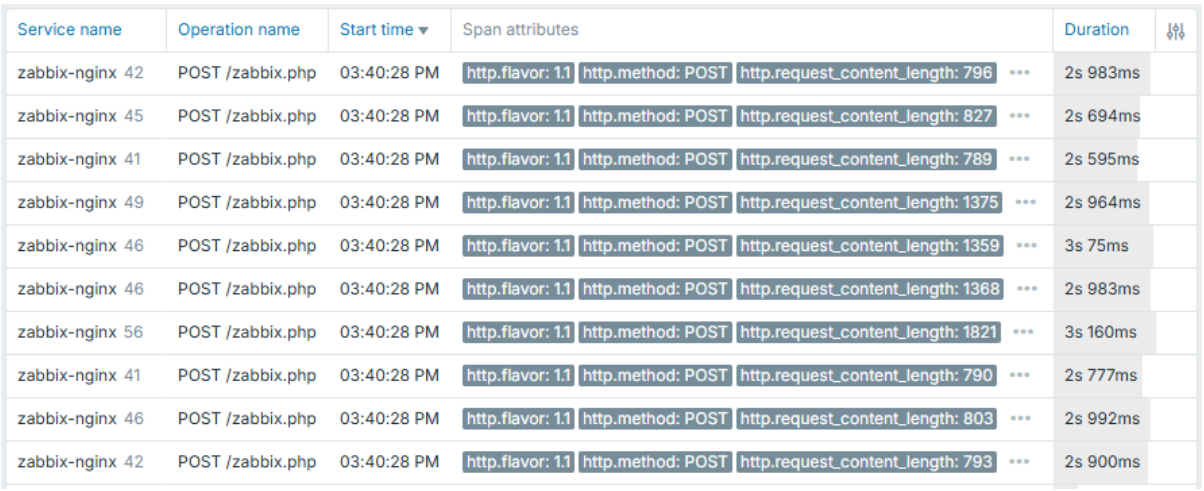
Sadaļā *APM* > *Traces* ir pieejami šādi elementi:

- Poga Kioska režīms - parāda tikai lapas saturu bez izvēlnes un galvenes.
- Laika perioda atlasītājs - atlasa laika periodu, par kuru rādīt izsekojumus.

- Filtrs - parāda tikai jūs interesējošos izsekojumus.
- Izsekojumu saraksts - ļauj skatīt izsekojumus, kas aprakstīti, izmantojot to saknes diapazonus.
- Izsekojuma panelis - laika skalā ļauj skatīt visus atlasītā izsekojuma diapazonus.

Traces saraksts

Katra rinda traces sarakstā attēlo vienu trace; kolonnās tiek parādīti tās saknes span dati.



Service name	Operation name	Start time ▾	Span attributes	Duration
zabbix-nginx 42	POST /zabbix.php	03:40:28 PM	http.flavor: 1.1 http.method: POST http.request_content_length: 796 •••	2s 983ms
zabbix-nginx 45	POST /zabbix.php	03:40:28 PM	http.flavor: 1.1 http.method: POST http.request_content_length: 827 •••	2s 694ms
zabbix-nginx 41	POST /zabbix.php	03:40:28 PM	http.flavor: 1.1 http.method: POST http.request_content_length: 789 •••	2s 595ms
zabbix-nginx 49	POST /zabbix.php	03:40:28 PM	http.flavor: 1.1 http.method: POST http.request_content_length: 1375 •••	2s 964ms
zabbix-nginx 46	POST /zabbix.php	03:40:28 PM	http.flavor: 1.1 http.method: POST http.request_content_length: 1359 •••	3s 75ms
zabbix-nginx 46	POST /zabbix.php	03:40:28 PM	http.flavor: 1.1 http.method: POST http.request_content_length: 1368 •••	2s 983ms
zabbix-nginx 56	POST /zabbix.php	03:40:28 PM	http.flavor: 1.1 http.method: POST http.request_content_length: 1821 •••	3s 160ms
zabbix-nginx 41	POST /zabbix.php	03:40:28 PM	http.flavor: 1.1 http.method: POST http.request_content_length: 790 •••	2s 777ms
zabbix-nginx 46	POST /zabbix.php	03:40:28 PM	http.flavor: 1.1 http.method: POST http.request_content_length: 803 •••	2s 992ms
zabbix-nginx 42	POST /zabbix.php	03:40:28 PM	http.flavor: 1.1 http.method: POST http.request_content_length: 793 •••	2s 900ms

Traces tiek attēlotas pielāgojamā tabulā. Tabulas galvenē esošā poga *Pielāgot tabulu* ļauj parādīt, paslēpt vai pārkārtot kolonnas, kā arī atiestatīt izkārtojumu uz noklusējuma iestatījumiem. Katras kolonnas izmēru var mainīt arī, velkot tās labo malu; veiciet dubultklikšķi uz malas, lai atiestatītu izmēru. Visi pielāgojumi tiek lietoti nekavējoties un saglabāti pašreizējam lietotājam.

Traces sarakstā var būt šādas kolonnas:

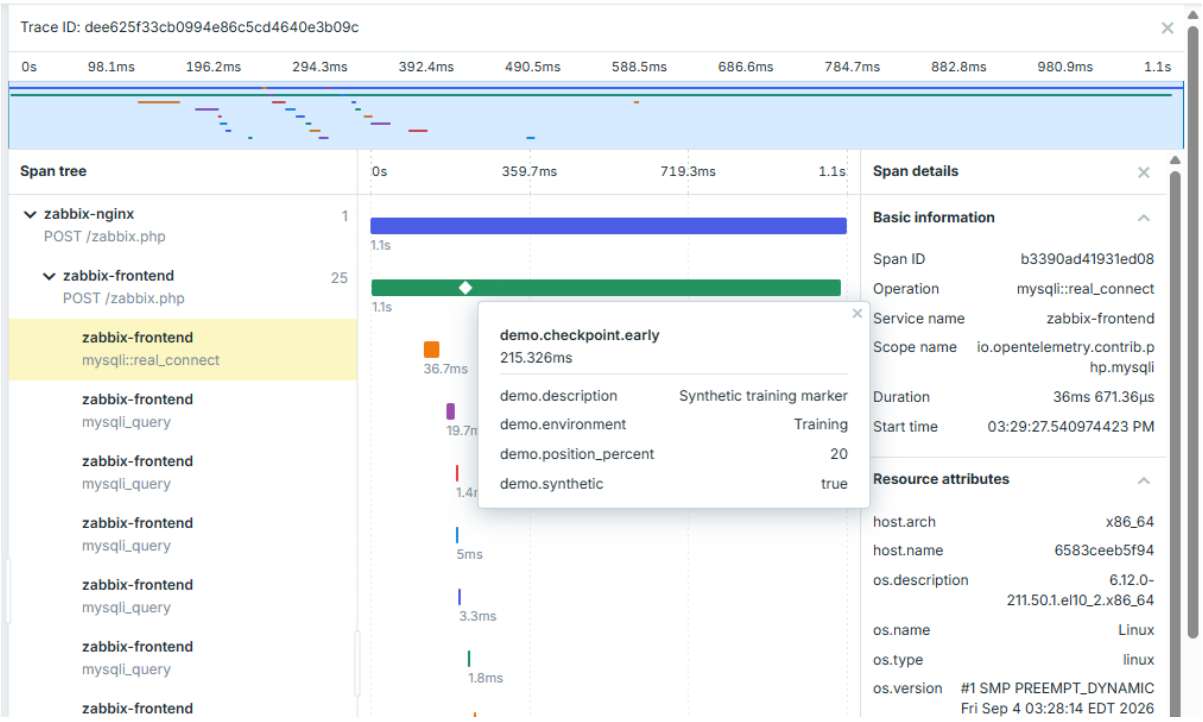
Note:
Katra kolonna atbilst kolonnai `otel_traces` tabulā telemetrijas datu avotā; kolonnu sarakstu skatiet sadaļā Atbalstītās telemetrijas datu kolonnas. Lielākajai daļai kolonnu ir tāds pats nosaukums kā datu avotā (piemēram, *Trace ID* atbilst kolonnai `TraceId`). Ja nosaukumi atšķiras, datu avota kolonna ir norādīta tālāk esošajā tabulā.

Column	Description
Trace ID	Trace ID, kas ir kopīgs visiem trace piederošajiem span.
Span ID	Saknes span ID.
Trace state	Papildu dati, ja tādi ir, ko izsekošanas rīki pievieno trace (skatiet W3C Trace Context).
Service name	Serviss, kas izveidoja saknes span, kam seko span skaits trace. Ja kādam span ir statuss *Error* (skatiet tālāk), sarkana ikona parāda to skaitu.
Operation name	Operācija, ko attēlo saknes span (piemēram, `POST /zabbix.php`). Datu avota kolonna: `SpanName`.
Span kind	Saknes span loma: **Unspecified** - nav norādīta. **Internal** - operācija, kas veikta lietotnē. **Server** - ienākoša pieprasījuma apstrāde. **Client** - izejošs pieprasījums. **Producer** - asinhronas operācijas izveide (piemēram, ziņojuma nosūtīšana). **Consumer** - asinhronas operācijas apstrāde.
Scope name	Instrumentācijas tvērums, kas izveidoja saknes span.
Scope version	Instrumentācijas tvēruma versija.
Status code	Operācijas statuss, ko attēlo saknes span: **Unset** - saknes span nav piešķirts skaidrs statuss. **Ok** - operācija tika veiksmīgi pabeigta. **Error** - operācija tika pabeigta ar kļūdu.
Status message	Papildu informācija, ja tāda ir, par statusu.
Start time	Datums un laiks, kad sākās saknes span. Datu avota kolonna: `Timestamp`.
Span attributes	Saknes span atribūti. Noklikšķiniet uz ••• trīs punktu pogas, lai parādītu visus atribūtus.
Resource attributes	Saknes span resursa atribūti. Noklikšķiniet uz ••• trīs punktu pogas, lai parādītu visus atribūtus.

Column	Description
Duration	Trace ilgums; pelēkā josla parāda tā ilgumu attiecībā pret citiem trace.

Traces panelis

Noklikšķiniet uz trases trašu sarakstā, lai atvērtu trases paneli, kurā laika skalā tiek parādīti visi trases span elementi: kādas darbības tika veiktas, kādā secībā un cik ilgs laiks bija nepieciešams katrai no tām.

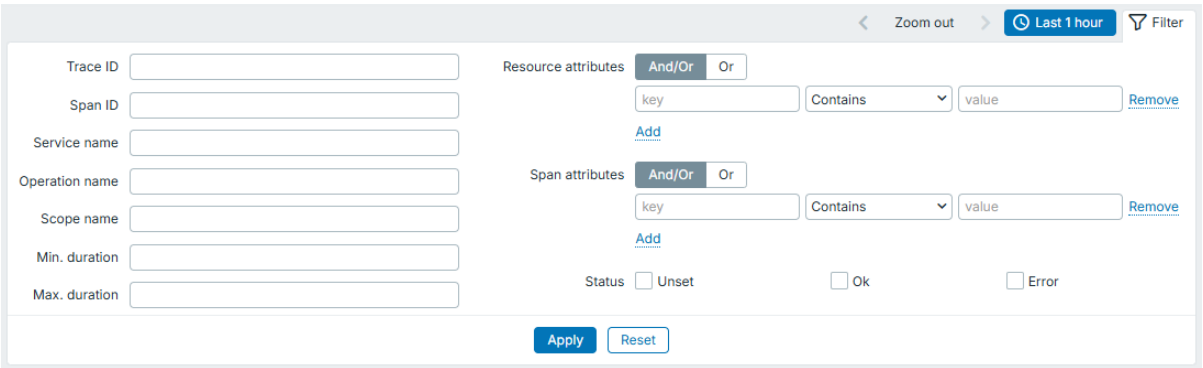


Trases panelī var būt šādi elementi:

Elements	Apraksts
Trases ID	Atlasītās trases ID.
Trases laika skalas pārskats	Visa trase ar span elementiem līdz 8 rindās (vairāki span elementi var būt vienā rindā). Velciet atlasītā diapazona robežas vai pašu diapazonu, lai tuvinātu span laika skalu šai trases daļai.
Span koks	Span elementi to hierarhijā; katram tiek parādīts tā serviss (treknrakstā), darbības nosaukums un tiešo pakārtoto span elementu skaits.
Span laika skala	Span elementi joslu veidā: novietojums un garums parāda, kad katrs span elements sākās un cik ilgs laiks bija nepieciešams tā izpildei. Marķieri parāda notikumus (piemēram, izņēmumus); novietojiet kursoru virs marķiera vai noklikšķiniet uz tā, lai skatītu detalizētu informāciju.
Span informācija	Noklikšķiniet uz span elementa span kokā, lai skatītu tā detalizēto informāciju un atribūtus.

Filtra izmantošana

Varat izmantot filtru, lai parādītu tikai tos izsekošanas ierakstus, kas jūs interesē.



Lai filtrētu pēc atribūtiem, ievadiet atribūta atslēgu, atlasiet operatoru un ievadiet atribūta vērtību.

Tiek atbalstīti šādi operatori:

- **Satur** - *(noklusējums)* atribūta vērtība satur ievadīto virkni.
- **Ir vienāds ar** - atribūta vērtība ir vienāda ar ievadīto virkni.
- **Nesatur** - atribūta vērtība nesatur ievadīto virkni.
- **Nav vienāds ar** - atribūta vērtība nav vienāda ar ievadīto virkni.
- **Pastāv** - atribūts ar ievadīto atslēgu pastāv.
- **Nepastāv** - atribūts ar ievadīto atslēgu nepastāv.

Ja pievienojat vairākus nosacījumus, atlasiet, kā tos novērtēt:

- **And/Or** - *(noklusējums)* jāizpilda visi nosacījumi; nosacījumi ar vienādu atribūta atslēgu tiek apvienoti ar *Or*.
- **Or** - jāizpilda vismaz viens nosacījums.

Metrikas

Pārskats

Sadaļā *APM* > *Metrikas* tiek rādīti no jūsu lietojumprogrammām un pakalpojumiem apkopotie metriku datu punkti.

Metrika ir mērījums (piemēram, pieprasījuma ilgums, apstrādāto pasūtījumu skaits vai atmiņas lietojums). Lietojumprogrammas nosūta savas vērtības kā **datu punktus**, kas satur noteiktā laikā izmērīto vērtību (vai apkopotās vērtības). Metrika ir līdzīga Zabbix vienumam, bet datu punkts ir līdzīgs šī vienuma vērtībai.

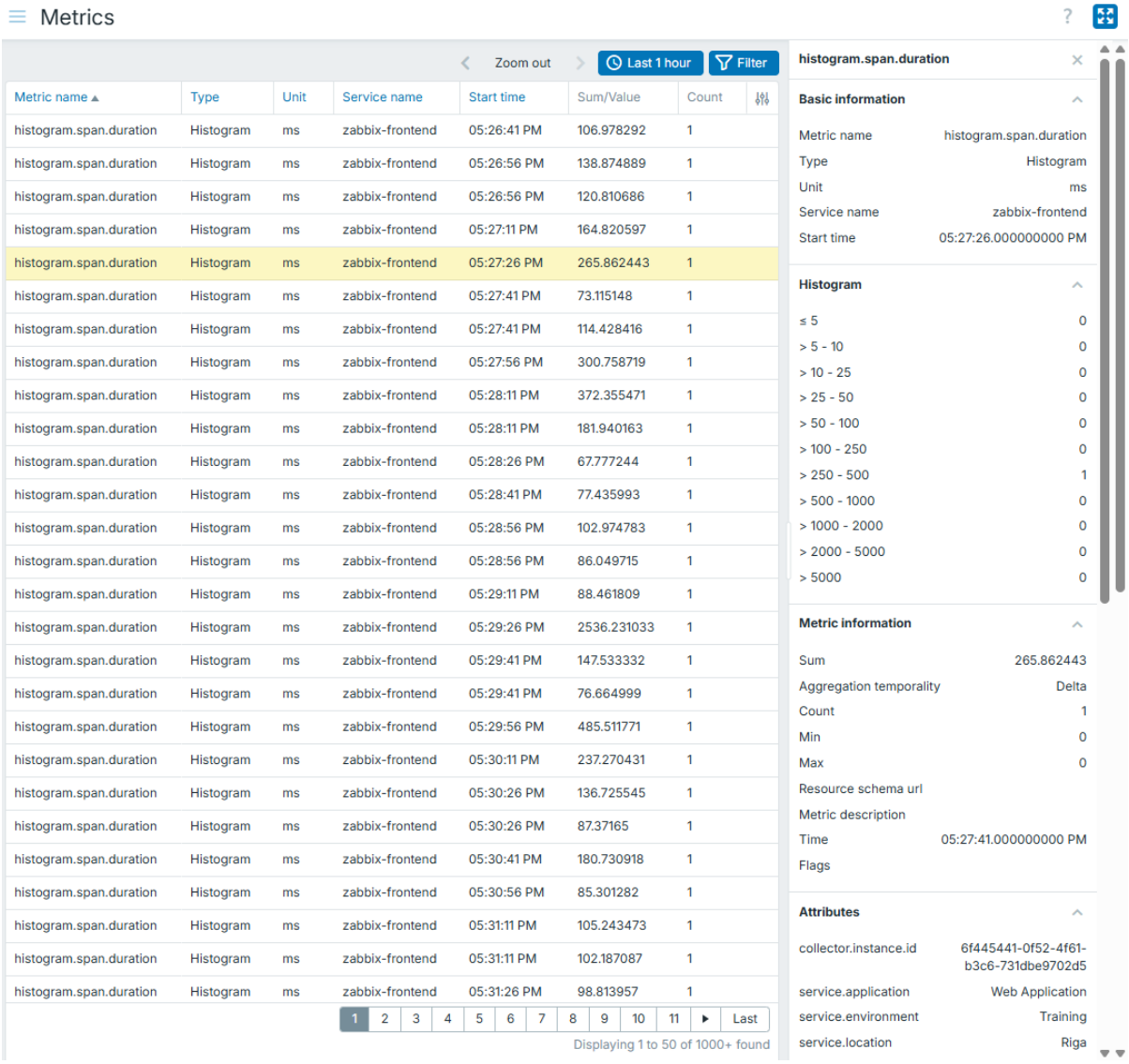
Datu punkts var būt kāds no šiem tipiem:

- **Gauge** - noteiktā laikā izmērīta vērtība (piemēram, pašreizējais atmiņas lietojums).
- **Sum** - mērījumu summa noteiktā laika periodā (piemēram, kopējais pieprasījumu skaits).
- **Histogram** - noteiktā laika periodā veikto mērījumu skaits, kas ietilpst katrā no vairākām fiksētām vērtību diapazonu grupām, ko sauc par groziem (piemēram, cik pieprasījumu izpilde ilga 0-100 ms, 100-250 ms vai 250-500 ms).
- **Exponential histogram** - līdzīgi histogrammai, taču katrs grozs ir par vienādu koeficientu platāks nekā iepriekšējais (piemēram, 1-2 ms, 2-4 ms, 4-8 ms).

Katru datu punktu izveido **pakalpojums** (piemēram, `zabbix-frontend`; to nosaka OpenTelemetry resursa atribūts `service.name`) un instrumentācijas tvērums, proti, koda daļa, kas to izveidoja (piemēram, bibliotēka, kas mēra HTTP pieprasījumu ilgumu).

Lai sniegtu kontekstu par katru mērījumu, datu punktiem ir arī atslēgas un vērtības **atribūti**: datu punkta atribūti apraksta, kas tika izmērīts, un atšķir vienas metrikas datu punktus (piemēram, `http.request.method: POST`), savukārt resursa atribūti apraksta pakalpojumu un vidi, kurā tas darbojas (piemēram, `host.name: eabf8c2a5ddc`).

Plašāku informāciju par metrikām skatiet OpenTelemetry dokumentācijā.

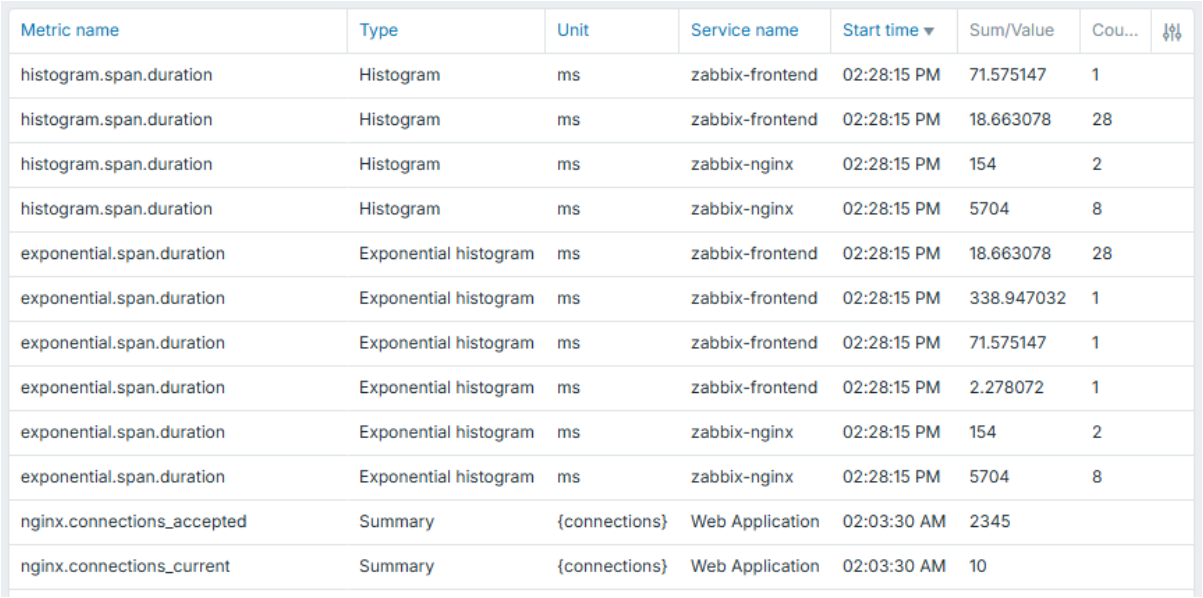


Sadaļā *APM* > *Metrikas* ir pieejami šādi elementi:

- Poga Kioska režīms - rādīt tikai lapas saturu bez izvēlnes un galvenes.
- Laika perioda atlasītājs - atlasīt laika periodu, par kuru rādīt datu punktus.
- Filtrs - rādīt tikai jūs interesējošos datu punktus.
- Metriku saraksts - skatīt datu punktus.
- Metrikas informācija - skatīt atlasītā datu punkta detalizētu informāciju.

Metriku saraksts

Katra rinda metriku sarakstā attēlo datu punktu.



Metric name	Type	Unit	Service name	Start time ▾	Sum/Value	Cou...
histogram.span.duration	Histogram	ms	zabbix-frontend	02:28:15 PM	71.575147	1
histogram.span.duration	Histogram	ms	zabbix-frontend	02:28:15 PM	18.663078	28
histogram.span.duration	Histogram	ms	zabbix-nginx	02:28:15 PM	154	2
histogram.span.duration	Histogram	ms	zabbix-nginx	02:28:15 PM	5704	8
exponential.span.duration	Exponential histogram	ms	zabbix-frontend	02:28:15 PM	18.663078	28
exponential.span.duration	Exponential histogram	ms	zabbix-frontend	02:28:15 PM	338.947032	1
exponential.span.duration	Exponential histogram	ms	zabbix-frontend	02:28:15 PM	71.575147	1
exponential.span.duration	Exponential histogram	ms	zabbix-frontend	02:28:15 PM	2.278072	1
exponential.span.duration	Exponential histogram	ms	zabbix-nginx	02:28:15 PM	154	2
exponential.span.duration	Exponential histogram	ms	zabbix-nginx	02:28:15 PM	5704	8
nginx.connections_accepted	Summary	{connections}	Web Application	02:03:30 AM	2345	
nginx.connections_current	Summary	{connections}	Web Application	02:03:30 AM	10	

Datu punkti tiek attēloti pielāgojamā tabulā. Poga *Pielāgot tabulu* tabulas galvenē ļauj rādīt, paslēpt vai pārkārtot kolonnas, kā arī atiestatīt izkārtojumu uz noklusējuma iestatījumiem. Varat arī mainīt katras kolonnas platumu, velkot tās labo malu; veiciet dubultklikšķi uz malas, lai atiestatītu platumu. Visi pielāgojumi tiek lietoti nekavējoties un saglabāti pašreizējam lietotājam.

Metriku sarakstā var būt šādas kolonnas:

Note:
Katra kolonna, izņemot *Type*, atbilst kolonnai telemetrijas datu avota telemetrijas datu avots tabulās (`otel_metrics_gauge`, `otel_metrics_sum`, `otel_metrics_histogram` un `otel_metrics_exponential_histogram`); kolonnu sarakstu skatiet sadaļā Atbalstītās telemetrijas datu kolonnas. Lielākajai daļai kolonnu ir tāds pats nosaukums kā datu avotā (piemēram, *Metric name* atbilst kolonnai `MetricName`). Ja nosaukumi atšķiras, datu avota kolonna ir norādīta tālāk esošajā tabulā.

Column	Description
Metric name	Metrika, kurai pieder datu punkts (piemēram, `http.server.duration`).
Type	Datu punkta tips: *Gauge*, *Sum*, *Histogram* vai *Exponential histogram*. To nosaka metrikas tabula, kurā atrodas datu punkts.
Unit	Metrikas vērtību mērvienība, ja tāda ir (piemēram, `s` vai `By`; skatiet UCUM). Datu avota kolonna: `MetricUnit`.
Service name	Serviss, kas izveidoja datu punktu.
Scope name	Instrumentācijas tvērums, kas izveidoja datu punktu.
Start time	Datums un laiks, kad sākās datu punkta mērījumu periods. Datu avota kolonna: `StartTimeUnix`.
Sum/Value	Visu mērījumu summa (*Histogram*, *Exponential histogram*) vai datu punkta vērtība (*Gauge*, *Sum*).
Count	Mērījumu skaits *Histogram* vai *Exponential histogram* datu punktā.
Time	Datu punkta datums un laiks; *Sum*, *Histogram* un *Exponential histogram* datu punktiem — mērījumu perioda beigas. Datu avota kolonna: `TimeUnix`.
Flags	Datu punkta karodziņi: **0** - nav iestatīts neviens karodziņš. **1** - datu punktam nav reģistrēta vērtība.
Aggregation temporality	Kā *Sum*, *Histogram* vai *Exponential histogram* datu punkta vērtības tiek apkopotas laika gaitā: **Unspecified** - nav norādīts. **Delta** - apkopots par periodu kopš iepriekšējā datu punkta. **Cumulative** - apkopots par periodu kopš sākuma laika.
Metric description	Metrikas apraksts, ja tāds ir.
Attributes	Datu punkta atribūti. Noklikšķiniet uz ⋯ trīs punktu pogas, lai parādītu visus atribūtus.
Resource attributes	Datu punkta resursa atribūti. Noklikšķiniet uz ⋯ trīs punktu pogas, lai parādītu visus atribūtus.
Scope attributes	Instrumentācijas tvēruma atribūti. Noklikšķiniet uz ⋯ trīs punktu pogas, lai parādītu visus atribūtus.

Metrikas informācija

Noklikšķiniet uz datu punkta metrikas sarakstā, lai atvērtu metrikas informācijas paneli, kurā tiek parādīta visa informācija par datu punktu.

exponential.span.calls

Basic information

Metric name	exponential.span.calls
Type	Summary
Unit	
Service name	zabbix-frontend
Start time	02:22:15.000000000 PM

Metric information

Value	1
Aggregation temporality	Delta
Is monotonic	true
Resource schema url	
Metric description	
Time	02:22:30.000000000 PM
Flags	

Attributes

Resource attributes

Scope attributes

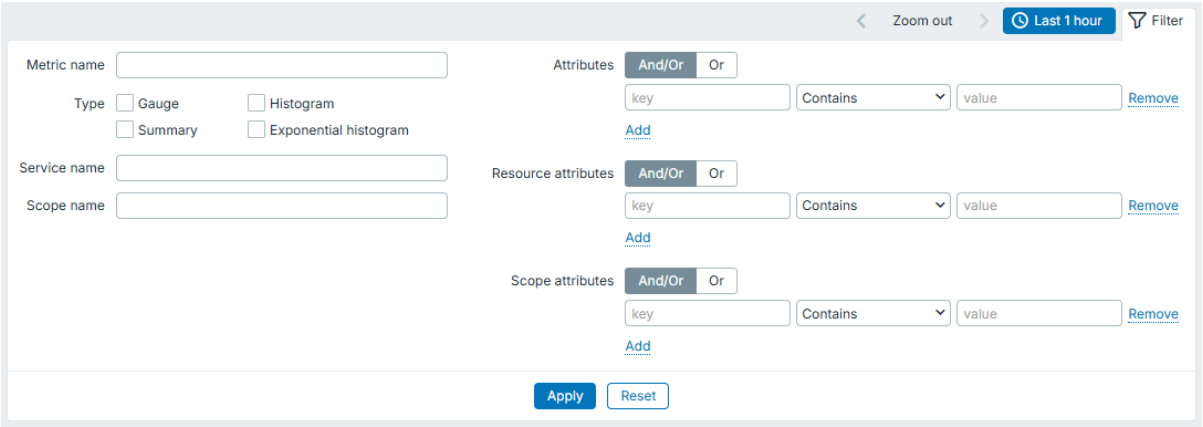
Scope

Metrikas informācijas panelī var būt šādas sadaļas:

Sadaļa	Apraksts
Pamatinformācija	Metrikas nosaukums, tips, mērvienība, pakalpojuma nosaukums un sākuma laiks.
Histogramma	Intervāli (piemēram, `> 5 – 10`), katram norādot tajā ietverto mērījumu skaitu. Tiek parādīts tikai *Histogrammas* datu punktiem.
Eksponenciālā histogramma	Intervāli (piemēram, `>= 249.7885 – 249.7887`), katram norādot tajā ietverto mērījumu skaitu. Tiek parādīts tikai *Eksponenciālās histogrammas* datu punktiem.
Metrikas informācija	Datu punkta vērtības, kas ir atkarīgas no tā tipa, kā arī metrikas apraksts, laiks, karodziņi un resursa shēmas URL.
Atribūti	Datu punkta atribūti.
Resursa atribūti	Resursa atribūti.
Tvēruma atribūti	Instrumentācijas tvēruma atribūti, ja tādi ir.
Tvērums	Instrumentācijas tvēruma nosaukums, versija un shēmas URL.

Filtra izmantošana

Varat izmantot filtru, lai parādītu tikai tos datu punktus, kas jūs interesē.



Lai filtrētu pēc atribūtiem, ievadiet atribūta atslēgu, atlasiet operatoru un ievadiet atribūta vērtību.

Tiek atbalstīti šādi operatori:

- **Satur** - *(noklusējums)* atribūta vērtība satur ievadīto virkni.
- **Vienāds ar** - atribūta vērtība ir vienāda ar ievadīto virkni.
- **Nesatur** - atribūta vērtība nesatur ievadīto virkni.
- **Nav vienāds ar** - atribūta vērtība nav vienāda ar ievadīto virkni.
- **Pastāv** - atribūts ar ievadīto atslēgu pastāv.
- **Nepastāv** - atribūts ar ievadīto atslēgu nepastāv.

Ja pievienojat vairākus nosacījumus, atlasiet, kā tos novērtēt:

- **And/Or** - *(noklusējums)* jāizpilda visi nosacījumi; nosacījumi ar vienādu atribūta atslēgu tiek apvienoti ar *Or*.
- **Or** - jāizpilda vismaz viens nosacījums.

Žurnāli

Pārskats

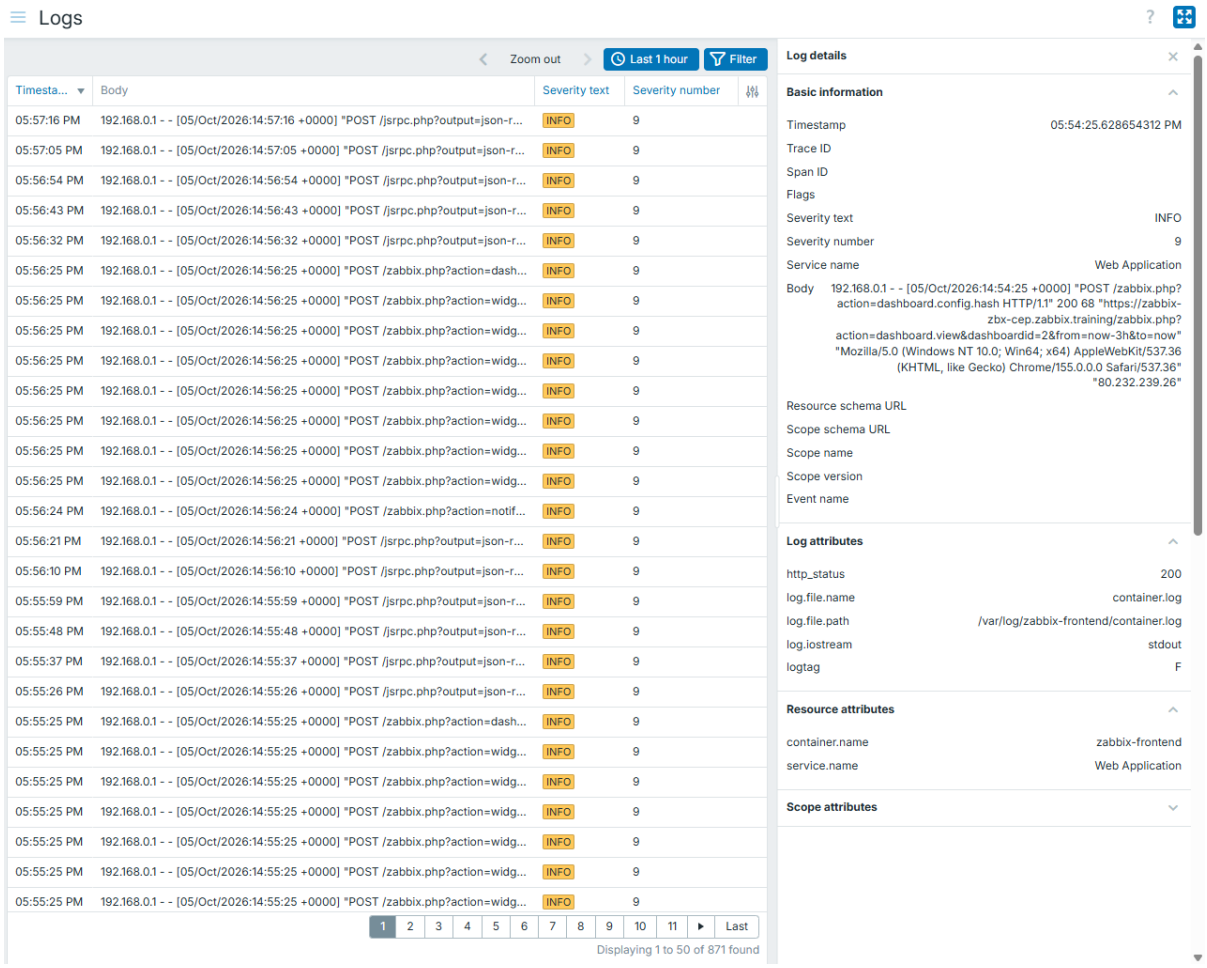
Sadaļā *APM* > *Žurnāli* tiek rādīti no jūsu lietojumprogrammām un pakalpojumiem apkopotie žurnāla ieraksti.

Žurnāla ieraksts ir notikuma teksta ieraksts (piemēram, neveiksmīgs maksājums vai datubāzes taimauts). Katram žurnāla ierakstam ir nopietnības pakāpe un ziņojums (saturs). Ja žurnāla ieraksts tika izveidots operācijas laikā, kas ir daļa no izsekojuma, tas satur arī šī izsekojuma un tvēruma ID.

Katru žurnāla ierakstu izveido **pakalpojums** (piemēram, `zabbix-frontend`; to nosaka OpenTelemetry resursa atribūts `service.name`) un instrumentācijas tvērums, tas ir, koda daļa, kas to izveidoja (piemēram, žurnalēšanas bibliotēka).

Lai nodrošinātu kontekstu par katru notikumu, žurnāla ierakstiem ir arī atslēgas un vērtības pāri jeb **atribūti**: žurnāla atribūti apraksta pašu notikumu (piemēram, `log.file.path: /var/log/zabbix/zabbix_frontend.log`), bet resursa atribūti apraksta pakalpojumu un vidi, kurā tas darbojas (piemēram, `host.name: eabf8c2a5ddc`).

Plašāku informāciju par žurnāliem skatiet OpenTelemetry dokumentācijā.

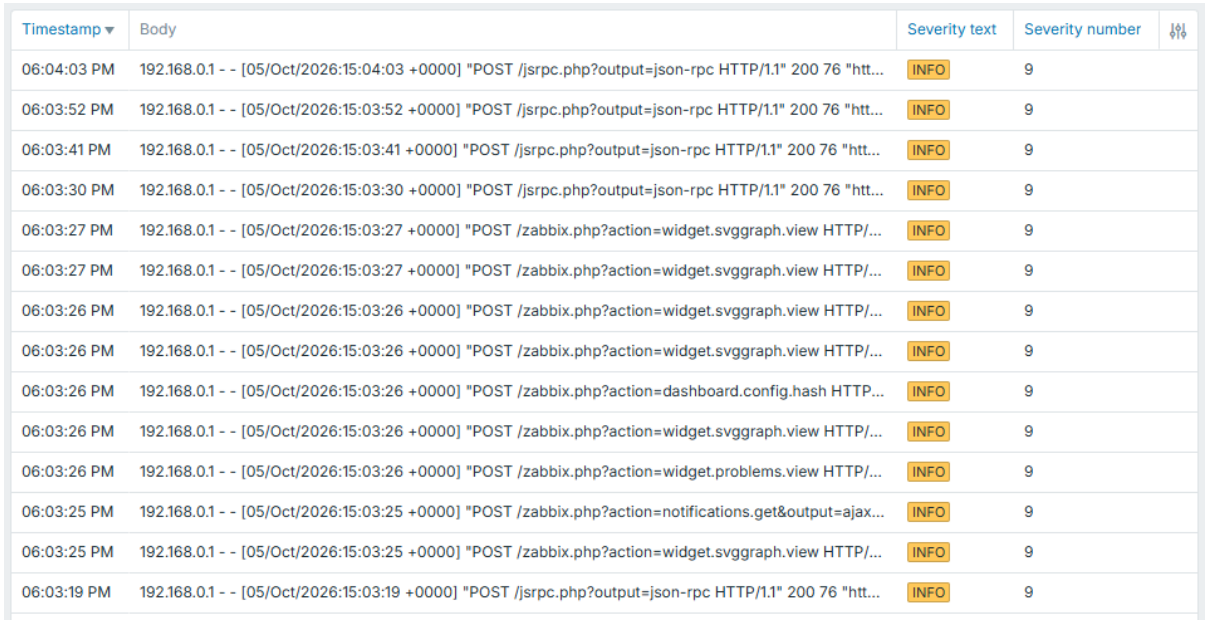


Sadaļā *APM* > *Žurnāli* ir pieejami šādi elementi:

- Poga Kioska režīms - rādīt tikai lapas saturu bez izvēlnes un galvenes.
- Laika perioda atlasītājs - atlasīt laika periodu, par kuru rādīt žurnāla ierakstus.
- Filtrs - rādīt tikai jūs interesējošos žurnāla ierakstus.
- Žurnāla saraksts - skatīt žurnāla ierakstus.
- Žurnāla informācija - skatīt atlasītā žurnāla ieraksta detalizētu informāciju.

Žurnālu saraksts

Katra rinda žurnālu sarakstā attēlo vienu žurnāla ierakstu.



Žurnāla ieraksti tiek attēloti pielāgojamā tabulā. Poga *Pielāgot tabulu* tabulas galvenē ļauj rādīt, paslēpt vai pārkārtot kolonnas,

kā arī atiestatīt izkārtojumu uz noklusējuma iestatījumiem. Varat arī mainīt katras kolonnas platumu, velkot tās labo malu; veiciet dubultklikšķi uz malas, lai atiestatītu platumu. Visi pielāgojumi tiek lietoti nekavējoties un saglabāti pašreizējam lietotājam.

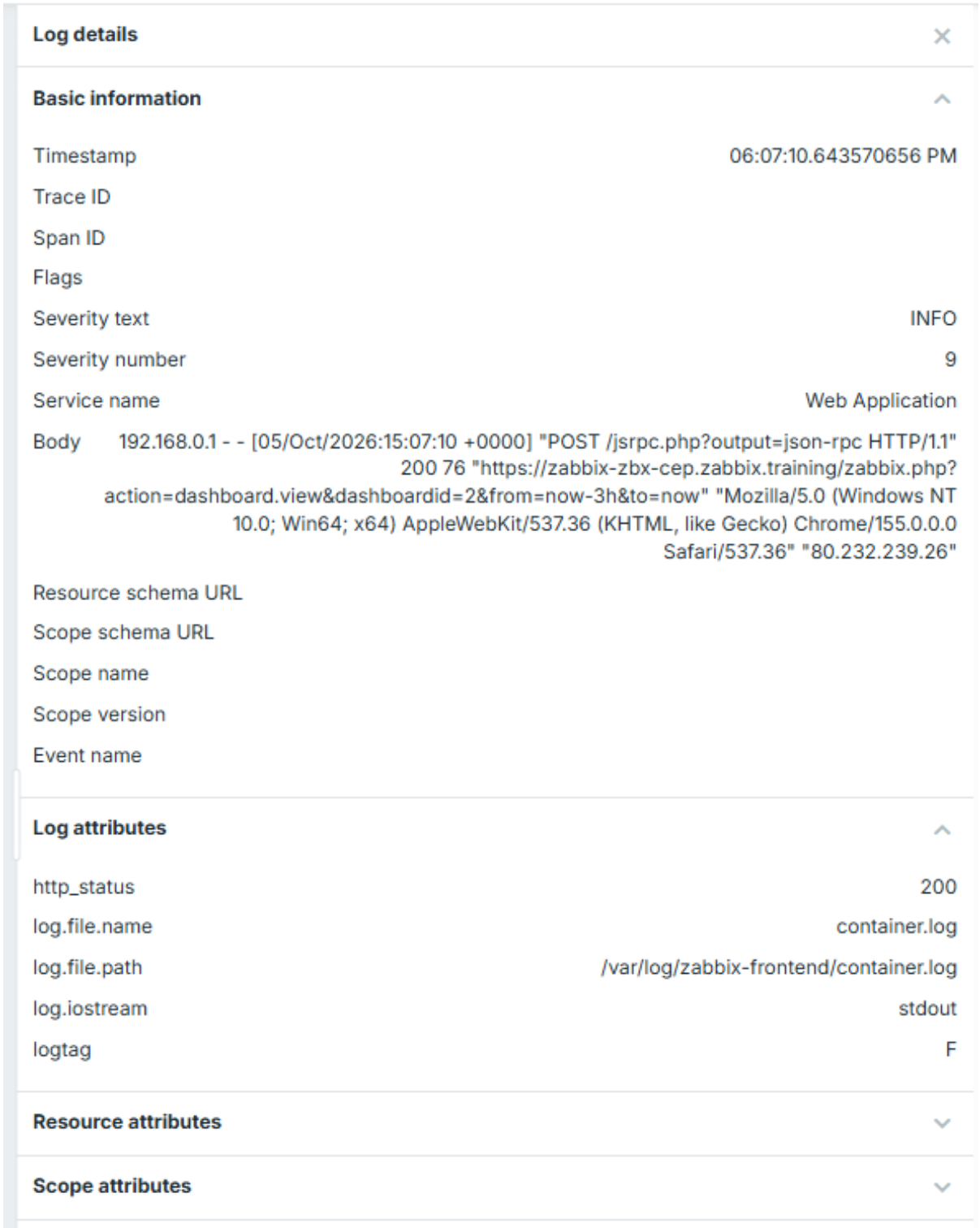
Žurnālu sarakstā var būt šādas kolonnas:

Note:
Katra kolonna atbilst kolonnai `otel_logs` tabulā telemetrijas datu avotā; kolonnu sarakstu skatiet sadaļā Atbalstītās telemetrijas datu kolonnas. Lielākajai daļai kolonnu ir tāds pats nosaukums kā datu avotā (piemēram, *Severity text* atbilst kolonnai `SeverityText`). Ja nosaukumi atšķiras, datu avota kolonna ir norādīta tālāk esošajā tabulā.

Kolonna	Apraksts
Laikspiedols	Datums un laiks, kad notika notikums.
Pamatteksts	Žurnāla ieraksta ziņojums.
Nopietnības teksts	Nopietnības līmenis, ko nodrošina lietojumprogramma (piemēram, `INFO`, `WARN` vai `ERROR`). Etiķetes krāsa ir atkarīga no nopietnības līmeņa.
Nopietnības skaitlis	Nopietnības līmenis kā skaitlis: **0** - UNSPECIFIED. **1-4** - TRACE. **5-8** - DEBUG. **9-12** - INFO. **13-16** - WARN. **17-20** - ERROR. **21-24** - FATAL.
Trace ID	Trace ID, ja tāds ir, kura laikā tika izveidots žurnāla ieraksts.
Span ID	Span ID, ja tāds ir, kura laikā tika izveidots žurnāla ieraksts.
Karodziņi	Žurnāla ieraksta trace karodziņi: **0** - trace netiek iztverts. **1** - trace tiek iztverts (ierakstīts). Datu avota kolonna: `TraceFlags`.
Pakalpojuma nosaukums	Pakalpojums, kas izveidoja žurnāla ierakstu.
Tvēruma nosaukums	Instrumentācijas tvērums, kas izveidoja žurnāla ierakstu.
Tvēruma versija	Instrumentācijas tvēruma versija.
Notikuma nosaukums	Notikuma tipa nosaukums, ja tāds ir, ko attēlo žurnāla ieraksts.
Žurnāla atribūti	Žurnāla ieraksta žurnāla atribūti. Noklikšķiniet uz ⋯ trīs punktu pogas, lai parādītu visus atribūtus.
Resursa atribūti	Žurnāla ieraksta resursa atribūti. Noklikšķiniet uz ⋯ trīs punktu pogas, lai parādītu visus atribūtus.
Tvēruma atribūti	Instrumentācijas tvēruma atribūti. Noklikšķiniet uz ⋯ trīs punktu pogas, lai parādītu visus atribūtus.

Žurnāla informācija

Noklikšķiniet uz žurnāla ieraksta žurnālu sarakstā, lai atvērtu paneli *Žurnāla informācija*, kurā tiek parādīta visa informācija par žurnāla ierakstu.

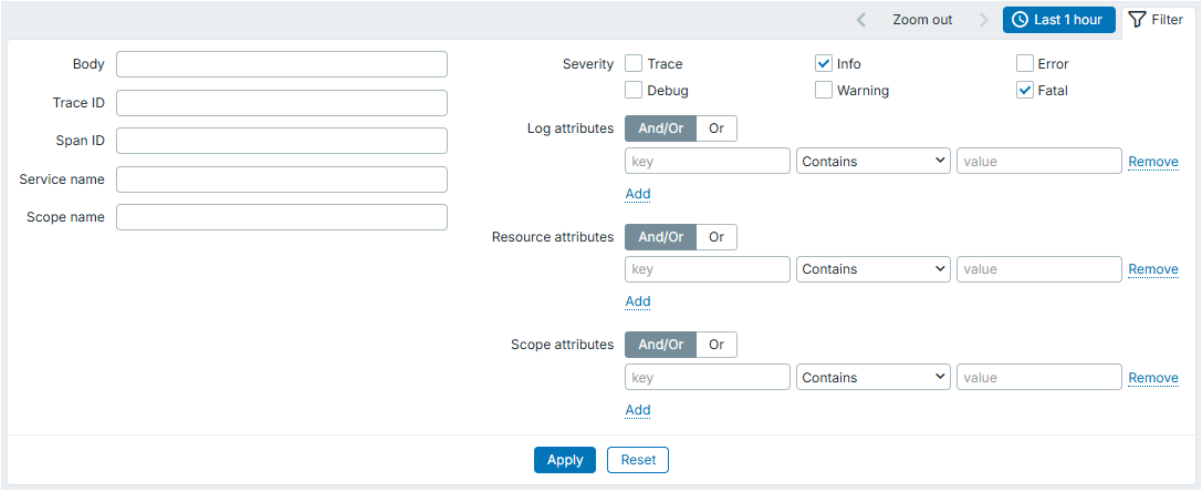


Panelī *Žurnāla informācija* var būt šādas sadaļas:

Sadaļa	Apraksts
Pamatinformācija	Žurnāla ieraksta izsekošanas ID, span ID un izsekošanas karodziņi.
Žurnāla atribūti	Žurnāla ieraksta žurnāla atribūti.
Resursa atribūti	Shēmas URL un resursa atribūti.
Tvēruma atribūti	Instrumentācijas tvēruma shēmas URL, nosaukums, versija un atribūti.

Filtra izmantošana

Varat izmantot filtru, lai parādītu tikai tos žurnāla ierakstus, kas jūs interesē.



Lai filtrētu pēc atribūtiem, ievadiet atribūta atslēgu, atlasiet operatoru un ievadiet atribūta vērtību.

Tiek atbalstīti šādi operatori:

- **Satur** - *(noklusējums)* atribūta vērtība satur ievadīto virkni.
- **Ir vienāds ar** - atribūta vērtība ir vienāda ar ievadīto virkni.
- **Nesatur** - atribūta vērtība nesatur ievadīto virkni.
- **Nav vienāds ar** - atribūta vērtība nav vienāda ar ievadīto virkni.
- **Pastāv** - atribūts ar ievadīto atslēgu pastāv.
- **Nepastāv** - atribūts ar ievadīto atslēgu nepastāv.

Ja pievienojat vairākus nosacījumus, atlasiet, kā tos novērtēt:

- **Un/Vai** - *(noklusējums)* jāizpilda visi nosacījumi; nosacījumi ar vienādu atribūta atslēgu tiek apvienoti ar *Vai*.
- **Vai** - jāizpilda vismaz viens nosacījums.

Pakalpojumi

Pārskats

Izvēlne Pakalpojumi ir paredzēta Zabbix pakalpojumu uzraudzības funkcijām.

Pakalpojumi

Pārskats

Šajā sadaļā varat skatīt vispārīgu visu Zabbix konfigurēto pakalpojumu statusu, pamatojoties uz jūsu infrastruktūru.

Pakalpojums var būt hierarhija, kas sastāv no vairākiem citu pakalpojumu līmeņiem, kurus sauc par “bērnu” pakalpojumiem. Tie ietekmē pakalpojuma kopējo statusu (skatiet arī pakalpojumu uzraudzības funkcionalitātes pārskatu).

Galvenās pakalpojuma stāvokļa kategorijas ir *OK* vai *Problēma*, kur stāvoklis *Problēma* tiek attēlots ar atbilstošo problēmas nopietnības nosaukumu un krāsu.

Skata režīmā varat uzraudzīt pakalpojumus, kā arī to statusu un citu informāciju. Šajā sadaļā varat arī konfigurēt pakalpojumu hierarhiju (pievienot vai rediģēt pakalpojumus un bērnu pakalpojumus), pārslēdzoties uz rediģēšanas režīmu.

Lai pārslēgtos no skata režīma uz rediģēšanas režīmu (un atpakaļ), noklikšķiniet uz attiecīgās pogas augšējā labajā stūrī:

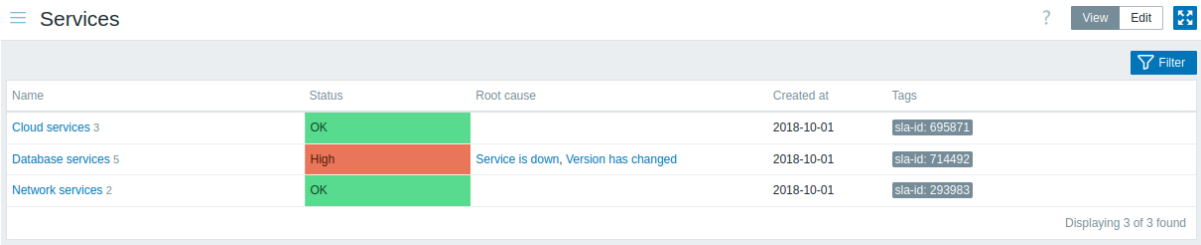
- 

- skatīt pakalpojumus
- 

- pievienot vai rediģēt pakalpojumus un bērnu pakalpojumus

Ņemiet vērā, ka piekļuve rediģēšanai ir atkarīga no lietotāja lomas iestatījumiem.

Pakalpojumu skatīšana



Tiek parādīts esošo pakalpojumu saraksts.

Parādītie dati:

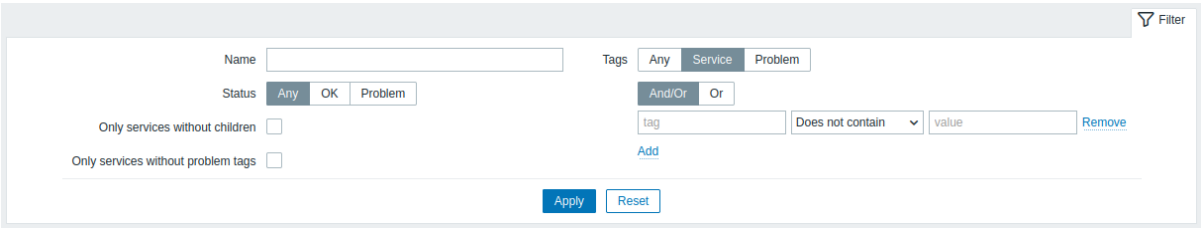
Parameter	Description
Name	Pakalpojuma nosaukums. Pakalpojuma nosaukums ir saite uz pakalpojuma informāciju. Skaitlis aiz nosaukuma norāda, cik bērnpakalpojumu ir šim pakalpojumam.
Status	Pakalpojuma stāvoklis: **OK** - nav problēmu **<problem color and severity>** - norāda uz problēmu un tās nopietnību. Ja ir vairākas problēmas, tiek parādīta viskritiskākās problēmas krāsa un nopietnība.
Root cause	Tiek uzskaitītas pamatproblēmas, kas tieši vai netieši ietekmē pakalpojuma stāvokli. Tās pašas problēmas tiek uzskaitītas tā, kā tās atgriež {SERVICE.ROOTCAUSE} makro. Noklikšķiniet uz problēmas nosaukuma, lai skatītu vairāk informācijas par to sadaļā *Monitoring* > *Problems*. Problēmas, kas neietekmē pakalpojuma stāvokli, sarakstā netiek iekļautas.
Created at	Tiek parādīts laiks, kad pakalpojums tika izveidots.
Tags	Tiek parādīti pakalpojuma tagi. Tagi tiek izmantoti, 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.



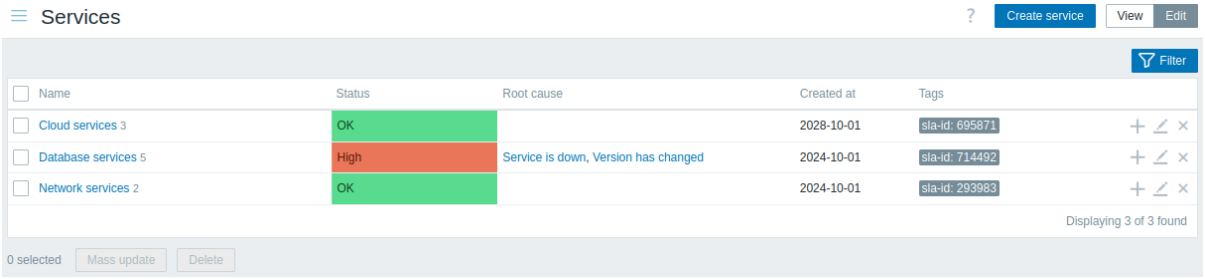
Parametrs	Apraksts
Name	Filtrēt pēc pakalpojuma nosaukuma.
Status	Filtrēt pēc pakalpojuma stāvokļa.
Only services without children	Atzīmējiet izvēles rūtiņu, lai attēlotu tikai pakalpojumus bez apakšpakalpojumiem. Šis parametrs ir pieejams tikai pakalpojuma rediģēšanas režīmā.
Only services without problem 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. Rediģēšanas režīmā saraksts tiek papildināts ar izvēles rūtiņām pirms ierakstiem, kā arī šīm papildu opcijām:

-  - pievienot šim pakalpojumam pakārtotu pakalpojumu
-  - rediģēt šo pakalpojumu
-  - dzēst šo pakalpojumu

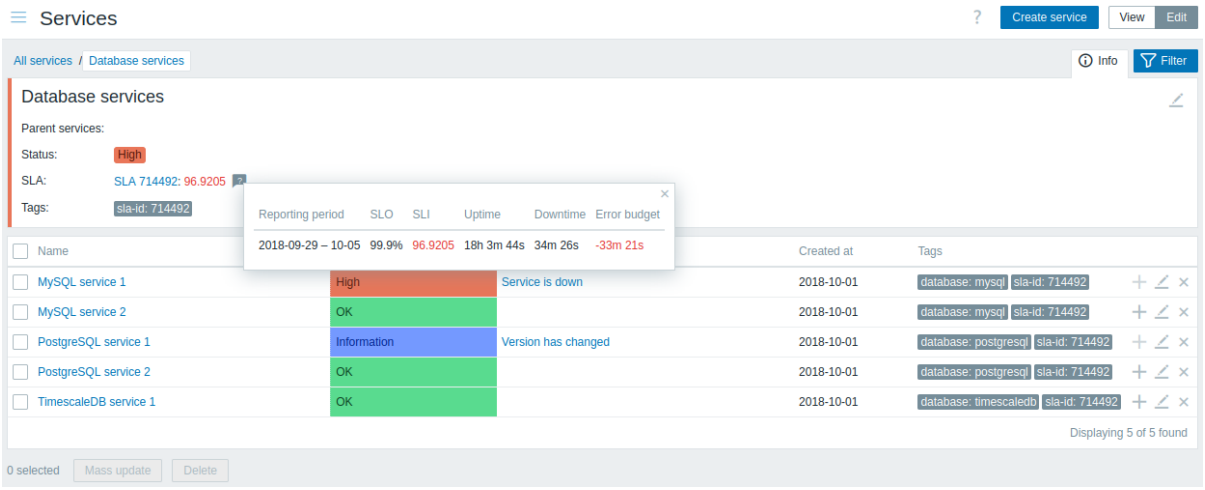


Lai konfigurētu jaunu pakalpojumu, noklikšķiniet uz pogas *Create service* augšējā labajā stūrī.

Pakalpojuma informācija

Lai piekļūtu pakalpojuma informācijai, noklikšķiniet uz pakalpojuma nosaukuma. Lai atgrieztos pie visu pakalpojumu saraksta, noklikšķiniet uz *All services*.

Pakalpojuma informācija ietver informācijas lodziņu un pakārtoto pakalpojumu 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 pašreizējie SLA, formātā `SLA name:service level indicator`. 'SLA name' ir arī saite uz šī pakalpojuma SLA pārskatu. Ja novietojat peli uz informācijas lodziņa blakus pakalpojuma līmeņa indikatoram (SLI), tiek parādīts uznirstošs informācijas saraksts ar SLI detaļām. Pakalpojuma līmeņa indikators parāda pašreizējo pakalpojuma līmeni procentos.
- Pakalpojuma tagi

Informācijas lodziņā ir arī saite uz pakalpojuma konfigurāciju.

Lai izmantotu pakārtoto pakalpojumu filtru, noklikšķiniet uz cilnes *Filter*.

Rediģēšanas režīmā pakārtoto pakalpojumu sarakstam tiek pievienotas papildu rediģēšanas opcijas:

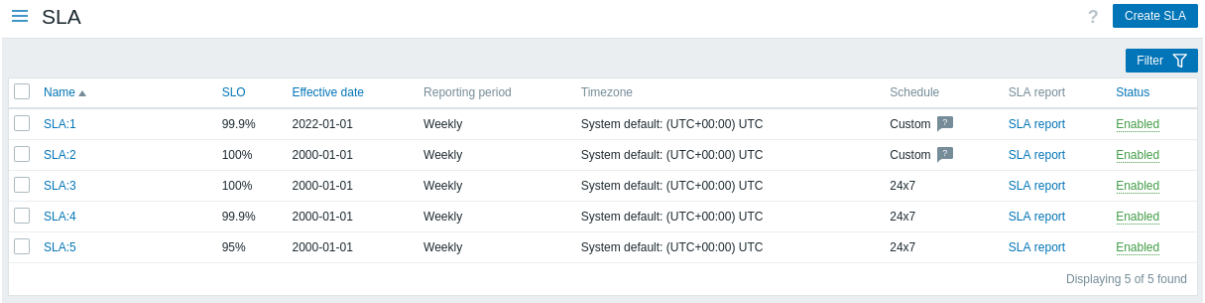
-  - pievienot šim pakalpojumam pakārtoto pakalpojumu
-  - rediģēt šo pakalpojumu
-  - dzēst šo pakalpojumu

SLA

Pārskats

Šajā sadaļā var skatīt un konfigurēt SLA.

SLA



Tiek parādīts konfigurēto SLA saraksts. *Piezīme*, ka tiks parādīti tikai tie SLA, kas attiecas uz lietotājam pieejamajiem pakalpojumiem (tikai lasīšanas režīmā, ja vien lietotāja lomai nav iespējota *Manage SLA*).

Parādītie dati:

Parameter	Description
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.
Reporting period	Tiek parādīts SLA pārskatā izmantotais periods - *daily*, *weekly*, *monthly*, *quarterly* vai *annually*.

Parameter	Description
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.

SLA pārskats

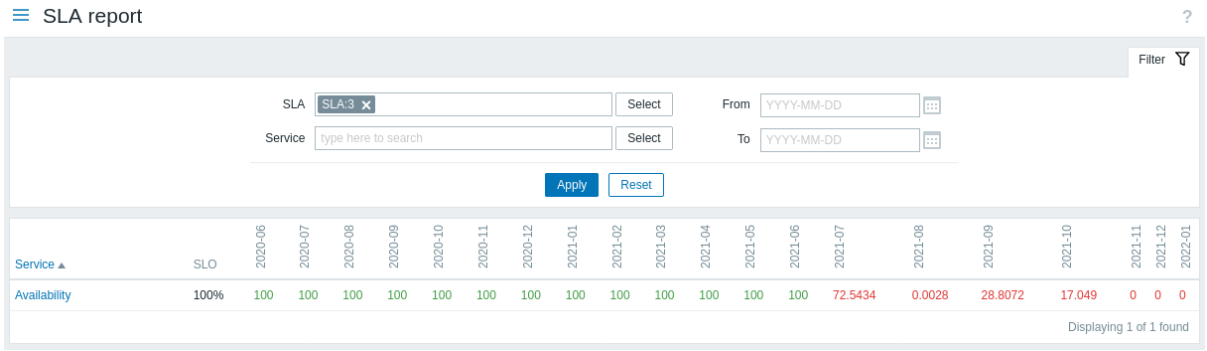
Pārskats

Šajā sadaļā var skatīt SLA pārskatus, pamatojoties uz filtrā atlasītajiem kritērijiem.

SLA pārskatus var attēlot arī kā informācijas paneļa logrīku.

Pārskats

Filtrs ļauj atlasīt pārskatu pēc SLA nosaukuma, kā arī pakalpojuma nosaukuma. Ir iespējams arī ierobežot attēlojamo periodu.

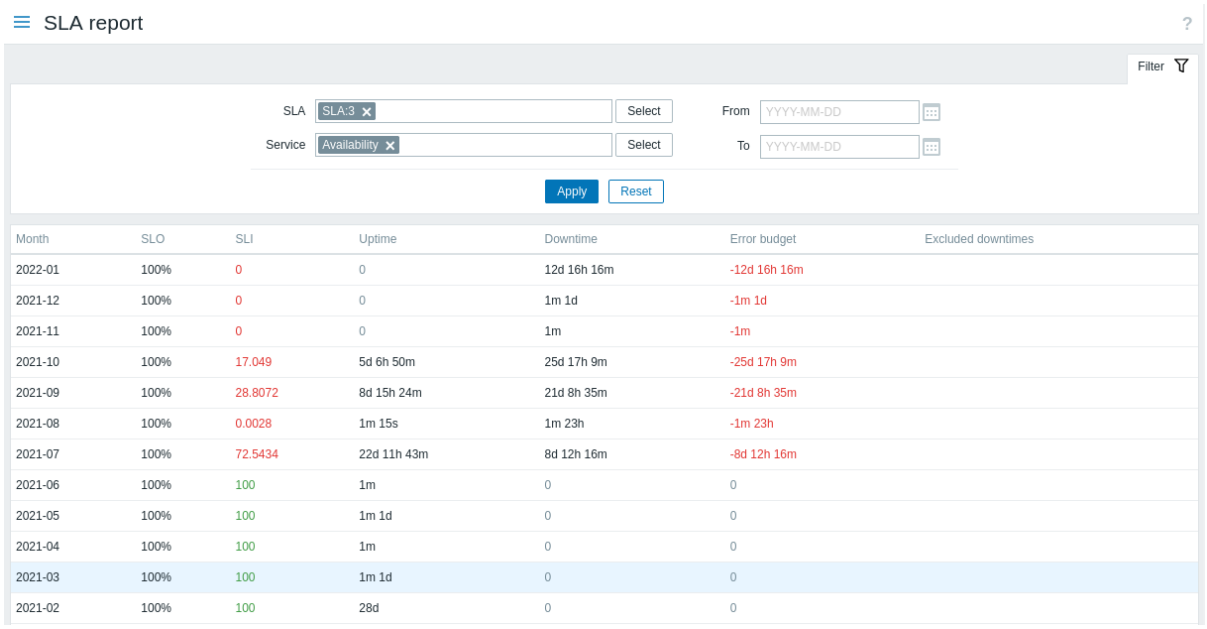


Katrā kolonnā (periodā) tiek attēlots attiecīgā perioda SLI. SLI, kas neatbilst iestatītajam SLO, tiek izcelti sarkanā krāsā.

Pārskatā tiek attēloti 20 periodi. Ja ir norādīts gan datums *No*, gan datums *Līdz*, var attēlot ne vairāk kā 100 periodus.

Pārskata detaļas

Ja pārskatā noklikšķināsiet uz pakalpojuma nosaukuma, varēsiet atvērt citu pārskatu, kas parāda detalizētāku skatu.



Ņemiet vērā, ka negatīvs problēmas ilgums neietekmē SLA aprēķinu vai pārskatu veidošanu.

Inventārs

Pārskats

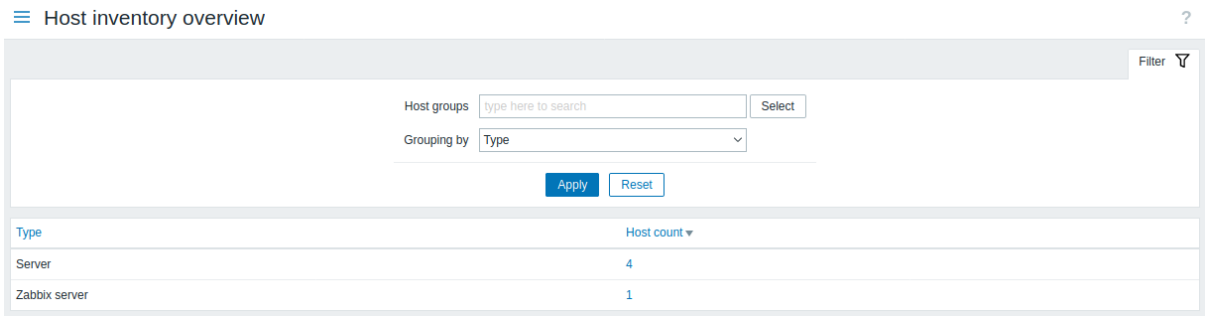
Izvēlnē Inventārs ir sadaļas, kas nodrošina hostu inventāra datu pārskatu pēc izvēlēta parametra, kā arī iespēju skatīt hosta inventāra detalizētu informāciju.

Pārskats

Pārskats

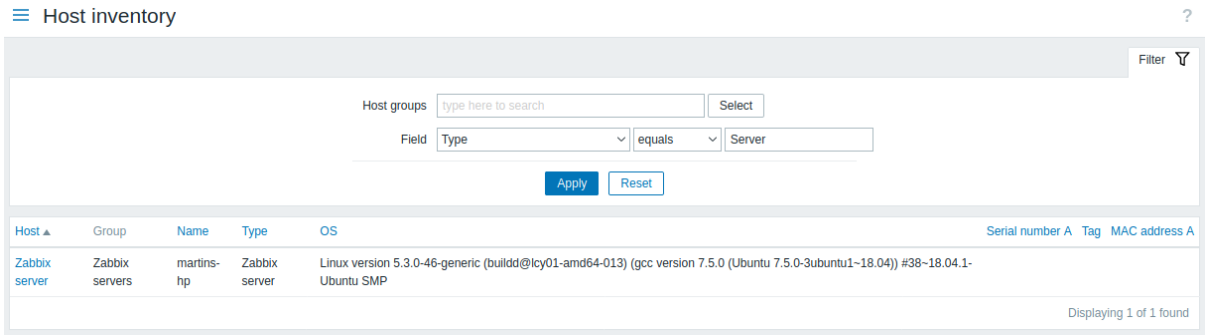
Sadaļa *Inventory* > *Overview* nodrošina veidus, kā iegūt host inventarizācijas datu pārskatu.

Lai tiktu parādīts pārskats, izvēlieties host grupas (vai nevienu) un inventarizācijas lauku, pēc kura rādīt datus. Tiks parādīts hostu skaits, kas atbilst katram izvēlētā lauka ierakstam.



Pārskata pilnīgums ir atkarīgs no tā, cik daudz inventarizācijas informācijas ir uzturēts hostos.

Skaitļi kolonnā *Host count* ir saites; tās noved pie šo hostu filtrēšanas tabulā *Host Inventories*.

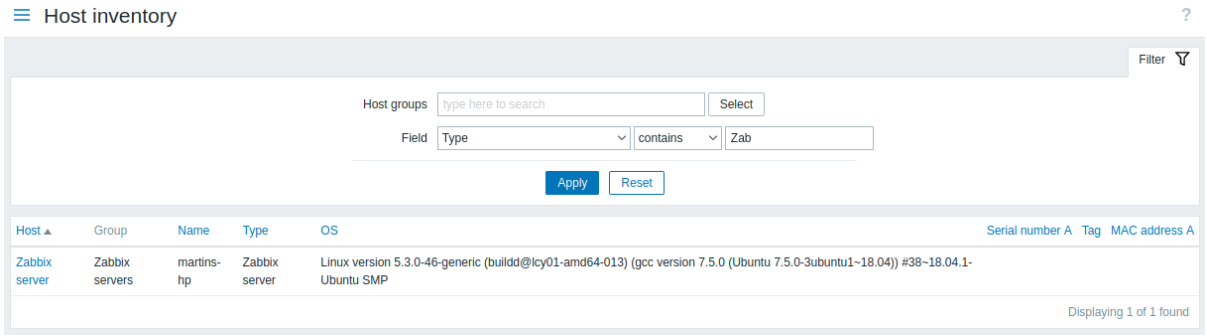


Hosts

Pārskats

Sadaļā *Inventory* > *Hosts* tiek attēloti hosts inventarizācijas dati.

Varat filtrēt hosts pēc hostu grupas(-ām) un pēc jebkura inventarizācijas lauka, lai parādītu tikai tos hosts, kas jūs interesē.

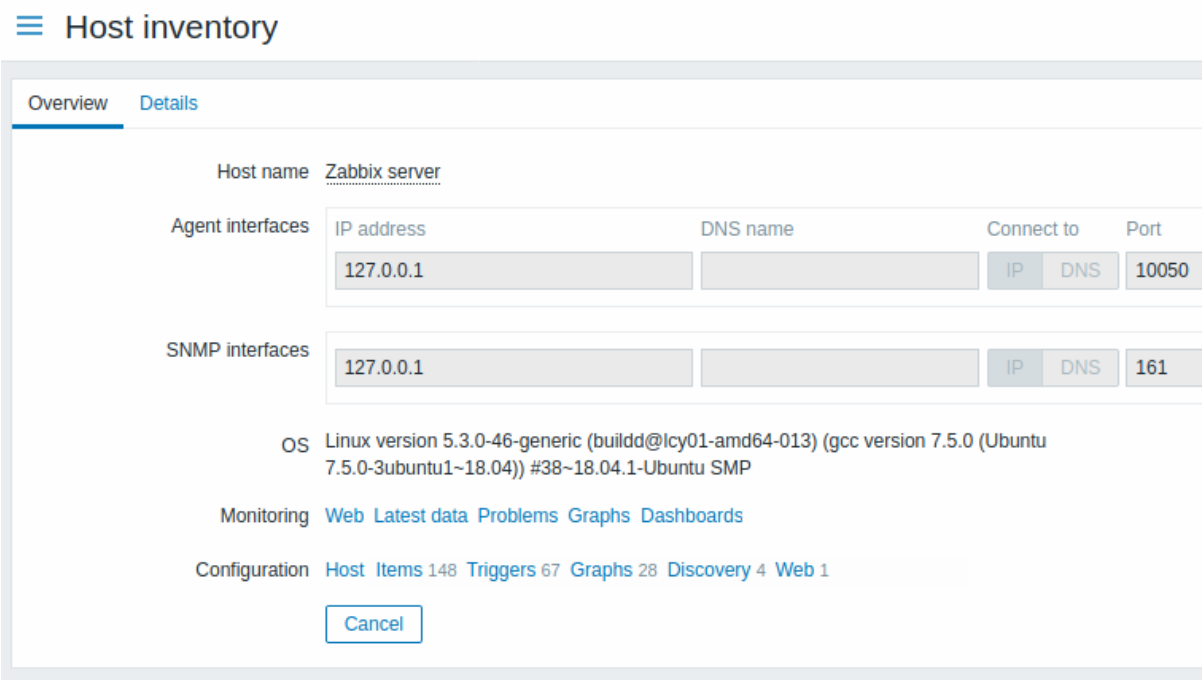


Lai parādītu visu hostu inventarizāciju, noklikšķiniet uz pogas *Reset*.

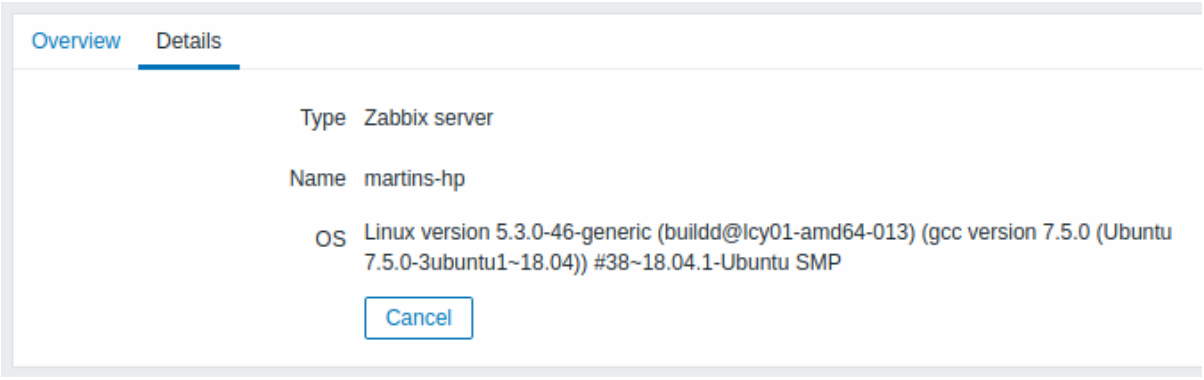
Lai gan tabulā tiek attēloti tikai daži galvenie inventarizācijas lauki, jūs varat arī skatīt visu pieejamo inventarizācijas informāciju par konkrēto hostu. Lai to izdarītu, noklikšķiniet uz hosta nosaukuma pirmajā kolonnā.

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.

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.

Sistēmas informācija

Pārskats

Sadaļā *Pārskati* > *Sistēmas informācija* tiek parādīts Zabbix servera un sistēmas svarīgāko datu kopsavilkums. Sistēmas dati tiek apkopoti, izmantojot iekšējos vienumus.

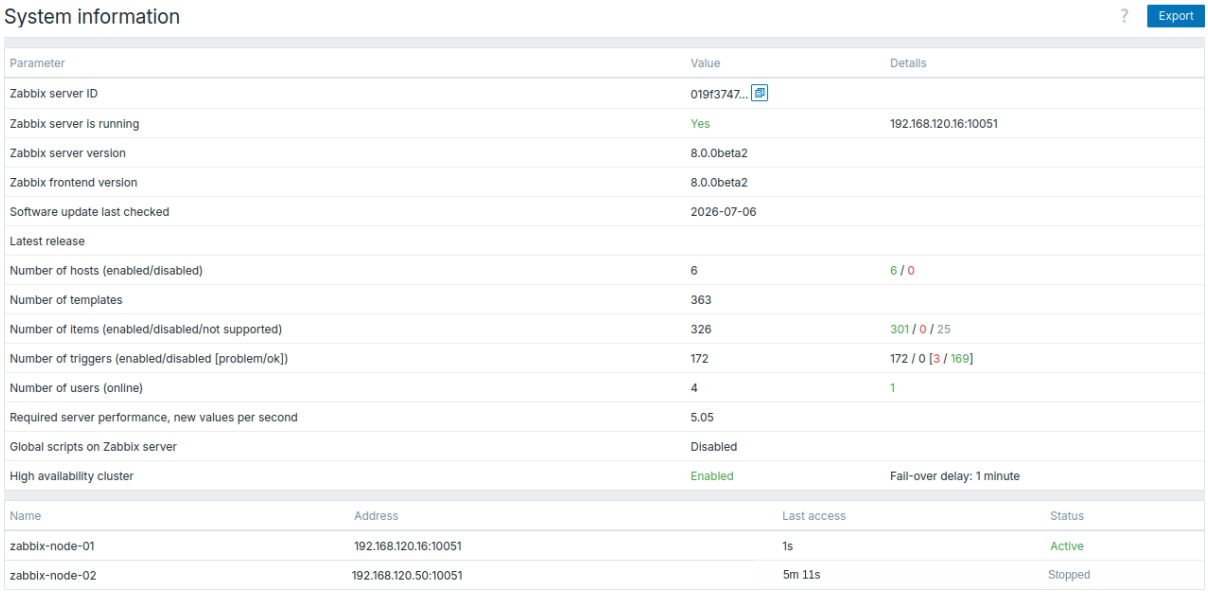
Sistēmas informācijas pārskats ir redzams tikai Zabbix *Super Admin* un Zabbix *Admin* lietotājiem.

Ņemiet vērā, ka augstas pieejamības iestatījumā ir iespējams novirzīt sistēmas informācijas avotu (servera instanci). Lai to izdarītu, rediģējiet lietotāja saskarnes konfigurācijas failu — atkomentējiet un iestatiet `$ZBX_SERVER` vai gan `$ZBX_SERVER`, gan `$ZBX_SERVER_PORT` uz citu serveri, nevis to, kas tiek rādīts kā aktīvs. Ņemiet vērā: ja tiek iestatīts tikai `$ZBX_SERVER`, `$ZBX_SERVER_PORT` tiks izmantota noklusējuma vērtība (10051).

Ja ir iespējots augstas pieejamības iestatījums, zem sistēmas statistikas tiek parādīts atsevišķs bloks ar detalizētu informāciju par augstas pieejamības mezgliem.

Sistēmas informācija ir pieejama arī kā informācijas paneļa logrīks.

Sistēmas statistika



Parādītie dati:

Parametrs	Vērtība	Detalizēta informācija
Zabbix servera ID	Zabbix servera unikālais identifikators (UUIDv7).	Servera ID tiek ģenerēts servera pirmās palaišanas laikā. Tiek parādītas pirmās 8 rakstzīmes, kam seko daudzpunkte. Novietojot kursoru virs servera ID vērtības vai noklikšķinot uz tās, tiek parādīts rīka padoms ar pilno vērtību. Noklikšķinot uz ikonas blakus vērtībai, ID tiek nokopēts starpliktuvē. Ja ir iespējota augsta pieejamība, katram mezglam ir savs mezgla ID komandrindas adresēšanai, taču visi mezgli tiek grupēti zem viena servera ID, ko ģenerē mezgls, kas tika palaists pirmais (vai iepriekš bija atsevišķs serveris), neatkarīgi no tā, kurš mezgls pašlaik ir aktīvs.
Zabbix serveris darbojas	Zabbix servera statuss: **Jā** - serveris darbojas **Nē** - serveris nedarbojas *Piezīme:* Lai parādītu pārējo informāciju, tīmekļa lietotāja saskarnei serverim ir jādarbojas un serverī jābūt palaistam vismaz vienam trapper procesam (parametram StartTrappers `zabbix_server.conf` failā jābūt > 0).	Zabbix servera atrašanās vieta un ports.
Zabbix servera versija	Tiek parādīts pašreizējais servera versijas numurs. *Piezīme:* Tas tiek parādīts tikai tad, ja Zabbix serveris darbojas.	Tiek parādīts servera versijas statuss: **Atjaunināta** - tiek izmantota jaunākā versija; **Pieejams jauns atjauninājums** - ir pieejama jaunāka versija; **Novecojusi** - šīs versijas pilnais atbalsta periods ir beidzies. Šī informācija ir pieejama tikai tad, ja Zabbix servera konfigurācijā ir iespējota programmatūras atjauninājumu pārbaude. Nekas netiek parādīts, ja pēdējā programmatūras atjauninājumu pārbaude veikta pirms vairāk nekā nedēļas vai nav datu par pašreizējo versiju.

Parametrs	Vērtība	Detalizēta informācija
Zabbix lietotāja saskarnes versija	Tiek parādīts Zabbix lietotāja saskarnes versijas numurs.	Tiek parādīts Zabbix lietotāja saskarnes versijas statuss: **Atjaunināta** - tiek izmantota jaunākā versija; **Pieejams jauns atjauninājums** - ir pieejama jaunāka versija; **Novecojusi** - šīs versijas pilnais atbalsta periods ir beidzies. Šī informācija ir pieejama tikai tad, ja Zabbix servera konfigurācijā ir iespējota programmatūras atjauninājumu pārbaude. Nekas netiek parādīts, ja pēdējā programmatūras atjauninājumu pārbaude veikta pirms vairāk nekā nedēļas vai nav datu par pašreizējo versiju.
Pēdējā programmatūras atjauninājumu pārbaude	Tiek parādīts pēdējās Zabbix programmatūras atjauninājumu pārbaudes datums. Šī informācija ir pieejama tikai tad, ja Zabbix servera konfigurācijā ir iespējota programmatūras atjauninājumu pārbaude.	
Jaunākais laidiens	Tiek parādīts jaunāka laidiena numurs (ja tāds ir pieejams) pašreizējai Zabbix versijai. Šī informācija ir pieejama tikai tad, ja Zabbix servera konfigurācijā ir iespējota programmatūras atjauninājumu pārbaude. Nekas netiek parādīts, ja pēdējā programmatūras atjauninājumu pārbaude veikta pirms vairāk nekā nedēļas vai nav datu par pašreizējo versiju.	Tiek parādīta saite uz jaunākā pieejamā Zabbix laidiena izlaiduma piezīmēm.
Hostu skaits	Tiek parādīts kopējais konfigurēto hostu skaits.	Uzraudzīto/neuzraudzīto hostu skaits.
Veidņu skaits	Tiek parādīts kopējais veidņu skaits.	
Vienumu skaits	Tiek parādīts kopējais vienumu skaits.	Uzraudzīto/atspējoto/neatbalstīto hostu līmeņa vienumu skaits. Vienumi atspējotos hostos tiek uzskaitīti kā atspējoti.
Trigeru skaits	Tiek parādīts kopējais trigeru skaits.	Iespējoto/atspējoto hostu līmeņa trigeru skaits; iespējoto trigeru sadalījums atbilstoši stāvokļiem “Problēma”/“OK”. Trigeri, kas uzskaitīti stāvoklī “OK”, ietver trigerus ar statusu “Nezināms”. Trigeri, kas ir atkarīgi no atspējotiem vienumiem vai piešķirti atspējotiem hostiem, tiek uzskaitīti kā atspējoti.
Lietotāju skaits	Tiek parādīts kopējais konfigurēto lietotāju skaits.	Tiešsaistē esošo lietotāju skaits.
Nepieciešamā servera veiktspēja, jaunas vērtības sekundē	Tiek parādīts paredzamais jauno vērtību skaits sekundē, ko apstrādā Zabbix serveris.	*Nepieciešamā servera veiktspēja* ir aptuvens novērtējums, kas var būt noderīgs kā vadlīnija. Lai iegūtu precīzu apstrādāto vērtību skaitu, izmantojiet `zabbix[wcache,values,all]` iekšējo vienumu. Aprēķinā tiek iekļauti iespējotie vienumi no uzraudzītajiem hostiem. Žurnāla vienumi tiek uzskaitīti kā viena vērtība katrā vienuma atjaunināšanas intervālā. Tiek uzskaitītas regulāra intervāla vērtības; elastīga intervāla un plānošanas intervāla vērtības netiek uzskaitītas. Aprēķins netiek pielāgots “nodata” uzturēšanas perioda laikā. Trapper vienumi netiek uzskaitīti.
Globālie skripti Zabbix serverī	Šajā laukā tiek parādīts **Atspējots**, ja globālie skripti Zabbix serverī ir atspējoti, servera konfigurācijā iestatot EnableGlobalScripts=0.	

Parametrs	Vērtība	Detalizēta informācija
Augstas pieejamības klasteris	Zabbix servera augstas pieejamības klastera statuss: **Atspējots** - atsevišķs serveris **Iespējots** - pastāv vismaz viens augstas pieejamības mezgls	Ja tas ir iespējots, tiek parādīta pārslēgšanās aizkave.

Sistēmas informācijā tiek parādīts arī kļūdas ziņojums šādos gadījumos:

- Izmantotajai datubāzei nav nepieciešamās rakstzīmju kopas vai kolācijas (UTF-8).
- Datubāzes versija ir zemāka vai augstāka par atbalstīto diapazonu (pieejams tikai lietotājiem ar *Super administratora lomas* tipu).
- TimescaleDB mājturība ir konfigurēta nepareizi (vēstures vai tendenču tabulas satur saspiestus fragmentus, bet opcijas *Ignorēt vienuma vēstures periodu* vai *Ignorēt vienuma tendenču periodu* ir atspējotas).

Pogas

Poga lapas augšējā labajā stūrī piedāvā šādu opciju:

Export	Eksportēt sistēmas informāciju JSON failā. Fails tiek saglabāts kā `Zabbix-system-information-{time}.json`, kur `{time}` ir Unix laikspiedols, kas norāda, kad dati tika sagatavoti serverī. Ja ir iespējota augsta pieejamība, tiek eksportēta arī augstas pieejamības mezglu informācija.

Augstas pieejamības mezgli

Ja ir iespējots augstas pieejamības klasteris, tiek parādīts papildu datu bloks, kurā redzams katra augstas pieejamības mezgla statuss.



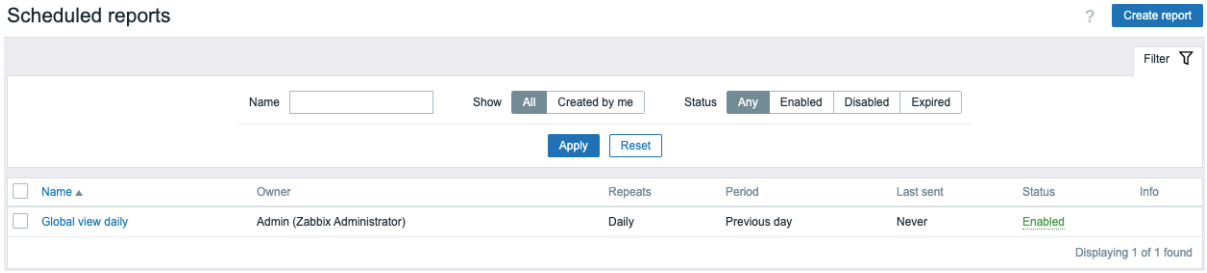
Parādītie dati:

Column	Description
Name	Mezgla nosaukums, kā tas definēts servera konfigurācijā.
Address	Mezgla IP adrese un ports.
Last access	Mezgla pēdējās piekļuves laiks. Novietojot kursoru virs šūnas, tiek parādīts pēdējās piekļuves laikspiedols pilnā formātā.
Status	Mezgla statuss. Tabulas rindas tiek kārtotas pēc šiem statusiem šādā prioritātes secībā: **Active** - mezgls darbojas un ir pieejams **Unavailable** - mezgls nav redzēts ilgāk nekā kļūmjpārslēgšanas aizkave (iespējams, vēlēsieties noskaidrot iemeslu) **Stopped** - mezgls ir apturēts vai nevarēja startēt (iespējams, vēlēsieties to startēt vai dzēst) **Standby** - mezgls darbojas un gaida

Plānotie pārskati

Pārskats

Sadaļā *Pārskati* > *Ieplānotie pārskati* lietotāji ar pietiekamām atļaujām var konfigurēt informācijas paneļu PDF versiju plānoto ģenerēšanu, kas tiks nosūtītas pa e-pastu norādītajiem adresātiem.



Atvēršanas ekrānā tiek parādīta informācija par ieplānotajiem pārskatiem, ko var filtrēt ērtākai navigācijai - skatiet tālāk sadaļu Filtra izmantošana.

Tiek parādīti šādi dati:

Kolonna	Apraksts
Nosaukums	Pārskata nosaukums. Noklikšķinot uz tā, tiek atvērta pārskata konfigurācijas veidlapa.
Īpašnieks	Lietotājs, kurš izveidoja pārskatu.
Atkārtojas	Pārskata ģenerēšanas biežums (katru dienu/nedēļu/mēnesi/gadu).
Periods	Laika periods, par kuru tiek sagatavots pārskats (iepriekšējā diena, nedēļa, mēnesis vai gads).
Pēdējoreiz nosūtīts	Datums un laiks, kad tika nosūtīts jaunākais pārskats.
Statuss	Pārskata pašreizējais statuss (iespējots/atspējots/derīguma termiņš beidzies). Lietotāji ar pietiekamām atļaujām var mainīt statusu, noklikšķinot uz tā - no "Iespējots" uz "Atspējots" (un otrādi); no "Derīguma termiņš beidzies" uz "Atspējots" (un otrādi). Lietotājiem bez pietiekamām tiesībām statuss nav noklikšķināms.
Informācija	Tiek parādītas informatīvas ikonas: Sarkana ikona norāda, ka pārskata ģenerēšana neizdevās; novietojot kursoru virs tās, tiek parādīts rīka padoms ar informāciju par kļūdu. Dzeltena ikona norāda, ka pārskats tika ģenerēts, taču nosūtīšana dažiem (vai visiem) adresātiem neizdevās vai arī pārskata derīguma termiņš ir beidzies; novietojot kursoru virs tās, tiek parādīts rīka padoms 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:

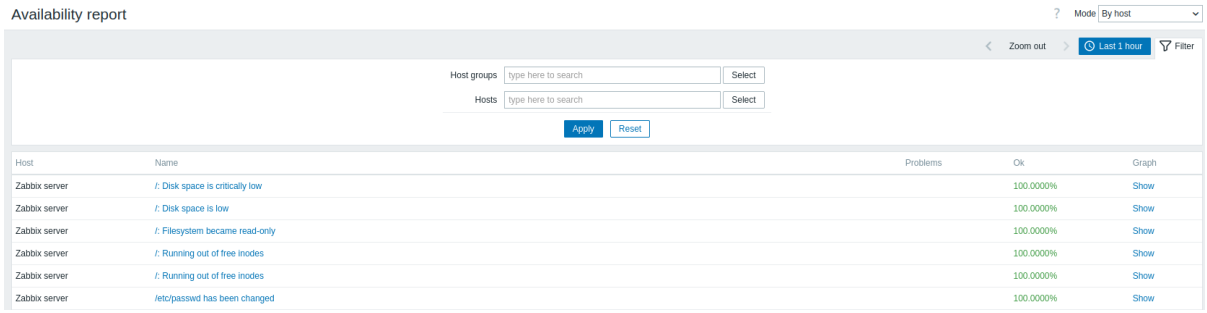
- Sarakstā atzīmējiet to atskaišu izvēles rūtiņas, kuras vēlaties atjaunināt
- Noklikšķiniet uz vajadzīgās pogas zem saraksta, lai veiktu izmaiņas (*Iespējot*, *Atspējot* vai *Dzēst*)

Pieejamības pārskats

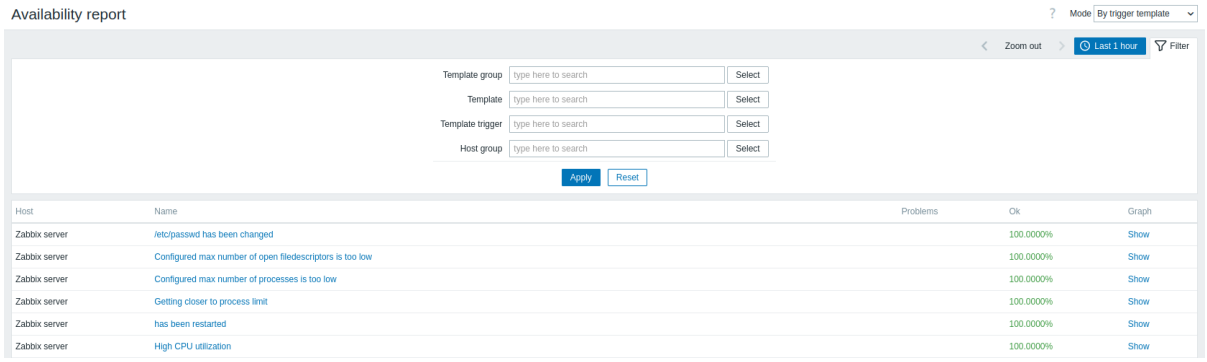
Pārskats

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 uzlabotu meklēšanas veiktspēju, dati tiek meklēti ar neatrisinātiem makro.

Filtrs atrodas zem sadaļas *Pieejamības pārskats* 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.

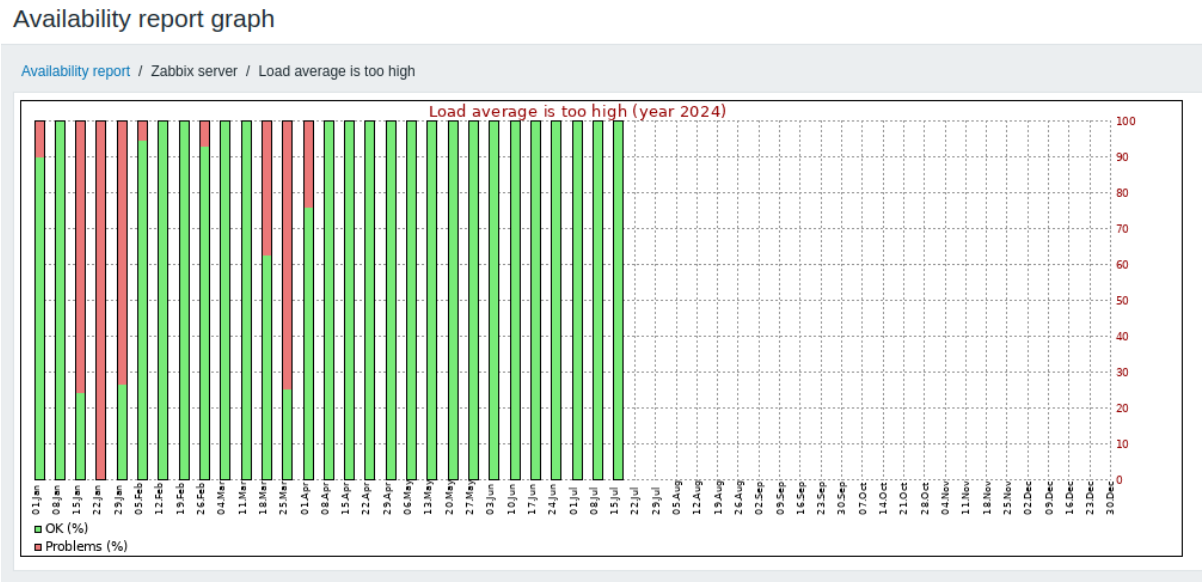
Filtrēšana pēc hosta

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 parādīta joslu diagrammā, kurā katra josla attēlo vienu pagājušo pašreizējā gada nedēļu.



Joslas zaļā krāsa apzīmē laiku, kad viss bija kārtībā, bet sarkanā - problēmu laiku.

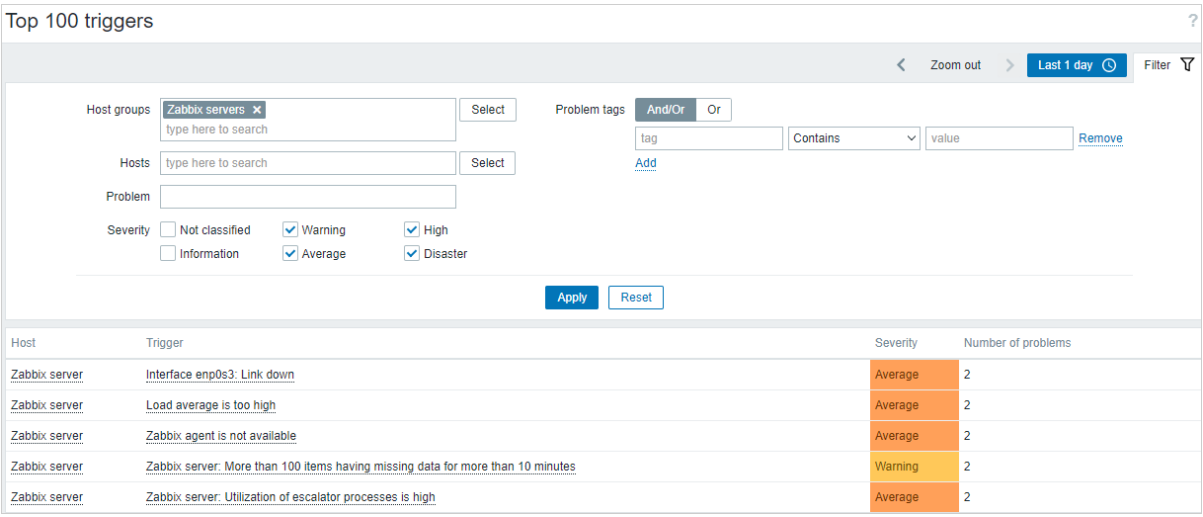
Uzturēšanas periodu ietekme

Uzturēšana automātiski neizslēdz laiku no pieejamības pārskata. Tikai uzturēšana, kas konfigurēta kā *No data collection*, aptur aptauju veikšanu (tāpēc problēmas netiek ģenerētas) un līdz ar to atstāj pieejamības procentu nemainīgu ietekmētajiem trigeriem.

Top 100 trigeri

Pārskats

Sadaļā *Pārskati > 100 populārākie trigeri* varat skatīt trigerus ar lielāko konstatēto problēmu skaitu atlasītajā periodā.



Gan hosts, gan trigera kolonnas ieraksti ir saites, kas piedāvā noderīgas iespējas:

- hosts - noklikšķinot uz hosta nosaukuma, tiek atvērta hosta izvēlne
- trigerim - noklikšķinot uz trigera nosaukuma, tiek atvērtas saites uz jaunākajiem notikumiem, vienkāršu 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 nopietnības. Norādot vecāku hostu grupu, tiek netieši atlasītas visas ligzdotās hostu grupas. Lai uzlabotu meklēšanas veiktspēju, dati tiek meklēti ar neatrisinātiem makro.

Filtrs atrodas zem sadaļas *100 populārākie trigeri* nosaukuma. To var atvērt un sakļaut, labajā pusē noklikšķinot uz cilnes *Filtrs*.

Laika perioda atlasītājs

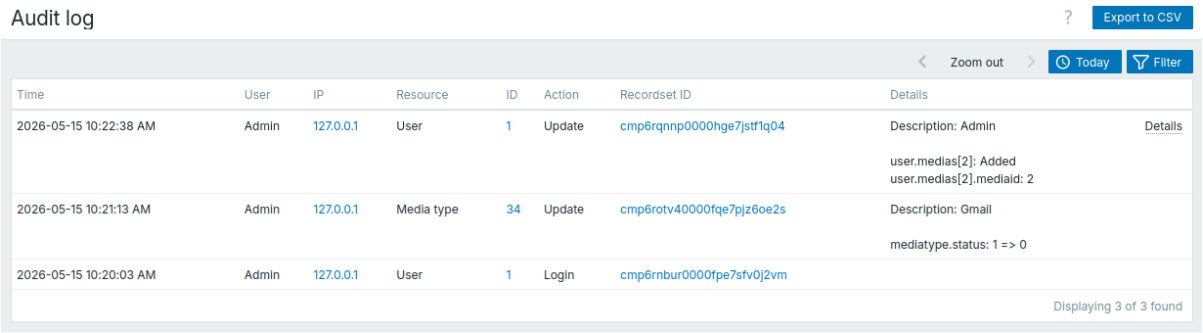
Laika perioda atlasītājs blakus filtram ļauj ar vienu peles klikšķi atlasīt bieži nepieciešamos periodus. Papildinformāciju skatiet sadaļā Laika perioda un hosta atlasītāji.

Audita žurnāls

Pārskats

Sadaļā *Reports* > *Audit log* var skatīt lietotāju un sistēmas darbību ierakstus.

Note:
Lai auditācijas ieraksti tiktu savākti un parādīti, sadaļā *Administration* > *Audit log* ir jāatzīmē izvēles rūtiņa *Enable audit logging*. Ja šis iestatījums nav iespējots, darbību vēsture netiks ierakstīta datubāzē un netiks parādīta auditācijas žurnālā.



Auditācijas žurnālā tiek parādīti šādi dati:

Column	Description
Time	Auditācijas ieraksta laikspiedols.
User	Lietotājs, kurš veica darbību.
IP	IP adrese, no kuras tika sākta darbība. Noklikšķinot uz hipersaites, auditācijas žurnāla ieraksti tiks filtrēti pēc šīs IP adreses.
Resource	Ietekmētā resursa tips (*API token*, *Action*, *Authentication*, *Autoregistration* u. c.).
ID	Ietekmētā resursa ID. Noklikšķinot uz hipersaites, auditācijas žurnāla ieraksti tiks filtrēti pēc šī resursa ID.
Action	Darbības tips (*Add*, *Configuration refresh*, *Delete*, *Execute*, *Failed login*, *History clear*, *Login*, *Logout*, *Push*, *Update*).
Recordset ID	Kopīgs ID visiem auditācijas žurnāla ierakstiem, kas izveidoti vienas un tās pašas darbības rezultātā. Piemēram, saistot veidni ar hostu, katrai mantotajai veidnes entītijai (vienums, trigeris u. c.) tiek izveidots atsevišķs auditācijas žurnāla ieraksts - visiem šiem ierakstiem būs viens un tas pats *Recordset ID*. Noklikšķinot uz hipersaites, auditācijas žurnāla ieraksti tiks filtrēti pēc šī *Recordset ID*.
Details	Resursa apraksts un detalizēta informācija par veikto darbību. Ja ieraksts satur vairāk nekā divas rindas, tiks parādīta papildu saite *Details*. Noklikšķiniet uz šīs saites, lai skatītu pilnu izmaiņu sarakstu.

Note:
Ja trapper vienums vai HTTP agent vienums (ar iespējotu trapping) ir saņēmis datus, auditācijas žurnālā ieraksts tiks pievienots tikai tad, ja dati tika nosūtīti, izmantojot `history.push` API metodi, nevis utilītu Zabbix sender.

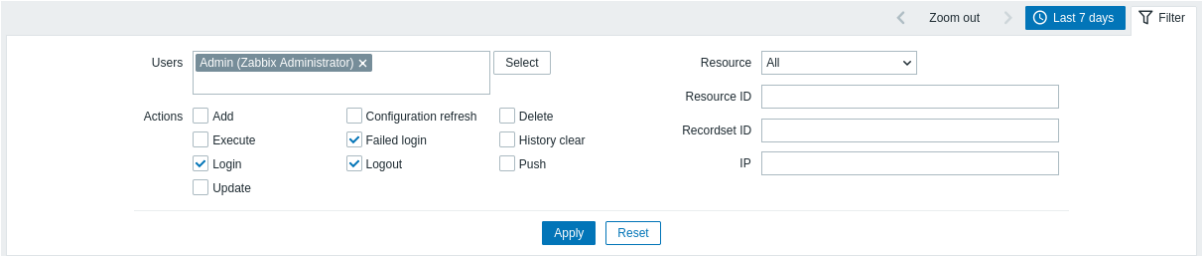
Pogas

Poga lapas augšējā labajā stūrī piedāvā šādu opciju:

Export to CSV	Eksportēt audita žurnāla ierakstus no visām lapām uz CSV failu. Ja ir lietots filtrs, tiks eksportēti tikai filtrētie ieraksti.

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.

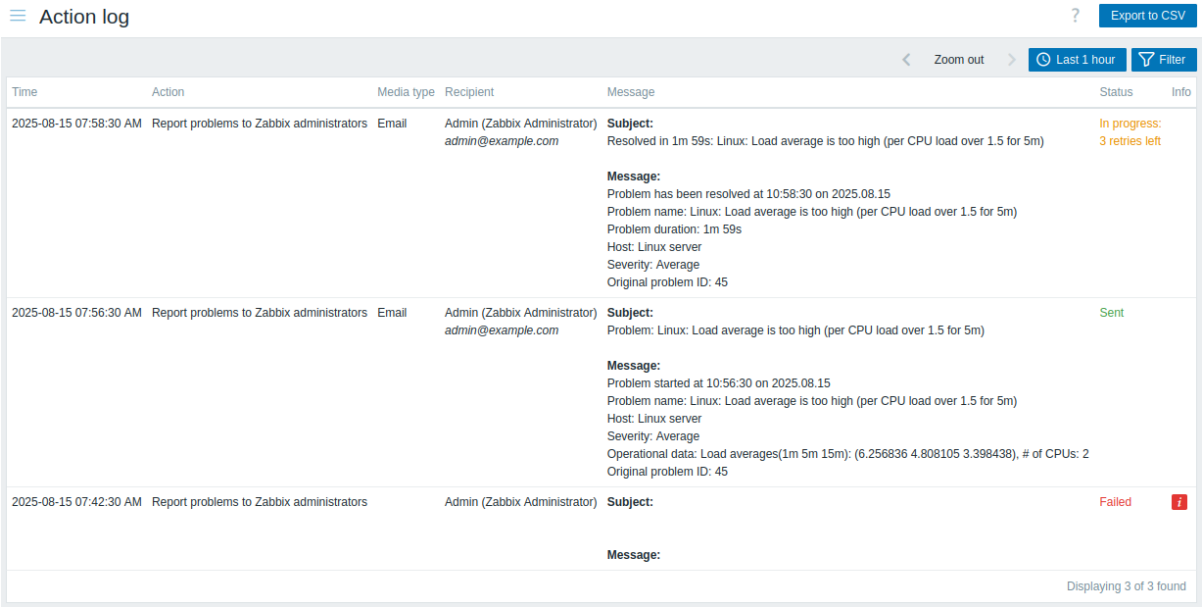
Laika perioda atlasītājs

Laika perioda atlasītājs blakus filtram ļauj ar vienu peles klikšķi atlasīt bieži nepieciešamos periodus. Papildinformāciju skatiet sadaļā Laika perioda un hosts atlasītāji.

Darbību žurnāls

Pārskats

Sadaļā *Reports* > *Action log* varat skatīt operāciju (paziņojumu, attālo komandu) detalizētu informāciju, kas izpildītas darbības ietvaros.



Attēlotie dati:

Column	Description
Time	Operācijas laika zīmogs.
Action	Darbības nosaukums, kas izraisīja operācijas.
Media type	Paziņojuma nosūtīšanai izmantotais medija tips (piem., Email, Jira u.c.). Operācijām, kas izpildīja attālās komandas, šī kolonna būs tukša.
Recipient	Informācija par paziņojuma saņēmēju - lietotājvārds, vārds un uzvārds (iekavās), kā arī papildu informācija atkarībā no medija tipa (e-pasts, lietotājvārds u.c.). Operācijām, kas izpildīja attālās komandas, šī kolonna būs tukša.
Message	Ziņojuma saturs, attālās komandas vai globālā skripta nosaukums. Attālā komanda no mērķa hosta ir atdalīta ar kolu: `<host>:<command>`. Piemēram, ja attālā komanda tika izpildīta Zabbix serverī, informācija būs šādā formātā: `Zabbix server:<command>`.

Column	Description
Status	Operācijas statuss: *In progress* - notiek paziņojuma nosūtīšanas operācija (tiek parādīts arī atlikušais reižu skaits, cik serveris vēl mēģinās nosūtīt paziņojumu) *Sent* - paziņojums ir nosūtīts *Executed* - attālā komanda ir izpildīta *Failed* - operācija nav pabeigta
Info	Kļūdas informācija (ja tāda ir) par operācijas izpildi.

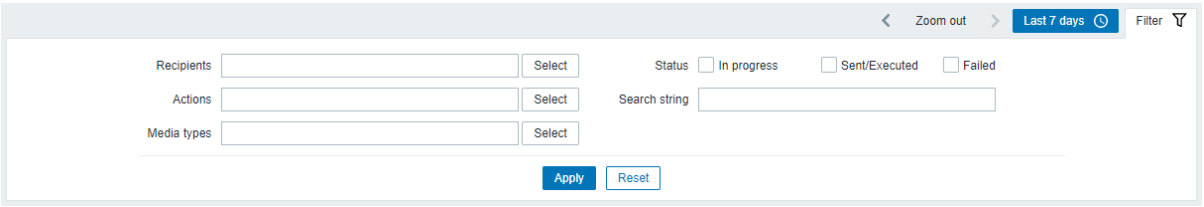
Pogas

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 joslas *Action log*. To var atvērt un sakļaut, noklikšķinot uz cilnes *Filter* lapas augšējā labajā stūrī.



Filtru var izmantot, lai sašaurinātu ierakstus pēc paziņojumu saņēmējiem, darbībām, multivides tipiem, statusa vai pēc ziņojuma/attālā komandas satura (*Search string*). Lai uzlabotu meklēšanas veiktspēju, dati tiek meklēti ar neatrisinātiem makro.

Laika perioda atlasītājs

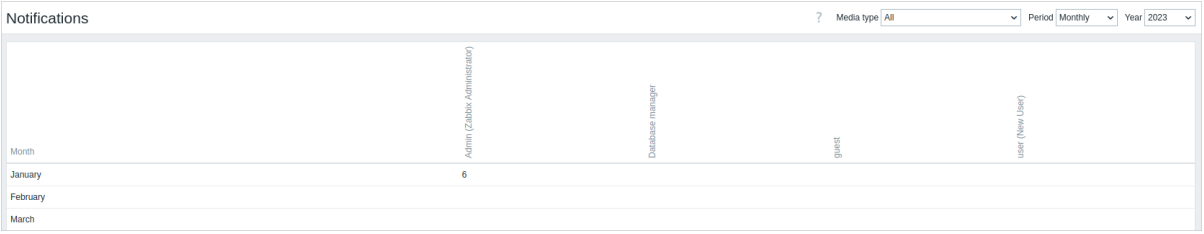
Laika perioda atlasītājs blakus filtram ļauj ar vienu peles klikšķi atlasīt bieži nepieciešamos periodus. Plašāku informāciju skatiet sadaļā Laika perioda un hosts atlasītāji.

Paziņojumi

Pārskats

Sadaļā *Reports* > *Notifications* tiek parādīts pārskats par katram lietotājam nosūtīto paziņojumu skaitu.

No nolaižamajiem sarakstiem augšējā labajā stūrī varat izvēlēties medija tipu (vai visus), periodu (dati par katru dienu/nedēļu/mēnesi/gadu) un gadu nosūtītajiem paziņojumiem.



Katrā kolonnā tiek parādīti kopējie dati vienam sistēmas lietotājam.

Datu vākšana

Pārskats

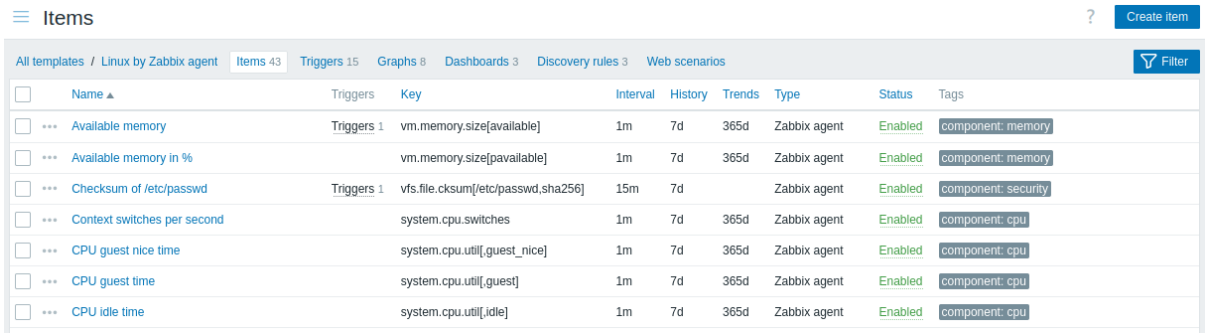
Šajā izvēlnē ir sadaļas, kas saistītas ar datu vākšanas konfigurēšanu.

Vienumi

Pārskats

Veidnes vienumu sarakstam var piekļūt, izvēlnē *Datu apkopošana* > *Veidnes* attiecīgajai veidnei noklikšķinot uz *Vienumi*.

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 parādīta tikai tad, ja filtrā ir atlasītas vairākas veidnes vai neviena veidne.
Nosaukums	Vienuma nosaukums tiek parādīts kā zila saite uz vienuma detalizēto informāciju. Noklikšķinot uz vienuma nosaukuma saites, tiek atvērta vienuma konfigurācijas forma. Ja vienums ir mantots no citas veidnes, pirms vienuma nosaukuma kā pelēka saite tiek parādīts veidnes nosaukums. Noklikšķinot uz veidnes saites, tiek atvērts vienumu saraksts šīs veidnes līmenī.
Trigeri	Novietojot peles rādītāju virs sadaļas Trigeri, tiek parādīts informācijas lodziņš ar vienumam piesaistītajiem trigeriem. Trigeru skaits tiek parādīts pelēkā krāsā.
Atslēga	Tiek parādīta vienuma atslēga.
Intervāls	Tiek parādīts pārbaužu veikšanas biežums.
Vēsture	Tiek parādīts, cik dienas tiks saglabāta vienuma datu vēsture.
Tendences	Tiek parādīts, cik dienas tiks saglabāta vienuma tendenču vēsture.
Tips	Tiek parādīts vienuma tips (Zabbix aģents, SNMP aģents, vienkārša pārbaude u. c.).
Statuss	Tiek parādīts vienuma statuss — *Iespējots* vai *Atspējots*. Noklikšķinot uz statusa, to var mainīt — no Iespējots uz Atspējots (un otrādi).
Birkas	Tiek parādītas vienuma birkas. Var tikt parādītas ne vairāk kā trīs birkas (nosaukuma:vērtības pāri). Ja birku ir vairāk, tiek parādīta saite "...", kas, novietojot peles rādītāju virs tās, ļauj skatīt visas birkas.

Lai konfigurētu jaunu vienumu, augšējā labajā stūrī noklikšķiniet uz pogas *Izveidot vienumu*.

Masveida rediģēšanas opcijas

Pogas zem saraksta piedāvā vairākas masveida rediģēšanas opcijas:

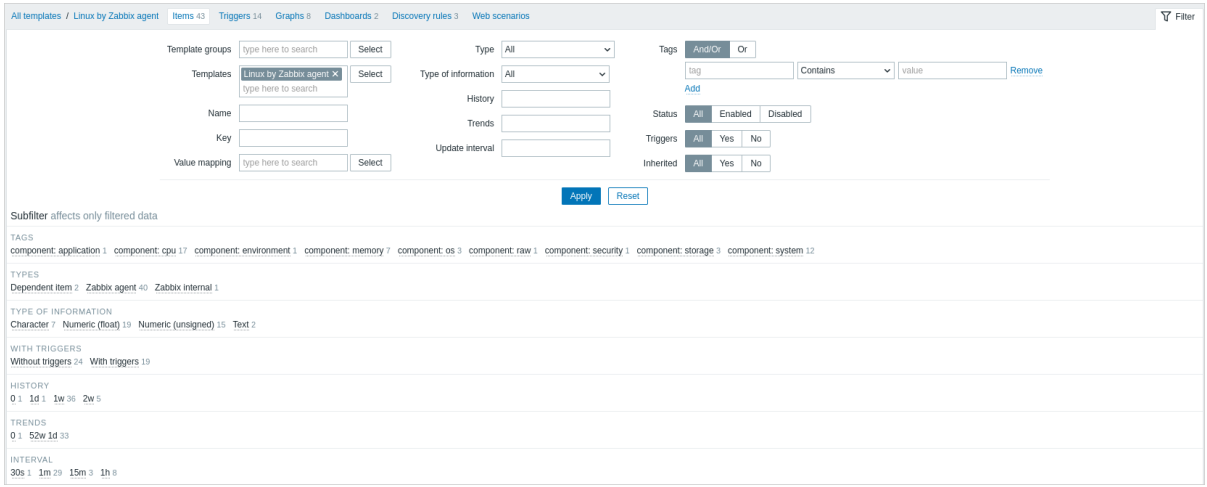
- *Iespējot* - mainīt vienuma statusu uz *Iespējots*.
- *Atspējot* - mainīt vienuma statusu uz *Atspējots*.
- *Kopēt* - kopēt vienumus uz citiem hosts vai veidnēm.
- *Masveida atjaunināšana* - atjaunināt vairākas īpašības vairākiem vienumiem vienlaikus.
- *Dzēst* - dzēst vienumus.

Lai izmantotu šīs opcijas, atzīmējiet izvēles rūtiņas pie attiecīgajiem vienumiem, pēc tam noklikšķiniet uz vajadzīgās pogas.

Filtra izmantošana

Vienumu sarakstā var būt daudz vienumu. Izmantojot filtru, varat atlasīt daļu no tiem, lai ātri atrastu vienumus, ko meklējat. Lai uzlabotu meklēšanas veiktspēju, dati tiek meklēti ar neatrisinātiem makro.

Ikona *Filter* ir pieejama augšējā labajā stūrī. Noklikšķinot uz tās, tiks atvērts filtrs, kurā var norādīt vēlamos filtrēšanas kritērijus.

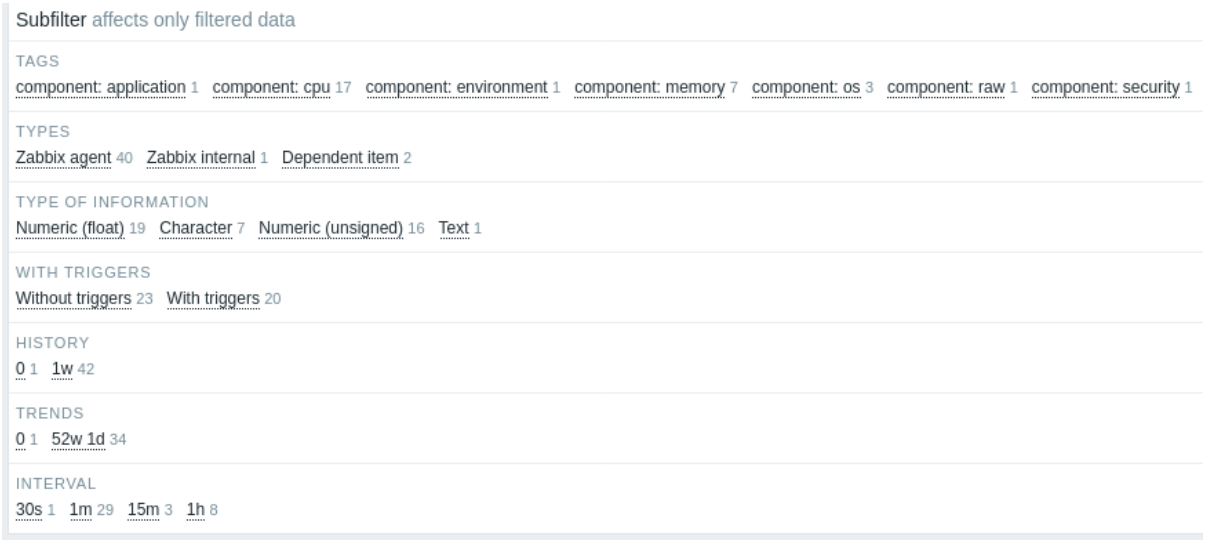


Parameter	Description
Template groups	Filtrēt pēc vienas vai vairākām veidņu grupām. Norādot vecāko veidņu grupu, netieši tiek atlasītas arī visas ligzdotās grupas.
Templates	Filtrēt pēc vienas vai vairākām veidnēm.
Name	Filtrēt pēc vienuma nosaukuma.
Key	Filtrēt pēc vienuma atslēgas.
Value mapping	Filtrēt pēc izmantotās vērtību kartes. Šis parametrs netiek rādīts, ja opcija *Templates* ir tukša.
Type	Filtrēt pēc vienuma tipa (Zabbix aģents, SNMP aģents utt.).
Type of information	Filtrēt pēc informācijas tipa (neparakstīts skaitlis, peldošais punkts utt.).
History	Filtrēt pēc tā, cik ilgi tiek glabāta vienuma vēsture.
Trends	Filtrēt pēc tā, cik ilgi tiek glabātas vienuma tendences.
Update interval	Filtrēt pēc vienuma atjaunināšanas intervāla.
Tags	Norādiet tagus, lai ierobežotu parādīto vienumu skaitu. Ir iespējams iekļaut, kā arī izslēgt konkrētus tagus un tagu vērtības. Var iestatīt vairākus nosacījumus. Tagu nosaukumu atbilstība vienmēr ir reģistrjutīga. Katrai nosacījuma iespējamas vairākas darbības: **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, nav reģistrjutīga) **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, nav reģistrjutīga) 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
Status	Filtrēt pēc vienuma statusa - *Enabled* vai *Disabled*.
Triggers	Filtrēt vienumus ar trigeriem vai bez tiem.
Inherited	Filtrēt mantotos vienumus, kas mantoti vai nav mantoti no saistītajām veidnēm.

Apakšfiltra izmantošana

Apakšfiltrs ļauj papildus modificēt galvenā filtra lietoto filtrēšanu.

Tas satur noklikšķināmas saites ātrai piekļuvei saistītajiem vienumiem. Vienumus saista kopīga entītija — tags, vienuma tips, vienuma atjaunināšanas intervāls u. c. Noklikšķinot uz entītijas, tā tiek izcelta ar pelēku fonu, un vienumi tiek nekavējoties filtrēti (galvenajā filtrā nav jānoklikšķina uz *Lietot*). Noklikšķinot uz citas entītijas, tā tiek pievienota filtrētajiem rezultātiem. Vēlreiz noklikšķinot uz entītijas, filtrēšana pēc tās tiek noņemta.



Apakšfiltri tiek ģenerēti, pamatojoties uz filtrētajiem datiem, kas ir ierobežoti līdz 1000 ierakstiem. Ja apakšfiltrā vēlaties redzēt vairāk ierakstu, jāpalielina parametra *Meklēšanas un filtra rezultātu ierobežojums* vērtība (sadaļā *Administrēšana* > *Vispārīgi* > *Lietotāja saskarne*).

Atšķirībā no galvenā filtra apakšfiltrs tiek atjaunināts ar katru tabulas atsvaidzināšanas pieprasījumu, lai tajā vienmēr būtu aktuāla informācija par pieejamajām filtrēšanas opcijām un to skaitu.

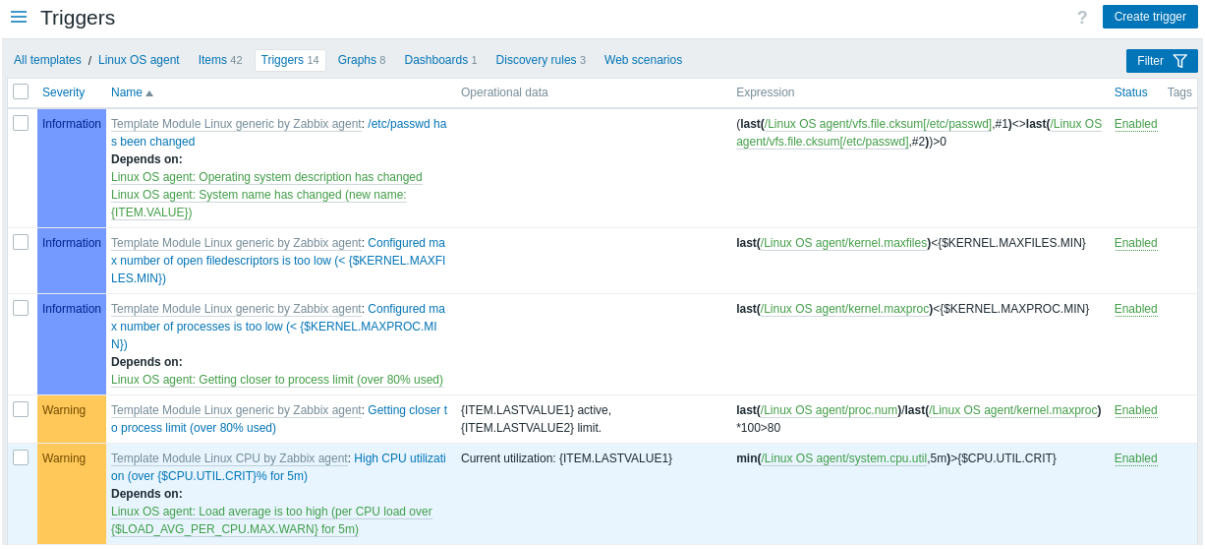
Horizontāli attēloto entītiju skaits ir ierobežots līdz 100. Ja entītiju ir vairāk, beigās tiek attēlota trīs punktu ikona; uz tās nevar noklikšķināt.

Skaitlis blakus katrai noklikšķināmajai entītijai norāda tajā sagrupēto vienumu skaitu (pamatojoties uz galvenā filtra rezultātiem). Kad entītija ir atlasīta, pie citām pieejamajām entītijām tiek attēloti skaitļi ar pluszīmi, kas norāda, cik vienumu var pievienot pašreizējai atlasei. Entītijas, kurās nav vienumu, netiek attēlotas, ja vien tās iepriekš nav atlasītas apakšfiltrā.

Trigeri

Pārskats

Veidnes trigeru sarakstam var piekļūt sadaļā *Datu apkopošana* > *Veidnes*, attiecīgajai veidnei noklikšķinot uz *Trigeri*.



Parādītie dati:

Kolonna	Apraksts
Būtiskums	Trigeru būtiskums tiek parādīts gan ar nosaukumu, gan šūnas fona krāsu.
Veidne	Veidne, kurai pieder trigeris. Noklikšķinot uz veidnes nosaukuma, tiek atvērta veidnes konfigurācijas forma. Šī kolonna tiek parādīta tikai tad, ja filtrā ir atlasītas vairākas veidnes vai nav atlasīta neviena veidne.

Kolonna	Apraksts
Nosaukums	Trigeru nosaukums tiek parādīts kā zila saite uz trigeru detalizētu informāciju. Noklikšķinot uz trigeru nosaukuma saites, tiek atvērta trigera konfigurācijas forma. Ja trigeris ir mantots no citas veidnes, pirms trigera nosaukuma kā pelēka saite tiek parādīts veidnes nosaukums. Noklikšķinot uz veidnes saites, tiek atvērts trigeru saraksts šīs veidnes līmenī.
Darbības dati	Trigera darbības datu definīcija, kas satur patvaļīgas virknes un makro, kuri dinamiski tiks aizstāti sadaļā *Uzraudzība* > *Problēmas*.
Izteiksme	Tiek parādīta trigera izteiksme. Izteiksmes veidnes-vienuma daļa tiek parādīta kā saite, kas ved uz vienuma konfigurācijas formu.
Statuss	Tiek parādīts trigera statuss - *Iespējots* vai *Atspējots*. Noklikšķinot uz statusa, to var mainīt - no Iespējots uz Atspējots (un otrādi).
Tagi	Ja trigeris satur tagus, šajā kolonnā tiek parādīts taga nosaukums un vērtība.

Lai konfigurētu jaunu trigeri, augšējā labajā stūrī noklikšķiniet uz pogas *Izveidot trigeri*.

Masveida rediģēšanas opcijas

Pogas zem saraksta piedāvā vairākas masveida rediģēšanas opcijas:

- *Iespējot* - mainīt trigeru statusu uz *Iespējots*
- *Atspējot* - mainīt trigeru statusu uz *Atspējots*
- *Kopēt* - kopēt trigerus uz citiem hostiem vai veidnēm
- *Masveida atjaunināšana* - atjaunināt vairākas īpašības vairākiem trigeriem vienlaikus
- *Dzēst* - dzēst trigerus

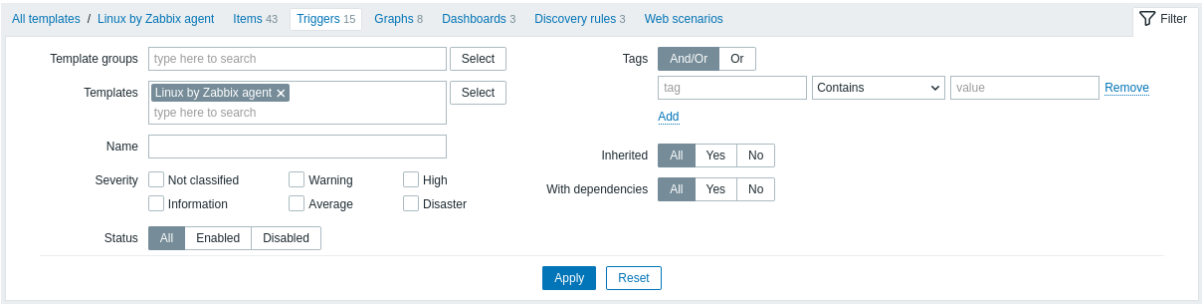
Lai izmantotu šīs opcijas, atzīmējiet izvēles rūtiņas pie attiecīgajiem trigeriem, pēc tam noklikšķiniet uz vajadzīgās pogas.

Filtra izmantošana

Varat izmantot filtru, lai parādītu tikai tos trigerus, kas jūs interesē.

Lai nodrošinātu labāku meklēšanas veiktspēju, dati tiek meklēti ar neatrisinātiem makro.

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 vecāko veidņu grupu, automātiski tiek atlasītas visas ligzdotās grupas.
Veidnes	Filtrēt pēc vienas vai vairākām veidnēm. Ja iepriekš ir atlasītas veidņu grupas, veidņu atlase ir ierobežota līdz šīm grupām.
Nosaukums	Filtrēt pēc trigeru nosaukuma.
Smaguma pakāpe	Atlasiet, lai filtrētu pēc vienas vai vairākām trigeru smaguma pakāpēm.
Statuss	Filtrēt pēc trigeru aktivizācijas statusa (Ieslēgts/Izslēgts).

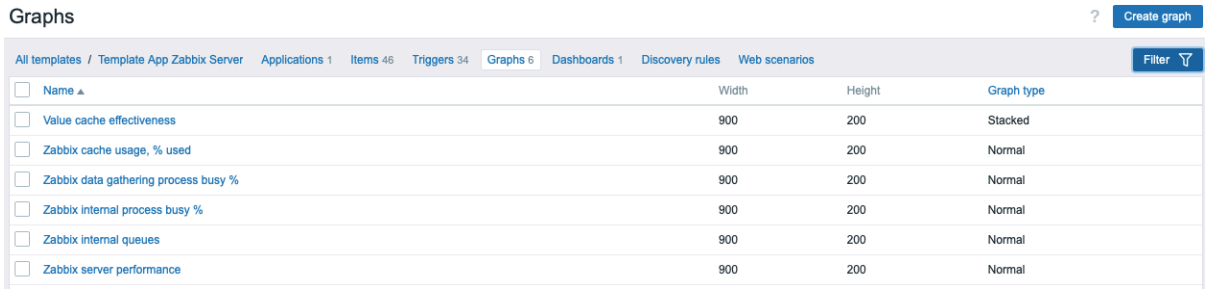
Parametrs	Apraksts
Tagi	Filtrēt pēc trigeru taga nosaukuma un vērtības. Ir iespējams iekļaut, kā arī izslēgt konkrētus tagus un tagu vērtības. Var iestatīt vairākus nosacījumus. Taga nosaukuma atbilstība vienmēr ir reģistrjutīga. Katram nosacījumam ir pieejami vairāki operatori: **Pastāv** - iekļaut norādītos taga nosaukumus **Vienāds ar** - iekļaut norādītos taga nosaukumus un vērtības (reģistrjutīgi) **Satur** - iekļaut norādītos taga nosaukumus, kuru taga vērtības satur ievadīto virkni (apakšvirknes atbilstība, nereģistrjutīgi) **Nepastāv** - izslēgt norādītos taga nosaukumus **Nav vienāds ar** - izslēgt norādītos taga nosaukumus un vērtības (reģistrjutīgi) **Nesatur** - izslēgt norādītos taga nosaukumus, kuru taga vērtības satur ievadīto virkni (apakšvirknes atbilstība, nereģistrjutīgi) Nosacījumiem ir pieejami divi aprēķina tipi: **Un/Vai** - jāizpildās visiem nosacījumiem, nosacījumi ar vienādu taga nosaukumu tiks grupēti pēc Vai nosacījuma **Vai** - pietiek, ja izpildās viens nosacījums Taga nosaukuma un taga vērtības laukos tiek atbalstīti makro un makro funkcijas.
Mantots	Filtrēt trigerus, kas mantoti (vai nav mantoti) no saistītajām veidnēm.
Ar atkarībām	Filtrēt trigerus ar atkarībām (vai bez tām).

Grafiki

Pārskats

Veidnes pielāgoto grafiku sarakstam var piekļūt sadaļā *Datu apkopošana* > *Veidnes*, attiecīgajai veidnei noklikšķinot uz *Grafiki*.

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 parādīta tikai tad, ja filtrā ir atlasītas vairākas veidnes vai 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 kā pelēka saite tiek parādīts veidnes nosaukums. 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 *Izkliedēts*.

Lai konfigurētu jaunu grafiku, augšējā labajā stūrī noklikšķiniet uz pogas *Izveidot grafiku*.

Masveida rediģēšanas opcijas

Pogas zem saraksta piedāvā dažas masveida rediģēšanas opcijas:

- *Kopēt* - kopēt grafikus uz citiem hosts vai veidnēm
- *Dzēst* - dzēst grafikus

Lai izmantotu šīs opcijas, atzīmējiet izvēles rūtiņas pie attiecīgajiem grafikiem, pēc tam noklikšķiniet uz vajadzīgās pogas.

Filtra izmantošana

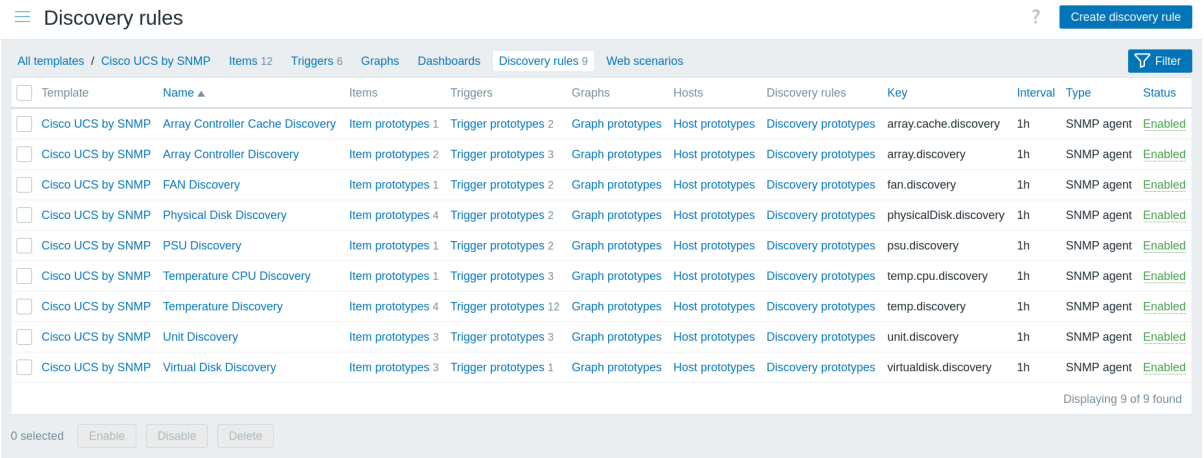
Jūs varat filtrēt grafikus pēc veidņu grupas un veidnes. Lai uzlabotu meklēšanas veiktspēju, dati tiek meklēti ar neatrisinātām makrokomandām.

Atklāšanas noteikumi

Pārskats

Veidnes zema līmeņa atklāšanas noteikumu sarakstam var piekļūt, atverot *Data collection* > *Templates* un attiecīgajai veidnei noklikšķinot uz *Discovery*.

Tiek parādīts esošo zema līmeņa atklāšanas noteikumu saraksts. Ir iespējams skatīt arī visus atklāšanas noteikumus neatkarīgi no veidnes vai visus konkrētas veidņu grupas atklāšanas noteikumus, mainot filtra iestatījumus.



Parādītie dati:

Kolonna	Apraksts
Template	Veidne, kurai pieder atklāšanas noteikums. Noklikšķinot uz veidnes nosaukuma, tiek atvērta veidnes konfigurācijas forma.
Name	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 kā pelēka saite tiek parādīts veidnes nosaukums. Noklikšķinot uz veidnes saites, tiek atvērts atklāšanas noteikumu saraksts šīs 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 veikšanas biežums.
Type	Tiek parādīts atklāšanai izmantotais vienuma tips (Zabbix aģents, SNMP aģents u. c.).
Status	Tiek parādīts atklāšanas noteikuma statuss - *Enabled* vai *Disabled*. Noklikšķinot uz statusa, to var mainīt - no Enabled uz Disabled (un otrādi).

Lai konfigurētu jaunu zema līmeņa atklāšanas noteikumu, augšējā labajā stūrī noklikšķiniet uz pogas *Create discovery rule*.

Masveida rediģēšanas opcijas

Pogas zem saraksta piedāvā dažas masveida rediģēšanas opcijas:

- *Iespējot* - mainīt zemā līmeņa atklāšanas noteikuma statusu uz *Iespējots*

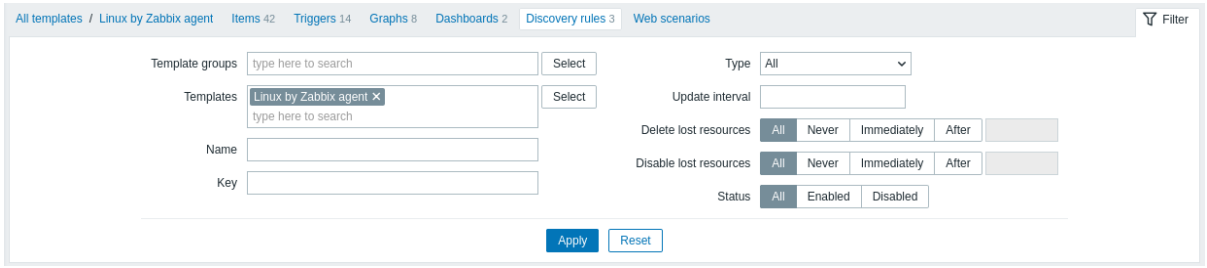
- *Atspējot* - mainīt zemā līmeņa atklāšanas noteikuma statusu uz *Atspējots*
- *Dzēst* - dzēst zemā līmeņa atklāšanas noteikumus

Lai izmantotu šīs opcijas, atzīmējiet izvēles rūtiņas pie attiecīgajiem atklāšanas noteikumiem, pēc tam noklikšķiniet uz vajadzīgās pogas.

Filtra izmantošana

Varat izmantot filtru, lai parādītu tikai tās atklāšanas noteikumus, kas jūs interesē. Lai uzlabotu meklēšanas veiktspēju, dati tiek meklēti ar neatrisinātiem makro.

Augšējā labajā stūrī ir pieejama ikona *Filtrs*. Noklikšķinot uz tās, tiks atvērts filtrs, kurā varat norādīt vēlamos filtrēšanas kritērijus, piemēram, veidni, atklāšanas noteikuma nosaukumu, vienuma atslēgu, vienuma tipu u. c.



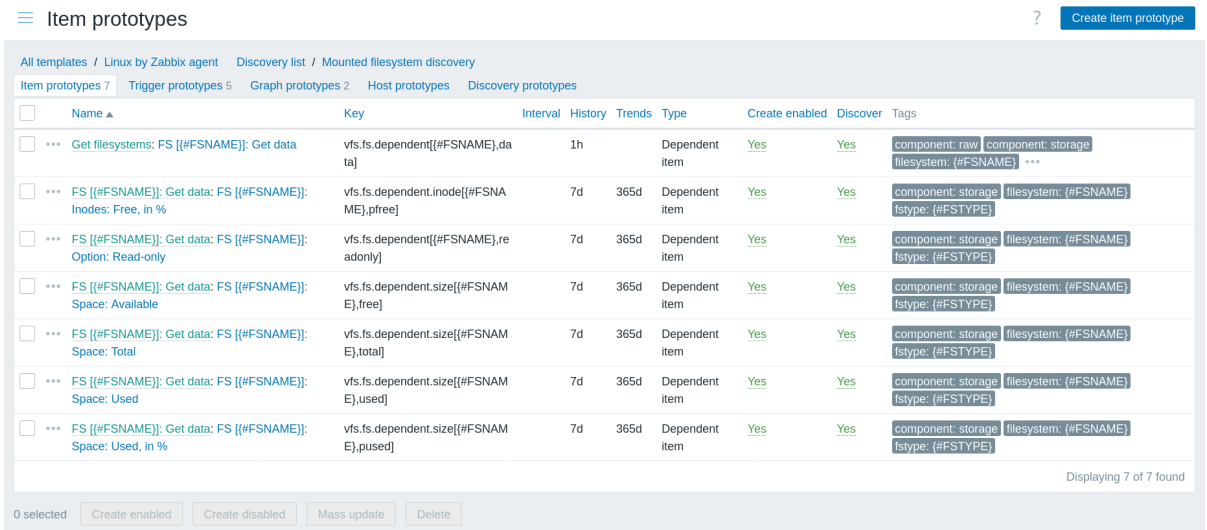
Parameter	Description
Template groups	Filtrēt pēc vienas vai vairākām veidņu grupām. Norādot vecāko veidņu grupu, automātiski tiek atlasītas arī visas ligzdotās grupas.
Templates	Filtrēt pēc vienas vai vairākām veidnēm.
Name	Filtrēt pēc atklāšanas noteikuma nosaukuma.
Key	Filtrēt pēc atklāšanas vienuma atslēgas.
Type	Filtrēt pēc atklāšanas vienuma tipa.
Update interval	Filtrēt pēc atjaunināšanas intervāla. Nav pieejams Zabbix trapper un atkarīgajiem vienumiem.
Delete lost resources	Filtrēt pēc *Delete lost resources* perioda.
Disable lost resources	Filtrēt pēc *Disable lost resources* perioda.
Status	Filtrēt pēc atklāšanas noteikuma statusa (All/Enabled/Disabled).

Vienumu prototipi

Pārskats

Šajā sadaļā tiek parādīti veidnes zema līmeņa atklāšanas noteikuma konfigurētie vienumu prototipi.

Ja veidne ir piesaistīta hostam, vienumu prototipi zema līmeņa atklāšanas laikā kļūs par pamatu reālu hosta vienumu izveidei.



Parādītie dati:

Column	Description
Name	Vienuma prototipa nosaukums tiek parādīts kā zila saite. Noklikšķinot uz nosaukuma, tiek atvērta vienuma prototipa konfigurācijas veidlapa. Ja vienuma prototips pieder piesaistītai veidnei, pirms vienuma prototipa nosaukuma kā pelēka saite tiek parādīts veidnes nosaukums. Noklikšķinot uz veidnes saites, tiek atvērts vienuma prototipu saraksts piesaistītās veidnes līmenī. Ja vienuma prototips ir tipa ”Dependent item”, pirms vienuma prototipa nosaukuma zaļā krāsā tiek parādīts galvenā vienuma nosaukums (tāpat kā vienuma prototipiem hosta līmenī).
Key	Tiek parādīta vienuma prototipa atslēga.
Interval	Tiek parādīts pārbaudes biežums.
History	Tiek parādīts dienu skaits, cik ilgi saglabāt vienuma datu vēsturi.
Trends	Tiek parādīts dienu skaits, cik ilgi saglabāt vienuma tendenču vēsturi.
Type	Tiek parādīts vienuma prototipa tips (Zabbix aģents, SNMP aģents, vienkārša pārbaude u. c.).
Create enabled	Izveidot vienumu, pamatojoties uz šo prototipu: **Yes** - iespējots **No** - atspējots. Varat pārslēgties starp vērtībām ”Yes” un ”No”, noklikšķinot uz tām.
Discover	Atklāt vienumu, pamatojoties uz šo prototipu: **Yes** - atklāt **No** - neatklāt. Varat pārslēgties starp vērtībām ”Yes” un ”No”, noklikšķinot uz tām.
Tags	Tiek parādītas vienuma prototipa birkas.

Lai konfigurētu jaunu vienuma prototipu, augšējā labajā stūrī noklikšķiniet uz pogas *Create item prototype*.

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

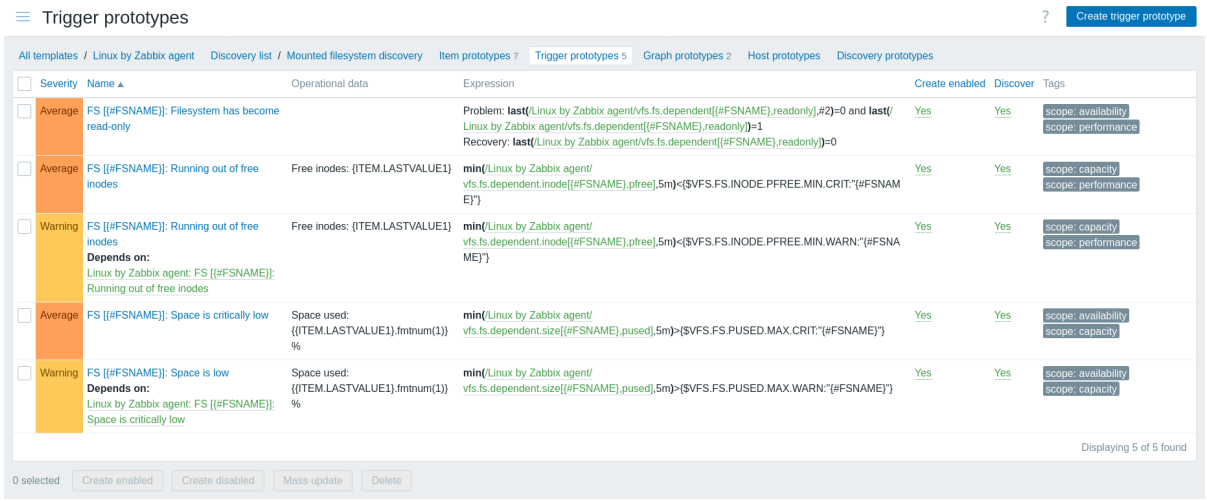
Lai izmantotu šīs opcijas, atzīmējiet izvēles rūtiņas attiecīgo vienumu prototipu priekšā, pēc tam noklikšķiniet uz vajadzīgās pogas.

Trigeru prototipi

Pārskats

Šajā sadaļā tiek parādīti veidnes zema līmeņa atklāšanas kārtulas konfigurētie trigeru prototipi.

Ja veidne ir saistīta ar hostu, trigeru prototipi zema līmeņa atklāšanas laikā kļūs par pamatu reālu hostu trigeru izveidei.



Parādītie dati:

Kolonna	Apraksts
Nopietnība	Trigeru nopietnība tiek parādīta gan ar nosaukumu, gan šūnas fona krāsu.

Kolonna	Apraksts
Nosaukums	Trigeru prototipa nosaukums, kas tiek parādīts kā zila saite. Noklikšķinot uz nosaukuma, tiek atvērta trigeru prototipa konfigurācijas veidlapa. Ja trigeru prototips pieder saistītai veidnei, pirms trigera nosaukuma kā pelēka saite tiek parādīts veidnes nosaukums. Noklikšķinot uz veidnes saites, saistītās veidnes līmenī tiek atvērts trigeru prototipu saraksts.
Izteiksme	Tiek parādīta trigera izteiksme. Izteiksmes veidnes vienuma daļa tiek parādīta kā saite, kas ved uz vienuma konfigurācijas veidlapu.
Darbības dati	Tiek parādīts trigera darbības datu formāts, kas satur patvaļīgas virknes un makro, kuri dinamiski tiks aizstāti sadaļā *Monitoring* > *Problems*.
Izveide iespējota	Trigera izveide, pamatojoties uz šo prototipu: **Jā** - iespējota **Nē** - atspējota. Varat pārslēgties starp vērtībām "Jā" un "Nē", noklikšķinot uz tām.
Atklāt	Trigera atklāšana, pamatojoties uz šo prototipu: **Jā** - atklāt **Nē** - neatklāt. Varat pārslēgties starp vērtībām "Jā" un "Nē", noklikšķinot uz tām.
Tagi	Tiek parādīti trigeru prototipa tagi.

Lai konfigurētu jaunu trigeru prototipu, augšējā labajā stūrī noklikšķiniet uz pogas *Izveidot trigeru prototipu*.

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

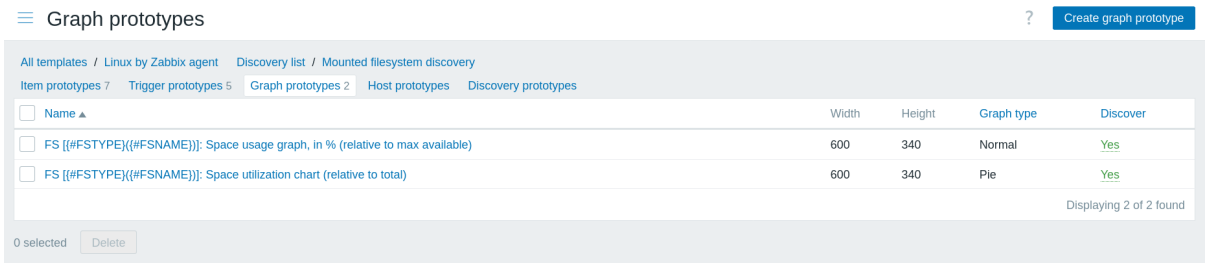
Lai izmantotu šīs opcijas, atzīmējiet izvēles rūtiņas attiecīgo trigeru prototipu priekšā, pēc tam noklikšķiniet uz vajadzīgās pogas.

Grafiku prototipi

Pārskats

Šajā sadaļā tiek parādīti veidnes zema līmeņa atklāšanas noteikuma konfigurētie grafiku prototipi.

Ja veidne ir piesaistīta hostam, grafiku prototipi zema līmeņa atklāšanas laikā kļūs par pamatu reālu hosta grafiku izveidei.



Parādītie dati:

Kolonna	Apraksts
Nosaukums	Grafika prototipa nosaukums, kas tiek parādīts kā zila saite. Noklikšķinot uz nosaukuma, tiek atvērta grafika prototipa konfigurācijas veidlapa. Ja grafika prototips pieder piesaistītai veidnei, pirms grafika nosaukuma kā pelēka saite tiek parādīts veidnes nosaukums. Noklikšķinot uz veidnes saites, piesaistītās veidnes līmenī tiek atvērts grafiku prototipu saraksts.
Platums	Tiek parādīts grafika prototipa platums.
Augstums	Tiek parādīts grafika prototipa augstums.
Tips	Tiek parādīts grafika prototipa tips - *Parasts*, *Sakrauts*, *Sektoru* vai *Izkliedēts*.
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, augšējā labajā stūrī noklikšķiniet uz pogas *Izveidot grafika prototipu*.

Masveida rediģēšanas opcijas

Pogas zem saraksta piedāvā dažas masveida rediģēšanas opcijas:

- *Dzēst* - dzēst šos grafiku prototipus

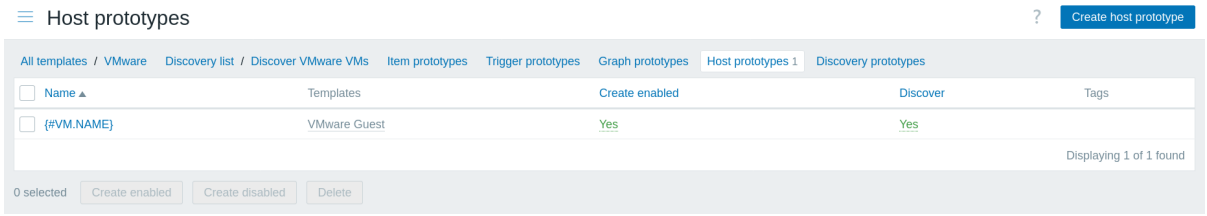
Lai izmantotu šīs opcijas, atzīmējiet izvēles rūtiņas pie attiecīgajiem grafiku prototipiem, pēc tam noklikšķiniet uz vajadzīgās pogas.

Hostu prototipi

Pārskats

Šajā sadaļā tiek parādīti veidnes zema līmeņa atklāšanas kārtulas konfigurētie hostu prototipi.

Ja veidne ir piesaistīta hostam, hostu prototipi būs pamats reālu hostu izveidei zema līmeņa atklāšanas laikā.



Parādītie dati:

Kolonna	Apraksts
Nosaukums	Hosta prototipa nosaukums, kas tiek parādīts kā zila saite. Noklikšķinot uz nosaukuma, tiek atvērta hosta prototipa konfigurācijas veidlapa. Ja hosta prototips pieder piesaistītai veidnei, pirms hosta nosaukuma kā pelēka saite tiek parādīts veidnes nosaukums. Noklikšķinot uz veidnes saites, piesaistītās veidnes līmenī tiek atvērts hostu prototipu saraksts.
Veidnes	Tiek parādītas hosta prototipa veidnes.
Izveide iespējota	Izveidot hostu, pamatojoties uz šo prototipu: **Jā** - iespējots **Nē** - atspējots. Varat pārslēgties starp vērtībām “Jā” un “Nē”, noklikšķinot uz tām.
Atklāt	Atklāt hostu, pamatojoties uz šo prototipu: **Jā** - atklāt **Nē** - neatklāt. Varat pārslēgties starp vērtībām “Jā” un “Nē”, noklikšķinot uz tām.
Tagi	Tiek parādīti hosta prototipa tagi.

Lai konfigurētu jaunu hosta prototipu, augšējā labajā stūrī noklikšķiniet uz pogas *Izveidot hosta prototipu*.

Masveida rediģēšanas opcijas

Pogas zem saraksta piedāvā dažas masveida rediģēšanas opcijas:

- *Izveidot iespējotu* - izveidot šos hosts kā *Iespējotus*
- *Izveidot atspējotu* - izveidot šos hosts kā *Atspējotus*
- *Dzēst* - dzēst šos hostu prototipus

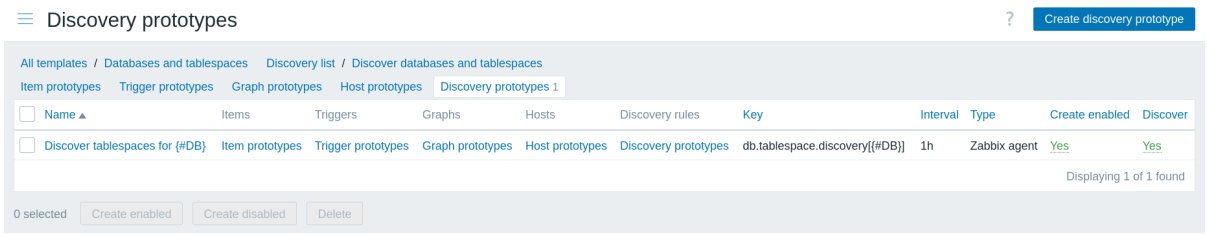
Lai izmantotu šīs opcijas, atzīmējiet izvēles rūtiņas pie attiecīgajiem hostu prototipiem, pēc tam noklikšķiniet uz vajadzīgās pogas.

Atklāšanas prototipi

Pārskats

Šajā sadaļā tiek parādīti veidnē konfigurētie zema līmeņa atklāšanas kārtulas atklāšanas prototipi.

Ja veidne ir piesaistīta hostam, atklāšanas prototipi kļūs par pamatu vecākās atklāšanas kārtulas reālo atklāšanas prototipu izveidei.



Parādītie dati:

Kolonna	Apraksts
Nosaukums	Atklāšanas prototipa nosaukums tiek parādīts kā zila saite. Noklikšķinot uz nosaukuma, tiek atvērta atklāšanas prototipa konfigurācijas veidlapa. Ja atklāšanas prototips pieder piesaistītai veidnei, pirms atklāšanas prototipa nosaukuma kā pelēka saite tiek parādīts veidnes nosaukums. Noklikšķinot uz veidnes saites, piesaistītās veidnes līmenī tiek atvērts atklāšanas prototipu saraksts.
Vienumi	Tiek parādīta saite uz vienumu prototipu sarakstu. Pelēkā krāsā tiek parādīts esošo vienumu prototipu skaits.
Trigeri	Tiek parādīta saite uz trigeru prototipu sarakstu. Pelēkā krāsā tiek parādīts esošo trigeru prototipu skaits.
Grafiki	Tiek parādīta saite uz grafiku prototipu sarakstu. Pelēkā krāsā tiek parādīts esošo grafiku prototipu skaits.
Hosti	Tiek parādīta saite uz hostu prototipu sarakstu. Pelēkā krāsā tiek parādīts esošo hostu prototipu skaits.
Atklāšanas kārtulas	Tiek parādīta saite uz atklāšanas prototipu sarakstu. Pelēkā krāsā tiek parādīts esošo atklāšanas prototipu skaits.
Atslēga	Tiek parādīta atklāšanai izmantotā vienuma atslēga.
Intervāls	Tiek parādīts atklāšanas veikšanas biežums.
Tips	Tiek parādīts atklāšanai izmantotais vienuma tips (Zabbix aģents, SNMP aģents u. c.).
Izveide iespējota	Tiek parādīts atklāšanas prototipa izveides statuss — izveide iespējota (*Jā*) vai izveide atspējota (*Nē*). Noklikšķinot uz statusa, to var mainīt — no "Jā" uz "Nē" (un otrādi).
Atklāt	Tiek parādīts atklāšanas prototipa atklāšanas statuss — atklāt (*Jā*) vai neatklāt (*Nē*). Noklikšķinot uz statusa, to var mainīt — no "Jā" uz "Nē" (un otrādi).

Lai konfigurētu jaunu atklāšanas prototipu, augšējā labajā stūrī noklikšķiniet uz pogas *Izveidot atklāšanas prototipu*.

Masveida rediģēšanas opcijas

Pogas zem saraksta piedāvā dažas masveida rediģēšanas opcijas:

- *Izveidot iespējotu* - izveidot šos atklāšanas prototipus kā *Iespējotus*
- *Izveidot atspējotu* - izveidot šos atklāšanas prototipus kā *Atspējotus*
- *Dzēst* - dzēst šos atklāšanas prototipus

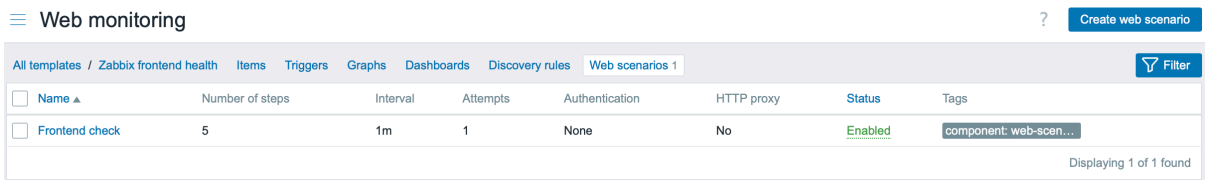
Lai izmantotu šīs opcijas, atzīmējiet izvēles rūtiņas pie attiecīgajiem atklāšanas prototipiem, pēc tam noklikšķiniet uz vajadzīgās pogas.

Tīmekļa scenāriji

Pārskats

Veidnes tīmekļa scenāriju sarakstam var piekļūt sadaļā *Datu apkopošana* > *Veidnes*, attiecīgajai veidnei noklikšķinot uz *Tīmeklis*.

Tiek parādīts esošo tīmekļa scenāriju saraksts.



Parādītie dati:

Kolonna	Apraksts
Nosaukums	Tīmekļa scenārija nosaukums. Noklikšķinot uz tīmekļa scenārija nosaukuma, tiek atvērta tīmekļa scenārija konfigurācijas veidlapa. 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.
Darbību skaits	Scenārijā ietverto darbību skaits.
Atjaunināšanas intervāls	Cik bieži tiek izpildīts scenārijs.
Mēģinājumi	Mēģinājumu skaits tīmekļa scenārija darbību izpildei.
Autentifikācija	Tiek parādīta autentifikācijas metode - Basic, NTLM, Kerberos, Digest vai None.
HTTP starpniekserveris	Tiek parādīts HTTP starpniekserveris vai "Nē", ja tas netiek izmantots.
Statuss	Tiek parādīts tīmekļa scenārija statuss - *Iespējots* vai *Atspējots*. Noklikšķinot uz statusa, to var mainīt.
Tagi	Tiek parādīti tīmekļa scenārija tagi. Var tikt parādīti ne vairāk kā trīs tagi (nosaukuma:vērtības pāri). Ja tagu ir vairāk, tiek parādīta saite "...", kas, novietojot peles rādītāju virs tās, ļauj skatīt visus tagus.

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* - maina scenārija statusu uz *Iespējots*
- *Atspējot* - maina scenārija statusu uz *Atspējots*
- *Dzēst* - dzēš tīmekļa scenārijus

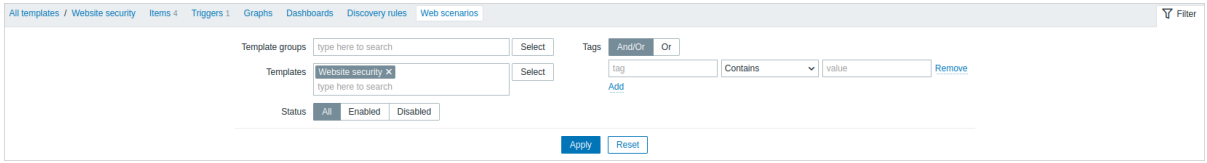
Lai izmantotu šīs opcijas, atzīmējiet izvēles rūtiņas attiecīgo tīmekļa scenāriju priekšā un pēc tam noklikšķiniet uz vajadzīgās pogas.

Filtra izmantošana

Varat izmantot filtru, lai parādītu tikai tos scenārijus, kas jūs interesē.

Lai uzlabotu meklēšanas veiktspēju, dati tiek meklēti ar neatrisinātiem makro.

Saite *Filter* ir pieejama virs tīmekļa scenāriju saraksta. Ja uz tās noklikšķināsiet, būs pieejams filtrs, kurā varat filtrēt scenārijus pēc veidņu grupas, veidnes, statusa un tagiem.

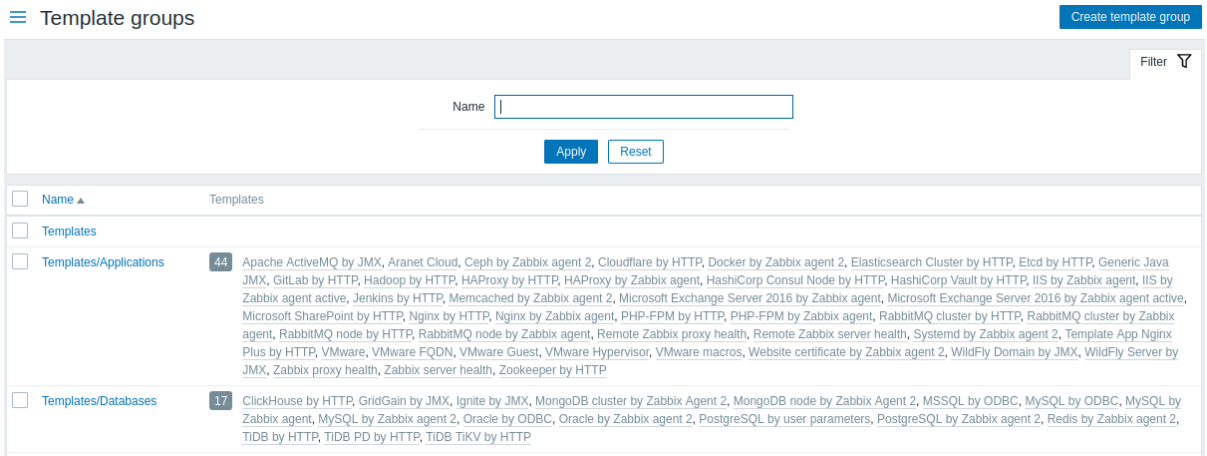


Veidņu grupas

Pārskats

Sadaļā *Datu apkopošana* > *Veidņu grupas* lietotāji var konfigurēt un uzturēt veidņu grupas.

Tiek parādīts esošo veidņu grupu saraksts ar to detalizētu informāciju. Veidņu grupas var meklēt un filtrēt pēc nosaukuma.



Parādītie dati:

Kolonna	Apraksts
Nosaukums	Veidņu grupas nosaukums. Noklikšķinot uz grupas nosaukuma, tiek atvērta veidņu grupas konfigurācijas veidlapa.
Veidnes	Grupā esošo veidņu skaits (parādīts pelēkā krāsā), kam seko grupas dalībnieku saraksts. Noklikšķinot uz veidnes nosaukuma, tiek atvērta veidnes konfigurācijas veidlapa. Noklikšķinot uz skaita, tiek atvērts šajā grupā esošo veidņu saraksts.

Masveida rediģēšanas opcijas

Lai vienlaikus dzēstu vairākas veidņu grupas, atzīmējiet izvēles rūtiņas pie attiecīgajām grupām, pēc tam noklikšķiniet uz pogas Delete zem saraksta.

Filtra izmantošana

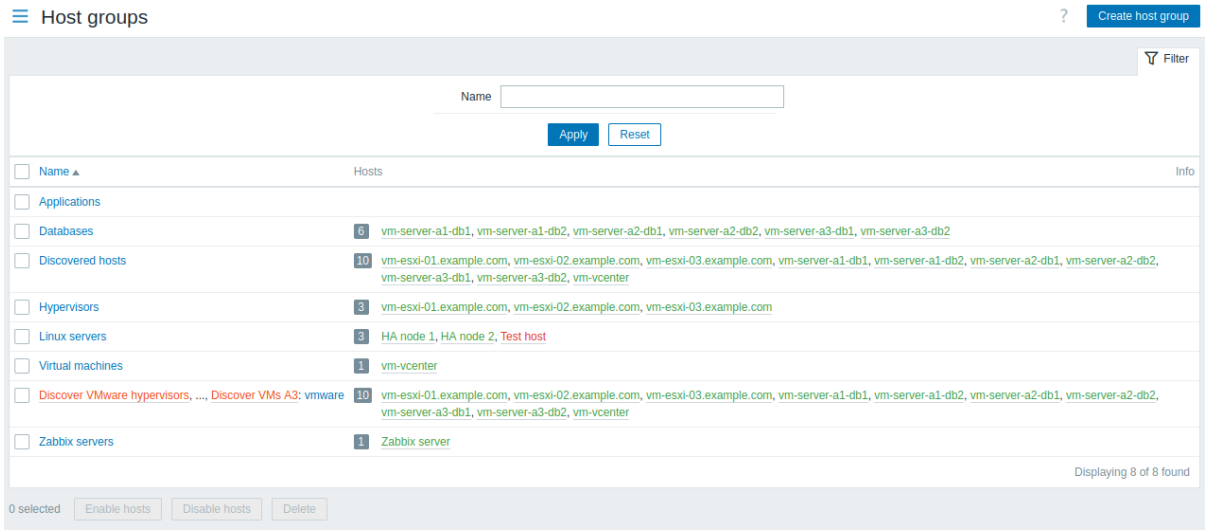
Varat izmantot filtru, lai parādītu tikai tās veidņu grupas, kas jūs interesē. Lai nodrošinātu labāku meklēšanas veiktspēju, dati tiek meklēti ar neatrisinātām makrokomandām.

Hostu grupas

Pārskats

Sadaļā *Datu apkopošana* > *Hostu grupas* lietotāji var konfigurēt un pārvaldīt hostu grupas.

Tiek parādīts esošo hostu grupu saraksts ar to detalizētu informāciju. Hostu grupas var meklēt un filtrēt pēc nosaukuma.



Parādītie dati:

Kolonna	Apraksts
Nosaukums	Hostu grupas nosaukums. Noklikšķinot uz grupas nosaukuma, tiek atvērta hostu grupas konfigurācijas veidlapa. Atklātās hostu grupas tiek parādītas ar zema līmeņa atklāšanas noteikumu nosaukumiem kā prefiksu. Noklikšķinot uz ZLA noteikuma nosaukuma, tiek atvērta hosta prototipa konfigurācijas veidlapa. Ņemiet vērā, ka atklātās hostu grupas tiek dzēstas, kad tiek dzēsti visi ZLA noteikumi, kas tās atklāja.
Hosti	Hostu skaits grupā (parādīts pelēkā krāsā), kam seko grupas dalībnieku saraksts. Noklikšķinot uz hosta nosaukuma, tiek atvērta hosta konfigurācijas veidlapa. Noklikšķinot uz skaita, visā hostu sarakstā tiek parādīti tikai tie hosti, kas pieder grupai.
Informācija	Tiek parādīta informācija par hostu grupas kļūdām (ja tādas ir).

Masveida rediģēšanas opcijas

Pogas zem saraksta piedāvā dažas masveida rediģēšanas opcijas:

- *Iespējot hosts* - mainīt visu grupā esošo hostu statusu uz ”Monitored”
- *Atspējot hosts* - mainīt visu grupā esošo hostu statusu uz ”Not monitored”

- *Dzēst* - dzēst hostu grupas

Lai izmantotu šīs opcijas, atzīmējiet izvēles rūtiņas attiecīgo hostu grupu priekšā un pēc tam noklikšķiniet uz vajadzīgās pogas.

Filtra izmantošana

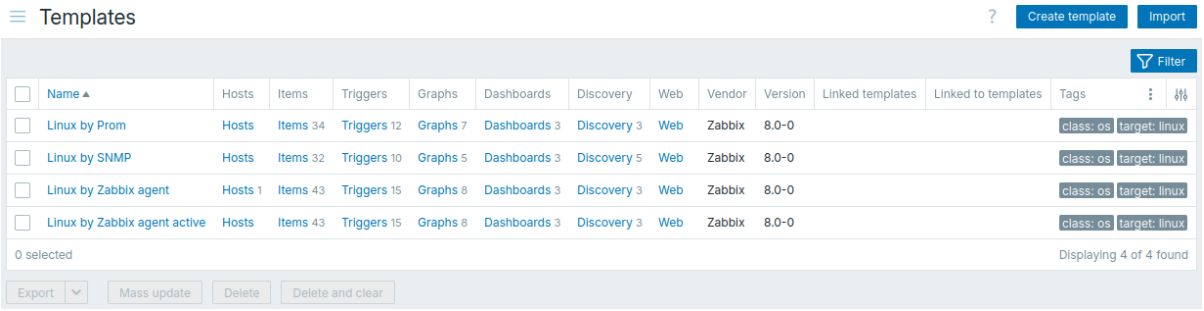
Varat izmantot filtru, lai parādītu tikai tās hostu grupas, kas jūs interesē. Labākai meklēšanas veiktspējai dati tiek meklēti ar neatrisinātām makrokomandām.

Veidnes

Pārskats

Sadaļā *Datu apkopošana* > *Veidnes* lietotāji var konfigurēt un uzturēt veidnes.

Tiek parādīts esošo veidņu saraksts ar to detalizētu informāciju.



Veidņu informācija tiek parādīta pielāgojamā tabulā. Tabulas galvenē esošā poga *Pielāgot tabulu* ļauj parādīt, paslēpt vai mainīt kolonnu secību, kā arī atiestatīt izkārtojumu uz noklusējuma iestatījumiem. Katras kolonnas izmēru var mainīt arī, velkot tās labo malu; veiciet dubultklikšķi uz malas, lai atiestatītu izmēru. Visi pielāgojumi tiek lietoti nekavējoties un vienmēr tiek saglabāti pašreizējam lietotājam.

Ir pieejamas šādas kolonnas:

Kolonna	Apraksts
Nosaukums	Veidnes nosaukums. Noklikšķinot uz veidnes nosaukuma, tiek atvērta veidnes konfigurācijas veidlapa.
Hosti	Rediģējamo hostu skaits, ar kuriem ir saistīta veidne; tikai lasāmi hosti netiek iekļauti. Noklikšķinot uz *Hosti*, tiek atvērts hostu saraksts, kurā tiek parādīti tikai ar veidni saistītie hosti.
Entītijas (vienumi, trigeri, grafiki, informācijas paneļi, atklāšana, tīmeklis)	Attiecīgo veidnes entītiju skaits (parādīts pelēkā krāsā). Noklikšķinot uz entītijas nosaukuma, pilnajā šīs entītijas sarakstā tiek atlasītas tikai tās entītijas, kas pieder veidnei.
Saistītās veidnes	Veidnes, kas ir saistītas ar šo veidni.
Saistīts ar veidnēm	Veidnes, ar kurām šī veidne ir saistīta.
Piegādātājs, *Versija*	Veidnes piegādātājs un versija; tiek parādīti, ja veidnes konfigurācijā ir ietverta šāda informācija, un tikai gatavām veidnēm, importētām veidnēm vai veidnēm, kas modificētas, izmantojot Template API. Gatavām veidnēm versija tiek parādīta šādi: Zabbix galvenā versija, atdalītājs (”-”), pārskatījuma numurs (palielināts ar katru jaunu veidnes versiju un atiestatīts ar katru Zabbix galveno versiju). Piemēram, 7.0-0, 7.0-3, 8.0-0, 8.0-3.
Birkas	Tiek parādītas veidnes birkas, makro neaizstājot. Novietojot kursoru virs vai noklikšķinot uz trīs punktu ikonas pēc parādītajām birkām, tiek atvērts uznirstošais logs ar visām birkām. Noklikšķinot uz pogas *Opcijas* kolonnas galvenē, var iestatīt parādāmo birku skaitu (1–3), izvēlēties, kā parādīt birku nosaukumus (pilnus, saīsinātus vai nerādīt), iestatīt birku parādīšanas prioritāti vai dublēt kolonnu (dublētās kolonnas var pārdēvēt vai dzēst).
Birkas vērtība	Tiek parādītas veidnes birku vērtības, makro neaizstājot. Noklikšķinot uz pogas *Opcijas* kolonnas galvenē, var pārdēvēt kolonnu, izvēlēties, kuras birku vērtības parādīt, vai dublēt kolonnu (dublētās kolonnas var pārdēvēt vai dzēst).

Kolonna	Apraksts
Pielāgots teksts	Tiek parādīts pielāgots teksts, makro vērtības vai abu kombinācija. Noklikšķinot uz pogas ⋮ *Opcijas* kolonnas galvenē, var pārdēvēt kolonnu, ievadīt pielāgoto tekstu vai makro, vai dublēt kolonnu (dublētās kolonnas var pārdēvēt vai dzēst). Atbalstītie makro: {HOST.ID}, {HOST.HOST}, {HOST.NAME}, {HOST.DESCRIPTION} un lietotāja makro.

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*.

Masveida rediģēšanas opcijas

Pogas zem saraksta piedāvā vairākas masveida rediģēšanas opcijas:

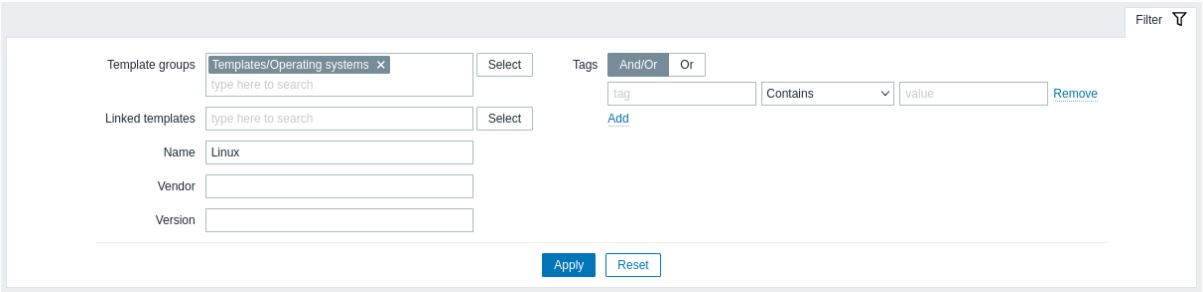
- *Eksportēt* - eksportēt veidni YAML, XML vai JSON failā;
- *Masveida atjaunināšana* - atjaunināt vairākus rekvizītus vairākiem veidnēm vienlaikus;
- *Dzēst* - dzēst veidni, atstājot tās saistītos objektus (vienumus, trigerus u. c.) pie hosts;
- *Dzēst un notīrīt* - dzēst veidni un tās saistītos objektus no hosts.

Lai izmantotu šīs opcijas, atzīmējiet izvēles rūtiņas pie attiecīgajām veidnēm, pēc tam noklikšķiniet uz vajadzīgās pogas.

Filtra izmantošana

Varat izmantot filtru, lai parādītu tikai tās veidnes, kas jūs interesē. Lai meklēšana būtu ātrāka, dati tiek meklēti ar neatrisinātiem makro.

Saite *Filter* ir pieejama zem pogām *Create template* un *Import*. Ja uz tās noklikšķināsiet, kļūs pieejams filtrs, kurā varat filtrēt veidnes pēc veidņu grupas, saistītajām veidnēm, nosaukuma un tagiem.



Parameter	Description
Template groups	Filtrēt pēc vienas vai vairākām veidņu grupām. Norādot vecāko veidņu grupu, automātiski tiek atlasītas arī visas ligzdotās grupas.
Linked templates	Filtrēt pēc tieši saistītām veidnēm.
Name	Filtrēt pēc veidnes nosaukuma.
Vendor	Filtrēt pēc veidnes piegādātāja.
Version	Filtrēt pēc veidnes versijas.

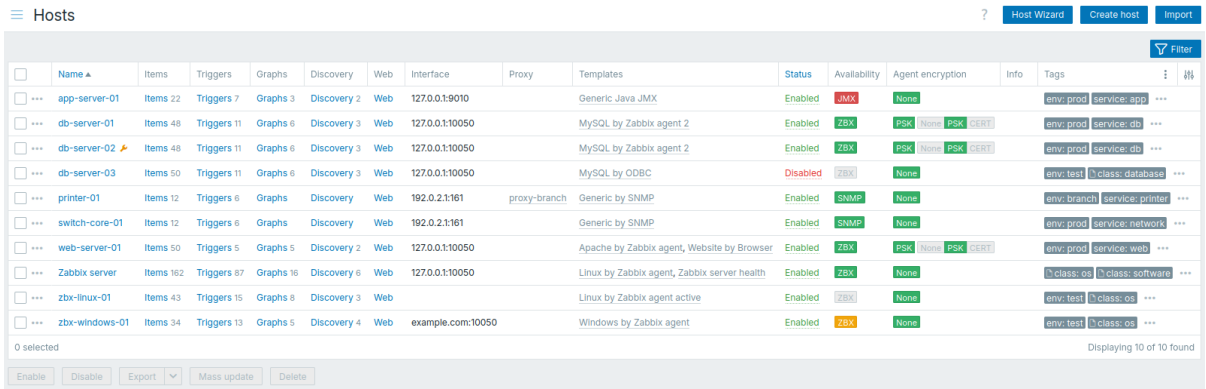
Parameter	Description
Tags	Filtrēt pēc veidnes taga nosaukuma un vērtības. Filtrēšana ir iespējama tikai pēc veidnes līmeņa tagiem (nevis mantotajiem). Ir iespējams iekļaut, kā arī izslēgt konkrētus tagus un tagu vērtības. Var iestatīt vairākus nosacījumus. Tagu nosaukumu atbilstība vienmēr ir reģistrjutīga. 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, nav reģistrjutīga); **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, nav reģistrjutīga). 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.

Hosts

Pārskats

Sadaļā *Datu apkopošana* > *Hosti* lietotāji var konfigurēt un uzturēt hostus.

Tiek parādīts esošo hostu saraksts ar to detalizētu informāciju.



Hostu detalizētā informācija tiek parādīta pielāgojamā tabulā. Tabulas galvenē esošā poga *Pielāgot tabulu* ļauj parādīt, paslēpt vai pārkārtot kolonnas, kā arī atiestatīt izkārtojumu uz noklusējuma iestatījumiem. Katras kolonnas izmēru var mainīt arī, velkot tās labo malu; veiciet dubultklikšķi uz malas, lai atiestatītu izmēru. Visi pielāgojumi tiek lietoti nekavējoties un vienmēr tiek saglabāti pašreizējam lietotājam.

Ir pieejamas šādas kolonnas:

Kolonna	Apraksts
Nosaukums	Hosta nosaukums. Noklikšķinot uz hosta nosaukuma, tiek atvērta hosta konfigurācijas veidlapa. Oranža uzgriežņa ikona aiz hosta nosaukuma norāda, ka hosts atrodas uzturēšanas režīmā. Novietojot peles rādītāju uz ikonas, tiek parādīta informācija par uzturēšanu.
Entītijas (vienumi, trigeri, grafiki, atklāšana, tīmeklis)	Noklikšķinot uz entītijas nosaukuma, tiek parādīti hosta vienumi, trigeri u. c. Attiecīgo entītiju skaits tiek parādīts pelēkā krāsā.
Saskarne	Tiek parādīta hosta galvenā saskarne.

Kolonna	Apraksts
Starpniekserveris	Šajā kolonnā tiek parādīti piešķirtie starpniekserveri: <Proxy name> - hosts tiek uzraudzīts, izmantojot atsevišķu starpniekserveri (pat ja starpniekserveris ir starpniekserveru grupas daļa); <Proxy group name: proxy name> - hosts tiek uzraudzīts, izmantojot starpniekserveru grupu, un Zabbix serveris ir piešķīris starpniekserveri hosta uzraudzībai; <Proxy group name> - hosts tiek uzraudzīts, izmantojot starpniekserveru grupu bez starpniekserveriem vai ja Zabbix serveris nav piešķīris starpniekserveri hosta uzraudzībai; Nekas - hosts netiek uzraudzīts, izmantojot ne starpniekserveri, ne starpniekserveru grupu. Šī kolonna tiek parādīta tikai tad, ja filtra opcijai *Uzrauga* ir iestatīta vērtība "Jebkurš", "Starpniekserveris" vai "Starpniekserveru grupa".
Veidnes	Tiek parādītas ar hostu saistītās veidnes. Ja saistītajā veidnē ir ietvertas citas veidnes, tās tiek parādītas iekavās, atdalot ar komatu. Noklikšķinot uz veidnes nosaukuma, tiek atvērta tās konfigurācijas veidlapa.
Statuss	Tiek parādīts hosta statuss - *Iespējots* vai *Atspējots*. Noklikšķinot uz statusa, to var mainīt manuāli. Atklātie hosti, kas ir pazuduši, tiek atzīmēti ar informācijas ikonu. Rīka padomā 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. Tāpat, ja iespējojat hostu, tā pieejamība tiks atjaunināta pēc tam, kad Zabbix serveris būs sinhronizējis konfigurācijas izmaiņas un aptaujājis hostu. Pieejamības ikonas attēlo tikai to konfigurēto saskarņu tipus (aģents, SNMP, IPMI, JMX). Novietojot peles rādītāju uz ikonas, tiek parādīts uznirstošais logs ar visu tā paša tipa saskarņu sarakstu, to 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 arī tad, ja hostam nav konfigurēta aģenta saskarne. Kolonna ir tukša hostiem, kuriem nav saskarņu. Atsevišķas 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 (noildze, 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 sadaļās Nezināms saskarnes statuss un Nesasniedzamas/nepieejamas 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. **Aktīvās pārbaudes pieejamība.** Ja hostam ir iespējota vismaz viena aktīvā pārbaude, aktīvās pārbaudes ietekmē arī kopējo aģenta saskarnes pieejamību, kā aprakstīts iepriekš. Lai noteiktu aktīvās pārbaudes pieejamību, aģenta aktīvo pārbaužu pavedienā tiek nosūtīti 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 pēdējais 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.
Aģenta šifrēšana	Tiek parādīts savienojumu ar hostu un no hosta šifrēšanas statuss: **None** - šifrēšana netiek izmantota; **PSK** - tiek izmantota iepriekš koplietota atslēga; **Cert** - tiek izmantots sertifikāts.
Informācija	Tiek parādīta ar hostu saistītā informācija par kļūdām (ja tāda ir).

Kolonna	Apraksts
Tagi	Tiek parādīti hosta tagi, neatrisinot makro. Novietojot peles rādītāju uz parādītajiem tagiem vai noklikšķinot uz ikonas ar trim punktiem, tiek atvērts uznirstošais logs ar visiem tagiem. Noklikšķinot uz pogas *Opcijas* kolonnas galvenē, var iestatīt parādāmo tagu skaitu (1–3), izvēlēties tagu nosaukumu attēlošanas veidu (pilns, saīsināts vai nerādīt), iestatīt tagu attēlošanas prioritāti vai dublēt kolonnu (dublētās kolonnas var pārdēvēt vai dzēst).
Tagu vērtība	Tiek parādītas hosta tagu vērtības, neatrisinot makro. Noklikšķinot uz pogas *Opcijas* kolonnas galvenē, var pārdēvēt kolonnu, izvēlēties, kuras tagu vērtības parādīt, vai dublēt kolonnu (dublētās kolonnas var pārdēvēt vai dzēst).
Pielāgots teksts	Tiek parādīts pielāgots teksts, makro vērtības vai abu kombinācija. Noklikšķinot uz pogas *Opcijas* kolonnas galvenē, var pārdēvēt kolonnu, ievadīt pielāgotu tekstu vai makro, vai dublēt kolonnu (dublētās kolonnas var pārdēvēt vai dzēst). Atbalstītie makro: `{HOST.*}` (izņemot `{HOST.METADATA}`), `{INVENTORY.*}` un lietotāja makro.

Lai atvērtu hosta izvēlni, noklikšķiniet uz pogas *Opcijas* pirms hosta nosaukuma.

Lai konfigurētu jaunu hostu, augšējā labajā stūrī noklikšķiniet uz pogas *Hosta vednis* vai *Izveidot hostu*.

Lai importētu hostu no YAML, XML vai JSON faila, augšējā labajā stūrī noklikšķiniet uz pogas *Importēt*.

Masveida rediģēšanas opcijas

Pogas zem saraksta piedāvā vairākas masveida rediģēšanas opcijas:

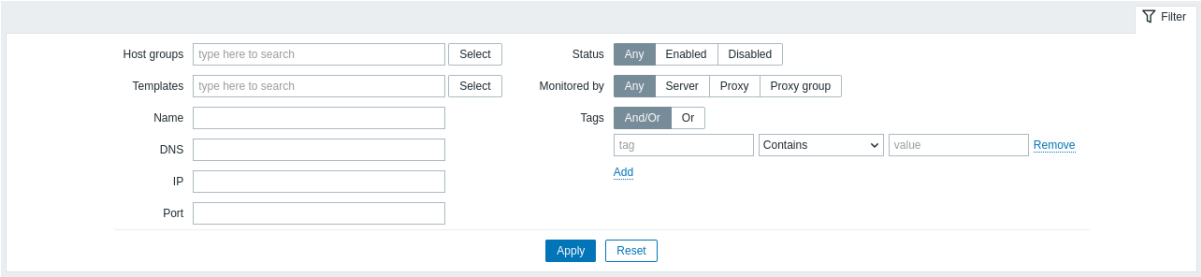
- *Iespējot* - mainīt hosta statusu uz *Uzraudzīts*;
- *Atspējot* - mainīt hosta statusu uz *Netiek uzraudzīts*;
- *Eksportēt* - eksportēt hostus YAML, XML vai JSON failā;
- *Masveida atjaunināšana* - atjaunināt vairākas īpašības vairākiem hostiem vienlaikus;
- *Dzēst* - dzēst hostus.

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 hosts, kas jūs interesē. Lai uzlabotu meklēšanas veiktspēju, dati tiek meklēti ar neatrisinātiem makro.

Ikona *Filter* ir pieejama augšējā labajā stūrī. Noklikšķinot uz tās, tiks atvērts filtrs, kurā varat norādīt vajadzīgos filtrēšanas kritērijus.



Parameter	Description
Host groups	Filtrēt pēc viena vai vairākiem host grupām. Norādot vecāko host grupu, netieši tiek atlasītas visas ligzdotās host grupas.
Templates	Filtrēt pēc saistītajām veidnēm.
Name	Filtrēt pēc redzamā host nosaukuma.
DNS	Filtrēt pēc DNS nosaukuma.
IP	Filtrēt pēc IP adreses.
Port	Filtrēt pēc porta numura.
Status	Filtrēt pēc host statusa.
Monitored by	Filtrēt hosts, kurus uzrauga Zabbix serveris, starpniekserveris vai starpniekserveru grupa.
Proxies	Filtrēt hosts, kurus uzrauga šeit norādītie starpniekserveri. Šis lauks ir pieejams tikai tad, ja laukā *Monitored by* ir atlasīts "Proxy".
Proxy groups	Filtrēt hosts, kurus uzrauga šeit norādītās starpniekserveru grupas. Šis lauks ir pieejams tikai tad, ja laukā *Monitored by* ir atlasīts "Proxy group".

Parameter	Description
Tags	Filtrēt pēc host taga nosaukuma un vērtības. Ir iespējams iekļaut, kā arī izslēgt konkrētus tagus un tagu vērtības. Var iestatīt vairākus nosacījumus. Taga nosaukuma atbilstība vienmēr ir reģistrjutīga. Katrai nosacījuma rindai ir pieejami vairāki operatori: **Exists** - iekļaut norādītos taga nosaukumus; **Equals** - iekļaut norādītos taga nosaukumus un vērtības (reģistrjutīgi); **Contains** - iekļaut norādītos taga nosaukumus, kuru taga vērtības satur ievadīto virkni (apakšvirknes atbilstība, nav reģistrjutīga); **Does not exist** - izslēgt norādītos taga nosaukumus; **Does not equal** - izslēgt norādītos taga nosaukumus un vērtības (reģistrjutīgi); **Does not contain** - izslēgt norādītos taga nosaukumus, kuru taga vērtības satur ievadīto virkni (apakšvirknes atbilstība, nav reģistrjutīga). 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.

Nezināms interfeisa statuss

Zabbix serveris parāda ”Unknown” statusu hosts interfeisam (Agent, SNMP, IPMI, JMX) šādos gadījumos:

- Hosts ir atspējots.
- Hosts ir iestatīts uzraudzībai ar starpniekserveri, citu starpniekserveri vai serveri, ja tas iepriekš tika uzraudzīts ar starpniekserveri.
- Hosts tiek uzraudzīts ar starpniekserveri, kas šķiet bezsaistē (nav saņemti atjauninājumi no starpniekservera maksimālā sirdspuksta intervāla laikā - 1 stunda).
- Visi hosts vienumi ar šāda tipa interfeisu ir atspējoti.
- Nav konfigurēti neviena šī interfeisa tipa aptaujātāji (piemēram, servera konfigurācijas parametrs `StartAgentPollers` vai `StartSNMPPollers` ir iestatīts uz 0).

Interfeisa pieejamība tiek iestatīta uz ”Unknown” pēc Zabbix servera konfigurācijas kešatmiņas sinhronizācijas.

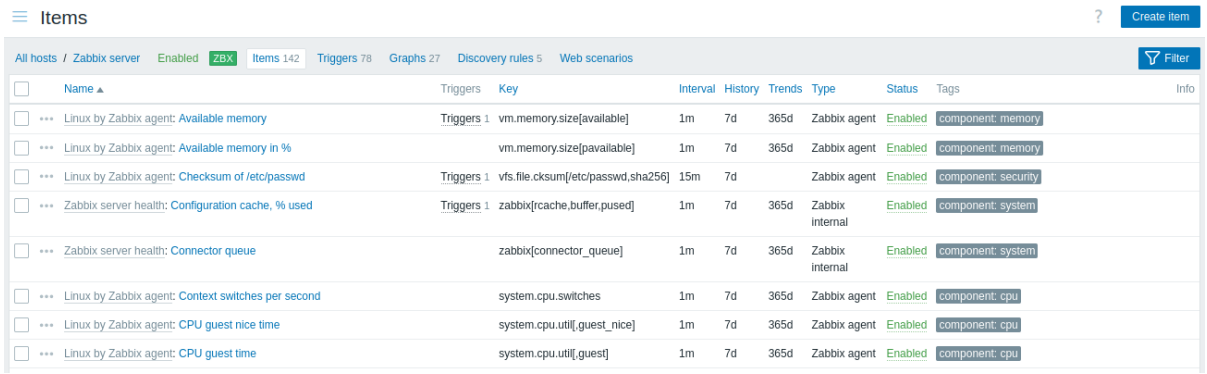
Interfeisa pieejamība (Available/Not available) hosts, ko uzrauga starpniekserveri, tiek atjaunota pēc starpniekservera konfigurācijas kešatmiņas sinhronizācijas.

Vienumi

Pārskats

Hosts vienumu sarakstam var piekļūt sadaļā *Datu apkopošana* > *Hosts*, noklikšķinot uz *Vienumi* pie attiecīgā hosts.

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 hosts nosaukuma, tiek atvērta hosts konfigurācijas forma. Šī kolonna tiek parādīta tikai tad, ja filtrā ir atlasīti vairāki hosts vai neviens hosts.
Nosaukums	Vienuma nosaukums tiek parādīts kā zila saite uz vienuma detalizēto informāciju. Noklikšķinot uz vienuma nosaukuma saites, tiek atvērta vienuma konfigurācijas forma. Ja hosts vienums pieder veidnei, pirms vienuma nosaukuma kā pelēka saite tiek parādīts veidnes nosaukums. Noklikšķinot uz veidnes saites, tiek atvērts vienumu saraksts veidnes līmenī. Ja vienums ir izveidots no vienuma prototipa, pirms tā nosaukuma oranžā krāsā tiek parādīts zema līmeņa atklāšanas noteikuma nosaukums. Noklikšķinot uz atklāšanas noteikuma nosaukuma, tiek atvērts vienumu prototipu saraksts.
Trigeri	Novietojot peles rādītāju virs trigeriem, tiek 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. *Ņemiet vērā*, ka pasīvos vienumus var pārbaudīt arī nekavējoties, nospiežot *Izpildīt tūlīt* pogu.
Vēsture	Tiek parādīts, cik dienas tiks saglabāta vienuma datu vēsture.
Tendences	Tiek parādīts, cik dienas tiks saglabāta vienuma tendenču vēsture.
Tips	Tiek parādīts vienuma tips (Zabbix aģents, SNMP aģents, vienkārša pārbaude u. c.).
Statuss	Tiek parādīts vienuma statuss — *Iespējots*, *Atspējots* vai *Netiek atbalstīts*. Statusu var mainīt manuāli, noklikšķinot uz tā: no Iespējots uz Atspējots (un otrādi), kā arī no Netiek atbalstīts uz Atspējots (un otrādi). Atklātie vienumi, kas ir pazuduši, tiek atzīmēti ar informācijas ikonu. Rīka padomā ir sniegta detalizēta informācija par to statusu.
Tagi	Tiek parādīti vienuma tagi. Var tikt parādīti ne vairāk kā trīs tagi (nosaukuma un vērtības pāri). Ja tagu ir vairāk, tiek parādīta saite "...", kas, novietojot virs tās peles rādītāju, ļauj skatīt visus tagus.
Informācija	Ja vienums darbojas pareizi, šajā kolonnā ikona netiek parādīta. Kļūdu gadījumā tiek parādīta kvadrātveida ikona ar burtu "i". Novietojiet peles rādītāju virs ikonas, lai skatītu rīka padomu ar kļūdas aprakstu.

Lai konfigurētu jaunu vienumu, augšējā labajā stūrī noklikšķiniet uz pogas *Izveidot vienumu*.

Masveida rediģēšanas opcijas

Pogas zem saraksta piedāvā vairākas masveida rediģēšanas opcijas:

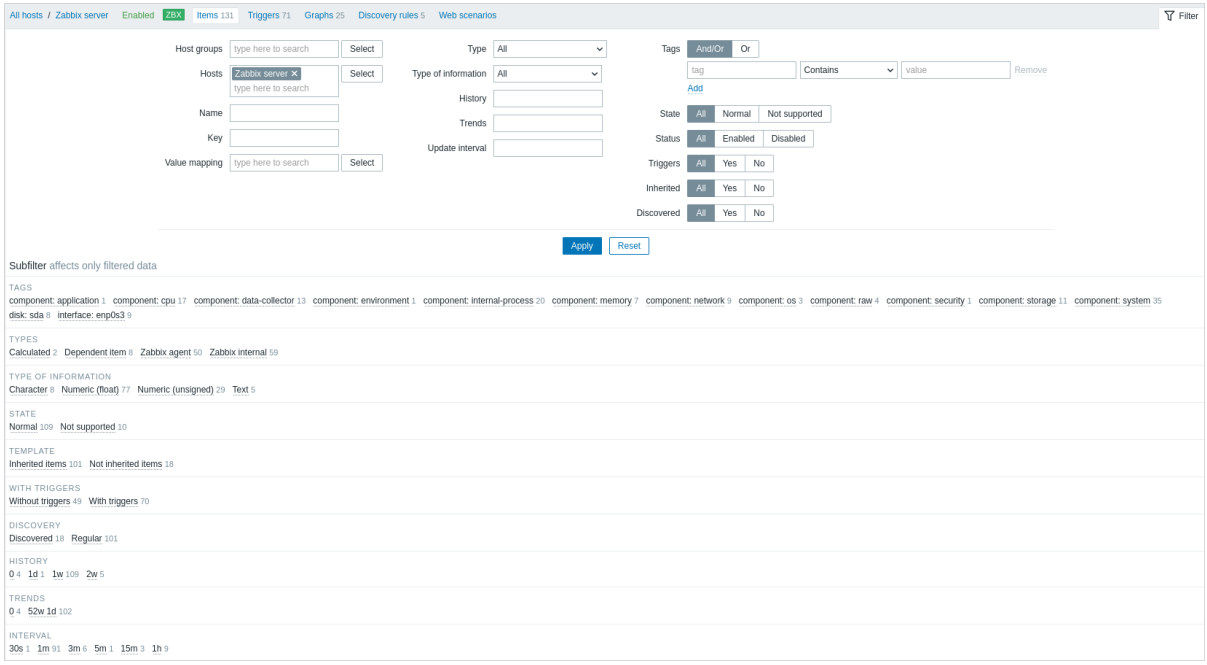
- *Iespējot* - mainīt vienuma statusu uz *Iespējots*
- *Atspējot* - mainīt vienuma statusu uz *Atspējots*
- *Izpildīt tagad* - nekavējoties izpildīt pārbaudi, lai iegūtu jaunās vienuma vērtības. Atbalstīts tikai **pasīvajām** pārbaudēm (skatiet sīkāku informāciju). Ņemiet vērā, ka, nekavējoties pārbaudot vērtības, konfigurācijas kešatmiņa netiek atjaunināta, tāpēc vērtības neatspoguļos pavisam nesen veiktās izmaiņas vienuma konfigurācijā.
- *Notīrīt vēsturi un tendences* - dzēst vienumu vēstures un tendenču datus.
- *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

Varat izmantot filtru, lai parādītu tikai tos vienumus, kas jūs interesē. Lai uzlabotu meklēšanas veiktspēju, dati tiek meklēti ar neatrisinātiem makro.

Ikona *Filter* ir pieejama augšējā labajā stūrī. Noklikšķinot uz tās, tiks atvērts filtrs, kurā varat norādīt vēlamos filtrēšanas kritērijus.



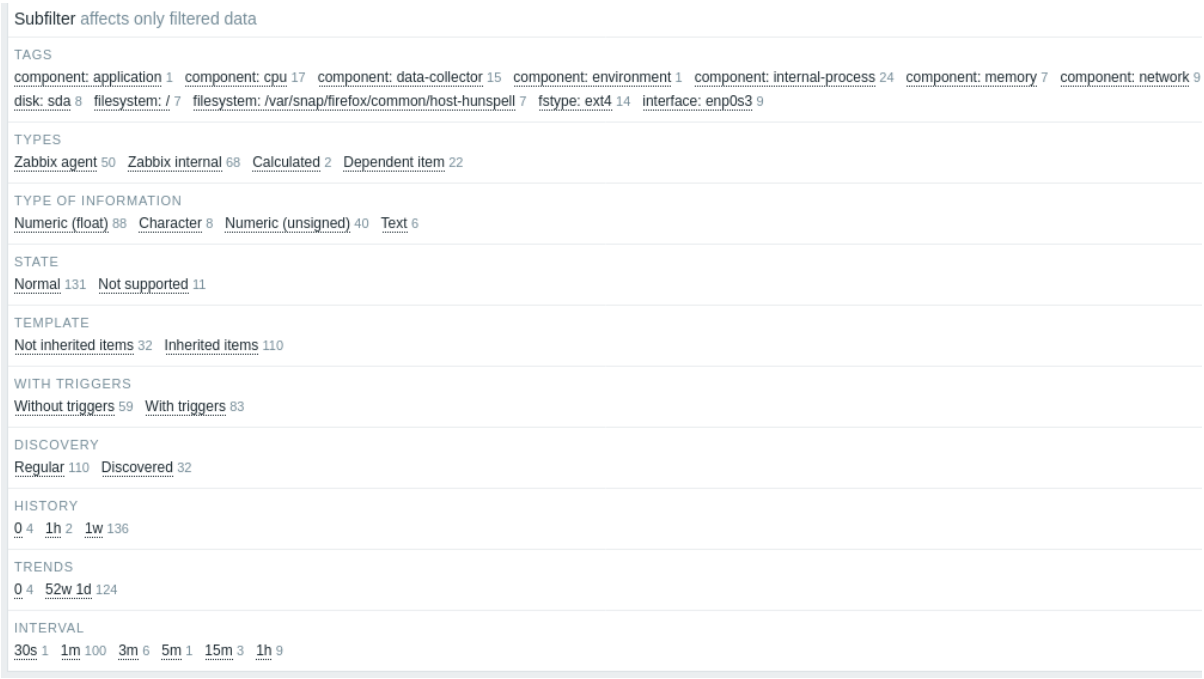
Parameter	Description
Host groups	Filtrēt pēc viena vai vairākiem hostu grupām. Norādot vecāko hostu grupu, automātiski tiek atlasītas arī visas ligzdotās hostu grupas. Hostu grupas, kurās ir tikai veidnes, nevar atlasīt.
Hosts	Filtrēt pēc viena vai vairākiem hosts.
Name	Filtrēt pēc vienuma nosaukuma.
Key	Filtrēt pēc vienuma atslēgas.
Value mapping	Filtrēt pēc izmantotās vērtību kartes. Šis parametrs netiek rādīts, ja opcija *Hosts* ir tukša.
Type	Filtrēt pēc vienuma tipa (Zabbix aģents, SNMP aģents utt.).
Type of information	Filtrēt pēc informācijas tipa (bez zīmes skaitlis, peldošais punkts utt.).
History	Filtrēt pēc tā, cik ilgi tiek glabāta vienuma vēsture.
Trends	Filtrēt pēc tā, cik ilgi tiek glabātas vienuma tendences.
Update interval	Filtrēt pēc vienuma atjaunināšanas intervāla.
Tags	Norādiet tagus, lai ierobežotu parādīto vienumu skaitu. Ir iespējams iekļaut, kā arī izslēgt konkrētus tagus un tagu vērtības. Var iestatīt vairākus nosacījumus. Tagu nosaukumu 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, nereģistrjutīgi) **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, nereģistrjutīgi) 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
State	Filtrēt pēc vienuma stāvokļa - *Normal* vai *Not supported*.
Status	Filtrēt pēc vienuma statusa - *Enabled* vai *Disabled*.
Triggers	Filtrēt vienumus ar trigeriem vai bez tiem.
Inherited	Filtrēt vienumus, kas ir mantoti no veidnes, vai nav mantoti.
Discovery	Filtrēt vienumus, kas ir atklāti vai nav atklāti ar zemā līmeņa atklāšanu.

Apakšfiltra izmantošana

Apakšfiltrs nodrošina galvenā filtra lietotā filtrējuma papildu modificēšanu.

Tas satur noklikšķināmas saites ātrai piekļuvei saistītajiem vienumiem. Vienumus saista kopīga entītija, piemēram, tags, vienuma

tips, vienuma stāvoklis, vienuma atjaunināšanas intervāls u. c. Noklikšķinot uz entītijas, tā tiek izcelta ar pelēku fonu, un vienumi tiek nekavējoties filtrēti (galvenajā filtrā nav jānoklikšķina uz *Lietot*). Noklikšķinot uz citas entītijas, tā tiek pievienota filtrētajiem rezultātiem. Vēlreiz noklikšķinot uz entītijas, filtrēšana tiek noņemta.



Apakšfiltri tiek ģenerēti, pamatojoties uz filtrētajiem datiem, kas ir ierobežoti līdz 1000 ierakstiem. Ja apakšfiltrā vēlaties skatīt vairāk ierakstu, jāpalielina parametra *Meklēšanas un filtrēšanas rezultātu ierobežojums* vērtība (sadaļā *Administrēšana* > *Vispārīgi* > *Lietotāja saskarne*).

Atšķirībā no galvenā filtra apakšfiltrs tiek atjaunināts ar katru tabulas atsvaidzināšanas pieprasījumu, lai vienmēr rādītu aktuālu informāciju par pieejamajām filtrēšanas opcijām un to skaitu.

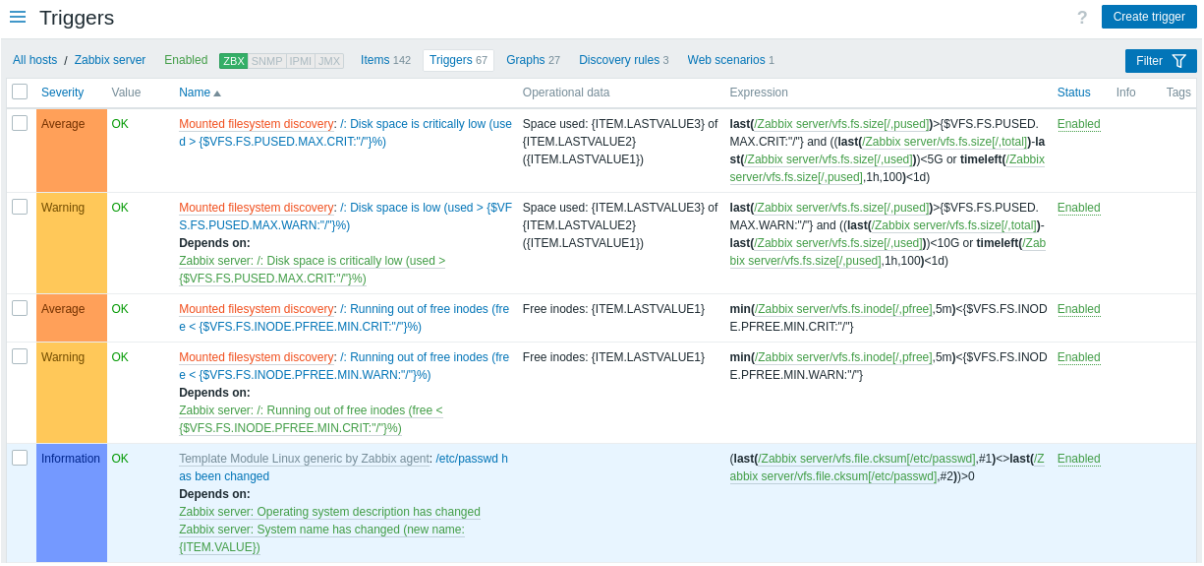
Horizontāli tiek rādītas ne vairāk kā 100 entītijas. Ja entītiju ir vairāk, beigās tiek parādīta trīs punktu ikona; uz tās nevar noklikšķināt.

Skaitlis blakus katrai noklikšķināmajai entītijai norāda tajā sagrupēto vienumu skaitu (pamatojoties uz galvenā filtra rezultātiem). Kad entītija ir atlasīta, blakus citām pieejamajām entītijām tiek rādīti skaitļi ar pluszīmi, kas norāda, cik vienumu var pievienot pašreizējai atlasei. Entītijas bez vienumiem netiek rādītas, izņemot gadījumus, kad tās iepriekš ir atlasītas apakšfiltrā.

Trigeri

Pārskats

Hosts trigeru sarakstam var piekļūt sadaļā *Datu apkopošana* > *Hosts*, noklikšķinot uz *Trigeri* pie attiecīgā hosts.



Parādītie dati:

Kolonna	Apraksts
Svarīgums	Trigera svarīgums tiek parādīts gan ar nosaukumu, gan šūnas fona krāsu.
Vērtība	Tiek parādīts trigera stāvoklis: **OK** - stāvoklis OK **PROBLEM** - problēmas stāvoklis
Hosts	Trigera hosts. Noklikšķinot uz hosts nosaukuma, tiek atvērta hosts konfigurācijas veidlapa. Šī kolonna tiek parādīta tikai tad, ja filtrā ir atlasīti vairāki hosts vai nav atlasīts neviens hosts.
Nosaukums	Trigera nosaukums tiek parādīts kā zila saite uz trigera detalizēto informāciju. Noklikšķinot uz trigera nosaukuma saites, tiek atvērta trigera konfigurācijas veidlapa. Ja hosts trigeris pieder veidnei, pirms trigera nosaukuma kā pelēka saite tiek parādīts veidnes nosaukums. Noklikšķinot uz veidnes saites, tiks atvērts trigeru saraksts veidnes līmenī. Ja trigeris ir izveidots no trigera prototipa, pirms tā nosaukuma oranžā krāsā tiek parādīts zema līmeņa atklāšanas noteikuma nosaukums. Noklikšķinot uz atklāšanas noteikuma nosaukuma, tiek atvērts trigeru prototipu saraksts.
Darbības dati	Trigera darbības datu definīcija, kas satur patvaļīgas virknes un makro, kuri sadaļā *Uzraudzība* > *Problēmas* tiks dinamiski aizstāti ar to vērtībām.
Izteiksme	Tiek parādīta trigera izteiksme. Izteiksmes hosts-vienuma daļa tiek parādīta kā saite, kas ved uz vienuma konfigurācijas veidlapu.
Statuss	Tiek parādīts trigera statuss - *Iespējots*, *Atspējots* vai *Nezināms*. Noklikšķinot uz statusa, to var manuāli mainīt - no Iespējots uz Atspējots (un otrādi), kā arī no Nezināms uz Atspējots (un otrādi). Atspējota trigera problēmas lietotāja saskarnē vairs netiek parādītas, taču tās netiek dzēstas. Zudušie atklātie trigeri tiek atzīmēti ar informācijas ikonu. Rīka padoms sniedz detalizētu informāciju par to statusu.
Informācija	Ja viss darbojas pareizi, šajā kolonnā ikona netiek parādīta. Kļūdu gadījumā tiek parādīta kvadrātveida ikona ar burtu ”i”. Novietojiet kursoru virs ikonas, lai skatītu rīka padomu ar kļūdas aprakstu.
Tagi	Ja trigeris satur tagus, šajā kolonnā tiek parādīts taga nosaukums un vērtība.

Lai konfigurētu jaunu trigeri, augšējā labajā stūrī noklikšķiniet uz pogas *Izveidot trigeri*.

Masveida rediģēšanas opcijas

Pogas zem saraksta piedāvā vairākas masveida rediģēšanas opcijas:

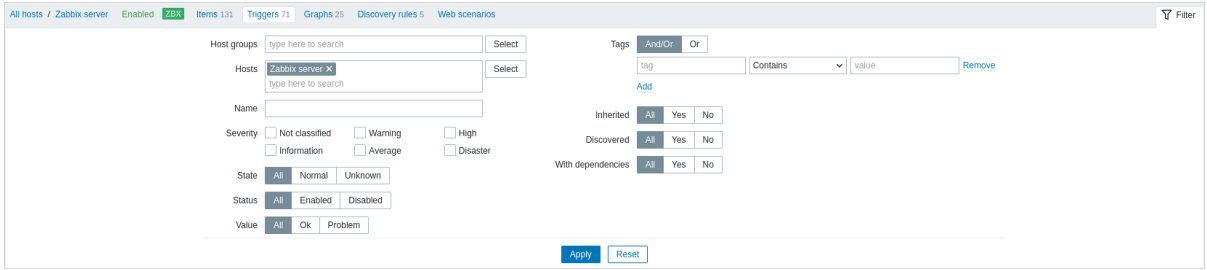
- *Iespējot* - mainīt trigeru statusu uz *Iespējots*.
- *Atspējot* - mainīt trigeru statusu uz *Atspējots*.
- *Kopēt* - kopēt trigerus uz citiem hosts vai veidnēm.
- *Masveida atjaunināšana* - atjaunināt vairākas īpašības vairākiem trigeriem vienlaikus.
- *Dzēst* - dzēst trigerus.

Lai izmantotu šīs opcijas, atzīmējiet izvēles rūtiņas attiecīgo trigeru priekšā, pēc tam noklikšķiniet uz vajadzīgās pogas.

Filtrēšanas izmantošana

Varat izmantot filtru, lai parādītu tikai tos trigerus, kas jūs interesē. Lai meklēšana būtu ātrāka, dati tiek meklēti ar neatrisinātiem makro.

Augšējā labajā stūrī ir pieejama ikona *Filtrs*. Noklikšķinot uz tās, tiks atvērts filtrs, kurā varat norādīt vajadzīgos filtrēšanas kritērijus.



Parameter	Description
Host groups	Filtrēt pēc vienas vai vairākām hostu grupām. Norādot vecāko hostu grupu, netieši tiek atlasītas arī visas ligzdotās hostu grupas. Hostu grupas, kurās ir tikai veidnes, nevar atlasīt.
Hosts	Filtrēt pēc viena vai vairākiem hosts. Ja iepriekš jau ir atlasītas hostu grupas, hostu atlase ir ierobežota līdz šīm grupām.

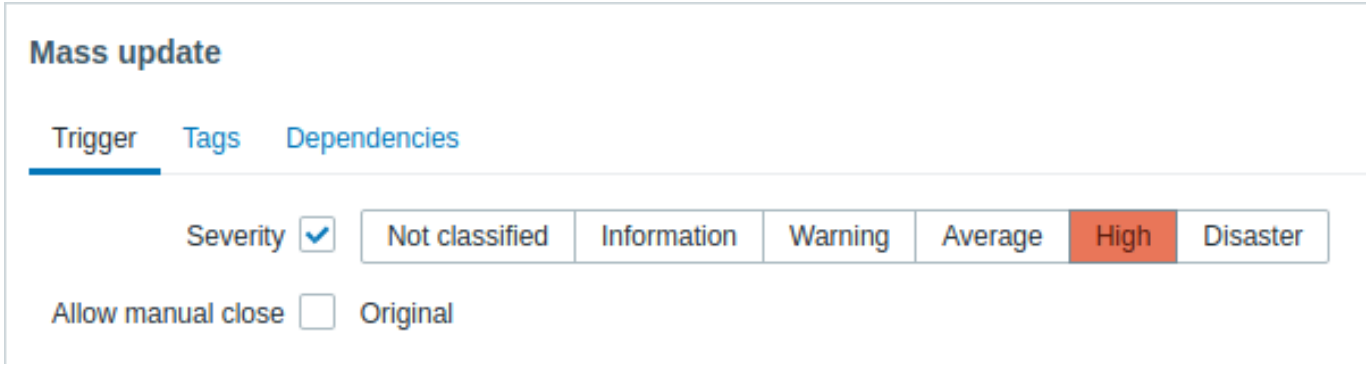
Parameter	Description
Name	Filtrēt pēc trigeru nosaukuma.
Severity	Atlasiet, lai filtrētu pēc vienas vai vairākām trigeru nopietnības pakāpēm.
State	Filtrēt pēc trigeru izteiksmes stāvokļa (Normal/Unknown).
Status	Filtrēt pēc trigeru aktivizācijas statusa (Enabled/Disabled).
Value	Filtrēt pēc trigeru vērtības.
Tags	Filtrēt pēc trigeru taga nosaukuma un vērtības. Ir iespējams iekļaut, kā arī izslēgt konkrētus tagus un tagu vērtības. Var iestatīt vairākus nosacījumus. Taga nosaukuma atbilstība vienmēr ir reģistrjutīga. Katram nosacījumam ir pieejami vairāki operatori: **Exists** - iekļaut norādītos taga nosaukumus **Equals** - iekļaut norādītos taga nosaukumus un vērtības (reģistrjutīgi) **Contains** - iekļaut norādītos taga nosaukumus, kuru taga vērtības satur ievadīto virkni (apakšvirknes atbilstība, nereģistrjutīgi) **Does not exist** - izslēgt norādītos taga nosaukumus **Does not equal** - izslēgt norādītos taga nosaukumus un vērtības (reģistrjutīgi) **Does not contain** - izslēgt norādītos taga nosaukumus, kuru taga vērtības satur ievadīto virkni (apakšvirknes atbilstība, nereģistrjutīgi) Nosacījumiem ir divi aprēķina veidi: **And/Or** - jāizpilda visi nosacījumi, nosacījumi ar vienādu taga nosaukumu tiks grupēti pēc Or nosacījuma **Or** - pietiek, ja ir izpildīts viens nosacījums Gan taga nosaukuma, gan taga vērtības laukos tiek atbalstīti makro un makro funkcijas.
Inherited	Filtrēt trigerus, kas ir mantoti (vai nav mantoti) no veidnes.
Discovered	Filtrēt trigerus, kas ir atklāti (vai nav atklāti) ar zemlīmeņa atklāšanu.
With dependencies	Filtrēt trigerus ar (vai bez) atkarībām.

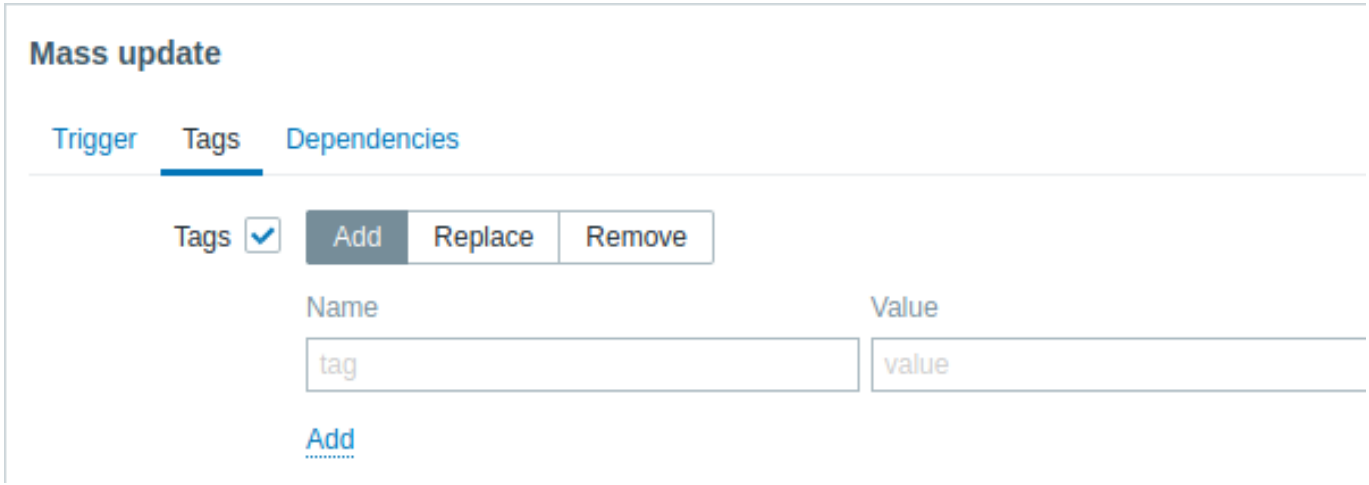
Masveida atjaunināšana

Izmantojot masveida atjaunināšanu, varat vienlaikus mainīt kādu atribūtu vairākiem trigeriem, tādējādi nav jāatver rediģēšanai katrs atsevišķais trigeris.

Lai masveidā atjauninātu trigerus, rīkojieties šādi:

- Atzīmējiet to trigeru izvēles rūtiņas, kurus vēlaties atjaunināt sarakstā
- Zem saraksta noklikšķiniet uz *Mass update*
- Atveriet cilni ar nepieciešamajiem atribūtiem (*Trigger*, *Tags* vai *Dependencies*)
- Atzīmējiet atjaunināmā atribūta izvēles rūtiņu

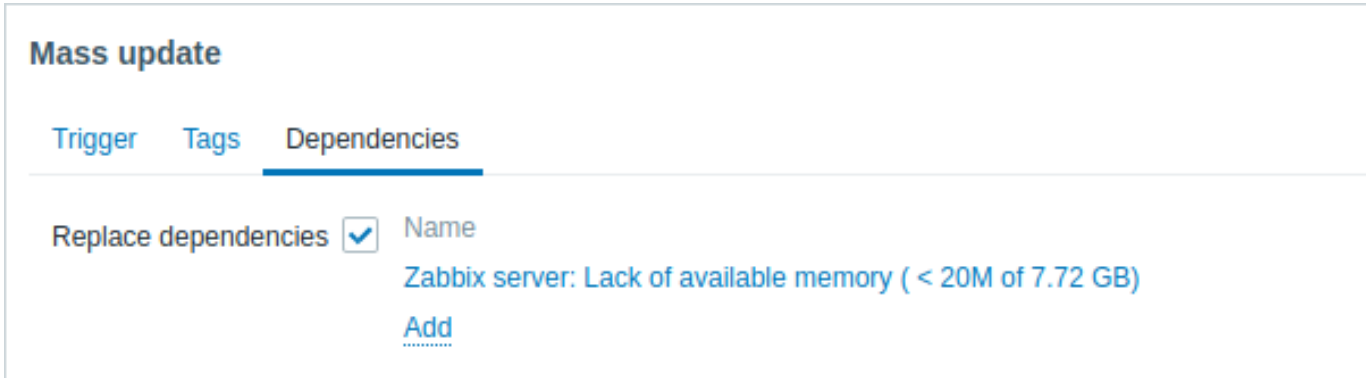




Izvēloties attiecīgo pogu tagu atjaunināšanai, ir pieejamas šādas opcijas:

- *Add* - pievienot trigeriem jaunus tagus;
- *Replace* - noņemt visus esošos taga tagus un aizstāt tos ar tālāk norādīto tagu vai tagiem;
- *Remove* - noņemt no trigeriem norādītos tagus.

Ņemiet vērā, ka tagi ar vienādu nosaukumu, bet atšķirīgām vērtībām, netiek uzskatīti par "dublikātiem" un tos var pievienot vienam un tam pašam trigerim.



Replace dependencies - noņems visas esošās trigeru atkarības un aizstās tās ar norādīto atkarību vai atkarībām.

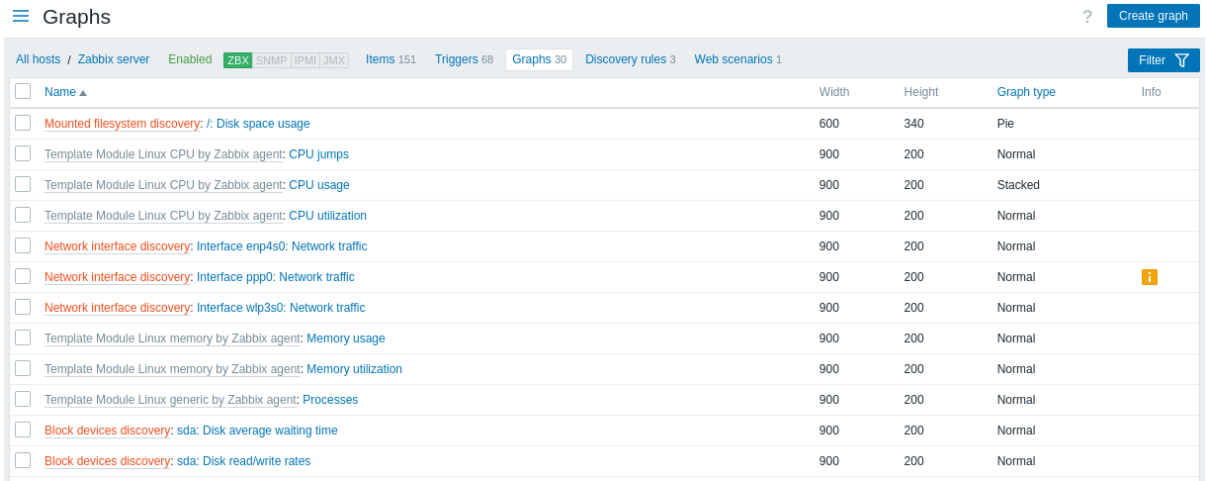
Lai lietotu izmaiņas, noklikšķiniet uz *Update*.

Grafiki

Pārskats

Pielāgoto grafiku sarakstam hosts var piekļūt, izvēlnē *Datu apkopošana* > *Hosts* attiecīgajam hostam noklikšķinot uz *Grafiki*.

Tiek parādīts esošo grafiku saraksts.



Parādītie dati:

Kolonna	Apraksts
Hosts	Grafika hosts. Noklikšķinot uz hosta nosaukuma, tiek atvērta hosta konfigurācijas veidlapa. Šī kolonna tiek parādīta tikai tad, ja filtrā ir atlasīti vairāki hosts vai neviens hosts.
Nosaukums	Pielāgotā grafika nosaukums, kas tiek parādīts kā zila saite uz grafika detalizēto informāciju. Noklikšķinot uz grafika nosaukuma saites, tiek atvērta grafika konfigurācijas veidlapa. Ja hosta grafiks pieder veidnei, pirms grafika nosaukuma kā pelēka saite tiek parādīts veidnes nosaukums. Noklikšķinot uz veidnes saites, tiks atvērts grafiku saraksts veidnes līmenī. Ja grafiks ir izveidots no grafika prototipa, pirms tā nosaukuma oranžā krāsā tiek parādīts zema līmeņa atklāšanas noteikuma nosaukums. Noklikšķinot uz atklāšanas noteikuma nosaukuma, tiek atvērts grafiku prototipu saraksts.
Platums	Tiek parādīts grafika platums.
Augstums	Tiek parādīts grafika augstums.
Grafika tips	Tiek parādīts grafika tips - *Parasts*, *Sakrauts*, *Sektoru* vai *Izkliedēts*.
Informācija	Ja grafiks darbojas pareizi, šajā kolonnā ikona netiek parādīta. Kļūdu gadījumā tiek parādīta kvadrātveida ikona ar burtu "i". Novietojiet kursoru virs ikonas, lai skatītu rīka padomu ar kļūdas aprakstu.

Lai konfigurētu jaunu grafiku, augšējā labajā stūrī noklikšķiniet uz pogas *Izveidot grafiku*.

Masveida rediģēšanas opcijas

Pogas zem saraksta piedāvā dažas masveida rediģēšanas opcijas:

- *Kopēt* - kopēt grafikus uz citiem hosts vai veidnēm
- *Dzēst* - dzēst grafikus

Lai izmantotu šīs opcijas, atzīmējiet izvēles rūtiņas pie attiecīgajiem grafikiem, pēc tam noklikšķiniet uz vajadzīgās pogas.

Filtra izmantošana

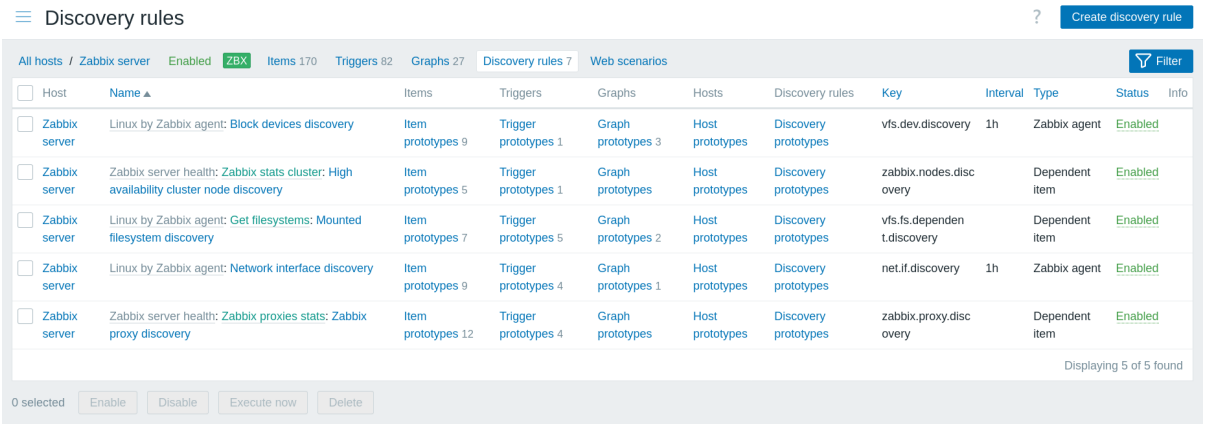
Jūs varat filtrēt grafikus pēc hostu grupas un hosta. Lai uzlabotu meklēšanas veiktspēju, dati tiek meklēti ar neatrisinātām makrokomandām.

Atklāšanas noteikumi

Pārskats

Hosts zema līmeņa atklāšanas noteikumu sarakstam var piekļūt sadaļā *Datu apkopošana* > *Hosts*, attiecīgajam hostam noklikšķinot uz *Atklāšana*.

Tiek parādīts esošo zema līmeņa atklāšanas noteikumu saraksts. Ir iespējams arī skatīt visus atklāšanas noteikumus neatkarīgi no hosta vai konkrētas hostu grupas atklāšanas noteikumus, mainot filtra iestatījumus.



Parādītie dati:

Kolonna	Apraksts
Hosts	Tiek parādīts redzamais hosta nosaukums. Noklikšķinot uz hosta nosaukuma, tiek atvērta hosta konfigurācijas veidlapa. Ja redzamais hosta nosaukums nav pieejams, tiek parādīts tehniskais hosta nosaukums.
Nosaukums	Noteikuma nosaukums tiek parādīts kā zila saite. Noklikšķinot uz noteikuma nosaukuma, tiek atvērta zema līmeņa atklāšanas noteikuma konfigurācijas veidlapa. 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.
Vienumi	Tiek parādīta saite uz vienumu prototipu sarakstu. Pelēkā krāsā tiek parādīts esošo vienumu prototipu skaits.
Trigeri	Tiek parādīta saite uz trigeru prototipu sarakstu. Pelēkā krāsā tiek parādīts esošo trigeru prototipu skaits.
Grafiki	Tiek parādīta saite uz grafiku prototipu sarakstu. Pelēkā krāsā tiek parādīts esošo grafiku prototipu skaits.
Hosts	Tiek parādīta saite uz hostu prototipu sarakstu. Pelēkā krāsā tiek parādīts esošo hostu prototipu skaits.
Atslēga	Tiek parādīta atklāšanai izmantotā vienuma atslēga.
Intervāls	Tiek parādīts atklāšanas veikšanas biežums. *Piezīme*: atklāšanu var veikt arī nekavējoties, zem saraksta nospiežot pogu *Izpildīt tūlīt*.
Tips	Tiek parādīts atklāšanai izmantotais vienuma tips (Zabbix aģents, SNMP aģents u. c.).
Statuss	Tiek parādīts atklāšanas noteikuma statuss — *Iespējots*, *Atspējots* vai *Netiek atbalstīts*. Noklikšķinot uz statusa, to var mainīt — no "Iespējots" uz "Atspējots" (un otrādi), kā arī no "Netiek atbalstīts" uz "Atspējots" (un otrādi).
Informācija	Ja viss ir kārtībā, šajā kolonnā ikona netiek parādīta. Kļūdu gadījumā tiek parādīta kvadrātveida ikona ar burtu "i". Novietojiet kursoru virs ikonas, lai skatītu rīka padomu ar kļūdas aprakstu.

Lai konfigurētu jaunu zema līmeņa atklāšanas noteikumu, augšējā labajā stūrī noklikšķiniet uz pogas *Izveidot atklāšanas noteikumu*.

Masveida rediģēšanas opcijas

Pogas zem saraksta piedāvā vairākas masveida rediģēšanas opcijas:

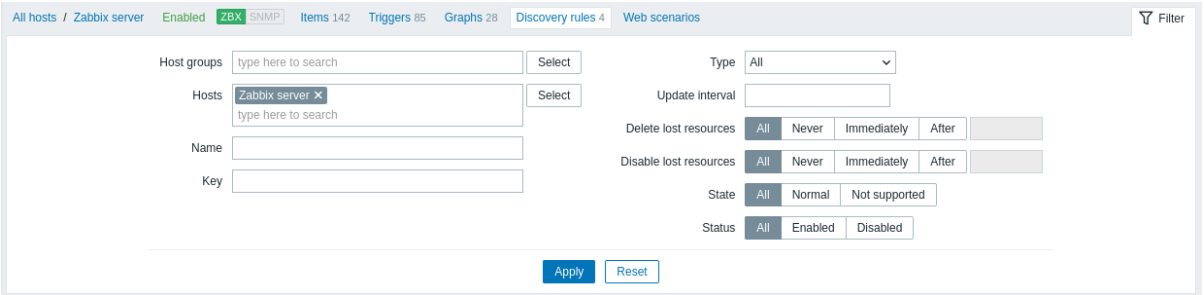
- *Iespējot* - mainīt zema līmeņa atklāšanas noteikuma statusu uz *Iespējots*.
- *Atspējot* - mainīt zema līmeņa atklāšanas noteikuma statusu uz *Atspējots*.
- *Izpildīt tagad* - nekavējoties veikt atklāšanu, pamatojoties uz atklāšanas noteikumiem. Skatiet sīkāku informāciju. Ņemiet vērā, ka, veicot atklāšanu nekavējoties, konfigurācijas kešatmiņa netiek atjaunināta, tāpēc rezultāts neatspoguļos pavisam nesenas izmaiņas atklāšanas noteikumu konfigurācijā.
- *Dzēst* - dzēst zema līmeņa atklāšanas noteikumus.

Lai izmantotu šīs opcijas, atzīmējiet izvēles rūtiņas pie attiecīgajiem atklāšanas noteikumiem, pēc tam noklikšķiniet uz vajadzīgās pogas.

Filtra izmantošana

Varat izmantot filtru, lai parādītu tikai tās atklāšanas noteikumus, kas jūs interesē. Lai uzlabotu meklēšanas veiktspēju, dati tiek meklēti ar neatrisinātiem makro.

Saite *Filter* 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.

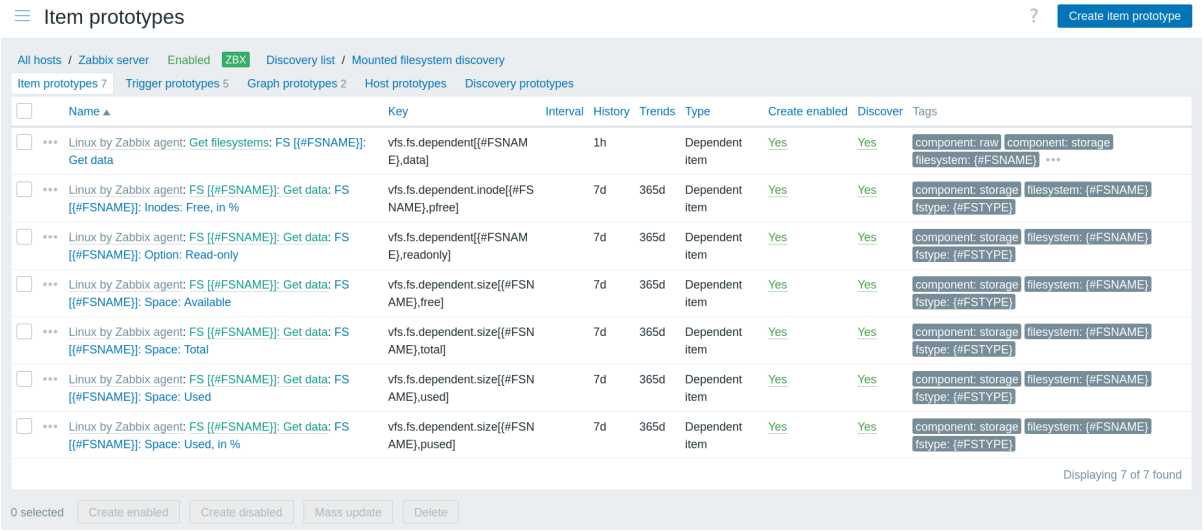


Parametrs	Apraksts
Host groups	Filtrēt pēc vienas vai vairākām hostu grupām. Norādot vecāko hostu grupu, netieši tiek atlasītas visas ligzdotās hostu grupas.
Hosts	Filtrēt pēc viena vai vairākiem hostiem.
Name	Filtrēt pēc atklāšanas noteikuma nosaukuma.
Key	Filtrēt pēc atklāšanas vienuma atslēgas.
Type	Filtrēt pēc atklāšanas vienuma tipa.
Update interval	Filtrēt pēc atjaunināšanas intervāla. Nav pieejams Zabbix trapper un atkarīgajiem vienumiem.
Delete lost resources	Filtrēt pēc *Delete lost resources* perioda.
Disable lost resources	Filtrēt pēc *Disable lost resources* perioda.
SNMP OID	Filtrēt pēc SNMP OID. Pieejams tikai tad, ja kā tips ir atlasīts *SNMP agent*.
State	Filtrēt pēc atklāšanas noteikuma stāvokļa (All/Normal/Not supported).
Status	Filtrēt pēc atklāšanas noteikuma statusa (All/Enabled/Disabled).

Vienumu prototipi

Pārskats

Šajā sadaļā tiek parādīti zema līmeņa atklāšanas noteikuma vienumu prototipi hostā. Vienumu prototipi ir pamats reālajiem hosta vienumiem, kas tiek izveidoti zema līmeņa atklāšanas laikā.



Parādītie dati:

Kolonna	Apraksts
Nosaukums	Vienuma prototipa nosaukums tiek parādīts kā zila saite. Noklikšķinot uz nosaukuma, tiek atvērta vienuma prototipa konfigurācijas veidlapa. Ja vienuma prototips pieder veidnei, pirms noteikuma nosaukuma kā pelēka saite tiek parādīts veidnes nosaukums. Noklikšķinot uz veidnes saites, tiek atvērts vienumu prototipu saraksts veidnes līmenī. Ja vienuma prototips ir tipa ”Atkarīgais vienums”, pirms vienuma prototipa nosaukuma zaļā krāsā tiek parādīts galvenā vienuma nosaukums.
Atslēga	Tiek parādīta vienuma prototipa atslēga.
Intervāls	Tiek parādīts pārbaudes biežums.
Vēsture	Tiek parādīts dienu skaits, cik ilgi saglabāt vienuma datu vēsturi.
Tendences	Tiek parādīts dienu skaits, cik ilgi saglabāt vienuma tendenču vēsturi.
Tips	Tiek parādīts vienuma prototipa tips (Zabbix aģents, SNMP aģents, vienkārša pārbaude u. c.).
Izveide iespējota	Vienums, kas izveidots, pamatojoties uz šo prototipu, ir: **Jā** - iespējots **Nē** - atspējots. Varat pārslēgties starp vērtībām “Jā” un “Nē”, noklikšķinot uz tām.

Kolonna	Apraksts
Atklāt	Vienums, kas atklāts, pamatojoties uz šo prototipu: **Jā** - atklāt **Nē** - neatklāt. Varat pārslēgties starp vērtībām “Jā” un “Nē”, noklikšķinot uz tām.
Tagi	Tiek parādīti vienuma prototipa tagi.

Lai konfigurētu jaunu vienuma prototipu, augšējā labajā stūrī noklikšķiniet uz pogas *Izveidot vienuma prototipu*.

Masveida rediģēšanas opcijas

Pogas zem saraksta piedāvā dažas masveida rediģēšanas opcijas:

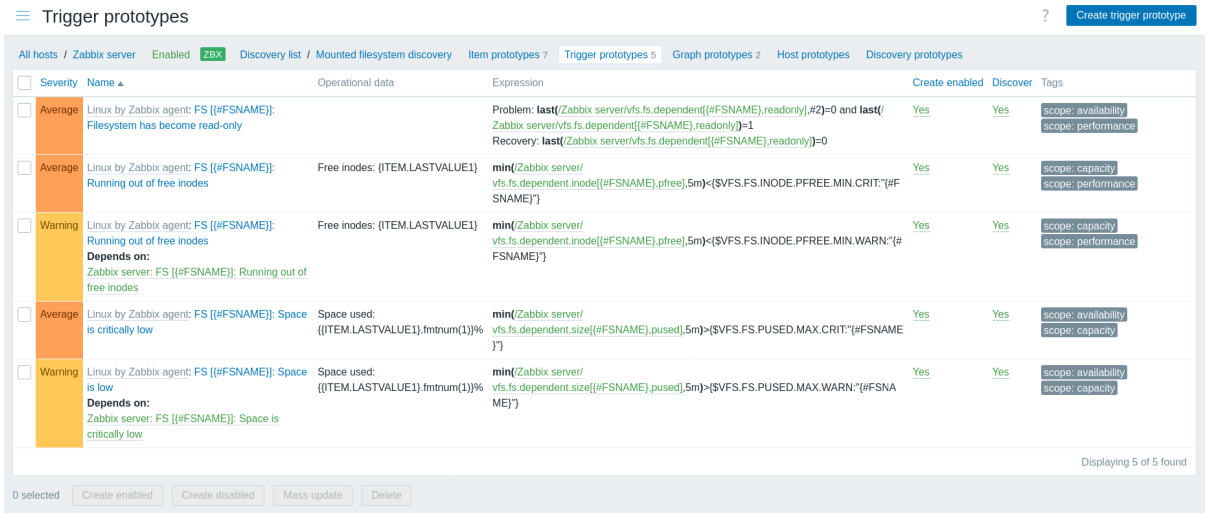
- *Izveidot iespējotus* - izveidot šos vienumus kā *Iespējotus*
- *Izveidot atspējotus* - izveidot šos vienumus kā *Atspējotus*
- *Masveida atjaunināšana* - masveidā atjaunināt šos vienumu prototipus
- *Dzēst* - dzēst šos vienumu prototipus

Lai izmantotu šīs opcijas, atzīmējiet izvēles rūtiņas attiecīgo vienumu prototipu priekšā, pēc tam noklikšķiniet uz vajadzīgās pogas.

Trigeru prototipi

Pārskats

Šajā sadaļā tiek parādīti zema līmeņa atklāšanas noteikuma trigeru prototipi hostā. Trigeru prototipi ir pamats reāliem hosta trigeriem, kas tiek izveidoti zema līmeņa atklāšanas laikā.



Parādītie dati:

Kolonna	Apraksts
Nosaukums	Trigeru prototipa nosaukums, kas tiek parādīts kā zila saite. Noklikšķinot uz nosaukuma, tiek atvērta trigeru prototipa konfigurācijas veidlapa. Ja trigeru prototips pieder saistītai veidnei, pirms trigera nosaukuma kā pelēka saite tiek parādīts veidnes nosaukums. Noklikšķinot uz veidnes saites, saistītās veidnes līmenī tiek atvērts trigeru prototipu saraksts.
Darbības dati	Tiek parādīts trigera darbības datu formāts, kas satur patvaļīgas virknes un makro, kuri tiks dinamiski aizstāti sadaļā *Monitoring* > *Problems*.
Izveide iespējota	Izveidot trigeri, pamatojoties uz šo prototipu: **Yes** - iespējots **No** - atspējots. Starp vērtībām “Yes” un “No” var pārslēgties, uz tām noklikšķinot.
Atklāt	Atklāt trigeri, pamatojoties uz šo prototipu: **Yes** - atklāt **No** - neatklāt. Starp vērtībām “Yes” un “No” var pārslēgties, uz tām noklikšķinot.
Tagi	Tiek parādīti trigeru prototipa tagi.

Lai konfigurētu jaunu trigeru prototipu, augšējā labajā stūrī noklikšķiniet uz pogas *Create trigger prototype*.

Masveida rediģēšanas opcijas

Pogas zem saraksta piedāvā dažas masveida rediģēšanas opcijas:

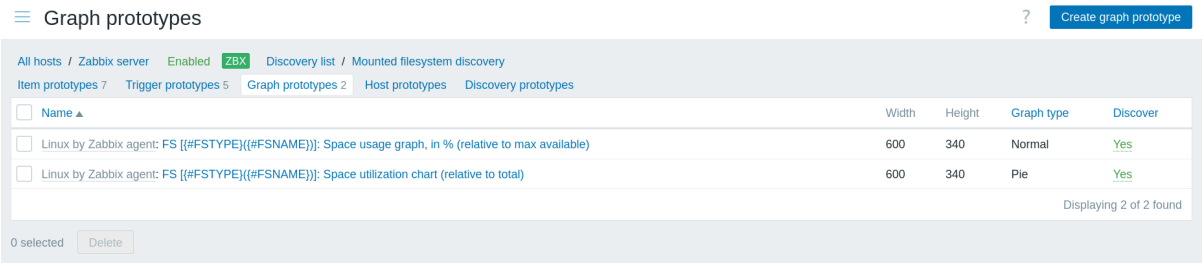
- *Izveidot iespējotus* - izveidot šos trigerus kā *Iespējotus*
- *Izveidot atspējotus* - izveidot šos trigerus kā *Atspējotus*
- *Masveida atjaunināšana* - masveidā atjaunināt šos trigeru prototipus
- *Dzēst* - dzēst šos trigeru prototipus

Lai izmantotu šīs opcijas, atzīmējiet izvēles rūtiņas attiecīgo trigeru prototipu priekšā, pēc tam noklikšķiniet uz vajadzīgās pogas.

Grafiku prototipi

Pārskats

Šajā sadaļā tiek parādīti zema līmeņa atklāšanas noteikuma grafiku prototipi hostā. Grafiku prototipi ir pamats reālajiem hosta grafikiem, kas tiek izveidoti zema līmeņa atklāšanas laikā.



Parādītie dati:

Column	Description
Name	Grafika prototipa nosaukums, kas tiek parādīts kā zila saite. Noklikšķinot uz nosaukuma, tiek atvērta grafika prototipa konfigurācijas veidlapa. Ja grafika prototips pieder saistītai veidnei, pirms grafika nosaukuma kā pelēka saite tiek parādīts veidnes nosaukums. Noklikšķinot uz veidnes saites, tiek atvērts grafiku prototipu saraksts saistītās veidnes līmenī.
Width	Tiek parādīts grafika prototipa platums.
Height	Tiek parādīts grafika prototipa augstums.
Type	Tiek parādīts 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:

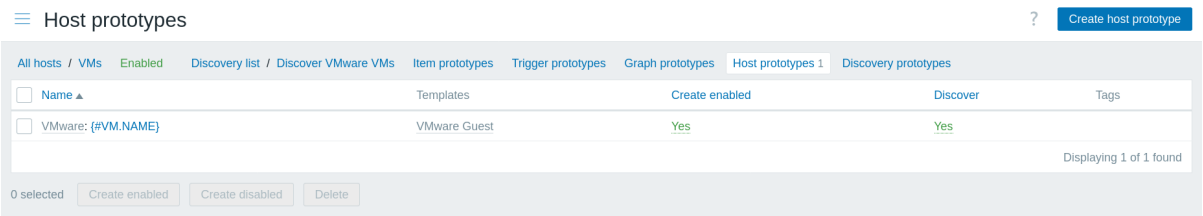
- *Dzēst* - dzēst šos grafiku prototipus

Lai izmantotu šīs opcijas, atzīmējiet izvēles rūtiņas pie attiecīgajiem grafiku prototipiem, pēc tam noklikšķiniet uz vajadzīgās pogas.

Hostu prototipi

Pārskats

Šajā sadaļā tiek parādīti zema līmeņa atklāšanas noteikuma hostu prototipi hostā. Hostu prototipi ir pamats reālajiem hostiem, kas tiek izveidoti zema līmeņa atklāšanas laikā.



Tiek parādīti šādi dati:

Kolonna	Apraksts
Name	Hosta prototipa nosaukums, kas tiek parādīts kā zila saite. Noklikšķinot uz nosaukuma, tiek atvērta hosta prototipa konfigurācijas forma. Ja hosta prototips pieder saistītai veidnei, pirms hosta nosaukuma kā pelēka saite tiek parādīts veidnes nosaukums. Noklikšķinot uz veidnes saites, saistītās veidnes līmenī tiek atvērts hosta prototipu saraksts.
Templates	Tiek parādītas hosta prototipa veidnes.
Create enabled	Izveidot hostu, pamatojoties uz šo prototipu: **Yes** - iespējots **No** - atspējots. Starp vērtībām “Yes” un “No” var pārslēgties, uz tām noklikšķinot.
Discover	Atklāt hostu, pamatojoties uz šo prototipu: **Yes** - atklāt **No** - neatklāt. Starp vērtībām “Yes” un “No” var pārslēgties, uz tām noklikšķinot.
Tags	Tiek parādītas 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:

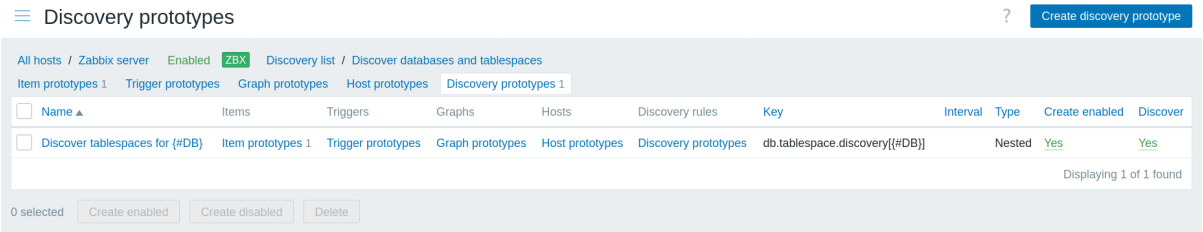
- *Izveidot iespējotus* - izveidot šos hostus kā *Iespējotus*
- *Izveidot atspējotus* - izveidot šos hostus kā *Atspējotus*
- *Dzēst* - dzēst šos hostu prototipus

Lai izmantotu šīs opcijas, atzīmējiet izvēles rūtiņas pie attiecīgajiem hostu prototipiem, pēc tam noklikšķiniet uz vajadzīgās pogas.

Atklāšanas prototipi

Pārskats

Šajā sadaļā tiek parādīti zema līmeņa atklāšanas noteikuma atklāšanas prototipi hostā. Atklāšanas prototipi ir vecākatklāšanas noteikuma ligzdotie atklāšanas noteikumi.



Parādītie dati:

Kolonna	Apraksts
Name	Atklāšanas prototipa nosaukums tiek parādīts kā zila saite. Noklikšķinot uz nosaukuma, tiek atvērta atklāšanas prototipa konfigurācijas veidlapa. Ja atklāšanas protips pieder saistītai veidnei, veidnes nosaukums pirms hosta nosaukuma tiek parādīts kā pelēka saite. Noklikšķinot uz veidnes saites, tiek atvērts atklāšanas prototipu saraksts saistītās veidnes līmenī.
Items	Tiek parādīta saite uz vienumu prototipu sarakstu. Pelēkā krāsā tiek parādīts esošo vienumu prototipu skaits.

Kolonna	Apraksts
Triggers	Tiek parādīta saite uz trigeru prototipu sarakstu. Pelēkā krāsā tiek parādīts esošo trigeru prototipu skaits.
Graphs	Tiek parādīta saite uz grafiku prototipu sarakstu. Pelēkā krāsā tiek parādīts esošo grafiku prototipu skaits.
Hosts	Tiek parādīta saite uz hostu prototipu sarakstu. Pelēkā krāsā tiek parādīts esošo hostu prototipu skaits.
Discovery rules	Tiek parādīta saite uz atklāšanas prototipu sarakstu. Pelēkā krāsā tiek parādīts esošo atklāšanas prototipu skaits.
Key	Tiek parādīta atklāšanai izmantotā vienuma atslēga.
Interval	Tiek parādīts atklāšanas veikšanas biežums. *Piezīme*: atklāšanu var veikt 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 prototipu izveides statuss — izveide iespējota (*Yes*) vai izveide atspējota (*No*). Noklikšķinot uz statusa, to var mainīt no Yes uz No (un otrādi).
Discover	Tiek parādīts atklāšanas prototipu atklāšanas statuss — atklāt (*Yes*) vai neatklāt (*No*). Noklikšķinot uz statusa, to var mainīt no Yes uz No (un otrādi).

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 iespējotu* - izveidot šos atklāšanas prototipus kā *Iespējotus*
- *Izveidot atspējotu* - izveidot šos atklāšanas prototipus kā *Atspējotus*
- *Dzēst* - dzēst šos atklāšanas prototipus

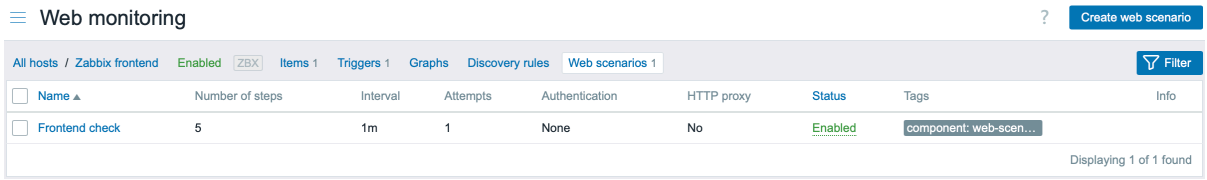
Lai izmantotu šīs opcijas, atzīmējiet izvēles rūtiņas pie attiecīgajiem atklāšanas prototipiem, pēc tam noklikšķiniet uz vajadzīgās pogas.

Tīmekļa scenāriji

Pārskats

Hosts tīmekļa scenāriju sarakstam var piekļūt, izvēloties *Data collection* > *Hosts* un noklikšķinot uz *Web* attiecīgajam hostam.

Tiek parādīts esošo tīmekļa scenāriju saraksts.



Parādītie dati:

Kolonna	Apraksts
Name	Tīmekļa scenārija nosaukums. Noklikšķinot uz tīmekļa scenārija nosaukuma, tiek atvērta tīmekļa scenārija konfigurācijas forma. Ja hosts tīmekļa scenārijs pieder veidnei, pirms tīmekļa scenārija nosaukuma kā pelēka saite tiek parādīts veidnes nosaukums. Noklikšķinot uz veidnes saites, tiks atvērts tīmekļa scenāriju saraksts veidnes līmenī.
Number of steps	Scenārijā ietverto darbību skaits.
Update interval	Cik bieži tiek izpildīts scenārijs.
Attempts	Mēģinājumu skaits tīmekļa scenārija darbību izpildei.
Authentication	Tiek parādīta autentifikācijas metode - Basic, NTLM, Kerberos, Digest vai None.
HTTP proxy	Tiek parādīts HTTP starpniekserveris vai “No”, ja tas netiek izmantots.
Status	Tiek parādīts tīmekļa scenārija statuss - *Enabled* vai *Disabled*. Noklikšķinot uz statusa, to var mainīt.
Tags	Tiek parādītas tīmekļa scenārija birkas. Var tikt parādītas ne vairāk kā trīs birkas (nosaukuma:vērtības pāri). Ja birku ir vairāk, tiek parādīta saite ”...”, kas, novietojot peles rādītāju virs tās, ļauj skatīt visas birkas.

Kolonna	Apraksts
Info	Ja viss darbojas pareizi, šajā kolonnā ikona netiek parādīta. Kļūdu gadījumā tiek parādīta kvadrātveida ikona ar burtu “i”. Novietojiet peles rādītāju virs ikonas, lai skatītu rīka padomu ar kļūdas aprakstu.

Lai konfigurētu jaunu tīmekļa scenāriju, augšējā labajā stūrī noklikšķiniet uz pogas *Create web scenario*.

Masveida rediģēšanas opcijas

Pogas zem saraksta piedāvā vairākas 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

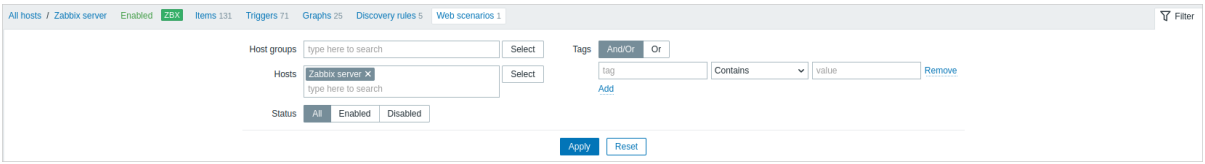
Lai izmantotu šīs opcijas, atzīmējiet izvēles rūtiņas pie attiecīgajiem tīmekļa scenārijiem, pēc tam noklikšķiniet uz vajadzīgās pogas.

Filtra izmantošana

Varat izmantot filtru, lai parādītu tikai tos scenārijus, kas jūs interesē.

Lai meklēšana būtu ātrāka, dati tiek meklēti ar neatrisinātiem makro.

Saite *Filter* ir pieejama virs tīmekļa scenāriju saraksta. Noklikšķinot uz tās, tiks atvērts filtrs, kurā varat norādīt vēlamos filtrēšanas kritērijus.



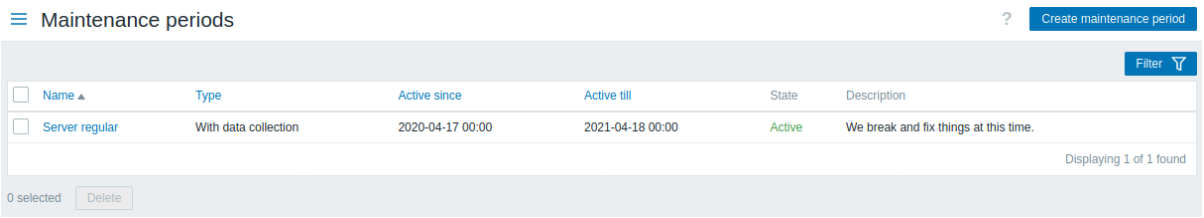
Parameter	Description
Host groups	Filtrēt pēc vienas vai vairākām hostu grupām. Norādot vecāko hostu grupu, netieši tiek atlasītas arī visas ligzdotās hostu grupas.
Hosts	Filtrēt pēc viena vai vairākiem hostiem. Ja iepriekš ir atlasītas hostu grupas, hostu atlase ir ierobežota līdz šīm grupām.
Status	Filtrēt pēc tīmekļa scenārija statusa.
Tags	Filtrēt pēc tīmekļa scenārija taga nosaukuma un vērtības. Iespējams iekļaut, kā arī izslēgt konkrētus 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, nav reģistrjutīga); **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, nav reģistrjutīga). Nosacījumiem ir pieejami divi aprēķina tipi: **And/Or** - jāizpildās visiem nosacījumiem, nosacījumi ar vienādu taga nosaukumu tiks grupēti pēc Or nosacījuma; **Or** - pietiek, ja ir izpildīts viens nosacījums.

Uzturēšana

Pārskats

Sadaļā *Datu apkopošana* > *Uzturēšana* lietotāji var konfigurēt un pārvaldīt uzturēšanas periodus hostiem.

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 veidlapa.
Tips	Tiek parādīts uzturēšanas tips: *Ar datu apkopošanu* vai *Bez datu apkopošanas*
Aktīvs no	Datums un laiks, kad izpildāmais uzturēšanas periods kļūst aktīvs. Piezīme: šis laiks neaktivizē uzturēšanas periodu; uzturēšanas periodi jāiestata atsevišķi.
Aktīvs līdz	Datums un laiks, kad izpildāmais uzturēšanas periods vairs nav aktīvs.
Stāvoklis	Uzturēšanas perioda stāvoklis: **Gaidāms** - drīzumā 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:

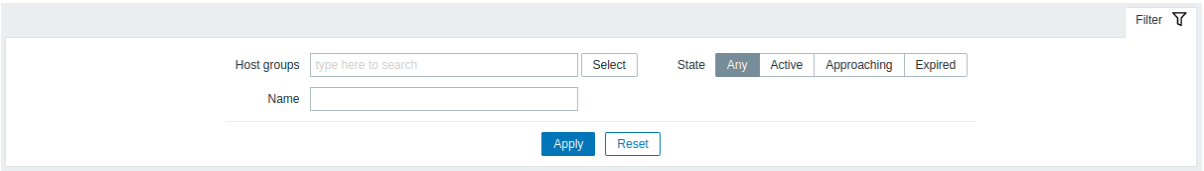
- *Dzēst* - dzēst uzturēšanas periodus

Lai izmantotu šo opciju, atzīmējiet izvēles rūtiņas pie attiecīgajiem uzturēšanas periodiem un noklikšķiniet uz *Dzēst*.

Filtra izmantošana

Varat izmantot filtru, lai parādītu tikai tos uzturēšanas periodus, kas jūs interesē. Lai uzlabotu meklēšanas veiktspēju, dati tiek meklēti ar neatrisinātiem makro.

Saite *Filter* ir pieejama virs uzturēšanas periodu saraksta. Ja uz tās noklikšķināsiet, būs pieejams filtrs, kurā varat filtrēt uzturēšanas periodus pēc hostu grupas, nosaukuma un stāvokļa.



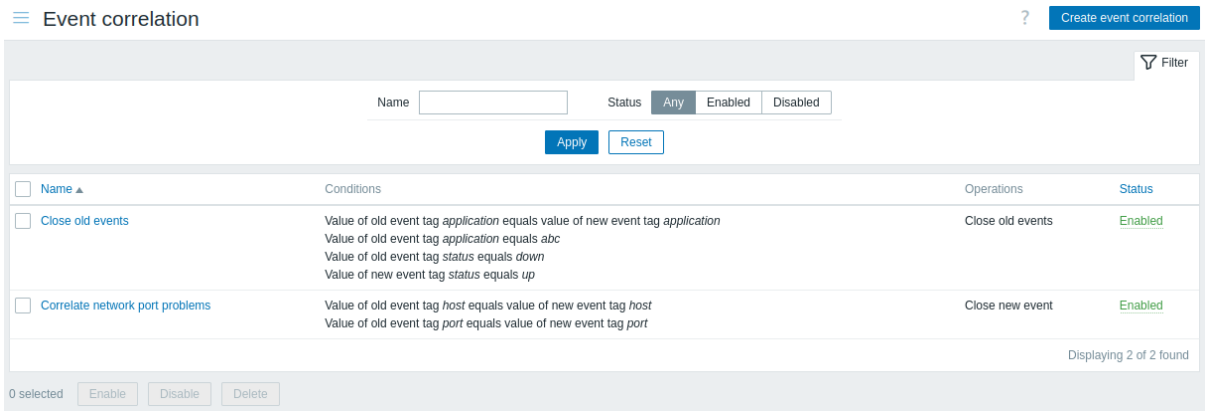
Rindu aprēķins apkopes laikā

Attention:
Zabbix starpniekserveris nav informēts par apkopes periodiem; sīkāku informāciju skatiet Rindu aprēķins apkopes laikā.

Notikumu korelācija

Pārskats

Sadaļā *Datu apkopošana* > *Notikumu korelācija* lietotāji var konfigurēt un uzturēt globālās korelācijas kārtulas Zabbix notikumiem.



Parādītie dati:

Kolonna	Apraksts
Nosaukums	Korelācijas kārtulas nosaukums. Noklikšķinot uz korelācijas kārtulas nosaukuma, tiek atvērta notikumu korelācijas konfigurācijas veidlapa.
Nosacījumi	Tiek parādīti korelācijas kārtulas nosacījumi.
Darbības	Tiek parādītas korelācijas kārtulas darbības.
Statuss	Tiek parādīts korelācijas kārtulas statuss - *Iespējota* vai *Atspējota*. Noklikšķinot uz statusa, to var mainīt.

Lai konfigurētu jaunu korelācijas kārtulu, augšējā labajā stūrī noklikšķiniet uz pogas *Izveidot notikumu korelāciju*.

Masveida rediģēšanas opcijas

Pogas zem saraksta piedāvā vairākas masveida rediģēšanas opcijas:

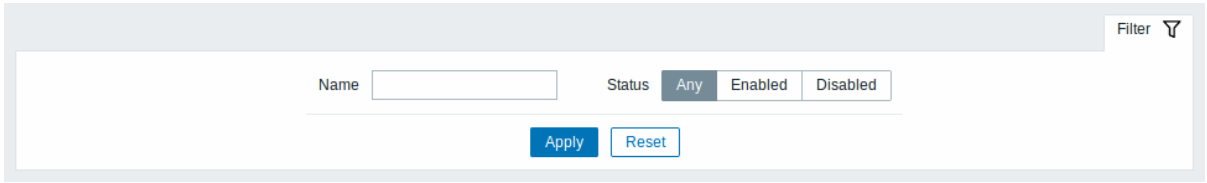
- *Iespējot* - mainīt korelācijas noteikuma statusu uz *Iespējots*
- *Atspējot* - mainīt korelācijas noteikuma statusu uz *Atspējots*
- *Dzēst* - dzēst korelācijas noteikumus

Lai izmantotu šīs opcijas, atzīmējiet izvēles rūtiņas pie attiecīgajiem korelācijas noteikumiem un pēc tam noklikšķiniet uz vajadzīgās pogas.

Filtra izmantošana

Varat izmantot filtru, lai parādītu tikai tās korelācijas noteikumus, kas jūs interesē. Lai uzlabotu meklēšanas veiktspēju, dati tiek meklēti ar neatrisinātiem makro.

Saite *Filter* ir pieejama virs korelācijas noteikumu saraksta. Ja uz tās noklikšķināsiet, kļūs pieejams filtrs, kurā varat filtrēt korelācijas noteikumus pēc nosaukuma un statusa.

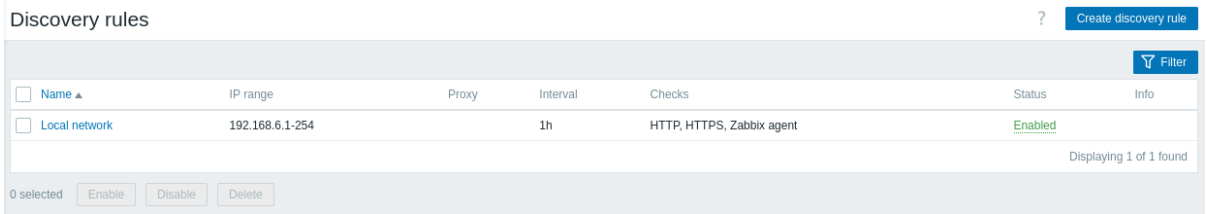


Atklāšana

Pārskats

Sadaļā *Datu apkopošana* > *Atklāšana* lietotāji var konfigurēt un pārvaldīt atklāšanas noteikumus.

Tiek parādīts esošo atklāšanas noteikumu saraksts ar to detalizētu informāciju.



Parādītie dati:

Kolonna	Apraksts
Nosaukums	Atklāšanas noteikuma nosaukums. Noklikšķinot uz atklāšanas noteikuma nosaukuma, tiek atvērta atklāšanas noteikuma konfigurācijas veidlapa.
IP diapazons	Tiek parādīts tīkla skenēšanai izmantojamais IP adrešu diapazons.
Starpniekserveris	Ja atklāšanu veic starpniekserveris, tiek parādīts tā nosaukums.
Intervāls	Tiek parādīts atklāšanas veikšanas biežums.
Pārbaudes	Tiek parādīti atklāšanai izmantoto pārbaužu veidi.
Statuss	Tiek parādīts atklāšanas noteikuma 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". Novietojiet kursoru virs ikonas, lai skatītu rīka padomu ar kļūdas aprakstu.

Lai konfigurētu jaunu atklāšanas noteikumu, augšējā labajā stūrī noklikšķiniet uz pogas *Izveidot atklāšanas noteikumu*.

Masveida rediģēšanas opcijas

Pogas zem saraksta piedāvā vairākas masveida rediģēšanas opcijas:

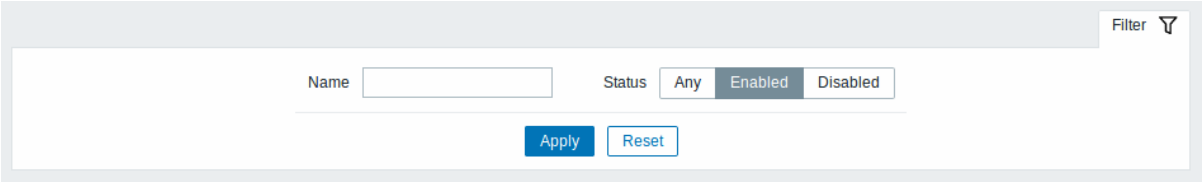
- *Iespējot* - maina atklāšanas noteikuma statusu uz *Iespējots*
- *Atspējot* - maina atklāšanas noteikuma statusu uz *Atspējots*
- *Dzēst* - dzēš atklāšanas noteikumus

Lai izmantotu šīs opcijas, atzīmējiet izvēles rūtiņas attiecīgo atklāšanas noteikumu priekšā un pēc tam noklikšķiniet uz vajadzīgās pogas.

Filtra izmantošana

Varat izmantot filtru, lai parādītu tikai tos atklāšanas noteikumus, kas jūs interesē. Lai nodrošinātu labāku meklēšanas veiktspēju, dati tiek meklēti ar neatrisinātiem makro.

Saite *Filtrs* ir pieejama virs atklāšanas noteikumu saraksta. Ja uz tās noklikšķināsiet, būs pieejams filtrs, kurā varat filtrēt atklāšanas noteikumus pēc nosaukuma un statusa.



Brīdinājumi

Pārskats

Šajā izvēlnē ir pieejamas sadaļas, kas saistītas ar brīdinājumu konfigurēšanu Zabbix.

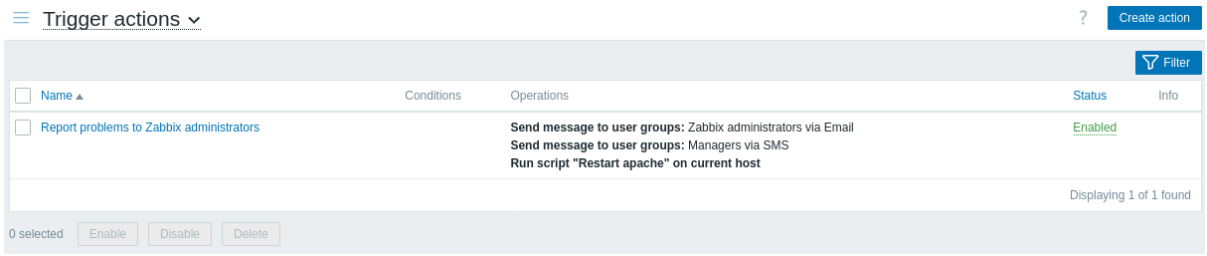
Darbības

Pārskats

Sadaļā *Brīdinājumi* > *Darbības* lietotāji var konfigurēt un pārvaldīt darbības.

Tiek parādītas darbības, kas piešķirtas atlasītajam notikumu avotam (trigeriem, pakalpojumiem, atklāšanai, automātiskai reģistrācijai, iekšējām darbībām).

Lai pārslēgtos uz citu notikumu avotu, izvēlnes sadaļā *Brīdinājumi* noklikšķiniet uz *Darbības*. Starp avotiem var pārslēgties arī, izmantojot virsraksta nolaižamo sarakstu augšējā kreisajā stūrī.



Parādītie dati:

Kolonna	Apraksts
Nosaukums	Darbības nosaukums. Noklikšķinot uz darbības nosaukuma, tiek atvērta darbības konfigurācijas veidlapa.
Nosacījumi	Tiek parādīti darbības nosacījumi.
Darbības	Tiek parādītas darbības operācijas. Operāciju sarakstā tiek parādīts arī paziņojumiem izmantotais multivides 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 paziņojums vai attāla komanda.
Statuss	Tiek parādīts darbības statuss - *Iespējota* vai *Atspējota*. Noklikšķinot uz statusa, to var mainīt. Skatiet sadaļu Eskalācijas, lai iegūtu papildinformāciju par to, kas notiek, ja darbība tiek atspējota notiekošas eskalācijas laikā.
Informācija	Ja viss darbojas pareizi, šajā kolonnā ikona netiek parādīta. Kļūdu gadījumā, piemēram, ja pēc jaunināšanas vai zema līmeņa atklāšanas trūkst darbības operāciju vai nosacījumu, tiek parādīta brīdinājuma ikona. Novietojiet kursoru virs ikonas, lai skatītu rīka padomu ar kļūdas aprakstu.

Lai konfigurētu jaunu darbību, augšējā labajā stūrī noklikšķiniet uz pogas *Izveidot darbību*.

Lietotājiem bez superadministratora tiesībām darbības tiek parādītas atbilstoši atļauju iestatījumiem. Tas nozīmē, ka dažos gadījumos lietotājs bez superadministratora tiesībām noteiktu atļauju ierobežojumu dēļ nevar skatīt pilnu darbību sarakstu. Darbība lietotājam bez superadministratora tiesībām tiek parādīta, ja ir izpildīti šādi nosacījumi:

- Lietotājam ir lasīšanas un rakstīšanas piekļuve hosts grupām, hostiem, veidnēm un trigeriem darbības nosacījumos
- Lietotājam ir lasīšanas un rakstīšanas piekļuve hosts 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:

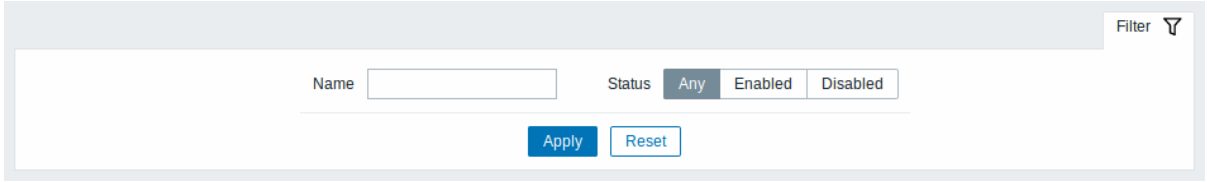
- *Iespējot* - mainīt darbības statusu uz *Iespējots*
- *Atspējot* - mainīt darbības statusu uz *Atspējots*
- *Dzēst* - dzēst darbības

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.



Multivides tipi

Pārskats

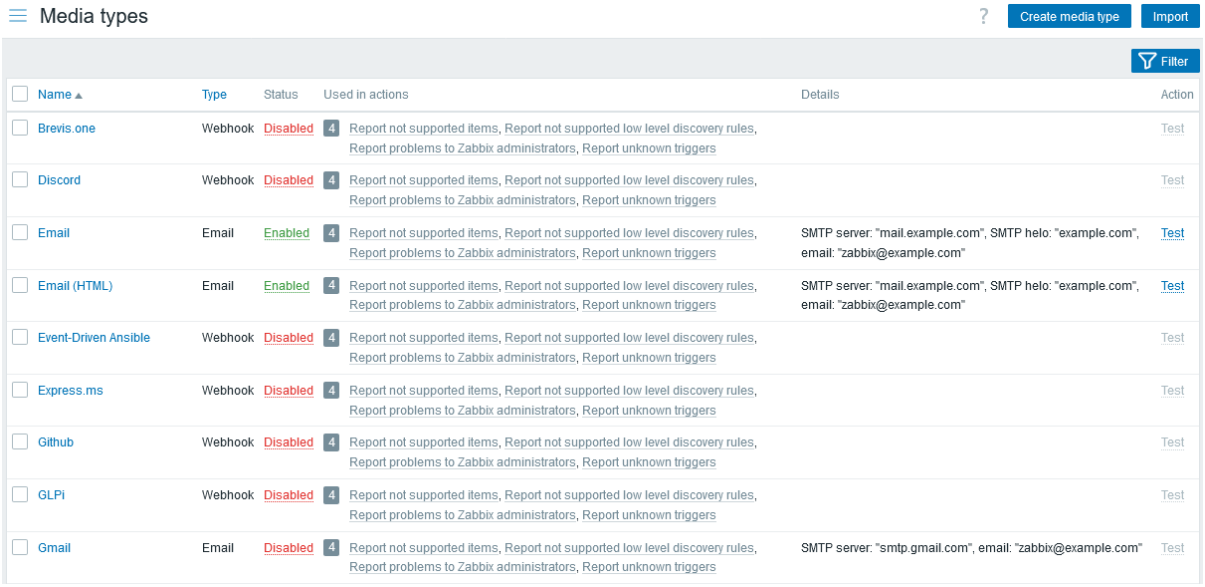
Sadaļā *Brīdinājumi* > *Multivides veidi* lietotāji var konfigurēt un uzturēt informāciju par multivides veidiem.

Šī izvēlne ir pieejama tikai lietotājiem ar lietotāja tipu SuperAdmin.

Informācija par multivides veidu ietver vispārīgus norādījumus par multivides izmantošanu kā paziņojumu piegādes kanālu. Konkrēta informācija, piemēram, atsevišķas e-pasta adreses, uz kurām jānosūta paziņojums, tiek glabāta pie atsevišķiem lietotājiem.

Tiek parādīts esošo multivides veidu saraksts ar to detalizētu informāciju.

Note:
Ja `$ZBX_FEATURE_FLAGS['media_type_denylist']` funkcionalitātes karodziņā lietotāja saskarnes konfigurācijas failā ir norādītas piegādes metodes (*E-pasts*, *SMS*, *Webhook*, *Skripts* vai *Push*), atbilstošie iepriekš konfigurētie multivides veidi šajā sadaļā tiek atspējoti.



Parādītie dati:

Kolonna	Apraksts
Nosaukums	Multivides veida nosaukums. Noklikšķinot uz nosaukuma, tiek atvērta multivides veida konfigurācijas veidlapa.
Tips	Tiek parādīts multivides veids (e-pasts, SMS u. c.).
Statuss	Tiek parādīts multivides veida statuss - *Iespējots* vai *Atspējots*. Noklikšķinot uz statusa, to var mainīt.
Izmantots darbībās	Tiek parādītas darbības, kurās tiek izmantots multivides veids, pirms tām norādot šo darbību kopējo skaitu. Noklikšķinot uz darbības nosaukuma, tiek atvērta darbības konfigurācijas veidlapa. Ja lietotājam nav atļauju darbībai, uz nosaukuma nevar noklikšķināt.
Detalizēta informācija	Tiek parādīta detalizēta informācija par multivides veidu.
Darbības	Ir pieejama šāda darbība: **Testēt** - noklikšķiniet, lai atvērtu testēšanas veidlapu, kurā varat ievadīt multivides veida parametrus (piemēram, saņēmēja adresi, testa tēmu un pamattekstu) un nosūtīt testa ziņojumu, lai pārbaudītu, vai konfigurētais multivides veids darbojas. Skatiet arī: Multivides veida testēšana.

Lai konfigurētu jaunu multivides veidu, augšējā labajā stūrī noklikšķiniet uz pogas *Izveidot multivides veidu*.

Lai importētu multivides veidu, augšējā labajā stūrī noklikšķiniet uz pogas *Importēt*.

Masveida rediģēšanas opcijas

Pogas zem saraksta piedāvā dažas masveida rediģēšanas opcijas:

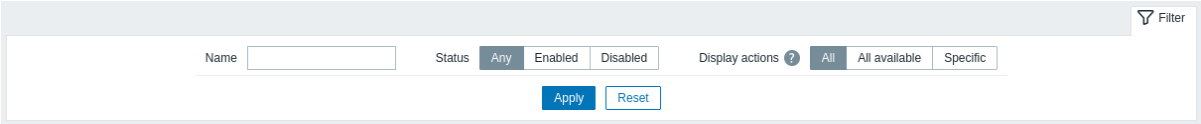
- *Iespējot* - mainīt multivides tipa statusu uz *Iespējots*
- *Atspējot* - mainīt multivides tipa statusu uz *Atspējots*
- *Eksportēt* - eksportēt multivides tipus YAML, XML vai JSON failā
- *Dzēst* - dzēst multivides tipus

Lai izmantotu šīs opcijas, atzīmējiet izvēles rūtiņas pirms attiecīgajiem multivides tipiem un pēc tam noklikšķiniet uz vajadzīgās pogas.

Filtra izmantošana

Varat izmantot filtru, lai parādītu tikai tos mediju tipus, kas jūs interesē. Lai uzlabotu meklēšanas veiktspēju, dati tiek meklēti ar neatrisinātiem makro.

Saite *Filter* ir pieejama virs mediju tipu saraksta. Noklikšķinot uz tās, kļūst pieejams filtrs, kurā varat filtrēt mediju tipus pēc nosaukuma un statusa. Turklāt varat izmantot filtru, lai kolonnā *Used in actions* parādītu darbības atbilstoši to mediju tipa izmantošanas tvērumam (noteiktam ar parametru *Send to media type* darbības operācijas detaļās).



Skripti

Pārskats

Sadaļā *Brīdinājumi* > *Skripti* var konfigurēt un uzturēt lietotāja definētus globālos skriptus.

Šī izvēlne ir pieejama tikai lietotājiem ar lietotāja tipu SuperAdmin.

Globālie skripti atkarībā no konfigurētā tvēruma un lietotāja atļaujām ir pieejami izpildei:

- No hosts izvēlnes dažādās lietotāja saskarnes vietās (*Informācijas panelis*, *Problēmas*, *Jaunākie dati*, *Kartes* u. c.).
- No notikuma izvēlnes.
- Tos var palaist kā darbības operāciju.

Skriptus izpilda Zabbix aģents, Zabbix serveris (starpniekserveris) vai tikai Zabbix serveris. Skatiet arī sadaļu Komandu izpilde.

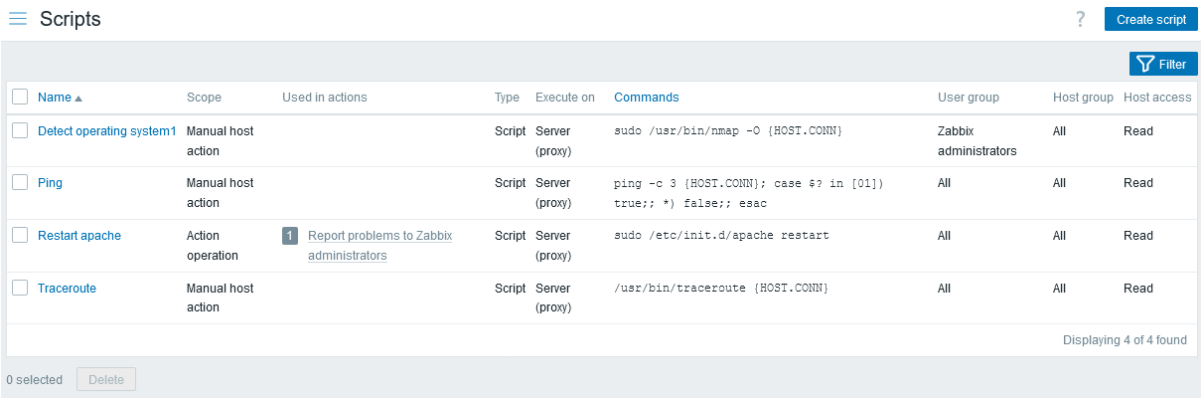
Attālās komandas pēc noklusējuma ir atspējotas gan Zabbix aģentā, gan Zabbix starpniekserverī. Tās var iespējot šādi:

- Attālajām komandām, ko izpilda Zabbix aģents, aģenta konfigurācijā katrai atļautajai komandai pievienojiet parametru `AllowKey=system.run[<command>,*]` vai `AllowKeyRegexp=^system\.run\[<command>,.*\]$`, kur `*` atbilst jebkuram izpildes režīmam (`wait` vai `nowait`).
- Attālajām komandām, ko izpilda Zabbix starpniekserveris, starpniekservera konfigurācijā iestatiet parametru `EnableRemoteCommands` uz 1. Ņemiet vērā, ka attālās komandas nav jāiespējo Zabbix starpniekserverī, ja tās izpilda Zabbix aģents uz hosts, ko uzrauga starpniekserveris.

Attention:
Neizmantojiet `AllowKey=system.run[*]` vai `AllowKeyRegexp=^system\.run\[.*\]$`, ja vien tas nav nepieciešams, jo tādējādi tiek atļauts izpildīt jebkuru komandu uz hosts. Drošības apsvērumu dēļ ieteicams atļaut tikai precīzi norādītas komandas, savukārt `system.run[*]` pēc noklusējuma ir aizliegts (`DenyKey=system.run[*]`).

Globālo skriptu izpildi Zabbix serverī var atspējot, servera konfigurācijā iestatot `EnableGlobalScripts=0`. Jaunās instalācijās, sākot ar Zabbix 7.0, globālo skriptu izpilde Zabbix serverī pēc noklusējuma ir atspējota.

Tiek parādīts esošo skriptu saraksts ar to detalizētu informāciju.



Parādītie dati:

Kolonna	Apraksts
Nosaukums	Skripta nosaukums. Noklikšķinot uz skripta nosaukuma, tiek atvērta skripta konfigurācijas veidlapa.
Tvērums	Skripta tvērums - darbības operācija, manuāla hosts darbība vai manuāla notikuma darbība. Šis iestatījums nosaka, kur skripts ir pieejams - darbības operāciju attālajās komandās, hosts izvēlnē vai notikuma izvēlnē.
Izmantots darbībās	Tiek parādītas visas darbības, kurās skripts tiek izmantots, pirms tām norādot šo darbību kopējo skaitu. Noklikšķinot uz darbības nosaukuma, tiek atvērta darbības konfigurācijas veidlapa. Ja lietotājam nav atļauju darbībai, nosaukums nav noklikšķināms.
Tips	Tiek parādīts skripta tips - *URL*, *Webhook*, *Skripts*, *SSH*, *Telnet* vai *IPMI* komanda.
Izpildīt uz	Tiek parādīts, vai skripts tiks izpildīts Zabbix aģentā, Zabbix starpniekserverī vai serverī, vai tikai Zabbix serverī.
Komandas	Tiek parādītas visas skripta ietvaros izpildāmās komandas. Webhook gadījumā šeit nekas netiek parādīts.
Lietotāju grupa	Tiek parādīta lietotāju grupa, kurai skripts ir pieejams (vai *Visi*, ja tas ir pieejams visām lietotāju grupām).
Hosts grupa	Tiek parādīta hosts grupa, kurai skripts ir pieejams (vai *Visas*, ja tas ir pieejams visām hosts grupām).
Piekļuve hosts	Tiek parādīts hosts grupai nepieciešamais atļauju līmenis - *Lasīšana* vai *Rakstīšana*. Skripta izpildei varēs piekļūt tikai lietotāji ar nepieciešamo atļauju līmeni.

Lai konfigurētu jaunu skriptu, augšējā labajā stūrī noklikšķiniet uz pogas *Izveidot skriptu*.

Masveida rediģēšanas opcijas

Poga zem saraksta piedāvā vienu masveida rediģēšanas opciju:

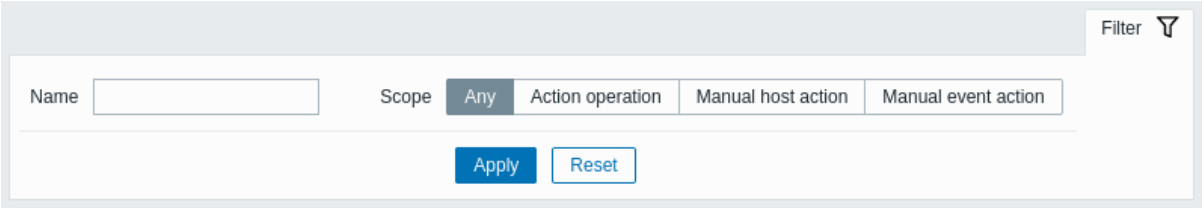
- *Dzēst* - dzēst skriptus

Lai izmantotu šo opciju, atzīmējiet izvēles rūtiņas pie attiecīgajiem skriptiem un noklikšķiniet uz *Dzēst*.

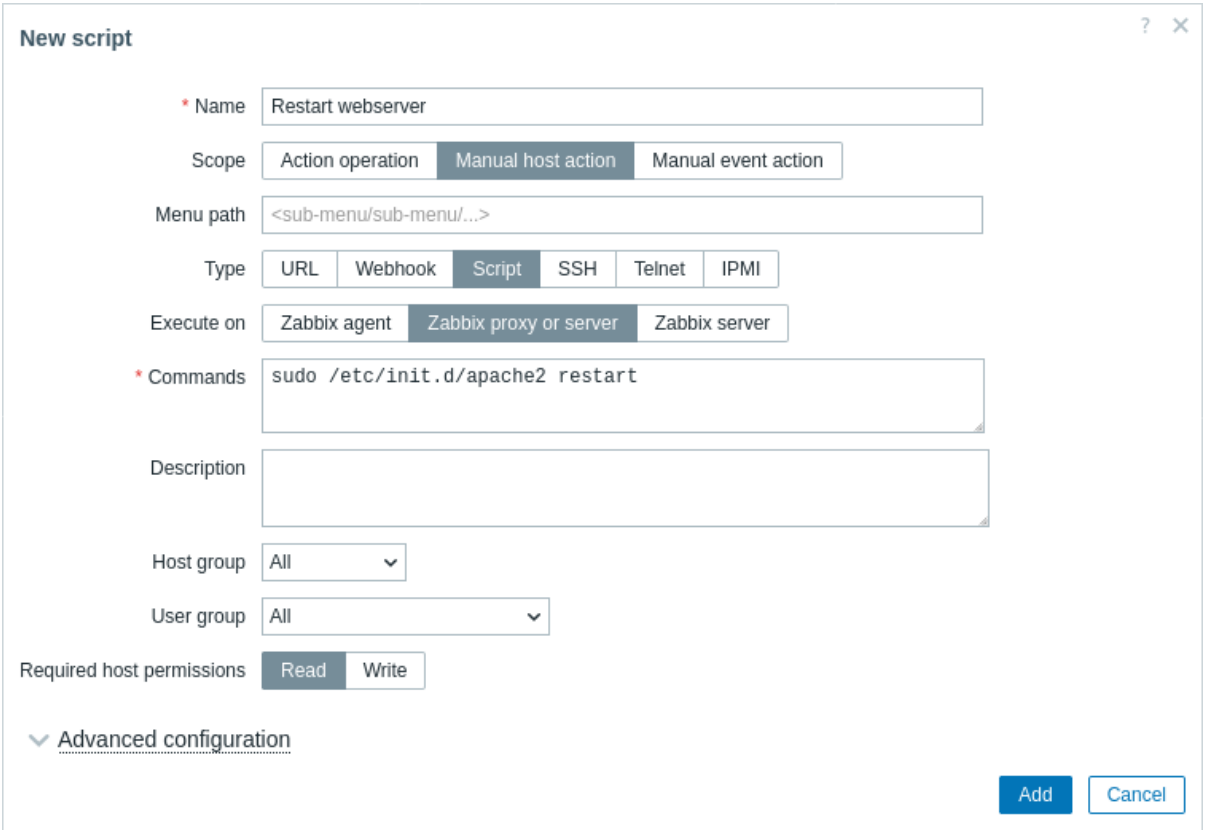
Filtra izmantošana

Varat izmantot filtru, lai parādītu tikai tos skriptus, kas jūs interesē. Lai uzlabotu meklēšanas veiktspēju, dati tiek meklēti ar neatrisinātiem makro.

Saite *Filtrs* ir pieejama virs skriptu saraksta. Ja uz tās noklikšķināsiet, kļūs pieejams filtrs, kurā varat filtrēt skriptus pēc nosaukuma un tvēruma.



Globāla skripta konfigurēšana



Note:
Lai uzlabotu drošību, globālajos skriptos vienkāršu makro vietā ieteicams izmantot makro funkcijas, jo makro netiek automātiski ekranēti.

Skripta atribūti:

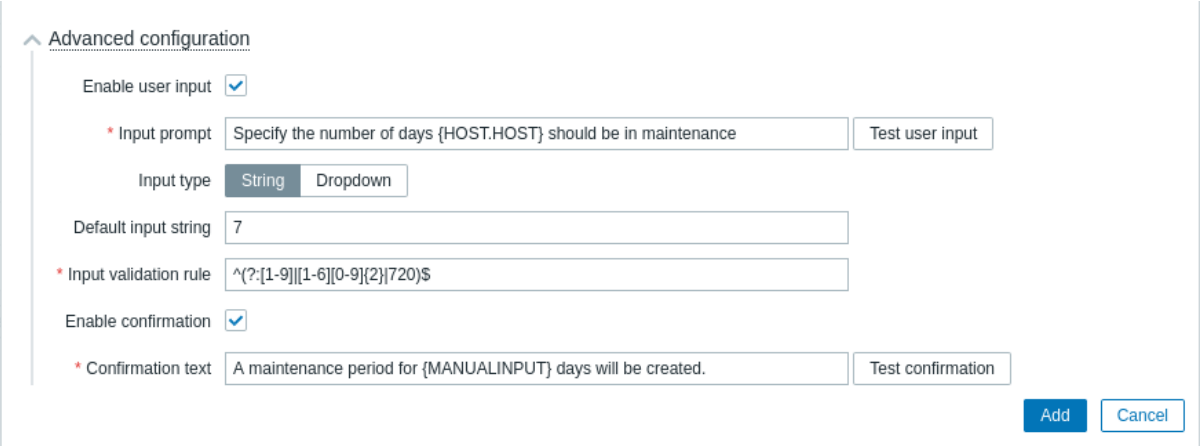
Parametrs	Apraksts
Nosaukums	Unikāls skripta nosaukums. Piemēram, `Clear /tmp filesystem`
Tvērums	Skripta tvērums - darbības operācija, manuāla darbība ar hostu vai manuāla darbība ar notikumu. Šis iestatījums nosaka, kur skriptu var izmantot - darbības operāciju attālajās komandās, hosta izvēlnē vai notikuma izvēlnē. Iestatot tvērumu uz "Darbības operācija", skripts kļūst pieejams visiem lietotājiem, kuriem ir piekļuve sadaļai *Brīdinājumi* > *Darbības*. Ja skripts faktiski tiek izmantots darbībā, tā tvērumu nevar mainīt no "darbības operācija". **Makro atbalsts** Tvērums ietekmē pieejamo makro diapazonu. Piemēram, skriptos tiek atbalstīti ar lietotāju saistīti makro (`{USER.*}`), lai varētu nodot informāciju par lietotāju, kurš palaida skriptu. Tomēr tie netiek atbalstīti, ja skripta tvērums ir darbības operācija, jo darbības operācijas tiek izpildītas automātiski. Makro `{MANUALINPUT}` ļauj skripta izpildes laikā norādīt manuālu ievadi. Tas tiek atbalstīts manuālas darbības ar hostu un manuālas darbības ar notikumu skriptos. Lai noskaidrotu, kuri citi makro tiek atbalstīti, atbalstīto makro tabulā meklējiet "Trigger-based notifications and commands/Trigger-based commands", "Manual host action scripts" un "Manual event action scripts". Ņemiet vērā: ja makro var tikt aizstāts ar vērtību, kurā ir atstarpes (piemēram, hosta nosaukums), neaizmirstiet to nepieciešamības gadījumā ievietot pēdiņās.
Izvēlnes ceļš	Vēlamais skripta izvēlnes ceļš. Piemēram, `Default` vai `Default/` skriptu parādīs attiecīgajā direktorijā. Izvēlnes var būt ligzdotas, piemēram, `Main menu/Sub menu1/Sub menu2`. Piekļūstot skriptiem, izmantojot uzraudzības sadaļu hosta/notikuma izvēlni, tie tiks sakārtoti atbilstoši norādītajām direktorijām. Šis lauks tiek parādīts tikai tad, ja *Tvērums* ir iestatīts uz "Manuāla darbība ar hostu" vai "Manuāla darbība ar notikumu".

Parametrs	Apraksts
Tips	Noklikšķiniet uz attiecīgās pogas, lai atlasītu skripta tipu: **URL**, **Webhook**, **Skripts**, **SSH**, **Telnet** vai **IPMI** komanda. Tips **URL** ir pieejams tikai tad, ja *Tvērums* ir iestatīts uz "Manuāla darbība ar hostu" vai "Manuāla darbība ar notikumu".
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 iepriekš sadaļu *Tvērums*). 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 kodēt URL formātā.
Atvērt jaunā logā	Nosaka, vai URL jāatver jaunā vai tajā pašā pārlūkprogrammas cilnē.
Skripta tips: **Webhook**	
Parametri	Norādiet webhook mainīgos kā atribūtu un vērtību pārus. Skatiet arī: Webhook multivides konfigurācija. Parametru vērtībās tiek atbalstīti makro un pielāgoti lietotāja makro. Makro atbalsts ir atkarīgs no skripta tvēruma (skatiet iepriekš sadaļu *Tvērums*).
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 iepriekš sadaļu *Tvērums*). Skatiet arī: Webhook multivides konfigurācija, Papildu Javascript objekti.
Noildze	JavaScript izpildes noildze (1-60 s, noklusējums: 30 s). Tiek atbalstīti laika sufiksi (piemēram, `30s`, `1m`).
Skripta tips: **Skripts**	
Izpildīt uz	Noklikšķiniet uz attiecīgās pogas, lai čaulas skriptu izpildītu: **Zabbix aģents** - skriptu izpildīs Zabbix aģents (ja vienums system.run ir atļauts) uz hosta **Zabbix starpniekserveris vai serveris** - skriptu izpildīs Zabbix starpniekserveris vai serveris atkarībā no tā, vai hostu uzrauga starpniekserveris vai serveris. Tas tiks izpildīts uz starpniekservera, ja to iespējo EnableRemoteCommands. Tas tiks izpildīts uz servera, ja globālos skriptus iespējo servera parametrs EnableGlobalScripts. **Zabbix serveris** - skriptu izpildīs tikai Zabbix serveris. Šī opcija nebūs pieejama, ja globālos skriptus atspējo servera parametrs EnableGlobalScripts.
Komandas	Ievadiet skriptā izpildāmo komandu pilno ceļu. Makro atbalsts ir atkarīgs no skripta tvēruma (skatiet iepriekš sadaļu *Tvērums*). Tiek atbalstīti pielāgoti lietotāja makro.
Skripta tips: **SSH**	
Autentifikācijas metode	Atlasiet autentifikācijas metodi - paroli vai publisko atslēgu.
Lietotājvārds	Ievadiet lietotājvārdu.
Parole	Ievadiet paroli. Šis lauks ir pieejams, ja kā autentifikācijas metode ir atlasīta "Parole".
Publiskās atslēgas fails	Ievadiet ceļu uz publiskās atslēgas failu. Šis lauks ir pieejams, ja kā autentifikācijas metode ir atlasīta "Publiskā atslēga".
Privātās atslēgas fails	Ievadiet ceļu uz privātās atslēgas failu. Šis lauks ir pieejams, ja kā autentifikācijas metode ir atlasīta "Publiskā atslēga".
Parole	Ievadiet paroles frāzi. Šis lauks ir pieejams, ja kā autentifikācijas metode ir atlasīta "Publiskā atslēga".
Ports	Ievadiet attālā SSH pakalpojuma portu mērķa hostā, ar kuru izveidos savienojumu Zabbix.
Komandas	Ievadiet komandas. Makro atbalsts ir atkarīgs no skripta tvēruma (skatiet iepriekš sadaļu *Tvērums*). Tiek atbalstīti pielāgoti lietotāja makro.
Skripta tips: **Telnet**	
Lietotājvārds	Ievadiet lietotājvārdu.
Parole	Ievadiet paroli.
Ports	Ievadiet attālā Telnet pakalpojuma portu mērķa hostā, ar kuru izveidos savienojumu Zabbix.
Komandas	Ievadiet komandas. Makro atbalsts ir atkarīgs no skripta tvēruma (skatiet iepriekš sadaļu *Tvērums*). Tiek atbalstīti pielāgoti lietotāja makro.
Skripta tips: **IPMI**	

Parametrs	Apraksts
Komanda	Ievadiet IPMI komandu. Makro atbalsts ir atkarīgs no skripta tvēruma (skatiet iepriekš sadaļu *Tvērums*). Tiek atbalstīti pielāgoti lietotāja makro.
Apraksts	Ievadiet skripta aprakstu.
Hostu grupa	Atlasiet hostu grupu, kurai skripts būs pieejams (vai *Visas*, lai tas būtu pieejams visām hostu grupām).
Lietotāju grupa	Atlasiet lietotāju grupu, kurai skripts būs pieejams (vai *Visas*, lai tas būtu pieejams visām lietotāju grupām). Šis lauks tiek parādīts tikai tad, ja *Tvērums* ir iestatīts uz "Manuāla darbība ar hostu" vai "Manuāla darbība ar notikumu".
Nepieciešamās atļaujas hostam	Atlasiet hostu grupas atļauju līmeni - *Lasīšana* vai *Rakstīšana*. Skripta izpildei varēs piekļūt tikai lietotāji ar nepieciešamo atļauju līmeni. Šis lauks tiek parādīts tikai tad, ja *Tvērums* ir iestatīts uz "Manuāla darbība ar hostu" vai "Manuāla darbība ar notikumu".
Papildu konfigurācija	Noklikšķiniet uz virsraksta *Papildu konfigurācija*, lai parādītu papildu konfigurācijas opcijas. Šis lauks tiek parādīts tikai tad, ja *Tvērums* ir iestatīts uz "Manuāla darbība ar hostu" vai "Manuāla darbība ar notikumu".

Papildu konfigurācija

Papildu konfigurācijas opcijas ir pieejamas izvēršamā 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: Manuāla lietotāja ievade.
Ievades uzvedne	Ievadiet pielāgotu tekstu, kas pieprasa lietotāja ievadi. Šis teksts tiks parādīts virs ievades lauka *Manuāla ievade* uznirstošajā logā. Lai skatītu *Manuāla ievade* uznirstošā loga priekšskatījumu, noklikšķiniet uz *Testēt lietotāja ievadi*. Priekšskatījums ļauj arī pārbaudīt, vai ievades virkne atbilst ievades validācijas nosacījumam (skatiet tālāk norādītos parametrus). 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, atdalot tās ar komatiem. Lai nolaižamajā sarakstā iekļautu tukšu opciju, pievienojiet papildu komatu saraksta sākumā, vidū vai beigās. Šis lauks tiek parādīts tikai tad, ja *Ievades tips* ir iestatīts uz "Nolaižamais saraksts".

Parametrs	Apraksts
Noklusējuma ievades virkne	Ievadiet lietotāja ievades noklusējuma virkni (vai atstājiet lauku tukšu). Šī virkne tiks validēta, izmantojot regulāro izteiksmi, kas norādīta laukā *Ievades validācijas nosacījums*. Šeit ievadītā vērtība pēc noklusējuma tiks parādīta *Manuāla ievade* uznirstošajā logā. Šis lauks tiek parādīts tikai tad, ja *Ievades tips* ir iestatīts uz “Virkne”.
Ievades validācijas nosacījums	Ievadiet regulāro izteiksmi, lai validētu lietotāja ievades virkni. Tiek atbalstītas globālās regulārās izteiksmes. Šis lauks tiek parādīts tikai tad, ja *Ievades tips* ir iestatīts uz “Virkne”.
Iespējot apstiprināšanu	Atzīmējiet izvēles rūtiņu, lai pirms skripta izpildes parādītu apstiprinājuma ziņojumu. Šī funkcija var būt īpaši noderīga potenciāli bīstamām darbībām (piemēram, atsāknēšanas skriptam) vai darbībām, kuru izpilde var aizņemt ilgu laiku.
Apstiprinājuma teksts	Ievadiet pielāgotu apstiprinājuma tekstu apstiprinājuma uznirstošajam logam, kas iespējots ar iepriekš norādīto izvēles rūtiņu (piemēram, *Attālā sistēma tiks atsāknēta. Vai tiešām vēlaties turpināt?*). Lai skatītu, kā teksts izskatīsies, noklikšķiniet uz *Testēt apstiprinājumu* blakus laukam. Tiek atbalstīti makro un pielāgoti lietotāja makro. *Piezīme:* makro netiks izvērsti, testējot apstiprinājuma ziņojumu.

Ja ir konfigurēta gan manuāla lietotāja ievade, gan apstiprinājuma ziņojums, tie tiks parādīti secīgos uznirstošajos logos.

Manuāla lietotāja ievade

Manuāla lietotāja ievade ļauj katrā skripta izpildes reizē norādīt pielāgotu parametru. Tādējādi nav nepieciešams izveidot vairākus līdzīgus lietotāja skriptus, kas atšķiras tikai ar vienu parametru.

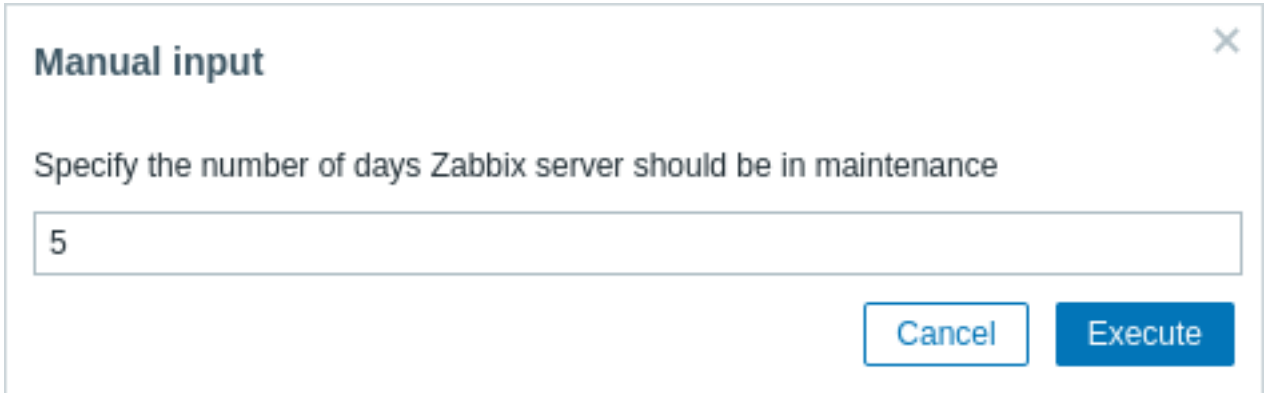
Piemēram, skripta izpildes laikā var būt nepieciešams norādīt citu veselu skaitli vai citu URL adresi.

Lai iespējotu manuālu lietotāja ievadi:

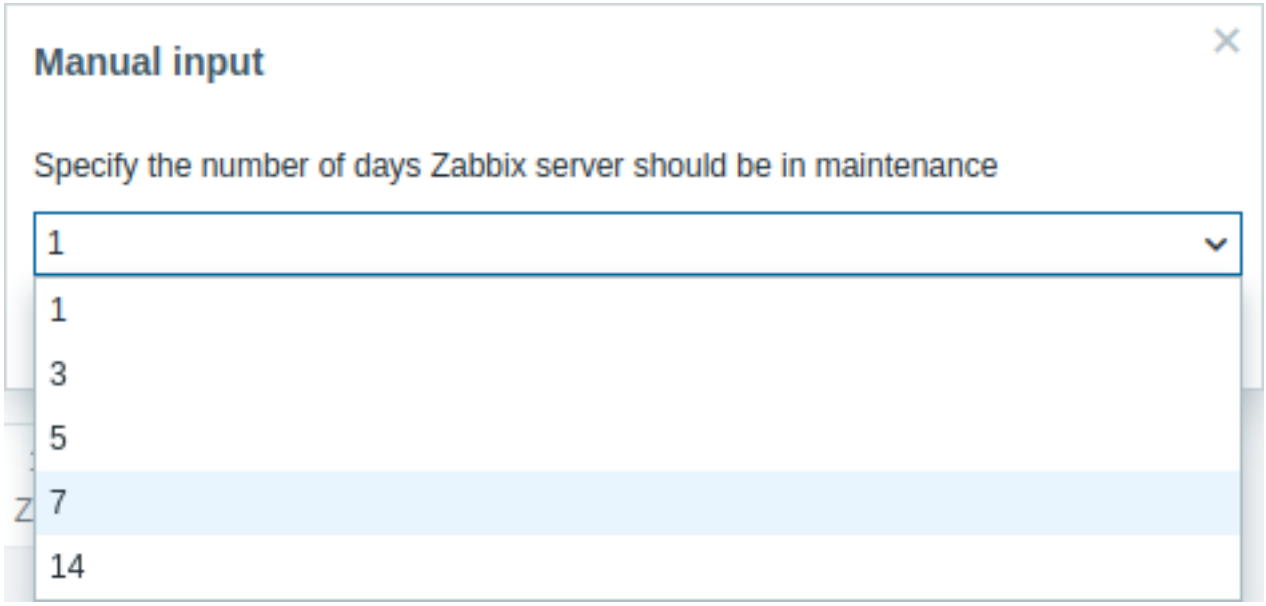
- izmantojiet makro {MANUALINPUT} skriptā (komandās, skriptā vai skripta parametrā), kur tas nepieciešams, vai URL skriptu URL laukā;
- papildu skripta konfigurācijā iespējojiet manuālu lietotāja ievadi un konfigurējiet ievades opcijas.

Ja lietotāja ievade ir iespējota, pirms skripta izpildes lietotājam tiks parādīts uznirstošais logs *Manuāla ievade*, kurā jānorāda pielāgota vērtība. Norādītā vērtība skriptā aizstās {MANUALINPUT}.

Atkarībā no konfigurācijas lietotājam tiks lūgts ievadīt virknes vērtību:



Vai arī atlasīt vērtību nolaižamajā sarakstā ar iepriekš noteiktām opcijām:



Manuāla lietotāja ievade ir pieejama tikai skriptiem, kuru tvērums ir “Manuāla darbība ar hostu” vai “Manuāla darbība ar notikumu”.

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:

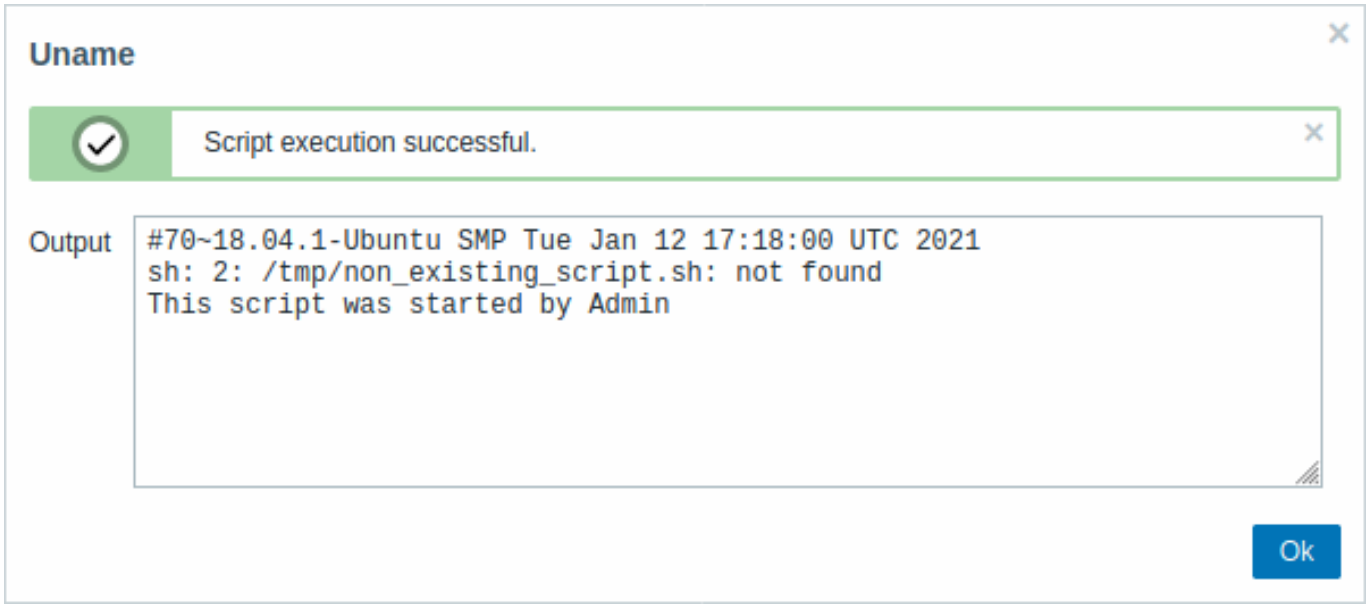
- Ja skripts beidzas veiksmīgi (izejas kods 0), atgrieztā vērtība ir ierobežota līdz 16MB (ieskaitot beigu atstarpes, kas tiek apgrieztas).
- Ja skripts beidzas ar kļūdu (izejas kods nav nulle), atgrieztā vērtība ir standarta kļūdu izvade, kas ierobežota līdz 2KB.

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}"
```



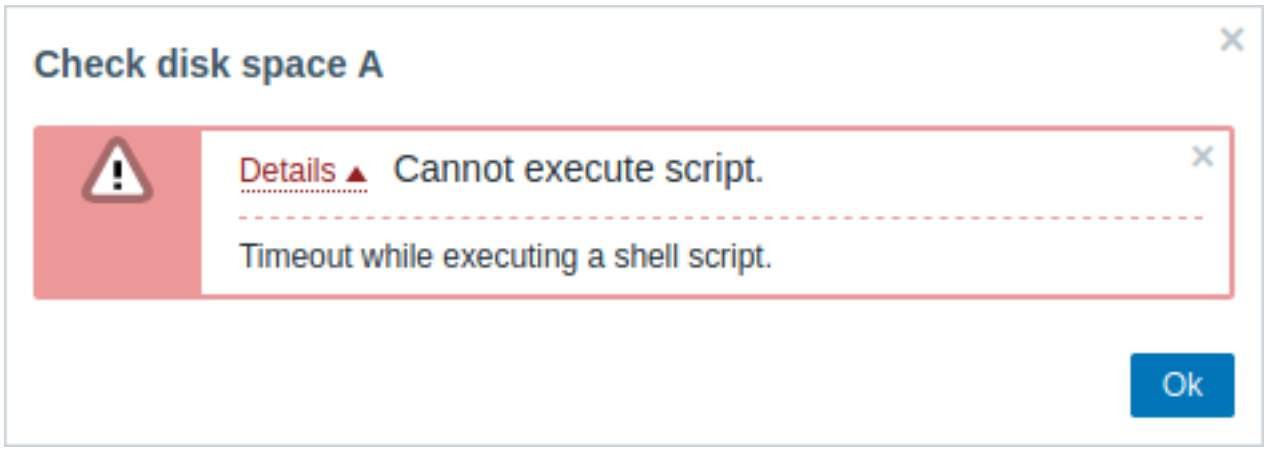
Skripta rezultātā netiek parādīts pats skripts.

Skripta izpildes noildze Zabbix aģentā

Var rasties situācija, kad skripta izpildes laikā iestājas noildze.

Tālāk skatiet skripta, kas darbojas Zabbix aģentā, piemēru un rezultātu logu:

```
sleep 5
df -h
```



Šajā gadījumā tiek parādīts šāds kļūdas ziņojums:

```
Timeout while executing a shell script.
```

Lai izvairītos no šādām situācijām, ieteicams optimizēt pašu skriptu (iepriekš minētajā piemērā — "5"), nevis mainīt parametru `Timeout` Zabbix aģenta konfigurācijā un Zabbix servera konfigurācijā.

Ja parametrs `Timeout` 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 parametru `Timeout` nepieciešams mainīt arī Zabbix servera konfigurācijā.

Skripta taimauts aktīvajiem aģentiem

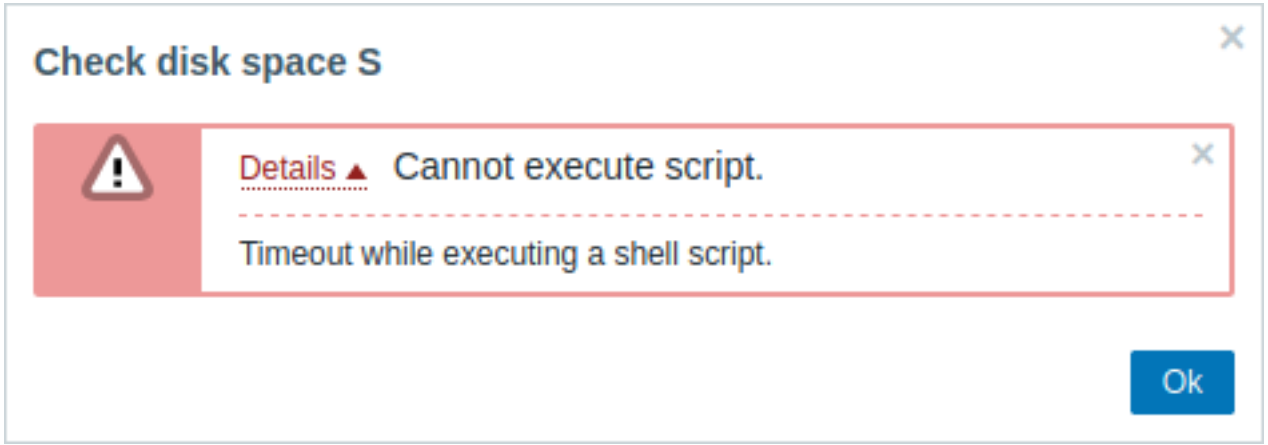
Skriptus ir iespējams izpildīt tikai uz aģentiem, kas darbojas aktīvajā režīmā. Kad skripta izpildi iedarbina darbība operation vai manuāla skripta izpilde, komanda tiek iekļauta aktīvās pārbaudes konfigurācijā un izpildīta, tiklīdz aktīvais aģents to saņem.

Manuālie skripti tiek nosūtīti aktīvajam aģentam kopā ar servera/starpniekservera taimautu skripta izpildei. Lūdzu, palieliniet noklusējuma serveris/starpniekserveris taimautu skripta izpildei. Taimautam jābūt lielākam par aktīvās pārbaudes atsvaidzināšanas biežumu, pretējā gadījumā taimauts tiek pārsniegts, pirms aktīvais aģents saņem skriptu un var atgriezt rezultātu.

Skripta taimauts Zabbix serverī/starpniekserverī

Tālāk skatiet skripta, kas darbojas Zabbix serverī, piemēru un rezultātu logu:

```
sleep 11
df -h
```



Ieteicams arī optimizēt pašu skriptu, nevis pielāgot `TrapperTimeout` parametru atbilstošai vērtībai (mūsu gadījumā — > 11), modificējot Zabbix servera konfigurāciju.

Lietotāji

Pārskats

Šajā izvēlnē ir sadaļas, kas saistītas ar lietotāju konfigurēšanu Zabbix. Šī izvēlne ir pieejama tikai SuperAdmin lietotāja tipa lietotājiem.

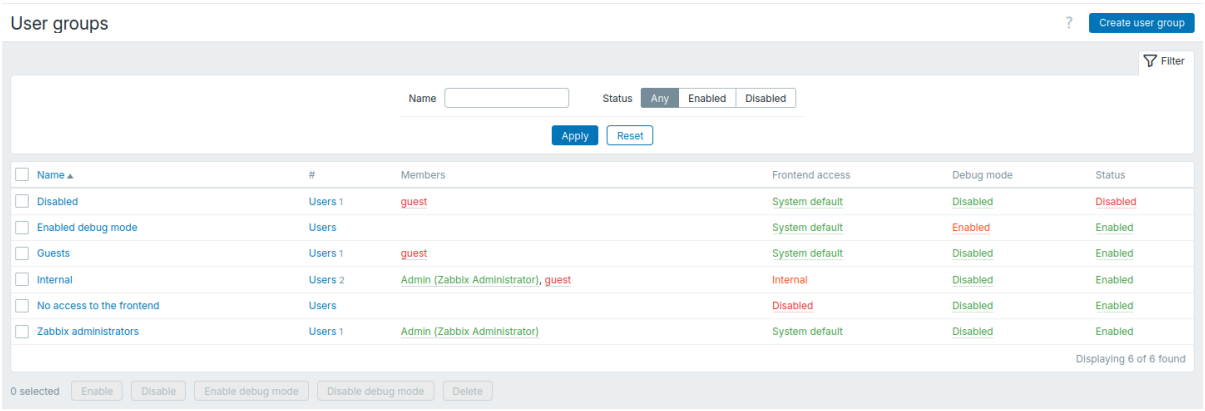
Lietotāju grupas

Pārskats

Sadaļā *Users* > *User groups* 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 veidlapa.
#	Lietotāju skaits grupā. Noklikšķinot uz *Lietotāji*, lietotāju sarakstā tiks parādīti attiecīgie lietotāji, izmantojot filtru.
Dalībnieki	Atsevišķu lietotāju lietotājvārdi lietotāju grupā (iekavās norādot vārdu un uzvārdu). Noklikšķinot uz lietotājvārda, tiek atvērta lietotāja konfigurācijas veidlapa. Lietotāji no atspējotām grupām tiek parādīti sarkanā krāsā.
Lietotāja saskarnes piekļuve	Tiek parādīts lietotāja saskarnes piekļuves līmenis: **Sistēmas noklusējums** - lietotāji tiek autentificēti, izmantojot Zabbix, LDAP vai HTTP (atkarībā no globāli iestatītās autentifikācijas metodes); **Iekšējā** - lietotāji tiek autentificēti, izmantojot Zabbix; tiek ignorēts, ja HTTP autentifikācija ir globālais noklusējums; **LDAP** - lietotāji tiek autentificēti, izmantojot LDAP; tiek ignorēts, ja HTTP autentifikācija ir globālais noklusējums; **Atspējota** - šai grupai ir aizliegta piekļuve Zabbix lietotāja saskarnei. 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 - *Iespējots* vai *Atspējots*. Noklikšķinot uz statusa, to var mainīt.

Lai konfigurētu jaunu lietotāju grupu, augšējā labajā stūrī noklikšķiniet uz pogas *Izveidot lietotāju grupu*.

Masveida rediģēšanas opcijas

Pogas zem saraksta piedāvā vairākas masveida rediģēšanas opcijas:

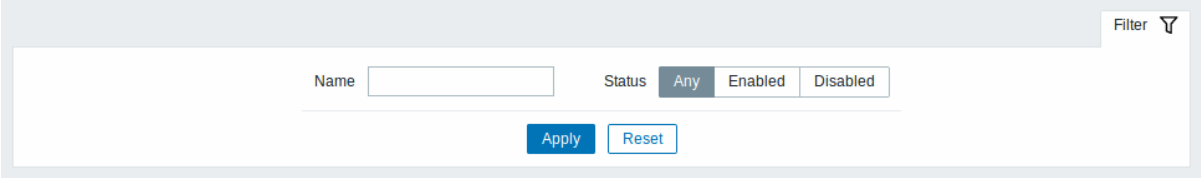
- *Iespējot* - mainīt lietotāju grupas statusu uz *Iespējots*
- *Atspējot* - mainīt lietotāju grupas statusu uz *Atspējots*
- *Iespējot atkļūdošanas režīmu* - iespējot lietotāju grupām atkļūdošanas režīmu
- *Atspējot atkļūdošanas režīmu* - atspējot lietotāju grupām atkļūdošanas režīmu
- *Dzēst* - dzēst lietotāju grupas

Lai izmantotu šīs opcijas, atzīmējiet izvēles rūtiņas pie attiecīgajām lietotāju grupām un pēc tam noklikšķiniet uz vajadzīgās pogas.

Filtra izmantošana

Varat izmantot filtru, lai parādītu tikai tās lietotāju grupas, kas jūs interesē. Lai uzlabotu meklēšanas veiktspēju, dati tiek meklēti ar neatrisinātiem makro.

Saite *Filter* ir pieejama virs lietotāju grupu saraksta. Ja uz tās noklikšķināsiet, būs pieejams filtrs, kurā varat filtrēt lietotāju grupas pēc nosaukuma un statusa.



Lietotāju lomas

Pārskats

Sadaļā *Lietotāji* > *Lietotāju lomas* varat izveidot lietotāju lomas.

Lietotāju lomas ļauj izveidot detalizēti pielāgotas atļaujas, pamatojoties uz sākotnēji atlasīto lietotāja tipu (*Lietotājs*, *Administrators*, *Superadministrators*).

Pēc lietotāja tipa atlasīšanas tiek piešķirtas visas šim lietotāja tipam pieejamās atļaujas (pēc noklusējuma tās ir atzīmētas).

Atļaujas var atsaukt tikai no tām, kas ir pieejamas konkrētajam lietotāja tipam; tās nevar paplašināt, pārsniedzot šim lietotāja tipam pieejamās atļaujas.

Nepieejamo atļauju izvēles rūtiņas ir pelēkotas; lietotāji nevarēs piekļūt elementam pat tad, ja pārlūkprogrammā ievadīs tiešu URL uz šo elementu.

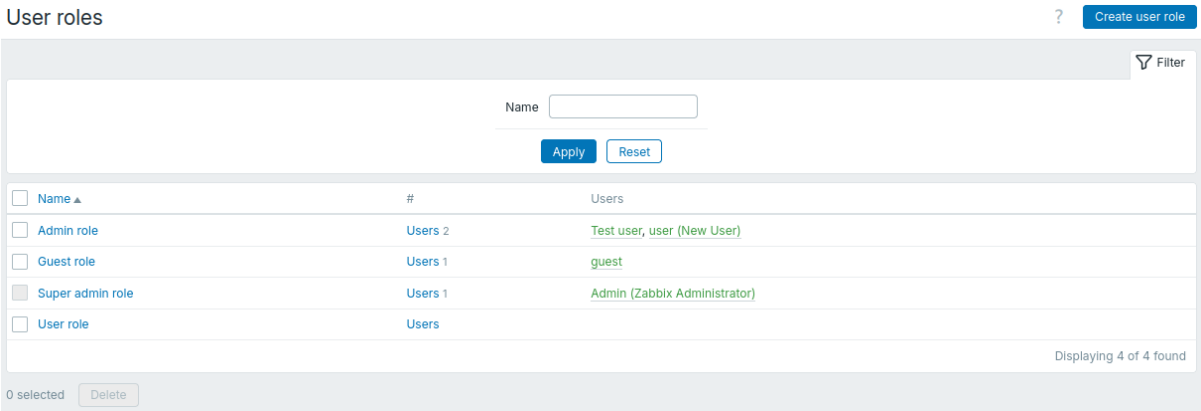
Note:
Piekļuves ierobežošana dažiem lietotāja saskarnes elementiem tikai neļauj atvērt attiecīgo lapu — tā neatceļ iespēju piekļūt pamatā esošajiem datiem citās saskarnes daļās.

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
- Superadministratora loma



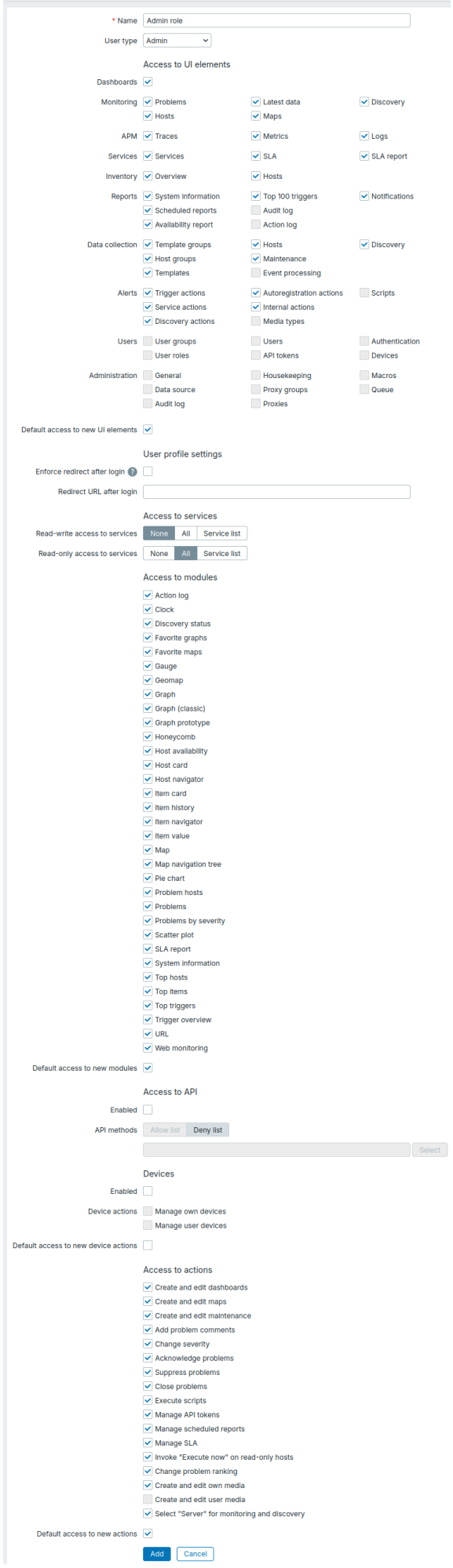
Šīs lomas ir balstītas uz galvenajiem lietotāju tipiem Zabbix. Tiek parādīts visu lietotāju saraksts, kuriem ir piešķirta attiecīgā loma. Lietotāji, kas ir iekļauti atspējotās grupās, tiek parādīti sarkanā krāsā. *Viesa loma* ir lietotāja tipa loma, kurai ir tikai atļaujas skatīt dažas lietotāja saskarnes sadaļas.

Note:
Noklusējuma *superadministratora lomu* nevar modificēt vai dzēst, jo Zabbix ir jābūt vismaz vienam superadministratora lietotājam ar neierobežotām privilēģijām. *Superadministratora* tipa lietotāji var mainīt savas lomas iestatījumus, bet ne lietotāja tipu.

Konfigurācija

Lai izveidotu jaunu lomu, augšējā labajā stūrī noklikšķiniet uz pogas *Create user role*. Lai atjauninātu esošu lomu, noklikšķiniet uz lomas nosaukuma, lai atvērtu konfigurācijas veidlapu.

User roles


* Name
Admin role
User type
Admin
Access to UI elements
Dashboards
Monitoring
Problems
Latest data
Discovery
Hosts
Maps
APM
Traces
Metrics
Logs
Services
Services
SLA
SLA report
Inventory
Overview
Hosts
Reports
System information
Top 100 triggers
Notifications
Scheduled reports
Audit log
Availability report
Action log
Data collection
Template groups
Hosts
Discovery
Host groups
Maintenance
Templates
Event processing
Alerts
Trigger actions
Autoregistration actions
Scripts
Service actions
Internal actions
Discovery actions
Media types
Users
User groups
Users
Authentication
User roles
API tokens
Devices
Administration
General
Housekeeping
Macros
Data source
Proxy groups
Queue
Audit log
Proxies
Default access to new UI elements
User profile settings
Enforce redirect after login
Redirect URL after login
Access to services
Read-write access to services
None
All
Service list
Read-only access to services
None
All
Service list
Access to modules
Action log
Clock
Discovery status
Favorite graphs
Favorite maps
Gauge
Geomap
Graph
Graph (classic)
Graph prototype
Honeycomb
Host availability
Host card
Host navigator
Item card
Item history
Item navigator
Item value
Map
Map navigation tree
Pie chart
Problem hosts
Problems
Problems by severity
Scatter plot
SLA report
System information
Top hosts
Top items
Top triggers
Trigger overview
URL
Web monitoring
Default access to new modules
Access to API
Enabled
API methods
Allow list
Deny list
Select
Devices
Enabled
Device actions
Manage own devices
Manage user devices
Default access to new device actions
Access to actions
Create and edit dashboards
Create and edit maps
Create and edit maintenance
Add problem comments
Change severity
Acknowledge problems
Suppress problems
Close problems
Execute scripts
Manage API tokens
Manage scheduled reports
Manage SLA
Invoke "Execute now" on read-only hosts
Change problem ranking
Create and edit own media
Create and edit user media
Select "Server" for monitoring and discovery
Default access to new actions
Add
Cancel

Tiek parādītas pieejamās atļaujas. Lai atsauktu noteiktu atļauju, noņemiet atzīmi no tās izvēles rūtiņas.

Tālāk ir aprakstītas pieejamās atļaujas, kā arī noklusējuma iestatījumi katrai iepriekš konfigurētajai Zabbix lietotāja lomai.

Noklusējuma atļaujas

Piekļuve lietotāja saskarnes elementiem

Noklusējuma piekļuve izvēlnes sadaļām ir atkarīga no lietotāja tipa. Detalizētu informāciju skatiet lapā Atļaujas.

Piekļuve citām opcijām

Dažas no tālāk tabulā aprakstītajām sadaļām var būt paslēptas:

- *Piekļuve moduļiem* ir paslēpta, ja `$ZBX_FEATURE_FLAGS['modules_config_enabled']` lietotāja saskarnes konfigurācijas failā ir iestatīts uz `false`.
- *Ierīces* ir paslēptas, ja `EnableMobileDevices` ir atspējots.

Parametrs	**Apraksts**	**Noklusējuma lietotāju lomas**			
		Super administratora loma	*Administratora 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 un rakstīšanas piekļuve pakalpojumiem	Atlasiet lasīšanas un rakstīšanas piekļuvi pakalpojumiem: **Nav** - nav nekādas piekļuves **Visiem** - lasīšanas un rakstīšanas piekļuve visiem pakalpojumiem **Pakalpojumu saraksts** - atlasiet pakalpojumus, kuriem piešķirt lasīšanas un rakstīšanas piekļuvi Ja lasīšanas un rakstīšanas piekļuve ir piešķirta, tai ir prioritāte pār tikai lasīšanas piekļuves iestatījumiem, un tā tiek dinamiski mantota pakārtotajos pakalpojumos.	Visiem	Visiem	Nav	Nav
Lasīšanas un rakstīšanas piekļuve pakalpojumiem ar tagu	Norādiet taga nosaukumu un, pēc izvēles, vērtību, lai papildus piešķirtu lasīšanas un rakstīšanas piekļuvi pakalpojumiem, kas atbilst tagam. Šī opcija ir pieejama, ja parametrā *Lasīšanas un rakstīšanas piekļuve pakalpojumiem* ir atlasīts *Pakalpojumu saraksts*. Ja lasīšanas un rakstīšanas piekļuve ir piešķirta, tai ir prioritāte pār tikai lasīšanas piekļuves iestatījumiem, un tā tiek dinamiski mantota pakārtotajos pakalpojumos.				

Tikai lasīšanas piekļuve pakalpojumiem	Atlasiet tikai lasīšanas piekļuvi pakalpojumiem: **Nav** - nav nekādas piekļuves **Visiem** - tikai lasīšanas piekļuve visiem pakalpojumiem **Pakalpojumu saraksts** - atlasiet pakalpojumus, kuriem piešķirt tikai lasīšanas piekļuvi Tikai lasīšanas piekļuvei nav prioritātes pār lasīšanas un rakstīšanas piekļuvi, un tā tiek dinamiski mantota pakārtotajos pakalpojumos.			Visiem	Visiem
Tikai lasīšanas piekļuve pakalpojumiem ar tagu	Norādiet taga nosaukumu un, pēc izvēles, vērtību, lai papildus piešķirtu tikai lasīšanas piekļuvi pakalpojumiem, kas atbilst tagam. Šī opcija ir pieejama, ja parametrā *Tikai lasīšanas piekļuve pakalpojumiem* ir atlasīts *Pakalpojumu saraksts*. Tikai lasīšanas piekļuvei nav prioritātes pār lasīšanas un rakstīšanas piekļuvi, un tā tiek dinamiski mantota pakārtotajos pakalpojumos.				
Piekļuve moduļiem					
<Moduļa nosaukums>	Atļaut vai liegt 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 jaunie moduļi un logrīki būs pieejami pēc Zabbix jauninašanas. Tā attiecas arī uz moduļiem un logrīkiem, kas pievienoti sadaļā *Administrēšana > Vispārīgi > Moduļi*.				
Piekļuve API					
Iespējots API metodes	Iespējot vai atspējot piekļuvi API. Atlasiet *Atļauju saraksts*, lai atļautu, vai *Aizliegumu saraksts*, lai liegtu meklēšanas laukā norādītās API metodes. Ņemiet vērā, ka nav iespējams dažas API metodes atļaut un citas liegt. Meklēšanas laukā sāciet rakstīt metodes nosaukumu un pēc tam atlasiet metodi automātiskās pabeigšanas sarakstā. Varat arī noklikšķināt uz pogas *Atlasīt* un atlasīt metodes no pilna saraksta, kas pieejams šim lietotāja tipam. Ņemiet vērā: ja kāda darbība sadaļā *Piekļuve darbībām* nav atlasīta, lietotāji nevarēs izmantot ar šo darbību saistītās API metodes. Tiek atbalstītas aizstājējzīmes. Piemēri: `dashboard.*` (visas API pakalpojuma “dashboard.” metodes), `*` (jebkura metode), `*.export` (metodes ar nosaukumu “.export” no visiem API pakalpojumiem). Ja metodes nav norādītas, noteikums *Atļauju/aizliegumu saraksts* tiks ignorēts.	Nē	Nē	Nē	Nē
Piekļuve darbībām					
Informācijas paneļu izveide un rediģēšana Kartes izveide un rediģēšana	Noņemot atzīmi no šīs izvēles rūtiņas, tiks atņemtas arī tiesības izmantot `.create`, `.update` un `.delete` API metodes attiecīgajiem elementiem.	Jā	Jā	Jā	Nē

Apkopes izveide un rediģēšana				Nē	
Problēmu komentāru pievienošana Mainīt nopietnību Apstiprināt problēmas Apspiest problēmas Aizvērt problēmas	Noņemot atzīmi no šīs izvēles rūtiņas, tiks atņemtas arī tiesības veikt attiecīgo darbību, izmantojot `event.acknowledge` API metodi.			Jā	
Izpildīt skriptus	Noņemot atzīmi no šīs izvēles rūtiņas, tiks atņemtas arī tiesības izmantot `script.execute` API metodi.				
Pārvaldīt API pilnvaras	Noņemot atzīmi no šīs izvēles rūtiņas, tiks atņemtas arī tiesības izmantot visas `token.` API metodes.				
Pārvaldīt ieplānotos pārskatus	Noņemot atzīmi no šīs izvēles rūtiņas, tiks atņemtas arī tiesības izmantot visas `report.` API metodes.			Nē	
Pārvaldīt SLA	Iespējot vai atspējot tiesības pārvaldīt SLA.				
Izmantot "Izpildīt tūlīt" tikai lasāmiem hosts	Atļaut lietotājiem izmantot opciju "Izpildīt tūlīt" sadaļā Jaunākie dati vienumiem tikai lasāmos hosts.			Jā	
Mainīt problēmu secību	Atļaut lietotājiem mainīt problēmas secību no cēloņa uz simptomu un otrādi.				
Izveidot un rediģēt savus multivides līdzekļus	Atļaut lietotājiem izveidot un rediģēt savus multivides līdzekļus.				
Izveidot un rediģēt lietotāju multivides līdzekļus	Atļaut lietotājiem izveidot un rediģēt citu lietotāju multivides līdzekļus. Šī opcija ir pieejama tikai super administratoriem.		Nē	Nē	
Atlasīt "Serveris" uzraudzībai un atklāšanai	Noņemot atzīmi no šīs izvēles rūtiņas, tiek ierobežota to hosts izveide, kurus uzrauga serveris, un tīkla atklāšanas noteikumu, ko izpilda serveris, izveide, kā arī vienumu testēšana ar serveri. Opcija *Serveris* būs atspējota hosta konfigurācijā parametram *Uzrauga*, tīkla atklāšanas noteikuma konfigurācijā parametram *Atklāj*, kā arī vienumu testēšanā parametram *Testēt ar*.		Jā	Nē	
Noklusējuma piekļuve jaunām darbībām	Šī opcija nosaka, kā jaunās darbības būs pieejamas pēc Zabbix jaunināšanas.		Jā	Jā	
Ierīces					
Iespējots	Iespējot vai atspējot piekļuvi *Lietotāja iestatījumi* > *Ierīces* (pieejams visiem lietotājiem) un *Lietotāji* > *Ierīces* (pieejams tikai super administratoriem).	Jā	Nē	Nē	Nē
Pārvaldīt savas ierīces	Atļaut lietotājiem pārvaldīt savas ierīces sadaļā *Lietotāja iestatījumi* > *Ierīces*.				
Pārvaldīt lietotāju ierīces	Atļaut lietotājiem pārvaldīt citiem lietotājiem piederošās ierīces sadaļā *Lietotāji* > *Ierīces*.				
Noklusējuma piekļuve jaunām darbībām	Šī opcija nosaka, kā jaunās darbības būs pieejamas pēc Zabbix jaunināšanas.				

Skatiet arī:

- Lietotāja konfigurēšana

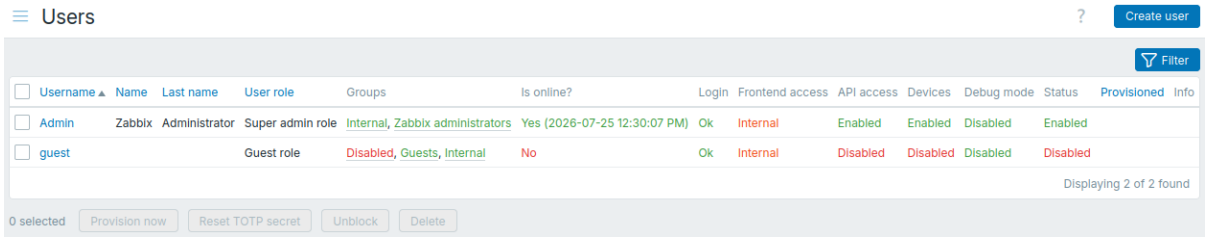
Lietotāji

Pārskats

Sadaļā *Lietotāji* > *Lietotāji* tiek pārvaldīti sistēmas lietotāji.

Lietotāji

Tiek parādīts esošo lietotāju saraksts ar to detalizētu informāciju.



Parādītie dati:

Kolonna	Apraksts
Lietotājvārds	Lietotājvārds, lai pieteiktos Zabbix. Noklikšķinot uz lietotājvārda, tiek atvērta lietotāja konfigurācijas veidlapa.
Vārds	Lietotāja vārds.
Uzvārds	Lietotāja uzvārds.
Lietotāja loma	Tiek parādīta lietotāja loma.
Grupas	Tiek uzskaitītas grupas, kuru dalībnieks ir lietotājs. Noklikšķinot uz lietotāju grupas nosaukuma, tiek atvērta lietotāju grupas konfigurācijas veidlapa. Atspējotās grupas tiek parādītas sarkanā krāsā.
Vai tiešsaistē?	Tiek parādīts lietotāja tiešsaistes statuss - *Jā* vai *Nē*. Iekavās tiek parādīts pēdējās lietotāja aktivitātes laiks.
Pieteikšanās	Tiek parādīts lietotāja pieteikšanās statuss - *Labi* vai *Bloķēts*. Lietotājs var tikt īslaicīgi bloķēts, ja tiek pārsniegts *Administrēšana > Vispārīgi > Citi* sadaļā iestatītais neveiksmīgo pieteikšanās mēģinājumu skaits (pēc noklusējuma pieci). Noklikšķinot uz *Bloķēts*, lietotāju var atbloķēt.
Lietotāja saskarnes piekļuve	Tiek parādīts lietotāja saskarnes piekļuves līmenis - *Sistēmas noklusējums*, *Iekšēja*, *LDAP* vai *Atspējota* atkarībā no iestatījuma visai lietotāju grupai.
API piekļuve	Tiek parādīts API piekļuves statuss - *Iespējota* vai *Atspējota* atkarībā no lietotāja lomai iestatītās vērtības.
Ierīces	Tiek parādīts ierīču piekļuves statuss - *Iespējota* vai *Atspējota* atkarībā no lietotāja lomai iestatītās vērtības. Šī kolonna tiek parādīta, ja ir iespējota `EnableMobileDevices`.
Atkļūdošanas režīms	Tiek parādīts atkļūdošanas režīma statuss - *Iespējots* vai *Atspējots* atkarībā no iestatījuma visai lietotāju grupai.
Statuss	Tiek parādīts lietotāja statuss - *Iespējots* vai *Atspējots* atkarībā no iestatījuma visai lietotāju grupai.
Nodrošināts	Tiek parādīts datums, kad lietotājs pēdējo reizi tika nodrošināts. Izmanto lietotājiem, kas izveidoti, izmantojot JIT nodrošināšanu no LDAP/SAML.
Informācija	Tiek parādīta informācija par kļūdām. Lietotājiem bez lietotāju grupām tiek parādīts dzeltens brīdinājums. Lietotājiem bez lomām, kā arī lietotājiem bez lomām un lietotāju grupām tiek parādīts sarkans brīdinājums.

Lai konfigurētu jaunu lietotāju, augšējā labajā stūrī noklikšķiniet uz pogas *Izveidot lietotāju*.

Masveida rediģēšanas opcijas

Zem saraksta esošās pogas piedāvā vairākas masveida rediģēšanas opcijas:

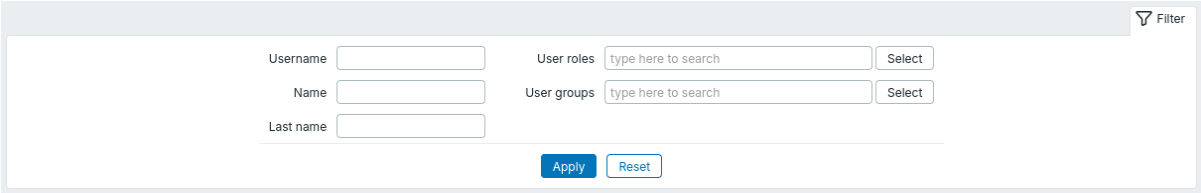
- *Nodrošināt tūlīt* - atjaunināt lietotāja informāciju no LDAP (šī opcija ir iespējota tikai tad, ja ir atlasīts LDAP lietotājs)
- *Atiestatīt TOTP noslēpumu* - atiestatīt lietotāju TOTP noslēpumus visām TOTP metodēm un dzēst lietotāja sesiju (šī opcija ir iespējota tikai tad, ja ir iespējota MFA; lietotājiem bez TOTP noslēpumiem sesija netiks dzēsta)
- *Atbloķēt* - atkārtoti iespējot sistēmas piekļuvi bloķētajiem lietotājiem
- *Dzēst* - dzēst lietotājus

Lai izmantotu šīs opcijas, atzīmējiet izvēles rūtiņas pirms attiecīgajiem lietotājiem un pēc tam noklikšķiniet uz nepieciešamās pogas.

Filtra izmantošana

Varat izmantot filtru, lai parādītu tikai tos lietotājus, kas jūs interesē. Lai uzlabotu meklēšanas veiktspēju, dati tiek meklēti ar neatrisinātiem makro.

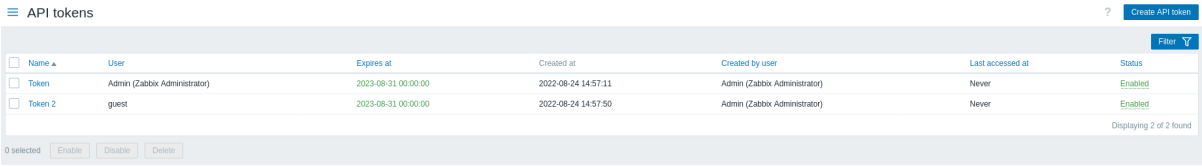
Virs lietotāju saraksta ir pieejama saite *Filtrs*. Noklikšķinot uz tās, kļūst pieejams filtrs, ar kuru varat filtrēt lietotājus pēc lietotājvārda, vārda, uzvārda, lietotāja lomas un lietotāju grupas.



API marķieri

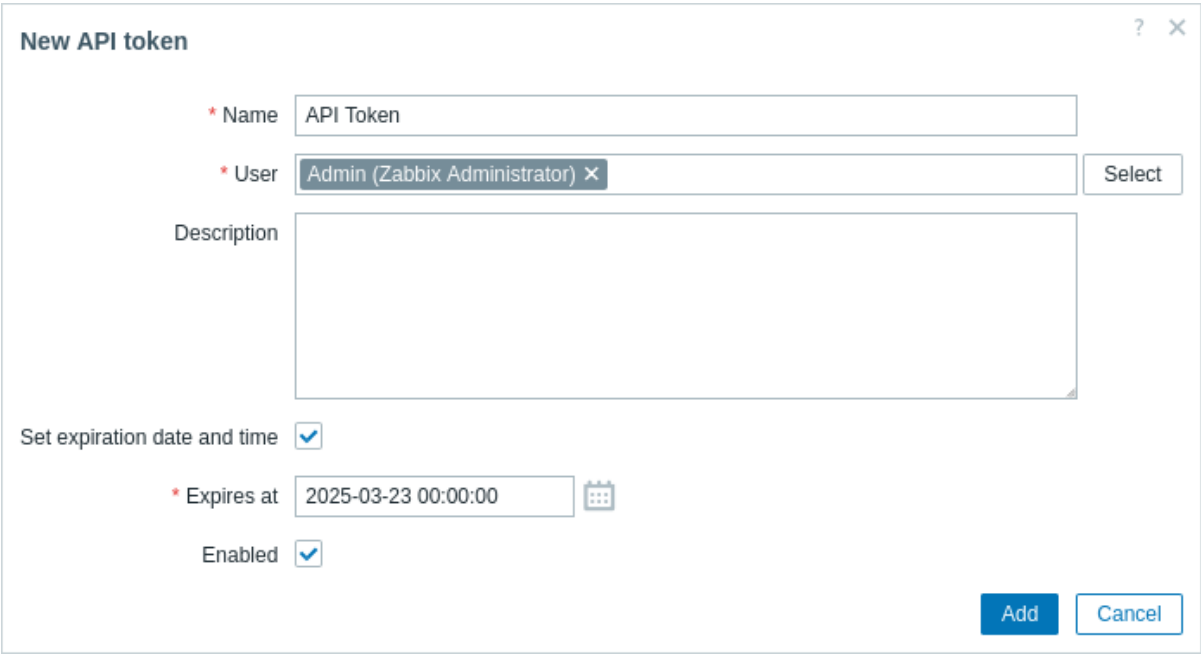
Pārskats

Šajā sadaļā varat izveidot un pārvaldīt API pilnvaras.
 API pilnvaras tiek izmantotas autentifikācijai Zabbix API.
 Katra API pilnvara tiek piešķirta vienam lietotājam un pārmanto šī lietotāja atļaujas. Ar API pilnvaru autentificēti pieprasījumi tiek autorizēti atbilstoši saistītā lietotāja lomai un atļaujām.



API pilnvaras varat filtrēt pēc nosaukuma, lietotājiem, kuriem pilnvaras ir piešķirtas, derīguma termiņa, lietotājiem, kuri izveidoja pilnvaras, vai statusa (iespējota/atspējota). Noklikšķiniet uz pilnvaras statusa sarakstā, lai ātri iespējotu/atspējotu pilnvaru. Varat arī masveidā iespējot/atspējot pilnvaras, atlasot tās sarakstā un pēc tam noklikšķinot uz pogām *Enable/Disable* zem saraksta.

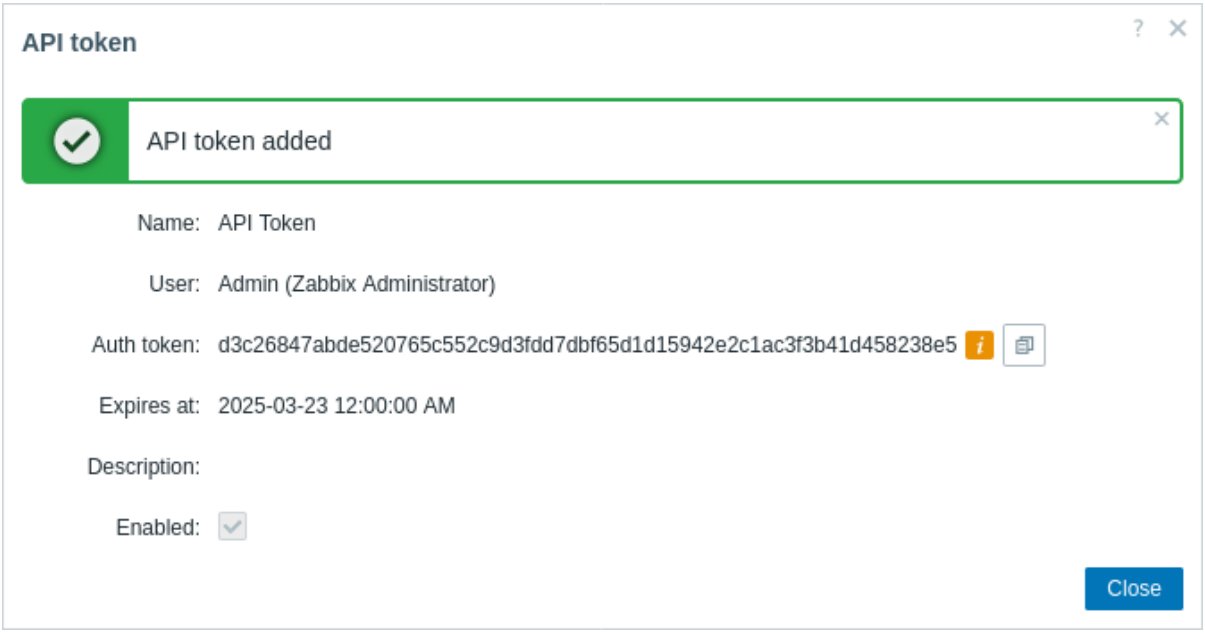
Lai izveidotu jaunu pilnvaru, augšējā labajā stūrī noklikšķiniet uz pogas *Create API token* un pēc tam pilnvaras konfigurācijas ekrānā aizpildiet obligātos laukus:



Parameter	Description
Name	Pilnvaras redzamais nosaukums.
User	Lietotājs, kuram jāpiešķir pilnvara. Lai ātri atlasītu lietotāju, sāciet rakstīt lietotājvārdu, vārdu vai uzvārdu un pēc tam automātiskās pabeigšanas sarakstā atlasiet vajadzīgo lietotāju. Varat arī noklikšķināt uz pogas *Select* un atlasīt lietotāju pilnajā lietotāju sarakstā. Pilnvaru var piešķirt tikai vienam lietotājam.
Description	Neobligāts pilnvaras apraksts.
Set expiration date and time	Noņemiet atzīmi no šīs izvēles rūtiņas, ja pilnvarai nedrīkst būt derīguma termiņa.
Expiry date	Noklikšķiniet uz kalendāra ikonas, lai atlasītu pilnvaras derīguma termiņu, vai ievadiet datumu manuāli formātā YYYY-MM-DD hh:mm:ss

Parameter	Description
Enabled	Noņemiet atzīmi no šīs izvēles rūtiņas, ja pilnvara jāizveido atspējotā stāvoklī.

Noklikšķiniet uz *Add*, lai izveidotu pilnvaru.



Nokopējiet vērtību *Auth token* un saglabājiet to drošā vietā **pirms lapas aizvēršanas**, pēc tam noklikšķiniet uz *Close*. Pilnvara tiks parādīta sarakstā.

Warning:
Vērtību *Auth token* vēlāk vairs nevarēs apskatīt. Tā ir pieejama tikai uzreiz pēc pilnvaras izveides. Ja saglabātā pilnvara tiek pazaudēta, tā būs jāģenerē no jauna, un tādējādi tiks izveidota jauna autorizācijas virkne.

Noklikšķiniet uz pilnvaras nosaukuma, lai rediģētu nosaukumu, aprakstu, derīguma termiņa iestatījumus vai pilnvaras statusu. Ņemiet vērā, ka nav iespējams mainīt lietotāju, kuram pilnvara ir piešķirta. Noklikšķiniet uz pogas *Update*, lai saglabātu izmaiņas. Ja pilnvara ir pazaudēta vai atklāta, varat noklikšķināt uz pogas *Regenerate*, lai ģenerētu jaunu pilnvaras vērtību. Tiks parādīts apstiprinājuma dialoglodziņš ar lūgumu apstiprināt šo darbību, jo pēc tās veikšanas iepriekš ģenerētā pilnvara kļūs nederīga.

Lietotāji, kuriem nav piekļuves izvēlnes sadaļai *Administration*, var skatīt un mainīt viņiem piešķirto pilnvaru informāciju tikai sadaļā *User profile* > *API tokens* section, ja viņu user role atļaujās ir iespējota opcija *Manage API tokens*.

Autentifikācija

Pārskats

Sadaļā *Lietotāji* > *Autentifikācija* varat norādīt Zabbix lietotāju autentifikācijas metodi 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.

Noklusējuma autentifikācijas metodi visā sistēmā var mainīt uz LDAP. Lai to izdarītu, atveriet cilni *LDAP* un konfigurējiet LDAP parametrus, pēc tam atgriezieties cilnē *Authentication* un pārslēdziet *Default authentication* atlasītāju uz LDAP.

Ņemiet vērā, ka autentifikācijas metodi var precīzāk konfigurēt lietotāju grupas līmenī. Pat ja LDAP autentifikācija ir iestatīta globāli, dažas lietotāju grupas joprojām var autentificēt, izmantojot Zabbix. Šīm grupām lietotāja saskarnes piekļuve ir jāiestata uz Internal.

Ja visā sistēmā tiek izmantota iekšējā autentifikācija, LDAP autentifikāciju var iespējot tikai noteiktām lietotāju grupām. Šādā gadījumā LDAP autentifikācijas informāciju var norādīt un izmantot konkrētām lietotāju grupām, kurām lietotāja saskarnes piekļuve pēc tam jāiestata 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.

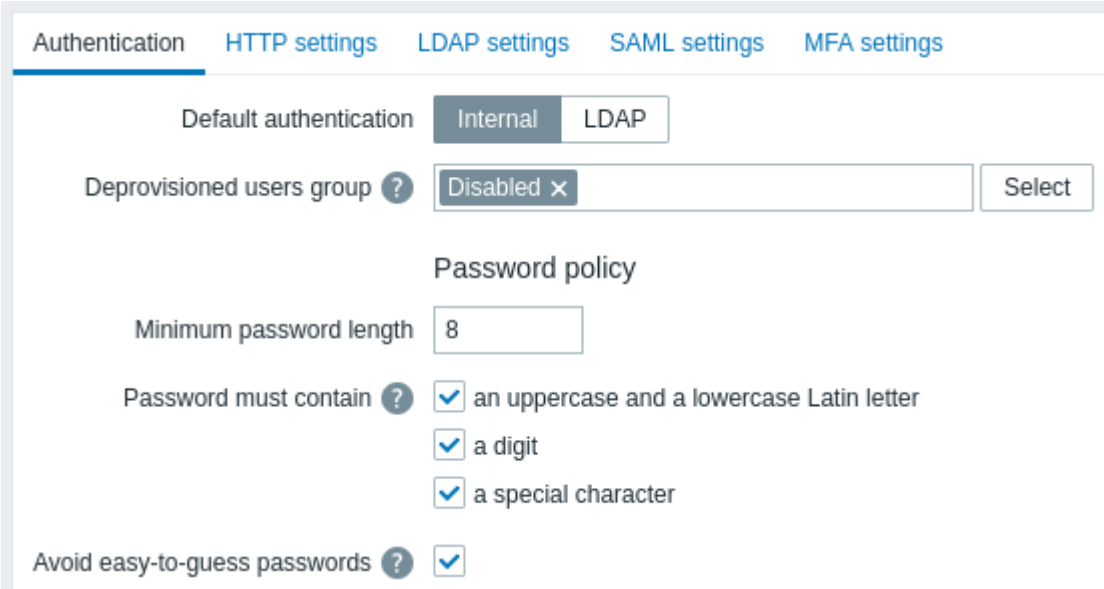
Zabbix atbalsta tieši laikā (JIT) nodrošināšanu, kas ļauj izveidot lietotāju kontus Zabbix, kad ārējais lietotājs pirmo reizi autentificējas, un nodrošināt šos lietotāju kontus. JIT nodrošināšana tiek atbalstīta LDAP un SAML.

Skatiet arī:

- HTTP autentifikācija
- LDAP autentifikācija
- SAML autentifikācija
- MFA autentifikācija

Konfigurācija

Cilne *Autentifikācija* ļauj iestatīt noklusējuma autentifikācijas metodi, norādīt grupu deaktivizētiem lietotājiem un iestatīt paroles sarežģītības prasības Zabbix lietotājiem.



Konfigurācijas parametri:

Parametrs	Apraksts
Noklusējuma autentifikācija	Atlasiet Zabbix noklusējuma autentifikācijas metodi - *Iekšējā* vai *LDAP*.
Deaktivizēto lietotāju grupa	Norādiet lietotāju grupu deaktivizētiem lietotājiem. Šis iestatījums ir nepieciešams tikai JIT nodrošināšanai attiecībā uz lietotājiem, kuri tika izveidoti Zabbix no LDAP vai SAML sistēmām, bet vairs nav jānodrošina. 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 norādīto rakstzīmju izmantošanu parolē: - lielais un mazais latīņu alfabēta burts - cipars - speciālā rakstzīme Novietojiet kursoru virs jautājuma zīmes, lai skatītu ieteikumu ar katrai opcijai atbilstošo rakstzīmju sarakstu.
Izvairīties no viegli uzminamām parolēm	Ja šī opcija ir atzīmēta, parole tiks pārbaudīta atbilstoši šādām prasībām: - nedrīkst saturēt lietotāja vārdu, uzvārdu vai lietotājvārdu - nedrīkst būt viena no bieži izmantotajām vai kontekstam specifiskajām parolēm. Bieži izmantoto un kontekstam specifisko paroļu saraksts tiek automātiski ģenerēts no NCSC "Top 100k passwords" saraksta, SecLists "Top 1M passwords" saraksta un Zabbix kontekstam specifisko paroļu saraksta. Iekšējiem lietotājiem nebūs atļauts iestatīt šajā sarakstā iekļautās paroles, jo šādas paroles to biežas izmantošanas dēļ tiek uzskatītas par nedrošām.

Paroles sarežģītības prasību izmaiņas neietekmēs esošās lietotāju paroles, taču, ja esošs lietotājs izvēlēsies mainīt paroli, jaunajai parolei būs jāatbilst pašreizējām prasībām. Ieteikums ar prasību sarakstu tiks parādīts blakus laukam *Parole* lietotāja profilā un lietotāja konfigurācijas veidlapā, kurai var piekļūt izvēlnē *Lietotāji* > *Lietotāji*.

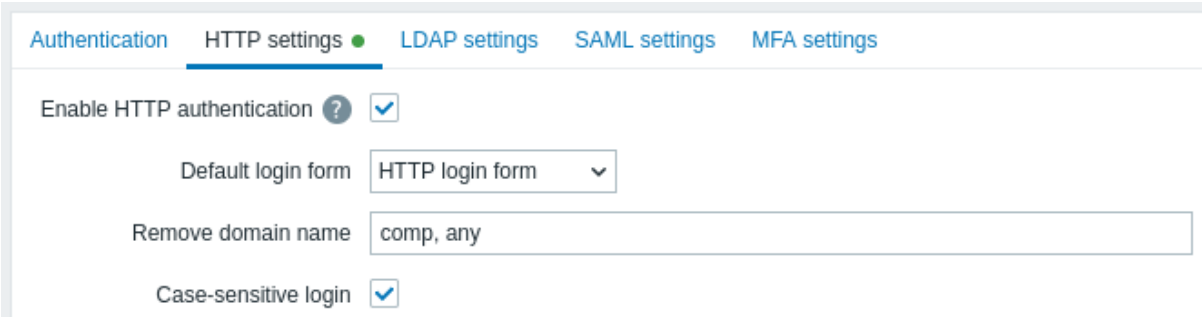
Pārskats

HTTP vai tīmekļa servera autentifikāciju autentifikācija (piemēram, BasicAuthentication, NTLM/Kerberos) var izmantot lietotājvārdu un paroļu pārbaudei. Ņemiet vērā, ka lietotājam ir jābūt arī Zabbix, tomēr tā Zabbix parole netiks izmantota.

Attention:
Esiet uzmanīgi! Pirms tās ieslēgšanas pārliecinieties, ka tīmekļa servera autentifikācija ir konfigurēta un darbojas pareizi.

Note:
HTTP autentifikācijas konfigurāciju var atspējot, iestatot `$ZBX_FEATURE_FLAGS['http_auth_enabled'] = false` lietotāja saskarnes konfigurācijas failā (`zabbix.conf.php`). Šādā gadījumā cilne *HTTP iestatījumi* ar HTTP autentifikācijas opcijām tiks paslēpta.

Konfigurācija



Konfigurācijas parametri:

Parametrs	Apraksts
Iespējot HTTP autentifikāciju	Atzīmējiet izvēles rūtiņu, lai iespējotu HTTP autentifikāciju. Novietojot peles kursoru virs , tiks parādīts ieteikums ar brīdinājumu, ka tīmekļa servera autentifikācijas gadījumā visus lietotājus (pat tos, kuriem lietotāja saskarnes piekļuve ir iestatīta uz LDAP/iekšēja) autentificēs tīmekļa serveris, nevis Zabbix.
Noklusējuma pieteikšanās veidlapa	Norādiet, uz kuru vietu novirzīt neautentificētus lietotājus: **Zabbix pieteikšanās veidlapa** - standarta Zabbix pieteikšanās lapa. **HTTP pieteikšanās veidlapa** - HTTP pieteikšanās lapa. Ieteicams iespējot autentifikāciju, kuras pamatā ir tīmekļa serveris, tikai `index_http.php` lapai. Ja *Noklusējuma pieteikšanās veidlapa* 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.
Reģistrjutīgs pieteikšanās vārds	Noņemiet atzīmi no izvēles rūtiņas, lai atspējotu lietotājvārdu reģistrjutīgu pieteikšanos (pēc noklusējuma tā ir iespējota). Reģistrjutīgas pieteikšanās atspējošana ļauj, piemēram, pieteikties kā "admin", pat ja Zabbix lietotājs ir "Admin" vai "ADMIN". Ņemiet vērā: ja reģistrjutīga pieteikšanās ir atspējota un ir vairāki Zabbix lietotāji ar līdzīgiem lietotājvārdiem (piemēram, Admin un admin), pieteikšanās šiem lietotājiem vienmēr tiks liegta ar šādu kļūdas ziņojumu: "Authentication failed: supplied credentials are not unique."

Note:
Iekšējiem lietotājiem, kuri nevar pieteikties, izmantojot HTTP akreditācijas datus (ja kā noklusējuma veidlapa ir iestatīta HTTP pieteikšanās veidlapa), kā rezultātā tiek saņemta kļūda 401, varat pievienot rindu `ErrorDocument 401 /index.php?form=default` pamatrežīma autentifikācijas direktīvām. Tā novirzīs uz parasto Zabbix pieteikšanās veidlapu.

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

LDAP lietotājiem ir iespējams konfigurēt JIT (just-in-time) **lietotāju nodrošināšanu**. Šādā gadījumā lietotājam nav obligāti jābūt jau izveidotam 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 pastāv *noklusējuma* LDAP serverī. Ja lietotājs pastāv 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 saistīts ar lietotājiem, kuri jau ir nodrošināti.

> **Attention:**
> Ja JIT nodrošināšana ir iespējota, cilnē *Authentication* ir jānorāda lietotāju grupa deprovisionētajiem lietotājiem.

JIT nodrošināšana ļauj arī atjaunināt nodrošināto lietotāju kontus, pamatojoties uz LDAP veiktajām izmaiņām. 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 viņš nepieder nevienai citai grupai, pievienots lietotāju grupai deprovisionētajiem lietotājiem. Nodrošinātie lietotāju konti tiek atjaunināti, pamatojoties uz konfigurēto nodrošināšanas periodu vai tad, kad lietotājs piesakās Zabbix.

Ņemiet vērā, ka fona nodrošināšanu veic Zabbix lietotāja saskarne, kad lietotājs ar to mijiedarbojas vai pārlūkprogrammā ir atvērta vismaz viena lietotāja saskarnes lapa. Lietotāju nodrošināšanai nav paredzēti īpaši fona procesi.

LDAP atbalsta trīs veidus, kā autentifikācijai un meklēšanai piesaistīties direktorijai:

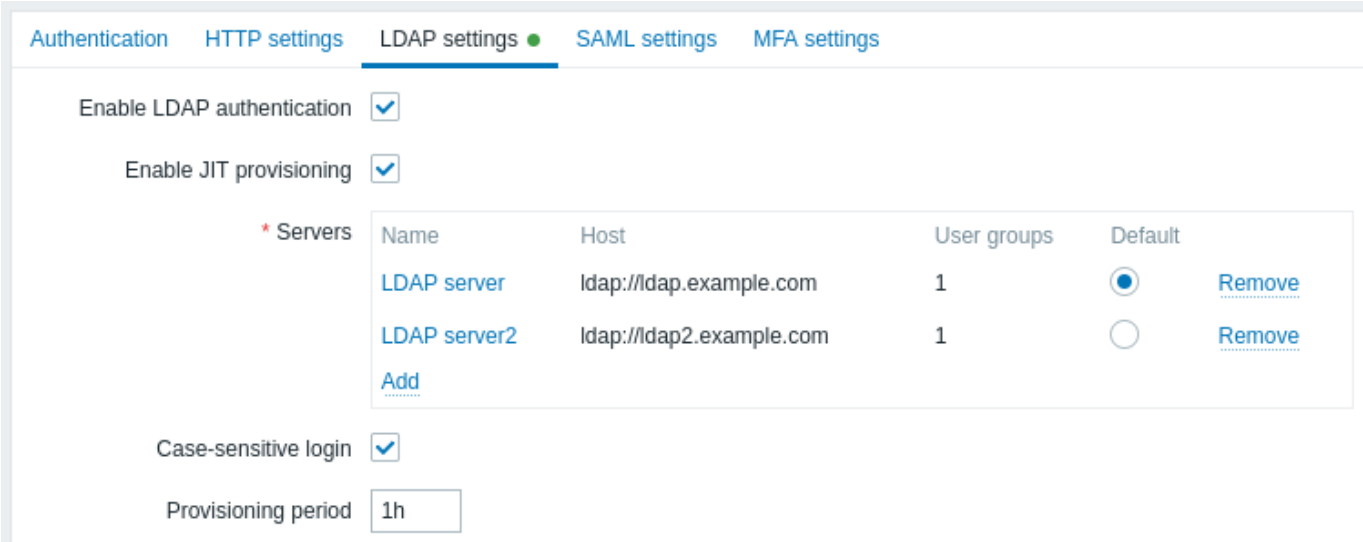
- Anonīma piesaiste - nav norādīts *Bind DN* / *Bind password*, un LDAP serveris atļauj anonīmus vaicājumus.
- Īpašs piesaistes lietotājs (pakalpojuma konts) - laukā *Bind DN* / *Bind password* ir iestatīts konkrēts LDAP konts, ko Zabbix izmanto meklēšanai un nodrošināšanai. Šī ir ieteicamā un elastīgākā opcija, 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 autentifikācija LDAP ārpus lietotāja pieteikšanās sesijas (piemēram, izmantojot pogu *Provision now* vai palaižot fona nodrošināšanu, kad lietotājs nav aktīvi pieteicies), nevar veikt autentifikāciju un tādēļ nedarbosies. Nodrošināšana un atjaunināšana tiešas lietotāja piesaistes gadījumā notiek tikai lietotāja pieteikšanās brīdī.

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.

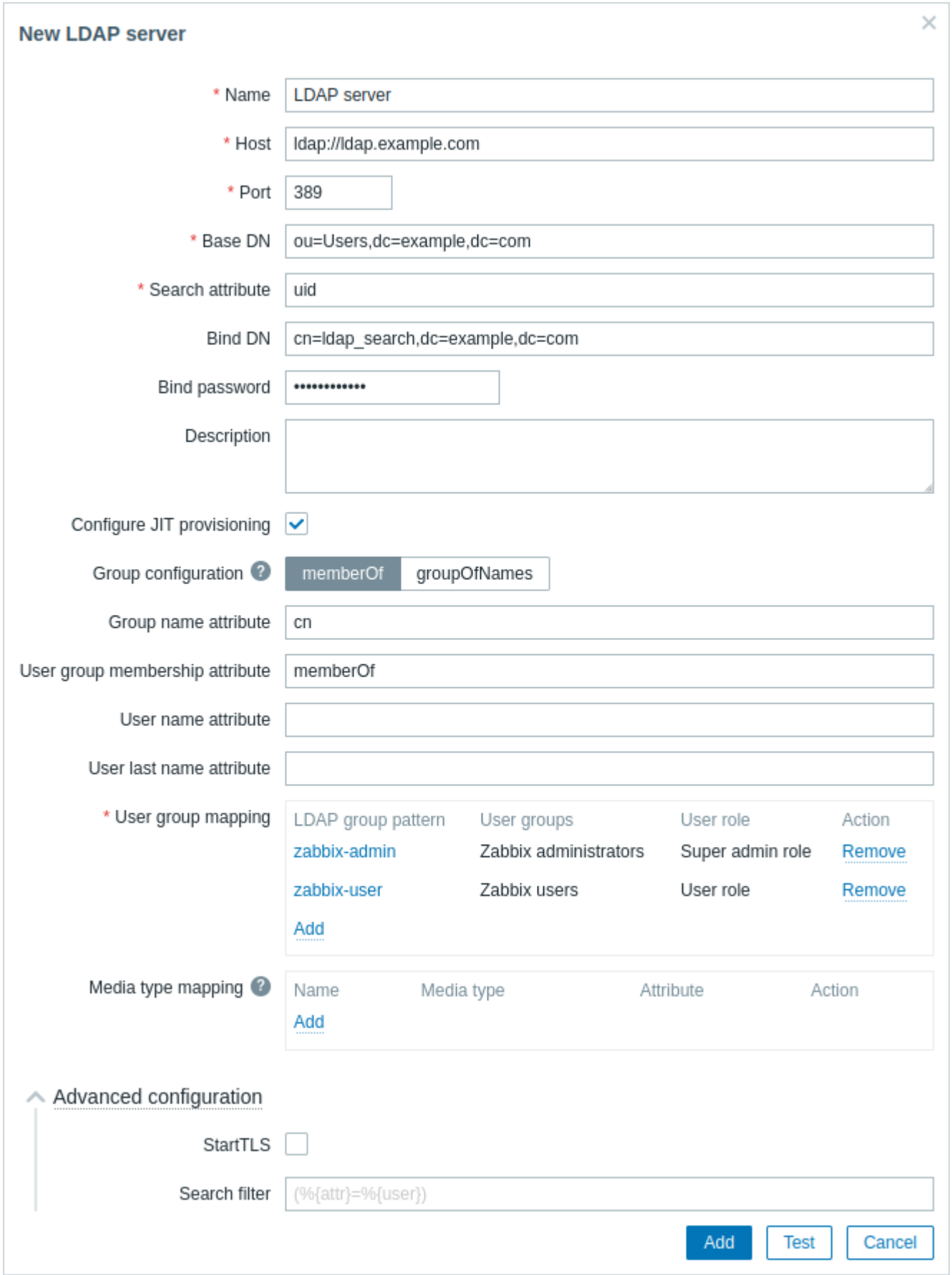
Konfigurācija



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 tālāk sadaļu LDAP servera konfigurācija).
Reģistrjutīga pieteikšanās	Noņemiet atzīmi no izvēles rūtiņas, lai atspējotu lietotājvārdu reģistrjutīgu pieteikšanos (pēc noklusējuma iespējota). Reģutīgas pieteikšanās atspējošana ļauj, piemēram, pieteikties kā ”admin”, pat ja Zabbix lietotājs ir ”Admin” vai ”ADMIN”. Ņemiet vērā: ja reģistrjutīga pieteikšanās ir atspējota un ir vairāki Zabbix lietotāji ar līdzīgiem lietotājvārdiem (piemēram, Admin un admin), pieteikšanās šiem lietotājiem vienmēr tiks liegta ar šādu kļūdas ziņojumu: ”Authentication failed: supplied credentials are not unique.”
Nodrošināšanas periods	Iestatiet nodrošināšanas periodu, t. i., cik bieži pieteicies lietotājs tiks nodrošināts, strādājot ar lietotāja saskarni.

LDAP servera konfigurācija



LDAP servera konfigurācijas parametri:

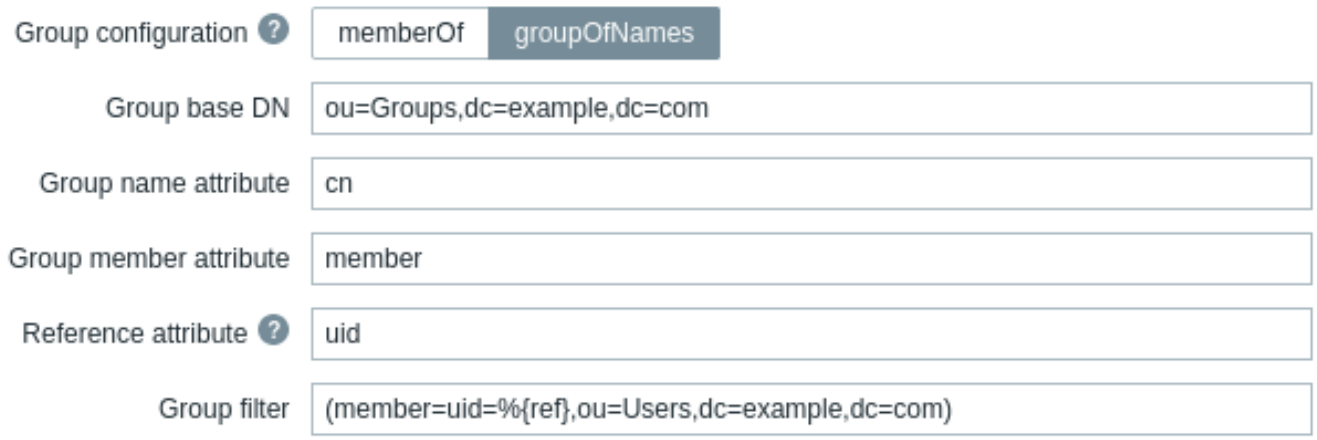
Parameter	Description
Name	LDAP servera nosaukums Zabbix konfigurācijā.

Parameter	Description
Host	LDAP servera resursdatora nosaukums, IP adrese 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 Izmantojot OpenLDAP 2.x.x un jaunāku versiju, var izmantot pilnu LDAP URI formā ldap://hostname:port vai ldaps://hostname:port.
Port	LDAP servera ports. Noklusējums 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 potenciāli atver domēna konfigurāciju neatļautiem lietotājiem (informācija par lietotājiem, datoriem, serveriem, grupām, pakalpojumiem utt.). Drošības apsvērumu dēļ atspējojiet anonīmas piesaistes LDAP resursdatoros un tā vietā izmantojiet autentificētu piekļuvi.
Bind password	LDAP konta parole 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 grupas konfigurācijas metodi: **memberOf** - meklējot lietotājus un viņu 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 `memberOf` atribūtā (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 (piem., `memberOf`). Piemēram, memberOf atribūts var saturēt šādu informāciju: `memberOf=cn=zabbix-admin,ou=Groups,dc=example,dc=com` Šis lauks ir pieejams tikai memberOf metodei.
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 modeli uz Zabbix lietotāju grupu un lietotāja lomu. Tas ir nepieciešams, lai noteiktu, kādu lietotāju grupu/lomu provisionētais lietotājs saņems Zabbix. Noklikšķiniet *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 visaugstākais atļauju līmenis.
Media type mapping	Kartējiet lietotāja LDAP media atribūtus (piem., e-pastu) uz Zabbix lietotāja medijiem paziņojumu sūtīšanai (atribūta vērtība tiek izmantota kā medija lauks *Send to*).
Advanced configuration	Noklikšķiniet uz galvenes *Advanced configuration*, lai parādītu papildu konfigurācijas opcijas (skatiet zemāk).
StartTLS	Atzīmējiet izvēles rūtiņu, lai, veidojot savienojumu ar LDAP serveri, izmantotu StartTLS darbību. Savienojums neizdosies, ja serveris neatbalsta StartTLS. StartTLS nevar izmantot ar serveriem, kas izmanto *ldaps* protokolu.

Parameter	Description
Search filter	Definējiet pielāgotu virkni, autentificējot lietotāju 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 noklusējumu: `(%{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. Autentificēšanās laikā vietturis %{user} tiks aizstāts ar lietotājvārdu, kas ievadīts pieteikšanās laikā.
 Izmantojot tiešo lietotāja piesaisti, Zabbix interaktīvās pieteikšanās laikā piekļūst tikai lietotāja akreditācijas datiem. Tāpēc provisioning uzdevumi, kas darbojas bez interaktīvas pieteikšanās (piemēram, poga *Provision now* vai `user.provision` API metode), ignorēs lietotājus, kuri autentificējas, izmantojot tiešo lietotāja piesaisti, jo Zabbix nevar piesaistīties viņu vārdā. Izmantojiet anonīmu piesaisti vai īpašu piesaistes lietotāju (pakalpojuma kontu), ja nepieciešams, lai provisioning un LDAP meklēšana darbotos no lietotāja saskarnes, neprasot, lai galalietotājs būtu pieteicies.

Tālāk norādītie lauki ir specifiski "groupOfNames" kā *Group configuration* metodei:



Parameter	Description
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 (piem., `member`).
Reference attribute	Norādiet grupas filtra atsauces atribūtu (skatiet lauku *Group filter*). Pēc tam grupas filtrā izmantojiet `%{ref}`, lai iegūtu vērtības šeit norādītajam atribūtam.
Group filter	Norādiet filtru, lai izgūtu grupu, kuras dalībnieks ir lietotājs. Piemēram, `(member=uid=%{ref},ou=Users,dc=example,dc=com)` atbildīs "User1", ja grupas atribūts member 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, `/etc/openldap/ldap.conf` konfigurācijas failā būs jāpievieno rinda `TLS_REQCERT allow`. Tas var samazināt savienojuma drošību ar LDAP katalogu.

Note:
Ieteicams izveidot atsevišķu LDAP kontu (*Bind DN*), lai veiktu piesaisti un meklēšanu LDAP serverī ar minimālām LDAP privilēģijām, nevis izmantotu reālus lietotāju kontus (kas tiek izmantoti pieteikšanās veikšanai Zabbix lietotāja saskarnē).
 Šāda pieeja nodrošina lielāku drošību un neprasa mainīt *Bind password*, kad lietotājs maina savu paroli LDAP serverī.
 Iepriekšējā tabulā tas ir konta nosaukums *ldap_search*.

Piekļuves pārbaude

Poga *Testēt* ļauj pārbaudīt lietotāja piekļuvi:

Parametrs	Apraksts
Pieteikšanās vārds	LDAP lietotāja vārds, ko pārbaudīt (iepriekš aizpildīts ar pašreizējā lietotāja vārdu no Zabbix lietotāja saskarnes). Šim lietotāja vārdam ir jābūt LDAP serverī. Zabbix neaktivizēs LDAP autentifikāciju, ja nevarēs autentificēt testa lietotāju.
Lietotāja parole	LDAP lietotāja parole, ko pārbaudīt.

SAML

Pārskats

SAML 2.0 autentifikāciju var izmantot, lai pieteiktos Zabbix.

Ja SAML pieteikšanās ir konfigurēta bez JIT lietotāju nodrošināšanas, lietotājam jau ir jāeksistē Zabbix; lietotāja Zabbix parole autentifikācijai netiek izmantota. Pēc veiksmīgas SAML autentifikācijas Zabbix salīdzina no SAML saņemto lietotājvārda atribūtu ar esošu lokālo Zabbix lietotājvārdu.

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 lietotāja saskarnē vēlaties izmantot SAML autentifikāciju, ir jāinstalē php-openssl.

Lai izmantotu SAML autentifikāciju, Zabbix jākonfigurē šādi:

1. Norādiet SP privāto atslēgu un sertifikātus. Lai aktivizētu SAML, izmantojot lietotāja saskarni, jānorāda IdP sertifikāts (ielīmējiet tekstu vai izvēlieties failu). Ja ir atlasīta opcija *Parakstīt* vai *Šifrēt*, jānorāda SP sertifikāts un SP privātā atslēga.

Ja lietotāja saskarnes konfigurācijas failā (`zabbix.conf.php`) ir iestatīts `$SSO['CERT_STORAGE'] = 'database'`, SAML konfigurēšanas laikā sertifikāta tekstu var ielīmēt vai sertifikāta failu augšupielādēt lietotāja saskarnē — faili failu sistēmā nav nepieciešami.

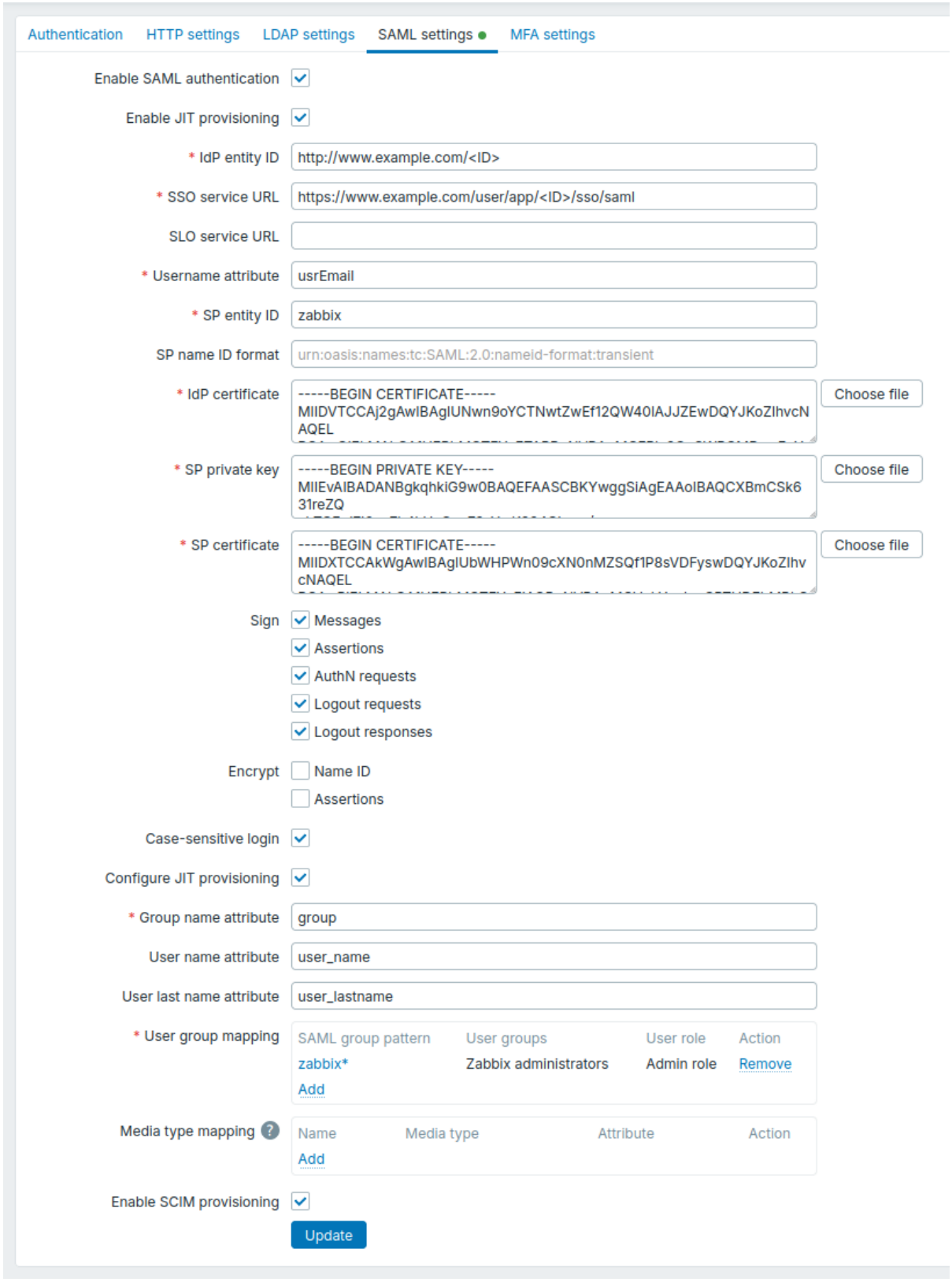
Ja konfigurācijas failā ir iestatīts `$SSO['CERT_STORAGE'] = 'file'`, sertifikātam jābūt pieejamam failu sistēmā (pēc noklusējuma mapē `ui/conf/certs` vai konfigurācijas failā norādītajā ceļā), un lietotāja saskarne sertifikātus datubāzē nesaglabās. Ņemiet vērā: ja `$SSO['CERT_STORAGE']` nav iestatīts vai ir komentēts, tiek pieņemta glabāšana failos.

Pēc noklusējuma Zabbix meklēs šādās atrašanās 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 materiālus var importēt, izmantojot lietotāja saskarni, ielīmējot tekstu vai augšupielādējot failus. Lai gan dažās vidēs ērtības labad ir atļauts importēt nešifrētus datus, stingri ieteicams izmantot šifrētu pārraidi (piemēram, iespējot HTTPS/TLS Zabbix lietotāja saskarnei). Ja SAML sertifikāti/atslēgas tiek glabātas datubāzē, iespējojiet TLS/SSL datubāzes savienojumiem, lai aizsargātu datplūsmu starp lietojumprogrammu un datubāzi.
 Šifrējiet datubāzes dublējumkopijas, kurās var būt SAML sertifikāti vai privātās atslēgas, un ierobežojiet piekļuvi dublējumkopiju failiem un datubāzes tabulām, kurās tiek glabāti SAML materiāli. Ierobežojiet datubāzes lietotāja privilēģijas līdz minimumam, kas nepieciešams.

2. Visus svarīgākos iestatījumus var konfigurēt Zabbix lietotāja saskarnē. Tomēr konfigurācijas failā ir iespējams norādīt papildu iestatījumus.



Zabbix lietotāja saskarnē pieejamie konfigurācijas parametri:

Parametrs	Apraksts
Iespējot SAML autentifikāciju	Atzīmējiet izvēles rūtiņu, lai iespējotu SAML autentifikāciju.
Iespējot JIT nodrošināšanu	Atzīmējiet izvēles rūtiņu, lai iespējotu JIT lietotāju nodrošināšanu.
IDP entītijas ID	Unikāls entītijas identifikators SAML identitātes nodrošinātājā.
SSO pakalpojuma URL	URL, uz kuru lietotāji tiks novirzīti, piesakoties.
SLO pakalpojuma URL	URL, uz kuru lietotāji tiks novirzīti, atsakoties. Ja lauks ir atstāts tukšs, SLO pakalpojums netiks izmantots.

Parametrs	Apraksts
Lietotājvārda atribūts	SAML atribūts, kas jāizmanto 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 entītijas ID	Unikāls pakalpojuma nodrošinātāja identifikators (ja tas nesakrīt, darbība tiks noraidīta). Var norādīt URL vai jebkuru datu virkni.
SP nosaukuma ID formāts	Atbildē pieprasiet noteiktu nosaukuma 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 sertifikāts	Pakalpojuma nodrošinātāja (SP) sertifikāta saturs SAML vienotās pieteikšanās (SSO) pakalpojuma iestatīšanai.
SP privātā atslēga	Pakalpojuma nodrošinātāja (SP) privātās atslēgas saturs SAML vienotās pieteikšanās (SSO) pakalpojuma iestatīšanai. Tā nodrošina drošu autentifikāciju un datu apmaiņu ar identitātes nodrošinātāju (IdP).
SP sertifikāts	Pakalpojuma nodrošinātāja (SP) sertifikāta saturs SAML vienotās pieteikšanās (SSO) pakalpojuma iestatīšanai.
Parakstīt	Noņemiet atzīmi no izvēles rūtiņām, lai atspējotu SAML parakstus attiecīgajām entītijām: *Ziņojumi* *Apliecinājumi* *AuthN pieprasījumi* *Atteikšanās pieprasījumi* *Atteikšanās atbildes*
Šifrēt	Atzīmējiet izvēles rūtiņas, lai atlasītu entītijas, kurām jāiespējo SAML šifrēšana: *Nosaukuma ID* *Apliecinājumi*
Reģistrjutīga pieteikšanās	Noņemiet atzīmi no izvēles rūtiņas, lai atspējotu lietotājvārdu reģistrjutīgu pieteikšanos (pēc noklusējuma iespējota). Reģistrjutīgas pieteikšanās atspējošana ļauj, piemēram, pieteikties kā "admin", pat ja Zabbix lietotājs ir "Admin" vai "ADMIN". Ņemiet vērā: ja reģistrjutīga pieteikšanās ir atspējota un ir vairāki Zabbix lietotāji ar līdzīgiem lietotājvārdiem (piemēram, Admin un admin), pieteikšanās šiem lietotājiem vienmēr tiks liegta ar šādu kļūdas ziņojumu: "Authentication failed: supplied credentials are not unique."
Konfigurēt JIT nodrošināšanu	Atzīmējiet šo izvēles rūtiņu, lai parādītu ar JIT lietotāju nodrošināšanu saistītās opcijas.
Grupas nosaukuma atribūts	Norādiet grupas nosaukuma atribūtu JIT lietotāju nodrošināšanai.
Lietotāja vārda atribūts	Norādiet lietotāja vārda atribūtu JIT lietotāju nodrošināšanai.
Lietotāja uzvārda atribūts	Norādiet lietotāja uzvārda atribūtu JIT lietotāju nodrošināšanai.
Lietotāju grupu kartēšana	Kartējiet SAML lietotāju grupas modeli uz Zabbix lietotāju grupu un lietotāja lomu. Tas ir nepieciešams, lai noteiktu, kādu lietotāju grupu/lomu nodrošinātais lietotājs saņems Zabbix. Noklikšķiniet uz *Pievienot*, lai pievienotu kartējumu. Lauks *SAML grupas modelis* atbalsta aizstājējzīmes. Grupas nosaukumam jāatbilst esošai grupai. 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.
Multivides veidu kartēšana	Kartējiet lietotāja SAML multivides atribūtus (piemēram, e-pastu) uz Zabbix lietotāja multividi paziņojumu sūtīšanai.

Parametrs	Apraksts
Iespējot SCIM nodrošināšanu	Atzīmējiet šo izvēles rūtiņu, lai iespējotu SCIM 2.0 nodrošināšanu.

Skatiet piemērus SAML identitātes nodrošinātāju konfigurēšanai pieteikšanās un lietotāju nodrošināšanas vajadzībām Zabbix:

- Microsoft Azure AD
- Okta
- Onelogin

Piezīmes par SCIM nodrošināšanu

SCIM nodrošināšanai identitātes nodrošinātāja pusē norādiet ceļu uz Zabbix lietotāja saskarni un pievienojiet tam `api_scim.php`, proti:

`https://<path-to-zabbix-ui>/api_scim.php`

Lietotāju atribūti, kas tiek izmantoti Zabbix (lietotājvārds, lietotāja vārds, lietotāja uzvārds un multivides atribūti), ir jāpievieno kā pielāgoti atribūti un, ja nepieciešams, ārējai nosaukumvietai jābūt tādai pašai kā lietotāja shēmai: `urn:ietf:params:scim:schemas:core`

Papildu iestatījumi

Papildu SAML parametrus var konfigurēt Zabbix lietotāja saskarnes konfigurācijas failā (`zabbix.conf.php`):

- `$SSO['SP_KEY']`
- `$SSO['SP_CERT']`
- `$SSO['IDP_CERT']`
- `$SSO['SETTINGS']`

Konfigurācijas piemēri

1. piemērs: Savienojuma izveide ar Zabbix instanci, izmantojot slodzes balansētāju ar TLS/SSL

Ja savienojuma izveidei ar Zabbix instanci tiek izmantots slodzes balansētājs un slodzes balansētājs izmanto TLS/SSL, bet Zabbix to neizmanto, norādiet parametrus `baseurl`, `strict` un `use_proxy_headers` šādi:

```
$SSO['SETTINGS'] = [
    'strict' => false,
    'baseurl' => 'https://zabbix.example.com/zabbix/',
    'use_proxy_headers' => true
];
```

2. piemērs: Pielāgotu SAML paraksta un īssavilkuma algoritmu konfigurēšana

```
$SSO['SETTINGS'] = [
    'security' => [
        'signatureAlgorithm' => 'http://www.w3.org/2001/04/xmldsig-more#rsa-sha384'
        'digestAlgorithm' => 'http://www.w3.org/2001/04/xmldsig-more#sha384',
        // ...
    ],
    // ...
];
```

3. piemērs: SSO konfigurēšana ar Kerberos autentifikāciju, izmantojot 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
```

MFA

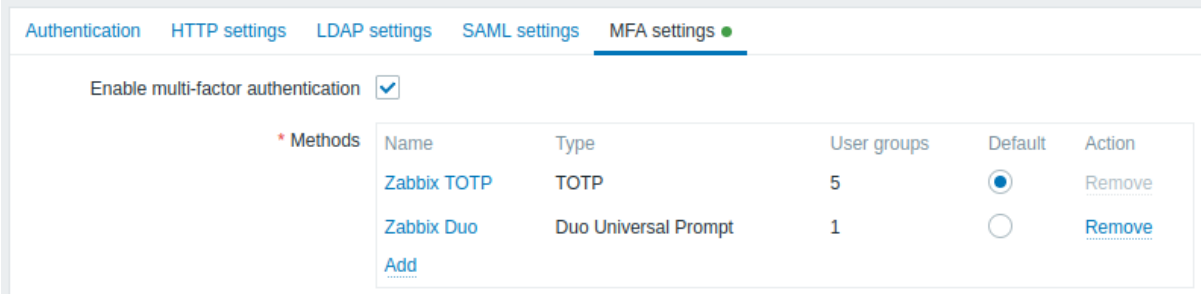
Pārskats

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.

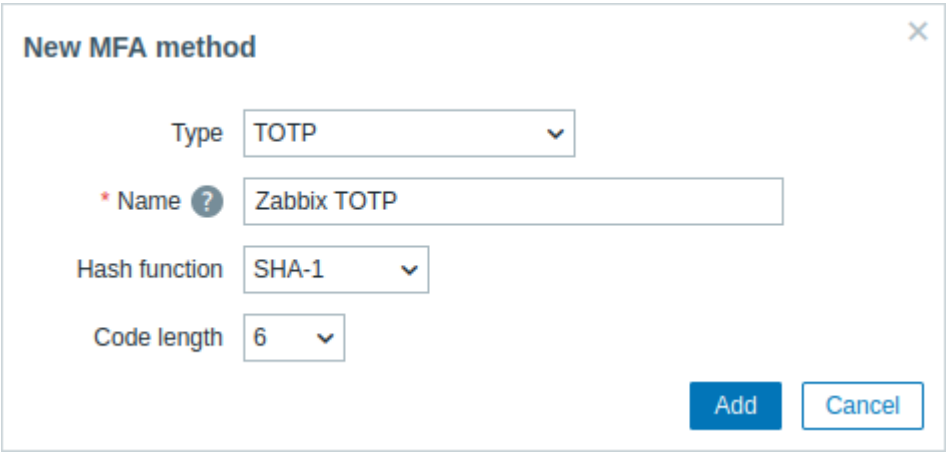
Konfigurācija



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”.

Parametrs	Apraksts
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

Lai izmantotu TOTP, lietotājiem ir jāapstiprina sava identitāte, izmantojot autentifikācijas lietotni (piemēram, Google Authenticator lietotni).

1. Dodieties uz MFA iestatījumiem Zabbix sadaļā *Users* > *Authentication* 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 *Add* un pēc tam uz pogas *Update*.

4. Dodieties uz *Users* > *User groups* un izveidojiet jaunu lietotāju grupu ar šādu konfigurāciju:

- Grupas nosaukums: TOTP group
- Lietotāji: Admin
- Daudzfaktoru autentifikācija: Default (vai ”Zabbix TOTP”, ja tā nav iestatīta kā noklusējuma)

5. Atteicieties 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 Google Authenticator lietotnē. Lietotne ģenerēs verifikācijas kodu, kas jāievada, lai pabeigtu pieteikšanās procesu.

7. Turpmākajām pieteikšanās reizēm iegūstiet verifikācijas kodu no Google Authenticator lietotnes un ievadiet to pieteikšanās laikā.

Duo Universal Prompt

Lai izmantotu Duo Universal Prompt, lietotājiem ir jāapstiprina sava identitāte, izmantojot autentifikācijas lietotni Duo Mobile.

Attention:
Duo Universal Prompt MFA metodei ir nepieciešama php-curl paplašinājuma instalēšana, piekļuve Zabbix, izmantojot HTTPS, un atļauja izejošajiem savienojumiem ar Duo serveriem. Turklāt, ja tīmekļa serverī ir iespējota Content Security Policy (CSP), pārliecinieties, ka savas virtuālā hosta konfigurācijas faila CSP direktīvai pievienojat "duo.com".

1. Reģistrējieties bezmaksas Duo administratora kontam vietnē Duo Signup.

2. Atveriet Duo Admin Panel, dodieties uz *Applications* > *Protect an Application*, meklējiet *Web SDK* lietotni un noklikšķiniet uz *Protect*.

3. Pierakstiet akreditācijas datus (Client ID, Client secret, API hostname), kas nepieciešami MFA metodes konfigurēšanai Zabbix.

4. Zabbix sadaļā *Users* > *Authentication* atveriet MFA iestatījumus un iespējojiet daudzfaktoru autentifikāciju.

5. Pievienojiet jaunu MFA metodi ar šādu konfigurāciju:

- Type: Duo Universal Prompt
- Name: Zabbix Duo
- API hostname: (use API hostname from Duo)
- Client ID: (use Client ID from Duo)
- Client secret: (use Client secret from Duo)

6. Noklikšķiniet uz pogas *Add* un pēc tam uz pogas *Update*.

7. Dodieties uz *Users* > *User groups* un izveidojiet jaunu lietotāju grupu ar šādu konfigurāciju:

- Group name: Duo group
- Users: Admin
- Multi-factor authentication: Default (or "Zabbix Duo" if it is not set as default)

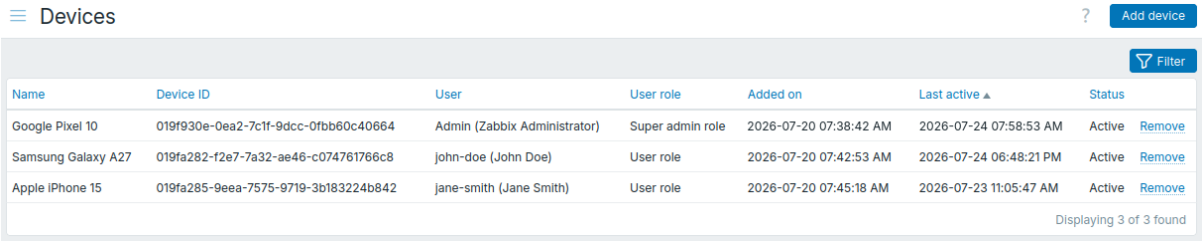
8. Izrakstieties no Zabbix un piesakieties atkārtoti, izmantojot savus akreditācijas datus. Pēc veiksmīgas pieteikšanās jums tiks piedāvāts reģistrēties MFA un jūs tiksiet novirzīts uz Duo. Pabeidziet Duo iestatīšanu un apstipriniet savu lietotāju ar tālruņa Duo lietotni, lai pieteiktos.

9. Turpmākajām pieteikšanām izmantojiet atbilstošo MFA metodi, ko nodrošina Duo lietotne (piemēram, verifikācijas koda iegūšanu, push paziņojumu apstiprināšanu vai hard keys izmantošanu), un pieteikšanās laikā ievadiet nepieciešamo informāciju.

Ierīces

Šī sadaļa ļauj superadministratoriem pārvaldīt mobilās ierīces, kas tiks izmantotas ar Zabbix mobilo lietotni.

Šī sadaļa tiek parādīta, ja ir iespējots `EnableMobileDevices`.



Lai pievienotu ierīci:

1. Noklikšķiniet uz *Add device*.
2. Atlasiet lietotāju un pēc tam noklikšķiniet uz *Add*. Lietotājam ir jābūt iespējotai piekļuvei ierīcēm.
3. Tiek parādīts QR kods. Lietotājam šis kods ir jāskenē savā Zabbix mobilajā lietotnē.

Note:
Kad pievienojat ierīci lietotājam, Zabbix lietotāja lietotāja multividei pievieno arī push paziņojumu multivides tipu. Kad ierīci noņemat, šī lietotāja multivide arī tiek noņemta. Ja ir konfigurēti vairāki push paziņojumu multivides tipi, Zabbix pievieno to, kas ir iespējots un alfabētiskā secībā atrodas pirmais.

Pievienotās ierīces tiek parādītas sarakstā, kuru varat filtrēt.

Varat arī noņemt ierīces. Kad ierīci noņemat, tās piekļuves akreditācijas dati tiek atsaukti un noņemti no Zabbix, pēc tam ierīce vairs nevar piekļūt Zabbix.

Sarakstā tiek parādīta šāda informācija:

Kolonna	Apraksts
Name	Ierīces nosaukums.
Device ID	Ierīces ID.
User	Lietotājs, kuram ierīce tika pievienota, vai *Inaccessible user*, ja šis lietotājs ir izdzēsts.
User role	Lietotāja loma.
Added on	Datums un laiks, kad ierīce tika pievienota.
Last active	Datums un laiks, kad ierīce pēdējo reizi bija aktīva.
Status	Ierīces statuss: **New** - ierīce tika pievienota, neskenējot QR kodu, un tā būs jāpievieno vēlreiz. **New (Token expired)** - ierīce tika pievienota, neskenējot QR kodu, un QR kodā ietvertā pilnvara ir beigusies; ierīce būs jāpievieno vēlreiz. **Active** - ierīce tika pievienota, QR kods tika noskenēts, un ierīce var izmantot Zabbix mobilo lietotni. **Unassigned** - ierīce tika pievienota, bet tās lietotājs tika izdzēsts, un ierīce nevar izmantot Zabbix mobilo lietotni.

Lietotāji ar pietiekamām atļaujām var pārvaldīt arī savas ierīces sadaļā *User settings* > *Devices*.

Administrēšana

Pārskats

Izvēlne Administration ir paredzēta Zabbix administratīvajām funkcijām. Šī izvēlne ir pieejama tikai SuperAdmin lietotāja tipa lietotājiem.

Datu avots

Pārskats

Sadaļā *Administrēšana* > *Datu avots* varat norādīt telemetrijas datu avotu, tas ir, datubāzi, kurā tiek glabāti telemetrijas dati (izsekojumi, metrikas un žurnāli). Telemetrijas dati tiek izmantoti lietojumprogrammu veiktspējas uzraudzībai (APM), kas palīdz noskaidrot, kā pieprasījumi plūst caur jūsu lietojumprogrammām un pakalpojumiem, novērtēt to veiktspēju un atrast lēnu vai neveiksmīgu darbību cēloni.

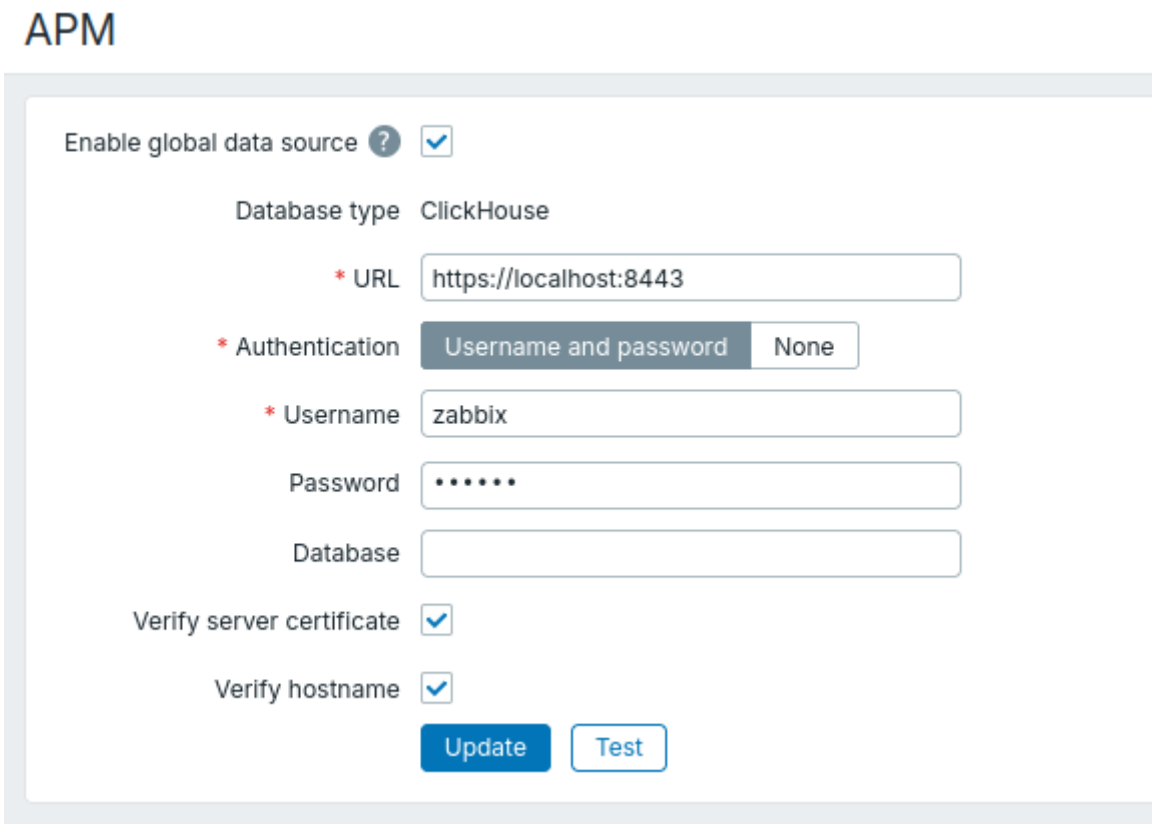
Telemetrijas datu avotam jāatbalsta OpenTelemetry.

Šajā sadaļā norādītais telemetrijas datu avots tiek saukts par globālo datu avotu, jo to koplieto Zabbix serveris, starpniekserveris un lietotāja saskarne. Tomēr varat to aizstāt jebkuram no šiem komponentiem, servera, starpniekservera vai lietotāja saskarnes konfigurācijā norādot citu telemetrijas datu avotu.

Vienums telemetrijas vaicājums vaicā šo datu avotu, ja vien komponenta konfigurācijā nav norādīts cits datu avots.

APM

Lai norādītu globālo datu avotu, atzīmējiet izvēles rūtiņu *Iespējot globālo datu avotu* un aizpildiet tālāk aprakstītos parametrus. Pirms saglabāšanas varat noklikšķināt uz *Testēt*, lai pārbaudītu savienojumu ar šo datu avotu (nepieciešams php-curl paplašinājums).



Parametrs	Apraksts
Datubāzes tips	Datubāzes tips ir kodēts kā *ClickHouse*, un to nevar mainīt. Skatiet arī: Prasības un ClickHouse iestatīšana.
URL	Ievadiet HTTP vai HTTPS URL, kas tiek izmantots savienojuma izveidei ar datubāzi.
Autentifikācija	Atlasiet autentifikācijas metodi, kas tiek izmantota, lai pieteiktos datubāzē: *Lietotājvārds un parole* vai *Nav*.
Lietotājvārds	Norādiet lietotājvārdu, kas tiek izmantots, lai pieteiktos datubāzē. Šis parametrs ir pieejams, ja parametram *Autentifikācija* ir iestatīta vērtība *Lietotājvārds un parole*.
Parole	Norādiet paroli, kas tiek izmantota, lai pieteiktos datubāzē, vai atstājiet lauku tukšu, ja datubāzes lietotājam nav paroles. Parole tiek atiestatīta, mainot URL. Šis parametrs ir pieejams, ja parametram *Autentifikācija* ir iestatīta vērtība *Lietotājvārds un parole*.
Datubāze	Ievadiet tās datubāzes nosaukumu, kurā tiek glabāti telemetrijas dati. Ja lauks ir tukšs, Zabbix izmanto ClickHouse lietotājam konfigurēto noklusējuma datubāzi.
Pārbaudīt servera sertifikātu	Atzīmējiet izvēles rūtiņu, lai pārbaudītu servera sertifikātu. Ja tā ir atzīmēta, saistītajām TLS opcijām jābūt iestatītām servera, starpniekservera vai lietotāja saskarnes konfigurācijā. Šis parametrs ir pieejams, ja *URL* satur HTTPS URL.
Pārbaudīt resursdatora nosaukumu	Atzīmējiet izvēles rūtiņu, lai pārbaudītu, vai *URL* norādītais resursdatora nosaukums atbilst servera sertifikātā norādītajam resursdatora nosaukumam. Ja tā ir atzīmēta, saistītajām TLS opcijām jābūt iestatītām servera, starpniekservera vai lietotāja saskarnes konfigurācijā. Šis parametrs ir pieejams, ja *URL* satur HTTPS URL.

Vispārīgi

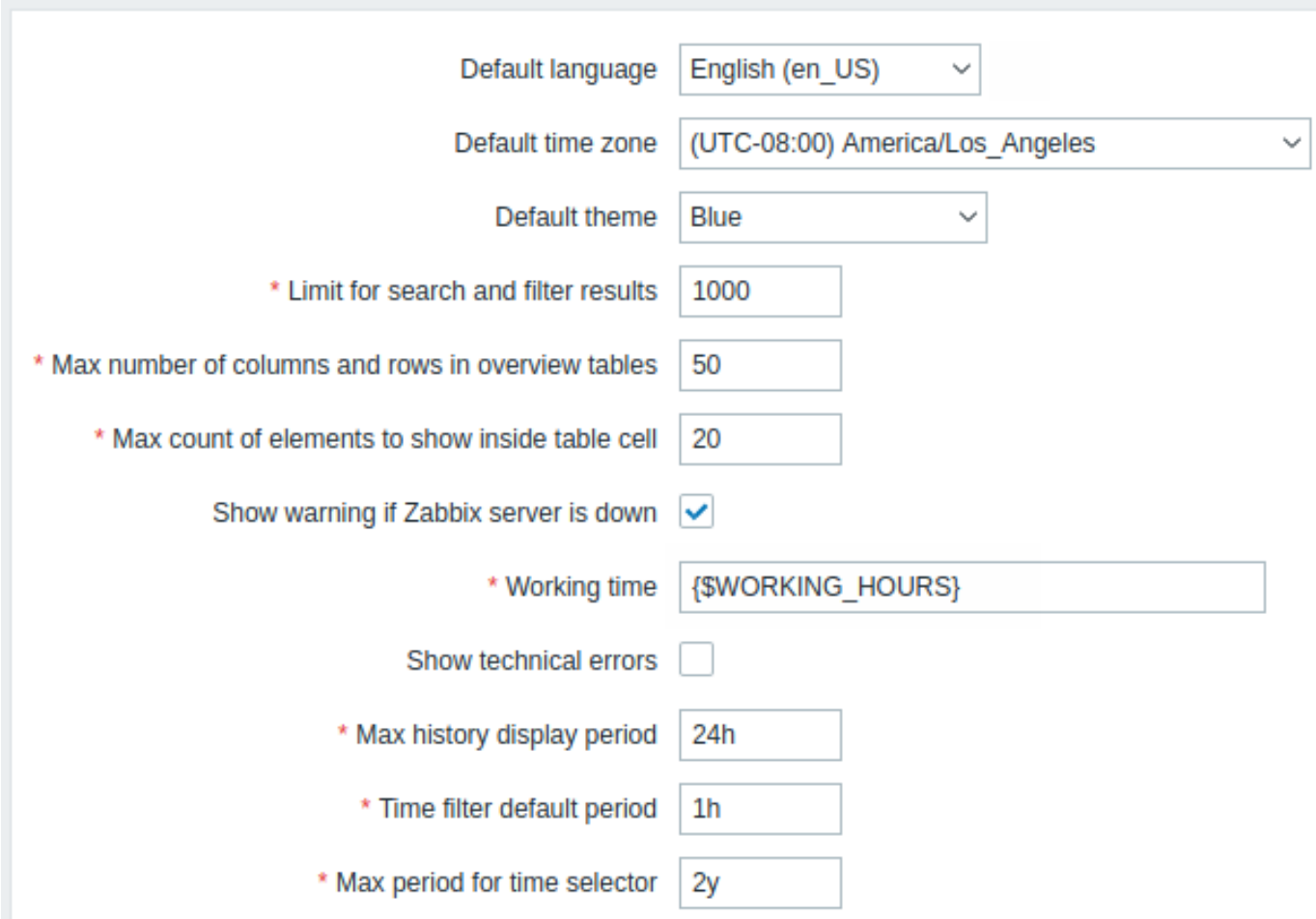
Pārskats

Sadaļā *Administration* > *General* ir vairākas apakšsadaļas, kas paredzētas lietotāja saskarnei saistīto noklusējuma iestatījumu konfigurēšanai un Zabbix pielāgošanai.

Pieejamo apakšsadaļu saraksts tiek parādīts, noklikšķinot uz *General* izvēlnes sadaļā *Administration*. Starp apakšsadaļām ir iespējams pārslēgties arī, izmantojot virsraksta nolaižamo izvēlni augšējā kreisajā stūrī.

GUI

Šī sadaļa nodrošina vairāku ar lietotāja saskarni saistītu noklusējuma iestatījumu pielāgošanu.



Konfigurācijas parametri:

Parameter	Description
Default language	Noklusējuma valoda lietotājiem, kuri savos profilos nav norādījuši valodu, un vieslietotājiem. Papildinformāciju skatiet sadaļā Papildu lietotāja saskarnes valodu instalēšana.
Default time zone	Noklusējuma laika josla lietotājiem, kuri savos profilos nav norādījuši laika joslu, un vieslietotājiem.
Default theme	Noklusējuma motīvs lietotājiem, kuri savos profilos nav norādījuši motīvu, un vieslietotājiem.
Limit for search and filter results	Maksimālais elementu (rindu) skaits, kas tiks parādīts lietotāja saskarnes sarakstā, piemēram, sadaļā *Datu vākšana* > *Hosts*. *Piezīme*: ja iestatīts, piemēram, ’50’, visos ietekmētajos lietotāja saskarnes sarakstos tiks parādīti tikai pirmie 50 elementi. Ja kāds saraksts satur vairāk nekā piecdesmit elementus, to norādīs ’+’ zīme tekstā *”Tiek parādīti 1 līdz 50 no atrastajiem **50+**”*. Tāpat, ja tiek izmantota filtrēšana un joprojām ir vairāk nekā 50 atbilstību, tiks parādīti tikai pirmie 50. Ņemiet vērā, ka šī parametra vērtības palielināšana var samazināt veiktspēju un palielināt atmiņas patēriņu tīmekļa servera pusē.
Max number of columns and rows in overview tables	Maksimālais kolonnu un rindu skaits, kas jāparāda informācijas paneļa logrīkā *Trigger overview*. Vienāds ierobežojums attiecas gan uz kolonnām, gan rindām. Ja ir vairāk rindu un/vai kolonnu, nekā tiek parādīts, sistēma tabulas apakšā parādīs brīdinājumu: ”Ne visi rezultāti tiek parādīti. Lūdzu, norādiet precīzākus meklēšanas kritērijus.”
Max count of elements to show inside table cell	Ierakstiem, kas tiek parādīti vienā tabulas šūnā, tiks rādīts ne vairāk kā šeit konfigurētais skaits.
Show warning if Zabbix server is down	Šis parametrs iespējo brīdinājuma ziņojuma parādīšanu pārlūkprogrammas logā, ja Zabbix serveris nav sasniedzams (iespējams, tas nedarbojas). Ziņojums paliek redzams pat tad, ja lietotājs ritina lapu uz leju. Novietojot kursoru virs tā, ziņojums īslaicīgi tiek paslēpts, lai atklātu zem tā esošo saturu.

Parameter	Description
Working time	Šis sistēmas mēroga parametrs nosaka darba stundas. Grafikos darba laiks tiek attēlots ar baltu fonu, bet ārpus darba laiks - ar pelēku fonu. Skatiet lapu Laika perioda specifikācija, lai uzzinātu laika formāta aprakstu. Tiek atbalstītas lietotāja makro.
Show technical errors	Ja atzīmēts, visi reģistrētie lietotāji varēs redzēt tehniskās kļūdas (PHP/SQL). Ja nav atzīmēts, informācija ir pieejama tikai Zabbix Super Admins un lietotājiem, kas pieder lietotāju grupām ar iespējotu atkļūdošanas režīmu.
Max history display period	Maksimālais laika periods, par kuru rādīt vēsturiskos datus sadaļā *Monitoring* > *Latest data*, hosts web scenario informāciju sadaļā *Monitoring* > *Hosts* un informācijas paneļa logrīkā *Top items*. Atļautais diapazons: 24 stundas (noklusējums) - 1 nedēļa. Tiek atbalstīti laika sufiksi, piemēram, 1w (viena nedēļa), 36h (36 stundas).
Time filter default period	Laika periods, kas pēc noklusējuma jāizmanto grafikos un informācijas 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 noklusējumu vai iepriekšējo lietotāja izvēli.
Max period for time selector	Maksimālais pieejamais laika periods grafikiem un informācijas paneļiem. Lietotāji nevarēs vizualizēt vecākus datus. Atļautais diapazons: 1 gads - 10 gadi (noklusējums: 2 gadi). Tiek atbalstīti laika sufiksi, piemēram, 1y (viens gads), 365w (365 nedēļas).

Autoreģistrācija

Šajā sadaļā varat konfigurēt šifrēšanas līmeni aktīvai aģenta autoreģistrācijai.



Ar zvaigznīti atzīmētie parametri ir obligāti.

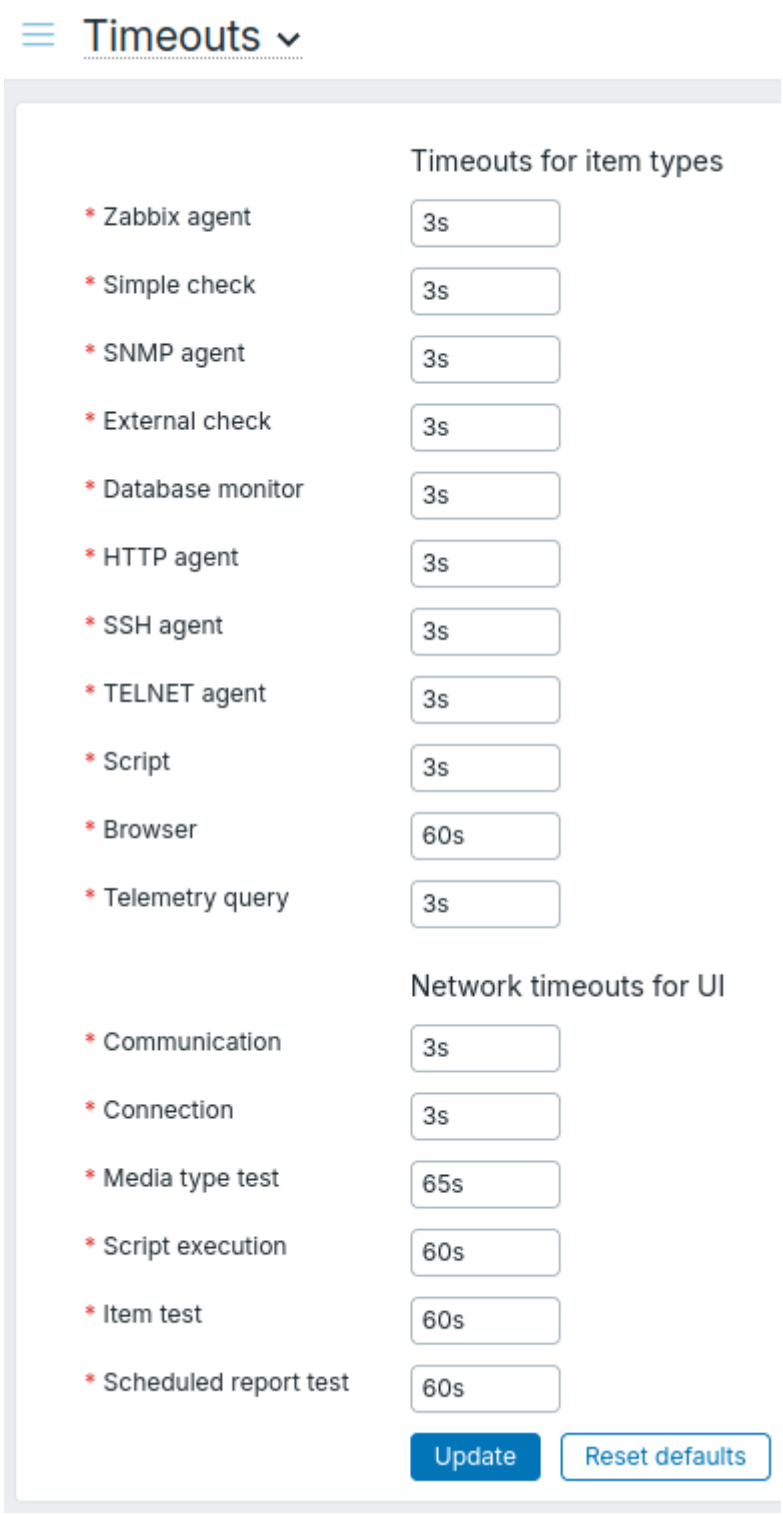
Konfigurācijas parametri:

Parameter	Description
Encryption level	Atlasiet vienu vai abas šifrēšanas līmeņa opcijas: **No encryption** - atļauti nešifrēti savienojumi **PSK** - atļauti ar TLS šifrēti savienojumi ar iepriekš koplietotu atslēgu
PSK identity	Ievadiet iepriekš koplietotās atslēgas identitātes virkni. Šis lauks ir pieejams tikai tad, ja kā *Encryption level* ir atlasīts 'PSK'. Neievadiet sensitīvu informāciju PSK identitātē, jo tā tiek pārsūtīta nešifrēti tīklā, lai saņēmējam norādītu, kuru PSK izmantot.
PSK	Ievadiet iepriekš koplietoto atslēgu (pāra skaits heksadecimālo rakstzīmju). Maksimālais garums: 512 heksadecimālās zīmes (256 baitu PSK), ja Zabbix izmanto GnuTLS vai OpenSSL bibliotēku, 64 heksadecimālās zīmes (32 baitu PSK), ja Zabbix izmanto mbed TLS (PolarSSL) bibliotēku. Piemērs: 1f87b595725ac58dd977beef14b97461a7c1045b9a1c963065002c5473194952 Šis lauks ir pieejams tikai tad, ja kā *Encryption level* ir atlasīts 'PSK'.

Skatiet arī: Droša autoreģistrācija

Noildzes

Šajā sadaļā ir iespējams iestatīt globālos vienumu tipu noildzes laikus un tīkla noildzes laikus. Visi šīs formas lauki ir obligāti.

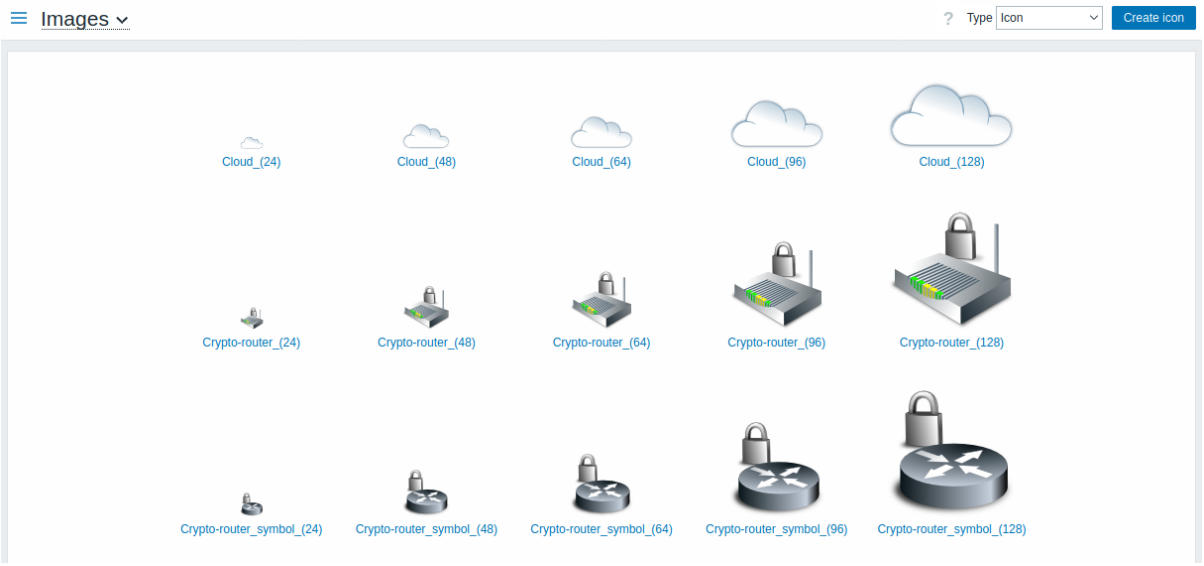


Parameter	Description
Noildzes vienumu tipiem	Cik sekundes Zabbix jāgaida pirms pārbaudes pārtraukšanas (atkarībā no tās tipa). Papildu informāciju un ierobežojumus skatiet vienuma noildze. Atļautais diapazons: 1 - 600 s (noklusējums: 3 s; noklusējums browser vienumiem: 60 s). Tiek atbalstīti laika sufiksi, piemēram, 30s, 1m, kā arī lietotāja makrosi. Ņemiet vērā, ka starpniekservera līmeņa noildzes, ja tās ir konfigurētas, pārraksta šos globālos iestatījumus. Turklāt, ja ir konfigurētas konkrētu vienumu noildzes, tās pārraksta gan globālos, gan starpniekservera iestatījumus.
Tīkla noildzes lietotāja saskarnei	

Parameter		Description
	Saziņa	Cik sekundes gaidīt pirms neaktīva ligzdas savienojuma aizvēršanas (ja savienojums ar Zabbix serveri iepriekš ir izveidots, bet lietotāja saskarne šajā laikā nevar pabeigt datu lasīšanas/sūtīšanas darbību, savienojums tiks pārtraukts). Atļautais diapazons: 1 - 300 s (noklusējums: 3 s).
	Savienojums	Cik sekundes gaidīt pirms mēģinājuma izveidot savienojumu ar Zabbix serveri pārtraukšanas. Atļautais diapazons: 1 - 30 s (noklusējums: 3 s).
	Multivides tipa tests	Cik sekundes gaidīt atbildi, testējot multivides tipu. Atļautais diapazons: 1 - 300 s (noklusējums: 65 s).
	Skripta izpilde	Cik sekundes gaidīt atbildi, izpildot skriptu. Atļautais diapazons: 1 - 300 s (noklusējums: 60 s). Šī noildze attiecas uz visu skripta izpildes ķēdi, kuras garums var būt dažāds. Piemēram, ja skripts tiek izpildīts uz aģenta, tas ir pilns pieprasījuma cikls caur serveri (iespējams, arī starpniekserveri) uz aģentu un atpakaļ. Skriptiem, kas tiek izpildīti uz aģenta, tiek piemērota servera vai starpniekservera noildze. Skriptiem, kas tiek izpildīti tikai uz aktīvā aģenta, visticamāk, būs jāpalielina noklusējuma servera/starpniekservera noildze. Servera/starpniekservera noildzei jābūt lielākai par aktīvās pārbaudes atsvaidzes biežumu, pretējā gadījumā servera/starpniekservera noildze var tikt pārsniegta, pirms aktīvais aģents vispār saņem skriptu. Skatiet arī: Skripta noildze.
	Vienuma tests	Cik sekundes gaidīt atgriezto datu saņemšanu, testējot vienumu. Atļautais diapazons: 1 - 600 s (noklusējums: 60 s).
	Plānotā pārskata tests	Cik sekundes gaidīt atgriezto datu saņemšanu, testējot plānoto pārskatu. Atļautais diapazons: 1 - 300 s (noklusējums: 60 s).

Attēli

Sadaļā Attēli tiek parādīti visi Zabbix pieejamie attēli.
Attēli tiek glabāti datubāzē.

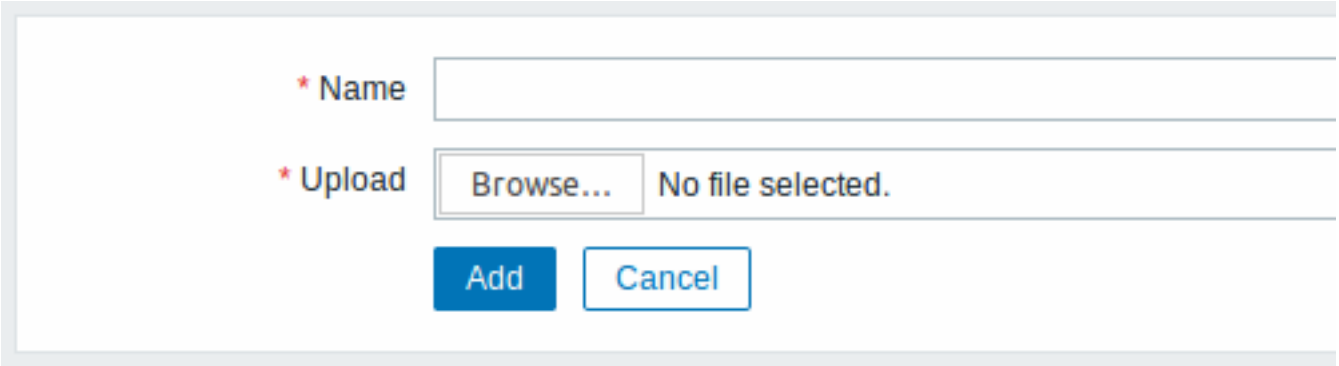


Nolaižamais saraksts *Tips* ļauj pārslēgties starp ikonu un fona attēliem:

- Ikonas tiek izmantotas tīkla karšu elementu attēlošanai.
- Foni tiek izmantoti kā tīkla karšu fona attēli.

Attēla pievienošana

Varat pievienot savu attēlu, augšējā labajā stūrī noklikšķinot uz pogas *Izveidot ikonu* vai *Izveidot fonu*.



Attēla atribūti:

Parametrs	Apraksts
Nosaukums	Unikāls attēla nosaukums.
Augšupielāde	Atlasiet failu (PNG, JPEG, GIF vai WebP) no lokālās sistēmas, lai to augšupielādētu Zabbix. *Ņemiet vērā*, ka var būt iespējams augšupielādēt arī citus formātus, kas augšupielādes laikā tiks konvertēti 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).

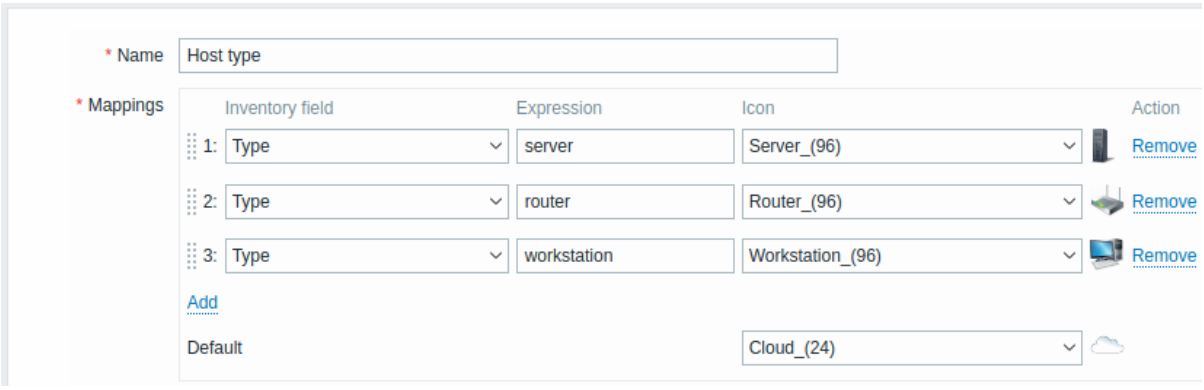
Note:
Augšupielādējamā faila maksimālo lielumu ierobežo `ZBX_MAX_IMAGE_SIZE` vērtība, kas ir 1024x1024 baiti jeb 1 MB.
Attēla augšupielāde var neizdoties, ja attēla lielums ir tuvu 1 MB un MySQL konfigurācijas parametram `max_allowed_packet` ir noklusējuma vērtība 1 MB.
Šādā gadījumā palieliniet parametra `max_allowed_packet` vērtību.

Ikonu kartēšana

Šī sadaļa ļauj izveidot noteiktu hosts kartēšanu ar noteiktām ikonām. Kartēšanas izveidei tiek izmantota hosts inventarizācijas lauka informācija.

Pēc tam kartējumus var izmantot tīkla kartes konfigurācijā, lai automātiski piešķirtu atbilstošas ikonas atbilstošajiem hosts.

Lai izveidotu jaunu ikonu karti, augšējā labajā stūrī noklikšķiniet uz *Izveidot ikonu karti*.



Konfigurācijas parametri:

Parameter	Description
Name	Ikonu kartes unikāls nosaukums.
Mappings	Kartējumu saraksts. Kartējumu secība nosaka, kuram būs prioritāte. Kartējumus var pārvietot sarakstā uz augšu un uz leju, izmantojot vilkšanu un nomešanu.
Inventory field	Hosts inventarizācijas lauks, kurā tiks meklēta atbilstība.
Expression	Regulārā izteiksme, kas apraksta atbilstību.
Icon	Ikona, ko izmantot, ja tiek atrasta atbilstība izteiksmei.
Default	Noklusējuma ikona, ko izmantot.

Regulārās izteiksmes

Šī sadaļa ļauj izveidot pielāgotas regulārās izteiksmes, kuras var izmantot vairākās vietās lietotāja saskarnē. Sīkāku informāciju skatiet sadaļā Regulārās izteiksmes.

Trigeru attēlošanas opcijas

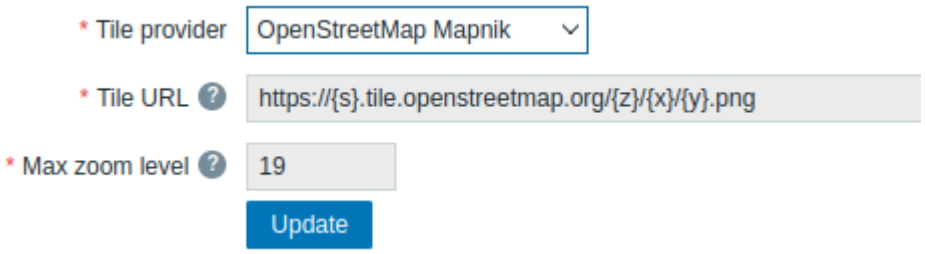
Šī sadaļa ļauj pielāgot, kā trigeru stāvoklis tiek attēlots lietotāja saskarnē, kā arī trigeru smaguma pakāpju nosaukumus un krāsas.



Parameter	Description
Use custom event status colors	Atzīmējot šo parametru, tiek ieslēgta apstiprinātu/neapstiprinātu problēmu krāsu pielāgošana.
Unacknowledged PROBLEM events, *Acknowledged PROBLEM events*, *Unacknowledged RESOLVED events*, *Acknowledged RESOLVED events*	Ievadiet jaunu krāsas kodu vai noklikšķiniet uz krāsas, lai no piedāvātās paletes izvēlētos citu. Ja ir atzīmēta izvēles rūtiņa *blinking*, trigeri pēc stāvokļa maiņas kādu laiku mirgos, lai tie būtu labāk pamanāmi.
Display OK triggers for	Laika periods, kurā tiek rādīti OK trigeri. Atļautais diapazons: 0 - 24 stundas. Tiek atbalstīti laika sufiksi, piemēram, 5m, 2h, 1d.
On status change triggers blink for	Trigeru mirgošanas ilgums. Atļautais diapazons: 0 - 24 stundas. Tiek atbalstīti laika sufiksi, piemēram, 5m, 2h, 1d.
Not classified, *Information*, *Warning*, *Average*, *High*, *Disaster*	Pielāgoti smaguma pakāpju nosaukumi un/vai krāsas, kas tiks rādītas sistēmas noklusējuma vietā. Ievadiet jaunu krāsas kodu vai noklikšķiniet uz krāsas, lai no piedāvātās paletes izvēlētos citu. Ņemiet vērā, ka šeit ievadītie pielāgotie smaguma pakāpju nosaukumi tiks izmantoti visās lokalizācijās. Ja tie ir jātulko citās valodās noteiktiem lietotājiem, skatiet lapu Customizing trigger severities.

Ģeogrāfiskās kartes

Šī sadaļa ļauj izvēlēties ģeogrāfisko karšu flīžu pakalpojuma sniedzēju un konfigurēt pakalpojuma sniedzēja iestatījumus Geomap paneļa sīkrīkam. 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, sniegto attēlu kvalitāti.



Parameter	Description
Tile provider	Izvēlieties vienu no pieejamajiem flīžu pakalpojumu sniedzējiem vai izvēlieties *Other*, lai pievienotu citu flīžu pakalpojumu sniedzēju vai pašmitinātas flīzes (skatiet Pielāgota flīžu pakalpojumu sniedzēja izmantošana).
Tile URL	URL veidne (līdz 2048 rakstzīmēm) flīžu slāņa ielādei un attēlošanai ģeogrāfiskajās kartēs. Šis lauks ir rediģējams tikai tad, ja *Tile provider* ir iestatīts uz *Other*. Tiek atbalstīti šādi vietturi: *{s}* apzīmē vienu no pieejamajiem apakšdomēniem; *{z}* apzīmē tālummaiņas līmeņa parametru URL; *{x}* un *{y}* apzīmē flīžu koordinātas; *{r}* var izmantot, lai URL pievienotu "@2x" un ielādētu retina flīzes. Piemērs: `https://{s}.example.com/{z}/{x}/{y}{r}.png`
Attribution text	Flīžu pakalpojumu sniedzēja atribūcijas teksts, kas tiks parādīts nelielā teksta lodziņā kartē. Šis lauks ir redzams tikai tad, ja *Tile provider* ir iestatīts uz *Other*.
Max zoom level	Kartes maksimālais tālummaiņas līmenis. Šis lauks ir rediģējams tikai tad, ja *Tile provider* ir iestatīts uz *Other*.

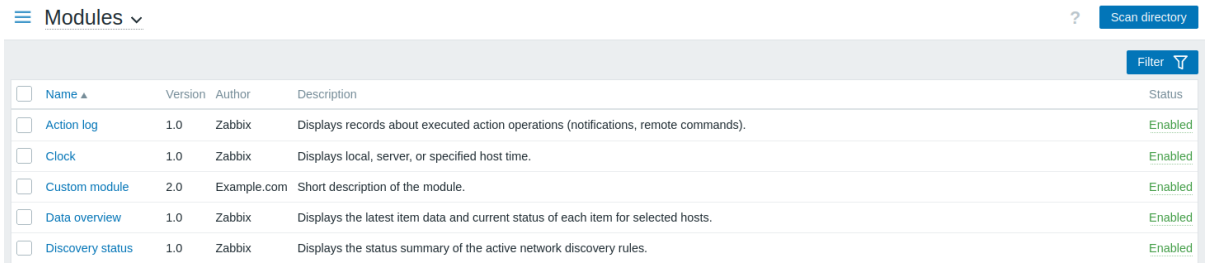
Izmantojot pielāgotu flīžu pakalpojuma sniedzēju

Geomap sīkrīks spēj ielādēt rastra flīžu attēlus no pielāgota pašmitināta vai trešās puses flīžu pakalpojuma sniedzēja. Lai izmantotu pielāgotu trešās puses flīžu pakalpojuma sniedzēja pakalpojumu vai pašmitinātu flīžu mapi vai serveri, laukā *Tile provider* atlasiet *Other* un laukā *Tile URL* norādiet pielāgoto URL, izmantojot atbilstošus vietturus.

Moduļi

Šajā sadaļā varat pārvaldīt pielāgotus, kā arī iebūvētos lietotāja saskarnes moduļus.

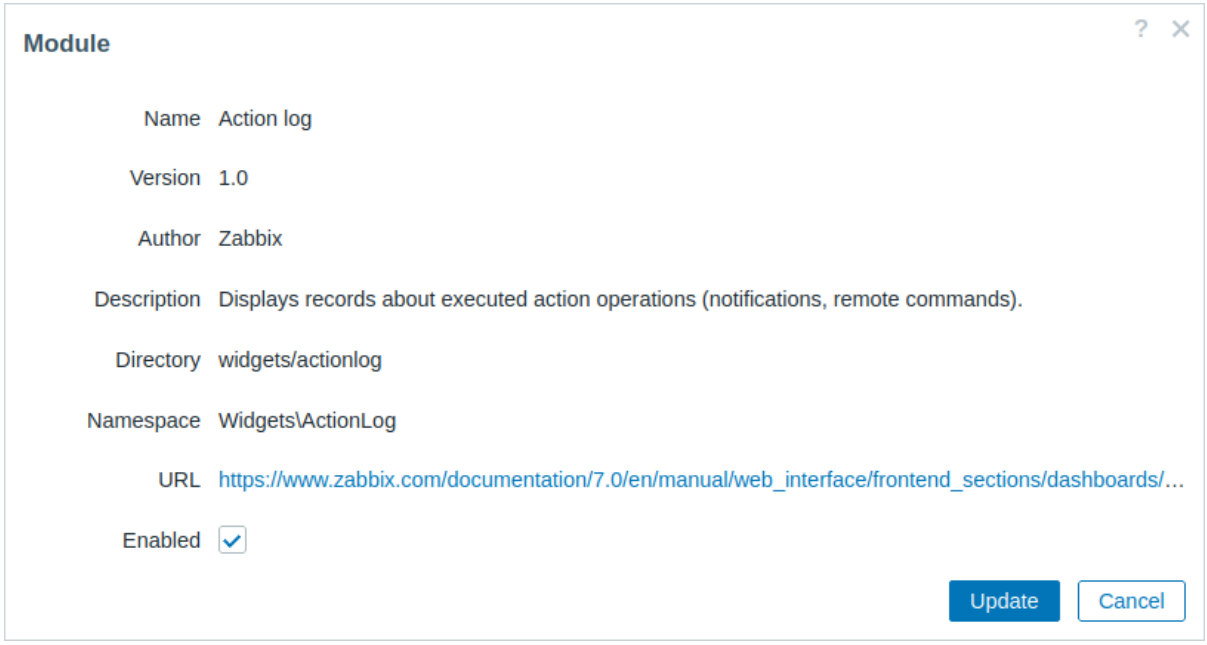
Note:
Ja `$ZBX_FEATURE_FLAGS['modules_config_enabled']` funkcionalitātes karodziņš lietotāja saskarnes konfigurācijas failā ir iestatīts uz `false`, piekļuve šai sadaļai ir atspējota.



Noklikšķiniet uz *Skenēt direktoriju*, lai reģistrētu vai atceltu jebkuru pielāgotu moduli. Reģistrētie moduļi tiks parādīti sarakstā; nereģistrētie moduļi tiks no saraksta noņemti.

Noklikšķiniet uz moduļa statusa sarakstā, lai moduli iespējotu vai atspējotu. Varat arī masveidā iespējot vai atspējot moduļus, atlasot tos sarakstā un pēc tam noklikšķinot uz pogas *Iespējot/Atspējot* zem saraksta.

Noklikšķiniet uz moduļa nosaukuma sarakstā, lai uznirstošajā logā skatītu tā informāciju.

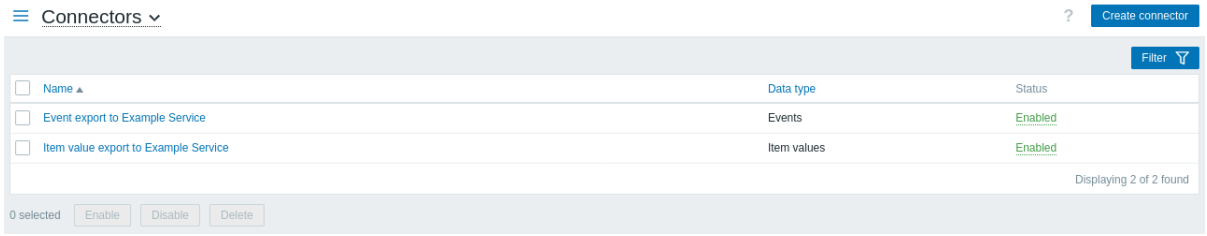


Moduļa statusu var atjaunināt arī moduļa informācijas uznirstošajā logā; lai to izdarītu, atzīmējiet vai noņemiet atzīmi no izvēles rūtiņas *Iespējots* un pēc tam noklikšķiniet uz *Atjaunināt*.

Moduļus var filtrēt pēc nosaukuma vai statusa (iespējots/atspējots).

Savienotāji

Šajā sadaļā varat konfigurēt savienotājus Zabbix datu straumēšanai uz ārējām sistēmām, izmantojot HTTP.



Noklikšķiniet uz *Izveidot savienotāju*, lai konfigurētu jaunu savienotāju.

Savienotājus varat filtrēt pēc nosaukuma vai statusa (iespējots/atspējots). Lai iespējotu vai atspējotu savienotāju, sarakstā noklikšķiniet uz tā statusa. Varat arī masveidā iespējot vai atspējot savienotājus, sarakstā tos atlasot un pēc tam zem saraksta noklikšķinot uz pogas *Iespējot/Atspējot*.

Citi

Šī sadaļa ļauj konfigurēt dažādus citus lietotāja saskarnes parametrus.



Parametrs	Apraksts
Lietotāja saskarnes URL	URL (līdz 2048 rakstzīmēm) uz Zabbix tīmekļa saskarni. 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 nepieejamības ziņojumam	Lietotāju grupa datubāzes nepieejamības trauksmes ziņojuma nosūtīšanai. Zabbix serveris ir atkarīgs no aizmugursistēmas datubāzes pieejamības. Tas nevar darboties bez datubāzes. Ja datubāze nav pieejama, Zabbix var paziņot izvēlētajiem lietotājiem. Paziņojumi tiks nosūtīti šeit iestatītajai 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 neapstāsies; 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	Reģistrēt SNMP slazdu, 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.

Slepeno datu glabāšana

Parameter	Description
Vault provider	Atlasiet slepeno datu pārvaldības programmatūru lietotāja makro vērtību glabāšanai - *HashiCorp Vault* (noklusējums) vai *CyberArk Vault*.
Resolve secret vault macros by	Atrisināt slepeno datu glabātuves makro, izmantojot: **Zabbix serveris** - slepenie dati tiek izgūti ar Zabbix serveris un, ja nepieciešams, pārsūtīti starpniekserveriem (noklusējums); **Zabbix serveris un starpniekserveris** - slepenie dati tiek izgūti gan ar Zabbix serveris, gan ar starpniekserveriem, ļaujot tiem neatkarīgi atrisināt makro.

Skatiet arī: Slepeno datu glabāšana.

Drošība

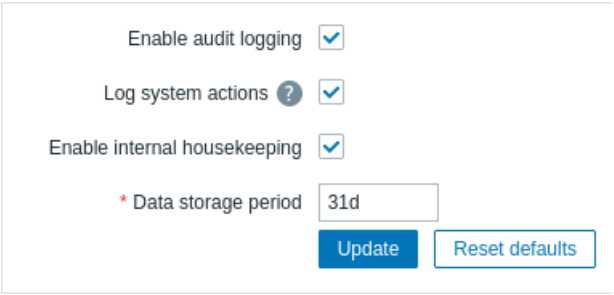
Parametrs	Apraksts
Validate URI schemes	Noņemiet atzīmi no šīs izvēles rūtiņas, lai atspējotu URI shēmu validāciju (pēc noklusējuma iespējota). Ja atzīmēta, varat norādīt ar komatiem atdalītu atļauto URI shēmu sarakstu (pēc noklusējuma: http,https,ftp,file,mailto,tel,ssh). Attiecas uz visiem laukiem 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.

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.

Tīrīšana

Pārskats

Sadaļa *Administrēšana > Housekeeping* ļauj konfigurēt Zabbix iekšējo housekeeping, kas automātiski noņem novecojušus vai lietotāja dzēstus datus no datubāzes. Tas novērš datubāzes bezgalīgu pieaugumu, kas patērētu nevajadzīgu diska vietu un paslik-tinātu datubāzes veiktspēju.

Housekeeping var iespējot un konfigurēt šādiem datu veidiem:

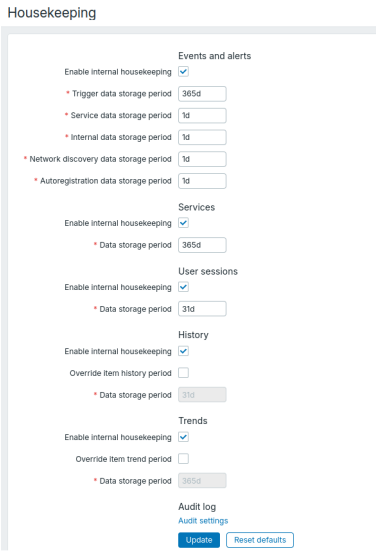
- Notikumi un brīdinājumi no trigeriem, pakalpojumiem, iekšējiem datiem, tīkla atklāšanas un automātiskās reģistrācijas
- Pakalpojumi

- Lietotāju sesijas
- Vēsture un tendences vienumiem

Note:
Auditācijas žurnāla housekeeping tiek konfigurēts atsevišķi.

Konfigurācija

Ir pieejami šādi parametri:



Parameter	Description
Enable internal housekeeping	Iespējot vai atspējot iekšējo housekeeper procesu (pēc noklusējuma iespējots). Ja tas ir iespējots, housekeeping procedūra automātiski noņem no datubāzes datus, kas pārsniedz *Data storage period*.
Data storage period	Norādiet, cik ilgi saglabāt datus, pirms housekeeper tos noņem. Obligāts, ja iekšējais housekeeping ir iespējots. Diapazons: 1 diena (1 stunda vēsturei) - 25 gadi; vai ”0”. Tiek atbalstīti laika sufiksi (piem., 1d, 1w). Parametram *Events and alerts* 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/atjaunošanās notikums, kas ir vecāks par *Data storage period*, netiks noņemts, ja tas ir saistīts ar neatrisinātu problēmu. Kad housekeeper noņem novecojušus objektus, tas vispirms noņem problēmas, pēc tam notikumus. Parametriem *History* un *Trends* datu glabāšanas periodi arī nosaka, cik ilgi dati ir redzami *Monitoring* > *Latest data*, pat ja iekšējais housekeeping ir atspējots. Ņemiet vērā, ka, dzēšot vienumu vai trigeri, tiek dzēstas arī tā problēmas un saistītie notikumi.
Override item history period	Ja iespējots, vienuma konfigurācijā norādītais vēstures/tendenču glabāšanas periods tiek pārrakstīts ar iestatījumu *Data storage period* (izņemot vienumus, kuriem ir iespējota opcija *Do not store*). Šo opciju var izmantot arī tad, ja iekšējais housekeeping ir atspējots un tiek izmantots ārējs housekeeper.
Override item trend period	

Ja tiek izmantots **ClickHouse**, kļūst pieejama sadaļa *ClickHouse data storage period*. Tā parāda datu glabāšanas periodu katram vienuma vērtības tipam. Lai mainītu periodu, skatiet ClickHouse datu glabāšanas perioda konfigurēšanu. Ņemiet vērā, ka housekeeper nedzēš datus no ClickHouse.

Ja tiek izmantots **Elasticsearch**, kļūst pieejama sadaļa *Elasticsearch data storage period*. Tā parāda vienuma vērtības tipus, kas saglabāti Elasticsearch, un datu glabāšanas periods tiek rādīts kā *Unknown*. Lai kontrolētu, cik ilgi dati tiek saglabāti, skatiet Index

Lifecycle Management. Ņemiet vērā, ka housekeeper nedzēš datus no Elasticsearch.

Ja tiek izmantots **TimescaleDB**, kļūst pieejama sadaļa *History, trends and audit log compression*.

Note:
TimescaleDB gadījumā iespējojiet *Override item history period*, *Override item trend period* un *Enable internal housekeeping* vēsturei un tendencēm, lai pilnībā izmantotu automātiskās sadalīšanas priekšrocības. Ja šīs opcijas ir atspējotas, dati, kas tiek glabāti vēstures un tendenču tabulās, joprojām tiks sadalīti, taču housekeeper nenometīs novecojušos sadalījumus, un tiks parādīti konfigurācijas brīdinājumi. Kad novecojušo sadalījumu nomešana ir iespējota, Zabbix serveris un lietotāja saskarne vairs nesekos dzēstajiem vienumiem, un šo vienumu vēsture tiks notīrīta, kad tiks dzēsts novecojis sadalījums.

Pēc noklusējuma housekeeper tiek izpildīts automātiski katru stundu. Izpildes biežumu var mainīt, iestatot Zabbix servera konfigurācijas parametru `HousekeepingFrequency`.

Poga *Reset defaults* ļauj atjaunot visas veiktās izmaiņas.

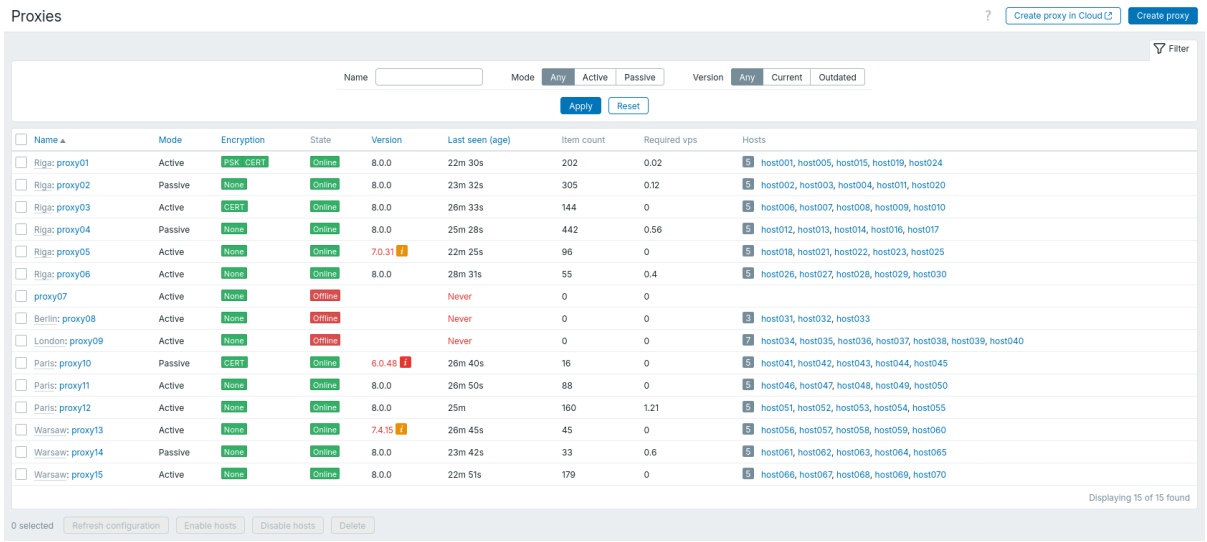
Starpniekserveri

Pārskats

Sadaļā *Administrēšana* > *Starpniekserveri* Zabbix lietotāja saskarnē var konfigurēt starpniekserverus izplatītai uzraudzībai.

Starpniekserveri

Tiek parādīts esošo starpniekserveru saraksts ar to detalizētu informāciju.



Parādītie dati:

Kolonna	Apraksts
Name	Starpniekservera nosaukums. Noklikšķinot uz starpniekservera nosaukuma, tiek atvērta starpniekservera konfigurācijas veidlapa. Ja starpniekserveris pieder starpniekserveru grupai, pirms starpniekservera nosaukuma kā pelēka saite tiek parādīts grupas nosaukums. Noklikšķinot uz grupas nosaukuma, tiek atvērta starpniekserveru grupas konfigurācijas veidlapa.
Mode	Starpniekservera režīms - *Active* vai *Passive*.
Encryption	Savienojumu no starpniekservera šifrēšanas statuss: **None** - šifrēšana netiek izmantota; **PSK** - tiek izmantota iepriekš koplietota atslēga; **Cert** - tiek izmantots sertifikāts.
State	Starpniekservera stāvoklis: **Unknown** - starpniekserveris tika izveidots, kamēr Zabbix serveris nedarbojās, vai serveris vēl nav atjauninājis stāvokli; **Online** - starpniekserveris ir sazinājies ar Zabbix serveri (pasīvais starpniekserveris atbildēja uz servera pieprasījumu; aktīvais starpniekserveris nosūtīja pieprasījumu) atteices pārslēgšanas periodā; **Offline** - starpniekserveris nav sazinājies ar Zabbix serveri atteices pārslēgšanas periodā.

Kolonna	Apraksts
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 konstatēja starpniekserveri.
Item count	Iespējoto vienumu skaits iespējotajos hosts, kas piešķirti starpniekserverim.
Required vps	Nepieciešamā starpniekservera veiktspēja (to vērtību skaits, kas jāapkopo sekundē).
Hosts	Iespējoto hosts skaits, kas piešķirti starpniekserverim, un hosts saraksts, kurus uzrauga starpniekserveris. Noklikšķinot uz hosts nosaukuma, tiek atvērta hosts konfigurācijas veidlapa.

Lai konfigurētu jaunu starpniekserveri, augšējā labajā stūrī noklikšķiniet uz pogas *Create proxy*. Lai izvietotu jaunu starpniekserveri Zabbix Cloud, blakus tai noklikšķiniet uz pogas *Create proxy in Cloud*. Tādējādi jaunā pārlūkprogrammas cilnē tiek atvērts Zabbix Cloud, kur varat izveidot starpniekservera mezglu. Kad starpniekservera mezgls ir izveidots, konfigurējiet starpniekserveri, izmantojot pogu *Create proxy*. Papildinformāciju skatiet sadaļā Zabbix Cloud starpniekservera iestatīšana.

Masveida rediģēšanas opcijas

Pogas zem saraksta piedāvā dažas masveida rediģēšanas opcijas:

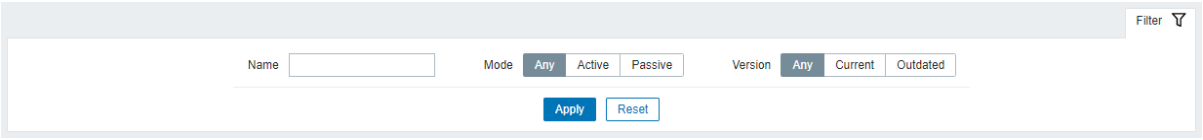
- *Konfigurācijas atsvaidzināšana* - atsvaidzina starpniekserveru konfigurāciju;
- *Iespējot hosts* - maina to hosts statusu, kurus uzrauga starpniekserveris, uz *Uzraudzīts*;
- *Atspējot hosts* - maina to hosts statusu, kurus uzrauga starpniekserveris, uz *Netiek uzraudzīts*;
- *Dzēst* - dzēš starpniekserverus.

Lai izmantotu šīs opcijas, atzīmējiet izvēles rūtiņas attiecīgo starpniekserveru priekšā, pēc tam noklikšķiniet uz vajadzīgās pogas.

Filtra izmantošana

Varat izmantot filtru, lai parādītu tikai jūs interesējošos starpniekserverus. Lai uzlabotu meklēšanas veiktspēju, dati tiek meklēti ar neatrisinātiem makro.

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.



Starpniekserveru grupas

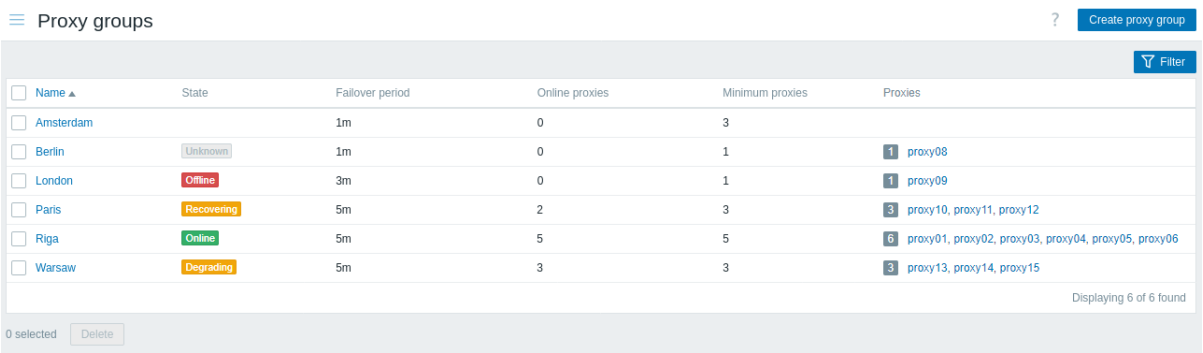
Pārskats

Sadaļā *Administration* > *Proxy groups* ir iespējams konfigurēt starpniekserveru grupas.

Starpniekserveru grupas tiek izmantotas starpniekserveru slodzes balansēšanā ar automatizētu hostu sadali starp starpniekserveriem un augstu pieejamību starp starpniekserveriem.

Starpniekserveru grupas

Tiek parādīts esošo starpniekserveru grupu saraksts ar to detalizētu informāciju.



Parādītie dati:

Kolonna	Apraksts
Nosaukums	Starpniekserveru grupas nosaukums. Noklikšķinot uz starpniekserveru grupas nosaukuma, tiek atvērta starpniekserveru grupas konfigurācijas veidlapa.
Stāvoklis	Starpniekserveru grupas stāvoklis (tiek parādīts, ja grupā ir vismaz viens starpniekserveris): **Nezināms** - grupa tika izveidota, kamēr Zabbix serveris nedarbojās, vai serveris vēl nav atjauninājis stāvokli; **Tiešsaistē** - minimālais starpniekserveru skaits kļūmjpārlēces periodā ir sazinājies ar Zabbix serveri; **Pasliktinās** - grupa drīzumā pāries bezsaistē, jo tiešsaistē esošo starpniekserveru skaits ir kļuvis mazāks par slieksni *Minimālais starpniekserveru skaits*; **Bezsaistē** - kļūmjpārlēces periodā ar Zabbix serveri ir sazinājies mazāks par minimālo starpniekserveru skaitu; **Atkopjas** - grupa drīzumā atgriezīsies tiešsaistē, jo tiešsaistē esošo starpniekserveru skaits ir sasniedzis slieksni *Minimālais starpniekserveru skaits*. Starpniekserveru grupas stāvokli var pārraudzīt, izmantojot vienumus zabbix[proxy group,<name>,state] un zabbix[proxy group,discovery].
Kļūmjpārlēces periods	Periods, kurā starpniekserveru grupas starpniekserverim ir jāsazinās ar Zabbix serveri, lai tas tiktu uzskatīts par tiešsaistē esošu.
Tiešsaistē esošie starpniekserveri	Tiešsaistē esošo starpniekserveru skaits (tiek parādīts sarkanā krāsā, ja tas ir mazāks par grupai noteikto minimumu).
Minimālais starpniekserveru skaits	Minimālais tiešsaistē esošo starpniekserveru skaits, kas nepieciešams, lai starpniekserveru grupa paliktu tiešsaistē.
Starpniekserveri	Grupā esošo starpniekserveru skaits un saraksts ar grupā esošajiem starpniekserveriem, kurā ir saites uz starpniekservera konfigurācijas veidlapu. Parādīto starpniekserveru maksimālo skaitu ierobežo vērtība *Maksimālais elementu skaits, ko rādīt tabulas šūnā*.

Lai konfigurētu jaunu starpniekserveru grupu, augšējā labajā stūrī noklikšķiniet uz pogas *Izveidot starpniekserveru grupas*.

Masveida rediģēšanas opcijas

Pogas zem saraksta piedāvā dažas masveida rediģēšanas opcijas:

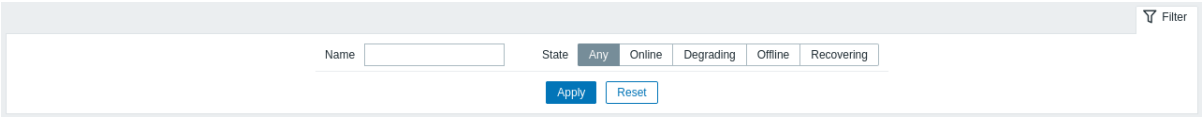
- *Dzēst* - dzēst starpniekserveru grupas.

Lai izmantotu šīs opcijas, atzīmējiet izvēles rūtiņas attiecīgo starpniekserveru grupu priekšā, pēc tam noklikšķiniet uz vajadzīgās pogas.

Filtra izmantošana

Varat izmantot filtru, lai parādītu tikai tos starpniekserveru grupas, kas jūs interesē. Lai nodrošinātu labāku meklēšanas veiktspēju, dati tiek meklēti ar neatrisinātiem makro.

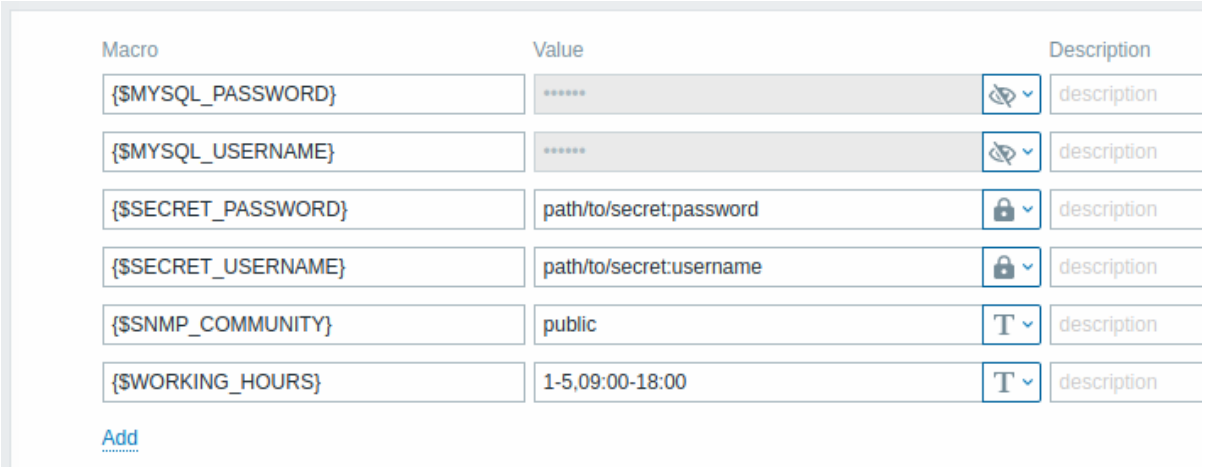
Saite *Filter* ir pieejama virs starpniekserveru grupu saraksta. Noklikšķinot uz tās, kļūst pieejams filtrs, kurā varat filtrēt starpniekserveru grupas pēc nosaukuma un statusa.



Makrokomandas

Pārskats

Šajā sadaļā varat definēt visas sistēmas lietotāja makro kā nosaukuma un vērtības pārus. Ņemiet vērā, ka makro vērtības var tikt saglabātas kā vienkāršs teksts, slepens teksts vai Vault noslēpums. Tiek atbalstīta arī apraksta pievienošana.



Rinda

Pārskats

Sadaļā *Administrēšana* > *Rinda* tiek parādīti vienumi, kas gaida atjaunināšanu.

Ideālā gadījumā, atverot šo sadaļu, tai visai vajadzētu būt ”zaļai”, kas nozīmē, ka rindā nav neviena vienuma. Ja visi vienumi tiek atjaunināti bez aizkaves, rindā nav neviena gaidoša vienuma. Tomēr nepietiekamas servera veiktspējas dēļ dažu vienumu atjaunināšana var aizkavēties, un informācija par to tiek parādīta šajā sadaļā. Plašāku informāciju skatiet sadaļā Rinda.

Note:
Zabbix starpniekserveris nezina par uzturēšanas periodiem; plašāku 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 aģents nedarbojas pareizi.

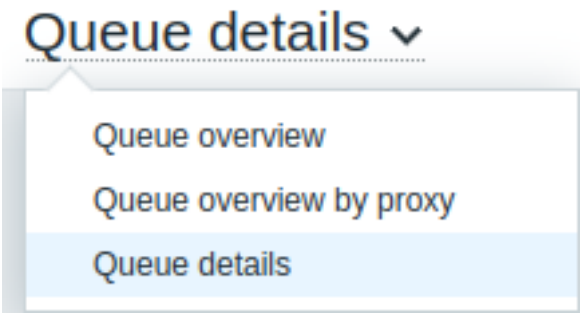
Sadaļā *Administrēšana* > *Rinda* ir pieejamas šādas lapas:

- Rindas pārskats - parāda rindu pēc vienuma tipa;
- Rindas pārskats pēc starpniekservera - parāda rindu pēc starpniekservera;
- Rindas informācija - parāda aizkavēto vienumu sarakstu.

Pieejamo lapu saraksts tiek parādīts, noklikšķinot uz *Rinda Administrēšana* izvēlnes sadaļā. Starp lapām var pārslēgties arī, izmantojot virsraksta nolaižamo sarakstu augšējā kreisajā stūrī.



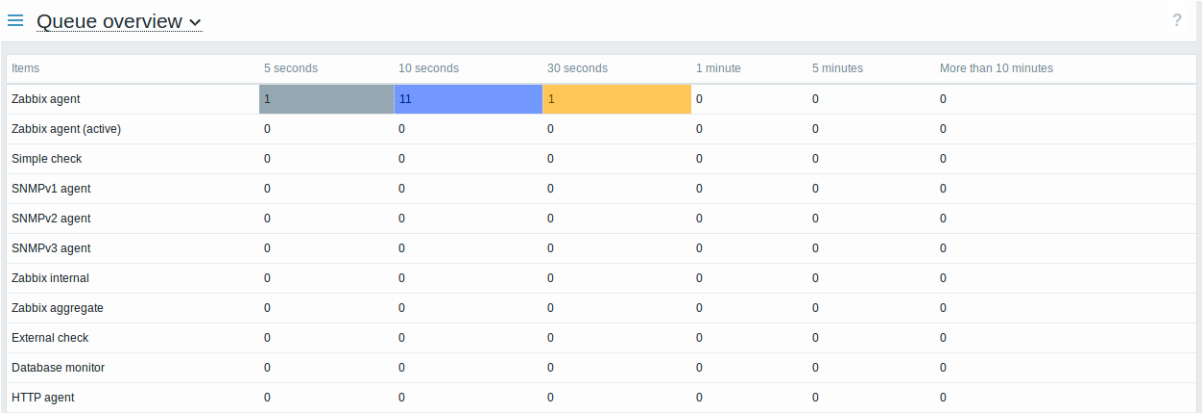
Trešā līmeņa izvēlne.



Virsraksta nolaižamais saraksts.

Pārskats pēc vienuma tipa

Šajā ekrānā ir viegli noteikt, vai problēma ir saistīta ar vienu vai vairākiem vienumu tipiem.

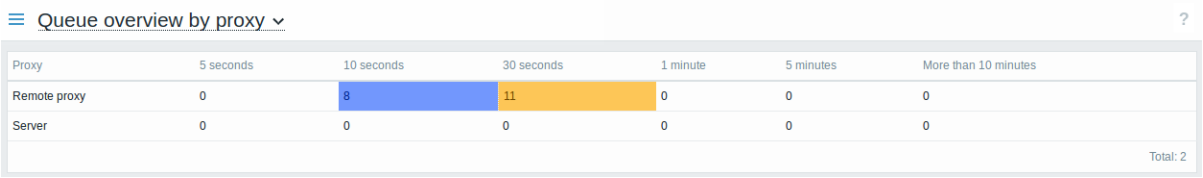


Items	5 seconds	10 seconds	30 seconds	1 minute	5 minutes	More than 10 minutes
Zabbix agent	1	11	1	0	0	0
Zabbix agent (active)	0	0	0	0	0	0
Simple check	0	0	0	0	0	0
SNMPv1 agent	0	0	0	0	0	0
SNMPv2 agent	0	0	0	0	0	0
SNMPv3 agent	0	0	0	0	0	0
Zabbix internal	0	0	0	0	0	0
Zabbix aggregate	0	0	0	0	0	0
External 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.



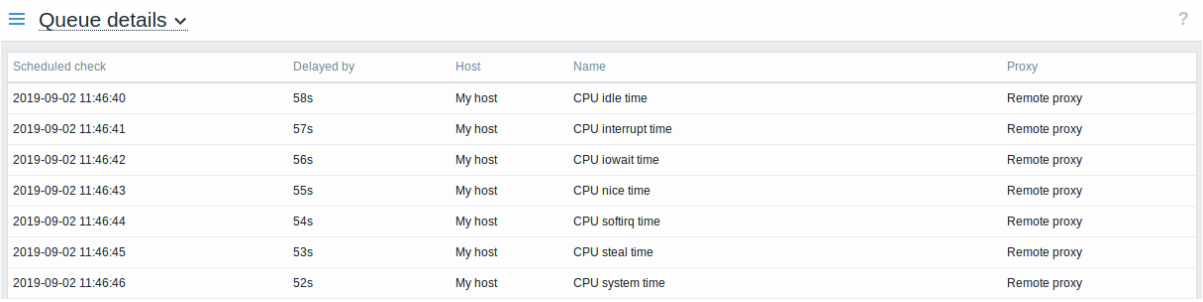
Proxy	5 seconds	10 seconds	30 seconds	1 minute	5 minutes	More than 10 minutes
Remote proxy	0	8	11	0	0	0
Server	0	0	0	0	0	0

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.



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

Var rasties situācija, kad dati netiek parādīti un tiek parādīts šāds kļūdas ziņojums:



Šajā gadījumā kļūdas ziņojums ir šāds:

```
Cannot display item queue. Permission denied
```

Tas notiek, ja PHP konfigurācijas parametri failā lietotāja saskarnes konfigurācijas fails - $ZBX_SERVER vai gan $ZBX_SERVER, gan $ZBX_SERVER_PORT - norāda uz esošu Zabbix serveri, kas izmanto citu datubāzi.

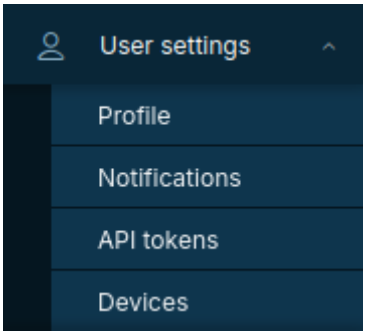
Lietotāja iestatījumi

Pārskats

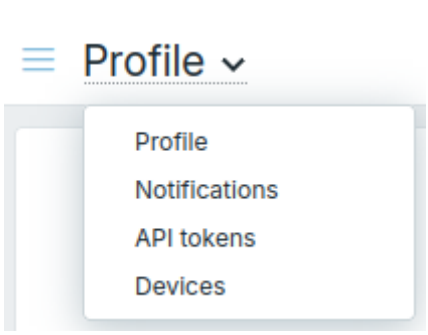
Atkarībā no jūsu lietotāja lomas, sadaļā *Lietotāja iestatījumi* var būt pieejamas šādas lapas:

- *Profils* vai *Lietotāja profils* - noteiktu Zabbix lietotāja saskarnes funkciju pielāgošanai;
- *Paziņojumi* - pašreizējā lietotāja paziņojumu pielāgošanai;
- *API marķieri* - pašreizējam lietotājam piešķirto API marķieru pārvaldībai;
- *Ierīces* - pašreizējam lietotājam piederošo mobilo ierīču pārvaldībai.

Pieejamo lapu saraksts tiek parādīts, noklikšķinot uz lietotāja ikonas Zabbix izvēlnes apakšdaļā (nav pieejams *guest* lietotājam). Starp lapām var arī pārslēgties, izmantojot virsraksta nolaižamo izvēlni augšējā kreisajā stūrī.



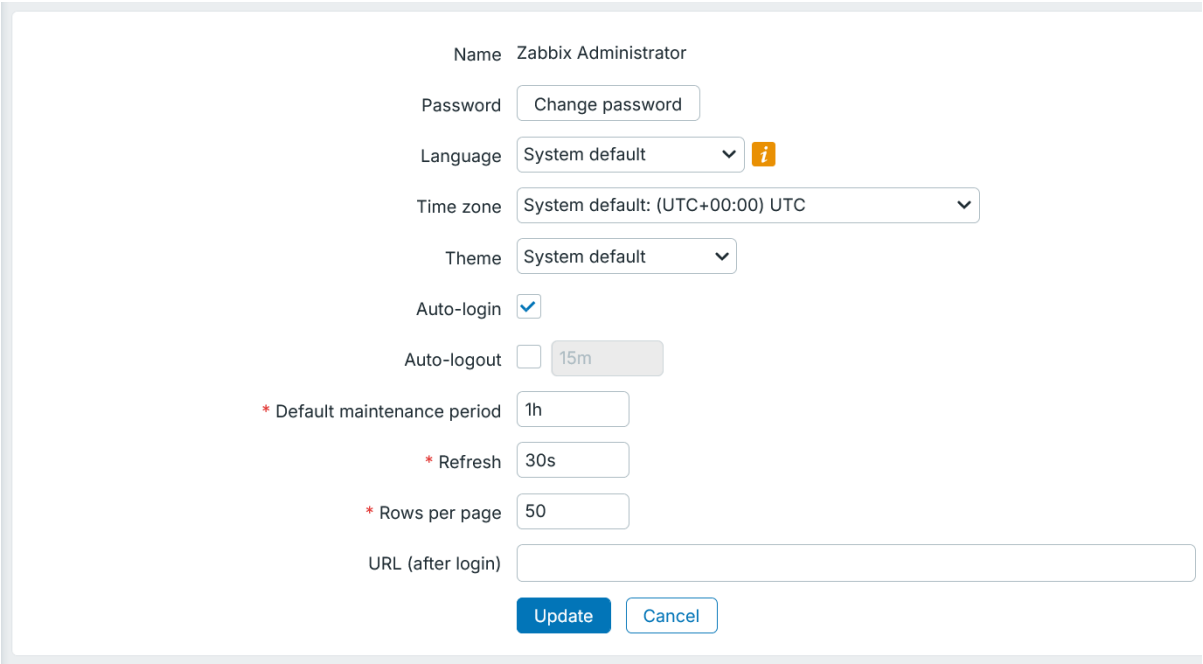
Trešā līmeņa izvēlne.



Virsraksta nolaižamā izvēlne.

Lietotāja profils

Sadaļā *Lietotāja profils* var iestatīt pielāgotu lietotāja saskarnes valodu, krāsu motīvu, sarakstos attēlojamo rindu skaitu u. c. opcijas. Šeit veiktās izmaiņas tiks piemērotas tikai pašreizējam lietotājam.



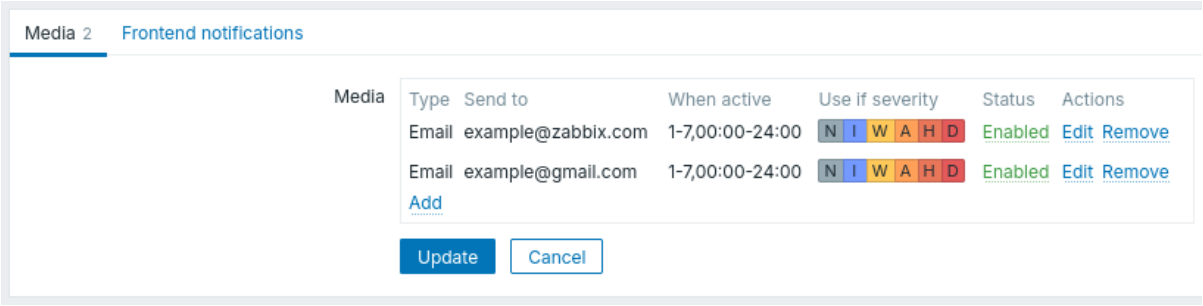
Parametrs	Apraksts
Name	Tiek attēlots lietotāja vārds un uzvārds. Ja vārds un uzvārds nav norādīti, tiek attēlots lietotājvārds.
Password	Noklikšķiniet uz pogas *Change password*, lai atvērtu trīs laukus: *Old password*, *New password*, *New password (once again)*. Pēc veiksmīgas paroles maiņas lietotājs tiks izrakstīts no visām aktīvajām sesijām. Ņemiet vērā, ka paroli var mainīt tikai lietotāji, kuri izmanto Zabbix iekšējo autentifikāciju.

Parametrs	Apraksts
Language	Atlasiet vēlamo lietotāja saskarnes valodu vai atlasiet **System default**, lai izmantotu noklusējuma sistēmas iestatījumus. Atlasot *English (en_US)*, lietotāja saskarnē tiks iespējots arī ASV laika un datuma formāts. Papildinformāciju skatiet sadaļā Papildu lietotāja saskarnes valodu instalēšana.
Time zone	Atlasiet laika joslu, lai lietotāja līmenī aizstātu globālo laika joslu, vai atlasiet **System default**, lai izmantotu globālos laika joslas iestatījumus.
Theme	Atlasiet krāsu motīvu savam profilam: **System default** - izmantot noklusējuma sistēmas iestatījumus; **Blue** - standarta zilais motīvs; **Blue (classic)** - standarta zilais motīvs ar mantotajiem fontiem; **Dark** - alternatīvs tumšais motīvs; **Dark blue** - tumši zilais motīvs; **Dark (classic)** - alternatīvs tumšais motīvs ar mantotajiem fontiem; **High-contrast light** - gaišs motīvs ar augstu kontrastu; **High-contrast dark** - tumšs motīvs ar augstu kontrastu.
Auto-login	Atzīmējiet šo izvēles rūtiņu, lai Zabbix jūs atcerētos un automātiski pieteiktu 30 dienas. Piesakoties ar iespējotu opciju *Remember for 30 days*: - Automātiskās izrakstīšanās taimeris tiek atiestatīts (sesija turpinās līdz 30 dienu perioda beigām). - Nākamajos apmeklējumos 30 dienu laikā tiek iespējota automātiskā pieteikšanās. Piesakoties bez opcijas *Remember for 30 days*: - Automātiskās pieteikšanās parametrs tiek notīrīts. - Automātiskā izrakstīšanās parasti notiek atbilstoši konfigurētajam taimautam. Šim nolūkam tiek izmantoti pārlūkprogrammas sīkfaili.
Auto-logout	Ja šī izvēles rūtiņa ir atzīmēta, jūs tiksiet automātiski izrakstīts pēc iestatītā sekunžu skaita (vismaz 90 sekundes, ne vairāk kā 1 diena). Tiek atbalstīti laika sufiksi, piemēram: 90s, 5m, 2h, 1d. Ņemiet vērā, ka šī opcija nedarbosies šādos gadījumos: * Kad *Monitoring* izvēlnes lapās fonā tiek atsvaidzināta informācija. Ja tiek atstātas atvērtas lapas, kurās dati tiek atsvaidzināti noteiktā laika intervālā (informācijas paneļi, grafiki, jaunākie dati u. c.), sesijas ilgums tiek attiecīgi pagarināts, tādējādi atspējojot automātiskās izrakstīšanās funkciju. * Ja piesakāties ar atzīmētu opciju *Remember me for 30 days*; šādā gadījumā automātiskās izrakstīšanās taimeris tiek ignorēts un sesija paliek aktīva visu periodu. * Ja autentifikācija tiek veikta, izmantojot SSO identitātes nodrošinātāju (IdP), automātiskā izrakstīšanās pārtrauc tikai Zabbix sesiju. SSO sesija ar IdP paliek aktīva, tāpēc lietotājs var atkārtoti pieteikties, neievadot paroli. *Auto-logout* var pieņemt arī vērtību "0", kas nozīmē, ka pēc profila iestatījumu atjaunināšanas automātiskās izrakstīšanās funkcija tiek atspējota.
Default maintenance period	Iestatiet noklusējuma uzturēšanas periodu ad hoc uzturēšanai.
Refresh	Iestatiet, cik bieži tiks atsvaidzināta informācija *Monitoring* izvēlnes lapās (vismaz 0 sekundes, ne vairāk kā 1 stunda). Tiek atbalstīti laika sufiksi, piemēram: 30s, 90s, 1m, 1h.
Rows per page	Iestatiet, cik rindu sarakstos tiks attēlots vienā lapā. Mazāks rindu skaits (un mazāks attēlojamo ierakstu skaits) nodrošina ātrāku ielādi.
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:

- aiz nosaukuma tiek parādīta dzeltena informācijas ikona;
- kolonnā *Statuss* tiek parādīts "Atspējots".

Note:
Lietotāja tiesības pašam mainīt savu mediju informāciju var tikt piešķirtas/atsauktas atbilstoši viņa lietotāja lomai (skatiet opciju *Izveidot un rediģēt savus medijus*). Super admin lietotāja tiesības mainīt citu lietotāju mediju informāciju arī var tikt piešķirtas/atsauktas atbilstoši viņa lietotāja lomai (skatiet opciju *Izveidot un rediģēt lietotāja medijus*)

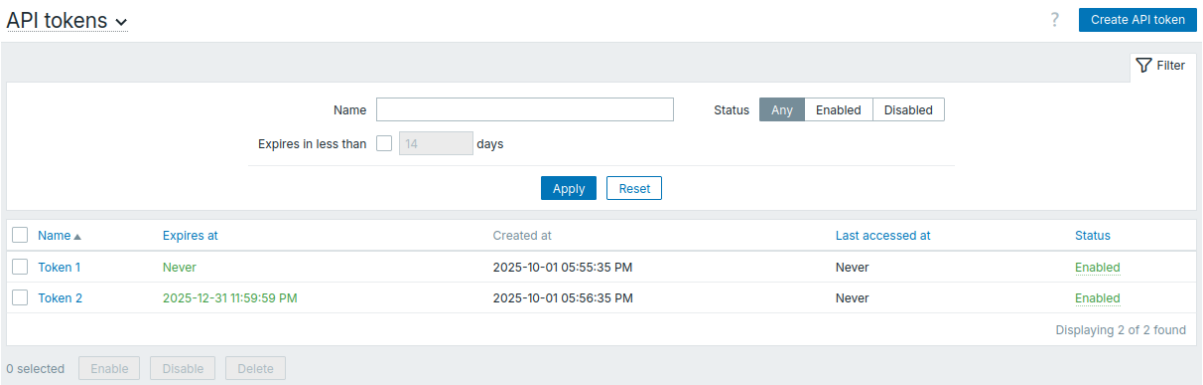
Ņemiet vērā, ka nodrošinātajiem lietotājiem:

- nodrošinātā lietotāja medijus nevar dzēst;
- nodrošinātā lietotāja medijus var atspējot/iespējot;
- tādus nodrošinātā lietotāja mediju laukus kā *Kad aktīvs*, *Lietot, ja smaguma pakāpe* 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 tokeni

Sadaļa *API tokeni* ļauj skatīt lietotājam piešķirtos tokenus, rediģēt tokena informāciju un izveidot jaunus tokenus. Šī sadaļa lietotājam ir pieejama tikai tad, ja lietotāja lomas iestatījumos ir atļauta darbība *Pārvaldīt API tokenus* lietotāja loma.



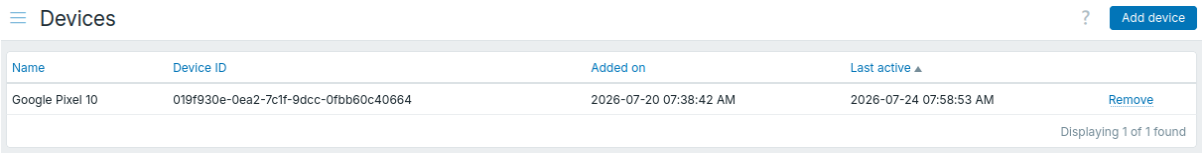
API tokenus var filtrēt pēc nosaukuma, derīguma termiņa vai statusa (*Iespējots/Atspējots*). Noklikšķiniet uz tokena statusa sarakstā, lai ātri iespējotu/atspējotu tokenu. Varat arī vienlaikus iespējot/atspējot vairākus tokenus, atlasot tos sarakstā un pēc tam noklikšķinot uz pogām *Iespējot/Atspējot* zem saraksta.

Attention:
Lietotāji nevar skatīt viņiem piešķirto tokenu *Auth token* vērtību Zabbix. *Auth token* vērtība tiek parādīta tikai vienu reizi - uzreiz pēc tokena izveides. Ja tokens ir pazaudēts, tas ir jāģenerē atkārtoti.

Ierīces

Sadaļā *Ierīces* varat pievienot vai noņemt savas mobilās ierīces, lai tās izmantotu kopā ar Zabbix mobilo lietotni.

Šī sadaļa tiek parādīta, ja ir iespējots `EnableMobileDevices`.



Lai pievienotu ierīci:

1. Noklikšķiniet uz *Pievienot ierīci*.
2. Noskenējiet QR kodu savā Zabbix mobilajā lietotnē.

Note:
Kad pievienojat ierīci, Zabbix jūsu lietotāja multividei pievieno arī push paziņojumu multivides tipu. Kad ierīci noņemat, šī lietotāja multivide tiek noņemta. Ja ir konfigurēti vairāki push paziņojumu multivides tipi, Zabbix pievieno to, kas ir iespējots un alfabētiskā secībā atrodas pirmais.

Pievienotās ierīces tiek parādītas sarakstā. Tajā tiek rādīts ierīču nosaukums, ID, kā arī datums un laiks, kad tās tika pievienotas un kad pēdējo reizi bija aktīvas.

Varat arī noņemt ierīces. Kad ierīci noņemat, tās piekļuves akreditācijas dati tiek atsaukti un noņemti no Zabbix, pēc tam ierīce vairs nevar piekļūt Zabbix.

Superadministratori var pārvaldīt arī citu lietotāju ierīces sadaļā *Lietotāji* > *Ierīces*.

Globālie paziņojumi

Pārskats

Globālie paziņojumi nodrošina iespēju attēlot aktuālās problēmas reāllaikā tieši jūsu pašreizējā ekrānā Zabbix lietotāja saskarnē.

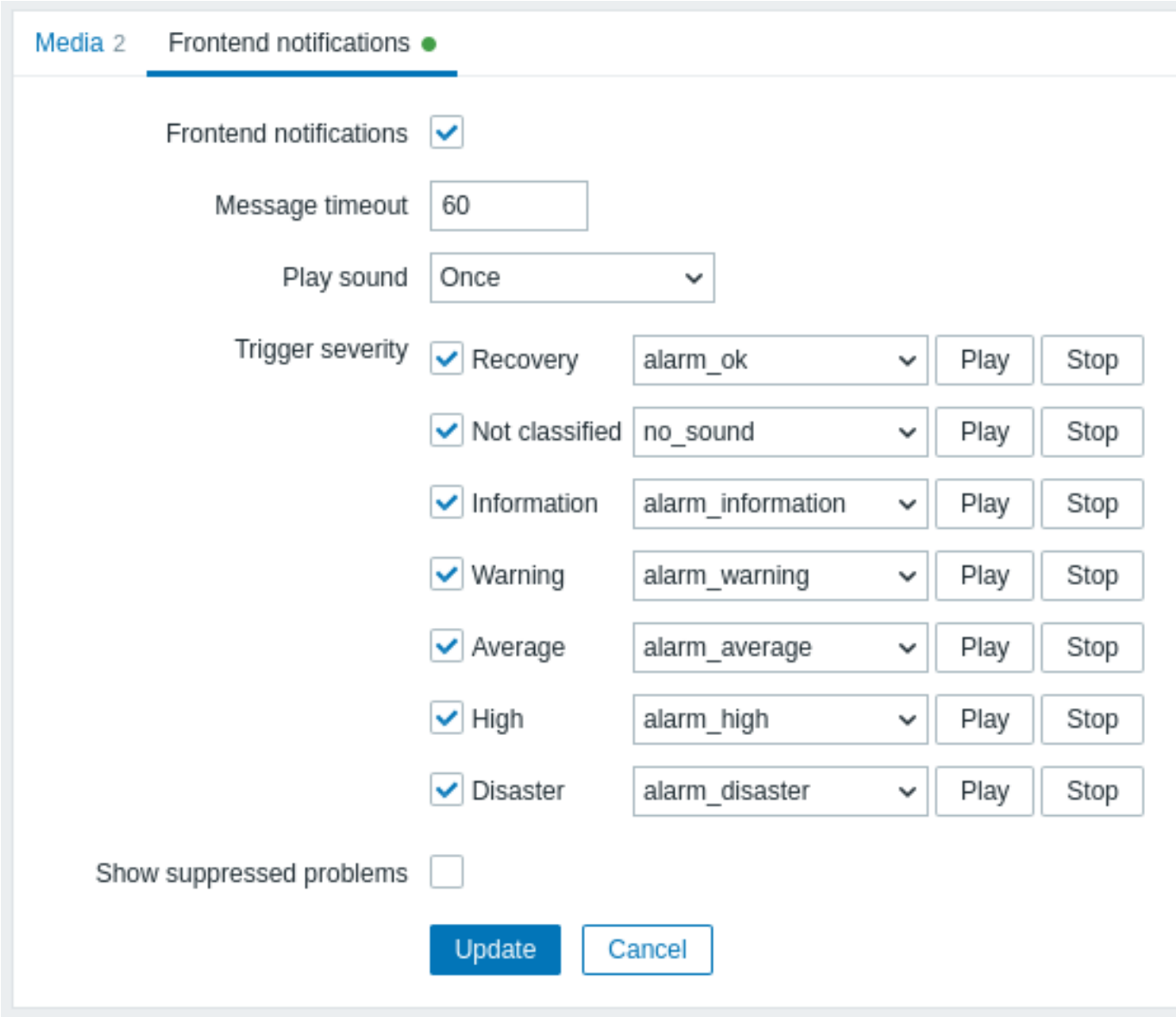
Bez globālajiem paziņojumiem, strādājot ārpus sadaļām *Problēmas* vai *Panelis*, jūs nesaņemtu nekādu informāciju par pašreizējām problēmām. Globālie paziņojumi nodrošina, ka šī informācija tiek attēlota neatkarīgi no jūsu pašreizējās atrašanās vietas Zabbix lietotāja saskarnē.

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

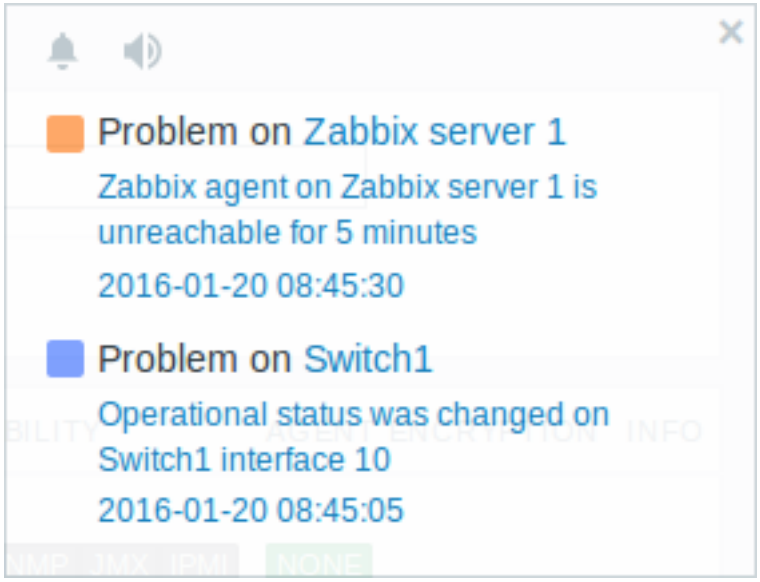
Globālos paziņojumus var iespējot katram lietotājam sadaļas Paziņojumi cilnē **Lietotāja saskarnes paziņojumi**.



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.

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.

Globālā meklēšana

Zabbix lietotāja saskarnē ir iespējams meklēt hosts, hostu grupas, veidnes un veidņu grupas.

Meklēšanas ievades lauks atrodas zem Zabbix logotipa izvēlnē. Meklēšanu var sākt, nospiežot *Enter* vai noklikšķinot uz meklēšanas ikonas.



Ja ir hosts, kura nosaukuma jebkurā daļā ir ietverta ievadītā virkne, tiks parādīta nolaižamā izvēlne ar visu šādu hostu sarakstu (atbilstošā daļa būs izcelta oranžā krāsā). Nolaižamajā izvēlnē hosts tiks parādīts arī tad, ja tā redzamais nosaukums atbilst kā meklēšanas virkne ievadītajam tehniskajam nosaukumam; šāds hosts tiks uzskaitīts, bet bez jebkādas izcelšanas.

Meklējamie atribūti

Hostus var meklēt pēc šādām īpašībām:

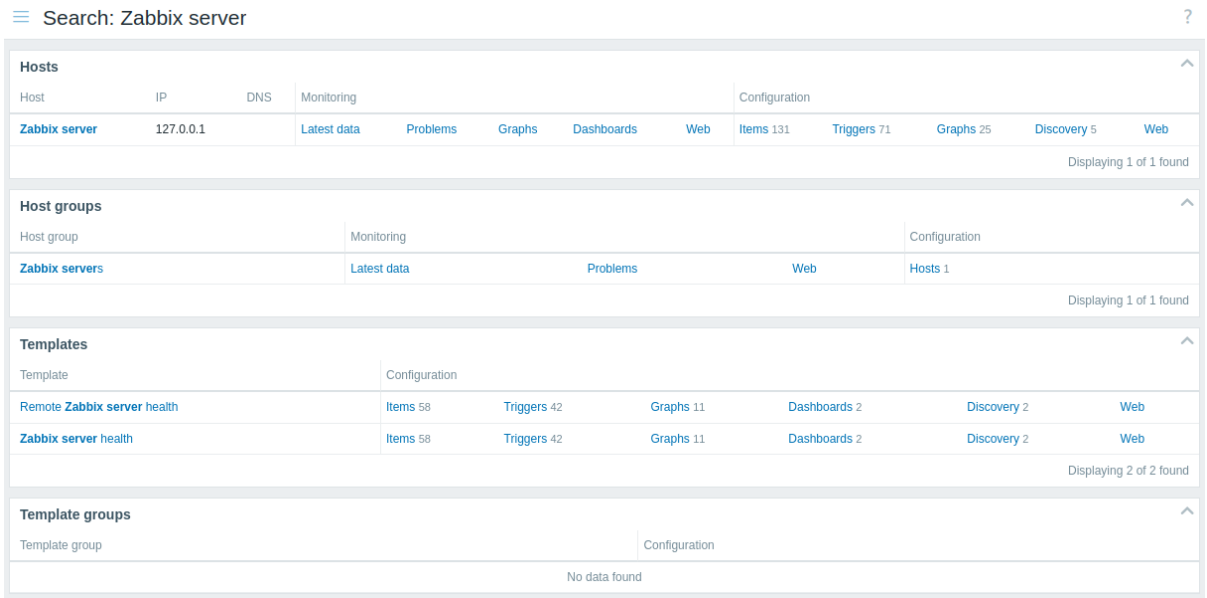
- Hosta nosaukums
- Redzamais nosaukums
- IP adrese
- DNS nosaukums

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 apakšdaļā, 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

Lietotāja saskarne maintenance mode

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.

Konfigurēšana

Lai iespējotu uzturēšanas režīmu, atveriet failu `maintenance.inc.php` (tas atrodas Zabbix HTML dokumentu direktorijas `/conf` mapē tīmekļa serverī) un noņemiet komentāra atzīmi šīm rindām:

```
// Uzturēšanas režīms.
define('ZBX_DENY_GUI_ACCESS', 1);

// To IP adrešu masīvs, kurām atļauts izveidot savienojumu ar lietotāja saskarni (neobligāti).
$ZBX_GUI_ACCESS_IP_RANGE = array('127.0.0.1');

// Brīdinājuma ekrānā parādītais ziņojums (neobligāti).
$ZBX_GUI_ACCESS_MESSAGE = 'Mēs atjauninām MySQL datubāzi līdz plkst. 15.00. Sekojiet jaunumiem...';
```

Note:
Vairumā gadījumu fails `maintenance.inc.php` atrodas tīmekļa servera Zabbix HTML dokumentu direktorijas mapē `/conf`. Tomēr dažas operētājsistēmas un tīmekļa serveri var izmantot citu atrašanās vietu.
Piemēram, atrašanās vieta:

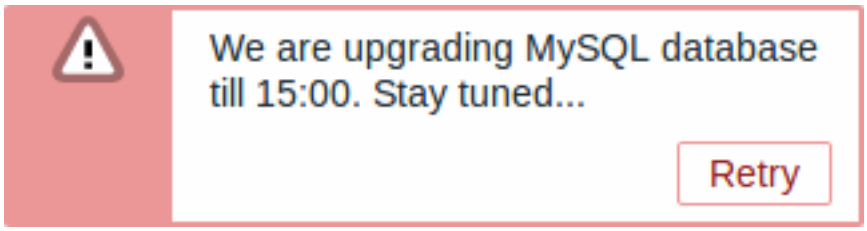
- SUSE un RedHat sistēmās ir `/etc/zabbix/web/maintenance.inc.php`.
- Uz Debian balstītās sistēmās ir `/usr/share/zabbix/conf/`.

Skatiet arī PHP failu kopēšana.

Parameter	Details
ZBX_DENY_GUI_ACCESS	Ja parametrs ir definēts ar jebkādu vērtību, tiks iespējots uzturēšanas režīms. Lai atspējotu uzturēšanas režīmu, komentējiet šo rindu vai izdzēsiet to.
ZBX_GUI_ACCESS_IP_RANGE	IP adrešu masīvs, kurām ir atļauts izveidot savienojumu ar lietotāja saskarni (neobligāts). Piemēram: `array('192.168.1.1', '192.168.1.2')`
ZBX_GUI_ACCESS_MESSAGE	Ziņojums, lai informētu lietotājus par uzturēšanas darbiem (neobligāts). 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.

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:

Parameter	Description	Example
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

Parameter	Description	Example
severities	Filtra opcija *Smagums*: atlasīto smagumu masīvs formātā `severities[*]=*` (aizstājiet * ar smaguma līmeni). Iespējamās vērtības: 0 - nav klasificēts; 1 - informācija; 2 - brīdinājums; 3 - vidējs; 4 - augsts; 5 - katastrofāls.	severities[3]=3
inventory	Filtra opcija *Host inventārs*: inventāra lauku masīvs `[field],[value]`	inventory[0][field]=type&invento
evaltype	Filtra opcija *Tagi*: taga novērtēšanas metode. Iespējamās vērtības: 0 - and/or; 2 - or.	evaltype=0
tags	Filtra opcija *Tagi*: definēto tagu 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 tagus*. Iespējamās vērtības: 0 - nav; 1 - viens; 2 - divi; 3 - trīs.	show_tags=3
tag_name_format	Filtra opcija *Taga nosaukums*. Iespējamās vērtības: 0 - pilns nosaukums; 1 - saīsināts nosaukums; 2 - nav.	tag_name_format=1
tag_priority	Filtra opcija *Tagu attēlošanas prioritāte*: ar komatiem atdalīta tagu attēlošanas prioritātes 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 *Apstiprinājuma statuss*. Iespējamās vērtības: 0 - visi; 1 - neapstiprinātie; 2 - apstiprinātie.	acknowledgement_status=0
acknowledged_by_me	Filtra opcija *Es*; atbalstīta tikai ar `acknowledgement_status=2`. Iespējamās vērtības: 0 - atspējots; 1 - iespējots.	acknowledged_by_me=1
compact_view	Filtra opcija *Kompakts skats*. Iespējamās vērtības: 0 - atspējots; 1 - iespējots.	compact_view=1

Parameter	Description	Example
highlight_row	Filtra opcija *Izcelt visu rindu*. Iespējamās vērtības: 0 - atspējots; 1 - iespējots.	highlight_row=1
filter_name	Filtra rekvizītu opcija *Nosaukums*: brīvas formas virkne	filter_name=Databases
filter_show_counter	Filtra rekvizītu opcija *Rādīt ierakstu skaitu*. Iespējamās vērtības: 0 - atspējots; 1 - iespējots.	filter_show_counter=1
filter_custom_time	Filtra rekvizītu 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 *Smagums*; name - kārtot pēc kolonnas *Problēma*.	sort=clock
sortorder	Rezultātu kārtošanas secība. Iespējamās vērtības: DESC - dilstoši; ASC - augoši.	sortorder=DESC
age_state	Filtra opcija *Vecums mazāks par*; atbalstīta tikai ar `show`=3. Iespējamās vērtības: 0 - atspējot parametru age; 1 - iespējot parametru age.	age_state=1
age	Filtra opcija *Vecums mazāks par*: vesels skaitlis, dienu skaits; atbalstīta tikai ar `age_state`=1 un `show`=3.	age=7
groupids	Filtra opcija *Host grupas*: host grupu ID masīvs.	groupids[]=4
hostids	Filtra opcija *Hosts*: host 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 (piem., now-1m); atbalstīts tikai ar `filter_custom_time`=1	from=now-2h
to	Datumu diapazona beigas, var būt relatīvas (piem., now-1m); atbalstīts tikai ar `filter_custom_time`=1	to=now

Skatiet arī: Problēmu lapas filtra opcijas.

Kioska režīms

Atbalstītajās lietotāja saskarnes lapās kioska režīmu var aktivizēt, izmantojot URL parametrus. Piemēram, informācijas paneļos:

- `/zabbix.php?action=dashboard.view&kiosk=1` - aktivizēt kioska režīmu
- `/zabbix.php?action=dashboard.view&kiosk=0` - aktivizēt parasto režīmu

Slaidrāde

Ir iespējams aktivizēt slaidrādi informācijas panelī:

- `/zabbix.php?action=dashboard.view&slideshow=on` - aktivizēt slaidrādi

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 noderīgi lietotājiem:

- ZBX_MIN_PERIOD

Minimālais grafika periods sekundēs. Pēc noklusējuma viena minūte.

- GRAPH_YAXIS_SIDE_DEFAULT

Y ass noklusējuma atrašanās vieta vienkāršajos grafikos un noklusējuma vērtība nolaižamajā sarakstā, pievienojot vienumus pielāgotajiem grafikiem. Iespējamās vērtības: 0 - pa kreisi, 1 - pa labi.

Noklusējums: 0

- ZBX_SESSION_NAME

Virkne, kas tiek izmantota kā Zabbix lietotāja saskarnes sesijas sīkfaila nosaukums.

Noklusējums: zbx_sessionid

- ZBX_DATA_CACHE_TTL

TTL noildze sekundēs, ko izmanto, lai padarītu nederīgu Vault atbildes datu kešatmiņu. Iestatiet 0, lai atspējotu Vault atbildes kešošanu.

Noklusējums: 60

- SUBFILTER_VALUES_PER_GROUP

Apakšfiltra vērtību skaits vienā grupā (piemēram, jaunākie dati apakšfiltrā).

Noklusējums: 1000

- ZBX_MAX_WIDGET_LINES

Maksimālais attēlojamo logrīka rindu skaits.

Noklusējums: 1000

Papildu lietotāja saskarnes valodas

Pārskats Lai Zabbix lietotāja saskarnē izmantotu jebkuru citu valodu, nevis angļu valodu, attiecīgajai lokalizācijai jābūt instalētai tīmekļa serverī. Turklāt, lai tulkojumi darbotos, ir nepieciešams PHP gettext paplašinājums.

Lokalizāciju instalēšana Lai uzskaitītu visas instalētās valodas, izpildiet:

```
locale -a
```

Ja dažas nepieciešamās valodas nav uzskaitītas, atveriet failu `/etc/locale.gen` un noņemiet komentāru no nepieciešamajām lokalizācijām. Tā kā Zabbix izmanto UTF-8 kodējumu, jums jāizvēlas lokalizācijas ar UTF-8 rakstzīmju kopu.

Tagad izpildiet:

```
locale-gen
```

Restartējiet tīmekļa serveri. Ņemiet vērā, ka, ja tīmekļa serveris izmanto PHP-FPM un tas ir instalēts, tas arī ir jārestartē kopā ar tīmekļa serveri.

Tagad lokalizācijām vajadzētu būt instalētām. Iespējams, lai jaunās valodas parādītos, pārlūkprogrammā būs jāpārlādē Zabbix lietotāja saskarne, izmantojot `Ctrl + F5`.

Zabbix instalēšana Ja Zabbix tiek instalēts tieši no Zabbix git repozitorija, tulkošanas faili ir jāģenerē manuāli. Lai ģenerētu tulkošanas failus, izpildiet:

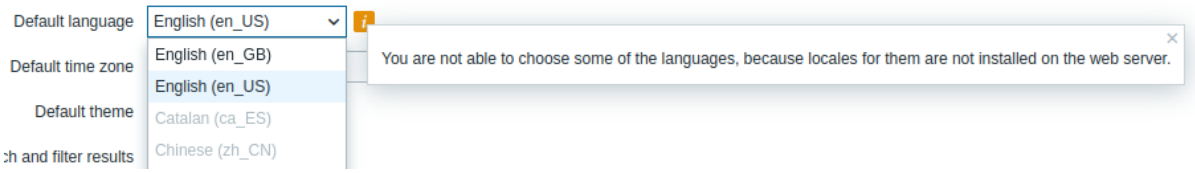
```
make gettext
locale/make_mo.sh
```

Šis solis nav nepieciešams, ja Zabbix tiek instalēts no pakotnēm vai avota `tar.gz` failiem.

Valodas izvēle Zabbix lietotāja saskarnē valodu var izvēlēties vairākos veidos:

- Instalējot lietotāja saskarni - lietotāja saskarnes instalēšanas vednī. Atlasītā valoda tiks iestatīta kā sistēmas noklusējuma valoda.
- Pēc instalēšanas sistēmas noklusējuma valodu var mainīt sadaļā *Administration* > *General* > *GUI*.
- Konkrēta lietotāja valodu var mainīt lietotāja profilā.

Ja valodai atbilstošā locale nav instalēta datorā, šī valoda Zabbix valodas atlasītājā būs pelēkota. Ja trūkst vismaz viena locale, blakus valodas atlasītājam tiek parādīta oranža ikona. Noklikšķinot uz šīs ikonas, tiks parādīts šāds ziņojums: "You are not able to choose some of the languages, because locales for them are not installed on the web server."



Izveidojot savu tēmu

Pārskats

Pēc noklusējuma Zabbix nodrošina vairākas iepriekš definētas lietotāja saskarnes tēmas. Lai izveidotu pielāgotu tēmu, izpildiet šeit sniegto soli pa solim procedūru.

1. solis: Izveidojiet savu motīvu

Lai izveidotu savu motīvu:

1. Izveidojiet CSS failu no nulles vai nokopējiet esošu motīvu (piemēram, `blue-theme.css`, `dark-theme.css` vai citu) no `assets/styles/` direktorijas un modificējiet to.
2. Saglabājiet savu motīvu (piemēram, `custom-theme.css`) `assets/styles/` direktorijā.

2. darbība: iespējojiet savu dizainu dizainu sarakstā

Lai jūsu pielāgotais dizains būtu pieejams lietotāja saskarnē, tas jāpievieno dizainu sarakstam, ko atgriež metode `APP::getThemes()`. To var izdarīt, APP klasē pārrakstot metodi `ZBase::getThemes()`:

1. Atveriet failu `include/classes/core/APP.php` un atrodiet klasi APP:

```
class APP extends ZBase {

}
```

2. Klasē pirms pēdējās noslēdzošās figūriekavas } pievienojiet šo metodi:

```
public static function getThemes() {
          return array_merge(parent::getThemes(), [
              'custom-theme' => _('Custom theme')
          ]);
}
```

kur:

- `custom-theme` - jūsu dizaina iekšējais identifikators; tam jāatbilst jūsu CSS faila nosaukumam bez paplašinājuma `.css`;
- `Custom theme` - parādāmais nosaukums, ko redzēsiet lietotāja saskarnē.

Lai pievienotu vairākus dizainus, uzskaitiet tos masīvā, atdalot katru ierakstu ar komatu. Komats pēc pēdējā ieraksta nav obligāts.

Piemēram:

```
    public static function getThemes() {
        return array_merge(parent::getThemes(), [
            'custom-theme' => _('Custom theme'),
            'anothertheme' => _('Another theme'),
            'onemoretheme' => _('One more theme')
        ]);
    }
```

3. darbība: mainiet grafiku krāsas

Lai mainītu grafiku krāsas, datubāzes tabulai `graph_theme` jāpievieno ieraksts jaunajam dizainam.

Ieraksta pievienošanas piemērs MySQL/MariaDB datubāzei:

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

```
mysql> USE zabbix;
mysql> INSERT INTO graph_theme (
    graphthemeid,
    theme,
    backgroundcolor,
    graphcolor,
    gridcolor,
    maingridcolor,
    gridbordercolor,
    textcolor,
    highlightcolor,
    leftpercentilecolor,
    rightpercentilecolor,
    nonworktimecolor,
    colorpalette
)
SELECT
    MAX(graphthemeid) + 1,
    'custom-theme',
    'FFFFFF',
    'FFFFFF',
    'CCD5D9',
    'ACBBC2',
    'ACBBC2',
    '1F2C33',
    'CC745E',
    '5ECCAB',
    'CC745E',
    'EBEBEB',
    '329F7E,C2583D,346D91,B26E44,CC6C91,7A6DC2,C4AA56,7C2731,BF75B8,73A350,B04833,633A60,879CCC,7FAD6C,324
FROM graph_theme;
```

4. darbība: Jaunās tēmas izmantošana

Zabbix lietotāja saskarnē šo tēmu var iestatīt kā noklusējuma tēmu vai atlasīt to lietotāja profilā.

Atkļūdošanas režīms

Pārskats

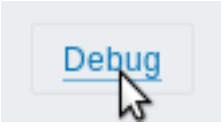
Atkļūdošanas režīmu var izmantot, lai diagnosticētu veiktspējas problēmas lietotāja saskarnes lapās.

Konfigurācija

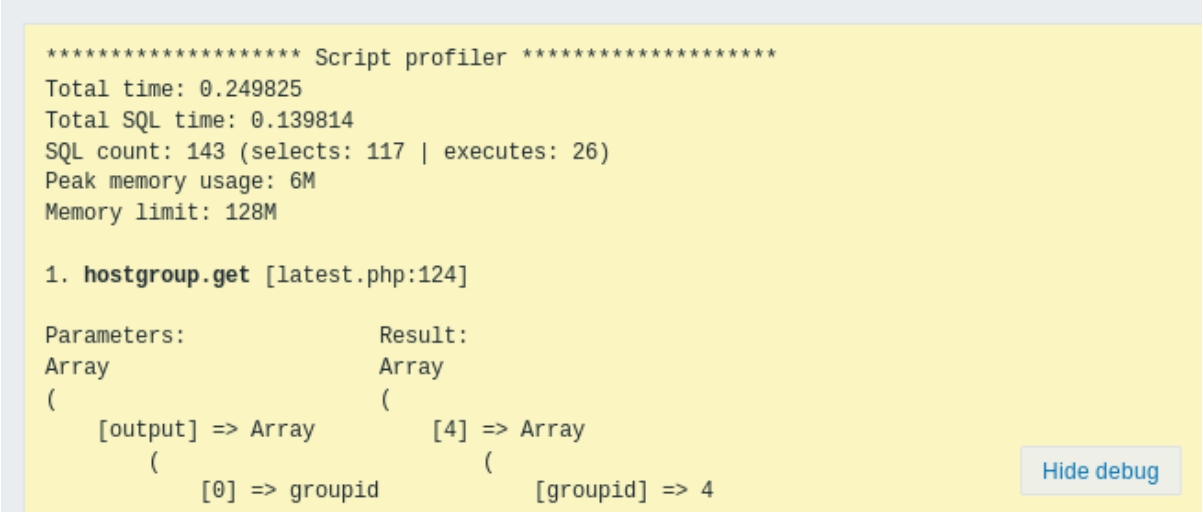
Atkļūdošanas režīmu var aktivizēt atsevišķiem lietotājiem, kuri pieder lietotāju grupai:

- konfigurējot lietotāju grupu;
- skatot konfigurētās lietotāju grupas.

Kad lietotāju grupai ir iespējots *Atkļūdošanas režīms*, tās lietotāji pārlūkprogrammas loga apakšējā labajā stūrī redzēs pogu *Debug*:



Noklikšķinot uz pogas *Debug*, zem lapas satura tiek atvērts jauns logs, kurā ir lapas SQL statistika, kā arī API izsaukumu un atsevišķu SQL priekšrakstu saraksts:



Ja lapā rodas veiktspējas problēmas, šo logu var izmantot, lai meklētu problēmas pamatcēloni.

Warning:
Iespējots *Atkļūdošanas režīms* negatīvi ietekmē lietotāja saskarnes veiktspēju.

Zabbix izmantotās sīkdatnes

Pārskats

Šajā lapā ir sniegts Zabbix izmantoto sīkdatņu saraksts.

Nosaukums	Apraksts	Vērtības	Beidzas/Max-Age	HttpOnly[1]	Secure[2]
ZBX_SESSION_NAME	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)	-	-

Nosaukums	Apraksts	Vērtības	Beidzas/Max-Age	HttpOnly[1]	Secure[2]
browserwarning_ignore	Vai ignorēt ziņojumu 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

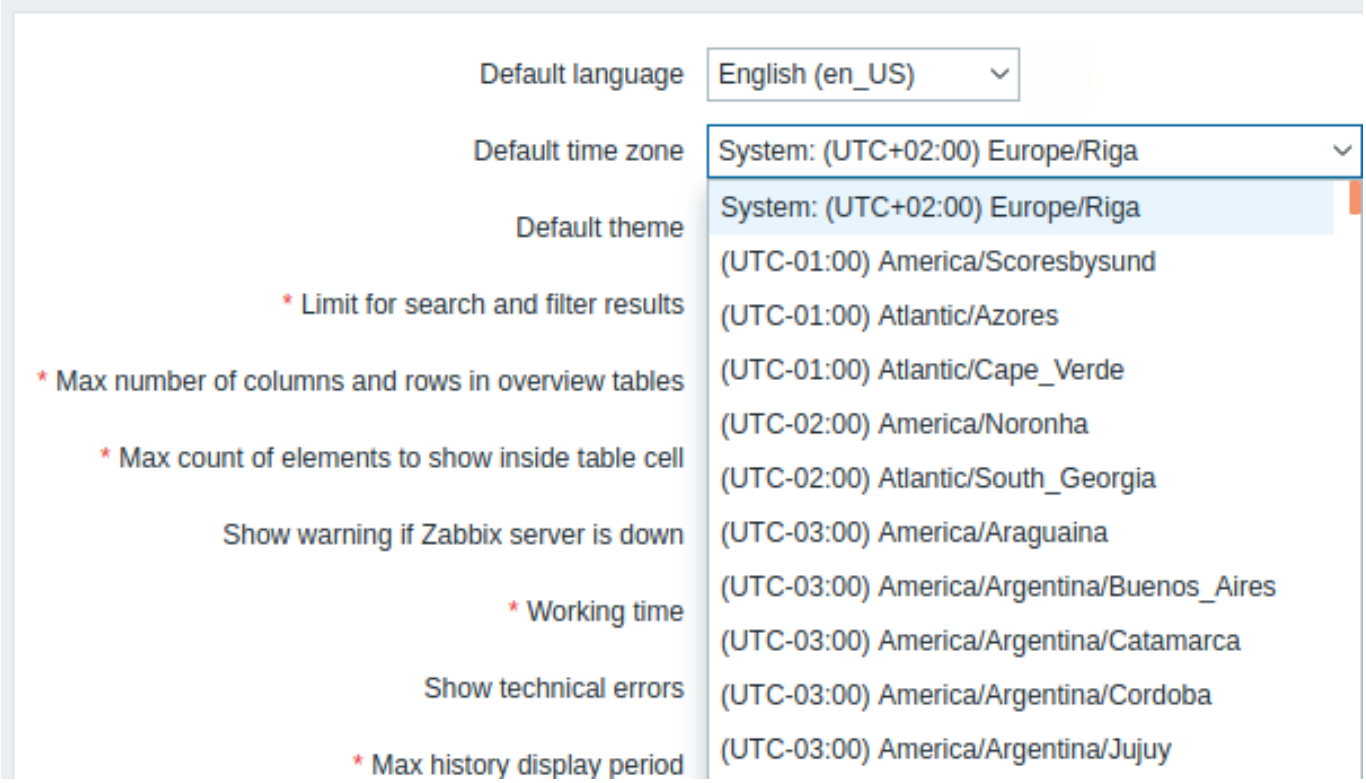
[1] Ja `HttpOnly` ir 'true', sīkdatne būs pieejama tikai caur HTTP protokolu. Tas nozīmē, ka sīkdatne nebūs pieejama skriptu valodām, piemēram, JavaScript. Šis iestatījums var efektīvi palīdzēt samazināt identitātes zādzību XSS uzbrukumu dēļ (lai gan to neatbalsta visas pārlūkprogrammas).

[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.

Laika joslas

Pārskats

Lietotāja saskarnes laika joslu var iestatīt globāli lietotāja saskarnē un pielāgot atsevišķiem lietotājiem.



Ja ir atlasīta opcija *System*, lietotāja saskarnei tiks izmantota tīmekļa servera laika josla (tostarp `date.timezone` vērtība failā `php.ini`, ja tā ir iestatīta), savukārt Zabbix serveris izmantos tās mašīnas laika joslu, kurā tas darbojas.

Note:
Zabbix serveris norādīto globālo vai lietotāja laika joslu izmantos tikai makrosu izvēršanai paziņojumos (piemēram, `{EVENT.TIME}` katram lietotājam var tikt izvērsts citā laika joslā) un laika ierobežojumam, kad tiek sūtīti paziņojumi (skatiet iestatījumu "When active" lietotāja multivides konfigurācijā).

Laika joslas izvēle neietekmē lietotāja saskarnes laika/datuma formātu. Tā vietā varat pielāgot saskarnes valodu (instalēšanas laikā vai sadaļā lietotāja iestatījumi) - atlasot *English (en_US)*, lietotāja saskarnē tiks iespējots arī ASV laika/datuma formāts.

Konfigurācija

Globālā laika josla:

- To var iestatīt manuāli, instalējot lietotāja saskarni.
- To var mainīt sadaļā *Administration* > *General* > *GUI*.

Lietotāja līmeņa laika josla:

- To var iestatīt, konfigurējot/atjauninot lietotāju.
- To var iestatīt katrs lietotājs savā lietotāja profilā.

Skatiet arī: Laika joslu saskaņošana, izmantojot plānošanas intervālus.

Paroles atiestatīšana

Pārskats Šajā sadaļā ir aprakstītas darbības, kas jāveic, lai Zabbix atiestatītu lietotāju paroles.

Darbības Ja esat aizmirsis savu Zabbix paroli un nevarat pieteikties, sazinieties ar Zabbix administratoru.

Super administratora lietotājs var mainīt visu lietotāju paroles lietotāja konfigurācijas veidlapā.

Ja Super administrators ir aizmirsis savu paroli un nevar pieteikties, jāizpilda tālāk norādītais SQL vaicājums, lai Super administratora lietotājam iestatītu 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*. Pirmajā pieteikšanās reizē noteikti nomainiet noklusējuma paroli.

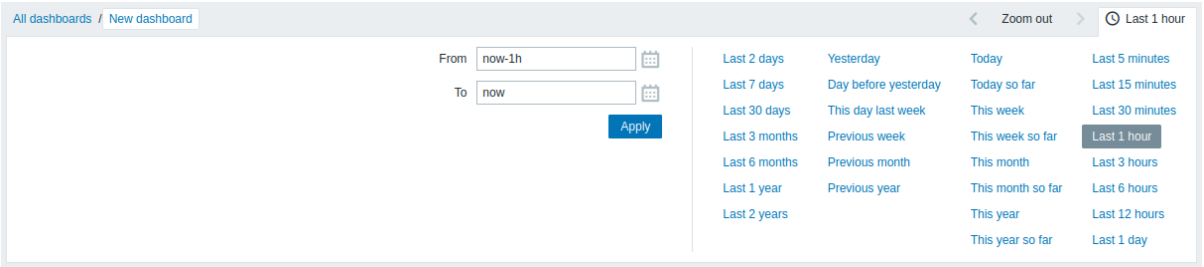
Laika periods un hostu atlasītāji

Pārskats

Šajā lapā aprakstīti informācijas paneļos pieejamie laika perioda un hostu atlasītāji. Šie atlasītāji ļauj dinamiski pielāgot attēlotos datus visos saderīgajos informācijas paneļa logrīkos, nemainot atsevišķu logrīku iestatījumus.

Laika perioda atlasītājs

Laika perioda atlasītājs ļauj ar vienu peles klikšķi atlasīt bieži nepieciešamos periodus. To var izvērst vai sakļaut, informācijas paneļa augšējā labajā stūrī noklikšķinot uz laika perioda atlasītāja pogas.



Opcijas, piemēram, *Šodien*, *Šī nedēļa* u. c., parāda visu periodu, tostarp nākotnē esošās stundas/dienas. Opcijas, piemēram, *Šodien līdz šim*, *Šī nedēļa līdz šim* u. c., parāda tikai pagājušās stundas.

Kad periods ir atlasīts, to var pārvietot uz priekšu vai atpakaļ laikā, noklikšķinot uz bultiņu pogām. Poga *Tālināt* ļauj samazināt periodu par 50% katrā virzienā.

Note:
Grafikiem (izņemot tos, kuriem ir *Pielāgots* laika periods) parādīto laika periodu var atlasīt arī, ar peles kreiso pogu iezīmējot apgabalu grafikā. Pēc peles kreisās pogas atlaišanas grafiks tiks pietuvināts iezīmētajam apgabalam. Tālināšana ir iespējama arī, veicot dubultklikšķi grafikā.

Lauki *No/Līdz* parāda atlasīto periodu 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`.

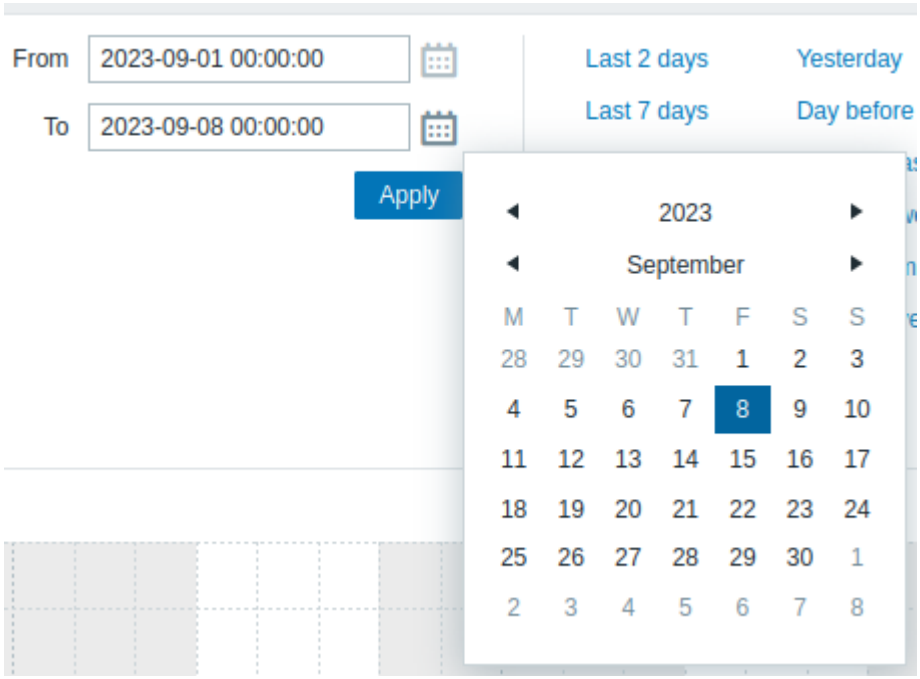
Tiek atbalstīti šādi relatīvā laika saīsinājumi:

- `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 var 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

No	Līdz	Atlasītais periods
`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, kura dati tiks parādīti saderīgos informācijas paneļa logrīkos.

Tas kļūst pieejams pēc informācijas paneļa rediģēšanas un saglabāšanas, ja vismaz vienam logrīkam ir konfigurēts parametrs *Ignorēt hostu*, iestatot informācijas paneli kā datu avotu.



Konfigurācijas parametri: Zabbix lietotāja saskarne

Pārskats Konfigurācijas parametri ļauj pielāgot Zabbix lietotāja saskarnes darbību.

Norādiet parametru vērtības lietotāja saskarnes konfigurācijas failā (`zabbix.conf.php`).

Šī lapa ir strukturēta šādi:

- Parametru pārskats (lai skatītu pilnu informāciju par parametru, noklikšķiniet uz tā nosaukuma)
- Parametru detalizēts apraksts

Parametrs	Apraksts
$DB['TYPE']	Datu bāzes tips.
$DB['SERVER']	Datu bāzes hosts nosaukums vai IP adrese.
$DB['PORT']	Datu bāzes ports.
$DB['DATABASE']	Datu bāzes nosaukums, ko Zabbix lietotāja saskarne izmanto, lai izveidotu savienojumu ar konfigurēto datu bāzi.
$DB['USER']	Lietotājvārds, ko Zabbix lietotāja saskarne izmanto, lai izveidotu savienojumu ar datu bāzi.
$DB['PASSWORD']	Parole, ko Zabbix lietotāja saskarne izmanto, lai izveidotu savienojumu ar datu bāzi.
$DB['SCHEMA']	PostgreSQL izmantotās datu bāzes shēmas nosaukums.
$DB['ENCRYPTION']	Iespējot vai atspējot TLS šifrēšanu savienojumiem starp Zabbix lietotāja saskarni un datu bāzi.
$DB['KEY_FILE']	Klienta privātās atslēgas faila ceļš, kas tiek izmantots datu bāzes TLS savienojumam.
$DB['CERT_FILE']	Klienta sertifikāta faila ceļš, kas tiek izmantots datu bāzes TLS savienojumam.
$DB['CA_FILE']	Certification Authority (CA) sertifikāta faila ceļš, kas tiek izmantots datu bāzes TLS savienojumam.
$DB['VERIFY_HOST']	Iespējot vai atspējot datu bāzes servera resursdatora nosaukuma pārbaudi TLS sertifikātā.
$DB['CIPHER_LIST']	Atļauto TLS šifru saraksts datu bāzes savienojumam.
$DB['VAULT']	Norādiet Vault noslēpumu pārvaldnieku, kas tiek izmantots datu bāzes akreditācijas datu glabāšanai.
$DB['VAULT_URL']	Vault noslēpumu pārvaldnieka URL, kas tiek izmantots datu bāzes akreditācijas datu izgūšanai.
$DB['VAULT_PREFIX']	Vault ceļa vai vaicājuma prefikss, kas tiek izmantots datu bāzes akreditācijas datu izgūšanai.
$DB['VAULT_DB_PATH']	Datu bāzes akreditācijas datu ceļš Vault noslēpumu pārvaldniekā.
$DB['VAULT_TOKEN']	Autentifikācijas pilnvara, kas tiek izmantota, lai piekļūtu Vault noslēpumu pārvaldniekam.
$DB['VAULT_APP_ROLE_ID']	AppRole lomas ID, kas tiek izmantots autentifikācijai HashiCorp Vault.
$DB['VAULT_APP_SECRET_ID']	AppRole slepenais ID, kas tiek izmantots autentifikācijai HashiCorp Vault.
$DB['VAULT_CERT_FILE']	Klienta sertifikāta faila ceļš, kas tiek izmantots autentifikācijai Vault noslēpumu pārvaldniekā.
$DB['VAULT_KEY_FILE']	Klienta privātās atslēgas faila ceļš, kas tiek izmantots autentifikācijai Vault noslēpumu pārvaldniekā.
$DB['VAULT_CACHE']	Iespējo vai atspējo no Vault noslēpumu pārvaldnieka izgūto akreditācijas datu kešošanu.
$ZBX_SERVER	Norādiet vērtību, lai aizstātu Zabbix resursdatora nosaukumu/IP adresi.

Parametrs	Apraksts
$ZBX_SERVER_PORT	Norādiet vērtību, lai aizstātu Zabbix servera portu.
$ZBX_SERVER_NAME	Zabbix servera nosaukums.
$IMAGE_FORMAT_DEFAULT	Noklusējuma attēla formāts.
$HISTORY_PROVIDERS[]	Vēstures datu glabāšanas nodrošinātāju konfigurācija Elasticsearch vai ClickHouse.
$SSO['SETTINGS']	SAML pierakstīšanās konfigurācijas iestatījumi.
$SSO['CERT_STORAGE']	Norādiet, kur tiek glabāti SAML sertifikāti.
$SSO['SP_KEY']	SAML pakalpojumu sniedzēja (SP) privātās atslēgas faila ceļš, ja privātā atslēga tiek glabāta failu sistēmā.
$SSO['SP_CERT']	SAML pakalpojumu sniedzēja (SP) sertifikāta faila ceļš, ja sertifikāts tiek glabāts failu sistēmā.
$SSO['IDP_CERT']	SAML identitātes nodrošinātāja sertifikāta faila ceļš, ja sertifikāts tiek glabāts failu sistēmā.
$ZBX_FEATURE_FLAGS['banner_ads']	Iespējot vai atspējot reklāmkarogu atbalstu.
$ZBX_FEATURE_FLAGS['http_auth']	Iespējot vai atspējot lietotāju HTTP autentifikāciju.
$ZBX_FEATURE_FLAGS['modules']	Iespējot vai atspējot pielāgotu moduļu atbalstu.
$ZBX_FEATURE_FLAGS['media_config']	Atspējot vai iespējot iepriekš konfigurētu multivides tipu rediģēšanu.
$ZBX_SERVER_TLS['ACTIVE']	Norāda, vai savienojumiem ar Zabbix serveri ir iespējots TLS.
$ZBX_SERVER_TLS['CA_FILE']	Certification Authority (CA) sertifikāta faila ceļš, kas tiek izmantots servera sertifikāta pārbaudei.
$ZBX_SERVER_TLS['KEY_FILE']	Klienta privātās atslēgas faila ceļš, kas atbilst klienta sertifikātam.
$ZBX_SERVER_TLS['CERT_FILE']	Klienta sertifikāta faila ceļš, ja ir nepieciešama savstarpēja TLS autentifikācija.
$ZBX_SERVER_TLS['CERTIFICATE_ISSUER']	Izdevēja atšķirīgais nosaukums (DN), kas jāsalīdzina ar servera sertifikātu.
$ZBX_SERVER_TLS['CERTIFICATE_SUBJECT']	Subjekta atšķirīgais nosaukums (DN), kas jāsalīdzina ar servera sertifikātu.
$TELEMETRY_PROVIDERS[]	Telemetrijas datu avota konfigurācija.

Parametru informācija

$DB['TYPE']

Datu bāzes tips.

Vērtības: `MYSQL` - MySQL, `POSTGRESQL` - PostgreSQL.

$DB['SERVER']

Datu bāzes hosta nosaukums vai IP adrese.

Ja `$DB['SERVER']` nav definēts, tiek izmantota noklusējuma Unix domēna ligzda. MySQL gadījumā, iestatot `$DB['SERVER']` uz `localhost`, savienojums tiek izveidots, izmantojot noklusējuma Unix ligzdu.
 PostgreSQL gadījumā `$DB['SERVER']` var saturēt vairākus ar komatiem atdalītus `host:port` ierakstus. Ja resursdatoram ar komatiem atdalītajā sarakstā ports nav definēts, kā noklusējuma ports tiek izmantota vērtība, kas definēta `$DB['PORT']`. Ja `$DB['SERVER']` ir norādīts ligzdas ceļš (piemēram, `/var/run/pgbouncer`), tiek izmantota šī Unix domēna ligzda.

Piemēri:

```
$DB['SERVER'] = 'localhost';
$DB['SERVER'] = 127.0.0.1;
```

$DB['PORT']

Datu bāzes ports.

Ja nav definēts, tiek izmantota vērtība 0.

$DB['DATABASE']

Zabbix lietotāja saskarnes izmantotais datubāzes nosaukums, lai izveidotu savienojumu ar konfigurēto datubāzi.

Piemērs:

```
$DB['DATABASE'] = 'zabbix';
```

$DB['USER']

Lietotājvārds, ko Zabbix lietotāja saskarne izmanto savienojumam ar datubāzi.

$DB['PASSWORD']

Parole, ko Zabbix lietotāja saskarne izmanto savienojuma izveidei ar datubāzi.

$DB['SCHEMA']

PostgreSQL izmantotās datubāzes shēmas nosaukums.

$DB['ENCRYPTION']

Iespējot vai atspējot šifrēšanu savienojumiem starp Zabbix lietotāja saskarni un datubāzi.

Vērtības: `true` - iespējota, `false` - atspējota.

$DB['KEY_FILE']

Klienta privātās atslēgas faila ceļš, kas tiek izmantots datubāzes TLS savienojumam.

Piemērs:

```
// MySQL:
$DB['KEY_FILE'] = '/etc/ssl/mysql/client-key.pem';
```

$DB['CERT_FILE']

Klienta sertifikāta faila ceļš, kas tiek izmantots datubāzes TLS savienojumam.

Piemērs:

```
// MySQL:
$DB['CERT_FILE'] = '/etc/ssl/mysql/client-cert.pem';
```

$DB['CA_FILE']

CA sertifikāta faila ceļš, kas tiek izmantots datubāzes TLS savienojumam.

Piemēri:

```
// MySQL:
$DB['CA_FILE'] = '/etc/ssl/mysql/ca.pem';
// PostgreSQL:
$DB['CA_FILE'] = '/etc/ssl/pgsql/root.crt';
```

$DB['VERIFY_HOST']

Iespējot vai atspējot datubāzes servera resursdatora nosaukuma pārbaudi TLS sertifikātā.

Vērtības: `true` - iespējots, `false` - atspējots.

$DB['CIPHER_LIST']

Atļauto TLS šifru saraksts datubāzes savienojumam.

Piemēri:

```
// MySQL:
$DB['CIPHER_LIST'] = 'TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256:TLS_AES_1
```

$DB['VAULT']

Norādiet Vault noslēpumu pārvaldnieku, kas tiek izmantots, lai glabātu datubāzes akreditācijas datus.

Vērtības: `CyberArk` vai `HashiCorp`.

Piemēri:

```
// CyberArk:
$DB['VAULT'] = 'CyberArk';
// HashiCorp:
$DB['VAULT'] = 'HashiCorp';
```

$DB['VAULT_URL']

Vault noslēpumu pārvaldnieka URL, kas tiek izmantots datubāzes akreditācijas datu izgūšanai.

Piemēri:

```
// CyberArk:
$DB['VAULT_URL'] = 'https://127.0.0.1:1858';
// HashiCorp:
$DB['VAULT_URL'] = 'https://localhost:8200';
```

$DB['VAULT_PREFIX']

Prefikss Vault ceļam vai vaicājumam, kas tiek izmantots datubāzes akreditācijas datu izgūšanai.

Piemērs:

```
// HashiCorp:
$DB['VAULT_PREFIX'] = '/v1/secret/data/zabbix/';
```

$DB['VAULT_DB_PATH']

Ceļš uz datubāzes akreditācijas datiem Vault noslēpumu pārvaldniekā.

Piemēri:

```
// CyberArk:
$DB['VAULT_DB_PATH'] = 'AppID=foo&Query=Safe=bar;Object=buzz';
// HashiCorp:
$DB['VAULT_DB_PATH'] = 'database';
```

$DB['VAULT_TOKEN']

Autentifikācijas pilnvara, ko izmanto, lai piekļūtu Vault noslēpumu pārvaldniekam.

Piemērs:

```
// For HashiCorp:
$DB['VAULT_TOKEN'] = 'hvs.CAESIIG_PILmULFYOsEyWHxkZ2mF2a8VPKNLE8eHqd4autYGGh4KHGh2cy5aeTY0NFNSaUp3ZnpWbDF1
```

$DB['VAULT_APP_ROLE_ID']

VaultAppSecretID lomas ID, ko izmanto autentifikācijai HashiCorp Vault.

Piemērs:

```
$DB['VAULT_APP_ROLE_ID'] = '6f1567f1-538d-7d70-e8e4-5d3e59471b2e';
```

$DB['VAULT_APP_SECRET_ID']

VaultAppSecretID, ko izmanto autentifikācijai HashiCorp Vault.

Piemērs:

```
$DB['VAULT_APP_SECRET_ID'] = '89dac8b1-f7d6-96a7-2ecf-f94d2de3d6f1';
```

$DB['VAULT_CERT_FILE']

Klienta sertifikāta faila ceļš, kas tiek izmantots autentifikācijai Vault noslēpumu pārvaldniekā.

Piemērs:

```
// CyberArk:
$DB['VAULT_CERT_FILE'] = 'conf/certs/cyberark-cert.pem';
```

$DB['VAULT_KEY_FILE']

Klienta privātās atslēgas faila ceļš, kas tiek izmantots autentifikācijai Vault noslēpumu pārvaldniekā.

Piemērs:

```
// CyberArk:
$DB['VAULT_KEY_FILE'] = 'conf/certs/cyberark-key.pem';
```

$DB['VAULT_CACHE']

Iespējo vai atspējo no Vault noslēpumu pārvaldnieka izgūto akreditācijas datu kešatmiņu.

Vērtības: `true` - iespējots, `false` - atspējots.

$ZBX_SERVER

Norādiet vērtību, lai aizstātu Zabbix hostname/IP adresi.
 Ja $ZBX_SERVER ir norādīts bez $ZBX_SERVER_PORT, tiek izmantots noklusējuma ports (10051).

$ZBX_SERVER_PORT

Norādiet vērtību, lai aizstātu Zabbix servera portu.
 Šis parametrs tiek izmantots kopā ar $ZBX_SERVER.

$ZBX_SERVER_NAME

Zabbix servera nosaukums.

$IMAGE_FORMAT_DEFAULT

Noklusējuma attēla formāts.

Noklusējums: IMAGE_FORMAT_PNG
 Vērtības: - IMAGE_FORMAT_PNG - PNG formāts - IMAGE_FORMAT_JPEG - JPEG formāts - IMAGE_FORMAT_GIF - GIF formāts

$HISTORY_PROVIDERS[]

Vēstures datu glabāšanas nodrošinātāju konfigurācija Elasticsearch vai ClickHouse.

Atbalstītie konfigurācijas parametri: - types - datu tipu masīvs, kas jāglabā ārējā krātuvē; - provider - vēstures datu nodrošinātāja tips: elasticsearch vai clickhouse; - url - vēstures datu nodrošinātāja URL; - db - datubāzes nosaukums (tiek izmantots ClickHouse); - username - datubāzes lietotājs (tiek izmantots ClickHouse); - password - datubāzes parole (tiek izmantota ClickHouse).

Piemēri:

```
// ClickHouse:
$HISTORY_PROVIDERS[] = [
    'types' => ['uint', 'dbl', 'str', 'log', 'text', 'json'],
    'provider' => 'clickhouse',
    'url' => 'http://localhost:8123',
    'db' => 'zabbix',
    'username' => 'zabbix',
    'password' => 'zabbix'
];
// Elasticsearch:
$HISTORY_PROVIDERS[] = [
    'types' => ['uint', 'dbl', 'str', 'log', 'text', 'json'],
    'provider' => 'elasticsearch',
    'url' => 'http://localhost:9200'
];
```

$SSO['SETTINGS']

SAML pierakstīšanās konfigurācijas iestatījumi. Masīvam $SSO['SETTINGS'] ir jāatbilst *SAML PHP Toolkit* bibliotēkas struktūrai (iekļauta kopā ar Zabbix).

Masīvā $SSO['SETTINGS'] var norādīt tikai šādas opcijas:

- 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 tiek iegūtas no datubāzes, un tās nevar ignorēt. Opcija debug tiks ignorēta.

Ja Zabbix lietotāja saskarne atrodas aiz starpniekservera vai slodzes balansētāja, var izmantot pielāgoto opciju use_proxy_headers:

- `false` (noklusējums) - ignorēt HTTP galvenes `X-Forwarded-*`;
- `true` - izmantot HTTP galvenes `X-Forwarded-*`, lai izveidotu bāzes URL.

Pilnu pieejamo konfigurācijas opciju aprakstu skatiet oficiālajā bibliotēkas dokumentācijā.

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',
        // ...
    ],
    // ...
];
```

$SSO['CERT_STORAGE']

Norādiet, kur tiek glabāti SAML sertifikāti.

Noklusējums: `file`
 Vērtības: - `database` - privātā atslēga un sertifikāti tiek glabāti datubāzē; - `file` - privātā atslēga un sertifikāti tiek glabāti failu sistēmā.

Piemērs:

```
$SSO['CERT_STORAGE'] = 'database';
```

$SSO['SP_KEY']

Ceļš uz SAML pakalpojumu sniedzēja (SP) privātās atslēgas failu, ja privātā atslēga tiek glabāta failu sistēmā.

Piemērs:

```
$SSO['SP_KEY'] = 'conf/certs/sp.key';
```

$SSO['SP_CERT']

Ceļš uz SAML pakalpojumu sniedzēja (SP) sertifikāta failu, ja sertifikāts tiek glabāts failu sistēmā.

Piemērs:

```
$SSO['SP_CERT'] = 'conf/certs/sp.crt';
```

$SSO['IDP_CERT']

Ceļš uz SAML identitātes nodrošinātāja sertifikāta failu, ja sertifikāts tiek glabāts failu sistēmā.

Piemērs:

```
$SSO['IDP_CERT'] = 'conf/certs/idp.crt';
```

$ZBX_FEATURE_FLAGS['banners_enabled']

Iespējot vai atspējot reklāmkarogu atbalstu.

Noklusējums: `true`
 Vērtības: `true` - iespējots, `false` - atspējots.

$ZBX_FEATURE_FLAGS['http_auth_enabled']

Iespējot vai atspējot lietotāju HTTP autentifikāciju.

Noklusējums: `true`
 Vērtības: `true` - iespējots, `false` - atspējots.

$ZBX_FEATURE_FLAGS['modules_config_enabled']

Iespējot vai atspējot piekļuvi moduļiem.

Noklusējums: `true`
 Vērtības: `true` - iespējots, `false` - atspējots.

$ZBX_FEATURE_FLAGS['media_type_denylist']

Atspējot konkrētu iepriekš konfigurētu multivides tipu rediģēšanu.

Vērtības: `email`, `script`, `sms`, `webhook`, `push`.

Piemērs:

```
$ZBX_FEATURE_FLAGS['media_type_denylist'] = ['webhook','push'];
```

$ZBX_SERVER_TLS['ACTIVE']

Norāda, vai TLS savienojums starp Zabbix lietotāja saskarni un Zabbix serveri ir iespējots.

Noklusējuma vērtība: `false`
 Vērtības: `true` - iespējots, `false` - atspējots.

$ZBX_SERVER_TLS['CA_FILE']

Ceļš uz sertificēšanas iestādes (CA) sertifikāta failu, kas tiek izmantots servera sertifikāta pārbaudei.

Piemērs:

```
$ZBX_SERVER_TLS['CA_FILE'] = '/etc/zabbix/ssl/ca.crt';
```

$ZBX_SERVER_TLS['KEY_FILE']

Klienta privātās atslēgas faila ceļš, kas atbilst klienta sertifikātam, kuru izmanto savstarpējai TLS autentifikācijai.

Piemērs:

```
$ZBX_SERVER_TLS['KEY_FILE'] = '/etc/zabbix/ssl/frontend.key';
```

$ZBX_SERVER_TLS['CERT_FILE']

Klienta sertifikāta faila ceļš, ja ir nepieciešama savstarpēja TLS autentifikācija.

Piemērs:

```
$ZBX_SERVER_TLS['CERT_FILE'] = '/etc/zabbix/ssl/frontend.crt';
```

$ZBX_SERVER_TLS['CERTIFICATE_ISSUER']

Izdevēja atšķirīgais nosaukums (DN), kas jāsalīdzina ar servera sertifikātu.

Piemērs:

```
$ZBX_SERVER_TLS['CERTIFICATE_ISSUER']  = 'CN=MyZabbixCA';
```

$ZBX_SERVER_TLS['CERTIFICATE_SUBJECT']

Sertifikāta subjekta atšķirīgais vārds (DN), kas jāsalīdzina ar servera sertifikātu.

Piemērs:

```
$ZBX_SERVER_TLS['CERTIFICATE_SUBJECT'] = 'CN=zabbix-server.example.com';
```

$TELEMETRY_PROVIDERS[]

Telemetrijas datu avota konfigurācija.

ClickHouse atbalstītie konfigurācijas parametri: - `url` - iestatiet HTTP vai HTTPS URL, kas tiek izmantots savienojuma izveidei ar telemetrijas datu avotu. - `db` - iestatiet datubāzes nosaukumu. - `username` - iestatiet lietotājvārdu, kas tiek izmantots, lai pieteiktos datubāzē. - `password` - iestatiet paroli, kas tiek izmantota, lai pieteiktos datubāzē. - `ssl_cert_file`, `ssl_key_file`, `ssl_key_password`, `ssl_verify_peer`, `ssl_verify_host`, `ssl_ca_location`, `ssl_ca_file` - iestatiet šīs opcijas, lai konfigurētu drošu (TLS/SSL) savienojumu ar datubāzi. - `vault_path` - iestatiet ceļu uz Vault noslēpumu, kurā tiek glabātas `username` un `password` vērtības. Neizmantojiet kopā ar opcijām `username` un `password`.

Piemērs:

```
$TELEMETRY_PROVIDERS[] = [
    'provider' => 'clickhouse',
    'url' => 'http://localhost:8123',
    'db' => 'zabbix',
    'username' => 'zabbix',
    'password' => 'zabbix'
];
```

API

Pārskats Zabbix API ļauj programmatiski izgūt un modificēt Zabbix konfigurāciju, kā arī nodrošina piekļuvi vēsturiskajiem datiem. To plaši izmanto, lai:

- Izveidotu jaunas lietojumprogrammas darbam ar Zabbix.
- Integrētu Zabbix trešās puses programmatūrā.
- Automatizētu rutīnas uzdevumus.

Zabbix API ir uz HTTP balstīts API, un tas tiek piegādāts kā daļa no 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.

Papildinformāciju par protokolu un JSON skatiet JSON-RPC 2.0 specifikācijā un JSON formāta sākumlapā.

Papildinformāciju par Zabbix funkcionalitātes integrēšanu jūsu Python lietojumprogrammās skatiet Python bibliotēkā Zabbix.

Note:
Lietotāju piekļuve Zabbix, tostarp gan konfigurācijai, gan vēsturiskajiem datiem, ir atkarīga no lietotāja tipa, piešķirtās lietotāja lomas un lietotāju grupām.

Struktūra API sastāv no vairākiem metodēm, kas nomināli ir grupētas atsevišķās API. Katra no metodēm veic vienu konkrētu uzdevumu. Piemēram, `host.create` metode pieder pie *host* API un tiek izmantota jaunu hosts izveidei. Vēsturiski API dažkārt tiek dēvētas par "klasēm".

Lielākajā daļā API ir vismaz četras metodes: `get`, `create`, `update` un `delete`, kas attiecīgi tiek izmantotas datu izgūšanai, izveidei, atjaunināšanai un dzēšanai, taču dažas API var nodrošināt pilnīgi atšķirīgu metožu kopu.

Pieprasījumu veikšana Kad esat iestatījis lietotāja saskarni, varat izmantot attālinātus HTTP pieprasījumus, lai izsauktu API. Lai to izdarītu, jums jānosūta HTTP POST pieprasījumi uz `api_jsonrpc.php` failu, kas atrodas lietotāja saskarnes direktorijā.
 Piemēram, ja jūsu Zabbix lietotāja saskarne ir instalēta zem `https://example.com/zabbix`, HTTP pieprasījums, lai izsauktu `apiinfo.version` metodi, varētu 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, kas iestatīta uz vienu no šīm vērtībām: `application/json-rpc`, `application/json` vai `application/jsonrequest`.

Pieprasījuma objektam jāietver šādas īpašības:

- `jsonrpc` - API izmantotā JSON-RPC protokola versija (Zabbix API ievieš JSON-RPC 2.0 versiju);
- `method` - API metode, kas tiek izsaukta;
- `params` - parametri, kas tiks nodoti API metodei;
- `id` - patvaļīgs pieprasījuma identifikators (ja tas nav norādīts, API pieprasījumu apstrādā kā paziņojumu).

Ja pieprasījums ir pareizs, API atgrieztajai atbildei vajadzētu izskatīties šādi:

```
{
    "jsonrpc": "2.0",
    "result": "8.0.0",
    "id": 1
}
```

Savukārt atbildes objektā ir šādas īpašības:

- `jsonrpc` - JSON-RPC protokola versija;
- `result` - metodes atgrieztie dati;
- `id` - atbilstošā pieprasījuma identifikators.

Autentifikācija Lai piekļūtu jebkuriem datiem Zabbix, jums ir jāizmanto viens no šiem veidiem:

- Izmantojiet esošu API tokenu, kas izveidots Zabbix lietotāja saskarnē, sadaļā *Users* > *API tokens*, vai izveidots, izmantojot Token API.
- Izmantojiet autentifikācijas tokenu, kas iegūts ar `user.login` metodi.

Piemēram, ja jūs vēlētos iegūt jaunu autentifikācijas tokenu, piesakoties kā standarta *Admin* lietotājs, 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 jūs norādītu akreditācijas datus pareizi, API atgrieztajā atbildē vajadzētu būt 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).

Piemēra darbplūsma Nākamā sadaļa detalizētāk iepazīstina ar vairākiem lietošanas piemēriem.

Hosts izgūšana Tagad jums ir derīgs lietotāja autentifikācijas tokens (turpmākajos piemēros attēlots kā mainīgais), ko var izmantot, lai piekļūtu datiem Zabbix. Piemēram, varat izmantot metodi `host.get`, lai izgūtu visu konfigurēto hosts 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 vaicājumu var nodot `--data` argumentā.

`data.json`

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": [
            "hostid",
            "host"
        ],
        "selectInterfaces": [
            "interfaceid",
```

```
            "ip"
        ]
    },
    "id": 2
}
```

Atbildes objekts saturēs pieprasītos datus par hosts:

```
{
    "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 uzskaitīt objekta īpašības, kuras vēlaties izgūt. Tādējādi jūs izvairīsieties no visa datu apjoma izgūšanas.

Jauna vienuma izveide Tagad izveidojiet jaunu vienumu uz hosts "Zabbix server", izmantojot datus, ko ieguvāt no iepriekšējā `host.get` pieprasījuma. 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_":
```

Veiksmīga atbilde saturēs jaunizveidotā vienuma ID, ko var izmantot, lai atsauktos uz vienumu turpmākajos pieprasījumos:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "24759"
        ]
    },
    "id": 3
}
```

Note:
Metode `item.create`, tāpat kā citas *izveides metodes*, var pieņemt arī objektu masīvus un izveidot vairākus vienumus ar vienu API izsaukumu.

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":"Processor load is too high
```

Veiksmīga 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}'
```

Veiksmīga atbilde saturēs atjaunināto vienuma ID:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "10092"
        ]
    },
    "id": 5
}
```

> **Note:**
> Metode `item.update`, tāpat kā 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
```

Veiksmīga atbilde saturēs atjaunināto trigeru ID:

```
{
    "jsonrpc": "2.0",
    "result": {
        "triggerids": [
            "13938",
            "13939"
        ]
    },
    "id": 6
}
```

Kļūdu apstrāde Līdz šim viss, ko esat mēģinājis, ir darbojies pareizi. Bet kas notiktu, ja jūs mēģinātu veikt nepareizu API izsaukumu? Mēģiniet izveidot vēl vienu hostu, izsaucot `host.create`, bet izlaižot obligāto `groups` parametru:

```
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,
```

Atbildē tad būs kļūdas ziņojums:

```
{
    "jsonrpc": "2.0",
    "error": {
        "code": -32602,
        "message": "Invalid params.",
        "data": "No groups for host \"Linux server\"."
    },
    "id": 7
}
```

Ja ir notikusi kļūda, atbildes objektā `result` īpašības vietā būs `error` īpašība 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 šāda veida kļūdas apstrādāt korekti.

API versijas Lai vienkāršotu API versēšanu, kopš Zabbix 2.0.4 API versija sakrīt ar paša Zabbix versiju. Varat izmantot metodi `apiinfo.version`, lai uzzinātu, ar kuru API versiju strādājat. Tas var būt noderīgi, lai pielāgotu savu lietojumprogrammu konkrētai versijai raksturīgu funkciju izmantošanai.

Zabbix garantē funkciju atpakaļsaderību vienas galvenās versijas ietvaros. Veicot ar atpakaļsaderību nesavietojamas izmaiņas starp galvenajiem laidieniem, Zabbix parasti atstāj vecās funkcijas kā novecojušas nākamajā laidienā un noņem tās tikai laidienā pēc tam. Dažkārt Zabbix var noņemt funkcijas starp galvenajiem laidieniem, nenodrošinot nekādu atpakaļsaderību. Ir svarīgi nekad nepaļauties uz novecojušām funkcijām un pēc iespējas ātrāk pāriet uz jaunākām alternatīvām.

> **Note:**
> Visas API veiktās izmaiņas varat sekot līdzi API izmaiņu žurnālā.

Papildu lasāmviela Tagad jums ir pietiekami daudz zināšanu, lai sāktu strādāt ar Zabbix API, tomēr neapstājieties šeit. Papildu 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

APM Zabbix API ļauj piekļūt telemetrijas datiem, kas apkopoti lietojumprogrammu veiktspējas uzraudzībai (APM).

Trases

Iegūstiet trases un to veidojošos spanus.

APM trases API | APM span API

Metrikas

Iegūstiet metrikas datu punktus.

APM metrikas API

Žurnāli

Iegūt žurnāla ierakstus.

APM žurnāla 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.

Service API

Datu apkopošana Zabbix API ļauj pārvaldīt jūsu uzraudzības sistēmas konfigurāciju.

Hosti un hostu grupas

Pārvaldiet hostu grupas, hostus un visu, kas ar tiem saistīts, tostarp hostu saskarnes, hostu makro un uzturēšanas periodus.

Host API | Host group API | User macro API | Value map API | Maintenance API

Vienumi

Definējiet vienumus uzraudzībai.

Vienumu API

Trigeri

Konfigurējiet trigerus, lai saņemtu paziņojumus par problēmām savā sistēmā.
Pārvaldiet trigeru atkarības.

Trigger 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

Ierīces

Pārvaldiet mobilās ierīces, ko izmantot ar Zabbix mobilo lietotni.

Ierīču 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

Zabbix informācija Iegūstiet Zabbix API versiju vai pēdējās atjaunināšanas laiku no Zabbix servera.

API info API | Server 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 izgūt 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*.

APM izsekošana

Šī klase ir paredzēta darbam ar izsekojumiem, kas apkopoti lietojumprogrammu veiktspējas uzraudzībai (APM).

Objektu atsauces:

- APM izsekošana
- Notikums
- Saite

Pieejamās metodes:

- apmtrace.get - izgūt izsekojumus

APM izsekošanas objekts

Tālāk norādītie objekti ir tieši saistīti ar `apmtrace` API.

Note:
Katrs rekvizīts, izņemot `span_count` un `error_count`, atbilst kolonnai `otel_traces` tabulā telemetrijas datu avotā. Lielākā daļa rekvizītu atbilst kolonnai ar tādu pašu nosaukumu (piemēram, rekvizīts `span_name` atbilst kolonnai `SpanName`). Ja nosaukumi atšķiras, datu avota kolonnas ir norādītas tālāk esošajās tabulās. Kolonnu sarakstu skatiet sadaļā Atbalstītās telemetrijas datu kolonnas.

APM izsekojums

APM izsekojuma objekts attēlo izsekojumu (ceļu, ko pieprasījums veic jūsu lietojumprogrammās un pakalpojumos). Izsekojums sastāv no viena vai vairākiem span; katrs span attēlo vienu darbību.

APM izsekojuma objekts ir balstīts uz izsekojuma saknes span (span, ar kuru sākas izsekojums un kuram nav vecāka span). Visi rekvizīti satur saknes span datus, izņemot `span_count` un `error_count`, kas tiek aprēķināti no visiem izsekojuma span.

APM izsekojuma objektam ir šādi rekvizīti:

Property	Type	Description
timestamp	integer	Unix laiks (nanosekundēs), kad sākās saknes span. Property behavior: - *read-only*

Property	Type	Description
traceid	string	Izsekojuma ID, kas ir kopīgs visiem span, kuri pieder izsekojumam. Property behavior: - *read-only*
spanid	string	Saknes span ID. Property behavior: - *read-only*
trace_state	string	Papildu dati, ja tādi ir, ko izsekošanas rīki pievieno izsekojumam (skatiet W3C Trace Context). Property behavior: - *read-only*
span_name	string	Darbības nosaukums, ko attēlo saknes span (piemēram, `POST /zabbix.php`). Property behavior: - *read-only*
span_kind	string	Saknes span loma. Iespējamās vērtības: `Unspecified` - span veids nav norādīts; `Internal` - darbība, kas veikta lietojumprogrammā; `Server` - ienākoša pieprasījuma apstrāde; `Client` - izejošs pieprasījums; `Producer` - asinhronas darbības izveide (piemēram, ziņojuma nosūtīšana); `Consumer` - asinhronas darbības apstrāde. Property behavior: - *read-only*
service_name	string	Pakalpojuma nosaukums, kas izveidoja saknes span (piemēram, `zabbix-nginx`), iegūts no OpenTelemetry `service.name` resursa atribūta. Property behavior: - *read-only*
resource_attributes	object	Saknes span atslēgas un vērtības resursa atribūti (piemēram, `host.name: eabf8c2a5ddc`). Property behavior: - *read-only*
scope_name	string	Instrumentācijas tvēruma nosaukums, kas izveidoja saknes span. Property behavior: - *read-only*
scope_version	string	Instrumentācijas tvēruma versija. Property behavior: - *read-only*
span_attributes	object	Saknes span atslēgas un vērtības span atribūti (piemēram, `http.method: POST`). Property behavior: - *read-only*
duration	integer	Izsekojuma ilgums nanosekundēs. Property behavior: - *read-only*

Property	Type	Description
status_code	string	Darbības statuss, ko attēlo saknes span. Iespējamās vērtības: `Unset` - saknes span nav piešķirts tiešs statuss; `Ok` - darbība tika veiksmīgi pabeigta; `Error` - darbība tika pabeigta ar kļūdu. Property behavior: - *read-only*
status_message	string	Papildu informācija, ja tāda ir, par saknes span statusu. Property behavior: - *read-only*
events	array	Saknes span notikumu objektu masīvs, tas ir, nozīmīgi laika punkti darbības laikā (piemēram, izņēmumi). Property behavior: - *read-only*
links	array	Saknes span saišu objektu masīvs, kas norāda uz saistītiem span tajā pašā vai citā izsekojumā. Property behavior: - *read-only*
span_count	integer	Kopējais span skaits izsekojumā. Property behavior: - *read-only*
error_count	integer	Izsekojumā esošo span skaits, kuriem `status_code` ir iestatīts uz `Error`. Property behavior: - *read-only*

Notikums

Notikuma objektam ir šādas īpašības.

Property	Type	Description
timestamp	integer	Unix laiks (nanosekundēs), kad notikums notika. Datu avota kolonna: `Events.Timestamp`
name	string	Notikuma nosaukums. Datu avota kolonna: `Events.Name`
attributes	object	Atslēgas un vērtības atribūti ar papildu informāciju par notikumu, ja tāda ir. Datu avota kolonna: `Events.Attributes`

Saite

Saistes objektam ir šādas īpašības.

Īpašība	Tips	Apraksts
traceid	string	Traces ID, kurā atrodas saistītais spans. Datu avota kolonna: `Links.TraceId`
spanid	string	Saistītā spana ID. Kopā ar `traceid` tas unikāli identificē saistīto spanu. Datu avota kolonna: `Links.SpanId`
trace_state	string	Saistītā spana trases stāvoklis, ja tāds ir (skatiet W3C Trace Context). Datu avota kolonna: `Links.TraceState`

Īpašība	Tips	Apraksts
attributes	object	Atslēgas un vērtības atribūti ar papildu informāciju par saiti. Datu avota kolonna: `Links.Attributes`

apmtrace.get

Apraksts

`integer/array apmtrace.get(object parameters)`

Šī metode ļauj izgūt izsekošanas datus atbilstoši norādītajiem parametriem.

> **Note:**
> Šī metode ir pieejama visu veidu 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
time_from	timestamp	Atgriezt tikai tās izsekošanas, kuru saknes spans sākās norādītajā laikā vai vēlāk. Parametra darbība: - *obligāts*
time_till	timestamp	Atgriezt tikai tās izsekošanas, kuru saknes spans sākās pirms norādītā laika. Parametra darbība: - *obligāts*
traceids	string/array	Atgriezt tikai izsekošanas ar norādītajiem ID.
resource_attributes	object/array	Atgriezt tikai tās izsekošanas, kuru saknes spanam ir norādītie resursa atribūti. Formāts: `[{"key": "<key>", "value": "<value>", "operator": "<operator>"}, ...]`. Tukšs masīvs atgriež visas izsekošanas. Iespējamās operatora vērtības: 0 - *(noklusējums)* Satur; 1 - Ir vienāds ar; 2 - Nesatur; 3 - Nav vienāds ar; 4 - Pastāv; 5 - Nepastāv.
resource_attributes_evaltype	integer	Novērtēšanas metode nosacījumiem parametrā `resource_attributes`. Iespējamās vērtības: 0 - *(noklusējums)* And/Or; 2 - Or.

Parametrs	Tips	Apraksts
span_attributes	object/array	Atgriezt tikai tās izsekošanas, kuru saknes spanam ir norādītie spana atribūti. Formāts: `[{"key": "<key>", "value": "<value>", "operator": "<operator>"}, ...]`. Tukšs masīvs atgriež visas izsekošanas. Iespējamās operatora vērtības: 0 - *(noklusējums)* Satur; 1 - Ir vienāds ar; 2 - Nesatur; 3 - Nav vienāds ar; 4 - Pastāv; 5 - Nepastāv.
span_attributes_evaltype	integer	Novērtēšanas metode nosacījumiem parametrā `span_attributes`. Iespējamās vērtības: 0 - *(noklusējums)* And/Or; 2 - Or.
min_duration	integer	Atgriezt tikai tās izsekošanas, kuru saknes spana `duration` ir lielāks par norādīto vērtību vai vienāds ar to (nanosekundēs).
max_duration	integer	Atgriezt tikai tās izsekošanas, kuru saknes spana `duration` ir mazāks par norādīto vērtību (nanosekundēs).
filter	object	Atgriezt tikai tos rezultātus, kas precīzi atbilst norādītajam filtram. Pieņem objektu, kurā atslēgas ir rekvizītu nosaukumi, bet vērtības ir viena vērtība vai vērtību masīvs, ar ko veikt atbilstības pārbaudi. Atbalstītie rekvizīti: `traceid`, `spanid`, `trace_state`, `span_name`, `span_kind`, `service_name`, `scope_name`, `scope_version`, `status_code`, `status_message`.
search	object	Atgriezt rezultātus, kas atbilst norādītajam paraugam (reģistrs netiek ņemts vērā). Pieņem objektu, kurā atslēgas ir rekvizītu nosaukumi, bet vērtības ir meklējamās virknes. Ja nav norādītas papildu opcijas, tiek veikta `LIKE "%...%"` meklēšana. Atbalstītie rekvizīti: `trace_state`, `span_name`, `service_name`, `scope_name`, `scope_version`, `status_message`.
sortfield	string/array	Kārtot rezultātu pēc norādītajiem rekvizītiem. Iespējamās vērtības: `timestamp`, `traceid`, `spanid`, `trace_state`, `span_name`, `span_kind`, `service_name`, `scope_name`, `scope_version`, `duration`, `status_code`, `status_message`.
offset	integer	Ierakstu skaits, kas jāizlaiž pirms rezultāta atgriešanas.
countOutput	boolean	Šie parametri ir aprakstīti atsauces komentāros.
excludeSearch	boolean	
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

Lēnāko izsekojumu izgūšana

Izgūstiet divus lēnākos `zabbix-nginx` servisa izsekojumus. Meklējiet izsekojumus no 10:00 līdz 11:00 2026. gada 1. oktobrī (UTC) un iekļaujiet tikai tos, kuru izpilde ilga 1 sekundi vai ilgāk.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "apmtrace.get",
    "params": {
        "output": ["timestamp", "traceid", "span_name", "service_name", "duration", "status_code", "span_c
        "time_from": 1790848800,
        "time_till": 1790852400,
        "filter": {
            "service_name": "zabbix-nginx"
        },
        "min_duration": 1000000000,
        "sortfield": "duration",
        "sortorder": "DESC",
        "limit": 2
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "timestamp": "1790849274910226417",
            "traceid": "8a4f1c7e2b9d4e6fa0c3b5d7e9f1a2c4",
            "span_name": "POST /zabbix.php",
            "service_name": "zabbix-nginx",
            "duration": "30412907815",
            "status_code": "Error",
            "span_count": "49",
            "error_count": "2"
        },
        {
            "timestamp": "1790850512338104952",
            "traceid": "dee625f33cb0994e86c5cd4640e3b09c",
            "span_name": "POST /zabbix.php",
            "service_name": "zabbix-nginx",
            "duration": "1104826531",
            "status_code": "Unset",
            "span_count": "27",
            "error_count": "0"
        }
    ],
    "id": 1
}
```

Skatiet arī

- APM span
- APM metrika
- APM žurnāls

Avots

CApmTrace::get() failā *ui/include/classes/api/services/CApmTrace.php*.

APM metrika

Šī klase ir paredzēta darbam ar lietojumprogrammu veiktspējas uzraudzībai (APM) apkopotajām metrikām.

Objektu atsauces:

- APM metrika
- Piemērs

Pieejamās metodes:

- apmmetric.get - izgūt metrikas datu punktus

APM metrikas objekts

Tālāk norādītie objekti ir tieši saistīti ar `apmmetric` API.

Note:
Katrs rekvizīts, izņemot `type`, atbilst kolonnai telemetrijas datu avota metrikas tabulās (`otel_metrics_gauge`, `otel_metrics_sum`, `otel_metrics_histogram` un `otel_metrics_exponential_histogram`). Lielākā daļa rekvizītu atbilst kolonnai ar tādu pašu nosaukumu (piemēram, rekvizīts `metric_name` atbilst kolonnai `MetricName`). Ja nosaukumi atšķiras, datu avota kolonnas ir norādītas tālāk esošajās tabulās. Kolonnu sarakstu skatiet sadaļā Atbalstītās telemetrijas datu kolonnas.

APM metrika

APM metrikas objekts attēlo metrikas datu punktu (mērījuma vērtību vai apkopotās vērtības noteiktā laikā).

APM metrikas objektam ir šādi rekvizīti:

Rekvizīts	Tips	Apraksts
type	integer	Datu punkta tips. Iespējamās vērtības: 0 - Gauge; 1 - Sum; 2 - Histogram; 3 - Exponential histogram. Tips tiek noteikts pēc tabulas, kurā atrodas datu punkts: `otel_metrics_gauge`, `otel_metrics_sum`, `otel_metrics_histogram` vai `otel_metrics_exponential_histogram`. Rekvizīta darbība: - *tikai lasāms*
resource_attributes	object	Datu punkta atslēgas un vērtības resursa atribūti (piemēram, `host.name: eabf8c2a5ddc`). Rekvizīta darbība: - *tikai lasāms*
resource_schema_url	string	Resursa atribūtu OpenTelemetry shēmas URL, ja tāds ir. Rekvizīta darbība: - *tikai lasāms*
scope_name	string	Instrumentācijas tvēruma nosaukums, kas izveidoja datu punktu. Rekvizīta darbība: - *tikai lasāms*
scope_version	string	Instrumentācijas tvēruma versija, ja tāda ir. Rekvizīta darbība: - *tikai lasāms*
scope_attributes	object	Instrumentācijas tvēruma atslēgas un vērtības atribūti. Rekvizīta darbība: - *tikai lasāms*

Rekvizīts	Tips	Apraksts
scope_schema_url	string	Instrumentācijas tvēruma un tā atribūtu OpenTelemetry shēmas URL, ja tāds ir. Rekvizīta darbība: - *tikai lasāms*
service_name	string	Servisa nosaukums, kas izveidoja datu punktu (piemēram, `zabbix-frontend`), iegūts no OpenTelemetry resursa atribūta `service.name`. Rekvizīta darbība: - *tikai lasāms*
metric_name	string	Metrikas nosaukums, kuram pieder datu punkts (piemēram, `http.server.request.duration`). Rekvizīta darbība: - *tikai lasāms*
metric_description	string	Metrikas apraksts, ja tāds ir. Rekvizīta darbība: - *tikai lasāms*
metric_unit	string	Metrikas vērtību mērvienība, ja tāda ir (piemēram, `s` vai `By`; skatiet UCUM). Rekvizīta darbība: - *tikai lasāms*
attributes	object	Datu punkta atslēgas un vērtības atribūti (piemēram, `http.request.method: POST`). Rekvizīta darbība: - *tikai lasāms*
start_time_unix	integer	Unix laiks (nanosekundēs), kad sākās datu punkta mērījumu periods. Rekvizīta darbība: - *tikai lasāms*
time_unix	integer	Datu punkta Unix laiks (nanosekundēs). Sum, Histogram un Exponential histogram datu punktiem tas ir mērījumu perioda beigu laiks. Rekvizīta darbība: - *tikai lasāms*
value	float	Datu punkta vērtība. Attiecas uz Gauge un Sum datu punktiem. Noklusējums: 0. Rekvizīta darbība: - *tikai lasāms*
flags	integer	Datu punkta karodziņu bitmaska. Ja ir iestatīts vismazāk nozīmīgais bits (vērtība 1), datu punktam nav reģistrēta vērtība. Rekvizīta darbība: - *tikai lasāms*
exemplars	array	Exemplar objektu masīvs, tas ir, atsevišķi mērījumi, kas saglabāti kā piemēri, kopā ar trasējumu un span, kura laikā tie tika reģistrēti. Rekvizīta darbība: - *tikai lasāms*

Rekvizīts	Tips	Apraksts
aggregation_temporality	integer	Veids, kādā datu punkta vērtības tiek apkopotas laika gaitā. Attiecas uz Sum, Histogram un Exponential histogram datu punktiem. Iespējamās vērtības: 0 - *(noklusējums)* Nav norādīts; 1 - Delta (apkopots par periodu kopš iepriekšējā datu punkta); 2 - Cumulative (apkopots par periodu kopš `start_time_unix`). Rekvizīta darbība: - *tikai lasāms*
is_monotonic	boolean	Norāda, vai summa laika gaitā var tikai palielināties vai palikt nemainīga. Attiecas uz Sum datu punktiem. Noklusējums: `false`. Rekvizīta darbība: - *tikai lasāms*
count	integer	Mērījumu skaits datu punktā. Attiecas uz Histogram un Exponential histogram datu punktiem. Noklusējums: 0. Rekvizīta darbība: - *tikai lasāms*
sum	float	Visu datu punkta mērījumu summa. Attiecas uz Histogram un Exponential histogram datu punktiem. Noklusējums: 0. Rekvizīta darbība: - *tikai lasāms*
bucket_counts	array	Veselu skaitļu masīvs, kurā norādīts mērījumu skaits katrā histogrammas segmentā. Segmentus nosaka `explicit_bounds`. Attiecas uz Histogram datu punktiem. Noklusējums: `[]`. Rekvizīta darbība: - *tikai lasāms*
explicit_bounds	array	Peldošā komata skaitļu masīvs ar robežām starp histogrammas segmentiem. Attiecas uz Histogram datu punktiem. Noklusējums: `[]`. Rekvizīta darbība: - *tikai lasāms*
min	float	Mazākā datu punkta mērījumu vērtība. Attiecas uz Histogram un Exponential histogram datu punktiem. Noklusējums: 0. Rekvizīta darbība: - *tikai lasāms*

Rekvizīts	Tips	Apraksts
max	float	Lielākā datu punkta mērījumu vērtība. Attiecas uz Histogram un Exponential histogram datu punktiem. Noklusējums: 0. Rekvizīta darbība: - *tikai lasāms*
scale	integer	Eksponenciālās histogrammas mērogs, kas nosaka segmentu robežas un izšķirtspēju. Lielāks mērogs nodrošina šaurākus segmentus un precīzāku izšķirtspēju; mazāks mērogs nodrošina platākus segmentus un rupjāku izšķirtspēju. Attiecas uz Exponential histogram datu punktiem. Noklusējums: 0. Rekvizīta darbība: - *tikai lasāms*
zero_count	integer	Mērījumu skaits eksponenciālās histogrammas nulles segmentā. Attiecas uz Exponential histogram datu punktiem. Noklusējums: 0. Rekvizīta darbība: - *tikai lasāms*
positive_bucket_counts	array	Veselu skaitļu masīvs, kurā norādīts pozitīvo mērījumu skaits katrā eksponenciālās histogrammas segmentā. Segmentu robežas nosaka `scale`; pirmā segmenta indeksu iestata `positive_offset`. Attiecas uz Exponential histogram datu punktiem. Noklusējums: `[]`. Rekvizīta darbība: - *tikai lasāms*
positive_offset	integer	Pirmā segmenta indekss masīvā `positive_bucket_counts`. Attiecas uz Exponential histogram datu punktiem. Noklusējums: 0. Rekvizīta darbība: - *tikai lasāms*
negative_bucket_counts	array	Veselu skaitļu masīvs, kurā norādīts negatīvo mērījumu skaits katrā eksponenciālās histogrammas segmentā. Segmentu robežas nosaka `scale`; pirmā segmenta indeksu iestata `negative_offset`. Attiecas uz Exponential histogram datu punktiem. Noklusējums: `[]`. Rekvizīta darbība: - *tikai lasāms*
negative_offset	integer	Pirmā segmenta indekss masīvā `negative_bucket_counts`. Attiecas uz Exponential histogram datu punktiem. Noklusējums: 0. Rekvizīta darbība: - *tikai lasāms*

Piemērs

Piemēra objektam ir šādi rekvizīti:

Rekvizīts	Tips	Apraksts
traceid	string	Traces ID, ja tāds ir, kura laikā tika reģistrēts mērījums. Datu avota kolonna: `Exemplars.TraceId`.
spanid	string	Span ID, ja tāds ir, kura laikā tika reģistrēts mērījums. Datu avota kolonna: `Exemplars.SpanId`.
value	float	Mērījuma vērtība. Datu avota kolonna: `Exemplars.Value`.
time_unix	integer	Unix laiks (nanosekundēs), kad tika reģistrēts mērījums. Datu avota kolonna: `Exemplars.TimeUnix`.
filtered_attributes	object	Mērījuma atslēgas un vērtības atribūti, kas nav iekļauti datu punkta `attributes`. Datu avota kolonna: `Exemplars.FilteredAttributes`.

apmmetric.get

Apraksts

`integer/array apmmetric.get(object parameters)`

Metode ļauj izgūt metrikas datu punktus atbilstoši norādītajiem parametriem.

> **Note:**
> Šī metode ir pieejama visu veidu 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.

Parameter	Type	Description
time_from	timestamp	Atgriezt tikai tos datu punktus, kuru laiks ir norādītajā laikā vai vēlāk. Parametra darbība: - *obligāts*
time_till	timestamp	Atgriezt tikai tos datu punktus, kuru laiks ir pirms norādītā laika. Parametra darbība: - *obligāts*
types	integer/array	Atgriezt tikai norādīto tipu datu punktus. Atbalstīto metriku tipu sarakstu skatiet APM metrikas objekta lapā.
with_flags_on	integer	Atgriezt tikai tos datu punktus, kuriem ir iestatīti visi norādītie karodziņi. Lai norādītu vairākus karodziņus, izmantojiet to vērtību summu.
with_flags_off	integer	Atgriezt tikai tos datu punktus, kuriem nav iestatīts neviens no norādītajiem karodziņiem. Lai norādītu vairākus karodziņus, izmantojiet to vērtību summu.

Parameter	Type	Description
resource_attributes	object/array	Atgriezt tikai tos datu punktus, kuriem ir norādītie resursa atribūti. Formāts: `[{"key": "<key>", "value": "<value>", "operator": "<operator>"}, ...]`. Tukšs masīvs atgriež visus datu punktus. Iespējamās operatora vērtības: 0 - *(noklusējums)* Satur; 1 - Ir vienāds ar; 2 - Nesatur; 3 - Nav vienāds ar; 4 - Pastāv; 5 - Nepastāv.
resource_attributes_evaltype	integer	Novērtēšanas metode nosacījumiem parametrā `resource_attributes`. Iespējamās vērtības: 0 - *(noklusējums)* And/Or; 2 - Or.
scope_attributes	object/array	Atgriezt tikai tos datu punktus, kuriem ir norādītie tvēruma atribūti. Formāts: `[{"key": "<key>", "value": "<value>", "operator": "<operator>"}, ...]`. Tukšs masīvs atgriež visus datu punktus. Iespējamās operatora vērtības: 0 - *(noklusējums)* Satur; 1 - Ir vienāds ar; 2 - Nesatur; 3 - Nav vienāds ar; 4 - Pastāv; 5 - Nepastāv.
scope_attributes_evaltype	integer	Novērtēšanas metode nosacījumiem parametrā `scope_attributes`. Iespējamās vērtības: 0 - *(noklusējums)* And/Or; 2 - Or.
attributes	object/array	Atgriezt tikai tos datu punktus, kuriem ir norādītie atribūti. Formāts: `[{"key": "<key>", "value": "<value>", "operator": "<operator>"}, ...]`. Tukšs masīvs atgriež visus datu punktus. Iespējamās operatora vērtības: 0 - *(noklusējums)* Satur; 1 - Ir vienāds ar; 2 - Nesatur; 3 - Nav vienāds ar; 4 - Pastāv; 5 - Nepastāv.
attributes_evaltype	integer	Novērtēšanas metode nosacījumiem parametrā `attributes`. Iespējamās vērtības: 0 - *(noklusējums)* And/Or; 2 - Or.
filter	object	Atgriezt tikai tos rezultātus, kas precīzi atbilst norādītajam filtram. Pieņem objektu, kurā atslēgas ir rekvizītu nosaukumi, bet vērtības ir viena vērtība vai vērtību masīvs, ar ko veikt salīdzināšanu. Atbalstītie rekvizīti: `resource_schema_url`, `scope_schema_url`, `scope_name`, `scope_version`, `service_name`, `metric_name`, `metric_unit`, `flags`.

Parameter	Type	Description
search	object	Atgriezt rezultātus, kas atbilst norādītajam paraugam (reģistrjutība netiek ņemta vērā). Pieņem objektu, kurā atslēgas ir rekvizītu nosaukumi, bet vērtības ir virknes, kurās meklēt. Ja nav norādītas papildu opcijas, tiek veikta `LIKE "%…%"` meklēšana. Atbalstītie rekvizīti: `resource_schema_url`, `scope_schema_url`, `scope_name`, `scope_version`, `service_name`, `metric_name`, `metric_description`, `metric_unit`.
sortfield	string/array	Kārtot rezultātu pēc norādītajiem rekvizītiem. Iespējamās vērtības: `type`, `resource_schema_url`, `scope_schema_url`, `scope_name`, `scope_version`, `service_name`, `metric_name`, `metric_unit`, `start_time_unix`, `time_unix`.
offset	integer	Ierakstu skaits, ko izlaist pirms rezultāta atgriešanas.
countOutput	boolean	Šie parametri ir aprakstīti atsauces komentāros.
excludeSearch	boolean	
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

Histogrammas datu punktu izgūšana

Izgūstiet jaunāko `http.server.request.duration` histogrammas metrikas datu punktu pakalpojumam `zabbix-frontend`.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "apmmetric.get",
    "params": {
        "output": ["metric_name", "metric_unit", "time_unix", "count", "sum", "min", "max", "explicit_boun
        "time_from": 1790848800,
        "time_till": 1790852400,
        "types": 2,
        "filter": {
            "service_name": "zabbix-frontend",
            "metric_name": "http.server.request.duration"
        },
        "sortfield": "time_unix",
        "sortorder": "DESC",
        "limit": 1
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": [
```

```
        {
            "metric_name": "http.server.request.duration",
            "metric_unit": "s",
            "time_unix": "1790852390000000000",
            "count": "1842",
            "sum": "412.773",
            "min": "0.0061",
            "max": "1.098",
            "explicit_bounds": ["0.005", "0.05", "0.1", "0.25", "0.5", "1"],
            "bucket_counts": ["0", "211", "904", "512", "167", "41", "7"],
            "exemplars": [
                {
                    "traceid": "dee625f33cb0994e86c5cd4640e3b09c",
                    "spanid": "3f8b2c6d1e9a7054",
                    "value": "1.098",
                    "time_unix": "1790850513439147623",
                    "filtered_attributes": {}
                }
            ]
        }
    ],
    "id": 1
}
```


Skatiet arī

- APM izsekojums
- APM spans
- APM žurnāls

Avots

CApmMetric::get() failā *ui/include/classes/api/services/CApmMetric.php*.

APM span

Šī klase ir paredzēta darbam ar lietojumprogrammu veiktspējas uzraudzībai (APM) apkopotajiem span.

Objektu atsauces:

- APM span

Pieejamās metodes:

- apmspan.get - izgūt span.

APM span objekts

Tālāk norādītie objekti ir tieši saistīti ar `apmspan` API.

> **Note:**
> Katrs rekvizīts atbilst kolonnai `otel_traces` tabulā telemetrijas datu avotā. Kolonnu sarakstu skatiet sadaļā Atbalstītās telemetrijas datu kolonnas.

APM span

APM span objekts attēlo span (vienu darbību trace ietvaros, piemēram, HTTP pieprasījumu, datubāzes vaicājumu vai funkcijas izsaukumu).

Span veido hierarhiju, kas sākas ar saknes span, kurš attēlo pieprasījuma pirmo darbību. Katrs cits span ir tā span bērnelements, kura darbības laikā tas tika sākts.

APM span objektam ir šādas īpašības:

Property	Type	Description
timestamp	integer	Unix laiks (nanosekundēs), kad span tika sākts. Property behavior: - *read-only*
traceid	string	Trace ID, kuram pieder span. Property behavior: - *read-only*
spanid	string	Span ID. Property behavior: - *read-only*
parent_spanid	string	Vecāka span ID. Tukšs saknes span gadījumā. Property behavior: - *read-only*
trace_state	string	Papildu dati, ja tādi ir, ko izsekošanas rīki pievieno trace (skatiet W3C Trace Context). Property behavior: - *read-only*
span_name	string	Span attēlotās darbības nosaukums (piemēram, `mysqli_query`). Property behavior: - *read-only*
span_kind	string	Span loma. Iespējamās vērtības: `Unspecified` - span veids nav norādīts; `Internal` - lietotnē veikta darbība; `Server` - ienākoša pieprasījuma apstrāde; `Client` - izejošs pieprasījums; `Producer` - asinhronas darbības izveide (piemēram, ziņojuma nosūtīšana); `Consumer` - asinhronas darbības apstrāde. Property behavior: - *read-only*
service_name	string	Pakalpojuma nosaukums, kas izveidoja span (piemēram, `zabbix-frontend`), iegūts no OpenTelemetry resursa atribūta `service.name`. Property behavior: - *read-only*
resource_attributes	object	Span atslēgas un vērtības resursa atribūti (piemēram, `host.name: eabf8c2a5ddc`). Property behavior: - *read-only*
scope_name	string	Instrumentācijas tvēruma nosaukums, kas izveidoja span. Property behavior: - *read-only*
scope_version	string	Instrumentācijas tvēruma versija. Property behavior: - *read-only*

Property	Type	Description
span_attributes	object	Span atslēgas un vērtības span atribūti (piemēram, `db.system: mysql`). Property behavior: - *read-only*
duration	integer	Span ilgums nanosekundēs. Property behavior: - *read-only*
status_code	string	Span attēlotās darbības statuss. Iespējamās vērtības: `Unset` - span nav piešķirts skaidrs statuss; `Ok` - darbība tika veiksmīgi pabeigta; `Error` - darbība tika pabeigta ar kļūdu. Property behavior: - *read-only*
status_message	string	Papildu informācija, ja tāda ir, par span statusu. Property behavior: - *read-only*
events	array	Span event objektu masīvs, tas ir, nozīmīgi laika punkti darbības laikā (piemēram, izņēmumi). Property behavior: - *read-only*
links	array	Span link objektu masīvs, kas norāda uz saistītiem span tajā pašā vai citā trace. Property behavior: - *read-only*

apmspan.get

Apraksts

`integer/array apmspan.get(object parameters)`

Šī metode ļauj izgūt spanus atbilstoši norādītajiem parametriem.

Note:
Šī metode ir pieejama visu veidu 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.

Parameter	Type	Description
time_from	timestamp	Atgriezt tikai tos intervālus, kas sākās norādītajā laikā vai vēlāk. Parametra darbība: - *obligāts*
time_till	timestamp	Atgriezt tikai tos intervālus, kas sākās pirms norādītā laika. Parametra darbība: - *obligāts*
traceids	string/array	Atgriezt tikai tos intervālus, kas pieder norādītajām trasēm.

Parameter	Type	Description
spanids	string/array	Atgriezt tikai tos intervālus ar norādītajiem ID.
parent_spanids	string/array	Atgriezt tikai tos intervālus ar norādītajiem vecāka intervāliem.
resource_attributes	object/array	Atgriezt tikai tos intervālus ar norādītajiem resursa atribūtiem. Formāts: `[{"key": "<key>", "value": "<value>", "operator": "<operator>"}, ...]`. Tukšs masīvs atgriež visus intervālus. Iespējamās operatora vērtības: 0 - *(noklusējuma)* Satur; 1 - Ir vienāds ar; 2 - Nesatur; 3 - Nav vienāds ar; 4 - Pastāv; 5 - Nepastāv.
resource_attributes_evaltype	integer	Izvērtēšanas metode nosacījumiem parametrā `resource_attributes`. Iespējamās vērtības: 0 - *(noklusējuma)* And/Or; 2 - Or.
span_attributes	object/array	Atgriezt tikai tos intervālus ar norādītajiem intervāla atribūtiem. Formāts: `[{"key": "<key>", "value": "<value>", "operator": "<operator>"}, ...]`. Tukšs masīvs atgriež visus intervālus. Iespējamās operatora vērtības: 0 - *(noklusējuma)* Satur; 1 - Ir vienāds ar; 2 - Nesatur; 3 - Nav vienāds ar; 4 - Pastāv; 5 - Nepastāv.
span_attributes_evaltype	integer	Izvērtēšanas metode nosacījumiem parametrā `span_attributes`. Iespējamās vērtības: 0 - *(noklusējuma)* And/Or; 2 - Or.
min_duration	integer	Atgriezt tikai tos intervālus, kuru ilgums ir lielāks par norādīto vērtību vai vienāds ar to, nanosekundēs.
max_duration	integer	Atgriezt tikai tos intervālus, kuru ilgums ir mazāks par norādīto vērtību, nanosekundēs.
filter	object	Atgriezt tikai tos rezultātus, kas precīzi atbilst norādītajam filtram. Pieņem objektu, kurā atslēgas ir rekvizītu nosaukumi, bet vērtības ir viena vērtība vai vērtību masīvs, ar ko veikt salīdzināšanu. Atbalstītie rekvizīti: `traceid`, `spanid`, `parent_spanid`, `trace_state`, `span_name`, `span_kind`, `service_name`, `scope_name`, `scope_version`, `status_code`, `status_message`.
search	object	Atgriezt rezultātus, kas atbilst norādītajam paraugam (reģistrjutība netiek ņemta vērā). Pieņem objektu, kurā atslēgas ir rekvizītu nosaukumi, bet vērtības ir meklējamās virknes. Ja nav norādītas papildu opcijas, tiks veikta `LIKE "%...%"` meklēšana. Atbalstītie rekvizīti: `trace_state`, `span_name`, `service_name`, `scope_name`, `scope_version`, `status_message`.

Parameter	Type	Description
sortfield	string/array	Kārtot rezultātu pēc norādītajiem rekvizītiem. Iespējamās vērtības: `timestamp`, `traceid`, `spanid`, `parent_spanid`, `trace_state`, `span_name`, `span_kind`, `service_name`, `scope_name`, `scope_version`, `duration`, `status_code`, `status_message`.
offset	integer	Ierakstu skaits, kas jāizlaiž pirms rezultāta atgriešanas.
countOutput	boolean	Šie parametri ir aprakstīti atsauces komentāros.
excludeSearch	boolean	
limit	integer	
output	query	
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

Izsekojuma span hierarhijas izgūšana

Izgūstiet pirmos četrus izsekojuma span, kas sakārtoti pēc sākuma laika, izsekojumam "dee625f33cb0994e86c5cd4640e3b09c". Īpašība `parent_spanid` parāda, kā span ir savstarpēji saistīti: saknes span nav vecāka, bet katrs pārējais span atsaucas uz span, kas to sāka.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "apmspan.get",
    "params": {
        "output": ["spanid", "parent_spanid", "service_name", "span_name", "duration", "status_code"],
        "time_from": 1790848800,
        "time_till": 1790852400,
        "traceids": "dee625f33cb0994e86c5cd4640e3b09c",
        "sortfield": "timestamp",
        "sortorder": "ASC",
        "limit": 4
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "spanid": "7c1a9e2f4b6d8a03",
            "parent_spanid": "",
            "service_name": "zabbix-nginx",
            "span_name": "POST /zabbix.php",
            "duration": "1104826531",
            "status_code": "Unset"
        },
        {
            "spanid": "3f8b2c6d1e9a7054",
            "parent_spanid": "7c1a9e2f4b6d8a03",
            "service_name": "zabbix-frontend",
```

```
            "span_name": "POST /zabbix.php",
            "duration": "1098230418",
            "status_code": "Unset"
        },
        {
            "spanid": "b3390ad41931ed08",
            "parent_spanid": "3f8b2c6d1e9a7054",
            "service_name": "zabbix-frontend",
            "span_name": "mysqli::real_connect",
            "duration": "36671360",
            "status_code": "Ok"
        },
        {
            "spanid": "e41f7a2c9d5b3086",
            "parent_spanid": "3f8b2c6d1e9a7054",
            "service_name": "zabbix-frontend",
            "span_name": "mysqli_query",
            "duration": "19704118",
            "status_code": "Ok"
        }
    ],
    "id": 1
}
```


Skatiet arī

- APM izsekošana
- APM žurnāls
- APM metrika

Avots

CApmSpan::get() failā *ui/include/classes/api/services/CApmSpan.php*.

APM žurnāls

Šī klase ir paredzēta darbam ar žurnāliem, kas apkopoti lietojumprogrammu veiktspējas uzraudzībai (APM).

Objektu atsauces:

- APM žurnāls

Pieejamās metodes:

- apmlog.get - izgūt žurnāla ierakstus

APM žurnāla objekts

Tālāk norādītie objekti ir tieši saistīti ar `apmlog` API.

> **Note:**
> Katrs rekvizīts atbilst kolonnai ar tādu pašu nosaukumu `otel_logs` tabulā telemetrijas datu avotā (piemēram, rekvizīts `severity_text` atbilst kolonnai `SeverityText`). Kolonnu sarakstu skatiet sadaļā Atbalstītās telemetrijas datu kolonnas.

APM žurnāls

APM žurnāla objekts attēlo žurnāla ierakstu (notikuma teksta ierakstu, piemēram, par neveiksmīgu maksājumu vai datubāzes taimautu). Žurnāla ieraksts satur informāciju par notikumu, kas noticis noteiktā laikā, tostarp tā nopietnības līmeni, ziņojumu un atribūtus. Ja žurnāla ieraksts tika izveidots operācijas laikā, kas ir daļa no izsekojuma, tas satur arī šī izsekojuma un tvēruma ID.

APM žurnāla objektam ir šādi rekvizīti:

Rekvizīts	Tips	Apraksts
timestamp	integer	Unix laiks (nanosekundēs), kad notika notikums. Rekvizīta darbība: - *tikai lasāms*
traceid	string	Izsekojuma ID, ja tāds ir, kura laikā tika izveidots žurnāla ieraksts. Rekvizīta darbība: - *tikai lasāms*
spanid	string	Tvēruma ID, ja tāds ir, kura laikā tika izveidots žurnāla ieraksts. Rekvizīta darbība: - *tikai lasāms*
trace_flags	integer	Žurnāla ieraksta izsekojuma konteksta izsekojuma karodziņi bitmaskas veidā. Ja ir iestatīts vismazāk nozīmīgais bits (vērtība 1), izsekojums tiek paraugots. Rekvizīta darbība: - *tikai lasāms*
severity_text	string	Žurnāla ieraksta nopietnības līmenis, ko nodrošina lietotne (piemēram, `INFO`, `WARN` vai `ERROR`). Rekvizīta darbība: - *tikai lasāms*
severity_number	integer	Žurnāla ieraksta nopietnības līmenis kā skaitlis. Iespējamās vērtības: 0 - UNSPECIFIED; 1-4 - TRACE; 5-8 - DEBUG; 9-12 - INFO; 13-16 - WARN; 17-20 - ERROR; 21-24 - FATAL. Rekvizīta darbība: - *tikai lasāms*
service_name	string	Pakalpojuma nosaukums, kas izveidoja žurnāla ierakstu (piemēram, `zabbix-frontend`), iegūts no OpenTelemetry resursa atribūta `service.name`. Rekvizīta darbība: - *tikai lasāms*
body	string	Žurnāla ieraksta ziņojums. Rekvizīta darbība: - *tikai lasāms*
resource_schema_url	string	OpenTelemetry resursa atribūtu shēmas URL, ja tāds ir. Rekvizīta darbība: - *tikai lasāms*
resource_attributes	object	Žurnāla ieraksta atslēgas un vērtības resursa atribūti (piemēram, `host.name: eabf8c2a5ddc`). Rekvizīta darbība: - *tikai lasāms*
scope_schema_url	string	Instrumentācijas tvēruma un tā atribūtu OpenTelemetry shēmas URL, ja tāds ir. Rekvizīta darbība: - *tikai lasāms*

Rekvizīts	Tips	Apraksts
scope_name	string	Instrumentācijas tvēruma nosaukums, kas izveidoja žurnāla ierakstu. Rekvizīta darbība: - *tikai lasāms*
scope_version	string	Instrumentācijas tvēruma versija, ja tāda ir. Rekvizīta darbība: - *tikai lasāms*
scope_attributes	object	Instrumentācijas tvēruma atslēgas un vērtības atribūti. Rekvizīta darbība: - *tikai lasāms*
log_attributes	object	Žurnāla ieraksta atslēgas un vērtības žurnāla atribūti (piemēram, `log.file.path: /var/log/zabbix/zabbix_frontend.log`). Rekvizīta darbība: - *tikai lasāms*
event_name	string	Notikuma tipa nosaukums, ko attēlo žurnāla ieraksts, ja tāds ir. Rekvizīta darbība: - *tikai lasāms*

apmlog.get

Apraksts

`integer/array apmlog.get(object parameters)`

Metode ļauj izgūt žurnāla ierakstus atbilstoši norādītajiem parametriem.

> **Note:**
> Šī metode ir pieejama visu veidu 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
time_from	timestamp	Atgriež tikai to notikumu žurnāla ierakstus, kas notikuši norādītajā laikā vai vēlāk. Parametra darbība: - *obligāts*
time_till	timestamp	Atgriež tikai to notikumu žurnāla ierakstus, kas notikuši pirms norādītā laika. Parametra darbība: - *obligāts*
traceids	string/array	Atgriež tikai žurnāla ierakstus, kas izveidoti norādīto izsekojumu laikā.
spanids	string/array	Atgriež tikai žurnāla ierakstus, kas izveidoti norādīto span laikā.
with_flags_on	integer	Atgriež tikai tos žurnāla ierakstus, kuriem ir iestatīti visi norādītie izsekojuma karodziņi. Lai norādītu vairākus karodziņus, izmantojiet to vērtību summu.
with_flags_off	integer	Atgriež tikai tos žurnāla ierakstus, kuriem nav iestatīts neviens no norādītajiem izsekojuma karodziņiem. Lai norādītu vairākus karodziņus, izmantojiet to vērtību summu.

Parametrs	Tips	Apraksts
resource_attributes	object/array	Atgriež tikai žurnāla ierakstus ar norādītajiem resursa atribūtiem. Formāts: `[{"key": "<key>", "value": "<value>", "operator": "<operator>"}, ...]`. Tukšs masīvs atgriež visus žurnāla ierakstus. Iespējamās operatora vērtības: 0 - *(noklusējums)* Satur; 1 - Ir vienāds ar; 2 - Nesatur; 3 - Nav vienāds ar; 4 - Pastāv; 5 - Nepastāv.
resource_attributes_evaltype	integer	Novērtēšanas metode nosacījumiem parametrā `resource_attributes`. Iespējamās vērtības: 0 - *(noklusējums)* Un/Vai; 2 - Vai.
scope_attributes	object/array	Atgriež tikai žurnāla ierakstus ar norādītajiem tvēruma atribūtiem. Formāts: `[{"key": "<key>", "value": "<value>", "operator": "<operator>"}, ...]`. Tukšs masīvs atgriež visus žurnāla ierakstus. Iespējamās operatora vērtības: 0 - *(noklusējums)* Satur; 1 - Ir vienāds ar; 2 - Nesatur; 3 - Nav vienāds ar; 4 - Pastāv; 5 - Nepastāv.
scope_attributes_evaltype	integer	Novērtēšanas metode nosacījumiem parametrā `scope_attributes`. Iespējamās vērtības: 0 - *(noklusējums)* Un/Vai; 2 - Vai.
log_attributes	object/array	Atgriež tikai žurnāla ierakstus ar norādītajiem žurnāla atribūtiem. Formāts: `[{"key": "<key>", "value": "<value>", "operator": "<operator>"}, ...]`. Tukšs masīvs atgriež visus žurnāla ierakstus. Iespējamās operatora vērtības: 0 - *(noklusējums)* Satur; 1 - Ir vienāds ar; 2 - Nesatur; 3 - Nav vienāds ar; 4 - Pastāv; 5 - Nepastāv.
log_attributes_evaltype	integer	Novērtēšanas metode nosacījumiem parametrā `log_attributes`. Iespējamās vērtības: 0 - *(noklusējums)* Un/Vai; 2 - Vai.
filter	object	Atgriež tikai tos rezultātus, kas precīzi atbilst norādītajam filtram. Pieņem objektu, kurā atslēgas ir rekvizītu nosaukumi, bet vērtības ir viena vērtība vai vērtību masīvs, ar ko veikt salīdzināšanu. Atbalstītie rekvizīti: `traceid`, `spanid`, `trace_flags`, `severity_text`, `severity_number`, `service_name`, `resource_schema_url`, `scope_schema_url`, `scope_name`, `scope_version`, `event_name`.

Parametrs	Tips	Apraksts
search	object	Atgriež rezultātus, kas atbilst norādītajam paraugam (reģistrjutība netiek ņemta vērā). Pieņem objektu, kurā atslēgas ir rekvizītu nosaukumi, bet vērtības ir meklējamās virknes. Ja nav norādītas papildu opcijas, tiek veikta LIKE "%...%" meklēšana. Atbalstītie rekvizīti: `severity_text`, `service_name`, `body`, `resource_schema_url`, `scope_schema_url`, `scope_name`, `scope_version`, `event_name`.
sortfield	string/array	Kārto rezultātu pēc norādītajiem rekvizītiem. Iespējamās vērtības: `timestamp`, `traceid`, `spanid`, `trace_flags`, `severity_text`, `severity_number`, `service_name`, `resource_schema_url`, `scope_schema_url`, `scope_name`, `scope_version`, `event_name`.
offset	integer	Ierakstu skaits, kas jāizlaiž pirms rezultāta atgriešanas.
countOutput	boolean	Šie parametri ir aprakstīti atsauces komentāros.
excludeSearch	boolean	
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

Izsekošanas kļūdu žurnālu izgūšana

Izgūstiet žurnāla ierakstus ar ERROR smaguma pakāpi (smaguma pakāpes numuri 17-20), kas ir saistīti ar izsekošanu "8a4f1c7e2b9d4e6fa0c3b5d7e9f1a2c4".

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "apmlog.get",
    "params": {
        "output": "extend",
        "time_from": 1790848800,
        "time_till": 1790852400,
        "traceids": "8a4f1c7e2b9d4e6fa0c3b5d7e9f1a2c4",
        "filter": {
            "severity_number": [17, 18, 19, 20]
        },
        "sortfield": "timestamp",
        "sortorder": "ASC"
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
```

```
            "timestamp": "1790849305091452376",
            "traceid": "8a4f1c7e2b9d4e6fa0c3b5d7e9f1a2c4",
            "spanid": "5d2e8f1a9c3b7046",
            "trace_flags": "1",
            "severity_text": "ERROR",
            "severity_number": "17",
            "service_name": "zabbix-frontend",
            "body": "mysqli_query(): (HY000/1205): Lock wait timeout exceeded; try restarting transaction"
            "resource_schema_url": "https://opentelemetry.io/schemas/1.26.0",
            "resource_attributes": {
                "service.name": "zabbix-frontend",
                "host.name": "6583ceeb5f94",
                "host.arch": "x86_64",
                "os.type": "linux"
            },
            "scope_schema_url": "",
            "scope_name": "zabbix-frontend",
            "scope_version": "",
            "scope_attributes": {},
            "log_attributes": {
                "log.file.path": "/var/log/zabbix/zabbix_frontend.log",
                "code.filepath": "/usr/share/zabbix/include/db.inc.php"
            },
            "event_name": ""
        }
    ],
    "id": 1
}
```


Žurnāla ziņojumu meklēšana

Iegūstiet laiku, pakalpojumu un ziņojumu 10 jaunākajiem žurnāla ierakstiem, kuru ziņojumā ir ietverts teksts ”timeout”.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "apmlog.get",
    "params": {
        "output": ["timestamp", "service_name", "severity_text", "body"],
        "time_from": 1790848800,
        "time_till": 1790852400,
        "search": {
            "body": "timeout"
        },
        "sortfield": "timestamp",
        "sortorder": "DESC",
        "limit": 10
    },
    "id": 1
}
```

Atbilde:


```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "timestamp": "1790849305091452376",
            "service_name": "zabbix-frontend",
            "severity_text": "ERROR",
            "body": "mysqli_query(): (HY000/1205): Lock wait timeout exceeded; try restarting transaction"
        },
        {
            "timestamp": "1790848961730284115",
```

```
            "service_name": "zabbix-frontend",
            "severity_text": "ERROR",
            "body": "mysqli_query(): (HY000/1205): Lock wait timeout exceeded; try restarting transaction"
        }
    ],
    "id": 1
}
```


Skatiet arī

- APM izsekojums
- APM spans
- APM metrika

Avots

CApmLog::get() failā *ui/include/classes/api/services/CApmLog.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)`

Šī metode ļauj izgūt atklātos hosts atbilstoši norādītajiem parametriem.

> **Note:**
> Šī metode ir pieejama jebkura tipa lietotājiem. Tiesības izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā User roles.

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)`

Šī metode ļauj izgūt atklātos pakalpojumus atbilstoši norādītajiem parametriem.

Note:
Šī metode ir pieejama jebkura tipa lietotājiem. Tiesības izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā Lietotāja 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	

Parametrs	Tips	Apraksts
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 noteikums

Atklāšanas noteikuma objekts definē tīkla atklāšanas noteikumu. Tam ir šādas īpašības.

Property	Type	Description
druleid	ID	Atklāšanas noteikuma ID. Īpašības darbība: - *tikai lasāms* - *obligāts* atjaunināšanas darbībām
iprange	string	Viens vai vairāki pārbaudāmie IP diapazoni, atdalīti ar komatiem. Papildinformāciju par atbalstītajiem IP diapazonu formātiem skatiet sadaļā tīkla atklāšanas konfigurēšana. Īpašības darbība: - *obligāts* izveides darbībām
name	string	Atklāšanas noteikuma nosaukums. Īpašības darbība: - *obligāts* izveides darbībām
delay	string	Atklāšanas noteikuma izpildes intervāls. Pieņem sekunžu skaitu (piemēram, 1, 60, 300), vērtību ar laika sufiksu (piemēram, 30s, 1m, 2h, 1d) vai lietotāja makro. Noklusējums: 1h.
proxyid	ID	Atklāšanai izmantotā starpniekservera ID.

Property	Type	Description
status	integer	Norāda, vai atklāšanas noteikums ir iespējots. Iespējamās vērtības: 0 - *(noklusējums)* iespējots; 1 - atspējots.
concurrency_max	integer	Maksimālais vienlaicīgo pārbaužu skaits katram atklāšanas noteikumam. Iespējamās vērtības: 0 - *(noklusējums)* neierobežots pārbaužu skaits; 1 - viena pārbaude; 2-999 - pielāgots pārbaužu skaits.
error	string	Kļūdas teksts, ja atklāšanas noteikuma izpildes laikā ir 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. Tiesības izsaukt šo metodi var atsaukt 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. Tiesības izsaukt šo metodi var atsaukt lietotāju lomu 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)`

Šī metode ļauj izgūt atklāšanas noteikumus atbilstoši norādītajiem parametriem.

> **Note:**
> Šī metode ir pieejama jebkura tipa lietotājiem. Tiesības izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Skatiet User roles, lai iegūtu papildu informāciju.

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.
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	

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

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. Tiesības izsaukt šo metodi var atsaukt lietotāju lomu 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”
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”

Īpašība	Tips	Apraksts
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”
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*

Īpašība	Tips	Apraksts
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)`

Šī metode ļauj izgūt atklāšanas pārbaudes atbilstoši norādītajiem parametriem.

> **Note:**
> Šī metode ir pieejama jebkura tipa lietotājiem. Tiesības 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
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.

Parametrs	Tips	Apraksts
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
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. Tiesības izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā User roles.

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 atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā 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. Tiesības izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā Lietotāja 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 atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā User roles.

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. Tiesības izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā Lietotāja 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:**
> Mezgli tiek izveidoti ar Zabbix serveris un tos nevar modificēt, izmantojot API.

Augstas pieejamības mezgla objektam ir šādas īpašības.

Property	Type	Description
ha_nodeid	ID	Mezgla ID.
name	string	Mezgla nosaukums, kas piešķirts, izmantojot `zabbix_server.conf` konfigurācijas ierakstu HANodeName. Tukšs serveris, kas darbojas patstāvīgā režīmā.
address	string	IP vai DNS nosaukums, no kura mezgls pieslēdzas.
port	integer	Ports, kurā mezgls darbojas.
lastaccess	integer	Heartbeat laiks, tas ir, pēdējā atjauninājuma 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)`

Šī metode ļauj izgūt augstas pieejamības klastera mezglu sarakstu atbilstoši norādītajiem parametriem.

> **Note:**
> Šī metode ir pieejama tikai *Super admin* lietotāja tipiem. Papildu informāciju skatiet 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 - *(default)* Iekšējā; 1 - LDAP.
http_auth_enabled	integer	HTTP autentifikācija. Iespējamās vērtības: 0 - *(default)* Atspējota; 1 - Iespējota. Property behavior: - *supported*, ja `$ZBX_FEATURE_FLAGS['http_auth_enabled']` ir iestatīts uz `true` lietotāja saskarnes konfigurācijas failā.
http_login_form	integer	Noklusējuma pieteikšanās forma. Iespējamās vērtības: 0 - *(default)* Zabbix pieteikšanās forma; 1 - HTTP pieteikšanās forma. Property behavior: - *supported*, ja `$ZBX_FEATURE_FLAGS['http_auth_enabled']` ir iestatīts uz `true` lietotāja saskarnes konfigurācijas failā.

Property	Type	Description
http_strip_domains	string	Noņemamais domēna nosaukums. Property behavior: - *supported*, ja `$ZBX_FEATURE_FLAGS['http_auth_enabled']` ir iestatīts uz `true` lietotāja saskarnes konfigurācijas failā.
http_case_sensitive	integer	HTTP reģistrjutīga pieteikšanās. Iespējamās vērtības: 0 - Izslēgta; 1 - *(default)* Ieslēgta. Property behavior: - *supported*, ja `$ZBX_FEATURE_FLAGS['http_auth_enabled']` ir iestatīts uz `true` lietotāja saskarnes konfigurācijas failā.
ldap_auth_enabled	integer	LDAP autentifikācija. Iespējamās vērtības: 0 - *(default)* Atspējota; 1 - Iespējota.
ldap_case_sensitive	integer	LDAP reģistrjutīga pieteikšanās. Iespējamās vērtības: 0 - Izslēgta; 1 - *(default)* Ieslēgta.
ldap_userdirectoryid	ID	Noklusējuma lietotāju direktorija ID LDAP autentifikācijai. Tiek izmantots lietotāju grupām, kurām `gui_access` ir iestatīts uz LDAP vai System default. Property behavior: - *required*, ja `ldap_auth_enabled` ir iestatīts uz ”Enabled”
saml_auth_enabled	integer	SAML autentifikācija. Iespējamās vērtības: 0 - *(default)* Atspējota; 1 - Iespējota.
saml_case_sensitive	integer	SAML reģistrjutīga pieteikšanās. Iespējamās vērtības: 0 - Izslēgta; 1 - *(default)* Ieslēgta.
passwd_min_length	integer	Minimālā paroles garuma prasība. Iespējamās vērtības ir no 1 līdz 70. Noklusējums: 8.
passwd_check_rules	integer	Paroles pārbaudes noteikumi. Iespējamās bitkartes vērtības: 0 - Pārbaudīt paroles garumu; 1 - Pārbaudīt, vai parole izmanto latīņu alfabēta lielos un mazos burtus; 2 - Pārbaudīt, vai parole izmanto ciparus; 4 - Pārbaudīt, vai parole izmanto speciālās rakstzīmes; 8 - *(default)* Pārbaudīt, vai parole nav bieži izmantoto paroļu sarakstā un nesatur vārda ”Zabbix” vai lietotāja vārda, uzvārda vai lietotājvārda atvasinājumus. Šis ir bitmaskas lauks; ir pieņemama jebkura iespējamo bitkartes vērtību summa (piemēram, 15 visu noteikumu pārbaudei).

Property	Type	Description
ldap_jit_status	integer	LDAP nodrošināšanas statuss. Iespējamās vērtības: 0 - Atspējots konfigurētajiem LDAP IdP; 1 - Iespējots konfigurētajiem LDAP IdP.
saml_jit_status	integer	SAML nodrošināšanas statuss. Iespējamās vērtības: 0 - Atspējots konfigurētajiem SAML IdP; 1 - Iespējots konfigurētajiem SAML IdP.
jit_provision_interval	string	Laika intervāls starp JIT nodrošināšanas pieprasījumiem pieteikušamies lietotājam. Pieņem sekunžu skaitu (piemēram, 3600) vai vērtību ar laika sufiksu (piemēram, 3600s, 60m, 1h, 1d, 1M, 1y), tostarp atbalsta mēnešus un gadus. Minimālā vērtība: 1h. Noklusējums: 1h.
disabled_usrgrpid	ID	Pieejams tikai LDAP nodrošināšanai. Lietotāju grupas ID, kurai piešķirt nenodrošināto lietotāju. Lietotāju grupai jābūt atspējotai, un pēc konfigurēšanas to nevar iespējot vai dzēst. Property behavior: - *required*, ja `ldap_jit_status` ir iestatīts uz "Enabled for configured LDAP IdPs" vai `saml_jit_status` ir iestatīts uz "Enabled for configured SAML IdPs"
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 ir iespējota MFA. Property behavior: - *required*, ja `mfa_status` ir iestatīts uz "Enabled"

authentication.get

Apraksts

`object authentication.get(object parameters)`

Šī metode ļauj izgūt autentifikācijas objektu atbilstoši norādītajiem parametriem.

> **Note:**
> Šī metode ir pieejama tikai lietotāja tipam *Super admin*. Tiesības izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Papildinformāciju skatiet sadaļā Lietotāja 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. Tiesības izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā User roles.

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 autoreģistrācijas objektu atbilstoši norādītajiem parametriem.

> **Note:**
> Šī metode ir pieejama tikai *Super admin* lietotāja tipam. Tiesības izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet User roles.

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. Tiesības izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet User roles.

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. Tiesības izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāja 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 tālāk norādītie rekvizīti.

Rekvizīts	Tips	Apraksts
actionid	ID	Darbības ID. Rekvizīta darbība: - *tikai lasāms* - *obligāts* atjaunināšanas operācijām
esc_period	string	Noklusējuma operācijas soļa ilgums. Tam jābūt vismaz 60 sekundēm. Var norādīt sekunžu skaitu (piemēram, 60, 300), vērtību ar laika sufiksu (piemēram, 30s, 1m, 2h) vai lietotāja makro. Rekvizīta darbība: - *atbalstīts*, ja `eventsource` ir iestatīts uz ”event created by a trigger”, ”internal event” vai ”event created on service status update”
eventsource	integer	Notikumu veids, ko apstrādās darbība. Atbalstīto notikumu veidu sarakstu skatiet notikuma `source` rekvizītā. Rekvizīta darbība: - *konstants* - *obligāts* izveides operācijām
name	string	Darbības nosaukums. Rekvizīta darbība: - *obligāts* izveides operācijām
status	integer	Norāda, vai darbība ir iespējota vai atspējota. Iespējamās vērtības: 0 - *(noklusējums)* iespējota; 1 - atspējota.
pause_symptoms	integer	Norāda, vai eskalācija ir jāaptur, ja notikums ir simptomu notikums. Iespējamās vērtības: 0 - neapturēt eskalāciju simptomu problēmām; 1 - *(noklusējums)* apturēt eskalāciju simptomu problēmām. Rekvizīta darbība: - *atbalstīts*, ja `eventsource` ir iestatīts uz ”event created by a trigger”
pause_suppressed	integer	Norāda, vai apkopes periodos eskalācija ir jāaptur. Iespējamās vērtības: 0 - neapturēt eskalāciju; 1 - *(noklusējums)* apturēt eskalāciju. Rekvizīta darbība: - *atbalstīts*, ja `eventsource` ir iestatīts uz ”event created by a trigger”

Rekvizīts	Tips	Apraksts
notify_if_canceled	integer	Norāda, vai jānosūta paziņojums, kad eskalācija tiek atcelta. Iespējamās vērtības: 0 - nesūtīt paziņojumu, kad eskalācija tiek atcelta; 1 - *(noklusējums)* nosūtīt paziņojumu, kad eskalācija tiek atcelta. Rekvizīta darbība: - *atbalstīts*, ja `eventsource` ir iestatīts uz ”event created by a trigger”

Darbības operācija

Darbības operācijas objekts definē operāciju, kas tiks veikta, kad darbība tiks izpildīta. Tam ir tālāk norādītās īpašības.

Īpašība	Tips	Apraksts
operationtype	integer	Operācijas tips. Iespējamās vērtības: 0 - nosūtīt ziņojumu; 1 - globālais skripts; 2 - pievienot hostu; 3 - noņemt hostu; 4 - pievienot hostu grupai; 5 - noņemt no hostu grupas; 6 - piesaistīt 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 Action object `eventsource` ir iestatīts uz ”event created by a trigger” vai ”event created on service status update”: 0 - ”send message”; 1 - ”global script”. Iespējamā vērtība, ja Action object `eventsource` ir iestatīts uz ”internal event”: 0 - ”send message”. Īpašības darbība: - *obligāta*
esc_period	string	Eskalācijas soļa ilgums sekundēs. Tam jābūt lielākam par 60 sekundēm. Pieņem sekunžu skaitu (piemēram, 60, 300), vērtību ar laika sufiksu (piemēram, `30s`, `1m`, `2h`) vai lietotāja makro. Ja iestatīta vērtība 0 vai 0s, tiks izmantots noklusējuma darbības eskalācijas periods. Noklusējums: 0s. Īpašības darbība: - *atbalstīta*, ja Action object `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: - *atbalstīta*, ja Action object `eventsource` ir iestatīts uz ”event created by a trigger”, ”internal event” vai ”event created on service status update”
esc_step_to	integer	Solis, ar kuru pabeigt eskalāciju. Noklusējums: 1. Īpašības darbība: - *atbalstīta*, ja Action object `eventsource` ir iestatīts uz ”event created by a trigger”, ”internal event” vai ”event created on service status update”
evaltype	integer	Operācijas nosacījumu izvērtēšanas metode. Iespējamās vērtības: 0 - *(noklusējums)* And/Or; 1 - And; 2 - Or.
opcommand	object	Izpildāmais globālais skripts. Globālajam skriptam jābūt definētai `scriptid` īpašībai. Īpašības darbība: - *obligāta*, ja `operationtype` ir iestatīts uz ”global script”
opcommand_grp	array	Hostu grupas, kurās izpildīt globālos skriptus. Hostu grupām jābūt definētai `groupid` īpašībai. Īpašības darbība: - *obligāta*, ja `operationtype` ir iestatīts uz ”global script” un `opcommand_hst` nav iestatīts
opcommand_hst	array	Hosts, kuros izpildīt globālos skriptus. Hostiem jābūt definētai `hostid` īpašībai. Īpašības darbība: - *obligāta*, ja `operationtype` ir iestatīts uz ”global script” un `opcommand_grp` nav iestatīts
opconditions	array	Darbības nosacījumi, kas tiek izmantoti trigeru darbībām. Darbības nosacījumu objekts ir detalizēti aprakstīts tālāk.
opgroup	array	Hostu grupas, kurām pievienot hostus. Hostu grupām jābūt definētai `groupid` īpašībai. Īpašības darbība: - *obligāta*, ja `operationtype` ir iestatīts uz ”add to host group” vai ”remove from host group”
opmessage	object	Objekts, kas satur datus par ziņojumu, kuru nosūta operācija. Darbības ziņojuma objekts ir detalizēti aprakstīts tālāk. Īpašības darbība: - *obligāta*, ja `operationtype` ir iestatīts uz ”send message”

Īpašība	Tips	Apraksts
opmessage_grp	array	Lietotāju grupas, kurām nosūtīt ziņojumus. Lietotāju grupām jābūt definētai `usrgrpid` īpašībai. Īpašības darbība: - *obligāta*, ja `operationtype` ir iestatīts uz ”send message” un `opmessage_usr` nav iestatīts
opmessage_usr	array	Lietotāji, kuriem nosūtīt ziņojumus. Lietotājiem jābūt definētai `userid` īpašībai. Īpašības darbība: - *obligāta*, ja `operationtype` ir iestatīts uz ”send message” un `opmessage_grp` nav iestatīts
optemplate	array	Veidnes, ko piesaistīt hostiem. Veidnēm jābūt definētai `templateid` īpašībai. Īpašības darbība: - *obligāta*, ja `operationtype` ir iestatīts uz ”link template” vai ”unlink template”
opinventory	object	Inventarizācijas režīms, kas jāiestata hostam. Inventarizācijai jābūt definētai `inventory_mode` īpašībai. Īpašības darbība: - *obligāta*, 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: - *atbalstīta*, 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.

Property	Type	Apraksts
conditiontype	integer	Nosacījuma tips. Iespējamās vērtības, ja Action object `eventsource` ir iestatīts uz ”event created by a trigger”: 0 - hostu grupa; 1 - hosts; 2 - trigeris; 3 - notikuma nosaukums; 4 - trigera smaguma pakāpe; 6 - laika periods; 13 - hostu veidne; 16 - problēma ir apspiesta; 25 - notikuma tags; 26 - notikuma taga vērtība. Iespējamās vērtības, ja Action object `eventsource` ir iestatīts uz ”event created by a discovery rule”: 7 - hosta IP adrese; 8 - atklātā servisa tips; 9 - atklātā servisa ports; 10 - atklāšanas statuss; 11 - darbspējas vai dīkstāves ilgums; 12 - saņemtā vērtība; 18 - atklāšanas noteikums; 19 - atklāšanas pārbaude; 20 - starpniekserveris; 21 - atklāšanas objekts; 29 - starpniekserveru grupa. Iespējamās vērtības, ja Action object `eventsource` ir iestatīts uz ”event created by active agent autoregistration”: 20 - starpniekserveris; 22 - hosta nosaukums; 24 - hosta metadati; 29 - starpniekserveru grupa. Iespējamās vērtības, ja Action object `eventsource` ir iestatīts uz ”internal event”: 0 - hostu grupa; 1 - hosts; 13 - hostu veidne; 23 - notikuma tips; 25 - notikuma tags; 26 - notikuma taga vērtība. Iespējamās vērtības, ja Action object `eventsource` ir iestatīts uz ”event created on service status update”: 25 - notikuma tags; 26 - notikuma taga vērtība; 27 - serviss; 28 - servisa nosaukums. Property behavior: - *required*
value	string	Vērtība, ar kuru salīdzināt. Property behavior: - *required*

Property	Type	Apraksts
value2	string	Sekundārā vērtība, ar kuru salīdzināt. Property behavior: - *required* if `eventsource` of Action object is set to ”event created by a trigger”, `conditiontype` is set to any possible value for trigger actions, and the type of condition (see below) is ”26” - *required* if `eventsource` of Action object is set to ”internal event”, `conditiontype` is set to any possible value for internal actions, and the type of condition (see below) is ”26” - *required* if `eventsource` of Action object is set to ”event created on service status update”, `conditiontype` is set to any possible value for service actions, and the type of condition (see below) is ”26”
formulaid	string	Patvaļīgs unikāls ID, ko izmanto, lai atsauktos uz nosacījumu pielāgotā izteiksmē. Var saturēt tikai lielos burtus. ID lietotājam jādefinē, modificējot filtra nosacījumus, taču pēc tam, kad tie tiek pieprasīti, tas tiks ģenerēts no jauna.
operator	integer	Nosacījuma operators. Iespējamās vērtības: 0 - *(default)* ir vienāds ar; 1 - nav vienāds ar; 2 - satur; 3 - nesatur; 4 - ir iekļauts; 5 - ir lielāks par vai vienāds ar; 6 - ir mazāks par vai vienāds ar; 7 - nav iekļauts; 8 - atbilst; 9 - neatbilst; 10 - Jā; 11 - Nē.

Note:
Lai labāk izprastu, kā izmantot filtrus ar dažādu tipu izteiksmēm, skatiet piemērus metožu action.get un action.create lapās.

Tālāk norādītie operatori un vērtības tiek atbalstīti katram nosacījuma tipam.

Nosacījums	Nosacījuma nosaukums	Atbalstītie operatori	Paredzamā vērtība
0	Hostu grupa	ir vienāds ar, nav vienāds ar	Hostu grupas ID.
1	Hosts	ir vienāds ar, nav vienāds ar	Hosta ID.
2	Trigeris	ir vienāds ar, nav vienāds ar	Trigera ID.
3	Notikuma nosaukums	satur, nesatur	Notikuma nosaukums.
4	Trigera smaguma pakāpe	ir vienāds ar, nav vienāds ar, ir lielāks par vai vienāds ar, ir mazāks par vai vienāds ar	Trigera smaguma pakāpe. Atbalstīto trigera smaguma pakāpju sarakstu skatiet trigera `severity` īpašībā.
5	Trigera vērtība	ir vienāds ar	Trigera vērtība. Atbalstīto trigera vērtību sarakstu skatiet trigera `value` īpašībā.
6	Laika periods	ir iekļauts, nav iekļauts	Laiks, kad notikums tika aktivizēts, kā laika periods.

Nosacījums	Nosacījuma nosaukums	Atbalstītie operatori	Paredzamā vērtība
7	Hosta IP adrese	ir vienāds ar, nav vienāds ar	Viens vai vairāki pārbaudāmie IP diapazoni, atdalīti ar komatiem. Papildinformāciju par atbalstītajiem IP diapazonu formātiem skatiet sadaļā tīkla atklāšanas konfigurēšana.
8	Atklātā servisa tips	ir vienāds ar, nav vienāds ar	Atklātā servisa tips. Servisa tips atbilst atklāšanas pārbaudes tipam, kas tika izmantots servisa noteikšanai. Atbalstīto tipu sarakstu skatiet atklāšanas pārbaudes type īpašībā.
9	Atklātā servisa ports	ir vienāds ar, nav vienāds ar	Viens vai vairāki portu diapazoni, atdalīti ar komatiem.
10	Atklāšanas statuss	ir vienāds ar	Atklātā objekta statuss. Iespējamās vērtības: 0 - hosts vai serviss darbojas; 1 - hosts vai serviss nedarbojas; 2 - hosts vai serviss ir atklāts; 3 - hosts vai serviss ir pazudis.
11	Darbspējas vai dīkstāves ilgums	ir lielāks par vai vienāds ar, ir mazāks par vai vienāds ar	Laiks sekundēs, kas norāda, cik ilgi atklātais objekts atrodas pašreizējā statusā.
12	Saņemtās vērtības	ir vienāds ar, nav vienāds ar, ir lielāks par vai vienāds ar, ir mazāks par vai vienāds ar, satur, nesatur	Vērtība, kas tiek atgriezta, veicot Zabbix aģenta, SNMPv1, SNMPv2 vai SNMPv3 atklāšanas pārbaudi.
13	Hostu veidne	ir vienāds ar, nav vienāds ar	Saistītās veidnes ID.
16	Problēma ir apspiesta	Jā, Nē	Vērtība nav nepieciešama: operatora ”Jā” izmantošana nozīmē, ka problēmai jābūt apspiestai, bet ”Nē” - ka tā nedrīkst būt apspiesta.
18	Atklāšanas noteikums	ir vienāds ar, nav vienāds ar	Atklāšanas noteikuma ID.
19	Atklāšanas pārbaude	ir vienāds ar, nav vienāds ar	Atklāšanas pārbaudes ID.
20	Starpniekserveris	ir vienāds ar, nav vienāds ar	Starpniekservera ID.
21	Atklāšanas objekts	ir vienāds ar	Objekta tips, kas aktivizēja atklāšanas notikumu. Iespējamās vērtības: 1 - atklāts hosts; 2 - atklāts serviss.
22	Hosta nosaukums	satur, nesatur, atbilst, neatbilst	Hosta nosaukums. Automātiskās reģistrācijas nosacījumos operatoriem *atbilst* un *neatbilst* tiek atbalstītas regulārās izteiksmes.
23	Notikuma tips	ir vienāds ar	Konkrēts iekšējais notikums. Iespējamās vērtības: 0 - vienums stāvoklī ”not supported”; 1 - vienums stāvoklī ”normal”; 2 - LLD noteikums stāvoklī ”not supported”; 3 - LLD noteikums stāvoklī ”normal”; 4 - trigeris stāvoklī ”unknown”; 5 - trigeris stāvoklī ”normal”.
24	Hosta metadati	satur, nesatur, atbilst, neatbilst	Automātiski reģistrētā hosta metadati. Operatoriem *atbilst* un *neatbilst* tiek atbalstītas regulārās izteiksmes.

Nosacījums	Nosacījuma nosaukums	Atbalstītie operatori	Paredzamā vērtība
25	Tags	ir vienāds ar, nav vienāds ar, satur, nesatur	Notikuma tags.
26	Taga vērtība	ir vienāds ar, nav vienāds ar, satur, nesatur	Notikuma taga vērtība.
27	Serviss	ir vienāds ar, nav vienāds ar	Servisa ID.
28	Servisa nosaukums	ir vienāds ar, nav vienāds ar	Servisa nosaukums.
29	Starpniekserveru grupa	ir vienāds ar, nav vienāds ar	Starpniekserveru grupas ID.

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. Tiesības izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā 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 atsaukt lietotāja lomu iestatījumos. Plašāku informāciju skatiet sadaļā Lietotāju lomas.
 Lietotājs nevarēs dzēst darbību, kuras filtra nosacījumos ir ietverts nepieejams starpniekserveris vai starpniekserveru grupa. Plašāku informāciju skatiet sadaļā lietotāju grupas konfigurācija (*Starpniekserveru piekļuves saraksts*).

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)`

Šī metode ļauj izgūt darbības atbilstoši norādītajiem parametriem.

> **Note:**
> Šī metode ir pieejama jebkura tipa lietotājiem. Tiesības izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā Lietotāja 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. Tiesības izsaukt šo metodi var atsaukt lietotāju lomu iestatījumos. Plašāku informāciju skatiet sadaļā User roles.

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 grafu prototipus.

> **Note:**
> Šī metode ir pieejama tikai *Admin* un *Super admin* lietotāju tipiem. Tiesības izsaukt metodi var atsaukt lietotāju lomu iestatījumos. Plašāku informāciju skatiet sadaļā 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 grafu prototipus.

> **Note:**
> Šī metode ir pieejama tikai *Admin* un *Super admin* lietotāju tipiem. Tiesības izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā 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)`

Šī metode ļauj izgūt grafiku prototipus atbilstoši norādītajiem parametriem.

> **Note:**
> Šī metode ir pieejama jebkura tipa lietotājiem. Tiesības 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
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 grafu prototipus.

Note:
Šī metode ir pieejama tikai *Admin* un *Super admin* lietotāju tipiem. Tiesības izsaukt šo metodi var atsaukt lietotāju lomu iestatījumos. Plašāku informāciju skatiet sadaļā 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)`

Šī metode ļauj izgūt grafika vienumus atbilstoši norādītajiem parametriem.

Note:
Šī metode ir pieejama jebkura tipa lietotājiem. Tiesības izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā User roles.

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. Tiesības izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā Lietotāja 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. Tiesības izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā 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. Tiesības izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāja 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. Tiesības izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet 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. Tiesības izsaukt šo metodi var atsaukt lietotāja lomu iestatījumos. Plašāku informāciju skatiet sadaļā 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. Tiesības izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā 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)`

Šī metode ļauj izgūt hostu prototipus atbilstoši norādītajiem parametriem.

Note:
Šī metode ir pieejama jebkura tipa lietotājiem. Tiesības izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāja 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. Tiesības izsaukt šo metodi var atsaukt lietotāju lomu 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*.

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

Hosts objektam ir šādas īpašības.

Property	Type	Description
hostid	ID	Hosts ID. Īpašības darbība: - *tikai lasāms* - *obligāts* atjaunināšanas darbībām
host	string	Hosts tehniskais nosaukums. Īpašības darbība: - *obligāts* izveides darbībām
description	text	Hosts apraksts.
flags	integer	Hosts izcelsme. Iespējamās vērtības: 0 - parasts hosts; 4 - no prototipa pārveidots hosts. Īpašības darbība: - *tikai lasāms*
inventory_mode	integer	Hosts 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)* noklusējuma; 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 - atzvanīšana; 2 - *(noklusējums)* lietotājs; 3 - operators; 4 - administrators; 5 - OEM.
ipmi_username	string	IPMI lietotājvārds.
maintenance_from	timestamp	Aktīvās apkopes sākuma laiks. Īpašības darbība: - *tikai lasāms*
maintenance_status	integer	Aktīvās apkopes statuss. Iespējamās vērtības: 0 - *(noklusējums)* apkope nenotiek; 1 - apkope notiek. Īpašības darbība: - *tikai lasāms*
maintenance_type	integer	Aktīvās apkopes veids. Iespējamās vērtības: 0 - *(noklusējums)* apkope ar datu apkopošanu; 1 - apkope bez datu apkopošanas. Īpašības darbība: - *tikai lasāms*
maintenanceid	ID	Apkopes ID, kas pašlaik ir aktīva hosts. Īpašības darbība: - *tikai lasāms*
name	string	Hosts redzamais nosaukums. Noklusējums: `host` īpašības vērtība.
monitored_by	integer	Avots, kas tiek izmantots hosts uzraudzībai. Iespējamās vērtības: 0 - *(noklusējums)* Zabbix serveris; 1 - starpniekserveris; 2 - starpniekserveru grupa. Īpašības darbība: - *obligāts*, ja `select_server_for_monitoring` ir iestatīts uz ”Disabled”.
proxyid	ID	Uzraudzībai izmantotā starpniekservera ID. Īpašības darbība: - *obligāts*, ja `monitored_by` ir iestatīts uz ”Proxy”

Property	Type	Description
proxy_groupid	ID	Uzraudzībai izmantotās starpniekserveru grupas ID. Īpašības darbība: - *obligāts*, ja `monitored_by` ir iestatīts uz ”Proxy group”
status	integer	Hosts statuss un funkcija. Iespējamās vērtības: 0 - *(noklusējums)* iespējots; 1 - atspējots.
tls_connect	integer	Savienojumi ar hosts. Iespējamās vērtības: 1 - *(noklusējums)* bez šifrēšanas; 2 - PSK; 4 - sertifikāts.
tls_accept	integer	Savienojumi no hosts. 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.
tls_psk_identity	string	PSK identitāte; tai jābūt saistītai tikai ar vienu PSK (starp automātisko reģistrāciju, hosts un starpniekserveriem). Neiekļaujiet sensitīvu informāciju PSK identitātē, jo tā tiek nosūtīta nešifrētā veidā pa tīklu, lai informētu saņēmēju, kuru PSK izmantot. Īpašības darbība: - *tikai rakstāms* - *obligāts*, ja `tls_connect` ir iestatīts uz ”PSK” vai `tls_accept` satur bitu ”PSK”
tls_psk	string	Iepriekš koplietota atslēga (PSK); tai jāsatur vismaz 32 heksadecimālie cipari. Īpašības darbība: - *tikai rakstāms* - *obligāts*, ja `tls_connect` ir iestatīts uz ”PSK” vai `tls_accept` satur bitu ”PSK”
active_available	integer	Hosts 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, izmantojot 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

ID	Īpašība	Type	Apraksts	Maksimālais garums
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
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 hosts.

Note:
Šī metode ir pieejama tikai *Admin* un *Super admin* lietotāju tipiem. Tiesības izsaukt šo metodi var atsaukt lietotāja lomas 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
}
```

Hosts izveide ar PSK šifrēšanu

Izveidojiet hostu ar nosaukumu "PSK host" un konfigurētu PSK šifrēšanu. Ņemiet vērā, ka hostam jābūt iepriekš konfigurētam PSK lietošanai.

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 hosts.

> **Note:**
> Šī metode ir pieejama tikai *Admin* un *Super admin* lietotāju tipiem. Tiesības izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāja 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 izgūt hosts atbilstoši norādītajiem parametriem.

> **Note:**
> Šī metode ir pieejama jebkura tipa lietotājiem. Tiesības izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Skatiet Lietotāja lomas, lai iegūtu papildu informāciju.

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.

Parametrs	Tips	Apraksts
severities	integer/array	Atgriezt hostus, kuriem ir tikai problēmas ar norādītajām nopietnības pakāpēm. Piemērojams tikai tad, ja problēmas objekts ir trigeris.
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 saistītos objektus visiem norādītajiem hostiem.

Note:
Šī metode ir pieejama tikai *Admin* un *Super admin* lietotāju tipiem. Tiesības izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā User roles.

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 hosts.

Note:
Šī metode ir pieejama tikai *Admin* un *Super admin* lietotāju tipiem. Tiesības izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāju lomas.

Parametri

`(object)` Parametri, kas satur atjaunināmo hostu ID un objektus, kas jānoņem.

Parametrs	Type	Apraksts
hostids	ID/array	Atjaunināmo 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 hosts.

> **Note:**
> Šī metode ir pieejama tikai *Admin* un *Super admin* lietotāju tipiem. Tiesības izsaukt šo metodi var atsaukt 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
}
```

Hosts šifrēšanas atjaunināšana

Atjauniniet hostu ”10590”, lai tas izmantotu tikai PSK šifrēšanu savienojumiem no hosta uz Zabbix serveri, un mainiet PSK identitāti un PSK atslēgu. Ņemiet vērā, ka hostam ir 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*.

Hosts saskarne

Šī klase nodrošina piekļuvi hosts saskarnēm.

Objektu atsauces:

- Hosts saskarne
  - Detalizēta informācija

Pieejamā metode:

- hostinterface.get - izgūt hosts saskarnes

Hosts saskarnes tiek izveidotas, atjauninātas un dzēstas, izmantojot hosts metodes.

Hosta saskarnes objekts

Tālāk norādītie objekti ir tieši saistīti ar `hostinterface` API.

Host interfeiss

Host interfeisa objektam ir šādas īpašības.

Property	Type	Description
interfaceid	ID	Interfeisa ID. Īpašības darbība: - *tikai lasāms* - *obligāts* atjaunināšanas darbībām
available	integer	Host interfeisa pieejamība. Iespējamās vērtības: 0 - *(noklusējums)* nezināms; 1 - pieejams; 2 - nepieejams. Īpašības darbība: - *tikai lasāms*
hostid	ID	Hosta, kuram pieder interfeiss, ID. Īpašības darbība: - *konstants* - *obligāts* izveides darbībām
type	integer	Interfeisa tips. Iespējamās vērtības: 1 - aģents; 2 - SNMP; 3 - IPMI; 4 - JMX. Īpašības darbība: - *obligāts* izveides darbībām
ip	string	Interfeisa izmantotā IP adrese. Var būt tukša, ja savienojums tiek izveidots, izmantojot DNS. Īpašības darbība: - *obligāts* izveides darbībām
dns	string	Interfeisa izmantotais DNS nosaukums. Var būt tukšs, ja savienojums tiek izveidots, izmantojot IP. Īpašības darbība: - *obligāts* izveides darbībām
port	string	Interfeisa izmantotais porta numurs. Var saturēt lietotāja makro. Īpašības darbība: - *obligāts* izveides darbībām
useip	integer	Norāda, vai savienojums jāizveido, izmantojot IP. Iespējamās vērtības: 0 - savienoties, izmantojot hosta DNS nosaukumu; 1 - savienoties, izmantojot hosta IP adresi. Īpašības darbība: - *obligāts* izveides darbībām

Property	Type	Description
main	integer	Norāda, vai interfeiss hostā tiek izmantots kā noklusējuma interfeiss. Par noklusējuma interfeisu hostā var iestatīt tikai vienu konkrēta tipa interfeisu. Iespējamās vērtības: 0 - nav noklusējuma; 1 - noklusējuma. Īpašības darbība: - *obligāts* izveides darbībām
details	object	Papildu details objekts interfeisam. Īpašības darbība: - *obligāts*, ja `type` ir iestatīts uz ”SNMP”
disable_until	timestamp	Nākamais nepieejama host interfeisa aptaujas laiks. Īpašības darbība: - *tikai lasāms*
error	string	Kļūdas teksts, ja host interfeiss nav pieejams. Īpašības darbība: - *tikai lasāms*
errors_from	timestamp	Laiks, kad host interfeiss kļuva nepieejams. 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.

Īpašība	Tips	Apraksts
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.get

Apraksts

`integer/array hostinterface.get(object parameters)`

Šī metode ļauj izgūt hostu saskarnes atbilstoši norādītajiem parametriem.

> **Note:**
> Šī metode ir pieejama visu veidu 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
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`.

Parametrs	Tips	Apraksts
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	
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*.

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

Īpašība	Tips	Apraksts
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 jaunus hostu grupas.

Note:
Šī metode ir pieejama tikai *Super admin* lietotāja tipam. Atļaujas izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet User roles.

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:

- tajā ir hosti, kas pieder tikai šai grupai;
- tā ir atzīmēta kā iekšēja;
- to izmanto hosta prototips;
- to izmanto globāls skripts;
- to izmanto korelācijas nosacījums.

Note:
Šī metode ir pieejama tikai *Admin* un *Super admin* lietotāju tipiem. Tiesības izsaukt metodi var atsaukt lietotāja lomu iestatījumos. Plašāku informāciju skatiet sadaļā 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)`

Šī metode ļauj izgūt hostu grupas atbilstoši norādītajiem parametriem.

Note:
Šī metode ir pieejama jebkura tipa lietotājiem. Atļaujas izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā Lietotāja 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.
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`.

Parametrs	Tips	Apraksts
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. Tiesības izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā Lietotāju lomas.

Parametri

`(object)` Parametri, kas satur atjauninámo 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*
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. Tiesības izsaukt šo metodi var atsaukt lietotāju lomu iestatījumos. Plašāku informāciju skatiet sadaļā 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 tagu filtrus visām host grupas apakšgrupām.

> **Note:**
> Šī metode ir pieejama tikai *Super admin* lietotāju tipiem. Tiesības 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|

|--|--|------|

|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. Tiesības izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet 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.

Mājturība

Iestatījumu objektam ir šādi rekvizīti.

Rekvizīts	Tips	Apraksts
hk_events_mode	integer	Iespējot iekšējo notikumu un brīdinājumu mājturību. Iespējamās vērtības: 0 - Atspējot; 1 - *(noklusējums)* Iespējot.
hk_events_trigger	string	Trigeru datu glabāšanas periods. Pieņem sekunžu skaitu (piemēram, 1, 60, 300) vai vērtību ar laika sufiksu (piemēram, 30s, 1m, 2h, 1d). Noklusējums: 365d.
hk_events_service	string	Pakalpojumu datu glabāšanas periods. Pieņem sekunžu skaitu (piemēram, 1, 60, 300) vai vērtību ar laika sufiksu (piemēram, 30s, 1m, 2h, 1d). Noklusējums: 1d.
hk_events_internal	string	Iekšējo datu glabāšanas periods. Pieņem sekunžu skaitu (piemēram, 1, 60, 300) vai vērtību ar laika sufiksu (piemēram, 30s, 1m, 2h, 1d). Noklusējums: 1d.
hk_events_discovery	string	Tīkla atklāšanas datu glabāšanas periods. Pieņem sekunžu skaitu (piemēram, 1, 60, 300) vai vērtību ar laika sufiksu (piemēram, 30s, 1m, 2h, 1d). Noklusējums: 1d.
hk_events_autoreg	string	Automātiskās reģistrācijas datu glabāšanas periods. Pieņem sekunžu skaitu (piemēram, 1, 60, 300) vai vērtību ar laika sufiksu (piemēram, 30s, 1m, 2h, 1d). Noklusējums: 1d.
hk_services_mode	integer	Iespējot iekšējo pakalpojumu mājturību. Iespējamās vērtības: 0 - Atspējot; 1 - *(noklusējums)* Iespējot.

Rekvizīts	Tips	Apraksts
hk_services	string	Pakalpojumu datu glabāšanas periods. Pieņem sekunžu skaitu (piemēram, 1, 60, 300) vai vērtību ar laika sufiksu (piemēram, 30s, 1m, 2h, 1d). Noklusējums: 365d.
hk_audit_mode	integer	Iespējot iekšējo audita mājturību. Iespējamās vērtības: 0 - Atspējot; 1 - *(noklusējums)* Iespējot.
hk_audit	string	Audita datu glabāšanas periods. Pieņem sekunžu skaitu (piemēram, 1, 60, 300) vai vērtību ar laika sufiksu (piemēram, 30s, 1m, 2h, 1d). Noklusējums: 31d.
hk_sessions_mode	integer	Iespējot iekšējo sesiju mājturību. Iespējamās vērtības: 0 - Atspējot; 1 - *(noklusējums)* Iespējot.
hk_sessions	string	Sesiju datu glabāšanas periods. Pieņem sekunžu skaitu (piemēram, 1, 60, 300) vai vērtību ar laika sufiksu (piemēram, 30s, 1m, 2h, 1d). Noklusējums: 31d.
hk_history_mode	integer	Iespējot iekšējo vēstures mājturību. Iespējamās vērtības: 0 - Atspējot; 1 - *(noklusējums)* Iespējot.
hk_history_global	integer	Ignorēt vienuma vēstures periodu. Iespējamās vērtības: 0 - Neignorēt; 1 - *(noklusējums)* Ignorēt.
hk_history	string	Vēstures datu glabāšanas periods. Pieņem sekunžu skaitu (piemēram, 1, 60, 300) vai vērtību ar laika sufiksu (piemēram, 30s, 1m, 2h, 1d). Noklusējums: 31d.
hk_trends_mode	integer	Iespējot iekšējo tendenču mājturību. Iespējamās vērtības: 0 - Atspējot; 1 - *(noklusējums)* Iespējot.
hk_trends_global	integer	Ignorēt vienuma tendenču periodu. Iespējamās vērtības: 0 - Neignorēt; 1 - *(noklusējums)* Ignorēt.
hk_trends	string	Tendenču datu glabāšanas periods. Pieņem sekunžu skaitu (piemēram, 1, 60, 300) vai vērtību ar laika sufiksu (piemēram, 30s, 1m, 2h, 1d). Noklusējums: 365d.
db_extension	string	Konfigurācijas karodziņš DB paplašinājumam. Ja šis karodziņš ir iestatīts uz ”timescaledb”, serveris maina savu darbību attiecībā uz mājturību un vienumu dzēšanu. Rekvizīta darbība: - *tikai lasāms*

Rekvizīts	Tips	Apraksts
compression_availability	integer	Vai datubāze (vai tās paplašinājums) atbalsta datu saspiešanu. Iespējamās vērtības: 0 - Nav pieejama; 1 - Pieejama. Rekvizīta darbība: - *tikai lasāms*
compression_status	integer	Iespējot TimescaleDB vēstures un tendenču saspiešanu. Iespējamās vērtības: 0 - *(noklusējums)* Izslēgts; 1 - Ieslēgts.
compress_older	string	Saspiest vēstures un tendenču ierakstus, kas ir vecāki par norādīto periodu. Pieņem sekunžu skaitu (piemēram, 1, 60, 300) vai vērtību ar laika sufiksu (piemēram, 30s, 1m, 2h, 1d). Noklusējums: 7d.

housekeeping.get

Apraksts

`object housekeeping.get(object parameters)`

Šī metode ļauj izgūt housekeeping objektu atbilstoši norādītajiem parametriem.

> **Note:**
> Šī metode ir pieejama jebkura tipa lietotājiem. Tiesības izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Skatiet User roles, lai iegūtu papildu informāciju.

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 housekeeping iestatījumus.

> **Note:**
> Šī metode ir pieejama tikai lietotāja tipam *Super admin*. Atļaujas izsaukt šo metodi var atsaukt lietotāju lomas iestatījumos. Plašāku informāciju skatiet sadaļā User roles.

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*.

Ierīce

Šī klase ir paredzēta darbam ar mobilajām ierīcēm, lai tās izmantotu kopā ar Zabbix mobilo lietotni.

Objektu atsauces:

- Ierīce

Pieejamās metodes:

- device.get - izgūt ierīces
- device.offboard - noņemt ierīces (izņemt no lietošanas)

Ierīces objekts

Ierīces objekts apzīmē mobilo ierīci, kas paredzēta lietošanai ar Zabbix mobilo lietotni.

Property	Type	Description
deviceid	ID	Ierīces ID (iekšējais, piemēram, 1, 2). Property behavior: - Tikai lasāms
userid	ID	Ierīces lietotāja ID. Property behavior: - Konstants
uuid	string	Ierīces universāli unikālais identifikators (UUIDv7) (ārējais, piemēram, `019f930e-0ea2-7c1f-9dcc-0fbb60c40664`). Property behavior: - Tikai lasāms
name	string	Ierīces nosaukums. Property behavior: - Tikai lasāms
status	integer	Ierīces statuss. Iespējamās vērtības: 0 - Jauna 1 - Aktīva 2 - Nepiešķirta Property behavior: - Tikai lasāms
activated_at	timestamp	Datums un laiks, kad ierīce tika pievienota mobilajā lietotnē, vai 0, ja tā vēl nav pievienota. Property behavior: - Tikai lasāms

Property	Type	Description
lastaccess	timestamp	Datums un laiks, kad ierīce pēdējo reizi tika autentificēta mobilajā lietotnē, vai 0, ja tā vēl nav autentificēta. Property behavior: - Tikai lasāms

device.get

Šī metode ļauj izgūt mobilās ierīces izmantošanai ar Zabbix mobilo lietotni.

Metode ir pieejama *User*, *Admin* un *Super admin* lietotājiem, ja ir iespējots `EnableMobileDevices` un lietotājam ir piekļuve ierīcēm. Piekļuvi API metodēm var atsaukt lietotāja lomas konfigurācijā.

User un *Admin* lietotāji var izgūt tikai savas ierīces.

Metodes paraksts: `integer/array device.get(object parameters)`

Metodes avots: `CDevice::get()` failā `ui/include/classes/api/services/CDevice.php`

Parametri

Parametrs	Tips	Apraksts
deviceids	ID/array	Atgriež tikai ierīces ar norādītajiem ierīču ID.
userids	ID/array	Atgriež tikai ierīces, kas pieder lietotājiem ar norādītajiem lietotāju ID.
roleids	ID/array	Atgriež tikai ierīces, kas pieder lietotājiem ar norādītajiem lietotāju lomu ID.
usrgrpids	ID/array	Atgriež tikai ierīces, kas pieder lietotājiem ar norādītajiem lietotāju grupu ID.
sortfield	string/array	Kārto rezultātu pēc norādītajām īpašībām. Iespējamās vērtības: `deviceid`, `name`.
countOutput	boolean	Šie parametri ir aprakstīti atsauces komentāros.
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Atgrieztās vērtības

Atgriež objektu masīvu vai izgūto objektu skaitu (ja tiek izmantots `countOutput`).

Piemēri

Ierīču izgūšana

Izgūstiet ierīces, kas pieder lietotājiem ar šādiem lietotāju ID: 1, 2, 3.
 Ņemiet vērā, ka lietotājam 2 (viesim) nav ierīču un ierīce 3 vēl nav pievienota vai autentificēta Zabbix mobilajā lietotnē.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "device.get",
    "params": {
        "output": "extend",
        "userids": [1, 2, 3]
    },
    "id": 1
}
```

Atbilde:


```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "deviceid": "1",
            "userid": "1",
            "uuid": "019f930e-0ea2-7c1f-9dcc-0fbb60c40664",
            "name": "Google Pixel 10",
            "status": "1",
            "activated_at": "1784533122",
            "lastaccess": "1784879933"
        },
        {
            "deviceid": "2",
            "userid": "1",
            "uuid": "019d00fb-9e87-7ad5-bb4e-978467254aff",
            "name": "Apple iPhone 15 Pro",
            "status": "1",
            "activated_at": "1784533225",
            "lastaccess": "0"
        },
        {
            "deviceid": "3",
            "userid": "3",
            "uuid": "019d00fb-9e87-7ad5-bb4e-6ed88a2297ee",
            "name": "",
            "status": "0",
            "activated_at": "0",
            "lastaccess": "0"
        }
    ],
    "id": 1
}
```


device.offboard

Šī metode ļauj noņemt (izņemt no lietošanas) mobilās ierīces, kas tiek izmantotas ar Zabbix mobilo lietotni.

Metode ir pieejama *User*, *Admin* un *Super admin* lietotājiem, ja ir iespējots `EnableMobileDevices` un lietotājam ir piekļuve ierīcēm. Piekļuvi API metodēm var atsaukt lietotāja lomas konfigurācijā.

User un *Admin* lietotāji var noņemt tikai savas ierīces.

Metodes paraksts: `object device.offboard(object parameters)`

Metodes avots: `CDevice::offboard()` failā `ui/include/classes/api/services/CDevice.php`

Parametri

Parametrs	Tips	Apraksts
uuid	string	Ierīces universāli unikālais identifikators (UUIDv7). Atbilst `deviceid`.
force	boolean	Iestatiet `true`, lai ignorētu kļūdu "Ierīce nav atrasta" un piespiedu kārtā noņemtu ierīci. Noklusējums: `false`

Atgrieztās vērtības

Atgriež objektu, kas satur noņemtās ierīces UUID.

Piemēri

Ierīces noņemšana

Noņem ierīci ar norādīto UUID, ignorējot kļūdu “Ierīce nav atrasta”.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "device.offboard",
    "params": {
        "uuid": "019d0106-5eef-7154-a159-bb29e35d38e5",
        "force": true
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "uuid": "019d0106-5eef-7154-a159-bb29e35d38e5"
    },
    "id": 1
}
```

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 šādi rekvizīti.

Rekvizīts	Tips	Apraksts
default_lang	string	Sistēmas noklusējuma valoda. Noklusējums: `en_US`.
default_timezone	string	Sistēmas noklusējuma laika josla. Noklusējums: `system` - sistēmas noklusējuma laika josla. Pilnu atbalstīto laika joslu sarakstu skatiet PHP dokumentācijā.
default_theme	string	Noklusējuma tēma. Iespējamās vērtības: `blue-theme` - *(noklusējums)* zila; `blue-classic-theme` - zila (klasiskā); `dark-theme` - tumša; `dark-blue-theme` - tumši zila; `dark-classic-theme` - tumša (klasiskā); `hc-light` - gaiša ar augstu kontrastu; `hc-dark` - tumša ar augstu kontrastu.

Rekvizīts	Tips	Apraksts
search_limit	integer	Meklēšanas un filtrēšanas rezultātu ierobežojums. Noklusējums: 1000.
max_overview_table_size	integer	Maksimālais kolonnu un rindu skaits Data overview un Trigger overview informācijas paneļa logrīkos. Noklusējums: 50.
max_in_table	integer	Maksimālais elementu skaits, kas tiek rādīts tabulas šūnā. Noklusējums: 50.
server_check_interval	integer	Rādīt brīdinājumu, ja Zabbix serveris nedarbojas. 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, kas nav Super administratora lietotāji, kā arī lietotājiem, kuri nav to lietotāju grupu dalībnieki, kurām ir iespējots atkļūdošanas režīms. Iespējamās vērtības: 0 - *(noklusējums)* nerādīt tehniskās kļūdas; 1 - rādīt tehniskās kļūdas.
history_period	string	Maksimālais periods vēsturisko datu rādīšanai Latest data, Web un Data overview informācijas paneļa logrīkos. Pieņem sekunžu skaitu (piemēram, 1, 60, 300) vai vērtību ar laika sufiksu (piemēram, 30s, 1m, 2h, 1d). Noklusējums: 24h.
period_default	string	Laika filtra noklusējuma periods. Pieņem sekunžu skaitu (piemēram, 1, 60, 300) vai vērtību ar laika sufiksu (piemēram, 30s, 1m, 2h, 1d, 1M, 1y), tostarp atbalsta mēnešus un gadus. Noklusējums: 1h.
max_period	string	Maksimālais laika filtra periods. Pieņem sekunžu skaitu (piemēram, 1, 60, 300) vai vērtību ar laika sufiksu (piemēram, 30s, 1m, 2h, 1d, 1M, 1y), tostarp atbalsta mēnešus un gadus. Noklusējums: 2y.
severity_color_0	string	“Not classified” prioritātes krāsa heksadecimāla krāsas koda formātā. Noklusējums: 97AAB3.
severity_color_1	string	“Information” prioritātes krāsa heksadecimāla krāsas koda formātā. Noklusējums: 7499FF.
severity_color_2	string	“Warning” prioritātes krāsa heksadecimāla krāsas koda formātā. Noklusējums: FFC859.
severity_color_3	string	“Average” prioritātes krāsa heksadecimāla krāsas koda formātā. Noklusējums: FFA059.
severity_color_4	string	“High” prioritātes krāsa heksadecimāla krāsas koda formātā. Noklusējums: E97659.
severity_color_5	string	“Disaster” prioritātes krāsa heksadecimāla krāsas koda formātā. Noklusējums: E45959.

Rekvizīts	Tips	Apraksts
severity_name_0	string	“Not classified” prioritātes nosaukums. Noklusējums: Not classified.
severity_name_1	string	“Information” prioritātes nosaukums. Noklusējums: Information.
severity_name_2	string	“Warning” prioritātes nosaukums. Noklusējums: Warning.
severity_name_3	string	“Average” prioritātes nosaukums. Noklusējums: Average.
severity_name_4	string	“High” prioritātes nosaukums. Noklusējums: High.
severity_name_5	string	“Disaster” prioritātes nosaukums. 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	Rādīt OK trigeru periodu. Pieņem sekunžu skaitu (piemēram, 1, 60, 300) vai vērtību ar laika sufiksu (piemēram, 30s, 1m, 2h). Noklusējums: 5m.
blink_period	string	Mirgošanas periods, mainoties trigera stāvoklim. Pieņem sekunžu skaitu (piemēram, 1, 60, 300) vai vērtību ar laika sufiksu (piemēram, 30s, 1m, 2h). Noklusējums: 2m.
problem_unack_color	string	Neapstiprinātu PROBLEM notikumu krāsa heksadecimāla krāsas koda formātā. Noklusējums: CC0000.
problem_ack_color	string	Apstiprinātu PROBLEM notikumu krāsa heksadecimāla krāsas koda formātā. Noklusējums: CC0000.
ok_unack_color	string	Neapstiprinātu RESOLVED notikumu krāsa heksadecimāla krāsas koda formātā. Noklusējums: 009900.
ok_ack_color	string	Apstiprinātu RESOLVED notikumu krāsa heksadecimāla krāsas koda formātā. Noklusējums: 009900.
problem_unack_style	integer	Neapstiprinātu PROBLEM notikumu mirgošana. Iespējamās vērtības: 0 - nerādīt mirgošanu; 1 - *(noklusējums)* rādīt mirgošanu.
problem_ack_style	integer	Apstiprinātu PROBLEM notikumu mirgošana. Iespējamās vērtības: 0 - nerādīt mirgošanu; 1 - *(noklusējums)* rādīt mirgošanu.

Rekvizīts	Tips	Apraksts
ok_unack_style	integer	Neapstiprinātu RESOLVED notikumu mirgošana. Iespējamās vērtības: 0 - nerādīt mirgošanu; 1 - *(noklusējums)* rādīt mirgošanu.
ok_ack_style	integer	Apstiprinātu RESOLVED notikumu mirgošana. Iespējamās vērtības: 0 - nerādīt mirgošanu; 1 - *(noklusējums)* rādīt mirgošanu.
url	string	Lietotāja saskarnes URL.
discovery_groupid	ID	To hostu grupas ID, kurā automātiski tiks ievietoti atklātie hosti.
default_inventory_mode	integer	Noklusējuma hostu inventarizācijas režīms. Iespējamās vērtības: -1 - *(noklusējums)* atspējots; 0 - manuāls; 1 - automātisks.
alert_usrgrpid	ID	Lietotāju grupas ID, kurai tiks nosūtīts paziņojums par datubāzes darbības pārtraukumu. Ja iestatīta vērtība "0", paziņojums netiks nosūtīts.
snmptrap_logging	integer	Reģistrēt žurnālā neatbilstošus SNMP trapus. Iespējamās vērtības: 0 - nereģistrēt žurnālā neatbilstošus SNMP trapus; 1 - *(noklusējums)* reģistrēt žurnālā 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 norādīto vērtību. Pieņem sekunžu skaitu (piemēram, 1, 60) vai vērtību ar laika sufiksu (piemēram, 30s, 1m). 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.
iframe_sandboxing_enabled	integer	Izmantot iframe smilškasti. Iespējamās vērtības: 0 - neizmantot; 1 - *(noklusējums)* izmantot.
iframe_sandboxing_exceptions	string	Iframe smilškastes izņēmumi.

Rekvizīts	Tips	Apraksts
connect_timeout	string	Savienojuma noildze ar Zabbix serveri. Iespējamais vērtību diapazons: 1-30s. Noklusējums: 3s. Rekvizīta darbība: - *obligāts*
socket_timeout	string	Tīkla noklusējuma noildze. Iespējamais vērtību diapazons: 1-300s. Noklusējums: 3s. Rekvizīta darbība: - *obligāts*
media_type_test_timeout	string	Tīkla noildze multivides tipa pārbaudei. Iespējamais vērtību diapazons: 1-300s. Noklusējums: 65s. Rekvizīta darbība: - *obligāts*
item_test_timeout	string	Tīkla noildze vienumu pārbaudēm. Iespējamais vērtību diapazons: 1-600s. Noklusējums: 60s. Rekvizīta darbība: - *obligāts*
script_timeout	string	Tīkla noildze skripta izpildei. Iespējamais vērtību diapazons: 1-300s. Noklusējums: 60s. Rekvizīta darbība: - *obligāts*
report_test_timeout	string	Tīkla noildze ieplānota pārskata pārbaudei. Iespējamais vērtību diapazons: 1-300s. Noklusējums: 60s. Rekvizīta darbība: - *obligāts*
device_link_timeout	string	Noildze mobilās ierīces pievienošanai izmantošanai ar Zabbix mobilo lietotni; attiecas uz saziņu starp Zabbix API un tilta adapteri. Iespējamais vērtību diapazons: 1-300s. Noklusējums: 60s. Rekvizīta darbība: - *obligāts*
auditlog_enabled	integer	Vai iespējot audita reģistrēšanu. Iespējamās vērtības: 0 - atspējot; 1 - *(noklusējums)* iespējot.

Rekvizīts	Tips	Apraksts
auditlog_mode	integer	Vai iespējot servera (System lietotāja) veikto zema līmeņa atklāšanas, tīkla atklāšanas un automātiskās reģistrācijas darbību audita reģistrēšanu. Iespējamās vērtības: 0 - atspējot; 1 - *(noklusējums)* iespējot.
ha_failover_delay	string	Pārslēgšanās aizkave sekundēs. Noklusējums: 1m. Rekvizīta darbība: - *tikai lasāms*
geomaps_tile_provider	string	Ģeokartes elementu 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 elementu URL. Rekvizīta darbība: - *atbalstīts*, ja `geomaps_tile_provider` ir iestatīts uz tukšu virkni
geomaps_max_zoom	integer	Ģeokartes maksimālais tālummaiņas līmenis. Iespējamās vērtības: 0-30. Rekvizīta darbība: - *atbalstīts*, ja `geomaps_tile_provider` ir iestatīts uz tukšu virkni
geomaps_attribution	string	Ģeokartes avota norādes teksts. Rekvizīta darbība: - *atbalstīts*, ja `geomaps_tile_provider` ir iestatīts uz tukšu virkni
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 serveri; 1 - serveri un starpniekserverus neatkarīgi.
timeout_zabbix_agent	string	Apstrādei izmantot ne vairāk kā `timeout_*` sekundes. Pieņem sekunžu skaitu (piemēram, 1, 60, 300) vai vērtību ar laika sufiksu (piemēram, 30s, 1m). Pieņem arī lietotāja makrosus. Iespējamais vērtību diapazons: 1-600s. Noklusējums: 3s. Noklusējums `timeout_browser`: 60s.
timeout_simple_check		Rekvizīta darbība: - *obligāts*

Rekvizīts	Tips	Apraksts
timeout_snmp_agent		
timeout_external_check		
timeout_db_monitor		
timeout_http_agent		
timeout_ssh_agent		
timeout_telnet_agent		
timeout_script		
timeout_browser		
timeout_telemetry_query		
apm_global_db	object	Globālā datu avota konfigurācija.

Globālais datu avots

Globālajam datu avota objektam ir šādi rekvizīti:

Rekvizīts	Tips	Apraksts
status	integer	Globālā datu avota statuss. Iespējamās vērtības: 0 - *(noklusējuma)* nav konfigurēts; 1 - konfigurēts.
url	string	URL, kas tiek izmantots savienojuma izveidei ar datu avotu. Rekvizīta darbība: - *obligāts*, ja lietotājs maina `status` uz 1 (konfigurēts).
authentication_type	integer	Autentifikācijas veids. Iespējamās vērtības: 0 - *(noklusējuma)* tiek izmantots lietotājvārds un parole; 1 - autentifikācija netiek izmantota. Rekvizīta darbība: - *obligāts*, ja lietotājs maina `status` uz 1 (konfigurēts).
username	string	Lietotājvārds, kas tiek izmantots, lai pieteiktos datu avotā. Rekvizīta darbība: - *atbalstīts*, ja `status` ir iestatīts uz 1 (konfigurēts) un `authentication_type` ir iestatīts uz 0 (lietotājvārds un parole). - *obligāts*, ja lietotājs maina `authentication_type` uz 0 (lietotājvārds un parole).
password	string	Parole, kas tiek izmantota, lai pieteiktos datu avotā. Šī parametra vērtība var būt tukša. Rekvizīta darbība: - *atbalstīts*, ja `status` ir iestatīts uz 1 (konfigurēts) un `authentication_type` ir iestatīts uz 0 (lietotājvārds un parole).
db	string	Datu bāzes nosaukums. Šī parametra vērtība var būt tukša. Rekvizīta darbība: - *atbalstīts*, ja `status` ir iestatīts uz 1 (konfigurēts).

Rekvizīts	Tips	Apraksts
ssl_verify_peer	integer	Pārbaudīt servera sertifikātu. Iespējamās vērtības: 0 - *(noklusējuma)* atspējots; 1 - iespējots. Rekvizīta darbība: - *atbalstīts*, ja `status` ir iestatīts uz 1 (konfigurēts) un `url` izmanto shēmu `https://`.
ssl_verify_host	integer	Pārbaudīt hosts. Iespējamās vērtības: 0 - atspējots; 1 - *(noklusējuma)* iespējots. Rekvizīta darbība: - *atbalstīts*, ja `status` ir iestatīts uz 1 (konfigurēts) un `url` izmanto shēmu `https://`.

settings.get

Apraksts

`object settings.get(object parameters)`

Šī metode ļauj izgūt iestatījumu objektu atbilstoši norādītajiem parametriem.

> **Note:**
> Šī metode ir pieejama jebkura tipa lietotājiem. Tiesības izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāja 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": {
        "alert_usrgrpid": "7",
```

```
        "apm_global_db": {
            "status": 1,
            "url": "https://localhost:8443",
            "authentication_type": 0,
            "username": "zabbix",
            "db": "",
            "ssl_verify_peer": 1,
            "ssl_verify_host": 1
        },
        "auditlog_enabled": "1",
        "auditlog_mode": "1",
        "blink_period": "2m",
        "connect_timeout": "3s",
        "custom_color": "0",
        "default_inventory_mode": "-1",
        "default_lang": "en_US",
        "default_theme": "blue-theme",
        "default_timezone": "system",
        "device_link_timeout": "60s",
        "discovery_groupid": "5",
        "geomaps_attribution": "",
        "geomaps_max_zoom": "0",
        "geomaps_tile_provider": "OpenStreetMap.Mapnik",
        "geomaps_tile_url": "",
        "ha_failover_delay": "1m",
        "history_period": "24h",
        "iframe_sandboxing_enabled": "1",
        "iframe_sandboxing_exceptions": "",
        "item_test_timeout": "60s",
        "login_attempts": "5",
        "login_block": "30s",
        "max_in_table": "50",
        "max_overview_table_size": "50",
        "max_period": "2y",
        "media_type_test_timeout": "65s",
        "ok_ack_color": "009900",
        "ok_ack_style": "1",
        "ok_period": "5m",
        "ok_unack_color": "009900",
        "ok_unack_style": "1",
        "period_default": "1h",
        "problem_ack_color": "CC0000",
        "problem_ack_style": "1",
        "problem_unack_color": "CC0000",
        "problem_unack_style": "1",
        "proxy_secrets_provider": "0",
        "report_test_timeout": "60s",
        "script_timeout": "60s",
        "search_limit": "1000",
        "server_check_interval": "10",
        "serverid": "019f4707-1a3e-70e6-aca2-286a2b696358",
        "severity_color_0": "97AAB3",
        "severity_color_1": "7499FF",
        "severity_color_2": "FFC859",
        "severity_color_3": "FFA059",
        "severity_color_4": "E97659",
        "severity_color_5": "E45959",
        "severity_name_0": "Nav klasificēts",
        "severity_name_1": "Informācija",
        "severity_name_2": "Brīdinājums",
        "severity_name_3": "Vidējs",
        "severity_name_4": "Augsts",
```

```
        "severity_name_5": "Katastrofa",
        "show_technical_errors": "0",
        "snmptrap_logging": "1",
        "socket_timeout": "3s",
        "timeout_browser": "60s",
        "timeout_db_monitor": "3s",
        "timeout_external_check": "3s",
        "timeout_http_agent": "3s",
        "timeout_script": "3s",
        "timeout_simple_check": "3s",
        "timeout_snmp_agent": "3s",
        "timeout_ssh_agent": "3s",
        "timeout_telemetry_query": "3s",
        "timeout_telnet_agent": "3s",
        "timeout_zabbix_agent": "3s",
        "uri_valid_schemes": "http,https,ftp,file,mailto,tel,ssh",
        "url": "",
        "validate_uri_schemes": "1",
        "vault_provider": "0",
        "work_period": "1-5,09:00-18:00",
        "x_frame_options": "SAMEORIGIN"
    },
    "id": 1
}
```


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 atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā User roles.

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*

Īpašība	Tips	Apraksts
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 jaunus ikonu kartējumus.

Note:
Šī metode ir pieejama tikai *Super admin* lietotāja tipam. Tiesības izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet User roles.

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 atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā User roles.

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)`

Šī metode ļauj izgūt ikonu kartes atbilstoši norādītajiem parametriem.

Note:
Šī metode ir pieejama jebkura tipa lietotājiem. Tiesības 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
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. Tiesības izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet User roles.

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*.

Informācija par serveri

Šī klase ir paredzēta informācijas iegūšanai par Zabbix serveri.

Pieejamās metodes:

- serverinfo.get - iegūt pēdējās atjaunināšanas laiku no Zabbix servera

serverinfo.get

Šī metode ļauj izgūt pēdējās atjaunināšanas laiku no norādītā Zabbix servera.

Metode ir pieejama lietotājiem ar lomām *User*, *Admin*, *Super admin*, kā arī neautentificētiem lietotājiem.

Metodes paraksts: `object serverinfo.get(object parameters)`

Metodes avots: `CServerInfo::get()` failā `ui/include/classes/api/services/CServerInfo.php`

Parametri

Parameteris	Type	Apraksts
serverid	ID	Zabbix servera ID. Parametra darbība: - Obligāts
output	query	Šis parametrs ir aprakstīts uzziņas komentāros.

Atgrieztās vērtības

Īpašība	Tips	Apraksts
serverid	ID	Zabbix servera ID.
lastaccess	timestamp	Pēdējās atjaunināšanas laiks no Zabbix servera. Augstas pieejamības (HA) klasterim tiek rādīta aktīvā mezgla `lastaccess` vērtība.

Piemēri

Servera informācijas izgūšana

Izgūt informāciju par norādīto Zabbix serveri.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "serverinfo.get",
    "params": {
        "serverid": "019fb1f9-22bd-74ea-bc03-51efb8b76aba",
        "output": "extend"
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "serverid": "019fb1f9-22bd-74ea-bc03-51efb8b76aba",
        "lastaccess": 1785337245
    },
    "id": 1
}
```

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
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.

Īpašība	Tips	Apraksts
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”
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.

Īpašība	Tips	Apraksts
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.
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.

Īpašība	Tips	Apraksts
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”
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”

Īpašība	Tips	Apraksts
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.
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*

Īpašība	Tips	Apraksts
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*
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`).

Īpašība	Tips	Apraksts
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.

Īpašība	Tips	Apraksts
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*
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.

Īpašība	Tips	Apraksts
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.
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.
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 jaunus kartējumus.

> **Note:**
> Šī metode ir pieejama jebkura tipa lietotājiem. Tiesības izsaukt šo metodi var atsaukt 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.

Parametrs	Tips	Apraksts
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 īpašības `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. Tiesības izsaukt šo metodi var atsaukt 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. Tiesības izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā User roles.

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	
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. Tiesības izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā 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. Tiesības izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāju lomas.

Parametri

`(object)` Parametri, kas definē eksportējamos objektus un izmantojamo formātu.

Parameter	Type	Description
format	string	Formāts, kādā dati ir jāeksportē. Iespējamās vērtības: `yaml` - YAML; `xml` - XML; `json` - JSON; `raw` - neapstrādāts PHP masīvs. Parameter behavior: - *required*
prettyprint	boolean	Padara izvadi cilvēkam lasāmāku, pievienojot atkāpes. Iespējamās vērtības: `true` - pievienot atkāpes; `false` - *(default)* nepievienot atkāpes.

Parameter	Type	Description
options	object	Eksportējamie objekti. Objektam `options` ir šādi parametri: `host_groups` - `(array)` eksportējamo host grupu ID; `hosts` - `(array)` eksportējamo hosts ID; `images` - `(array)` eksportējamo attēlu ID; `maps` - `(array)` eksportējamo karšu ID; `mediaTypes` - `(array)` eksportējamo multivides tipu ID; `template_groups` - `(array)` eksportējamo veidņu grupu ID; `templates` - `(array)` eksportējamo veidņu ID; `dashboards` - `(array)` eksportējamo informācijas paneļu ID; `global_regexes` - `(array)` eksportējamo globālo regulāro izteiksmju ID. *Admin* un *User* tipa lietotāji var eksportēt tikai tos objektus, kuriem viņiem ir *read-only* vai *read-write* atļauja, kā arī attēlus, bet ne multivides tipus. Parameter behavior: - *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. Tiesības izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā Lietotāja lomas.

Parametri

`(object)` Parametri, kas satur importējamus datus un noteikumus, kā šie dati jāapstrādā.

Parameter	Type	Description
format	string	Serializētās virknes formāts. Iespējamās vērtības: `yaml` - YAML; `xml` - XML; `json` - JSON. Parametra uzvedība: - *required*
source	string	Serializēta virkne, kas satur konfigurācijas datus. Parametra uzvedība: - *required*
rules	object	Noteikumi, kā importēt jaunus un esošus objektus. Tipa *Admin* lietotāji var importēt tikai tos objektus, kuriem viņiem ir *read-write* atļauja, kā arī kartes. Piemēram, hostu un tā entītijas (vienumus, trigerus, grafikus u. c.) var importēt tikai tad, ja lietotāja lietotāju grupai ir atļauja hostu grupai, kurai piederēs importētais hosts. Attēlus un mediju tipus tipa *Admin* lietotāji nevar importēt. `rules` parametrs ir detalizēti aprakstīts tabulā zemāk. Parametra uzvedība: - *required*
returnMissingObjects	boolean	Noteikumi, vai atbildes objektā jāiekļauj informācija par trūkstošajiem objektiem; noklusējums: `false`.

Attention:
Ja noteikumi nav norādīti, konfigurācija netiks atjaunināta.

`rules` objekts atbalsta šādus parametrus.

Parameter	Type	Description
dashboards	object	Noteikumi, kā importēt paneļus. Atbalstītie parametri: `createMissing` - `(boolean)` ja iestatīts uz `true`, tiks izveidoti jauni paneļi; noklusējums: `false`; `updateExisting` - `(boolean)` ja iestatīts uz `true` un sistēmā ir konkrēts importējams panelis, tas tiks pilnībā atjaunināts. Lapas un sīkrīki tiks pilnībā pārrakstīti ar jaunajiem; noklusējums: `false`.
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 importētajos datos, tiks dzēsti no datubāzes; noklusējums: `false`.

Parameter	Type	Description
global_regexes	object	Noteikumi, kā importēt globālās regulārās izteiksmes. Atbalstītie parametri: `createMissing` - `(boolean)` ja iestatīts uz `true`, tiks izveidotas jaunas globālās regulārās izteiksmes; noklusējums: `false`; `updateExisting` - `(boolean)` ja iestatīts uz `true`, esošās globālās regulārās izteiksmes tiks atjauninātas, to apakšizteiksmes pilnībā pārrakstot; 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 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`.
hosts	object	Noteikumi, kā importēt hostus. Atbalstītie parametri: `createMissing` - `(boolean)` ja iestatīts uz `true`, tiks izveidoti jauni hosts; noklusējums: `false`; `updateExisting` - `(boolean)` ja iestatīts uz `true`, esošie hosts 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 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`.
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 importētajos datos, tiks dzēsti no datubāzes; noklusējums: `false`.

Parameter	Type	Description
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`.
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 importētajos datos, tiks dzēsti no datubāzes; 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`.
templateLinkage	object	Noteikumi, kā importēt veidņu saites. Atbalstītie parametri: `createMissing` - `(boolean)` ja iestatīts uz `true`, veidnes, kas nav saistītas ar importējamo hostu vai veidni, bet ir importētajos datos, tiks saistītas; noklusējums: `false`; `deleteMissing` - `(boolean)` ja iestatīts uz `true`, veidnes, kas ir saistītas ar importējamo hostu vai veidni, bet nav importētajos datos, tiks atsaistītas, nenoņemot no atsaistītajām veidnēm mantotās entītijas (vienumus, trigerus u. c.); 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`.
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 importētajos datos, tiks dzēsti no datubāzes; noklusējums: `false`.

Parameter	Type	Description
valueMaps	object	Noteikumi, kā importēt hostu vai veidņu 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 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. Tiesības izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā User roles.

Parametri

`(object)` Parametri, kas satur iespējamos importējamos datus un noteikumus, kā šie dati jāapstrādā.

Parameter	Type	Description
format	string	Serializētās virknes formāts. Iespējamās vērtības: `yaml` - YAML; `xml` - XML; `json` - JSON. Parametra uzvedība: - *required*
source	string	Serializēta virkne, kas satur konfigurācijas datus. Parametra uzvedība: - *required*
rules	object	Noteikumi, kā salīdzināt jaunus un esošus objektus. *Admin* un *User* tipa lietotāji var salīdzināt jaunus objektus tikai ar esošajiem, kuriem tiem ir *read-only* vai *read-write* atļauja. Parametrs `rules` ir detalizēti aprakstīts tabulā zemāk. Parametra uzvedība: - *required*
returnMissingObjects	boolean	Noteikumi, vai atbildes objektam jāietver informācija par trūkstošajiem objektiem; noklusējums: `false`.

Attention:
Ja netiek norādīti nekādi noteikumi, nebūs ko atjaunināt, un rezultāts būs tukšs.

Note:
Salīdzināšana tiks veikta tikai host grupām un veidnēm. Trigeri un grafiki tiks salīdzināti tikai importētajām veidnēm, visi pārējie tiks uzskatīti par ”jauniem”.

Objekts `rules` atbalsta šādus parametrus.

Parameter	Type	Description
dashboards	object	Noteikumi, kā importēt paneļus. Atbalstītie parametri: `createMissing` - `(boolean)` ja iestatīts uz `true`, tiks izveidoti jauni paneļi; noklusējums: `false`; `updateExisting` - `(boolean)` ja iestatīts uz `true` un konkrēts importējams panelis ir sistēmā, tas tiks pilnībā atjaunināts. Lapas un sīkrīki tiks pilnībā pārrakstīti ar jaunajiem; noklusējums: `false`.
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`.
global_regexes	object	Noteikumi, kā importēt globālās regulārās izteiksmes. Atbalstītie parametri: `createMissing` - `(boolean)` ja iestatīts uz `true`, tiks izveidotas jaunas globālās regulārās izteiksmes; noklusējums: `false`; `updateExisting` - `(boolean)` ja iestatīts uz `true`, esošās globālās regulārās izteiksmes tiks atjauninātas, to apakšizteiksmes pilnībā pārrakstot; 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 host grupas. Atbalstītie parametri: `createMissing` - `(boolean)` ja iestatīts uz `true`, tiks izveidotas jaunas host grupas; noklusējums: `false`; `updateExisting` - `(boolean)` ja iestatīts uz `true`, esošās host grupas tiks atjauninātas; 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`.

Parameter	Type	Description
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`. Šim parametram nebūs nekādas ietekmes uz izvadi. Tas ir atļauts tikai saderībai 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`. Šim parametram nebūs nekādas ietekmes uz izvadi. Tas ir atļauts tikai saderībai ar `configuration.import`.
mediaTypes	object	Noteikumi, kā importēt multivides tipus. Atbalstītie parametri: `createMissing` - `(boolean)` ja iestatīts uz `true`, tiks izveidoti jauni multivides tipi; noklusējums: `false`; `updateExisting` - `(boolean)` ja iestatīts uz `true`, esošie multivides tipi tiks atjaunināti; noklusējums: `false`. Šim parametram nebūs nekādas ietekmes uz izvadi. Tas ir atļauts tikai saderībai ar `configuration.import`.
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`.
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`.

Parameter	Type	Description
templateLinkage	object	Noteikumi, kā importēt veidņu saites. Atbalstītie parametri: `createMissing` - `(boolean)` ja iestatīts uz `true`, veidnes, kas nav saistītas ar importējamo hostu vai veidni, bet ir iekļautas importētajos datos, tiks saistītas; noklusējums: `false`; `deleteMissing` - `(boolean)` ja iestatīts uz `true`, veidnes, kas ir saistītas ar importējamo hostu vai veidni, bet nav iekļautas importētajos datos, tiks atsaistītas, nenoņemot no atsaistītajām veidnēm mantotās entītijas (vienumus, trigerus utt.); 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`.
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 host 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.

Īpašība	Tips	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, kas jāizpilda, lai veiktu konfigurētās korelācijas darbības. Tam ir šādas īpašības.

Property	Type	Description
conditions	array	Filtru rezultātu filtrēšanai izmantojamo filtra nosacījumu kopa. Nosacījumi tiks sakārtoti tādā secībā, kādā tie ir izvietoti formulā. Īpašības uzvedība: - *required*
evaltype	integer	Filtra nosacījumu izvērtēšanas metode. Iespējamās vērtības: 0 - And/Or; 1 - And; 2 - Or; 3 - Custom expression. Īpašības uzvedība: - *required*
eval_formula	string	Ģenerēta izteiksme, kas tiks izmantota filtra nosacījumu izvērtēšanai. Izteiksme satur ID, kas atsaucas uz konkrētiem filtra nosacījumiem pēc to `formulaid`. Vērtība `eval_formula` ir vienāda ar `formula` vērtību filtriem ar pielāgotu izteiksmi. Īpašības uzvedība: - *read-only*

Property	Type	Description
formula	string	Lietotāja definēta izteiksme, kas jāizmanto filtru ar pielāgotu izteiksmi nosacījumu izvērtēšanai. Izteiksmē jāietver 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 uzvedība: - *required* if `evaltype` is set to "custom expression"

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.

Īpašība	Tips	Apraksts
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 jaunus korelācijas ierakstus.

Note:
Šī metode ir pieejama tikai *Super admin* lietotāja tipam. Tiesības izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet User roles.

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 lietotāja tipam *Super admin*. Atļaujas izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā User roles.

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)`

Šī metode ļauj izgūt korelācijas atbilstoši norādītajiem parametriem.

Note:
Šī metode ir pieejama jebkura tipa lietotājiem. Tiesības izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā Lietotāja 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. Tiesības izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet User roles.

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.

Īpašība	Tips	Apraksts
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. Lietotājam nav atļauts izveidot automātisku makrosu. Lai atjauninātu automātisku makrosu, tas ir jāpārveido par manuālu.
config	object/array	Makrosa konfigurācija, kas nosaka, kā makross tiek attēlots Host Wizard.
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"
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 host makro.

Note:
Šī metode ir pieejama tikai *Admin* un *Super admin* lietotāju tipiem. Tiesības izsaukt šo metodi var atsaukt 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

Hosts makro izveide

Izveidojiet hosts 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 makro.

> **Note:**
> Šī metode ir pieejama tikai *Super admin* lietotāja tipam. Atļaujas izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā User roles.

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

Globāla makro izveide

Izveidojiet globālu makro {$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 host makro.

> **Note:**
> Šī metode ir pieejama tikai *Admin* un *Super admin* lietotāju tipiem. Tiesības izsaukt šo metodi var atsaukt 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 makro.

> **Note:**
> Šī metode ir pieejama tikai *Super admin* lietotāja tipam. Tiesības izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā User roles.

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)`

Šī metode ļauj izgūt host un globālos makro atbilstoši norādītajiem parametriem.

> **Note:**
> Šī metode ir pieejama jebkura tipa lietotājiem. Tiesības izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā Lietotāja 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.
selectHosts	query	Atgriezt hostus, kuriem hosts makross pieder, īpašībā `hosts`. Izmanto tikai, izgūstot hosts makrosus.
selectTemplateGroups	query	Atgriezt veidņu grupas, kurām veidnes makross pieder, īpašībā `templategroups`. Izmanto tikai, izgūstot veidņu makrosus.
selectTemplates	query	Atgriezt veidnes, kurām hosts makross pieder, īpašībā `templates`. Izmanto tikai, izgūstot hosts makrosus.
sortfield	string/array	Kārtot rezultātu pēc norādītajām īpašībām. Iespējamās vērtības: `macro`.
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

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 host makro.

> **Note:**
> Šī metode ir pieejama tikai *Admin* un *Super admin* lietotāju tipiem. Tiesības izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā 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 makro.

> **Note:**
> Šī metode ir pieejama tikai *Super admin* lietotāja tipam. Tiesības izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā User roles.

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 šādi rekvizīti.

Property	Type	Description
userid	ID	Lietotāja ID. Rekvizīta darbība: - *tikai lasāms* - *obligāts* atjaunināšanas darbībām
username	string	Lietotāja vārds. Rekvizīta darbība: - *obligāts* izveides darbībām - *tikai lasāms* nodrošinātajiem lietotājiem, ja lietotājs ir saistīts ar lietotāju direktoriju (`userdirectoryid` ir iestatīts uz derīgu vērtību, kas nav ”0”), lietotāju direktorijas nodrošināšanas statuss ir iespējots (`provision_status` User directory object ir iestatīts uz ”1”) un visas LDAP vai SAML nodrošināšanas autentifikācijas statuss ir iespējots (`ldap_jit_status` Authentication object ir iestatīts uz ”Enabled for configured LDAP IdPs” vai `saml_jit_status` Authentication object 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. Rekvizīta 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 kartēm.
attempt_clock	timestamp	Pēdējā neveiksmīgā pieteikšanās mēģinājuma laiks. Rekvizīta darbība: - *tikai lasāms*

Property	Type	Description
attempt_failed	integer	Nesen veikto neveiksmīgo pieteikšanās mēģinājumu skaits. Rekvizīta darbība: - *tikai lasāms*
attempt_ip	string	IP adrese, no kuras tika veikts pēdējais neveiksmīgais pieteikšanās mēģinājums. Rekvizīta darbība: - *tikai lasāms*
autologin	integer	Vai iespējot automātisko pieteikšanos. Iespējamās vērtības: 0 - *(noklusējums)* automātiskā pieteikšanās atspējota; 1 - automātiskā pieteikšanās iespējota.
autologout	string	Lietotāja sesijas ilgums. Pieņem sekunžu skaitu (piemēram, 1, 60, 300) vai vērtību ar laika sufiksu (piemēram, 30s, 1m, 2h). Ja iestatīts uz 0s, sesija nekad nebeigsies. Noklusējums: 15m.
default_maintenance_period	string	Noklusējuma uzturēšanas periods uzturēšanai, kas izveidota, izmantojot saīsni. Pieņem sekundes vai laika vienību ar sufiksu (300s, 5m, 1h, 1d, 1w). Norādītā vērtība tiks noapaļota uz leju līdz pilnām minūtēm. Noklusējums: 1h.
lang	string	Lietotāja valodas kods, piemēram, en_US. Noklusējums: default - sistēmas noklusējuma valoda.
name	string	Lietotāja vārds.
provisioned	integer	Vai lietotājs ir nodrošināts. Iespējamās vērtības: 0 - nav nodrošināts; 1 - nodrošināts. Rekvizīta darbība: - *tikai lasāms*
refresh	string	Automātiskās atsvaidzināšanas periods. Pieņem sekunžu skaitu (piemēram, 1, 60, 300) vai vērtību ar laika sufiksu (piemēram, 30s, 1m, 1h). Noklusējums: 30s.
rows_per_page	integer	Vienumu rindu skaits, kas jāparāda vienā lapā. Noklusējums: 50.
surname	string	Lietotāja uzvārds.
theme	string	Lietotāja tēma. Iespējamās vērtības: default - *(noklusējums)* sistēmas noklusējuma tēma; blue-theme - zila; blue-classic-theme - zila (klasiskā); dark-theme - tumša; dark-blue-theme - tumši zila; dark-classic-theme - tumša (klasiskā); hc-light - gaiša ar augstu kontrastu; hc-dark - tumša ar augstu kontrastu.
ts_provisioned	timestamp	Laiks, kad tika veikta pēdējā nodrošināšanas darbība. Rekvizīta darbība: - *tikai lasāms* - *atbalstīts Super admin* tipa lietotājiem

Property	Type	Description
url	string	Tās lapas URL, uz kuru pēc pieteikšanās novirzīt lietotāju.
userdirectoryid	ID	Lietotāju direktorijas ID, ar kuru lietotājs ir saistīts. Tiek izmantots nodrošināšanai (izveidei vai atjaunināšanai), kā arī tāda lietotāja pieteikšanai, kurš ir saistīts ar lietotāju direktoriju. Pieteikšanās darbībām šī rekvizīta vērtībai būs prioritāte pār lietotāju grupu `userdirectoryid` rekvizīta vērtību, ja lietotājs pieder šai grupai. Noklusējums: 0. Rekvizīta darbība: - *tikai lasāms* - *atbalstīts Super admin* tipa lietotājiem
timezone	string	Lietotāja laika josla, piemēram, `Europe/London`, `UTC`. Noklusējums: `default` - sistēmas noklusējuma laika josla. Pilnu atbalstīto laika joslu sarakstu skatiet PHP dokumentācijā.

Multivide

Multivides objektam ir šādas īpašības.

Property	Type	Description
mediaid	ID	Lietotāja multivides ID. Īpašības darbība: - *tikai lasāms*
mediatypeid	ID	Lietotāja multivides izmantotā multivides tipa ID. Īpašības darbība: - *obligāts*
sendto	string/array	Saņēmēja adrese, lietotājvārds vai cits identifikators. Ja Multivides tipa `type` ir iestatīts uz ”Email” vai ”Push”, vērtības tiek attēlotas kā masīvs. ”Push” gadījumā masīvā var būt aizstājējzīme (*) vai viens vai vairāki mobilo ierīču ID. Ja Multivides tipa `type` ir iestatīts uz ”Script”, ”SMS” vai ”Webhook”, vērtība tiek attēlota kā virkne. Īpašības darbība: - *obligāts*
active	integer	Norāda, vai multivide ir iespējota. Iespējamās vērtības: 0 - *(noklusējums)* iespējota; 1 - atspējota.

Property	Type	Description
severity	integer	Trigeru nopietnības līmeņi, par kuriem jānosūta paziņojumi. Iespējamās bitkartes vērtības: 1 - Nav klasificēts; 2 - Informācija; 4 - Brīdinājums; 8 - Vidējs; 16 - Augsts; 32 - Katastrofa. Šis ir bitmaskas lauks; ir pieņemama jebkura iespējamo bitkartes vērtību summa (piemēram, 48 — Vidējs, Augsts un Katastrofa). Noklusējums: 63.
period	string	Laiks, kad paziņojumus var nosūtīt, norādīts kā laika periods vai ar semikolu atdalīti lietotāja makrosi. Noklusējums: 1-7,00:00-24:00.
provisioned	integer	Norāda, vai lietotājs ir nodrošināts. Iespējamās vērtības: 0 - nav nodrošināts; 1 - nodrošināts.
userdirectory_mediaid	ID	Lietotāju direktorija multivides kartējuma ID nodrošinātai multividei. 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.

Parameter	Type	Description
extend	boolean	Vai pagarināt lietotāja sesiju. Noklusējuma vērtība: ”true”. Ja vērtība ir iestatīta uz ”false”, tas ļauj pārbaudīt lietotāja sesiju, to nepagarinot.
sessionid	string	Parameter behavior: - *supported* if `sessionid` is set Lietotāja autentifikācijas marķieris. Parameter behavior: - *required* if `token` is not set

Parameter	Type	Description
token	string	Lietotāja API marķieris. Parameter behavior: - *required* if `sessionid` is not set

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 lietotāja tipam *Super admin*. Atļaujas izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā Lietotāju lomas.

Note:
Lietotāja paroles stiprums tiek pārbaudīts atbilstoši paroles politikas noteikumiem, ko definē Authentication API. Plašāku informāciju skatiet sadaļā authentication objekts.

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 to lietotāju grupai un izveidojiet tam jaunu multividi.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "user.create",
    "params": {
        "username": "John",
        "passwd": "Doe12345",
        "roleid": "5",
        "usrgrps": [
            {
```

```
                "usrgrpid": "7"
            }
        ],
        "medias": [
            {
                "mediatypeid": "1",
                "sendto": [
                    "support@company.com"
                ],
                "active": 0,
                "severity": 63,
                "period": "1-7,00:00-24:00"
            }
        ]
    },
    "id": 1
}
```

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. Tiesības izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā User roles.

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)`

Šī metode ļauj izgūt lietotājus atbilstoši norādītajiem parametriem.

> **Note:**
> Šī metode ir pieejama jebkura tipa lietotājiem. Tiesības izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Skatiet Lietotāja lomas, lai iegūtu papildu informāciju.

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

Izgūt 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",
            "default_maintenance_period": "1h",
            "refresh": "30s",
            "theme": "default",
            "attempt_failed": "0",
            "attempt_ip": "",
            "attempt_clock": "0",
            "rows_per_page": "50",
            "timezone": "default",
            "roleid": "3",
            "userdirectoryid": "0",
            "ts_provisioned": "0",
            "provisioned": "0"
        },
        {
            "userid": "2",
            "username": "guest",
            "name": "",
            "surname": "",
            "url": "",
            "autologin": "0",
            "autologout": "15m",
            "lang": "default",
```

```
            "default_maintenance_period": "1h",
            "refresh": "30s",
            "theme": "default",
            "attempt_failed": "0",
            "attempt_ip": "",
            "attempt_clock": "0",
            "rows_per_page": "50",
            "timezone": "default",
            "roleid": "4",
            "userdirectoryid": "0",
            "ts_provisioned": "0",
            "provisioned": "0"
        }
    ],
    "id": 1
}
```

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",
      "url": "",
      "autologin": "1",
      "autologout": "0",
      "lang": "default",
      "default_maintenance_period": "1h",
      "refresh": "30s",
      "theme": "default",
      "attempt_failed": "0",
      "attempt_ip": "",
      "attempt_clock": "0",
      "rows_per_page": "50",
      "timezone": "default",
      "roleid": "3",
      "userdirectoryid": "0",
      "ts_provisioned": "0",
      "provisioned": "0",
      "gui_access": "1",
      "debug_mode": "0",
```

```
    "users_status": "0",
    "usrgrps": [
      {
        "usrgrpid": "7",
        "name": "Zabbix administrators",
        "gui_access": "0",
        "users_status": "0",
        "debug_mode": "0",
        "userdirectoryid": "0",
        "mfa_status": "0",
        "mfaid": "0",
        "proxy_mode": "0",
        "proxy_group_mode": "0"
      },
      {
        "usrgrpid": "13",
        "name": "Internal",
        "gui_access": "1",
        "users_status": "0",
        "debug_mode": "0",
        "userdirectoryid": "0",
        "mfa_status": "0",
        "mfaid": "0",
        "proxy_mode": "0",
        "proxy_group_mode": "0"
      }
    ],
    "medias": [],
    "mediatypes": [],
    "role": {
      "roleid": "3",
      "name": "Super admin role",
      "type": "3",
      "readonly": "1"
    }
  },
  {
    "userid": "2",
    "username": "guest",
    "name": "",
    "surname": "",
    "url": "",
    "autologin": "0",
    "autologout": "15m",
    "lang": "default",
    "default_maintenance_period": "1h",
    "refresh": "30s",
    "theme": "default",
    "attempt_failed": "0",
    "attempt_ip": "",
    "attempt_clock": "0",
    "rows_per_page": "50",
    "timezone": "default",
    "roleid": "4",
    "userdirectoryid": "0",
    "ts_provisioned": "0",
    "provisioned": "0",
    "gui_access": "1",
    "debug_mode": "0",
    "users_status": "1",
    "usrgrps": [
      {
```

```
          "usrgrpid": "8",
          "name": "Guests",
          "gui_access": "0",
          "users_status": "0",
          "debug_mode": "0",
          "userdirectoryid": "0",
          "mfa_status": "0",
          "mfaid": "0",
          "proxy_mode": "0",
          "proxy_group_mode": "0"
        },
        {
          "usrgrpid": "9",
          "name": "Disabled",
          "gui_access": "0",
          "users_status": "1",
          "debug_mode": "0",
          "userdirectoryid": "0",
          "mfa_status": "0",
          "mfaid": "0",
          "proxy_mode": "0",
          "proxy_group_mode": "0"
        },
        {
          "usrgrpid": "13",
          "name": "Internal",
          "gui_access": "1",
          "users_status": "0",
          "debug_mode": "0",
          "userdirectoryid": "0",
          "mfa_status": "0",
          "mfaid": "0",
          "proxy_mode": "0",
          "proxy_group_mode": "0"
        }
      ],
      "medias": [],
      "mediatypes": [],
      "role": {
        "roleid": "4",
        "name": "Guest role",
        "type": "1",
        "readonly": "0"
      }
    }
  ],
  "id": 1
}
```


Lietotāja datu izgūšana

Izgūt lietotāja ar ID ”2” datus.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "user.get",
    "params": {
        "output": ["userid", "username"],
        "selectRole": "extend",
        "userids": "2"
    },
    "id": 1
```

```
}
```

Atbilde:

```
{
  "jsonrpc": "2.0",
  "result": [
    {
      "userid": "2",
      "username": "guest",
      "role": {
        "roleid": "4",
        "name": "Guest role",
        "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 jums pieteikties API un ģenerēt autentifikācijas žetonu.

Warning:
Izmantojot šo metodi, jums arī jāizpilda user.logout, lai novērstu liela skaita atvērtu sesiju ierakstu izveidi.

Attention:
Šī metode ir pieejama tikai neautentificētiem lietotājiem, kuri nepieder nevienai lietotāju grupai ar iespējotu daudzfaktoru autentifikāciju.

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 jums atteikties no API un padara pašreizējo autentifikācijas marķieri nederīgu.

Note:
Šī metode ir pieejama jebkura tipa lietotājiem. Tiesības izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāja 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 metodi var atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāja 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 lietotāju TOTP noslēpumus.

Note:
Šī metode ir pieejama tikai *Super admin* lietotāja tipam. Tiesības izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā 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 atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā User roles.

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. Tiesības izsaukt metodi var atsaukt lietotāja lomu iestatījumos. Plašāku informāciju skatiet Lietotāja lomas.

> **Note:**
> Lietotāja paroles stiprums tiek validēts saskaņā ar paroles politikas noteikumiem, kas definēti 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
usrgrps	array	Lietotāju grupas, ar kurām aizstāt esošās lietotāju grupas. Lietotāju grupām jābūt definētam tikai rekvizītam `usrgrpid`.
medias	array	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 tālāk norādītie rekvizīti.

Rekvizīts	Tips	Apraksts
userdirectoryid	ID	Lietotāju direktorija ID. Ja lietotāju direktorijs tiek dzēsts, Lietotāja objekta rekvizīta `userdirectoryid` vērtība tiek iestatīta uz ”0” visiem lietotājiem, kas ir saistīti ar dzēsto lietotāju direktoriju. Rekvizīta darbība: - *tikai lasāms* - *obligāts* atjaunināšanas darbībām
idp_type	integer	Identitātes nodrošinātāja lietotāju direktorijam izmantotā autentifikācijas protokola tips. Ņemiet vērā, ka var pastāvēt tikai viens SAML tipa lietotāju direktorijs. Iespējamās vērtības: 1 - LDAP tipa lietotāju direktorijs; 2 - SAML tipa lietotāju direktorijs. Rekvizīta darbība: - *obligāts* izveides darbībām
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* Rekvizīta darbība: - *obligāts*, ja `provision_status` ir iestatīts uz ”Enabled” un Autentifikācijas objekta `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. Lietotāja nodrošināšanas laikā tas tiek izmantots kā Lietotāja objekta rekvizīta `name` vērtība. 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. Lietotāja nodrošināšanas laikā tas tiek izmantots kā Lietotāja objekta rekvizīta `surname` vērtība. Piemēri: *sn*, *surname*, *lastName*
provision_status	integer	Lietotāju direktorija nodrošināšanas statuss. Iespējamās vērtības: 0 - *(noklusējums)* Disabled (lietotāju, kas izveidoti, izmantojot šo lietotāju direktoriju, nodrošināšana ir atspējota); 1 - Enabled (lietotāju, kas izveidoti, izmantojot šo lietotāju direktoriju, nodrošināšana ir iespējota; papildus jābūt iespējotam LDAP vai SAML nodrošināšanas statusam (`ldap_jit_status` vai Autentifikācijas objekta `saml_jit_status`)).
provision_groups	array	Nodrošināšanas grupu kartējumu objektu masīvs LDAP/SAML lietotāju grupu modeļa kartēšanai uz Zabbix lietotāju grupu un lietotāja lomu. Rekvizīta darbība: - *obligāts*, ja `provision_status` ir iestatīts uz ”Enabled”
provision_media	array	Multivides tipu kartējumu objektu masīvs lietotāja LDAP/SAML multivides atribūtu (piemēram, e-pasta) kartēšanai uz Zabbix lietotāja multividi paziņojumu nosūtīšanai.

Rekvizīts	Tips	Apraksts
LDAP specifiskie rekvizīti:		
name	string	Unikāls lietotāju direktorija nosaukums. Rekvizīta darbība: - *obligāts*, ja `idp_type` ir iestatīts uz "User directory of type LDAP"
host	string	LDAP servera resursdatora nosaukums, IP adrese vai URI. URI jāietver shēma (`ldap://` vai `ldaps://`), hosts un ports (neobligāts). Piemēri: *host.example.com* *127.0.0.1* *ldap://ldap.example.com:389* Rekvizīta darbība: - *obligāts*, ja `idp_type` ir iestatīts uz "User directory of type LDAP"
port	integer	LDAP servera ports. Rekvizīta darbība: - *obligāts*, ja `idp_type` ir iestatīts uz "User directory of type LDAP"
base_dn	string	LDAP lietotāju direktorija bāzes ceļš uz lietotāju kontiem. Piemēri: *ou=Users,dc=example,dc=org* *ou=Users,ou=system* (OpenLDAP) *DC=company,DC=com* (Microsoft Active Directory) *uid=%{user},dc=example,dc=com* (tiešai lietotāja saistīšanai; vietturis "*%{user}*" ir obligāts) Rekvizīta darbība: - *obligāts*, ja `idp_type` ir iestatīts uz "User directory of type LDAP"
search_attribute	string	LDAP lietotāju direktorija atribūts, pēc kura identificē lietotāja kontu, izmantojot pieteikšanās pieprasījumā sniegto informāciju. Piemēri: *uid* (OpenLDAP) *sAMAccountName* (Microsoft Active Directory) Rekvizīta darbība: - *obligāts*, ja `idp_type` ir iestatīts uz "User directory of type LDAP"
bind_dn	string	LDAP servera konts saistīšanai un meklēšanai LDAP serverī. Tiešai lietotāja saistīšanai un anonīmai saistīšanai `bind_dn` jābūt tukšam. Piemēri: *uid=ldap_search,ou=system* (OpenLDAP) *CN=ldap_search,OU=user_group,DC=company,DC=com* (Microsoft Active Directory) *CN=Admin,OU=Users,OU=Zabbix,DC=zbx,DC=local* Rekvizīta darbība: - *atbalstīts*, ja `idp_type` ir iestatīts uz "User directory of type LDAP"
bind_password	string	LDAP serverī saistīšanai un meklēšanai izmantotā konta LDAP parole. Tiešai lietotāja saistīšanai un anonīmai saistīšanai `bind_password` jābūt tukšam. Rekvizīta darbība: - *atbalstīts*, ja `idp_type` ir iestatīts uz "User directory of type LDAP"

Rekvizīts	Tips	Apraksts
description	string	Lietotāju direktorija apraksts.
group_basedn	string	Rekvizīta darbība: - *atbalstīts*, ja `idp_type` ir iestatīts uz "User directory of type LDAP" LDAP lietotāju direktorija bāzes ceļš uz grupām; tiek izmantots, lai konfigurētu lietotāja dalības pārbaudi LDAP lietotāju direktorijā. Tiek ignorēts lietotāja nodrošināšanas laikā, ja ir iestatīts `group_membership`. Piemērs: *ou=Groups,dc=example,dc=com*
group_filter	string	Rekvizīta darbība: - *atbalstīts*, ja `idp_type` ir iestatīts uz "User directory of type LDAP" Filtra virkne LDAP lietotāju direktorija grupu izgūšanai, kurās lietotājs ir dalībnieks; tiek izmantota, lai konfigurētu lietotāja dalības pārbaudi LDAP lietotāju direktorijā. Tiek ignorēta lietotāja nodrošināšanas laikā, ja ir iestatīts `group_membership`. Atbalstītie `group_filter` vietturi: *%{attr}* - meklēšanas atribūts (aizstāts ar rekvizīta `search_attribute` vērtību); *%{groupattr}* - grupas atribūts (aizstāts ar rekvizīta `group_member` vērtību); *%{host}* - LDAP servera resursdatora nosaukums, IP adrese vai URI (aizstāts ar rekvizīta `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, kurā "User1" ir dalībnieks; - *(%{groupattr}=cn=%{ref},ou=Users,ou=Zabbix,DC=example,DC=com)* atbildīs "User1", ja LDAP grupas objekts satur rekvizītā `group_member` norādīto atribūtu ar vērtību "*cn=User1,ou=Users,ou=Zabbix,DC=example,DC=com*", un atgriezīs grupu, kurā "User1" ir dalībnieks.
group_member	string	Rekvizīta 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 lietotāja nodrošināšanas laikā, ja ir iestatīts `group_membership`.
group_membership	string	Rekvizīta 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* Rekvizīta darbība: - *atbalstīts*, ja `idp_type` ir iestatīts uz "User directory of type LDAP"

Rekvizīts	Tips	Apraksts
search_filter	string	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})* Rekvizīta darbība: - *atbalstīts*, ja `idp_type` ir iestatīts uz "User directory of type LDAP"
start_tls	integer	LDAP servera konfigurācijas opcija, kas ļauj aizsargāt saziņu ar LDAP serveri, izmantojot Transport Layer Security (TLS). Ņemiet vērā, ka `start_tls` jāiestata uz "Disabled" hostiem, kas izmanto `ldaps://` protokolu. Iespējamās vērtības: 0 - *(noklusējums)* Disabled; 1 - Enabled. Rekvizīta darbība: - *atbalstīts*, ja `idp_type` ir iestatīts uz "User directory of type LDAP"
user_ref_attr	string	LDAP lietotāju direktorija atribūts, ko izmanto, lai atsauktos uz lietotāja objektu. `user_ref_attr` vērtība tiek izmantota, lai iegūtu vērtības no norādītā lietotāju direktorija atribūta un ievietotu tās *%{ref}* vietturā vietā `group_filter` virknē. Piemēri: *cn*, *uid*, *member*, *uniqueMember* Rekvizīta darbība: - *atbalstīts*, ja `idp_type` ir iestatīts uz "User directory of type LDAP"
SAML specifiskie rekvizīti:		
idp_entityid	string	URI, kas identificē identitātes nodrošinātāju un tiek izmantots saziņai ar identitātes nodrošinātāju SAML ziņojumos. Piemērs: *https://idp.example.com/idp* Rekvizīta darbība: - *obligāts*, ja `idp_type` ir iestatīts uz "User directory of type SAML"
sp_entityid	string	URL vai jebkura virkne, kas identificē identitātes nodrošinātāja pakalpojumu sniedzēju. Piemēri: *https://idp.example.com/sp* *zabbix* Rekvizīta darbība: - *obligāts*, ja `idp_type` ir iestatīts uz "User directory of type SAML"

Rekvizīts	Tips	Apraksts
username_attribute	string	SAML lietotāju direktorija atribūts (arī SCIM atribūts, ja `scim_status` ir iestatīts uz "SCIM provisioning is enabled"), kas satur lietotāja lietotājvārdu. Autentificēšanas laikā tas tiek salīdzināts ar Lietotāja objekta rekvizīta `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*
sso_url	string	Rekvizīta darbība: - *obligāts*, ja `idp_type` ir iestatīts uz "User directory of type SAML" Identitātes nodrošinātāja SAML vienotās pieteikšanās pakalpojuma URL, uz kuru Zabbix nosūtīs SAML autentifikācijas pieprasījumus. Piemērs: *http://idp.example.com/idp/sso/saml*
slo_url	string	Rekvizīta darbība: - *obligāts*, ja `idp_type` ir iestatīts uz "User directory of type SAML" Identitātes nodrošinātāja SAML vienotās atteikšanās pakalpojuma URL, uz kuru Zabbix nosūtīs SAML atteikšanās pieprasījumus. Piemērs: *https://idp.example.com/idp/slo/saml*
encrypt_nameid	integer	Rekvizīta darbība: - *atbalstīts*, ja `idp_type` ir iestatīts uz "User directory of type SAML" Norāda, vai SAML name ID ir jāšifrē. Iespējamās vērtības: 0 - *(noklusējums)* Nešifrēt name ID; 1 - Šifrēt name ID.
encrypt_assertions	integer	Rekvizīta darbība: - *atbalstīts*, ja `idp_type` ir iestatīts uz "User directory of type SAML" Norāda, vai SAML apgalvojumi ir jāšifrē. Iespējamās vērtības: 0 - *(noklusējums)* Nešifrēt apgalvojumus; 1 - Šifrēt apgalvojumus.
nameid_format	string	Rekvizīta darbība: - *atbalstīts*, ja `idp_type` ir iestatīts uz "User directory of type SAML" Identitātes nodrošinātāja pakalpojumu sniedzēja SAML 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*
scim_status	integer	Rekvizīta darbība: - *atbalstīts*, ja `idp_type` ir iestatīts uz "User directory of type SAML" Norāda, vai SAML SCIM nodrošināšana ir iespējota vai atspējota. Iespējamās vērtības: 0 - *(noklusējums)* SCIM nodrošināšana ir atspējota; 1 - SCIM nodrošināšana ir iespējota. Rekvizīta darbība: - *atbalstīts*, ja `idp_type` ir iestatīts uz "User directory of type SAML"

Rekvizīts	Tips	Apraksts
sign_assertions	integer	Norāda, vai SAML apgalvojumi jāparaksta ar SAML parakstu. Iespējamās vērtības: 0 - Neparakstīt apgalvojumus; 1 - *(noklusējums)* Parakstīt apgalvojumus. Rekvizīta darbība: - *atbalstīts*, ja `idp_type` ir iestatīts uz "User directory of type SAML"
sign_authn_requests	integer	Norāda, vai SAML AuthN pieprasījumi jāparaksta ar SAML parakstu. Iespējamās vērtības: 0 - Neparakstīt AuthN pieprasījumus; 1 - *(noklusējums)* Parakstīt AuthN pieprasījumus. Rekvizīta darbība: - *atbalstīts*, ja `idp_type` ir iestatīts uz "User directory of type SAML"
sign_messages	integer	Norāda, vai SAML ziņojumi jāparaksta ar SAML parakstu. Iespējamās vērtības: 0 - Neparakstīt ziņojumus; 1 - *(noklusējums)* Parakstīt ziņojumus. Rekvizīta darbība: - *atbalstīts*, ja `idp_type` ir iestatīts uz "User directory of type SAML"
sign_logout_requests	integer	Norāda, vai SAML atteikšanās pieprasījumi jāparaksta ar SAML parakstu. Iespējamās vērtības: 0 - Neparakstīt atteikšanās pieprasījumus; 1 - *(noklusējums)* Parakstīt atteikšanās pieprasījumus. Rekvizīta darbība: - *atbalstīts*, ja `idp_type` ir iestatīts uz "User directory of type SAML"
sign_logout_responses	integer	Norāda, vai SAML atteikšanās atbildes jāparaksta ar SAML parakstu. Iespējamās vērtības: 0 - Neparakstīt atteikšanās atbildes; 1 - *(noklusējums)* Parakstīt atteikšanās atbildes. Rekvizīta darbība: - *atbalstīts*, ja `idp_type` ir iestatīts uz "User directory of type SAML"
idp_certificate	string	SAML vienotās pieteikšanās (SSO) pakalpojuma iestatīšanai paredzētā pakalpojumu sniedzēja (SP) sertifikāta saturs. Piemēri: *-----BEGIN CERTIFICATE-----* * MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA7... ...more encoded data... -----END CERTIFICATE-----* Rekvizīta darbība: - *atbalstīts*, ja `idp_type` ir iestatīts uz "User directory of type SAML" un `$SSO['CERT_STORAGE']` lietotāja saskarnes konfigurācijas failā ir iestatīts uz `database` - *tikai rakstāms*

Rekvizīts	Tips	Apraksts
sp_private_key	string	SAML vienotās pieteikšanās (SSO) pakalpojuma iestatīšanai paredzētā pakalpojumu sniedzēja (SP) privātās atslēgas saturs. Nodrošina drošu autentifikāciju un datu apmaiņu ar identitātes nodrošinātāju (IdP). Piemēri: *-----BEGIN CERTIFICATE----- MIIEvQIBADANBgkqhkiG9w0BA... ...more encoded data... -----END CERTIFICATE-----* Rekvizīta darbība: - *atbalstīts*, ja `idp_type` ir iestatīts uz "User directory of type SAML" un `$SSO['CERT_STORAGE']` lietotāja saskarnes konfigurācijas failā ir iestatīts uz `database` - *tikai rakstāms*
sp_certificate	string	SAML vienotās pieteikšanās (SSO) pakalpojuma iestatīšanai paredzētā pakalpojumu sniedzēja (SP) sertifikāta saturs. Piemēri: *-----BEGIN CERTIFICATE----- MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA7... ...more encoded data... -----END CERTIFICATE-----* Rekvizīta darbība: - *atbalstīts*, ja `idp_type` ir iestatīts uz "User directory of type SAML" un `$SSO['CERT_STORAGE']` lietotāja saskarnes konfigurācijas failā ir iestatīts uz `database` - *tikai rakstāms*
idp_certificate_hash	string	`idp_certificate` vērtības md5 jaucējvērtība. Tukšam `idp_certificate` tiek atgriezta tukša virkne. Rekvizīta darbība: - *atbalstīts*, ja `idp_type` ir iestatīts uz "User directory of type SAML" un `$SSO['CERT_STORAGE']` lietotāja saskarnes konfigurācijas failā ir iestatīts uz `database` - *tikai lasāms*
sp_private_key_hash	string	`sp_private_key` vērtības md5 jaucējvērtība. Tukšam `sp_private_key` tiek atgriezta tukša virkne. Rekvizīta darbība: - *atbalstīts*, ja `idp_type` ir iestatīts uz "User directory of type SAML" un `$SSO['CERT_STORAGE']` lietotāja saskarnes konfigurācijas failā ir iestatīts uz `database` - *tikai lasāms*
sp_certificate_hash	string	`sp_certificate` vērtības md5 jaucējvērtība. Tukšam `sp_certificate` tiek atgriezta tukša virkne. Rekvizīta darbība: - *atbalstīts*, ja `idp_type` ir iestatīts uz "User directory of type SAML" un `$SSO['CERT_STORAGE']` lietotāja saskarnes konfigurācijas failā ir iestatīts uz `database` - *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*
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*

Īpašība	Type	Apraksts
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 direktorijas. Lietotāju direktoriju nevar dzēst, ja tā tiek tieši izmantota vismaz vienai lietotāju grupai.
 Noklusējuma LDAP lietotāju direktoriju nevar dzēst, ja `authentication.ldap_configured` ir iestatīts uz 1 vai ja paliek vairāk lietotāju direktoriju.

> **Note:**
> Šī metode ir pieejama tikai lietotāja tipam *Super admin*.

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)`

Šī metode ļauj izgūt lietotāju direktorijas 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	

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

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 un lietotāju mediji tiks piešķirti lietotājam). Šāda veida pārbaudei API pieprasījums jāveic uz lietotāju direktoriju, kuram `provision_status` ir iestatīts uz 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 šādi rekvizīti.

Property	Type	Description
usrgrpid	ID	Lietotāju grupas ID. Rekvizīta darbība: - *tikai lasāms* - *obligāts* atjaunināšanas darbībām
name	string	Lietotāju grupas nosaukums. Rekvizīta darbība: - *obligāts* izveides darbībām

Property	Type	Description
debug_mode	integer	Norāda, vai atkļūdošanas režīms ir iespējots vai atspējots. Iespējamās vērtības: 0 - *(noklusējums)* atspējots; 1 - iespējots.
gui_access	integer	Grupas lietotāju lietotāja saskarnes autentifikācijas metode. Iespējamās vērtības: 0 - *(noklusējums)* izmantot sistēmas noklusējuma autentifikācijas metodi; 1 - izmantot iekšējo autentifikāciju; 2 - izmantot LDAP autentifikāciju; 3 - atspējot piekļuvi lietotāja saskarnei.
mfa_status	integer	Norāda, vai grupas lietotājiem MFA ir iespējota vai atspējota. Iespējamās vērtības: 0 - atspējota (visām konfigurētajām MFA metodēm); 1 - iespējota (visām konfigurētajām MFA metodēm).
mfaid	ID	MFA metode, kas tiek izmantota grupas lietotājiem. Rekvizīta darbība: - *atbalstīts Super admin* tipa lietotājiem - *atbalstīts*, ja Authentication object `mfa_status` ir iestatīts uz "Enabled"
proxy_mode	integer	Režīms, kas nosaka piekļuves apstrādi starpniekserveriem, kuri nav iekļauti starpniekserveru grupās. Iespējamās vērtības: 0 - liegto sarakstu; 1 - *(noklusējums)* atļauto sarakstu. Rekvizīta darbība: - *atbalstīts Super admin* tipa lietotājiem.
proxy_group_mode	integer	Režīms, kas nosaka piekļuves apstrādi starpniekserveru grupām (un to starpniekserveriem). Iespējamās vērtības: 0 - liegto sarakstu; 1 - *(noklusējums)* atļauto sarakstu. Rekvizīta darbība: - *atbalstīts Super admin* tipa lietotājiem.
users_status	integer	Norāda, vai lietotāju grupa ir iespējota vai atspējota. Lietotājiem, kuriem ir atcelta piešķiršana, lietotāju grupu nevar iespējot. Iespējamās vērtības: 0 - *(noklusējums)* iespējota; 1 - atspējota.
userdirectoryid	ID	Autentifikācijai izmantotā lietotāju direktorija ID. Rekvizīta 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 atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā User roles.

Parametri

`(object/array)` Lietotāju grupas, kas jāizveido.

Papildus standarta lietotāju grupas rekvizītiem metode pieņem šādus parametrus.

Parameter	Type	Description
hostgroup_rights	object/array	Saimniekdatoru grupu atļaujas, kas jāpiešķir lietotāju grupai.
proxies	object/array	Starpniekserveri, kas jāsaista ar lietotāju grupu. Starpniekserveriem drīkst būt definēts tikai rekvizīts `proxyid`.
proxy_groups	object/array	Starpniekserveru grupas, kas jāsaista ar lietotāju grupu. Starpniekserveru grupām drīkst būt definēts tikai rekvizīts `proxy_groupid`.
templategroup_rights	object/array	Veidņu grupu atļaujas, kas jāpiešķir lietotāju grupai.
tag_filters	array	Uz tagiem balstītas atļaujas, kas jāpiešķir lietotāju grupai.
users	object/array	Lietotāji, kas jāpievieno lietotāju grupai. Lietotājam drīkst būt definēts tikai rekvizīts `userid`.

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:
Deprovisioned lietotāju grupa (lietotāju grupa, kas norādīta `disabled_usrgrpid` Authentication) nevar tikt dzēsta.

Note:
Šī metode ir pieejama tikai *Super admin* lietotāja tipam. Atļaujas izsaukt metodi var atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā User roles.

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)`

Šī metode ļauj izgūt lietotāju grupas atbilstoši norādītajiem parametriem.

> **Note:**
> Šī metode ir pieejama jebkura tipa lietotājiem. Tiesības 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
mfaids	ID/array	Atgriež tikai lietotāju grupas ar norādītajām MFA metodēm. Parametra darbība: - *atbalstīts* lietotājiem ar tipu *Super admin*
mfa_status	integer	Atgriež tikai lietotāju grupas ar norādīto MFA statusu. Atbalstīto statusu sarakstu skatiet lietotāju grupas lapā.

Parametrs	Tips	Apraksts
proxyids	ID/array	Atgriež tikai lietotāju grupas, kurās ir norādītie starpniekserveri. Parametra darbība: - *atbalstīts* lietotājiem ar tipu *Super admin*.
proxy_groupids	ID/array	Atgriež tikai lietotāju grupas, kurās ir norādītās starpniekserveru grupas. Parametra darbība: - *atbalstīts* lietotājiem ar tipu *Super admin*.
selectProxies	ID/array	Atgriež `proxies` rekvizītu ar lietotāju grupas starpniekserveriem. Parametra darbība: - *atbalstīts* lietotājiem ar tipu *Super admin*.
selectProxyGroups	ID/array	Atgriež `proxy_groups` rekvizītu ar lietotāju grupas starpniekserveru grupām. Parametra darbība: - *atbalstīts* lietotājiem ar tipu *Super admin*.
status	integer	Atgriež tikai lietotāju grupas ar norādīto statusu. Atbalstīto statusu sarakstu skatiet lietotāju grupas lapā.
userids	ID/array	Atgriež tikai lietotāju grupas, kurās ir norādītie lietotāji.
usrgrpids	ID/array	Atgriež tikai lietotāju grupas ar norādītajiem ID.
selectTagFilters	query	Atgriež lietotāju grupu uz tagiem balstītās atļaujas `tag_filters` rekvizītā.
selectUsers	query	Atgriež lietotājus no lietotāju grupas `users` rekvizītā. Ierobežojumus atkarībā no lietotāja tipa skatiet `user.get`.
selectHostGroupRights	query	Atgriež lietotāju grupas atļaujas uz hostu grupām `hostgroup_rights` rekvizītā. Piekļuves līmeņu sarakstu hostu grupām skatiet lietotāju grupas lapā.
selectTemplateGroupRights	query	Atgriež lietotāju grupas atļaujas uz veidņu grupām `templategroup_rights` rekvizītā. Piekļuves līmeņu sarakstu veidņu grupām skatiet lietotāju grupas lapā.
limitSelects	integer	Ierobežo apakšvaicājumu atgriezto ierakstu skaitu.
output	query	Atgriežamie lietotāju grupas objekta rekvizīti. Lietotāji ar tipu *Admin* un *User* var izgūt tikai šādus rekvizītus: `usrgrpid`, `name`, `gui_access`, `users_status`, `debug_mode`, `mfa_status`. Noklusējums: `extend`.
sortfield	string/array	Kārto rezultātu pēc norādītajiem rekvizītiem. 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. Tiesības izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā User roles.

Parametri

`(object/array)` Lietotāju grupu rekvizīti, kas jāatjaunina.

Rekvizīts `usrgrpid` ir jādefinē katrai lietotāju grupai; visi pārējie rekvizīti nav obligāti. Tiks atjaunināti tikai nodotie rekvizīti, bet visi pārējie paliks nemainīti.

Papildus standarta lietotāju grupu rekvizītiem metode pieņem šādus parametrus.

Parameter	Type	Description
hostgroup_rights	object/array	Hosts grupu atļaujas, ar kurām aizstāt pašreizējās lietotāju grupai piešķirtās atļaujas.
proxies	object/array	Starpniekserveri, kas jāsaista ar lietotāju grupu. Starpniekserveriem jābūt definētam tikai rekvizītam `proxyid`.
proxy_groups	object/array	Starpniekserveru grupas, kas jāsaista ar lietotāju grupu. Starpniekserveru grupām jābūt definētam tikai rekvizītam `proxy_groupid`.
templategroup_rights	object/array	Veidņu grupu atļaujas, ar kurām aizstāt pašreizējās lietotāju grupai piešķirtās atļaujas.

Parameter	Type	Description
tag_filters	array	Uz tagiem balstītas atļaujas, ar kurām aizstāt pašreizējās lietotāju grupai piešķirtās atļaujas.
users	object/array	Lietotāji, ar kuriem aizstāt pašreizējos lietotāju grupai piešķirtos lietotājus. Lietotājam jābūt definētam tikai rekvizītam `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.

Zema līmeņa atklāšanas kārtulas prototips

Zema līmeņa atklāšanas kārtulas prototipa objektam ir šādi rekvizīti.

Rekvizīts	Tips	Apraksts
itemid	ID	Zema līmeņa atklāšanas kārtulas prototipa ID. Rekvizīta darbība: - *tikai lasāms* - *obligāts* atjaunināšanas darbībām
ruleid	ID	Vecākās zema līmeņa atklāšanas kārtulas vai zema līmeņa atklāšanas kārtulas prototipa ID. Rekvizīta darbība: - *tikai lasāms* - *obligāts* izveides darbībām

Rekvizīts	Tips	Apraksts
delay	string	Zema līmeņa atklāšanas kārtulas prototipa atjaunināšanas intervāls. Pieņem sekunžu skaitu (piemēram, 1, 60, 300), vērtību ar laika sufiksu (piemēram, 30s, 1m, 2h, 1d) vai, pēc izvēles, vienu vai vairākus pielāgotus intervālus, kas atdalīti ar semikoliem. Pielāgotie intervāli var būt elastīgo un plānošanas intervālu kombinācija. Pieņem lietotāja makro. Ja tas tiek izmantots, vērtībai jābūt vienam makro. Vairāki makro vai makro kopā ar tekstu netiek atbalstīti. Elastīgos intervālus var norādīt kā divus makro, 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 Rekvizīta 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	hosts ID, kuram pieder zema līmeņa atklāšanas kārtulas prototips. Rekvizīta darbība: - *konstants* - *obligāts* izveides darbībām
flags	integer	Zema līmeņa atklāšanas kārtulas prototipa izcelsme. Iespējamās vērtības: 3 - zema līmeņa atklāšanas kārtulas prototips; 7 - atklāts zema līmeņa atklāšanas kārtulas prototips. Rekvizīta darbība: - *tikai lasāms*
interfaceid	ID	Zema līmeņa atklāšanas kārtulas prototipa hosts saskarnes ID. Rekvizīta darbība: - *obligāts*, ja zema līmeņa atklāšanas kārtulas prototips pieder hosts un type ir iestatīts uz ”Zabbix agent”, ”IPMI agent”, ”JMX agent” vai ”SNMP agent” - *atbalstīts*, ja zema līmeņa atklāšanas kārtulas prototips pieder hosts un type ir iestatīts uz ”Simple check”, ”External check”, ”SSH agent”, ”TELNET agent” vai ”HTTP agent”
key_	string	Zema līmeņa atklāšanas kārtulas prototipa atslēga. Nepieciešams vismaz viens zema līmeņa atklāšanas makro. Rekvizīta darbība: - *obligāts* izveides darbībām - *tikai lasāms* mantotiem objektiem
name	string	Zema līmeņa atklāšanas kārtulas prototipa nosaukums. Rekvizīta darbība: - *obligāts* izveides darbībām - *tikai lasāms* mantotiem objektiem

Rekvizīts	Tips	Apraksts
type	integer	Zema līmeņa atklāšanas kārtulas prototipa tips. Iespējamās vērtības: 0 - Zabbix agent; 2 - Zabbix trapper; 3 - Simple check; 5 - Zabbix internal; 7 - Zabbix agent (active); 10 - External check; 11 - Database monitor; 12 - IPMI agent; 13 - SSH agent; 14 - TELNET agent; 16 - JMX agent; 18 - Dependent item; 19 - HTTP agent; 20 - SNMP agent; 21 - Script; 22 - Browser; 23 - Nested. Rekvizīta darbība: - *obligāts* izveides darbībām - *tikai lasāms* mantotiem objektiem
url	string	URL virkne. Atbalsta lietotāja makro, {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {HOST.PORT}, {ITEM.ID}, {ITEM.KEY}. Rekvizīta darbība: - *obligāts*, ja `type` ir iestatīts uz ”HTTP agent” - *tikai lasāms* mantotiem objektiem
allow_traps	integer	Atļaut vērtību aizpildīt līdzīgi kā trapper vienumam. Iespējamās vērtības: 0 - *(noklusējums)* nepieņemt ienākošos datus; 1 - pieņemt ienākošos datus. Rekvizīta 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. Rekvizīta darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”SSH agent” vai ”HTTP agent” - *tikai lasāms* mantotiem objektiem (ja `type` ir iestatīts uz ”HTTP agent”)
description	string	Zema līmeņa atklāšanas kārtulas prototipa apraksts.
error	string	Kļūdas teksts, ja rodas problēmas ar zema līmeņa atklāšanas kārtulas prototipa vērtības atjaunināšanu. Rekvizīta darbība: - *tikai lasāms*

Rekvizīts	Tips	Apraksts
follow_redirects	integer	Datu iegūšanas laikā sekot atbildes novirzēm. Iespējamās vērtības: 0 - nesekot novirzēm; 1 - *(noklusējums)* sekot novirzēm. Rekvizīta darbība: - *atbalstīts*, ja `type` ir iestatīts uz "HTTP agent" - *tikai lasāms* mantotiem objektiem
headers	array	Galveņu masīvs, kas tiks nosūtīts, veicot HTTP pieprasījumu. Rekvizīta darbība: - *atbalstīts*, ja `type` ir iestatīts uz "HTTP agent" - *tikai lasāms* mantotiem objektiem
http_proxy	string	HTTP(S) starpniekservera savienojuma virkne. Rekvizīta darbība: - *atbalstīts*, ja `type` ir iestatīts uz "HTTP agent" - *tikai lasāms* mantotiem objektiem
ipmi_sensor	string	IPMI sensors. Rekvizīta 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* mantotiem objektiem
jmx_endpoint	string	JMX aģenta pielāgotā savienojuma virkne. Noklusējums: service:jmx:rmi:///jndi/rmi://{HOST.CONN}:{HOST.PORT}/jmxrmi Rekvizīta darbība: - *atbalstīts*, ja `type` ir iestatīts uz "JMX agent"
lifetime	string	Laika periods, pēc kura vienumi, kas vairs netiek atklāti, tiks dzēsti. Pieņem sekunžu skaitu (piemēram, 1, 60, 300), vērtību ar laika sufiksu (piemēram, `30s`, `1m`, `2h`, `1d`) vai lietotāja makro. Noklusējums: `7d`.
lifetime_type	integer	Scenārijs zaudēto zema līmeņa atklāšanas resursu dzēšanai. Iespējamās vērtības: 0 - *(noklusējums)* dzēst pēc dzīves cikla sliekšņa sasniegšanas; 1 - nedzēst; 2 - dzēst nekavējoties.
enabled_lifetime	string	Laika periods, pēc kura vienumi, kas vairs netiek atklāti, tiks atspējoti. Pieņem sekunžu skaitu (piemēram, 1, 60, 300), vērtību ar laika sufiksu (piemēram, `30s`, `1m`, `2h`, `1d`) vai lietotāja makro. Noklusējums: `0`.
enabled_lifetime_type	integer	Scenārijs zaudēto zema līmeņa atklāšanas resursu atspējošanai. Iespējamās vērtības: 0 - atspējot pēc dzīves cikla sliekšņa sasniegšanas; 1 - neatspējot; 2 - *(noklusējums)* atspējot nekavējoties.
master_itemid	ID	Galvenā vienuma ID. Atklāšanas kārtula nevar būt galvenais vienums citai atklāšanas kārtulai. Rekvizīta darbība: - *obligāts*, ja `type` ir iestatīts uz "Dependent item" - *tikai lasāms* mantotiem objektiem

Rekvizīts	Tips	Apraksts
output_format	integer	Vai atbilde jāpārveido par JSON. Iespējamās vērtības: 0 - *(noklusējums)* saglabāt neapstrādātus datus; 1 - pārveidot par JSON. Rekvizīta darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”HTTP agent” - *tikai lasāms* mantotiem objektiem
params	string	Papildu parametri atkarībā no zema līmeņa atklāšanas kārtulas prototipa tipa: - izpildāmais skripts SSH un Telnet zema līmeņa atklāšanas kārtulām; - SQL vaicājums datubāzes pārraudzības zema līmeņa atklāšanas kārtulām; - formula aprēķinātām zema līmeņa atklāšanas kārtulām; - skripts skripta un pārlūkprogrammas zema līmeņa atklāšanas kārtulām. Rekvizīta darbība: - *obligāts*, ja `type` ir iestatīts uz ”Database monitor”, ”SSH agent”, ”TELNET agent”, ”Script” vai ”Browser” - *tikai lasāms* mantotiem 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` rekvizītiem, kur `name` jābūt unikālam. Rekvizīta darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”Script” vai ”Browser” - *tikai lasāms* mantotiem objektiem
password	string	Autentifikācijas parole. Rekvizīta 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* mantotiem objektiem (ja `type` ir iestatīts uz ”HTTP agent”)
post_type	integer	`posts` rekvizītā saglabāto POST datu pamatteksta tips. Iespējamās vērtības: 0 - *(noklusējums)* neapstrādāti dati; 2 - JSON dati; 3 - XML dati. Rekvizīta darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”HTTP agent” - *tikai lasāms* mantotiem objektiem
posts	string	HTTP(S) pieprasījuma pamatteksta dati. Rekvizīta 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* mantotiem objektiem
privatekey	string	Privātās atslēgas faila nosaukums. Rekvizīta darbība: - *obligāts*, ja `type` ir iestatīts uz ”SSH agent” un `authtype` ir iestatīts uz ”public key”

Rekvizīts	Tips	Apraksts
publickey	string	Publiskās atslēgas faila nosaukums. Rekvizīta 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, veicot HTTP pieprasījumu. Rekvizīta darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”HTTP agent” - *tikai lasāms* mantotiem objektiem
request_method	integer	Pieprasījuma metodes tips. Iespējamās vērtības: 0 - *(noklusējums)* GET; 1 - POST; 2 - PUT; 3 - HEAD. Rekvizīta darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”HTTP agent” - *tikai lasāms* mantotiem 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. Iespējamā vērtība, ja `request_method` ir iestatīts uz ”HEAD”: 1 - galvenes. Rekvizīta darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”HTTP agent” - *tikai lasāms* mantotiem objektiem
snmp_oid	string	SNMP OID. Rekvizīta darbība: - *obligāts*, ja `type` ir iestatīts uz ”SNMP agent” - *tikai lasāms* mantotiem objektiem
ssl_cert_file	string	Publiskās SSL atslēgas faila ceļš. Rekvizīta darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”HTTP agent” - *tikai lasāms* mantotiem objektiem
ssl_key_file	string	Privātās SSL atslēgas faila ceļš. Rekvizīta darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”HTTP agent” - *tikai lasāms* mantotiem objektiem
ssl_key_password	string	SSL atslēgas faila parole. Rekvizīta darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”HTTP agent” - *tikai lasāms* mantotiem objektiem

Rekvizīts	Tips	Apraksts
state	integer	Zema līmeņa atklāšanas kārtulas prototipa stāvoklis. Iespējamās vērtības: 0 - *(noklusējums)* normāls; 1 - netiek atbalstīts. Rekvizīta darbība: - *tikai lasāms*
status	integer	Zema līmeņa atklāšanas kārtulas prototipa statuss. Iespējamās vērtības: 0 - *(noklusējums)* iespējots zema līmeņa atklāšanas kārtulas prototips; 1 - atspējots zema līmeņa atklāšanas kārtulas prototips.
status_codes	string	Nepieciešamo HTTP statusa kodu diapazoni, atdalīti ar komatiem. Komatu atdalītajā sarakstā var izmantot arī lietotāja makro. Piemērs: 200,200-{$M},{$M},200-400 Rekvizīta darbība: - *atbalstīts*, ja `type` ir iestatīts uz "HTTP agent" - *tikai lasāms* mantotiem objektiem
templateid	ID	Vecākās veidnes zema līmeņa atklāšanas kārtulas prototipa ID. Rekvizīta darbība: - *tikai lasāms*
timeout	string	Vienuma datu iegūšanas pieprasījuma taimauts. Pieņem sekunžu skaitu (piemēram, 1, 60, 300) vai vērtību ar laika sufiksu (piemēram, 30s, 1m). Pieņem arī lietotāja makro. Iespējamais vērtību diapazons: 1-600s. Noklusējums: "" - izmantot starpniekservera vai globālos iestatījumus. Rekvizīta 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* mantotiem objektiem
trapper_hosts	string	Atļautie hosts. Rekvizīta darbība: - *atbalstīts*, ja `type` ir iestatīts uz "Zabbix trapper" vai ja `type` ir iestatīts uz "HTTP agent" un `allow_traps` ir iestatīts uz "Allow incoming data". Ja rekvizīts nav norādīts, tam automātiski tiek iestatīts `{$TRAPPER.ALLOWED_HOSTS}` lietotāja makro (noklusējuma vērtība). Pēc noklusējuma `{$TRAPPER.ALLOWED_HOSTS}` ir `127.0.0.1,::1`. Ja rekvizītam ir iestatīta tukša virkne, ienākošie savienojumi no visiem hosts tiks noraidīti.
username	string	Autentifikācijas lietotājvārds. Rekvizīta 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* mantotiem objektiem (ja `type` ir iestatīts uz "HTTP agent")

Rekvizīts	Tips	Apraksts
uuid	string	Universāls unikālais identifikators, ko izmanto importēto zema līmeņa atklāšanas kārtulas prototipu sasaistīšanai ar jau esošajiem. Ja nav norādīts, tas tiek ģenerēts automātiski. Rekvizīta darbība: - *atbalstīts*, ja zema līmeņa atklāšanas kārtulas prototips pieder veidnei
verify_host	integer	Vai pārbaudīt, ka savienojuma hosts nosaukums atbilst hosts sertifikātā norādītajam nosaukumam. Iespējamās vērtības: 0 - *(noklusējums)* nepārbaudīt; 1 - pārbaudīt. Rekvizīta darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”HTTP agent” - *tikai lasāms* mantotiem objektiem
verify_peer	integer	Vai pārbaudīt, ka hosts sertifikāts ir autentisks. Iespējamās vērtības: 0 - *(noklusējums)* nepārbaudīt; 1 - pārbaudīt. Rekvizīta darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”HTTP agent” - *tikai lasāms* mantotiem 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”
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 šādi rekvizīti.

Rekvizīts	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, vai JSON satur kļūdu; 17 - Pārbaudīt, vai XML satur kļūdu; 20 - Atmest nemainīto vērtību 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. Rekvizīta darbība: - *obligāts*
params	string	Papildu parametri, kas tiek izmantoti priekšapstrādes opcijai. Vairāki parametri tiek atdalīti ar jaunās rindas (\n) rakstzīmi. Rekvizīta darbība: - *obligāts*, ja `type` ir iestatīts uz ”Regulārā izteiksme” (5), ”XML XPath” (11), ”JSONPath” (12), ”Atbilst regulārajai izteiksmei” (14), ”Neatbilst regulārajai izteiksmei” (15), ”Pārbaudīt, vai JSON satur kļūdu” (16), ”Pārbaudīt, vai XML satur kļūdu” (17), ”Atmest nemainīto vērtību 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)

Rekvizīts	Tips	Apraksts
error_handler	integer	Darbības tips, kas tiek izmantots priekšapstrādes soļa kļūmes gadījumā. Iespējamās vērtības: 0 - Kļūdas ziņojumu iestata Zabbix serveris; 1 - Atmest vērtību; 2 - Iestatīt pielāgotu vērtību; 3 - Iestatīt pielāgotu kļūdas ziņojumu. Rekvizīta darbība: - *obligāts*, ja `type` ir iestatīts uz ”Regulārā izteiksme” (5), ”XML XPath” (11), ”JSONPath” (12), ”Atbilst regulārajai izteiksmei” (14), ”Neatbilst regulārajai izteiksmei” (15), ”Pārbaudīt, vai JSON satur kļūdu” (16), ”Pārbaudīt, vai XML satur kļūdu” (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. Rekvizīta darbība: - *obligāts*, 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
5	Regulārā izteiksme	pattern[1]	output[2]		0, 1, 2, 3
11	XML XPath	path[3]			0, 1, 2, 3
12	JSONPath	path[3]			0, 1, 2, 3
14	Atbilst regulārajai izteiksmei	pattern[1]			0, 1, 2, 3
15	Neatbilst regulārajai izteiksmei	pattern[1]			0, 1, 2, 3
16	Pārbaudīt, vai JSON satur kļūdu	path[3]			0, 1, 2, 3
17	Pārbaudīt, vai XML satur kļūdu	path[3]			0, 1, 2, 3
20	Atmest nemainīto vērtību ar sirdspukstu	seconds^4, 5^			
21	JavaScript	script[2]			

Priekšapstrādes tips	Nosaukums	1. parametrs	2. parametrs	3. parametrs	Atbalstītie kļūdu apstrādātāji
23	Prometheus uz JSON	pattern^5, 6^			0, 1, 2, 3
24	CSV uz JSON	character[2]	character[2]	0,1	0, 1, 2, 3
25	Aizstāt	search string[2]	replacement[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 - Integer no BITS		0, 1, 2, 3
29	SNMP walk uz JSON[7]	Field name[2]	OID prefix[2]	Formāts: 0 - Nemainīts 1 - UTF-8 no Hex-STRING 2 - MAC no Hex-STRING 3 - Integer no BITS	0, 1, 2, 3
30	SNMP get vērtība	Formāts: 1 - UTF-8 no Hex-STRING 2 - MAC no Hex-STRING 3 - Integer 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 makro
 [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 makro.
 [7] Atbalsta vairākus "Field name,OID prefix,Format records" 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.

Īpašība	Type	Apraksts
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*
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”

Īpašība	Tips	Apraksts
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*
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 kārtulas prototipa ignorēšanas darbības periods

LLD kārtulas prototipa ignorēšanas darbības periods ir atjaunināšanas intervāla vērtība, kas iestatīta atklātajam vienumam. Tam ir šādi rekvizīti:

Property	Type	Description
delay	string	Ignorēt vienuma prototipa atjaunināšanas intervālu. Pieņem sekunžu skaitu (piemēram, 1, 60, 300), vērtību ar laika sufiksu (piemēram, 30s, 1m, 2h, 1d) vai, pēc izvēles, vienu vai vairākus pielāgotus intervālus, kas visi atdalīti ar semikoliem. Pielāgotie intervāli var būt elastīgo un plānošanas intervālu kombinācija. Pieņem lietotāja makro vai LLD makro. Ja tiek izmantots makro, vērtībai jābūt vienam makro. Vairāki makro vai makro kombinācijā ar tekstu netiek atbalstīti. Elastīgos intervālus var norādīt kā divus makro, 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` Rekvizīta darbība: - *obligāts*

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. Tiesības izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Papildu 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. Tiesības izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā 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)`

Šī metode ļauj izgūt LLD noteikumu prototipus atbilstoši norādītajiem parametriem.

Note:
Šī metode ir pieejama jebkura tipa lietotājiem. Tiesības izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāja 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`.

Parametrs	Tips	Apraksts
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.
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. Tiesības izsaukt šo metodi var atsaukt lietotāju lomu iestatījumos. Plašāku informāciju skatiet sadaļā User roles.

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 noteikums

Zema līmeņa atklāšanas noteikuma objektam ir šādi rekvizīti.

Rekvizīts	Tips	Apraksts
itemid	ID	LLD noteikuma ID. Rekvizīta darbība: - *tikai lasāms* - *obligāts* atjaunināšanas darbībām
delay	string	LLD noteikuma atjaunināšanas intervāls. Pieņem sekunžu skaitu (piemēram, 1, 60, 300), vērtību ar laika sufiksu (piemēram, `30s`, `1m`, `2h`, `1d`) vai, pēc izvēles, vienu vai vairākus pielāgotus intervālus, kas atdalīti ar semikoliem. Pielāgotie intervāli var būt elastīgo un plānošanas intervālu kombinācija. Pieņem lietotāja makro. Ja tiek izmantots makro, vērtībai jābūt vienam makro. Vairāki makro vai makro kombinācijā ar tekstu netiek atbalstīti. Elastīgos intervālus var norādīt kā divus makro, 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` Rekvizīta 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	Tā hosta ID, kuram pieder LLD noteikums. Rekvizīta darbība: - *konstants* - *obligāts* izveides darbībām
flags	integer	Atklāšanas noteikuma izcelsme. Iespējamās vērtības: 1 - zema līmeņa atklāšanas noteikums; 5 - no prototipa pārveidots zema līmeņa atklāšanas noteikums. Rekvizīta darbība: - *tikai lasāms*
interfaceid	ID	LLD noteikuma hosta saskarnes ID. Rekvizīta darbība: - *obligāts*, ja LLD noteikums pieder hostam un `type` ir iestatīts uz "Zabbix agent", "IPMI agent", "JMX agent" vai "SNMP agent" - *atbalstīts*, ja LLD noteikums pieder hostam un `type` ir iestatīts uz "Simple check", "External check", "SSH agent", "TELNET agent" vai "HTTP agent"

Rekvizīts	Tips	Apraksts
key_	string	LLD noteikuma atslēga. Rekvizīta darbība: - *obligāts* izveides darbībām - *tikai lasāms* mantotiem objektiem
name	string	LLD noteikuma nosaukums. Rekvizīta darbība: - *obligāts* izveides darbībām - *tikai lasāms* mantotiem objektiem
type	integer	LLD noteikuma tips. Iespējamās vērtības: 0 - Zabbix agent; 2 - Zabbix trapper; 3 - Simple check; 5 - Zabbix internal; 7 - Zabbix agent (active); 10 - External check; 11 - Database monitor; 12 - IPMI agent; 13 - SSH agent; 14 - TELNET agent; 16 - JMX agent; 18 - Dependent item; 19 - HTTP agent; 20 - SNMP agent; 21 - Script; 22 - Browser; 23 - Nested. Šis tips ir atļauts tikai tad, ja noteikuma vecākelements ir veidne (tiek pieņemts, ka atklāšanas laikā tā ir saistīta ar hosta prototipu) vai atklātais hosts (kura saknē ir LLD noteikums). Rekvizīta darbība: - *obligāts* izveides darbībām - *tikai lasāms* mantotiem objektiem
url	string	URL virkne. Atbalsta lietotāja makro, {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {HOST.PORT}, {ITEM.ID}, {ITEM.KEY}. Rekvizīta darbība: - *obligāts*, ja `type` ir iestatīts uz ”HTTP agent” - *tikai lasāms* mantotiem objektiem
allow_traps	integer	Atļaut vērtības aizpildīšanu līdzīgi kā trapper vienumam. Iespējamās vērtības: 0 - *(noklusējums)* nepieņemt ienākošos datus; 1 - pieņemt ienākošos datus. Rekvizīta darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”HTTP agent”

Rekvizīts	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 - pamata; 2 - NTLM; 3 - Kerberos; 4 - Digest. Rekvizīta darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”SSH agent” vai ”HTTP agent” - *tikai lasāms* mantotiem objektiem (ja `type` ir iestatīts uz ”HTTP agent”)
description	string	LLD noteikuma apraksts.
error	string	Kļūdas teksts, ja rodas problēmas ar LLD noteikuma vērtības atjaunināšanu. Rekvizīta darbība: - *tikai lasāms*
follow_redirects	integer	Datu aptaujas laikā sekot atbildes novirzēm. Iespējamās vērtības: 0 - nesekot novirzēm; 1 - *(noklusējums)* sekot novirzēm. Rekvizīta darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”HTTP agent” - *tikai lasāms* mantotiem objektiem
headers	array	Galveņu masīvs, kas tiks nosūtīts, veicot HTTP pieprasījumu. Rekvizīta darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”HTTP agent” - *tikai lasāms* mantotiem objektiem
http_proxy	string	HTTP(S) starpniekservera savienojuma virkne. Rekvizīta darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”HTTP agent” - *tikai lasāms* mantotiem objektiem
ipmi_sensor	string	IPMI sensors. Rekvizīta 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* mantotiem objektiem
jmx_endpoint	string	JMX aģenta pielāgotā savienojuma virkne. Noklusējums: service:jmx:rmi:///jndi/rmi://{HOST.CONN}:{HOST.PORT}/jmxrmi Rekvizīta darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”JMX agent”
lifetime	string	Laika periods, pēc kura vienumi, kas vairs netiek atklāti, tiks dzēsti. Pieņem sekunžu skaitu (piemēram, 3600), vērtību ar laika sufiksu (piemēram, `3600s`, `180m`, `24h`, `1d`) vai lietotāja makro. Noklusējums: `7d`.

Rekvizīts	Tips	Apraksts
lifetime_type	integer	Scenārijs zaudēto LLD resursu dzēšanai. Iespējamās vērtības: 0 - *(noklusējums)* dzēst pēc dzīves cikla sliekšņa sasniegšanas; 1 - nedzēst; 2 - dzēst nekavējoties.
enabled_lifetime	string	Laika periods, pēc kura vienumi, kas vairs netiek atklāti, tiks atspējoti. Pieņem sekunžu skaitu (piemēram, 1, 60, 300), vērtību ar laika sufiksu (piemēram, 30s, 1m, 2h, 1d) vai lietotāja makro. Noklusējums: 0.
enabled_lifetime_type	integer	Scenārijs zaudēto LLD resursu atspējošanai. Iespējamās vērtības: 0 - atspējot pēc dzīves cikla sliekšņa sasniegšanas; 1 - neatspējot; 2 - *(noklusējums)* atspējot nekavējoties.
master_itemid	ID	Galvenā vienuma ID. Atklāšanas noteikums nevar būt galvenais vienums citam atklāšanas noteikumam. Rekvizīta darbība: - *obligāts*, ja `type` ir iestatīts uz ”Dependent item” - *tikai lasāms* mantotiem objektiem
output_format	integer	Vai atbilde ir jāpārveido par JSON. Iespējamās vērtības: 0 - *(noklusējums)* saglabāt neapstrādātā formātā; 1 - pārveidot par JSON. Rekvizīta darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”HTTP agent” - *tikai lasāms* mantotiem objektiem
params	string	Papildu parametri atkarībā no LLD noteikuma tipa: - izpildāmais skripts SSH un Telnet LLD noteikumiem; - SQL vaicājums datubāzes pārraudzības LLD noteikumiem; - formula aprēķinātajiem LLD noteikumiem; - skripts skripta un pārlūkprogrammas LLD noteikumiem. Rekvizīta darbība: - *obligāts*, ja `type` ir iestatīts uz ”Database monitor”, ”SSH agent”, ”TELNET agent”, ”Script” vai ”Browser” - *tikai lasāms* mantotiem 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` rekvizītiem, kur `name` jābūt unikālam. Rekvizīta darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”Script” vai ”Browser” - *tikai lasāms* mantotiem objektiem
password	string	Autentifikācijas parole. Rekvizīta 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* mantotiem objektiem (ja `type` ir iestatīts uz ”HTTP agent”)

Rekvizīts	Tips	Apraksts
post_type	integer	`posts` rekvizītā saglabāto POST datu pamatteksta tips. Iespējamās vērtības: 0 - *(noklusējums)* neapstrādāti dati; 2 - JSON dati; 3 - XML dati. Rekvizīta darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”HTTP agent” - *tikai lasāms* mantotiem objektiem
posts	string	HTTP(S) pieprasījuma pamatteksta dati. Rekvizīta 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* mantotiem objektiem
privatekey	string	Privātās atslēgas faila nosaukums. Rekvizīta darbība: - *obligāts*, ja `type` ir iestatīts uz ”SSH agent” un `authtype` ir iestatīts uz ”public key”
publickey	string	Publiskās atslēgas faila nosaukums. Rekvizīta 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, veicot HTTP pieprasījumu. Rekvizīta darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”HTTP agent” - *tikai lasāms* mantotiem objektiem
request_method	integer	Pieprasījuma metodes tips. Iespējamās vērtības: 0 - *(noklusējums)* GET; 1 - POST; 2 - PUT; 3 - HEAD. Rekvizīta darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”HTTP agent” - *tikai lasāms* mantotiem objektiem
retrieve_mode	integer	Kura atbildes daļa ir 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. Rekvizīta darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”HTTP agent” - *tikai lasāms* mantotiem objektiem

Rekvizīts	Tips	Apraksts
snmp_oid	string	SNMP OID. Rekvizīta darbība: - *obligāts*, ja `type` ir iestatīts uz ”SNMP agent” - *tikai lasāms* mantotiem objektiem
ssl_cert_file	string	Publiskās SSL atslēgas faila ceļš. Rekvizīta darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”HTTP agent” - *tikai lasāms* mantotiem objektiem
ssl_key_file	string	Privātās SSL atslēgas faila ceļš. Rekvizīta darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”HTTP agent” - *tikai lasāms* mantotiem objektiem
ssl_key_password	string	SSL atslēgas faila parole. Rekvizīta darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”HTTP agent” - *tikai lasāms* mantotiem objektiem
state	integer	LLD noteikuma stāvoklis. Iespējamās vērtības: 0 - *(noklusējums)* normāls; 1 - netiek atbalstīts. Rekvizīta darbība: - *tikai lasāms*
status	integer	LLD noteikuma statuss. Iespējamās vērtības: 0 - *(noklusējums)* iespējots LLD noteikums; 1 - atspējots LLD noteikums.
status_codes	string	Nepieciešamo HTTP statusa kodu diapazoni, atdalīti ar komatiem. Atbalsta arī lietotāja makro kā daļu no komatiem atdalīta saraksta. Piemērs: `200,200-{$M},{$M},200-400` Rekvizīta darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”HTTP agent” - *tikai lasāms* mantotiem objektiem
templateid	ID	Vecākelementa veidnes LLD noteikuma ID. Rekvizīta darbība: - *tikai lasāms*

Rekvizīts	Tips	Apraksts
timeout	string	Vienuma datu aptaujas pieprasījuma taimauts. Pieņem sekunžu skaitu (piemēram, 1, 60, 300) vai vērtību ar laika sufiksu (piemēram, 30s, 1m). Pieņem arī lietotāja makro. Iespējamo vērtību diapazons: 1-600s. Noklusējums: "" - izmantot starpniekservera/globālos iestatījumus. Rekvizīta 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* mantotiem objektiem, - ja tiek lietota noklusējuma vērtība un vienums pieder hostam, kuru pārrauga nepieejams starpniekserveris vai starpniekserveru grupa, tiek izmantots globālais taimauts.
trapper_hosts	string	Atļautie hosti. Rekvizīta darbība: - *atbalstīts*, ja `type` ir iestatīts uz "Zabbix trapper" vai ja `type` ir iestatīts uz "HTTP agent" un `allow_traps` ir iestatīts uz "Allow incoming data". Ja rekvizīts nav norādīts, tam automātiski tiek iestatīts `{$TRAPPER.ALLOWED_HOSTS}` lietotāja makro (noklusējuma vērtība). Pēc noklusējuma `{$TRAPPER.ALLOWED_HOSTS}` ir `127.0.0.1,::1`. Ja rekvizīts ir iestatīts uz tukšu virkni, ienākošie savienojumi no visiem hostiem tiks noraidīti.
username	string	Autentifikācijas lietotājvārds. Rekvizīta 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* mantotiem objektiem (ja `type` ir iestatīts uz "HTTP agent")
uuid	string	Universāls unikāls identifikators, ko izmanto importēto LLD noteikumu sasaistīšanai ar jau esošajiem noteikumiem. Ja nav norādīts, tas tiek ģenerēts automātiski. Rekvizīta darbība: - *atbalstīts*, ja LLD noteikums pieder veidnei
verify_host	integer	Vai pārbaudīt, ka savienojuma hosta nosaukums atbilst hosta sertifikātā norādītajam nosaukumam. Iespējamās vērtības: 0 - *(noklusējums)* nepārbaudīt; 1 - pārbaudīt. Rekvizīta darbība: - *atbalstīts*, ja `type` ir iestatīts uz "HTTP agent" - *tikai lasāms* mantotiem objektiem

Rekvizīts	Tips	Apraksts
verify_peer	integer	Vai pārbaudīt, ka hosta sertifikāts ir autentisks. Iespējamās vērtības: 0 - *(noklusējums)* nepārbaudīt; 1 - pārbaudīt. Rekvizīta darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”HTTP agent” - *tikai lasāms* mantotiem 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*

Īpašība	Tips	Apraksts
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 šādi rekvizīti.

Property	Type	Description
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, vai JSON satur kļūdu; 17 - Pārbaudīt, vai XML satur kļūdu; 20 - Atmest nemainīto vērtību 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. Rekvizīta darbība: - *required*
params	string	Papildu parametri, kas tiek izmantoti priekšapstrādes opcijā. Vairāki parametri tiek atdalīti ar jaunas rindas (\n) rakstzīmi. Rekvizīta darbība: - *required*, ja `type` ir iestatīts uz ”Regulārā izteiksme” (5), ”XML XPath” (11), ”JSONPath” (12), ”Atbilst regulārajai izteiksmei” (14), ”Neatbilst regulārajai izteiksmei” (15), ”Pārbaudīt, vai JSON satur kļūdu” (16), ”Pārbaudīt, vai XML satur kļūdu” (17), ”Atmest nemainīto vērtību 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)

Property	Type	Description
error_handler	integer	Darbības tips, kas tiek izmantots priekšapstrādes soļa kļūmes gadījumā. Iespējamās vērtības: 0 - Kļūdas ziņojumu iestata Zabbix serveris; 1 - Atmest vērtību; 2 - Iestatīt pielāgotu vērtību; 3 - Iestatīt pielāgotu kļūdas ziņojumu. Rekvizīta darbība: - *required*, ja `type` ir iestatīts uz ”Regulārā izteiksme” (5), ”XML XPath” (11), ”JSONPath” (12), ”Atbilst regulārajai izteiksmei” (14), ”Neatbilst regulārajai izteiksmei” (15), ”Pārbaudīt, vai JSON satur kļūdu” (16), ”Pārbaudīt, vai XML satur kļūdu” (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. Rekvizīta darbība: - *required*, ja `error_handler` ir iestatīts uz ”Iestatīt pielāgotu vērtību” vai ”Iestatīt pielāgotu kļūdas ziņojumu”

Katram priekšapstrādes tipam tiek atbalstīti šādi parametri un kļūdu apstrādātāji.

Preprocessing type	Name	Parameter 1	Parameter 2	Parameter 3	Supported error handlers
5	Regulārā izteiksme	pattern[1]	output[2]		0, 1, 2, 3
11	XML XPath	path[3]			0, 1, 2, 3
12	JSONPath	path[3]			0, 1, 2, 3
14	Atbilst regulārajai izteiksmei	pattern[1]			0, 1, 2, 3
15	Neatbilst regulārajai izteiksmei	pattern[1]			0, 1, 2, 3
16	Pārbaudīt, vai JSON satur kļūdu	path[3]			0, 1, 2, 3
17	Pārbaudīt, vai XML satur kļūdu	path[3]			0, 1, 2, 3
20	Atmest nemainīto vērtību ar sirdspukstu	seconds^4, 5^			
21	JavaScript	script[2]			

Preprocessing type	Name	Parameter 1	Parameter 2	Parameter 3	Supported error handlers
23	Prometheus uz JSON	pattern^5, 6^			0, 1, 2, 3
24	CSV uz JSON	character[2]	character[2]	0,1	0, 1, 2, 3
25	Aizstāt	search string[2]	replacement[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 - Integer no BITS		0, 1, 2, 3
29	SNMP walk uz JSON[7]	Field name[2]	OID prefix[2]	Formāts: 0 - Nemainīts 1 - UTF-8 no Hex-STRING 2 - MAC no Hex-STRING 3 - Integer no BITS	0, 1, 2, 3
30	SNMP get vērtība	Formāts: 1 - UTF-8 no Hex-STRING 2 - MAC no Hex-STRING 3 - Integer 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 "Field name,OID prefix,Format records" ierakstus, kas atdalīti ar jaunas 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.

Īpašība	Tips	Apraksts
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”

Īpašība	Type	Apraksts
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”
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 (atjaunināšanas 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 atjaunināšanas intervāla vērtība, kas iestatīta atklātajam vienumam. Tam ir šādi rekvizīti:

Property	Type	Description
delay	string	Ignorēt vienuma prototipa atjaunināšanas intervālu. Pieņem sekunžu skaitu (piemēram, 1, 60, 300), vērtību ar laika sufiksu (piemēram, 30s, 1m, 2h, 1d) vai, pēc izvēles, vienu vai vairākus pielāgotus intervālus, kas visi atdalīti ar semikoliem. Pielāgotie intervāli var būt elastīgo un plānošanas intervālu kombinācija. Pieņem lietotāja makro vai LLD makro. Ja tiek izmantots makro, vērtībai jābūt vienam makro. Vairāki makro vai makro kombinācijā ar tekstu netiek atbalstīti. Elastīgos intervālus var norādīt kā divus makro, 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` Rekvizīta darbība: - *obligāts*

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 pārrakstīšanas operācijas taga

LLD noteikuma pārrakstīšanas operācijas taga objekts satur taga nosaukumu un vērtību, kas tiek iestatīti atklātajam objektam. Tam ir šādas īpašības:

Property	Type	Description
tag	string	Jauns taga nosaukums. Pieņem arī lietotāja makro un LLD makro. Property behavior: - *required*
value	string	Jauna taga vērtība. Pieņem arī lietotāja makro un LLD makro.

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. Tiesības izsaukt šo metodi var atsaukt 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 noteikumus.

Note:
Šī metode ir pieejama tikai lietotāju tipiem *Admin* un *Super admin*. Atļaujas izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā Lietotāju lomas.
 Lietotājs nevarēs dzēst LLD noteikumu, kas atklāj entītijas, izmantojot nepieejamu starpniekserveri. Plašāku informāciju skatiet sadaļā lietotāju grupas konfigurācija (*Starpniekserveru piekļuves saraksts*).

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 noteikumus atbilstoši norādītajiem parametriem.

Note:
Šī metode ir pieejama jebkura tipa lietotājiem. Tiesības 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.
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	

Parametrs	Tips	Apraksts
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

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. Tiesības izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā Lietotāju lomas.

Parametri

`(object/array)` Atjaunināmo 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.
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 noteikumam

Pievienojiet filtru, lai *{#FSTYPE}* makro saturs atbilstu *@File systems for discovery* regulārajai izteiksmei.

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*

Lomu noteikumi

Lomu noteikumu objektam ir šādi rekvizīti:

Property	Type	Description
ui	array	UI elementu objektu masīvs.
ui.default_access	integer	Vai piekļuve jauniem UI elementiem ir iespējota. Iespējamās vērtības: 0 - Atspējota; 1 - *(noklusējums)* Iespējota.
services.read.mode	integer	Tikai lasīšanas piekļuve pakalpojumiem. Iespējamās vērtības: 0 - Tikai lasīšanas piekļuve pakalpojumiem, kas norādīti rekvizītā `services.read.list` vai atbilst rekvizītam `services.read.tag`; 1 - *(noklusējums)* Tikai lasīšanas piekļuve visiem pakalpojumiem.
services.read.list	array	Service objektu masīvs. Norādītajiem pakalpojumiem, tostarp pakārtotajiem pakalpojumiem, lietotāja lomai tiks piešķirta tikai lasīšanas piekļuve. Tikai lasīšanas piekļuve neignorēs lasīšanas un rakstīšanas piekļuvi pakalpojumiem. Rekvizīta darbība: - *atbalstīts*, ja `services.read.mode` ir iestatīts uz "0"
services.read.tag	object	Service tag objektu masīvs. Pakalpojumiem, kas atbilst tagam, tostarp pakārtotajiem pakalpojumiem, lietotāja lomai tiks piešķirta tikai lasīšanas piekļuve. Tikai lasīšanas piekļuve neignorēs lasīšanas un rakstīšanas piekļuvi pakalpojumiem. Rekvizīta darbība: - *atbalstīts*, ja `services.read.mode` ir iestatīts uz "0"
services.write.mode	integer	Lasīšanas un rakstīšanas piekļuve pakalpojumiem. Iespējamās vērtības: 0 - *(noklusējums)* Lasīšanas un rakstīšanas piekļuve pakalpojumiem, kas norādīti rekvizītā `services.write.list` vai atbilst rekvizītam `services.write.tag`; 1 - Lasīšanas un rakstīšanas piekļuve visiem pakalpojumiem.
services.write.list	array	Service objektu masīvs. Norādītajiem pakalpojumiem, tostarp pakārtotajiem pakalpojumiem, lietotāja lomai tiks piešķirta lasīšanas un rakstīšanas piekļuve. Lasīšanas un rakstīšanas piekļuve ignorēs tikai lasīšanas piekļuvi pakalpojumiem. Rekvizīta darbība: - *atbalstīts*, ja `services.write.mode` ir iestatīts uz "0"

Property	Type	Description
services.write.tag	object	Service tag objektu masīvs. Pakalpojumiem, kas atbilst tagam, tostarp pakārtotajiem pakalpojumiem, lietotāja lomai tiks piešķirta lasīšanas un rakstīšanas piekļuve. Lasīšanas un rakstīšanas piekļuve ignorēs tikai lasīšanas piekļuvi pakalpojumiem. Rekvizīta darbība: - *atbalstīts*, ja `services.write.mode` ir iestatīts uz "0"
modules	array	Moduļu 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 - *(noklusējums)* Atspējota; 1 - Iespējota.
api.mode	integer	Režīms API metožu apstrādei, kas norādītas rekvizītā `api`. Iespējamās vērtības: 0 - *(noklusējums)* Noraidīto saraksts; 1 - Atļauto saraksts.
api	array	API metožu masīvs.
actions	array	Darbību objektu masīvs.
actions.default_access	integer	Vai piekļuve jaunām darbībām ir iespējota. Iespējamās vērtības: 0 - Atspējota; 1 - *(noklusējums)* Iespējota.
devices.access	integer	Vai ir iespējota piekļuve lietotāja mobilajām ierīcēm izmantošanai ar Zabbix mobilo lietotni. Iespējamās vērtības: 0 - (*noklusējums lietotājam un administratoram*) Atspējota; 1 - (*noklusējums superadministratoram*) Iespējota. Rekvizīta darbība: - *atbalstīts*, ja `EnableMobileDevices` ir iespējots
devices.actions	array	Ierīču darbību objektu masīvs. Rekvizīta darbība: - *atbalstīts*, ja `devices.access` ir iestatīts uz "1"
devices.actions.default_access	integer	Vai piekļuve jaunām ierīču darbībām ir iespējota. Iespējamās vērtības: 0 - (*noklusējums lietotājam un administratoram*) Atspējota; 1 - (*noklusējums superadministratoram*) Iespējota. Rekvizīta darbība: - *atbalstīts*, ja `EnableMobileDevices` ir iespējots

Lietotāja saskarnes elements

Lietotāja saskarnes elementa objektam ir šādas īpašības:

Property	Type	Description
name	string	Lietotāja saskarnes elementa nosaukums. Iespējamās vērtības, ja Role objekta `type` ir iestatīts uz ”User”, ”Admin” vai ”Super admin”: `monitoring.dashboard` - *Dashboards*; `monitoring.problems` - *Monitoring* > *Problems*; `monitoring.hosts` - *Monitoring* > *Hosts*; `monitoring.latest_data` - *Monitoring* > *Latest data*; `monitoring.maps` - *Monitoring* > *Maps*; `services.services` - *Services* > *Services*; `services.sla_report` - *Services* > *SLA report*; `inventory.overview` - *Inventory* > *Overview*; `inventory.hosts` - *Inventory* > *Hosts*; `reports.availability_report` - *Reports* > *Availability report*; `reports.top_triggers` - *Reports* > *Triggers top 100*. Iespējamās vērtības, ja Role objekta `type` ir iestatīts uz ”Admin” vai ”Super admin”: `monitoring.discovery` - *Monitoring* > *Discovery*; `services.sla` - *Services* > *SLA*; `reports.scheduled_reports` - *Reports* > *Scheduled reports*; `reports.notifications` - *Reports* > *Notifications*; `configuration.template_groups` - *Data collection* > *Template groups*; `configuration.host_groups` - *Data collection* > *Host groups*; `configuration.templates` - *Data collection* > *Templates*; `configuration.hosts` - *Data collection* > *Hosts*; `configuration.maintenance` - *Data collection* > *Maintenance*; `configuration.discovery` - *Data collection* > *Discovery*; `configuration.trigger_actions` - *Alerts* > *Actions* > *Trigger actions*; `configuration.service_actions` - *Alerts* > *Actions* > *Service actions*; `configuration.discovery_actions` - *Alerts* > *Actions* > *Discovery actions*; `configuration.autoregistration_actions` - *Alerts* > *Actions* > *Autoregistration actions*; `configuration.internal_actions` - *Alerts* > *Actions* > *Internal actions*. Iespējamās vērtības, ja Role objekta `type` ir iestatīts uz ”Super admin”: `reports.system_info` - *Reports* > *System information*; `reports.audit` - *Reports* > *Audit log*; `reports.action_log` - *Reports* > *Action log*; `configuration.event_correlation` - *Data collection* > *Event correlation*; `administration.media_types` - *Alerts* > *Media types*; `administration.scripts` - *Alerts* > *Scripts*; `administration.user_groups` - *Users* > *User groups*; `administration.user_roles` - *Users* > *User roles*; `administration.users` - *Users* > *Users*; `administration.api_tokens` - *Users* > *API tokens*; `administration.authentication` - *Users* > *Authentication*; `administration.linked_devices` - *Users* > *Devices*; `administration.general` - *Administration* > *General*; `administration.audit_log` - *Administration* > *Audit log*; `administration.housekeeping` - *Administration* > *Housekeeping*; `administration.proxy_groups` - *Administration* > *Proxy groups*; `administration.proxies` - *Administration* > *Proxies*; `administration.macros` - *Administration* > *Macros*; `administration.queue` - *Administration* > *Queue*. Īpašības darbība: - *required*

Property	Type	Description
status	integer	Norāda, vai piekļuve lietotāja saskarnes elementam ir iespējota. Iespējamās vērtības: 0 - Atspējots; 1 - *(default)* 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:

Property	Type	Description
name	string	Darbības nosaukums. Iespējamās vērtības, ja Role objekta `type` ir iestatīts uz ”User”, ”Admin” vai ”Super admin”: `edit_dashboards` - Izveidot un rediģēt informācijas paneļus; `edit_maps` - Izveidot un rediģēt kartes; `add_problem_comments` - Pievienot problēmu komentārus; `change_severity` - Mainīt problēmas nopietnību; `acknowledge_problems` - Apstiprināt problēmas; `suppress_problems` - Apspiest problēmas; `close_problems` - Aizvērt problēmas; `execute_scripts` - Izpildīt skriptus; `manage_api_tokens` - Pārvaldīt API pilnvaras; `change_problem_ranking` - Mainīt problēmas klasifikāciju no cēloņa uz simptomu un otrādi; `edit_own_media` - Atļaut lietotājiem izveidot un rediģēt savus multivides līdzekļus. 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, `select_server_for_monitoring` - Atļaut lietotājiem izveidot hostus, kurus uzrauga serveris, kā arī tīkla atklāšanas noteikumi, ko izpilda serveris. Iespējamā vērtība, ja Role objekta `type` ir iestatīts uz ”User” vai ”Admin”: `invoke_execute_now` - Atļaut lietotājiem, kuriem hostā ir tikai lasīšanas tiesības, izpildīt vienumu pārbaudes. Iespējamā vērtība, ja Role objekta `type` ir iestatīts uz ”Super admin”: `edit_user_media` - Atļaut lietotājiem izveidot un rediģēt multivides līdzekļus citiem lietotājiem. Īpašības darbība: - *required*
status	integer	Vai piekļuve darbības veikšanai ir iespējota. Iespējamās vērtības: 0 - Atspējota; 1 - *(noklusējums)* Iespējota.

Ierīces darbība

Ierīces darbības objektam ir šādas īpašības:

Property	Type	Description
name	string	Ierīces darbības nosaukums. Iespējamās vērtības, ja Role objekta `type` ir iestatīts uz ”User”, ”Admin” vai ”Super admin”: `manage_own` - Atļaut lietotājiem pievienot vai noņemt savas mobilās ierīces. Iespējamās vērtības, ja Role objekta `type` ir iestatīts uz ”Super admin”: `manage_user` - Atļaut lietotājiem pievienot vai noņemt mobilo ierīču lietotājiem. Īpašības darbība: - *required*
status	integer	Norāda, vai piekļuve darbības veikšanai ir iespējota. Iespējamās vērtības: 0 - Atspējota; 1 - (*default*) 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. Tiesības izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā User roles.

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 atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā User roles.

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)`

Šī metode ļauj izgūt lomas atbilstoši norādītajiem parametriem.

Note:
Šī metode ir pieejama jebkura tipa lietotājiem. Tiesības izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāju lomas.

Parametri

`(object)` Parametri, kas nosaka vēlamo izvadi.

Metode atbalsta šādus parametrus.

Parameter	Type	Description
roleids	ID/array	Atgriež tikai lomas ar norādītajiem ID.
current	boolean	Ja iestatīts uz `true`, atgriež tikai pašreizējā lietotāja lomu.
selectRules	query	Atgriež `rules` rekvizītu ar lomas noteikumiem.
selectUsers	query	Atgriež `users` rekvizītu ar lietotājiem, kuriem ir piešķirta šī loma. Skatiet `user.get`, lai uzzinātu par ierobežojumiem, kas balstīti uz lietotāja tipu.
sortfield	string/array	Kārto rezultātu pēc norādītajiem rekvizītiem. Iespējamās vērtības: `roleid`, `name`.
countOutput	boolean	Šie parametri ir aprakstīti atsauces komentāros.
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 atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet User roles.

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.

Multivides tips

Multivides tipa objektam ir šādas īpašības.

Īpašība	Tips	Apraksts
mediatypeid	ID	Multivides tipa ID. Īpašības darbība: - *tikai lasāms* - *obligāts* atjaunināšanas darbībām
name	string	Multivides tipa nosaukums. Īpašības darbība: - *obligāts* izveides darbībām
type	integer	Multivides tipa izmantotais transporta veids. Iespējamās vērtības: 0 - E-pasts; 1 - Skripts; 2 - SMS; 4 - Webhook; 5 - Push. Īpašības darbība: - *obligāts* izveides darbībām
exec_path	string	Skripta faila nosaukums (piemēram, notification.sh), kas atrodas direktorijā, kura norādīta servera konfigurācijas parametrā AlertScriptsPath. Īpašības darbība: - *obligāts*, ja `type` ir iestatīts uz ”Script”
gsm_modem	string	GSM modema seriālās ierīces nosaukums. Īpašības darbība: - *obligāts*, ja `type` ir iestatīts uz ”SMS”
passwd	string	Autentifikācijas parole. Īpašības darbība: - *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.
smtp_email	string	E-pasta adrese, no kuras tiks nosūtīti paziņojumi. Īpašības darbība: - *obligāts*, ja `type` ir iestatīts uz ”Email”
smtp_helo	string	SMTP HELO. Īpašības darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”Email”
smtp_server	string	SMTP serveris. Īpašības darbība: - *obligāts*, ja `type` ir iestatīts uz ”Email”
smtp_port	integer	SMTP servera ports, ar kuru izveidot savienojumu. Noklusējums: 25. Īpašības darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”Email”

Īpašība	Tips	Apraksts
smtp_security	integer	Izmantojamais SMTP savienojuma drošības līmenis. Iespējamās vērtības: 0 - *(noklusējums)* None; 1 - STARTTLS; 2 - SSL/TLS. Īpašības darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”Email”
smtp_verify_host	integer	SMTP SSL resursdatora pārbaude. Iespējamās vērtības: 0 - *(noklusējums)* Nē; 1 - Jā. Īpašības darbība: - *atbalstīts*, ja `smtp_security` ir iestatīts uz ”STARTTLS” vai ”SSL/TLS”
smtp_verify_peer	integer	SMTP SSL partnera pārbaude. Iespējamās vērtības: 0 - *(noklusējums)* Nē; 1 - Jā. Īpašības darbība: - *atbalstīts*, ja `smtp_security` ir iestatīts uz ”STARTTLS” vai ”SSL/TLS”
smtp_authentication	integer	Izmantojamā SMTP autentifikācijas metode. Iespējamās vērtības: 0 - *(noklusējums)* None; 1 - Normal password; 2 - OAuth token. OAuth autentifikācija nav atļauta *Office365 relay* e-pasta pakalpojumu sniedzējam. Īpašības darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”Email”
redirection_url	string	Zabbix lietotāja saskarnes URL, uz kuru novirzīt atpakaļ OAuth autorizācijas laikā. Noklusējums: API iestatījumu īpašības `url` vērtība ar daļu `zabbix.php?action=oauth.authorize` Īpašības darbība: - *obligāts*, ja `smtp_authentication` ir iestatīts uz ”OAuth token”
client_id	string	OAuth autorizācijas serverī reģistrētais klienta identifikators. Īpašības darbība: - *obligāts*, ja `smtp_authentication` ir iestatīts uz ”OAuth token”
client_secret	string	OAuth autorizācijas serverī reģistrētais klienta noslēpums. Pieejams tikai lietotājam ar Super Admin tipu. Īpašības darbība: - *obligāts*, ja `smtp_authentication` ir iestatīts uz ”OAuth token”
authorization_url	string	OAuth URL ar parametriem piekļuves un atsvaidzināšanas pilnvaru iegūšanai. Īpašības darbība: - *obligāts*, ja `smtp_authentication` ir iestatīts uz ”OAuth token”

Īpašība	Tips	Apraksts
token_url	string	OAuth URL autorizācijas pilnvaras apmaiņai pret piekļuves un atsvaidzināšanas pilnvarām. Šo URL serveris izmanto arī nederīgas piekļuves pilnvaras atsvaidzināšanai. Īpašības darbība: - *obligāts*, ja `smtp_authentication` ir iestatīts uz ”OAuth token”
tokens_status	integer	Bitu maska pilnvaru statusam. Iespējamās vērtības: 0 - (noklusējums) Abas pilnvaras satur nederīgu vērtību; 1 - Piekļuves pilnvara satur derīgu vērtību; 2 - Atsvaidzināšanas pilnvara satur derīgu vērtību; 3 - Abas pilnvaras satur derīgu vērtību. Īpašības darbība: - *atbalstīts*, ja `smtp_authentication` ir iestatīts uz ”OAuth token”
access_token	string	OAuth piekļuves pilnvaras vērtība. Īpašības darbība: - *obligāts*, ja `smtp_authentication` ir iestatīts uz ”OAuth token”
access_token_updated	timestamp	Pēdējo `access_token` izmaiņu laikspiedols, ko serveris veicis, atsvaidzinot ar `refresh_token`, vai API veicis pilnvaru izmaiņu laikā. Īpašības darbība: - *atbalstīts*, 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 pilnvaru izmaiņu laikā. Laikspiedolu aprēķina, pieskaitot `access_token_updated` vērtību. Īpašības darbība: - *atbalstīts*, ja `smtp_authentication` ir iestatīts uz ”OAuth token”
refresh_token	string	OAuth atsvaidzināšanas pilnvaras vērtība. Īpašības darbība: - *obligāts*, ja `smtp_authentication` ir iestatīts uz ”OAuth token”
status	integer	Norāda, vai multivides tips ir iespējots. Iespējamās vērtības: 0 - *(noklusējums)* Iespējots; 1 - Atspējots.
username	string	Lietotājvārds. Īpašības darbība: - *atbalstīts*, ja `smtp_authentication` ir iestatīts uz ”Normal password”
maxsessions	integer	Maksimālais vienlaikus apstrādājamo paziņojumu skaits. Iespējamā vērtība, 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 paziņojuma nosūtīšanai. Iespējamās vērtības: 1-100. Noklusējums: 3.

Īpašība	Tips	Apraksts
attempt_interval	string	Intervāls starp atkārtotiem mēģinājumiem. Pieņem sekunžu skaitu (piemēram, 1, 30, 60) vai vērtību ar laika sufiksu (piemēram, 30s, 1m, 1h). Iespējamās vērtības: 0-1h. Noklusējums: 10s.
message_format	integer	Ziņojuma formāts. Iespējamās vērtības: 0 - Vienkāršs teksts; 1 - *(noklusējums)* HTML.
script	text	Īpašības darbība: - *atbalstīts*, ja `type` ir iestatīts uz "Email" Webhook skripta pamatteksts (JavaScript).
timeout	string	Īpašības darbība: - *obligāts*, ja `type` ir iestatīts uz "Webhook" Webhook skripta taimauts. Pieņem sekunžu skaitu (piemēram, 1, 60) vai vērtību ar laika sufiksu (piemēram, 30s, 1m). Iespējamās vērtības: 1-60s. Noklusējums: 30s.
process_tags	integer	Īpašības darbība: - *atbalstīts*, ja `type` ir iestatīts uz "Webhook" Apstrādāt JSON īpašību vērtības Webhook skripta atbildē kā tagus. Šie tagi tiek pievienoti visiem esošajiem problēmu tagiem. Iespējamās vērtības: 0 - *(noklusējums)* Ignorēt Webhook skripta atbildi; 1 - Apstrādāt Webhook skripta atbildi kā tagus.
show_event_menu	integer	Īpašības darbība: - *atbalstīts*, ja `type` ir iestatīts uz "Webhook" Iekļaut notikumu izvēlnē ierakstu, kas novirza uz pielāgotu URL. problem.get un event.get izvadei tiek pievienota arī `urls` īpašība. Iespējamās vērtības: 0 - *(noklusējums)* Neiekļaut notikumu izvēlnes ierakstu vai `urls` īpašību; 1 - Iekļaut notikumu izvēlnes ierakstu un `urls` īpašību.
event_menu_url	string	Īpašības darbība: - *atbalstīts*, ja `type` ir iestatīts uz "Webhook" URL, kas tiek izmantots notikumu izvēlnes ierakstā un `urls` īpašībā, ko atgriež problem.get un event.get.
event_menu_name	string	Īpašības darbība: - *atbalstīts*, ja `show_event_menu` ir iestatīts uz "Include event menu entry and `urls` property" Nosaukums, kas tiek izmantots notikumu izvēlnes ierakstā un `urls` īpašībā, ko atgriež problem.get un event.get. Īpašības darbība: - *atbalstīts*, ja `show_event_menu` ir iestatīts uz "Include event menu entry and `urls` property"

Īpašība	Tips	Apraksts
parameters	array	Webhook vai skripta parametri. Īpašības darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”Webhook” vai ”Script”
description	text	Multivides 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*
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.

Īpašība	Type	Apraksts
message	string	Ziņojuma teksts.

mediatype.create

Apraksts

`object mediatype.create(object/array mediaTypes)`

Šī metode ļauj izveidot jaunus multivides tipus.

Note:
Šī metode ir pieejama tikai lietotāja tipam *Super admin*. Atļaujas izsaukt šo metodi var atsaukt lietotāju lomu iestatījumos. Plašāku informāciju skatiet sadaļā Lietotāju lomas.
 Ja `$ZBX_FEATURE_FLAGS['media_type_denylist']` funkcionalitātes karogā lietotāja saskarnes konfigurācijas failā ir norādītas piegādes metodes, lietotājam nav atļauts izsaukt šo metodi attiecīgajiem multivides tipiem.

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 multivides tipus.

> **Note:**
> Šī metode ir pieejama tikai lietotājiem ar *Super administratora* lietotāja tipu. Atļaujas izsaukt šo metodi var atsaukt lietotāju lomu iestatījumos. Plašāku informāciju skatiet sadaļā Lietotāju lomas.
 Ja `$ZBX_FEATURE_FLAGS['media_type_denylist']` funkcionalitātes karodziņā lietotāja saskarnes konfigurācijas failā ir norādītas piegādes metodes, lietotājam nav atļauts izsaukt šo metodi attiecīgajiem multivides tipiem.

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)

Metode ļauj izgūt multivides tipus atbilstoši norādītajiem parametriem.

Note:
Šī metode ir pieejama visu veidu lietotājiem. Atļaujas izsaukt metodi var atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā Lietotāju lomas.
 Ja $ZBX_FEATURE_FLAGS['media_type_denylist'] funkcionalitātes karodziņā lietotāja saskarnes konfigurācijas failā ir norādītas piegādes metodes, lietotājam nav atļauts izsaukt šo metodi attiecīgajiem multivides tipiem.

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.

Parametrs	Tips	Apraksts
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.
countOutput	boolean	Iespējamās Mediju tipa objekta īpašības *Admin* un *User* tipa lietotājiem: `name`, `description`. 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 multivides tipus.

Note:
Šī metode ir pieejama tikai lietotāja tipam *Super admin*. Atļaujas izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā Lietotāju lomas.
 Ja `$ZBX_FEATURE_FLAGS['media_type_denylist']` funkcionalitātes karodziņā lietotāja saskarnes konfigurācijas failā ir norādītas piegādes metodes, lietotājam nav atļauts izsaukt šo metodi attiecīgajiem multivides tipiem.

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”
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 jaunus MFA metodes.

> **Note:**
> Šī metode ir pieejama tikai *Super admin* lietotāja tipam. Atļaujas izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet 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. Tiesības izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Papildinformāciju skatiet sadaļā User roles.

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)`

Šī metode ļauj izgūt MFA metodes atbilstoši norādītajiem parametriem.

> **Note:**
> Šī metode ir pieejama tikai *Super admin* lietotāja tipam. Atļaujas izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet User roles.

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`.

Parametrs	Tips	Apraksts
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 atsaukt lietotāja lomu iestatījumos. Plašāku informāciju skatiet Lietotāja lomas.

Parametri

`(object/array)` Atjauninámo 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.

Property	Type	Description
moduleid	ID	Moduļa ID, kas saglabāts datubāzē. Īpašības uzvedība: - *tikai lasāms* - *obligāts* atjaunināšanas operācijām
id	string	Unikāls moduļa ID, ko izstrādātājs definējis moduļa failā `manifest.json`. Iespējamās vērtības iebūvētajiem moduļiem: skatiet īpašības "type" aprakstu sadaļā Dashboard widget. Īpašības uzvedība: - *obligāts* izveides operācijām
relative_path	string	Ceļš uz moduļa direktoriju attiecībā pret Zabbix lietotāja saskarnes direktoriju. Iespējamās vērtības: `widgets/*` - iebūvēto logrīku moduļiem; `modules/*` - trešo pušu moduļiem. Īpašības uzvedība: - *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 lietotāja tipam *Super admin*. Atļaujas izsaukt šo metodi var atsaukt lietotāju lomu iestatījumos. Plašāku informāciju skatiet sadaļā Lietotāju lomas.
 Ja lietotāja saskarnes konfigurācijas failā `$ZBX_FEATURE_FLAGS['modules_config_enabled']` ir iestatīts uz `false`, lietotājam nebūs atļauts izsaukt šo metodi.

> **Attention:**
> Moduļu faili ir manuāli jāizpako pareizajās apakšdirektorijās atbilstoši moduļu rekvizītam `relative_path`.

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 lietotāja tipam *Super admin*. Atļaujas izsaukt šo metodi var atsaukt lietotāju lomu iestatījumos. Plašāku informāciju skatiet sadaļā Lietotāju lomas.
 Ja lietotāja saskarnes konfigurācijas failā `$ZBX_FEATURE_FLAGS['modules_config_enabled']` ir iestatīts uz `false`, lietotājam nebūs atļauts izsaukt šo metodi.

> **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 lietotāja tipam *Super admin*. Atļaujas izsaukt šo metodi var atsaukt lietotāju lomu iestatījumos. Plašāku informāciju skatiet sadaļā Lietotāju lomas.
 Ja lietotāja saskarnes konfigurācijas failā `$ZBX_FEATURE_FLAGS['modules_config_enabled']` ir iestatīts uz `false`, lietotājam nebūs atļauts izsaukt šo metodi.

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 lietotāja tipam *Super admin*. Atļaujas izsaukt šo metodi var atsaukt lietotāju lomu iestatījumos. Plašāku informāciju skatiet sadaļā Lietotāju lomas.
 Ja lietotāja saskarnes konfigurācijas failā `$ZBX_FEATURE_FLAGS['modules_config_enabled']` ir iestatīts uz `false`, lietotājam nebūs atļauts izsaukt šo metodi.

Parametri

`(object/array)` Atjaunināmā 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.

Īpašība	Tips	Apraksts
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.
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.

Īpašība	Tips	Apraksts
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.
- Noņemt 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.
- Slāpēt notikumu. Ja notikums jau ir slāpēts, šī darbība tiks izlaista.
- Noņemt notikuma slāpēšanu. Ja notikums nav slāpēts, šī darbība tiks izlaista.
- Mainīt notikuma rangu uz cēloni. Ja notikums jau ir ierindots kā cēlonis, šī darbība tiks izlaista.
- Mainīt notikuma rangu uz simptomu. Ja notikums jau ir ierindots kā simptoms, šī darbība tiks izlaista.

Attention:
Var atjaunināt tikai trigera notikumus.
 Var atjaunināt tikai problēmu notikumus.
 Lai aizvērtu notikumu vai mainītu notikuma nopietnību, ir nepieciešamas lasīšanas/rakstīšanas tiesības trigerim.
 Lai aizvērtu notikumu, trigerī ir jābūt atļautai manuālai aizvēršanai.

Note:
Šī metode ir pieejama jebkura tipa lietotājiem. Tiesības izsaukt metodi var atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā Lietotāju lomas.

Parametri

`(object/array)` Parametri, kas satur to notikumu ID un atjaunināšanas darbības, kas jāveic.

Parameter	Type	Description
eventids	ID/array	To notikumu ID, kuri jāapstiprina. Parameter behavior: - *required*
action	integer	Notikuma atjaunināšanas darbība(s). Iespējamās bitkartes vērtības: 1 - aizvērt problēmu; 2 - apstiprināt notikumu; 4 - pievienot ziņojumu; 8 - mainīt nopietnību; 16 - atcelt 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 pieņemama jebkura iespējamo bitkartes vērtību summa (piemēram, 6, lai apstiprinātu notikumu un atstātu ziņojumu, vai 34, lai apstiprinātu un apspiestu notikumu). Parameter behavior: - *required*
cause_eventid	ID	Cēloņa notikuma ID. Parameter behavior: - *required* if `action` contains the ”change event rank to symptom” bit
message	string	Ziņojuma teksts. Parameter behavior: - *required* if `action` contains the ”add message” bit
severity	integer	Jaunā notikumu nopietnība. Iespējamās vērtības: 0 - nav klasificēts; 1 - informācija; 2 - brīdinājums; 3 - vidēja; 4 - augsta; 5 - katastrofāla. Parameter behavior: - *required* if `action` contains the ”change severity” bit
suppress_until	integer	Unix laikspiedols, līdz kuram notikums ir jāapslāpē. Ja iestatīta vērtība ”0”, apspiešana būs beztermiņa. Parameter behavior: - *required* if `action` contains the ”suppress event” bit

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
}
```

Notikuma slāpēšana uz nenoteiktu laiku

Slāpēt vienu notikumu bez derīguma termiņa.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
```

```
    "method": "event.acknowledge",
    "params": {
        "eventids": "20427",
        "action": 32,
        "suppress_until": 0
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "eventids": [
            20427
        ]
    },
    "id": 1
}
```

Notikuma atcelšana no slāpēšanas

Atceliet viena iepriekš slāpēta notikuma slāpēšanu.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "event.acknowledge",
    "params": {
        "eventids": "20427",
        "action": 64
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "eventids": [
            20427
        ]
    },
    "id": 1
}
```

Apstiprināt un slāpēt notikumu

Apstipriniet vienu vai vairākus notikumus, pievienojiet ziņojumu un slāpējiet tos vienā pieprasījumā.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "event.acknowledge",
    "params": {
        "eventids": [
            "20427",
            "20428"
        ],
        "action": 38,
        "message": "Slāpēšana plānotas apkopes laikā.",
        "suppress_until": 0
    },
```

```
    "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 dzēsta entītijas notikumus, ja šie notikumi vēl nav noņemti ar housekeeper.

Note:
Šī metode ir pieejama jebkura tipa lietotājiem. Tiesības izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā Lietotāja lomas.

Parametri

`(object)` Parametri, kas definē vēlamo izvadi.

Metode atbalsta šādus parametrus.

Parameter	Type	Description
eventids	ID/array	Atgriezt tikai notikumus ar norādītajiem ID.
groupids	ID/array	Atgriezt tikai notikumus, ko izveidojuši objekti, kas pieder norādītajām host grupām.
hostids	ID/array	Atgriezt tikai notikumus, ko izveidojuši objekti, kas pieder norādītajiem hosts.
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 - trigeris 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 tos notikumus, kuriem ir veikta norādītā notikuma atjaunināšanas darbība. Vairāku darbību gadījumā kā bitmasku izmantojiet jebkuru pieņemamu bitu vērtību summu (piemēram, 34 apstiprināšanai un notikuma slāpēšanai).

Parameter	Type	Description
action_userids	ID/array	Atgriezt tikai notikumus ar norādītajiem lietotāju ID, kuri veica notikuma atjaunināšanas darbības.
suppressed	boolean	Ja iestatīts uz `true`, atgriezt tikai slāpētos notikumus.
symptom	boolean	Ja iestatīts uz `true`, atgriezt tikai simptomu notikumus.
severities	integer/array	Atgriezt tikai notikumus ar norādītajām notikumu nopietnībām. Attiecas tikai tad, ja object ir trigeris.
trigger_severities	integer/array	Atgriezt tikai notikumus ar norādītajām trigeru nopietnībām. Attiecas tikai tad, ja object ir trigeris.
evaltype	integer	Tagu novērtēšanas metode. Iespējamās vērtības: 0 - *(noklusējums)* And/Or; 2 - Or.
tags	array	Atgriezt tikai notikumus ar norādītajiem tagiem. Formāts: `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`. Tukšs masīvs atgriež visus notikumus. Iespējamās operatora vērtības: 0 - *(noklusējums)* Contains; 1 - Equals; 2 - Does not contain; 3 - Does not equal; 4 - Exists; 5 - Does not exist.
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 pēc tā.
time_till	timestamp	Atgriezt tikai notikumus, kas izveidoti norādītajā laikā vai pirms tā.
problem_time_from	timestamp	Atgriež tikai tos notikumus, kas bija problēmas stāvoklī no `problem_time_from`, neatkarīgi no to pašreizējā statusa. Attiecas tikai tad, ja avots ir trigeris notikums un object ir trigeris. Šis parametrs ir obligāts, ja ir norādīts `problem_time_till`.
problem_time_till	timestamp	Atgriež tikai tos notikumus, kas bija problēmas stāvoklī līdz `problem_time_till`, neatkarīgi no to pašreizējā statusa. Attiecas tikai tad, ja avots ir trigeris notikums un object ir trigeris. Šis parametrs ir obligāts, ja ir norādīts `problem_time_from`.
value	integer/array	Atgriezt tikai notikumus ar norādītajām vērtībām.

Parameter	Type	Description
selectAcknowledges	query	Atgriezt `acknowledges` īpašību ar notikumu atjauninājumiem. Notikumu atjauninājumi ir sakārtoti apgrieztā hronoloģiskā secībā. Notikuma atjauninājuma objektam ir šādas īpašības: `acknowledgeid` - `(ID)` apstiprinājuma ID; `userid` - `(ID)` lietotāja ID, kurš atjaunināja notikumu; `clock` - `(timestamp)` laiks, kad notikums tika atjaunināts; `message` - `(string)` ziņojuma teksts; `action` - `(integer)` veiktais atjaunināšanas darbības veids, skatiet `event.acknowledge`; `old_severity` - `(integer)` notikuma nopietnība pirms šīs atjaunināšanas darbības; `new_severity` - `(integer)` notikuma nopietnība pēc šīs atjaunināšanas darbības; `suppress_until` - `(timestamp)` laiks, līdz kuram notikums tiks slāpēts; `taskid` - `(ID)` uzdevuma ID, ja pašreizējais notikums tiek pakļauts prioritātes maiņai; `maintenanceid` - `(ID)` apkopes ID; `username` - `(string)` lietotāja lietotājvārds, kurš atjaunināja notikumu; `name` - `(string)` lietotāja vārds, kurš atjaunināja notikumu; `surname` - `(string)` lietotāja uzvārds, kurš atjaunināja notikumu. Atbalsta `count`.
selectAlerts	query	Atgriezt `alerts` īpašību ar notikuma ģenerētajiem brīdinājumiem. Brīdinājumi ir sakārtoti apgrieztā hronoloģiskā secībā.
selectHosts	query	Atgriezt `hosts` īpašību ar hosts, kas satur objektu, kurš izveidoja notikumu. Atbalstīts tikai notikumiem, ko ģenerējuši trigeri, vienumi vai LLD noteikumi.
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 apkopju un manuālo slāpēšanu sarakstu: `maintenanceid` - `(ID)` apkopes ID; `userid` - `(ID)` lietotāja ID, kurš slāpēja notikumu; `suppress_until` - `(integer)` laiks, līdz kuram notikums ir slāpēts.
selectTags	query	Atgriezt `tags` īpašību ar notikuma tagiem.
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.
editable	boolean	
excludeSearch	boolean	
limit	integer	
output	query	

Parameter	Type	Description
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Atgrieztās vērtības

`(integer/array)` Atgriež vienu no šiem:

- objektu masīvu
- iegūto objektu skaitu, ja ir izmantots parametrs `countOutput`, bet parametrs `groupBy` nav izmantots
- objektu masīvu ar agregācijas rezultātiem, ja ir izmantots parametrs `groupBy`

Piemēri

Trigeru notikumu izgūšana

Izgūt 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": "Load average is too high (per CPU load over 1.5 for 5m)",
            "severity": "0",
            "r_eventid": "0",
            "c_eventid": "0",
            "correlationid": "0",
            "userid": "0",
            "suppressed": "0",
            "cause_eventid": "0",
            "acknowledges": [],
            "opdata": "Load averages(1m 5m 15m): (1.6328 3.0522 2.2515), # of CPUs: 2",
            "suppression_data": [],
            "tags": [
                {
```

```
                "tag": "scope",
                "value": "capacity"
            },
            {
                "tag": "scope",
                "value": "performance"
            },
            {
                "tag": "component",
                "value": "cpu"
            },
            {
                "tag": "class",
                "value": "os"
            },
            {
                "tag": "target",
                "value": "linux"
            }
        ],
        "urls": []
    },
    {
        "eventid": "4",
        "source": "0",
        "object": "0",
        "objectid": "22395",
        "clock": "1728657737",
        "value": "1",
        "acknowledged": "1",
        "ns": "460759366",
        "name": "Load average is too high (per CPU load over 1.5 for 5m)",
        "severity": "3",
        "r_eventid": "20",
        "c_eventid": "0",
        "correlationid": "0",
        "userid": "0",
        "suppressed": "1",
        "cause_eventid": "0",
        "acknowledges": [
            {
                "acknowledgeid": "1",
                "userid": "1",
                "clock": "1728657938",
                "message": "Testing environment. Please disregard this alert.",
                "action": "38",
                "old_severity": "0",
                "new_severity": "0",
                "suppress_until": "1728744338",
                "taskid": "0",
                "maintenanceid": "0",
                "username": "Admin",
                "name": "Zabbix",
                "surname": "Administrator"
            }
        ],
        "opdata": "Load averages(1m 5m 15m): (1.6328 3.0522 2.2515), # of CPUs: 2",
        "suppression_data": [
            {
                "maintenanceid": "0",
                "suppress_until": "1728744338",
                "userid": "1"
```

```
                }
            ],
            "tags": [
                {
                    "tag": "scope",
                    "value": "capacity"
                },
                {
                    "tag": "scope",
                    "value": "performance"
                },
                {
                    "tag": "component",
                    "value": "cpu"
                },
                {
                    "tag": "class",
                    "value": "os"
                },
                {
                    "tag": "target",
                    "value": "linux"
                }
            ],
            "urls": []
        }
    ],
    "id": 1
}
```

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, ko apstiprinājis norādītais lietotājs

Notikumu izgūšana, ko 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",
            "suppressed": "0",
            "cause_eventid": "0",
            "acknowledges": [
                {
                    "userid": "10",
                    "action": "2"
                }
            ],
            "opdata": "",
            "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.

Property	Type	Description
serviceid	ID	Pakalpojuma ID. Property behavior: - *tikai lasāms* - *obligāts* atjaunināšanas darbībām
algorithm	integer	Statusa aprēķināšanas noteikums. Piemērojams tikai tad, ja pastāv bērna pakalpojumi. Iespējamās vērtības: 0 - iestatīt statusu uz OK; 1 - viskritiskākais, ja visiem bērniem ir problēmas; 2 - viskritiskākais no bērna pakalpojumiem. Property behavior: - *obligāts* izveides darbībām
name	string	Pakalpojuma nosaukums. Property behavior: - *obligāts* izveides darbībām
sortorder	integer	Pakalpojuma pozīcija, ko izmanto kārtošanai. Iespējamās vērtības: 0-999. Property behavior: - *obligāts* izveides darbībām
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 smaguma pakāpēm); 2 - samazināt izplatīto statusu par norādīto `propagation_value` (par 1 līdz 5 smaguma pakāpēm); 3 - ignorēt šo pakalpojumu - statuss vispār netiek izplatīts uz vecāka pakalpojumu; 4 - iestatīt fiksētu pakalpojuma statusu, izmantojot norādīto `propagation_value`. Property behavior: - *obligāts*, ja ir iestatīts `propagation_value`

Property	Type	Description
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 - Katastrofāls. Iespējamās vērtības, ja `propagation_rule` ir iestatīts uz ”4”: -1 - OK; 0 - Nav klasificēts; 1 - Informācija; 2 - Brīdinājums; 3 - Vidējs; 4 - Augsts; 5 - Katastrofāls. Property behavior: - *obligāts*, ja ir iestatīts `propagation_rule`
status	integer	Vai pakalpojums ir OK vai problēmas stāvoklī. Ja pakalpojums ir problēmas stāvoklī, `status` ir vienāds ar vai nu viskritiskākās problēmas smaguma pakāpi, vai bērna pakalpojuma augstāko statusu problēmas stāvoklī. Ja pakalpojums ir OK stāvoklī, `status` ir vienāds ar: -1. Property behavior: - *tikai lasāms*
description	string	Pakalpojuma apraksts.
uuid	string	Universāls unikāls identifikators, ko izmanto importēto pakalpojumu sasaistīšanai ar jau esošajiem. Ja nav norādīts, tiek ģenerēts automātiski.
created_at	integer	Unix laikspiedols, kad pakalpojums tika izveidots.
readonly	integer	Piekļuve pakalpojumam. Iespējamās vērtības: 0 - Lasīšana un rakstīšana; 1 - Tikai lasāms. Property behavior: - *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 maiņu. Tam ir šādas īpašības.

Property	Type	Description
clock	timestamp	Laiks, kad notikusi pakalpojuma stāvokļa maiņa.
value	integer	Pakalpojuma statuss. Skatiet pakalpojuma `status` īpašību, lai iegūtu iespējamo vērtību sarakstu.

Problēmas tags

Problēmas tagi ļauj saistīt pakalpojumus ar problēmu notikumiem. Problēmas taga objektam ir šādas īpašības.

Property	Type	Description
tag	string	Problēmas taga nosaukums. Property behavior: - *required*
operator	integer	Kartēšanas nosacījuma operators. Iespējamās vērtības: 0 - *(default)* 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. Tiesības izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Skatiet Lietotāja lomas, lai iegūtu papildu informāciju.

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. Tiesības izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā User roles.

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)`

Šī metode ļauj izgūt pakalpojumus atbilstoši norādītajiem parametriem.

> **Note:**
> Šī metode ir pieejama jebkura tipa lietotājiem. Tiesības izsaukt šo metodi var atsaukt 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. Tiesības izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Skatiet Lietotāju lomas, lai iegūtu papildu informāciju.

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.

Informācijas panelis

Informācijas paneļa objektam ir šādas īpašības.

Property	Type	Description
dashboardid	ID	Informācijas paneļa ID. Property behavior: - *tikai lasāms* - *obligāts* atjaunināšanas operācijām
name	string	Informācijas paneļa nosaukums. Property behavior: - *obligāts* izveides operācijām
userid	ID	Lietotāja ID, kurš ir informācijas paneļa īpašnieks.
private	integer	Informācijas paneļa koplietošanas veids. Iespējamās vērtības: 0 - publisks informācijas panelis; 1 - *(noklusējums)* privāts informācijas panelis.
display_period	integer	Noklusējuma slaidrādes intervāls (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.

Informācijas paneļa lapa

Informācijas paneļa lapas objektam ir šādas īpašības.

Īpašība	Tips	Apraksts
dashboard_pageid	ID	Informācijas paneļa lapas ID. Īpašības darbība: - *tikai lasāms*
name	string	Informācijas paneļa lapas nosaukums. Noklusējums: tukša virkne.

Īpašība	Tips	Apraksts
display_period	integer	Informācijas paneļa slaidrādes intervāls (sekundēs). Iespējamās vērtības: 0, 10, 30, 60, 120, 600, 1800, 3600. Noklusējums: 0 (tiks izmantots noklusējuma slaidrādes intervāls).
widgets	array	Informācijas 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 informācijas paneļus.

Note:
Šī metode ir pieejama jebkura tipa lietotājiem. Tiesības izsaukt šo metodi var atsaukt 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 informācijas paneļus.

Note:
Šī metode ir pieejama jebkura tipa lietotājiem. Tiesības izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā 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)`

Šī metode ļauj izgūt informācijas paneļus atbilstoši norādītajiem parametriem.

Note:
Šī metode ir pieejama jebkura tipa lietotājiem. Tiesības 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
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 informācijas paneļus.

> **Note:**
> Šī metode ir pieejama jebkura tipa lietotājiem. Tiesības izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāja 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 informācijas paneļa logrīka lauku objektiem ļauj konfigurēt *Darbību žurnāla* logrīku `dashboard.create` un `dashboard.update` metodēs.

> **Attention:**
> Logrīka `fields` īpašības netiek validētas informācijas paneļa izveides vai atjaunināšanas laikā. Tas ļauj lietotājiem modificēt iebūvētos logrīkus un izveidot pielāgotus logrīkus, taču rada arī risku izveidot vai atjaunināt logrīkus nepareizi. Lai nodrošinātu veiksmīgu *Darbību žurnāla* logrīka izveidi vai atjaunināšanu, skatiet tālākajās tabulās aprakstīto parametru darbību.

Parametri

Tālāk norādīti parametri, kurus atbalsta logrīks *Darbību žurnāls*.

Parametrs	tips	nosaukums	vērtība
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.
Adresāti	11	userids.0	Lietotāja ID. Piezīme: Lai konfigurētu vairākus lietotājus, izveidojiet atsevišķu informācijas paneļa logrīka lauka objektu katram lietotājam, īpašuma nosaukumā izmantojot secīgi palielinātu numuru.
Darbības	12	actionids.0	Darbības ID. Piezīme: Lai konfigurētu vairākas darbības, izveidojiet atsevišķu informācijas paneļa logrīka lauka objektu katrai darbībai, īpašuma nosaukumā izmantojot secīgi palielinātu numuru.
Multivides veidi	13	mediatypeids.0	Multivides veida ID. Piezīme: Lai konfigurētu vairākus multivides veidus, izveidojiet atsevišķu informācijas paneļa logrīka lauka objektu katram multivides veidam, īpašuma nosaukumā izmantojot secīgi palielinātu numuru.
Statuss	0	statuses.0	0 - Notiek; 1 - Nosūtīts/izpildīts; 2 - Neizdevās. Piezīme: Lai konfigurētu vairākas vērtības, izveidojiet atsevišķu informācijas paneļa logrīka lauka objektu katrai vērtībai, īpašuma nosaukumā izmantojot secīgi palielinātu numuru.
Meklēšanas virkne	1	message	Jebkura virknes vērtība.
Laika periods	1	time_period._referenceDASHBOARD._timeperiod	`DASHBOARD._timeperiod` - iestatīt informācijas paneļa laika perioda atlasītāju kā datu avotu; `ABCDE._timeperiod` - iestatīt saderīgu logrīku (ar tā *Atsauce* parametru iestatītu uz "`ABCDE`") kā datu avotu. Noklusējums: `DASHBOARD._timeperiod` Varat arī iestatīt laika periodu 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 `time_period.to` ir iestatīts
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 `time_period.from` ir iestatīts
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 - Adresāts (augošā secībā); 12 - Adresāts (dilstošā secībā).

Parametrs	tips	nosaukums	vērtība
Rādīt rindas	0	show_lines	Iespējamās vērtības ir 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 attiecīgā informācijas paneļa logrīka lauku objektu iespējamās rekvizītu vērtības ļauj konfigurēt *Pulksteņa* logrīku `dashboard.create` un `dashboard.update` metodēs.

Attention:
Logrīka `fields` rekvizīti informācijas paneļa izveides vai atjaunināšanas laikā netiek validēti. Tas ļauj lietotājiem modificēt iebūvētos logrīkus un izveidot pielāgotus logrīkus, taču rada arī risku izveidot vai atjaunināt logrīkus nepareizi. Lai nodrošinātu veiksmīgu *Pulksteņa* logrīka izveidi vai atjaunināšanu, skatiet tālākajās 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 attiecīgo informācijas paneļa logrīka lauku objektu iespējamās rekvizītu vērtības ļauj konfigurēt logrīku *Atklāšanas statuss* `dashboard.create` un `dashboard.update` metodēs.

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 attiecīgajiem informācijas paneļa logrīka lauka objektiem pieejamās rekvizītu vērtības ļauj konfigurēt logrīku *Iecienītākie grafiki*, izmantojot `dashboard.create` un `dashboard.update` metodes.

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 attiecīgo informācijas paneļa logrīka lauku objektu iespējamās rekvizītu vērtības ļauj konfigurēt logrīku *Iecienītās kartes*, izmantojot `dashboard.create` un `dashboard.update` metodes.

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 informācijas paneļa logrīka lauka objektiem ļauj konfigurēt *Mērinstrumenta* logrīku `dashboard.create` un `dashboard.update` metodēs.

Attention:
Logrīka `fields` īpašības informācijas paneļa izveides vai atjaunināšanas laikā netiek validētas. Tas ļauj lietotājiem modificēt iebūvētos logrīkus un izveidot pielāgotus logrīkus, taču rada arī risku nepareizi izveidot vai atjaunināt logrīkus. Lai nodrošinātu veiksmīgu *Mērinstrumenta* logrīka izveidi vai atjaunināšanu, skatiet tālākajās 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.

Papildu konfigurācija

Gauge logrīkam tiek atbalstīti šādi papildu konfigurācijas parametri.

Note:
Skaitlis rekvizīta *Thresholds* nosaukumā (piemēram, thresholds.0.color) norāda sliekšņa vietu augošā secībā sakārtotā sarakstā. 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, tostarp makrosi. Atbalstītie makrosi: `{HOST.*}`, `{ITEM.*}`, `{INVENTORY.*}`, lietotāja makrosi. Noklusējums: `{ITEM.NAME}`.
Izmērs	0	desc_size	Iespējamās vērtības ir no 1 līdz 100. Noklusējums: 15.
Vertikālais novietojums	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			
Decimāldaļas	0	decimal_places	Iespējamās vērtības ir no 1 līdz 10. Noklusējums: 2.
Izmērs	0	value_size	Iespējamās vērtības ir 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 ir 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”
Novietojums	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 tam iestatīta kāda no tālāk norādītajām 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 ir no 1 līdz 100. Noklusējums: 20.
Rādītājs			
Krāsa	1	needle_color	Heksadecimāls krāsas kods (piemēram, FF0000). Noklusējums: "" (tukšs). Parametra darbība: - *atbalstīts*, ja informācijas paneļa logrīka lauka objektam *Rādīt* ir iestatīta vērtība “Vērtības loks” vai *Rādīt loku* 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 informācijas paneļa logrīka lauka objektam *Rādīt* ir iestatīta vērtība “Vērtības loks” vai *Rādīt loku* ir iestatīts uz “Iespējots”
Izmērs	0	scale_size	Iespējamās vērtības ir no 1 līdz 100. Noklusējums: 15. Parametra darbība: - *atbalstīts*, ja informācijas paneļa logrīka lauka objektam *Rādīt* ir iestatīta vērtība “Vērtības loks” vai *Rādīt loku* ir iestatīts uz “Iespējots”
Decimāldaļas	0	scale_decimal_places	Iespējamās vērtības ir no 1 līdz 10. Noklusējums: 0. Parametra darbība: - *atbalstīts*, ja informācijas paneļa logrīka lauka objektam *Rādīt* ir iestatīta vērtība “Vērtības loks” vai *Rādīt loku* 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 ir iestatīti *Sliekšņi* un informācijas paneļa logrīka lauka objektam *Rādīt* ir iestatīta vērtība “Vērtības loks” vai *Rādīt loku* ir iestatīts uz “Iespējots”
Rādīt loku	0	th_show_arc	0 - *(noklusējums)* Atspējots; 1 - Iespējots. Parametra darbība: - *atbalstīts*, ja ir iestatīti *Sliekšņi*
Loka izmērs	0	th_arc_size	Iespējamās vērtības ir no 1 līdz 100. Noklusējums: 5. Parametra darbība: - *atbalstīts*, ja *Rādīt loku* 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 attiecīgo informācijas paneļa logrīka lauku objektu iespējamās rekvizītu vērtības ļauj konfigurēt *Geomap* logrīku `dashboard.create` un `dashboard.update` metodēs.

Attention:
Logrīka `fields` rekvizīti informācijas paneļa izveides vai atjaunināšanas laikā netiek validēti. Tas ļauj lietotājiem modificēt iebūvētos logrīkus un izveidot pielāgotus logrīkus, taču rada arī risku izveidot vai atjaunināt logrīkus nepareizi. Lai nodrošinātu veiksmīgu *Geomap* logrīka izveidi vai atjaunināšanu, skatiet tālāk tabulās aprakstīto parametru darbību.

Parametri

Tālāk norādītie parametri tiek atbalstīti *Ģeokartes* logrīkam.

Parameter	type	name	value
Atjaunināšanas intervāls	0	rf_rate	0 - Bez atjaunināšanas; 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 atsevišķu informācijas paneļa logrīka lauka objektu katrai hostu grupai, lauka nosaukuma rekvizītā izmantojot palielinātu numuru. Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes informācijas panelī.

Parameter	type	name	value
Hostu grupas (logrīks)	1	groupids._reference	Hostu grupas ID vietā: `ABCDE._hostgroupids` - iestatiet saderīgu logrīku (kura *Atsauces* parametram ir iestatīta vērtība "`ABCDE`") kā hostu grupu datu avotu. Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes informācijas panelī.
Hosti	3	hostids.0	Hosta ID. Piezīme: Lai konfigurētu vairākus hostus, izveidojiet atsevišķu informācijas paneļa logrīka lauka objektu katram hostam, lauka nosaukuma rekvizītā izmantojot palielinātu numuru. Ja ir vairāki hosti, parametram *Hostu grupas* jābūt vai nu vispār nekonfigurētam, vai konfigurētam ar vismaz vienu hostu grupu, kurai pieder konfigurētie hosti. Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes informācijas panelī.
Hosti (logrīks/informācijas panelis)	1	hostids._reference	Hosta ID vietā: `DASHBOARD.hostids` - iestatiet informācijas paneļa hostu atlasītāju kā hostu datu avotu; `ABCDE._hostids` - iestatiet saderīgu logrīku (kura *Atsauces* parametram ir iestatīta vērtība "`ABCDE`") kā hostu datu avotu. Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes informācijas panelī.
Tagi			
Novērtēšanas tips	0	evaltype	0 - *(noklusējums)* Un/Vai; 2 - Vai. Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes informācijas panelī.
Taga nosaukums	1	tags.0.tag	Jebkura virknes vērtība. Piezīme: Skaitlis rekvizīta nosaukumā norāda taga secību tagu novērtēšanas sarakstā. Parametra darbība: - *obligāts*, konfigurējot *Tagus* Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes informācijas panelī.
Operators	0	tags.0.operator	0 - Satur; 1 - Ir vienāds ar; 2 - Nesatur; 3 - Nav vienāds ar; 4 - Pastāv; 5 - Nepastāv. Piezīme: Skaitlis rekvizīta nosaukumā norāda taga secību tagu novērtēšanas sarakstā. Parametra darbība: - *obligāts*, konfigurējot *Tagus* Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes informācijas panelī.

Parameter	type	name	value
Taga vērtība	1	tags.0.value	Jebkura virknes vērtība. Piezīme: Skaitlis rekvizīta nosaukumā norāda taga secību tagu novērtēšanas sarakstā. Parametra darbība: - *obligāts*, konfigurējot *Tagus* Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes informācijas panelī.
Sākotnējais skats	1	default_view	Ar komatiem atdalīti *platuma grādi*, *garuma grādi*, *tālummaiņas līmenis* (*neobligāts*, iespējamās vērtības ir no 0 līdz 30). Piemērs: `40.6892494,-74.0466891,10`.
Grupēšana			
Grupēšanas režīms	0	clustering_mode	0 - *(noklusējums)* Automātiski; 1 - Tālummaiņas līmenis.
Grupēšanas tālummaiņas līmenis	0	clustering_zoom_level	Iespējamās vērtības ir no 0 līdz 30. Noklusējums: 0. Parametra darbība: - *atbalstīts*, ja *Grupēšanas režīms* ir iestatīts uz "Tālummaiņas līmenis"
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 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 rekvizītu vērtības attiecīgajiem informācijas paneļa logrīka lauku objektiem ļauj konfigurēt *Graph* logrīku `dashboard.create` un `dashboard.update` metodēs.

Attention:
Logrīka `fields` rekvizīti informācijas paneļa izveides vai atjaunināšanas laikā netiek validēti. Tas ļauj lietotājiem modificēt iebūvētos logrīkus un izveidot pielāgotus logrīkus, taču rada arī risku izveidot vai atjaunināt logrīkus nepareizi. Lai nodrošinātu veiksmīgu *Graph* logrīka izveidi vai atjaunināšanu, skatiet tālākajās 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

Datu kopas konfigurēšanai tiek atbalstīti tālāk norādītie parametri.

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 informācijas panelī, jāiestata tikai veidnē konfigurētie vienumi. Piezīme: Lai konfigurētu vairākus vienumus, izveidojiet atsevišķu informācijas paneļa logrīka lauka objektu katram vienumam, rekvizīta nosaukumā izmantojot palielinātu skaitli. Parametra darbība: - *obligāts*, ja *Datu kopas tips* ir iestatīts uz "Vienumu saraksts" un *Vienumi (logrīks)* nav iestatīti
Vienumi (logrīks)	1	ds.0.itemids.0._reference	Vienuma ID vietā: `ABCDE._itemid` - iestatiet saderīgu logrīku (ar tā *Atsauce* parametru, kas iestatīts uz "`ABCDE`") kā vienumu datu avotu. Piezīme: Lai konfigurētu vairākus logrīkus, izveidojiet atsevišķu informācijas paneļa logrīka lauka objektu katram logrīkam, rekvizīta nosaukumā izmantojot palielinātu skaitli. Parametra darbība: - *obligāts*, ja *Datu kopas tips* ir iestatīts uz "Vienumu saraksts" un *Vienumi* nav iestatīti
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 informācijas panelī.
Vienumu šabloni	1	ds.0.items.0	Vienuma nosaukums vai šablons (piemēram, "*: Apstrādāto *vērtību skaits sekundē"). Konfigurējot logrīku veidnes informācijas panelī, jāiestata tikai veidnē konfigurēto vienumu šabloni. 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 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 - Pakāpieni; 3 - Josla.
Sakrauts	0	ds.0.stacked	0 - *(noklusējums)* Atspējots; 1 - Iespējots. Parametra darbība: - *atbalstīts*, ja *Zīmēšana* ir iestatīta uz “Līnija”, “Pakāpieni” vai “Josla”
Platums	0	ds.0.width	Iespējamās vērtības ir no 1 līdz 10. Noklusējums: 1. Parametra darbība: - *atbalstīts*, ja *Zīmēšana* ir iestatīta uz “Līnija” vai “Pakāpieni”
Punkta izmērs	0	ds.0.pointsize	Iespējamās vērtības ir no 1 līdz 10. Noklusējums: 3. Parametra darbība: - *atbalstīts*, ja *Zīmēšana* ir iestatīta uz “Punkti”
Caurspīdīgums	0	ds.0.transparency	Iespējamās vērtības ir no 1 līdz 10. Noklusējums: 5.
Aizpildījums	0	ds.0.fill	Iespējamās vērtības ir no 1 līdz 10. Noklusējums: 3. Parametra darbība: - *atbalstīts*, ja *Zīmēšana* ir iestatīta uz “Līnija” vai “Pakāpieni”
Trūkstošie dati	0	ds.0.missingdatafunc	0 - *(noklusējums)* Nekas; 1 - Savienot; 2 - Uzskatīt par 0; 3 - Pēdējais zināmais. Parametra darbība: - *atbalstīts*, ja *Zīmēšana* ir iestatīta uz “Līnija” vai “Pakāpieni”
Ignorēt hostu	1	ds.0.override_hostid.ABCDE.name	`ABCDE._hostid` - iestatiet saderīgu logrīku (ar tā “Atsauce” parametru, kas iestatīts uz “`ABCDE`”) kā hostu datu avotu; `DASHBOARD._hostid` - iestatiet informācijas paneļa hostu atlasītāju kā hostu datu avotu. Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes informācijas 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` u. c.). 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 u. c.). 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 tagi			
Novērtēšanas tips	0	ds.0.item_tags_evaltype	0 - *(noklusējums)* Un/Or; 2 - Or.
Taga nosaukums	1	ds.0.item_tags.0.tag	Jebkura virknes vērtība. Piezīme: Skaitlis rekvizīta nosaukumā norāda taga secību tagu novērtēšanas sarakstā. Parametra darbība: - *obligāts*, ja tiek konfigurēti *Vienumu tagi*
Operators	0	ds.0.item_tags.0.operator	0 - Satur; 1 - Ir vienāds ar; 2 - Nesatur; 3 - Nav vienāds ar; 4 - Pastāv; 5 - Nepastāv. Piezīme: Skaitlis rekvizīta nosaukumā norāda taga secību tagu novērtēšanas sarakstā. Parametra darbība: - *obligāts*, ja tiek konfigurēti *Vienumu tagi*
Taga vērtība	1	ds.0.item_tags.0.value	Jebkura virknes vērtība. Piezīme: Skaitlis rekvizīta nosaukumā norāda taga secību tagu novērtēšanas sarakstā. Parametra darbība: - *obligāts*, ja tiek konfigurēti *Vienumu tagi*

Attēlošanas opcijas

Tālāk norādītie parametri tiek atbalstīti, konfigurējot *Attēlošanas opcijas*.

Parameter	type	name	value
Vēsturisko datu atlase	0	source	0 - *(noklusējums)* Automātiski; 1 - Vēsture; 2 - Tendences.
Vienkāršie trigeri	0	simple_triggers	0 - *(noklusējums)* Atspējots; 1 - Iespējots.
Darba laiks	0	working_time	0 - *(noklusējums)* Atspējots; 1 - Iespējots.
Hosts nosaukumi etiķetēs	0	show_hostnames	0 - *(noklusējums)* Automātiski; 1 - Rādīt; 2 - Paslēpt. Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes informācijas panelī.
Procentiles līnija (kreisā)			
Statuss	0	percentile_left	0 - *(noklusējums)* Atspējots; 1 - Iespējots. Parametra darbība: - *atbalstīts*, ja *Y ass* (konfigurācijā *Datu kopa*) ir iestatīta uz "Kreisā"
Vērtība	0	percentile_left_value	Iespējamās vērtības ir no 1 līdz 100. Parametra darbība: - *atbalstīts*, ja *Y ass* (konfigurācijā *Datu kopa*) ir iestatīta uz "Kreisā"
Procentiles līnija (labā)			
Statuss	0	percentile_right	0 - *(noklusējums)* Atspējots; 1 - Iespējots. Parametra darbība: - *atbalstīts*, ja *Y ass* (konfigurācijā *Datu kopa*) ir iestatīta uz "Labā"
Vērtība	0	percentile_right_value	Iespējamās vērtības ir no 1 līdz 100. Parametra darbība: - *atbalstīts*, ja *Y ass* (konfigurācijā *Datu kopa*) ir iestatīta uz "Labā"

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

Leģendas konfigurēšanai tiek atbalstīti šādi parametri.

Parametrs	tips	nosaukums	vērtība
Rādīt leģendu	0	legend	0 - Atspējots; 1 - *(noklusējums)* Iespējots.
Rādīt min./vid./maks.	0	legend_statistic	0 - *(noklusējums)* Atspējots; 1 - Iespējots. Parametra darbība: - *atbalstīts*, ja *Rādīt leģendu* ir iestatīts uz “Iespējots”

Parametrs	tips	nosaukums	vērtība
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	Iespējamās vērtības ir no 1 līdz 10. Noklusējums: 1. Parametra darbība: - *atbalstīts*, ja *Rādīt leģendu* ir iestatīts uz "Iespējots"
Kolonnu skaits	0	legend_columns	Iespējamās vērtības ir no 1 līdz 4. Noklusējums: 4. Parametra darbība: - *atbalstīts*, ja *Rādīt leģendu* ir iestatīts uz "Iespējots" un *Rādīt min./vid./maks.* 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*

Pārrakstīšanas noteikumi

Pārrakstīšanas noteikumu konfigurēšanai tiek atbalstīti šādi parametri.

Note:
Pirmais skaitlis rekvizīta 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 raksti	1	or.0.hosts.0	Hosta nosaukums vai raksts (piemēram, `Zabbix*`). Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes informācijas panelī. Parametra darbība: - *obligāts*, ja tiek konfigurēti *pārrakstīšanas noteikumi*
Vienumu raksti	1	or.0.items.0	Vienuma nosaukums vai raksts (piemēram, `*: Number of processed *values per second`). Konfigurējot logrīku veidnes informācijas panelī, jāiestata tikai to vienumu raksti, kuri konfigurēti veidnē. Parametra darbība: - *obligāts*, ja tiek konfigurēti *pārrakstīšanas noteikumi*
Pamatkrāsa	1	or.0.color	Heksadecimāls krāsas kods (piemēram, `FF0000`).
Platums	0	or.0.width	Iespējamās vērtības ir no 1 līdz 10.
Zīmēšana	0	or.0.type	0 - Līnija; 1 - Punkti; 2 - Pakāpieni; 3 - Josla.

Parametrs	type	name	value
Caurspīdīgums	0	or.0.transparency	Iespējamās vērtības ir no 1 līdz 10.
Aizpildījums	0	or.0.fill	Iespējamās vērtības ir no 1 līdz 10.
Punkta izmērs	0	or.0.pointsize	Iespējamās vērtības ir no 1 līdz 10.
Trūkstošie dati	0	or.0.missingdatafunc	0 - Nav; 1 - Savienoti; 2 - Uzskatīt par 0; 3 - Pēdējie zināmie.
Y ass	0	or.0.axisy	0 - Kreisā; 1 - Labā.
Apgriezt vērtības	0	or.0.invert_values	0 - (noklusējums) Atspējots; 1 - Iespējots.
Laika nobīde	1	or.0.timeshift	Derīga laika virkne (piemēram, 3600, 1h u. c.). 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 attiecīgajiem informācijas paneļa logrīka lauku objektiem iespējamās rekvizītu vērtības ļauj konfigurēt logrīku *Graph (classic)* metodēs `dashboard.create` un `dashboard.update`.

Attention:
Logrīka `fields` rekvizīti informācijas paneļa izveides vai atjaunināšanas laikā netiek validēti. Tas ļauj lietotājiem modificēt iebūvētos logrīkus un izveidot pielāgotus logrīkus, taču rada arī risku izveidot vai atjaunināt logrīkus nepareizi. Lai nodrošinātu veiksmīgu *Graph (classic)* logrīka izveidi vai atjaunināšanu, skatiet tālākajās 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 attiecīgo informācijas paneļa logrīka lauku objektu iespējamās rekvizītu vērtības ļauj konfigurēt logrīku *Grafika prototips* `dashboard.create` un `dashboard.update` metodēs.

Attention:
Logrīka `fields` rekvizīti informācijas paneļa izveides vai atjaunināšanas laikā netiek validēti. Tas ļauj lietotājiem modificēt iebūvētos logrīkus un izveidot pielāgotus logrīkus, taču rada arī risku izveidot vai atjaunināt logrīkus nepareizi. Lai nodrošinātu veiksmīgu logrīka *Grafika prototips* izveidi vai atjaunināšanu, skatiet tālāk tabulās aprakstīto parametru darbību.

Parametri

Tālāk norādīti parametri, ko atbalsta *Graph prototype* logrīks.

Parameter	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.
Graph prototype	7	graphid.0	Graph prototype ID. Parametra darbība: - *obligāts*, ja *Avots* ir iestatīts uz ”Graph prototype”
Item prototype	5	itemid.0	Item prototype ID. Parametra darbība: - *obligāts*, ja *Avots* ir iestatīts uz ”Simple graph prototype”

Parameter	type	name	value
Laika periods	1	time_period._referenceDASHBOARD	`DASHBOARD._timeperiod` - iestatīt informācijas paneļa laika perioda atlasītāju kā datu avotu; `ABCDE._timeperiod` - iestatīt saderīgu logrīku (ar tā *Atsauce* parametru, kas iestatīts uz "`ABCDE`") kā datu avotu. Noklusējums: `DASHBOARD._timeperiod` Varat arī iestatīt laika periodu tikai parametros *No* un *Līdz*.
No	1	time_period.from	Derīga laika virkne absolūtajā (`YYYY-MM-DD hh:mm:ss`) vai relatīvajā 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 absolūtajā (`YYYY-MM-DD hh:mm:ss`) vai relatīvajā 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`
Rādīt leģendu	0	show_legend	0 - Atspējots; 1 - *(noklusējums)* Iespējots.
Ignorēt hostu	1	override_hostid._referenceABCDE	`ABCDE._hostid` - iestatīt saderīgu logrīku (ar tā *Atsauce* parametru, 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.
Kolonnas	0	columns	Iespējamās vērtības ir no 1 līdz 24. Noklusējums: 2.
Rindas	0	rows	Iespējamās vērtības ir no 1 līdz 16. Noklusējums: 1.
Atsauce	1	reference	Jebkura 5 rakstzīmju virkne (piemēram, `ABCDE` vai `JBPNL`). Šai vērtībai ir jābūt unikālai informācijas 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 attiecīgo informācijas paneļa logrīka lauku objektu iespējamās rekvizītu vērtības ļauj konfigurēt *Honeycomb* logrīku `dashboard.create` un `dashboard.update` metodēs.

Attention:
Logrīka `fields` rekvizīti informācijas paneļa izveides vai atjaunināšanas laikā netiek validēti. Tas ļauj lietotājiem modificēt iebūvētos logrīkus un izveidot pielāgotus logrīkus, taču rada arī risku izveidot vai atjaunināt logrīkus nepareizi. Lai nodrošinātu veiksmīgu *Honeycomb* logrīka izveidi vai atjaunināšanu, skatiet tālākajās tabulās aprakstīto parametru darbību.

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

Honeycomb logrīkam tiek atbalstīti tālāk norādītie papildu konfigurācijas parametri.

Note:
Skaitlis rekvizīta *Thresholds* nosaukumā (piemēram, thresholds.0.color) norāda sliekšņa vietu augošā secībā sakārtotā sarakstā. 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, tostarp makro. Atbalstītie makro: {HOST.*}, {ITEM.*}, {INVENTORY.*}, lietotāja makro. Noklusējums: {HOST.NAME} Parametra darbība: - *atbalstīts*, ja *Tips* ir iestatīts uz "Teksts"
Cipari aiz komata	0	primary_label_decimal_places	Iespējamās vērtības ir no 0 līdz 6. Noklusējums: 2. Parametra darbība: - *atbalstīts*, ja *Tips* ir iestatīts uz "Vērtība"
Izmērs (tips)	0	primary_label_size_type	0 - *(noklusējums)* Automātisks; 1 - Pielāgots.
Izmērs	0	primary_label_size	Iespējamās vērtības ir 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: balstīts uz Settings object un User object theme: 1F2C33 vērtībai "blue-theme", "blue-classic-theme" vai "hc-light"; EEEEEE vērtībai "dark-theme", "dark-classic-theme" vai "hc-dark"; E5F1FF vērtībai "dark-blue-theme".
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 tas iestatīts uz kādu no tālāk norādītajām ar laiku saistītajām mērvienībām: unixtime, uptime, s.
Sekundārā etiķete			
Tips	0	secondary_label_type	0 - Teksts; 1 - *(noklusējums)* Vērtība.

Parametrs	type	name	value
Teksts	1	secondary_label	Jebkura virknes vērtība, tostarp makro. Atbalstītie makro: {HOST.*}, {ITEM.*}, {INVENTORY.*}, lietotāja makro. Noklusējums: {{ITEM.LASTVALUE}.fmtnum(2)} Parametra darbība: - *atbalstīts*, ja *Tips* ir iestatīts uz "Teksts"
Cipari aiz komata	0	secondary_label_decimal_places	Iespējamās vērtības ir no 0 līdz 6. Noklusējums: 2. Parametra darbība: - *atbalstīts*, ja *Tips* ir iestatīts uz "Vērtība"
Izmērs (tips)	0	secondary_label_size_type	0 - *(noklusējums)* Automātisks; 1 - Pielāgots.
Izmērs	0	secondary_label_size	Iespējamās vērtības ir 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: balstīts uz Settings object un User object theme: 1F2C33 vērtībai "blue-theme", "blue-classic-theme" vai "hc-light"; EEEEEE vērtībai "dark-theme", "dark-classic-theme" vai "hc-dark"; E5F1FF vērtībai "dark-blue-theme".
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 tas iestatīts uz kādu no tālāk norādītajām ar laiku saistītajām mērvienībām: unixtime, uptime, s.
Fona krāsa			
Fona krāsa	1	bg_color	Heksadecimāls krāsas kods (piemēram, FF0000). Noklusējums: balstīts uz Settings object un User object theme: D9E7ED vērtībai "blue-theme" un "blue-classic-theme"; 3D5059 vērtībai "dark-theme" un "dark-classic-theme"; 004FA8 vērtībai "dark-blue-theme"; AAD7E9 vērtībai "hc-light"; 335463 vērtībai "hc-dark".
Sliekšņi			

Parametrs	type	name	value
Krāsu interpolācija	0	interpolation	0 - Atspējota; 1 - *(noklusējums)* Iespējota.
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 attiecīgo informācijas paneļa logrīka lauku objektu iespējamās rekvizītu vērtības ļauj konfigurēt logrīku *Hosts pieejamība*, izmantojot `dashboard.create` un `dashboard.update` metodes.

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 rekvizītu vērtības attiecīgajiem informācijas paneļa logrīka lauku objektiem ļauj konfigurēt *Hosts kartītes* logrīku `dashboard.create` un `dashboard.update` metodēs.

Attention:
Logrīka `fields` rekvizīti informācijas paneļa izveides vai atjaunināšanas laikā netiek validēti. Tas ļauj lietotājiem modificēt iebūvētos logrīkus un izveidot pielāgotus logrīkus, taču rada arī risku izveidot vai atjaunināt logrīkus nepareizi. Lai nodrošinātu veiksmīgu *Hosts kartītes* logrīka izveidi vai atjaunināšanu, skatiet tālāk 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.

Parametrs	type	name	value
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.
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 attiecīgajiem informācijas paneļa logrīka lauka objektiem iespējamās rekvizītu vērtības ļauj konfigurēt logrīku *Hosts navigator* `dashboard.create` un `dashboard.update` metodēs.

Attention:
Logrīka `fields` rekvizīti informācijas paneļa izveides vai atjaunināšanas laikā netiek validēti. Tas ļauj lietotājiem modificēt iebūvētos logrīkus un izveidot pielāgotus logrīkus, taču rada arī risku izveidot vai atjaunināt logrīkus nepareizi. Lai nodrošinātu veiksmīgu logrīka *Hosts navigator* izveidi vai atjaunināšanu, lūdzu, skatiet tālākajās tabulās aprakstīto parametru darbību.

Parametri

Tālāk norādītie parametri tiek atbalstīti *Hosts navigator* 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.
Hosts grupas	2	groupids.0	Hosts grupas ID. Piezīme: Lai konfigurētu vairākas hosts grupas, izveidojiet atsevišķu informācijas paneļa logrīka lauka objektu katrai hosts grupai, īpašuma nosaukumā izmantojot palielinātu numuru. Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes informācijas panelī.

Parametrs	type	name	value
Hosts grupas (logrīks)	1	groupids._reference	Hosts grupas ID vietā: `ABCDE._hostgroupids` - iestatiet saderīgu logrīku (ar tā *Reference* parametru, kas iestatīts uz "ABCDE") kā hosts grupu datu avotu. Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes informācijas panelī.
Hosts modeļi	1	hosts.0	Hosts nosaukums vai modelis. Piezīme: Lai konfigurētu vairākus hosts modeļus, izveidojiet atsevišķu informācijas paneļa logrīka lauka objektu katram hosts modelim, īpašuma nosaukumā izmantojot palielinātu numuru. Ja ir vairāki hosts modeļi, parametrs *Hosts grupas* nedrīkst būt konfigurēts vai arī tam jābūt konfigurētam ar vismaz vienu hosts grupu, kurai pieder hosts, kas atbilst konfigurētajiem hosts modeļiem. Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes informācijas panelī.
Hosts statuss	0	status	-1 - *(noklusējums)* Jebkurš; 0 - Iespējots; 1 - Atspējots. Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes informācijas panelī.
Hosts tagi			
Novērtēšanas tips	0	host_tags_evaltype	0 - *(noklusējums)* Un/Vai; 2 - Vai. Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes informācijas panelī.
Taga nosaukums	1	host_tags.0.tag	Jebkura virknes vērtība. Piezīme: Skaitlis īpašuma nosaukumā norāda taga secību tagu novērtēšanas sarakstā. Parametra darbība: - *obligāts*, ja tiek konfigurēti *Hosts tagi* Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes informācijas panelī.
Operators	0	host_tags.0.operator	0 - Satur; 1 - Ir vienāds ar; 2 - Nesatur; 3 - Nav vienāds ar; 4 - Pastāv; 5 - Nepastāv. Piezīme: Skaitlis īpašuma nosaukumā norāda taga secību tagu novērtēšanas sarakstā. Parametra darbība: - *obligāts*, ja tiek konfigurēti *Hosts tagi* Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes informācijas panelī.

Parametrs	type	name	value
Taga vērtība	1	host_tags.0.value	Jebkura virknes vērtība. Piezīme: Skaitlis īpašuma nosaukumā norāda taga secību tagu novērtēšanas sarakstā. Parametra darbība: - *obligāts*, ja tiek konfigurēti *Hosts tagi* Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes informācijas panelī.
Nopietnības pakāpe	0	severities.0	0 - Nav klasificēts; 1 - Informācija; 2 - Brīdinājums; 3 - Vidēja; 4 - Augsta; 5 - Katastrofa. Noklusējums: tukšs (visi iespējoti). Piezīme: Lai konfigurētu vairākas vērtības, izveidojiet atsevišķu informācijas paneļa logrīka lauka objektu katrai vērtībai, īpašuma nosaukumā izmantojot palielinātu numuru.
Rādīt uz-turēšanā es-ošos hosts	0	maintenance	0 - *(noklusējums)* Atspējots; 1 - Iespējots.
Rādīt prob-lē-mas	0	show_problems	0 - Visas; 1 - *(noklusējums)* Neapslāpētās; 2 - Nevienu.
Grupēt pēc			
Atribūts	0	group_by.0.attribute	0 - Hosts grupa; 1 - Taga vērtība; 2 - Nopietnības pakāpe. Piezīme: Skaitlis īpašuma 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: Skaitlis īpašuma nosaukumā norāda grupēšanas atribūtu, kas iestatīts parametrā *Atribūts*. Parametra darbība: - *obligāts*, ja tiek konfigurēts *Grupēt pēc* un *Atribūts* ir iestatīts uz "Taga vērtība"
Hosts ier-obežo-jums	0	show_lines	Iespējamās vērtības ir no 1 līdz 9999. Noklusējums: 100. Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes informācijas panelī.

Parametrs	type	name	value
Reference	1	reference	Jebkura virknes vērtība, kas sastāv no 5 rakstzīmēm (piemēram, `ABCDE` vai `JBPNL`). Šai vērtībai ir jābūt unikālai informācijas 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 attiecīgo informācijas paneļa logrīka lauku objektu iespējamās rekvizītu vērtības ļauj konfigurēt logrīku *Vienumu kartīte* `dashboard.create` un `dashboard.update` metodēs.

Attention:
Logrīka `fields` rekvizīti informācijas paneļa izveides vai atjaunināšanas laikā netiek validēti. Tas ļauj lietotājiem modificēt iebūvētos logrīkus un izveidot pielāgotus logrīkus, taču rada arī risku izveidot vai atjaunināt logrīkus nepareizi. Lai nodrošinātu veiksmīgu logrīka *Vienumu kartīte* izveidi vai atjaunināšanu, skatiet tālākajās 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
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.

Sparkline

Ja *Rādīt* ir iestatīts uz "Jaunākie dati", tiek atbalstīti šādi parametri.

Parametrs	tips	nosaukums	vērtība
Platums	0	sparkline.width	Iespējamās vērtības ir no 0 līdz 10. Noklusējums: 1.
Aizpildījums	0	sparkline.fill	Iespējamās vērtības ir no 0 līdz 10. Noklusējums: 3.
Krāsa	1	sparkline.color	Heksadecimāls krāsas kods (piemēram, `FF0000`). Noklusējums: `42A5F5`.

Parametrs	tips	nosaukums	vērtība
Laika periods	1	sparkline.time_period.from	DASHBOARD._timeperiod - iestatīt informācijas paneļa laika perioda atlasītāju kā datu avotu; ABCDE._timeperiod - iestatīt saderīgu logrīku (ar `reference` parametru, kas vienāds ar ABCDE) kā datu avotu. Noklusējums: "" (tukšs) Varat arī iestatīt laika periodu tikai parametros *No* un *Līdz*.
No	1	sparkline.time_period.from	Datuma laika virkne absolūtā (`YYYY-MM-DD hh:mm:ss`) vai relatīvā laika sintaksē (`now`, `now/d`, `now/w-1w` u. c.). Noklusējums: now-1h. Parametra darbība: - *atbalstīts*, ja *Laika periods* nav iestatīts - *obligāts*, ja ir iestatīts `sparkline.time_period.to`
Līdz	1	sparkline.time_period.to	Datuma laika virkne absolūtā (`YYYY-MM-DD hh:mm:ss`) vai relatīvā laika sintaksē (`now`, `now/d`, `now/w-1w` u. c.). Noklusējums: now. Parametra darbība: - *atbalstīts*, ja *Laika periods* nav iestatīts - *obligāts*, ja ir iestatīts `time_period.to`
Vēsturiskie 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 attiecīgo informācijas paneļa logrīka lauku objektu iespējamās rekvizītu vērtības ļauj konfigurēt *Vienumu vēstures* logrīku `dashboard.create` un `dashboard.update` metodēs.

Attention:
Logrīka `fields` rekvizīti informācijas paneļa izveides vai atjaunināšanas laikā netiek validēti. Tas ļauj lietotājiem modificēt iebūvētos logrīkus un izveidot pielāgotus logrīkus, taču rada arī risku izveidot vai atjaunināt logrīkus nepareizi. Lai nodrošinātu veiksmīgu *Vienumu vēstures* logrīka izveidi vai atjaunināšanu, skatiet tālākajās tabulās aprakstīto parametru darbību.

Parametri

Tālāk norādītie parametri tiek atbalstīti *Vienumu vēstures* logrīkā.

Parametrs	tips	nosaukums	vērtība
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.
Izkārtojums	0	layout	0 - *(noklusējums)* Horizontāls; 1 - Vertikāls.
Kolonnas (skatiet tālāk)			
Rādīt rindas	0	show_lines	Iespējamās vērtības ir no 1 līdz 100. Noklusējums: 25.
Ignorēt hostu	1	override_hostid._reference	`ABCDE._hostid` - iestatīt saderīgu logrīku (kura *Atsauce* parametram iestatīta vērtība "`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.
Papildu konfigurācija (skatiet tālāk)			
Atsauce	1	reference	Jebkura virknes vērtība, kas sastāv no 5 rakstzīmēm (piemēram, `ABCDE` vai `JBPNL`). Šai vērtībai ir jābūt unikālai informācijas panelī, kuram pieder logrīks. Parametra darbība: - *obligāts*

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*

Parametrs	type	name	value
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.

Ja konfigurētais *vienums* ir rakstzīmju, teksta vai žurnāla tipa vienums, tiek atbalstīti tālāk norādītie kolonnu parametri.

Parameter	type	name	value
Izcēlumi			
Izcēlums	1	columns.0.highlights.color	Heksadecimāls krāsas kods (piemēram, FF0000).
Slieksnis	1	columns.0.highlights.pattern	Jebkura regulārā izteiksme.
Rādīšana	0	columns.0.display	1 - *(noklusējums)* Kā ir; 4 - HTML; 5 - Viena rinda.
Viena rinda	0	columns.0.max_length	Iespējamās vērtības ir no 1 līdz 500. Noklusējums: 100. Parametra darbība: - *atbalstīts*, ja *Rādīšana* ir iestatīta uz “Viena rinda”
Izmantot vienplatuma fontu	0	columns.0.monospace_font	0 - *(noklusējums)* Izmantot noklusējuma fontu; 1 - Izmantot vienplatuma fontu.

Parameter	type	name	value
Rādīt vietējo laiku	0	columns.0.local_time	0 - *(noklusējums)* Rādīt laika zīmogu; 1 - Rādīt vietējo laiku. Parametra darbība: - *atbalstīts*, ja *vienums* ir iestatīts kā žurnāla tipa vienums un *Rādīt laika zīmogu* 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 laikspiedolu	0	show_timestamp	0 - *(noklusējums)* Atspējots; 1 - Iespējots.
Rādīt kolonnas galveni	0	show_column_header	0 - Izslēgts; 1 - Horizontāli; 2 - *(noklusējums)* Vertikāli.
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` 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 attiecīgo informācijas paneļa logrīka lauku objektu iespējamās rekvizītu vērtības ļauj konfigurēt *Vienumu navigatora* logrīku `dashboard.create` un `dashboard.update` metodēs.

Attention:
Logrīka `fields` rekvizīti informācijas paneļa izveides vai atjaunināšanas laikā netiek validēti. Tas ļauj lietotājiem modificēt iebūvētos logrīkus un izveidot pielāgotus logrīkus, taču rada arī risku izveidot vai atjaunināt logrīkus nepareizi. Lai nodrošinātu veiksmīgu *Vienumu navigatora* logrīka izveidi vai atjaunināšanu, skatiet tālāk tabulās aprakstīto parametru darbību.

Parametri

Logrīkam *Vienumu navigators* tiek atbalstīti šādi parametri.

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 atsevišķu informācijas paneļa logrīka lauka objektu katrai hostu grupai, īpašuma nosaukumā izmantojot palielinātu numuru. Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes informācijas panelī.
Hostu grupas (logrīks)	1	groupids._reference	Hostu grupas ID vietā: `ABCDE._hostgroupids` - iestatiet saderīgu logrīku (kura *Atsauce* parametrs ir iestatīts uz "`ABCDE`") kā hostu grupu datu avotu. Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes informācijas panelī.
Hosti	3	hostids.0	Hosta ID. Piezīme: Lai konfigurētu vairākus hostus, izveidojiet atsevišķu informācijas paneļa logrīka lauka objektu katram hostam, īpašuma nosaukumā izmantojot palielinātu numuru. Vairākiem hostiem parametrs *Hostu grupas* ir vai nu jāatstāj nekonfigurēts, vai arī jākonfigurē ar vismaz vienu hostu grupu, kurai pieder konfigurētie hosti. Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes informācijas panelī.
Hosti (logrīks/informācijas panelis)	1	hostids._reference	Hosta ID vietā: `DASHBOARD.hostid` - iestatiet informācijas paneļa hostu atlasītāju kā hostu datu avotu; `ABCDE._hostid` - iestatiet saderīgu logrīku (kura *Atsauce* parametrs ir iestatīts uz "`ABCDE`") kā hostu datu avotu. Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes informācijas panelī.
Hostu tagi			
Novērtēšanas tips	0	host_tags_evaltype	0 - *(noklusējums)* Un/Vai; 2 - Vai. Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes informācijas panelī.
Taga nosaukums	1	host_tags.0.tag	Jebkura virknes vērtība. Piezīme: Numurs īpašuma nosaukumā norāda taga secību tagu novērtēšanas sarakstā. Parametra darbība: - *obligāts*, ja tiek konfigurēti *Hostu tagi* Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes informācijas panelī.

Parametrs	type	name	value
Operators	0	host_tags.0.operator	0 - Satur; 1 - Ir vienāds ar; 2 - Nesatur; 3 - Nav vienāds ar; 4 - Pastāv; 5 - Nepastāv. Piezīme: Numurs īpašuma nosaukumā norāda taga secību tagu novērtēšanas sarakstā. Parametra darbība: - *obligāts*, ja tiek konfigurēti *Hostu tagi* Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes informācijas panelī.
Taga vērtība	1	host_tags.0.value	Jebkura virknes vērtība. Piezīme: Numurs īpašuma nosaukumā norāda taga secību tagu novērtēšanas sarakstā. Parametra darbība: - *obligāts*, ja tiek konfigurēti *Hostu tagi* Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes informācijas panelī.
Vienumu rak-sti	1	items.0	Vienuma nosaukums vai raksts. Piezīme: Lai konfigurētu vairākus vienumu rakstus, izveidojiet atsevišķu informācijas paneļa logrīka lauka objektu katram vienumu rakstam, īpašuma nosaukumā izmantojot palielinātu numuru.
Vienumu tagi			
Novērtēšanas tips	0	item_tags_evaltype	0 - *(noklusējums)* Un/Vai; 2 - Vai.
Taga nosaukums	1	item_tags.0.tag	Jebkura virknes vērtība. Piezīme: Numurs īpašuma nosaukumā norāda taga secību tagu novērtēšanas sarakstā. Parametra darbība: - *obligāts*, ja tiek konfigurēti *Vienumu tagi*
Operators	0	item_tags.0.operator	0 - Satur; 1 - Ir vienāds ar; 2 - Nesatur; 3 - Nav vienāds ar; 4 - Pastāv; 5 - Nepastāv. Piezīme: Numurs īpašuma nosaukumā norāda taga secību tagu novērtēšanas sarakstā. Parametra darbība: - *obligāts*, ja tiek konfigurēti *Vienumu tagi*
Taga vērtība	1	item_tags.0.value	Jebkura virknes vērtība. Piezīme: Numurs īpašuma nosaukumā norāda taga secību tagu novērtēšanas sarakstā. Parametra darbība: - *obligāts*, ja tiek konfigurēti *Vienumu tagi*

Parametrs	type	name	value
Stāvoklis	0	state	-1 - *(noklusējums)* Visi; 0 - Normāls; 1 - Netiek atbalstīts.
Rādīt problēmas	0	show_problems	0 - Visas; 1 - *(noklusējums)* Neapslāpētās; 2 - Nevienu.
Grupēt pēc			
Atribūts	0	group_by.0.attribute	0 - Hostu grupa; 1 - Hosta nosaukums; 2 - Hosta taga vērtība; 3 - Vienuma taga vērtība. Piezīme: Numurs īpašuma 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 īpašuma nosaukumā norāda grupēšanas atribūtu, kas iestatīts parametrā *Atribūts*. Parametra darbība: - *obligāts*, ja tiek konfigurēts *Grupēt pēc* un *Atribūts* ir iestatīts uz ”Hosta taga vērtība” vai ”Vienuma taga vērtība”
Vienumu ierobežojums	0	show_lines	Iespējamās vērtības ir no 1 līdz 9999. Noklusējums: 100.
Atsauce	1	reference	Jebkura virknes vērtība, kas sastāv no 5 rakstzīmēm (piemēram, `ABCDE` vai `JBPNL`). Šai vērtībai ir jābūt unikālai informācijas 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 informācijas paneļa logrīka lauku objektiem ļauj konfigurēt *Vienuma vērtība* logrīku `dashboard.create` un `dashboard.update` metodēs.

Attention:
Logrīka `fields` īpašības informācijas paneļa izveides vai atjaunināšanas laikā netiek validētas. Tas ļauj lietotājiem modificēt iebūvētos logrīkus un izveidot pielāgotus logrīkus, taču rada arī risku izveidot vai atjaunināt logrīkus nepareizi. Lai nodrošinātu veiksmīgu logrīka *Vienuma vērtība* izveidi vai atjaunināšanu, skatiet tālākajās tabulās aprakstīto parametru darbību.

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.
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

Ja *Rādīt* ir iestatīts uz “Apraksts”, tiek atbalstīti tālāk norādītie papildu konfigurācijas parametri.

Parametrs	type	name	value
Apraksts	1	description	Jebkura virknes vērtība, tostarp makro. Atbalstītie makro: {HOST.*}, {ITEM.*}, {INVENTORY.*}, lietotāja makro. Noklusējums: {ITEM.NAME}.
Horizontālais novietojums	0	desc_h_pos	0 - Pa kreisi; 1 - *(noklusējums)* Centrā; 2 - Pa labi.
Vertikālais novietojums	0	desc_v_pos	Diviem vai vairākiem elementiem (Apraksts, Vērtība, Laiks) nevar būt vienāds *Horizontālais novietojums* un *Vertikālais novietojums*. 0 - Augšā; 1 - Vidū; 2 - *(noklusējums)* Apakšā.
Izmērs	0	desc_size	Diviem vai vairākiem elementiem (Apraksts, Vērtība, Laiks) nevar būt vienāds *Horizontālais novietojums* un *Vertikālais novietojums*. Iespējamās vērtības ir 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

Ja *Rādīt* ir iestatīts uz “Vērtība”, tiek atbalstīti šādi papildu konfigurācijas parametri.

Parametrs	type	name	value
Cipari aiz komata			
Cipari aiz komata	0	decimal_places	Iespējamās vērtības ir no 1 līdz 10. Noklusējums: 2.
Izmērs	0	decimal_size	Iespējamās vērtības ir no 1 līdz 100. Noklusējums: 35.
Pozīcija			
Horizontālā pozīcija	0	value_h_pos	0 - pa kreisi; 1 - *(noklusējums)* centrā; 2 - pa labi.
Vertikālā pozīcija	0	value_v_pos	Diviem vai vairākiem elementiem (Apraksts, Vērtība, Laiks) nevar būt vienāda *Horizontālā pozīcija* un *Vertikālā pozīcija*. 0 - augšā; 1 - *(noklusējums)* vidū; 2 - apakšā.
Izmērs	0	value_size	Diviem vai vairākiem elementiem (Apraksts, Vērtība, Laiks) nevar būt vienāda *Horizontālā pozīcija* un *Vertikālā pozīcija*. Iespējamās vērtības ir no 1 līdz 100. Noklusējums: 45.
Treknraksts	0	value_bold	0 - atspējots; 1 - *(noklusējums)* iespējots.

Parametrs	type	name	value
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 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

Tiek atbalstīti tālāk norādītie papildu konfigurācijas parametri, ja *Rādīt* ir iestatīts uz “Laiks”.

Parametrs	type	name	value
Horizontālais novietojums	0	time_h_pos	0 - Pa kreisi; 1 - *(noklusējums)* Centrā; 2 - Pa labi. Diviem vai vairākiem elementiem (Apraksts, Vērtība, Laiks) nevar būt vienāds *Horizontālais novietojums* un *Vertikālais novietojums*.
Vertikālais novietojums	0	time_v_pos	0 - *(noklusējums)* Augšā; 1 - Vidū; 2 - Apakšā. Diviem vai vairākiem elementiem (Apraksts, Vērtība, Laiks) nevar būt vienāds *Horizontālais novietojums* un *Vertikālais novietojums*.
Izmērs	0	time_size	Iespējamās vērtības ir 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).

Parametrs	type	name	value
Izmaiņu indikatora ↕ krāsa	1	updown_color	Heksadecimāls krāsas kods (piemēram, FF0000). Noklusējums: "" (tukšs).

Sparkline

Ja *Rādīt* ir iestatīts uz "Sparkline", tiek atbalstīti tālāk norādītie papildu konfigurācijas parametri.

Parametrs	type	name	value
Platums	0	sparkline.width	Iespējamās vērtības ir no 0 līdz 10. Noklusējums: 1.
Aizpildījums	0	sparkline.fill	Iespējamās vērtības ir no 0 līdz 10. Noklusējums: 3.
Krāsa	1	sparkline.color	Heksadecimāls krāsas kods (piemēram, FF0000). Noklusējums: 42A5F5.
Laika periods	1	sparkline.time_period.from	DASHBOARD._timeperiod - iestatīt informācijas paneļa laika perioda atlasītāju kā datu avotu; ABCDE._timeperiod - iestatīt saderīgu logrīku (ar reference parametru, kas ir vienāds ar ABCDE) kā datu avotu. Noklusējums: DASHBOARD._timeperiod Varat arī iestatīt laika periodu 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 u. c.). 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 u. c.). Parametra darbība: - *obligāts*, ja ir iestatīts sparkline.time_period.from
Vēsturiskie 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 rekvizītu vērtības attiecīgajiem informācijas paneļa logrīka lauku objektiem ļauj konfigurēt logrīku *Karte* `dashboard.create` un `dashboard.update` metodēs.

Attention:
Logrīka `fields` rekvizīti informācijas paneļa izveides vai atjaunināšanas laikā netiek validēti. Tas ļauj lietotājiem modificēt iebūvētos logrīkus un izveidot pielāgotus logrīkus, taču rada arī risku izveidot vai atjaunināt logrīkus nepareizi. Lai nodrošinātu veiksmīgu logrīka *Karte* izveidi vai atjaunināšanu, skatiet tālāk 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

Parametrs		type	name	value
	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 informācijas paneļa logrīka lauku objektiem ļauj konfigurēt logrīku *Kartes navigācijas koks* `dashboard.create` un `dashboard.update` metodēs.

Attention:
Logrīka `fields` īpašības informācijas paneļa izveides vai atjaunināšanas laikā netiek validētas. Tas ļauj lietotājiem modificēt iebūvētos logrīkus un izveidot pielāgotus logrīkus, taču rada arī risku izveidot vai atjaunināt logrīkus nepareizi. Lai nodrošinātu veiksmīgu logrīka *Kartes navigācijas koks* izveidi vai atjaunināšanu, skatiet tālākajās tabulās aprakstīto parametru darbību.

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.

Parametrs	type	name	value
	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 attiecīgajiem informācijas paneļa logrīka lauku objektiem iespējamās rekvizītu vērtības ļauj konfigurēt *Sektoru diagrammas* logrīku `dashboard.create` un `dashboard.update` metodēs.

Attention:
Logrīka `fields` rekvizīti informācijas paneļa izveides vai atjaunināšanas laikā netiek validēti. Tas ļauj lietotājiem modificēt iebūvētos logrīkus un izveidot pielāgotus logrīkus, taču rada arī risku izveidot vai atjaunināt logrīkus nepareizi. Lai nodrošinātu veiksmīgu *Sektoru diagrammas* logrīka izveidi vai atjaunināšanu, skatiet tālākajās 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

Datu kopas konfigurēšanai tiek atbalstīti tālāk norādītie parametri.

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 informācijas panelī, jāiestata tikai veidnē konfigurētie vienumi. Piezīme: Lai konfigurētu vairākus vienumus, izveidojiet atsevišķu informācijas paneļa logrīka lauka objektu katram vienumam, rekvizīta nosaukumā palielinot skaitli. Parametra darbība: - *obligāts*, ja *Datu kopas tips* ir iestatīts uz "Vienumu saraksts" un *Vienumi (logrīks)* nav iestatīti
Vienumi (logrīks)	1	ds.0.itemids.0._reference	Vienuma ID vietā: `ABCDE._itemid` - iestatiet saderīgu logrīku (kura *Atsauces* parametrs ir iestatīts uz "`ABCDE`") kā vienumu datu avotu. Piezīme: Lai konfigurētu vairākus logrīkus, izveidojiet atsevišķu informācijas paneļa logrīka lauka objektu katram logrīkam, rekvizīta nosaukumā palielinot skaitli. Parametra darbība: - *obligāts*, ja *Datu kopas tips* ir iestatīts uz "Vienumu saraksts" un *Vienumi* nav iestatīti
Krāsa	1	ds.0.color.0	Heksadecimāls krāsas kods (piemēram, `FF0000`). Parametra darbība: - *obligāts*, ja *Datu kopas tips* ir iestatīts uz "Vienumu saraksts"

Parametrs	type	name	value
Vienuma tips	0	ds.0.type.0	0 - *(noklusējums)* Parasts; 1 - Kopsumma. Vērtību "Kopsumma" 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	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 informācijas panelī.
Vienumu šabloni	1	ds.0.items.0	Vienuma nosaukums vai šablons (piemēram, "*: Apstrādāto *vērtību skaits sekundē"). Konfigurējot logrīku veidnes informācijas panelī, jāiestata tikai veidnē konfigurēto vienumu šabloni. 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
Krāsu palete	0	ds.0.color_palette	Krāsu paletes indekss. Iespējamās vērtības ir no 0 līdz 11. Noklusējums: 0. Parametra darbība: - *atbalstīts*, ja *Datu kopas tips* ir iestatīts uz "Vienumu šabloni" un *Krāsa* nav iestatīta
Apkopošanas funkcija	0	ds.0.aggregate_function	1 - minimums; 2 - maksimums; 3 - vidējais; 4 - skaits; 5 - summa; 6 - pirmais; 7 - *(noklusējums)* pēdējais.
Datu kopas apkopošana	0	ds.0.dataset_aggregation	0 - *(noklusējums)* nav; 1 - minimums; 2 - maksimums; 3 - vidējais; 4 - skaits; 5 - summa. Parametra darbība: - *atbalstīts*, ja *Vienuma tips* ir iestatīts uz "Kopsumma"
Datu kopas etiķete	1	ds.0.data_set_label	Jebkura virknes vērtība. Noklusējums: "" (tukša).
Vienumu tagi			
Novērtēšanas tips	0	ds.0.item_tags_evaltype	0 - *(noklusējums)* Un/Vai; 2 - Vai.

Parametrs	type	name	value
Taga nosaukums	1	ds.0.item_tags.0.tag	Jebkura virknes vērtība. Piezīme: Skaitlis rekvizīta nosaukumā norāda taga secību tagu novērtēšanas sarakstā. Parametra darbība: - *obligāts*, ja tiek konfigurēti *Vienumu tagi*
Operators	0	ds.0.item_tags.0.operator	0 - Satur; 1 - Ir vienāds ar; 2 - Nesatur; 3 - Nav vienāds ar; 4 - Pastāv; 5 - Nepastāv. Piezīme: Skaitlis rekvizīta nosaukumā norāda taga secību tagu novērtēšanas sarakstā. Parametra darbība: - *obligāts*, ja tiek konfigurēti *Vienumu tagi*
Taga vērtība	1	ds.0.item_tags.0.value	Jebkura virknes vērtība. Piezīme: Skaitlis rekvizīta nosaukumā norāda taga secību tagu novērtēšanas sarakstā. Parametra darbība: - *obligāts*, ja tiek konfigurēti *Vienumu tagi*

Attēlošanas opcijas

Tālāk norādītie parametri tiek atbalstīti, konfigurējot *Attēlošanas opcijas*.

Parametrs	type	name	value
Vēsturisko datu atlase	0	source	0 - *(noklusējums)* Automātiski; 1 - Vēsture; 2 - Tendences.
Zīmēšana	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 *Zīmēšana* ir iestatīta uz “Gredzenveida diagramma”
Līnijas platums	0	stroke	Iespējamās vērtības ir no 0 līdz 10. Noklusējums: 0. Parametra darbība: - *atbalstīts*, ja *Zīmēšana* ir iestatīta uz “Gredzenveida diagramma”
Rādīt kopējo vērtību	0	total_show	0 - *(noklusējums)* Atspējots; 1 - Iespējots. Parametra darbība: - *atbalstīts*, ja *Zīmēšana* ir iestatīta uz “Gredzenveida diagramma”

Parametrs	type	name	value
Izmērs	0	value_size_type	0 - *(noklusējums)* Automātiski; 1 - Pielāgots.
Izmērs (pielāgotā izmēra vērtība)	0	value_size	Parametra darbība: - *atbalstīts*, ja *Rādīt kopējo vērtību* ir iestatīts uz "Iespējots" Iespējamās vērtības ir no 1 līdz 100. Noklusējums: 20.
Decimāldaļas	0	decimal_places	Parametra darbība: - *atbalstīts*, ja *Rādīt kopējo vērtību* ir iestatīts uz "Iespējots" Iespējamās vērtības ir no 0 līdz 6. Noklusējums: 2.
Mērvienības (izvēles rūtiņa)	0	units_show	Parametra darbība: - *atbalstīts*, ja *Rādīt kopējo vērtību* ir iestatīts uz "Iespējots" 0 - *(noklusējums)* Atspējots; 1 - Iespējots.
Mērvienības (vērtība)	1	units	Parametra darbība: - *atbalstīts*, ja *Rādīt kopējo vērtību* ir iestatīts uz "Iespējots" Jebkura virknes vērtība.
Treknraksts	0	value_bold	Parametra darbība: - *atbalstīts*, ja *Mērvienības* (izvēles rūtiņa) ir iestatīta uz "Iespējots" 0 - *(noklusējums)* Atspējots; 1 - Iespējots.
Krāsa	1	value_color	Parametra darbība: - *atbalstīts*, ja *Rādīt kopējo vērtību* ir iestatīts uz "Iespējots" Heksadecimāls krāsas kods (piemēram, FF0000).
Atstarpe starp sektoriem	0	space	Parametra darbība: - *atbalstīts*, ja *Rādīt kopējo vērtību* ir iestatīts uz "Iespējots" Iespējamās vērtības ir no 0 līdz 10. Noklusējums: 1.
Apvienot sektorus, kas ir mazāki par N% (izvēles rūtiņa)	0	merge	0 - *(noklusējums)* Atspējots; 1 - Iespējots.
Apvienot sektorus, kas ir mazāki par N% (vērtība)	0	merge_percent	Iespējamās vērtības ir no 1 līdz 10. Noklusējums: 1. Parametra darbība: - *atbalstīts*, ja *Apvienot sektorus, kas ir mazāki par N%* (izvēles rūtiņa) ir iestatīta uz "Iespējots"

Parametrs	type	name	value
Apvienot sektorus, 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
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

Leģenda

Leģendas konfigurēšanai tiek atbalstīti šādi parametri.

Parametrs	tips	nosaukums	vērtība
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 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"

Parametrs	tips	nosaukums	vērtība
Rindu skaits/ Maksimālais rindu skaits	0	legend_lines	Iespējamās vērtības ir no 1 līdz 10. Noklusējums: 1. Parametra darbība: - *atbalstīts*, ja *Rādīt leģendu* ir iestatīts uz "Iespējots"
Kolonnu skaits	0	legend_columns	Iespējamās vērtības ir 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 īpašību vērtības attiecīgajiem informācijas paneļa logrīka lauku objektiem ļauj konfigurēt logrīku *Problem hosts* metodēs `dashboard.create` un `dashboard.update`.

Attention:
Logrīka `fields` īpašības informācijas paneļa izveides vai atjaunināšanas laikā netiek validētas. Tas ļauj lietotājiem modificēt iebūvētos logrīkus un izveidot pielāgotus logrīkus, taču rada arī risku izveidot vai atjaunināt logrīkus nepareizi. Lai nodrošinātu veiksmīgu logrīka *Problem hosts* izveidi vai atjaunināšanu, skatiet tālāk 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.

Parametrs	type	name	value
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 *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*

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 apspiestās problēmas	0	show_suppressed	0 - *(noklusējums)* Atspējots; 1 - Iespējots.
Paslēpt grupas bez problēmām	0	hide_empty_groups	0 - *(noklusējums)* Atspējots; 1 - Iespējots. Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes panelī.
Problēmu atšķirošana	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 attiecīgās paneļa logrīka lauku objektu iespējamās rekvizītu vērtības ļauj konfigurēt *Problēmas* 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 rada arī risku izveidot vai atjaunināt logrīkus nepareizi. Lai nodrošinātu veiksmīgu *Problēmas* logrīka izveidi vai atjaunināšanu, skatiet tālākajās tabulās aprakstīto parametru darbību.

Parametri

Tālāk norādīti *Problēmas* logrīkam atbalstītie parametri.

Parameter	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)* Jaunākās problēmas; 2 - Vēsture; 3 - Problēmas.
Hostu grupas	2	groupids.0	Hostu grupas ID. Piezīme: Lai konfigurētu vairākas hostu grupas, izveidojiet atsevišķu informācijas paneļa logrīka lauka objektu katrai hostu grupai, īpašuma nosaukumā izmantojot palielinātu numuru. Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes informācijas panelī.
Hostu grupas (logrīks)	1	groupids._reference	Hostu grupas ID vietā: `ABCDE._hostgroupids` - iestatiet saderīgu logrīku (kura *Reference* parametram iestatīta vērtība "ABCDE") kā hostu grupu datu avotu. Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes informācijas panelī.

Parameter	type	name	value
Izslēgt hostu grupas	2	exclude_groupids.0	Hostu grupas ID. Piezīme: Lai izslēgtu vairākas hostu grupas, izveidojiet atsevišķu informācijas paneļa logrīka lauka objektu katrai hostu grupai, īpašuma nosaukumā izmantojot palielinātu numuru. Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes informācijas panelī.
Hosti	3	hostids.0	Hosta ID. Piezīme: Lai konfigurētu vairākus hostus, izveidojiet atsevišķu informācijas paneļa logrīka lauka objektu katram hostam, īpašuma nosaukumā izmantojot palielinātu numuru. Ja ir vairāki hosti, parametrs *Hostu grupas* 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 informācijas panelī.
Hosti (logrīks/informācijas panelis)	1	hostids._reference	Hosta ID vietā: `DASHBOARD.hostids` - iestatiet informācijas paneļa hostu atlasītāju kā hostu datu avotu; `ABCDE._hostids` - iestatiet saderīgu logrīku (kura *Reference* parametram iestatīta vērtība "`ABCDE`") kā hostu datu avotu. Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes informācijas panelī.
Problēma	1	problem	Problēmas notikuma nosaukums (reģistrjutīgs, pilns nosaukums vai tā daļa).
Būtiskums	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 atsevišķu informācijas paneļa logrīka lauka objektu katrai vērtībai, īpašuma nosaukumā izmantojot palielinātu numuru.
Problēmu tagi			
Novērtēšanas tips	0	evaltype	0 - *(noklusējums)* Un/Vai; 2 - Vai.
Taga nosaukums	1	tags.0.tag	Jebkura virknes vērtība. Piezīme: Skaitlis īpašuma nosaukumā norāda taga secību tagu novērtēšanas sarakstā. Parametra darbība: - *obligāts*, ja tiek konfigurēti *Problēmu tagi*

Parameter	type	name	value
Operators	0	tags.0.operator	0 - Satur; 1 - Ir vienāds ar; 2 - Nesatur; 3 - Nav vienāds ar; 4 - Pastāv; 5 - Nepastāv. Piezīme: Skaitlis īpašuma nosaukumā norāda taga secību tagu novērtēšanas sarakstā. Parametra darbība: - *obligāts*, ja tiek konfigurēti *Problēmu tagi*
Taga vērtība	1	tags.0.value	Jebkura virknes vērtība. Piezīme: Skaitlis īpašuma nosaukumā norāda taga secību tagu novērtēšanas sarakstā. Parametra darbība: - *obligāts*, ja tiek konfigurēti *Problēmu tagi*
Rādīt tagus	0	show_tags	0 - *(noklusējums)* Nevienu; 1 - 1; 2 - 2; 3 - 3.
Taga nosaukums (formāts)	0	tag_name_format	0 - *(noklusējums)* Pilns; 1 - Saīsināts; 2 - Neviens. Parametra darbība: - *atbalstīts*, ja *Rādīt tagus* ir iestatīts uz ”1”, ”2” vai ”3”
Tagu rādīšanas prioritāte	1	tag_priority	Ar komatiem atdalīts tagu saraksts. Parametra darbība: - *atbalstīts*, ja *Rādīt tagus* ir iestatīts uz ”1”, ”2” vai ”3”
Rādīt darbības datus	0	show_opdata	0 - *(noklusējums)* Nevienu; 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)* visas; 1 - Neapstiprinātas; 2 - Apstiprinātas.
Manis apstiprinātās	0	acknowledged_by_me	0 - *(noklusējums)* Atspējots; 1 - Iespējots.

Parameter	type	name	value
Kārtot ierakstus pēc	0	sort_triggers	1 - Būtiskums (dilstošā secībā); 2 - Hosts (augošā secībā); 3 - Laiks (augošā secībā); 4 - *(noklusējums)* Laiks (dilstošā secībā); 13 - Būtiskums (augošā secībā); 14 - Hosts (dilstošā secībā); 15 - Problēma (augošā secībā); 16 - Problēma (dilstošā secībā). Visām vērtībām, izņemot ”Laiks (dilstošā secībā)” un ”Laiks (augošā secībā)”, parametram *Rādīt laika skalu* jābūt iestatītam uz ”Atspējots”. Vērtības ”Hosts (augošā secībā)” un ”Hosts (dilstošā secībā)” netiek atbalstītas, ja logrīks tiek konfigurēts veidnes informācijas panelī.
Rādīt laika skalu	0	show_timeline	0 - Atspējots; 1 - *(noklusējums)* Iespējots. Parametra darbība: - *atbalstīts*, ja *Kārtot ierakstus pēc* ir iestatīts uz ”Laiks (dilstošā secībā)” vai ”Laiks (augošā secībā)”
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 no 1 līdz 100. Noklusējums: 25.
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 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 attiecīgajiem informācijas paneļa logrīka lauku objektiem pieejamās rekvizītu vērtības ļauj konfigurēt logrīku *Problēmas pēc smaguma pakāpes*, izmantojot `dashboard.create` un `dashboard.update` metodes.

Attention:
Logrīka `fields` rekvizīti informācijas paneļa izveides vai atjaunināšanas laikā netiek validēti. Tas ļauj lietotājiem modificēt iebūvētos logrīkus un izveidot pielāgotus logrīkus, taču rada arī risku izveidot vai atjaunināt logrīkus nepareizi. Lai nodrošinātu veiksmīgu logrīka *Problēmas pēc smaguma pakāpes* izveidi vai atjaunināšanu, skatiet tālākajās 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.
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.

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.
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*
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"

Parametrs	type	name	value
Rādīt darbības 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.
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 attiecīgo informācijas paneļa logrīka lauku objektu iespējamās rekvizītu vērtības ļauj konfigurēt *Izkliedes diagrammas* logrīku `dashboard.create` un `dashboard.update` metodēs.

Attention:
Logrīka `fields` rekvizīti informācijas paneļa izveides vai atjaunināšanas laikā netiek validēti. Tas ļauj lietotājiem modificēt iebūvētos logrīkus un izveidot pielāgotus logrīkus, taču rada arī risku izveidot vai atjaunināt logrīkus nepareizi. Lai nodrošinātu veiksmīgu *Izkliedes diagrammas* logrīka izveidi vai atjaunināšanu, 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 ***Datu kopa: Vienumu saraksts***	0	ds.0.dataset_type	0 - Vienumu saraksts; 1 - *(noklusējums)* Vienumu šabloni.
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 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 skaitli rekvizīta nosaukumā. Parametra darbība: - *obligāts*, ja *Datu kopas tips* ir iestatīts uz "Vienumu saraksts" un *X ass* nav iestatīta

Parametrs	type	name	value
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 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 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.
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.

Parametrs	type	name	value
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.
Aizstāt hostu	1	ds.0.override_hostid._reference	`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.

Parametrs	type	name	value
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).
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.

Parametrs	type	name	value
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 īpašību vērtības attiecīgajiem informācijas paneļa logrīka lauku objektiem ļauj konfigurēt *SLA pārskata* logrīku `dashboard.create` un `dashboard.update` metodēs.

Attention:
Logrīka `fields` īpašības informācijas paneļa izveides vai atjaunināšanas laikā netiek validētas. Tas ļauj lietotājiem modificēt iebūvētos logrīkus un izveidot pielāgotus logrīkus, taču rada arī risku izveidot vai atjaunināt logrīkus nepareizi. Lai nodrošinātu veiksmīgu *SLA pārskata* logrīka izveidi vai atjaunināšanu, skatiet tālākajās tabulās aprakstīto parametru darbību.

Parametri

SLA pārskata logrīkam tiek atbalstīti šādi parametri.

Parametrs	tips	nosaukums	vērtība
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	Pakalpojuma ID.
Rādīt periodus	0	show_periods	Iespējamās vērtības ir 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` u. c.).
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` u. c.).

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 informācijas paneļa logrīka lauku objektiem ļauj konfigurēt *Sistēmas informācija* logrīku `dashboard.create` un `dashboard.update` metodēs.

Attention:
Logrīka `fields` īpašības informācijas paneļa izveides vai atjaunināšanas laikā netiek validētas. Tas ļauj lietotājiem modificēt iebūvētos logrīkus un izveidot pielāgotus logrīkus, taču rada arī risku izveidot vai atjaunināt logrīkus nepareizi. Lai nodrošinātu veiksmīgu *Sistēmas informācija* logrīka izveidi vai atjaunināšanu, skatiet tālāk tabulās aprakstīto parametru darbību.

Parametri

Tālāk norādītie parametri ir atbalstīti *System Information* logrīkam.

Parameter	type	name	value
Refresh interval	0	rf_rate	0 - Nav atsvaidzināšanas; 10 - 10 sekundes; 30 - 30 sekundes; 60 - 1 minūte; 120 - 2 minūtes; 600 - 10 minūtes; 900 - *(noklusējums)* 15 minūtes.
Show	0	info_type	0 - *(noklusējums)* Sistēmas statistika; 1 - Augstas pieejamības mezgli.
Show software update check details	0	show_software_update_check_details	0 - *(noklusējums)* Atspējots; 1 - Iespējots. Parameter behavior: - *supported* ja `AllowSoftwareUpdateCheck` Zabbix serveris konfigurācijā ir iespējots un *Show* ir iestatīts uz "System stats"

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 informācijas paneļa logrīka lauku objektiem ļauj konfigurēt *Top hosts* logrīku `dashboard.create` un `dashboard.update` metodēs.

Attention:
Logrīka `fields` īpašības netiek validētas informācijas paneļa izveides vai atjaunināšanas laikā. Tas ļauj lietotājiem modificēt iebūvētos logrīkus un izveidot pielāgotus logrīkus, taču rada arī risku izveidot vai atjaunināt logrīkus nepareizi. Lai nodrošinātu veiksmīgu *Top hosts* logrīka izveidi vai atjaunināšanu, skatiet tālāk tabulās aprakstīto parametru darbību.

Parametri

Tālāk norādīti parametri, kas tiek atbalstīti *Top hosts* logrīkam.

Parameter	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 atsevišķu informācijas paneļa logrīka lauka objektu katrai hostu grupai, īpašuma nosaukumā izmantojot palielinātu numuru. Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes informācijas panelī.
Hostu grupas (logrīks)	1	groupids._reference	Hostu grupas ID vietā: `ABCDE._hostgroupids` - iestatiet saderīgu logrīku (kura *Reference* parametrs ir iestatīts uz "`ABCDE`") kā hostu grupu datu avotu. Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes informācijas panelī.
Hosti	3	hostids.0	Hosta ID. Piezīme: Lai konfigurētu vairākus hostus, izveidojiet atsevišķu informācijas paneļa logrīka lauka objektu katram hostam, īpašuma nosaukumā izmantojot palielinātu numuru. Ja ir vairāki hosti, parametram *Hostu grupas* jābūt vai nu vispār nekonfigurētam, vai konfigurētam ar vismaz vienu hostu grupu, kurai pieder konfigurētie hosti. Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes informācijas panelī.
Hosti (logrīks/informācijas panelis)	1	hostids._reference	Hosta ID vietā: `DASHBOARD.hostids` - iestatiet informācijas paneļa hostu atlasītāju kā hostu datu avotu; `ABCDE._hostids` - iestatiet saderīgu logrīku (kura *Reference* parametrs ir iestatīts uz "`ABCDE`") kā hostu datu avotu. Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes informācijas panelī.
Hostu tagi			
Novērtēšanas tips	0	evaltype	0 - *(noklusējums)* Un/Vai; 2 - Vai. Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes informācijas panelī.

Parameter	type	name	value
Taga nosaukums	1	tags.0.tag	Jebkura virknes vērtība. Piezīme: Skaitlis īpašuma nosaukumā norāda taga secību tagu novērtēšanas sarakstā. Parametra darbība: - *obligāts*, ja tiek konfigurēti *Hostu tagi* Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes informācijas panelī.
Operators	0	tags.0.operator	0 - satur; 1 - ir vienāds ar; 2 - nesatur; 3 - nav vienāds ar; 4 - pastāv; 5 - nepastāv. Piezīme: Skaitlis īpašuma nosaukumā norāda taga secību tagu novērtēšanas sarakstā. Parametra darbība: - *obligāts*, ja tiek konfigurēti *Hostu tagi* Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes informācijas panelī.
Taga vērtība	1	tags.0.value	Jebkura virknes vērtība. Piezīme: Skaitlis īpašuma nosaukumā norāda taga secību tagu novērtēšanas sarakstā. Parametra darbība: - *obligāts*, ja tiek konfigurēti *Hostu tagi* Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes informācijas panelī.
Rādīt hostus uzturēšanas laikā	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)* pirmie N; 3 - pēdējie N.
Hostu ierobežojums	0	show_lines	Iespējamās vērtības ir no 1 līdz 1000. Noklusējums: 10. Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes informācijas 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

Ja *Data* ir iestatīts uz “Item value”, tiek atbalstīti tālāk norādītie parametri.

Note:
Pirmais skaitlis *Thresholds* rekvizīta nosaukumā (piemēram, columnsthresholds.0.color.0) norāda kolonnu, kurai ir konfigurēti sliekšņi, savukārt otrais skaitlis norāda sliekšņa vietu augošā secībā sakārtotā sarakstā. 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"`).

Parameter	type	name	value
Vienums	1	columns.0.item	Derīgs vienuma nosaukums. Konfigurējot logrīku veidnes informācijas panelī, jāiestata tikai veidnē konfigurētie vienumi.
Rādīt vienuma vērtību kā	0	columns.0.display_value_as	0 - *(noklusējums)* Skaitliski; 1 - Teksts; 2 - Binări.
Attēlojums	0	columns.0.display	1 - *(noklusējums)* Kā ir; 2 - Josla; 3 - Indikatori; 6 - Sparkline. Parametra darbība: - *atbalstīts*, ja *Rādīt vienuma vērtību kā* ir iestatīts uz “Skaitliski”
Min.	1	columns.0.min	Jebkura skaitliska vērtība. Parametra darbība: - *atbalstīts*, ja *Rādīt vienuma vērtību kā* ir iestatīts uz “Skaitliski” un *Attēlojums* ir iestatīts uz “Josla” vai “Indikatori”
Maks.	1	columns.0.max	Jebkura skaitliska vērtība. Parametra darbība: - *atbalstīts*, ja *Rādīt vienuma vērtību kā* ir iestatīts uz “Skaitliski” un *Attēlojums* ir iestatīts uz “Josla” vai “Indikatori”

Parameter	type	name	value
Decimāldaļu skaits	0	columns.0.decimal_places	Iespējamās vērtības ir no 0 līdz 10. Noklusējums: 2. Parametra darbība: - *atbalstīts*, ja *Rādīt vienuma vērtību kā* ir iestatīts uz “Skaitliski”
Sparkline			
Platums	0	columns.0.sparkline.width	Iespējamās vērtības ir no 0 līdz 10. Noklusējums: 1. Parametra darbība: - *atbalstīts*, ja *Rādīt vienuma vērtību kā* ir iestatīts uz “Skaitliski” un *Attēlojums* ir iestatīts uz “Sparkline”
Aizpildījums	0	columns.0.sparkline.fill	Iespējamās vērtības ir no 0 līdz 10. Noklusējums: 3. Parametra darbība: - *atbalstīts*, ja *Rādīt vienuma vērtību kā* ir iestatīts uz “Skaitliski” un *Attēlojums* ir iestatīts uz “Sparkline”
Krāsa	1	columns.0.sparkline.color	Heksadecimāls krāsas kods (piemēram, `FF0000`). Noklusējums: `42A5F5`. Parametra darbība: - *atbalstīts*, ja *Rādīt vienuma vērtību kā* ir iestatīts uz “Skaitliski” un *Attēlojums* ir iestatīts uz “Sparkline”
Laika periods	1	columns.0.sparkline.time_period.reference	`DASHBOARD._timeperiod` - iestatīt informācijas paneļa laika perioda atlasītāju kā datu avotu; `ABCDE._timeperiod` - iestatīt saderīgu logrīku (kura `reference` parametra vērtība ir `ABCDE`) kā datu avotu. Noklusējums: `DASHBOARD._timeperiod` Varat arī iestatīt laika periodu tikai parametros *No* un *Līdz*. Parametra darbība: - *atbalstīts*, ja *Rādīt vienuma vērtību kā* ir iestatīts uz “Skaitliski” un *Attēlojums* ir iestatīts uz “Sparkline”
No	1	columns.0.sparkline.time_period.from	Laika vērtība absolūtajā (`YYYY-MM-DD hh:mm:ss`) vai relatīvā laika sintaksē (`now`, `now/d`, `now/w-1w` u. c.). Parametra darbība: - *atbalstīts*, ja *Rādīt vienuma vērtību kā* ir iestatīts uz “Skaitliski” un *Attēlojums* ir iestatīts uz “Sparkline”
Līdz	1	columns.0.sparkline.time_period.to	Laika vērtība absolūtajā (`YYYY-MM-DD hh:mm:ss`) vai relatīvā laika sintaksē (`now`, `now/d`, `now/w-1w` u. c.). Parametra darbība: - *atbalstīts*, ja *Rādīt vienuma vērtību kā* ir iestatīts uz “Skaitliski” un *Attēlojums* ir iestatīts uz “Sparkline” - *obligāts*, ja ir iestatīts `columns.0.sparkline.time_period.from`
Vēsturiskie dati	0	columns.0.sparkline.history	0 *(noklusējums)* Automātiski; 1 - Vēsture; 2 - Tendences. Parametra darbība: - *atbalstīts*, ja *Rādīt vienuma vērtību kā* ir iestatīts uz “Skaitliski” un *Attēlojums* ir iestatīts uz “Sparkline”

Parameter	type	name	value
Sliekšņi			
Krāsa	1	columnsthresholds.0.color	Heksadecimāls krāsas kods (piemēram, `FF0000`). Parametra darbība: - *atbalstīts*, ja *Rādīt vienuma vērtību kā* ir iestatīts uz "Skaitliski"
Slieksnis	1	columnsthresholds.0.threshold	Jebkura loka vērtība. Parametra darbība: - *atbalstīts*, ja *Rādīt vienuma vērtību kā* ir iestatīts uz "Skaitliski"
Izcēlumi			
Krāsa	1	columns.0.highlights.color	Heksadecimāls krāsas kods (piemēram, `FF0000`). Parametra darbība: - *atbalstīts*, ja *Rādīt vienuma vērtību kā* ir iestatīts uz "Teksts"
Raksts	1	columns.0.highlights.pattern	Jebkura virknes vērtība. Parametra darbība: - *atbalstīts*, ja *Rādīt vienuma vērtību kā* ir iestatīts uz "Teksts"
Rādīt sīktēlu	0	columns.0.show_thumbnail	0 - *(noklusējums)* Atspējots; 1 - Iespējots. Parametra darbība: - *atbalstīts*, ja *Rādīt vienuma vērtību kā* ir iestatīts uz "Bināri"
Apkopošanas funkcija	0	columns.0.aggregate_function	0 - *(noklusējums)* netiek izmantota; 1 - min.; 2 - maks.; 3 - vid.; 4 - skaits; 5 - summa; 6 - pirmais; 7 - pēdējais.
Laika periods	1	columns.0.time_period.reference	`DASHBOARD._timeperiod` - iestatīt informācijas paneļa laika perioda atlasītāju kā datu avotu; `ABCDE._timeperiod` - iestatīt saderīgu logrīku (kura `reference` parametra vērtība ir `ABCDE`) kā datu avotu. Noklusējums: `DASHBOARD._timeperiod` Varat arī iestatīt laika periodu tikai parametros *No* un *Līdz*. Parametra darbība: - *atbalstīts*, ja *Apkopošanas funkcija* ir iestatīta uz "min.", "maks.", "vid.", "skaits", "summa", "pirmais" vai "pēdējais"
No	1	columns.0.time_period.from	Derīga laika virkne absolūtajā (`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 un *Apkopošanas funkcija* ir iestatīta uz "min.", "maks.", "vid.", "skaits", "summa", "pirmais" vai "pēdējais" - *obligāts*, ja ir iestatīts `columns.0.time_period.to`
Līdz	1	columns.0.time_period.to	Derīga laika virkne absolūtajā (`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 un *Apkopošanas funkcija* ir iestatīta uz "min.", "maks.", "vid.", "skaits", "summa", "pirmais" vai "pēdējais" - *obligāts*, ja ir iestatīts `columns.0.time_period.from`

Parameter	type	name	value
Vēsturiskie dati	0	columns.0.history	0 - *(noklusējums)* Automātiski; 1 - Vēsture; 2 - Tendences. Parametra darbība: - *atbalstīts*, ja *Rādīt vienuma vērtību kā* ir iestatīts uz "Skaitliski"
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*

Teksts

Tālāk norādītie parametri tiek atbalstīti, ja *Data* ir iestatīts uz "Text".

Parameter	type	name	value
Text	1	columns.0.text	Jebkura teksta vērtība, tostarp makro. Atbalstītie makro: `{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 īpašību vērtības attiecīgajiem informācijas paneļa logrīka lauku objektiem ļauj konfigurēt logrīku *Top items* metodēs `dashboard.create` un `dashboard.update`.

Attention:
Logrīka `fields` īpašības informācijas paneļa izveides vai atjaunināšanas laikā netiek validētas. Tas ļauj lietotājiem modificēt iebūvētos logrīkus un izveidot pielāgotus logrīkus, taču rada arī risku izveidot vai atjaunināt logrīkus nepareizi. Lai nodrošinātu veiksmīgu logrīka *Top items* izveidi vai atjaunināšanu, lūdzu, skatiet tālākajās 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.

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	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.

Parametrs	type	name	value
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 Vienumi (skatīt zemāk)	0	show_problems	0 - Visas; 1 - *(noklusējums)* Nenoklusinātās; 2 - Nav.

Papildu konfigurācija

Tālāk norādītie papildu konfigurācijas parametri tiek atbalstīti logrīkā *Top items*.

Parameter	type	name	value
Hostsakārtošana			
Kārtot pēc	0	host_ordering_order_by	0 - *(noklusējums)* Hosts nosaukums; 3 - Vienuma vērtība.
Vienumu šabloni	1	host_ordering_item.0	Vienuma nosaukums vai šablons (piemēram, ”*: Number of processed *values per second”). Piezīme: Lai konfigurētu vairākus vienumu šablonus, izveidojiet atsevišķu informācijas paneļa logrīka lauka objektu katram vienumu šablonam, īpašuma nosaukumā izmantojot palielinātu numuru. Konfigurējot logrīku veidnes informācijas panelī, jāiestata tikai to vienumu šabloni, kuri ir 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.
Ierobežojums	0	host_ordering_limit	Iespējamās vērtības ir no 1 līdz 100. Noklusējums: 10.
Vienumu sakārtošana			
Kārtot pēc	0	item_ordering_order_by	1 - Hosts; 2 - Vienuma nosaukums; 3 - *(noklusējums)* Vienuma vērtība.
Hosts šabloni	1	item_ordering_host.0	Hosts nosaukums vai šablons. Piezīme: Lai konfigurētu vairākus hosts šablonus, izveidojiet atsevišķu informācijas paneļa logrīka lauka objektu katram hosts šablonam, īpašuma nosaukumā izmantojot palielinātu numuru. Parametra darbība: - *obligāts*, ja *Kārtot pēc* ir iestatīts uz “Hosts”

Parameter	type	name	value
Secība	0	item_ordering_order	2 - *(noklusējums)* Augšējie N; 3 - Apakšējie N.
Ierobežojums	0	item_ordering_limit	Iespējamās vērtības ir no 1 līdz 100. Noklusējums: 10.

Kolonnas

Note:
Pirmais skaitlis rekvizīta nosaukumā (piemēram, columns.0.items.0, columns.0.item_tags_evaltype) apzīmē konkrēto kolonnu, savukārt otrais skaitlis, ja tāds ir, apzīmē konfigurēto entītiju (piemēram, vienumu šablonu, tagu).

Parametrs	type	name	value
Vienumu šabloni	1	columns.0.items.0	Vienuma nosaukums vai šablons (piemēram, "*: Number of processed *values per second"). Piezīme: Lai konfigurētu vairākus vienumu šablonus, izveidojiet atsevišķu informācijas paneļa logrīka lauka objektu katram vienumu šablonam, rekvizīta nosaukumā palielinot otro skaitli par vienu. Konfigurējot logrīku veidnes informācijas panelī, jāiestata tikai to vienumu šabloni, kuri konfigurēti veidnē. Parametra darbība: - *obligāts*
Vienumu tagi			
Novērtēšanas tips	0	columns.0.item_tags_evaltype	0 - *(noklusējums)* And/Or; 2 - Or.
Taga nosaukums	1	columns.0.item_tags.0.tag	Jebkura virknes vērtība. Piezīme: Otrais skaitlis rekvizīta nosaukumā norāda taga secību tagu novērtēšanas sarakstā. Parametra darbība: - *obligāts*, ja tiek konfigurēti *Vienumu tagi*
Operators	0	columns.0.item_tags.0.operator	0 - Satur; 1 - Ir vienāds ar; 2 - Nesatur; 3 - Nav vienāds ar; 4 - Pastāv; 5 - Nepastāv. Piezīme: Otrais skaitlis rekvizīta nosaukumā norāda taga secību tagu novērtēšanas sarakstā. Parametra darbība: - *obligāts*, ja tiek konfigurēti *Vienumu tagi*
Taga vērtība	1	columns.0.item_tags.0.value	Jebkura virknes vērtība. Piezīme: Otrais skaitlis rekvizīta nosaukumā norāda taga secību tagu novērtēšanas sarakstā. Parametra darbība: - *obligāts*, ja tiek konfigurēti *Vienumu tagi*
Pamatkrāsa	1	columns.0.base_color	Heksadecimāls krāsas kods (piemēram, FF0000).
Rādīt vērtību kā	0	columns.0.display_value_as	1 - *(noklusējums)* Skaitlisku; 2 - Tekstu.

Parametrs	type	name	value
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 ”Skaitlisku” un *Attēlojums* ir iestatīts uz ”Josla” vai ”Indikatori”
Maks.	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 ”Skaitlisku” un *Attēlojums* ir iestatīts uz ”Josla” vai ”Indikatori”
Sparkline			
Platums	0	columns.0.sparkline.width	Iespējamās vērtības ir no 0 līdz 10. Noklusējums: 1. Parametra darbība: - *atbalstīts*, ja *Attēlojums* ir iestatīts uz ”Sparkline”
Aizpildījums	0	columns.0.sparkline.fill	Iespējamās vērtības ir no 0 līdz 10. Noklusējums: 3. Parametra darbība: - *atbalstīts*, ja *Attēlojums* ir iestatīts uz ”Sparkline”
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 informācijas paneļa laika perioda atlasītāju kā datu avotu; ABCDE._timeperiod - iestatīt saderīgu logrīku (ar reference parametru, kas vienāds ar ABCDE) kā datu avotu. Noklusējums: DASHBOARD._timeperiod Varat arī iestatīt laika periodu tikai parametros *No* un *Līdz*. Parametra darbība: - *atbalstīts*, ja *Attēlojums* ir iestatīts uz ”Sparkline”
No	1	columns.0.sparkline.time_period.from	Datuma laika virkne absolūtajā (YYYY-MM-DD hh:mm:ss) vai relatīvā laika sintaksē (now, now/d, now/w-1w u. c.). Parametra darbība: - *atbalstīts*, ja *Attēlojums* ir iestatīts uz ”Sparkline”
Līdz	1	columns.0.sparkline.time_period.to	Datuma laika virkne absolūtajā (YYYY-MM-DD hh:mm:ss) vai relatīvā laika sintaksē (now, now/d, now/w-1w u. c.). Parametra darbība: - *atbalstīts*, ja *Attēlojums* ir iestatīts uz ”Sparkline”

Parametrs	type	name	value
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 ”Skaitlisku”
Slieksnis	1	columns.0.thresholds.threshold	Jebkura skaitliska vērtība. Tiek atbalstīti sufiksi (piemēram, ”1d”, ”2w”, ”4K”, ”8G”). Parametra darbība: - *atbalstīts*, ja *Rādīt vērtību kā* ir iestatīts uz ”Skaitlisku”
Izcēlumi			
Izcēlums	1	columns.0.highlights.color	Heksadecimāls krāsas kods (piemēram, FF0000). Parametra darbība: - *atbalstīts*, ja *Rādīt vērtību kā* ir iestatīts uz ”Tekstu”
Slieksnis	1	columns.0.highlights.pattern	Jebkura regulārā izteiksme. Parametra darbība: - *atbalstīts*, ja *Rādīt vērtību kā* ir iestatīts uz ”Tekstu”
Cipari aiz komata **Papildu konfigurācija**	0	columns.0.decimal_places	Iespējamās vērtības ir no 0 līdz 10. Noklusējums: 2.
Apkopošanas funkcija	0	columns.0.aggregate_function	Ja *Rādīt vērtību kā* ir iestatīts uz ”Skaitlisku”: 0 - *(noklusējums)* netiek izmantota; 1 - minimums; 2 - maksimums; 3 - vidējā vērtība; 4 - skaits; 5 - summa; 6 - pirmā; 7 - pēdējā. Ja *Rādīt vērtību kā* ir iestatīts uz ”Tekstu”: 0 - *(noklusējums)* netiek izmantota; 4 - skaits; 6 - pirmā; 7 - pēdējā.
Laika periods	1	columns.0.time_period.data_set	DASHBOARD._timeperiod - iestatīt informācijas paneļa laika perioda atlasītāju kā datu avotu; ABCDE._timeperiod - iestatīt saderīgu logrīku (ar reference parametru, kas vienāds ar ABCDE) kā datu avotu. Noklusējums: DASHBOARD._timeperiod Varat arī iestatīt laika periodu tikai parametros *No* un *Līdz*. Parametra darbība: - *atbalstīts*, ja *Apkopošanas funkcija* ir iestatīta uz ”minimums”, ”maksimums”, ”vidējā vērtība”, ”skaits”, ”summa”, ”pirmā”, ”pēdējā”

Parametrs	type	name	value
No	1	columns.0.time_period.from	Laika periods sākuma vērtība. Definēta laika virkne absolūtajā (`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 un *Apkopošanas funkcija* ir iestatīta uz ”minimums”, ”maksimums”, ”vidējā vērtība”, ”skaits”, ”summa”, ”pirmā”, ”pēdējā” - *obligāts*, ja ir iestatīts `columns.0.time_period.to`
Līdz	1	columns.0.time_period.to	Laika periods beigu vērtība. Definēta laika virkne absolūtajā (`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 un *Apkopošanas funkcija* ir iestatīta uz ”minimums”, ”maksimums”, ”vidējā vērtība”, ”skaits”, ”summa”, ”pirmā”, ”pēdējā” - *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 ”Skaitlisku”
Apkopot	0	columns.0.aggregate_function	0 - *(noklusējums)* Katrs vienums 1 - Apvienoti. Parametra darbība: - *atbalstīts*, ja *Rādīt vērtību kā* ir iestatīts uz ”Skaitlisku” un *Attēlojums* ir iestatīts uz ”Kā ir”, ”Josla” vai ”Indikatori”.
Apvienotās apkopošanas funkcija	0	columns.0.column_aggregate_function	1 - minimums; 2 - maksimums; 3 - vidējā vērtība; 4 - skaits; 5 - summa *(noklusējums)*. Parametra darbība: - *obligāts*, ja tiek konfigurēts *Apkopot*.
Apvienotās kolonnas nosaukums	1	columns.0.combined_column_name	Jebkura virkne vērtība. Parametra darbība: - *obligāts*, ja tiek konfigurēts *Apkopot*.

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 informācijas paneļa logrīka lauka objektiem ļauj konfigurēt *Biežākie trigeri* logrīku `dashboard.create` un `dashboard.update` metodēs.

Attention:
Logrīka `fields` īpašības netiek validētas informācijas paneļa izveides vai atjaunināšanas laikā. Tas ļauj lietotājiem modificēt iebūvētos logrīkus un izveidot pielāgotus logrīkus, taču rada arī risku nepareizi izveidot vai atjaunināt logrīkus. Lai nodrošinātu veiksmīgu *Biežākie trigeri* logrīka izveidi vai atjaunināšanu, skatiet tālākajās tabulās aprakstīto parametru darbību.

Parametri

Tālāk norādītie parametri tiek atbalstīti *Top triggers* logrīkā.

Parameter	type	name	value
Atjaunināšanas intervāls	0	rf_rate	0 - *(noklusējums)* Bez atjaunināšanas; 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 atsevišķu informācijas paneļa logrīka lauka objektu katrai hostu grupai, lauka nosaukuma rekvizītā izmantojot palielinātu numuru. Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes informācijas panelī.

Parameter	type	name	value
Hosti	3	hostids.0	Hosta ID. Piezīme: Lai konfigurētu vairākus hostus, izveidojiet atsevišķu informācijas paneļa logrīka lauka objektu katram hostam, lauka nosaukuma rekvizītā izmantojot palielinātu numuru. Vairāku hostu gadījumā parametram *Hostu grupas* jābūt vai nu vispār nekonfigurētam, vai konfigurētam ar vismaz vienu hostu grupu, kurai pieder konfigurētie hosti. Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes informācijas panelī.
Problēma	1	problem	Problēmas notikuma nosaukums (reģistrjutīgs, pilns nosaukums vai tā daļa).
Nopietnības pakāpe	0	severities.0	Trigeru nopietnības pakāpes. 0 - Nav klasificēts; 1 - Informācija; 2 - Brīdinājums; 3 - Vidēja; 4 - Augsta; 5 - Katastrofa. Noklusējums: tukšs (visas iespējotas). Piezīme: Lai konfigurētu vairākas vērtības, izveidojiet atsevišķu informācijas paneļa logrīka lauka objektu katrai vērtībai, lauka nosaukuma rekvizītā izmantojot palielinātu numuru.
Problēmas tagi			
Novērtēšanas tips	0	evaltype	0 - *(noklusējums)* And/Or; 2 - Or.
Taga nosaukums	1	tags.0.tag	Jebkura virknes vērtība. Piezīme: Numurs rekvizīta nosaukumā norāda taga secību tagu novērtēšanas sarakstā. Parametra darbība: - *obligāts*, ja tiek konfigurēti *Problēmas tagi*
Operators	0	tags.0.operator	0 - Satur; 1 - Ir vienāds ar; 2 - Nesatur; 3 - Nav vienāds ar; 4 - Pastāv; 5 - Nepastāv. Piezīme: Numurs rekvizīta nosaukumā norāda taga secību tagu novērtēšanas sarakstā. Parametra darbība: - *obligāts*, ja tiek konfigurēti *Problēmas tagi*
Taga vērtība	1	tags.0.value	Jebkura virknes vērtība. Piezīme: Numurs rekvizīta nosaukumā norāda taga secību tagu novērtēšanas sarakstā. Parametra darbība: - *obligāts*, ja tiek konfigurēti *Problēmas tagi*

Parameter	type	name	value
Laika periods	1	time_period._referenceDASHBOARD._timeperiod - iestatīt informācijas paneļa laika perioda atlasītāju kā datu avotu; ABCDE._timeperiod - iestatīt saderīgu logrīku (ar tā *Reference* parametru iestatītu uz "ABCDE") kā datu avotu. Noklusējums: DASHBOARD._timeperiod
No	1	time_period.from	Varat arī iestatīt laika periodu 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 u. c.). Parametra darbība: - *atbalstīts*, ja *Laika periods* nav iestatīts - *obligāts*, ja time_period.to ir iestatīts
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 time_period.from ir iestatīts
Trigeru ierobežojums	0	show_lines	Iespējamās vērtības ir 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 īpašību vērtības attiecīgajiem informācijas paneļa logrīka lauku objektiem ļauj konfigurēt *Trigeru pārskata* logrīku `dashboard.create` un `dashboard.update` metodēs.

Attention:
Logrīka `fields` īpašības informācijas paneļa izveides vai atjaunināšanas laikā netiek validētas. Tas ļauj lietotājiem modificēt iebūvētos logrīkus un izveidot pielāgotus logrīkus, taču rada arī risku izveidot vai atjaunināt logrīkus nepareizi. Lai nodrošinātu veiksmīgu *Trigeru pārskata* logrīka izveidi vai atjaunināšanu, skatiet tālākajās 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.
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*

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 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 rekvizītu vērtības attiecīgajiem informācijas paneļa logrīka lauka objektiem ļauj konfigurēt *URL* logrīku `dashboard.create` un `dashboard.update` metodēs.

Attention:
Logrīka `fields` rekvizīti informācijas paneļa izveides vai atjaunināšanas laikā netiek validēti. Tas ļauj lietotājiem modificēt iebūvētos logrīkus un izveidot pielāgotus logrīkus, taču rada arī risku izveidot vai atjaunināt logrīkus nepareizi. Lai nodrošinātu veiksmīgu *URL* logrīka izveidi vai atjaunināšanu, skatiet tālākajās 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 attiecīgo informācijas paneļa logrīka lauku objektu iespējamās rekvizītu vērtības ļauj konfigurēt *Tīmekļa uzraudzības* logrīku `dashboard.create` un `dashboard.update` metodēs.

Attention:
Logrīka `fields` rekvizīti informācijas paneļa izveides vai atjaunināšanas laikā netiek validēti. Tas ļauj lietotājiem modificēt iebūvētos logrīkus un izveidot pielāgotus logrīkus, taču rada arī risku izveidot vai atjaunināt logrīkus nepareizi. Lai nodrošinātu veiksmīgu *Tīmekļa uzraudzības* logrīka izveidi vai atjaunināšanu, skatiet tālākajās tabulās aprakstīto parametru darbību.

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.

Parametrs	type	name	value
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			
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*

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: 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.
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. Ja tas ir norādīts, ir iespējams papildus izgūt arī nesen atrisinātas problēmas. Periods, kas nosaka, cik sena ir "nesen", ir definēts sadaļā *Administrēšana* > *Vispārīgi*. Problēmas, kas tika atrisinātas pirms šī perioda, netiek saglabātas problēmu tabulā. Lai izgūtu problēmas, kas tika atrisinātas senāk, izmantojiet `event.get` metodi.

Attention:
Šī metode var atgriezt dzēsta objekta problēmas, ja šīs problēmas vēl nav noņēmis housekeeper.

Note:
Šī metode ir pieejama jebkura tipa lietotājiem. Tiesības izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā Lietotāja lomas.

Parametri

`(object)` Parametri, kas definē vēlamo izvadi.

Metode atbalsta šādus parametrus.

Parameter	Type	Description
eventids	ID/array	Atgriež tikai problēmas ar norādītajiem ID.
groupids	ID/array	Atgriež tikai problēmas, ko izveidojuši objekti, kas pieder norādītajām hostu grupām.
hostids	ID/array	Atgriež tikai problēmas, ko izveidojuši objekti, kas pieder norādītajiem hostiem.
objectids	ID/array	Atgriež tikai problēmas, ko izveidojuši norādītie objekti.
source	integer	Atgriež tikai problēmas ar norādīto tipu. Skatiet problēmas notikuma objekta lapu, lai iegūtu atbalstīto notikumu tipu sarakstu. Noklusējums: 0 - problēma, ko izveidojis trigeris.
object	integer	Atgriež tikai problēmas, ko izveidojuši norādītā tipa objekti. Skatiet problēmas notikuma objekta lapu, lai iegūtu atbalstīto objektu tipu sarakstu. Noklusējums: 0 - trigeris.
acknowledged	boolean	Ja iestatīts uz `true`, atgriež tikai apstiprinātas problēmas.

Parameter	Type	Description
action	integer	Atgriež tikai problēmas, kurām ir veikta norādītā notikuma atjaunināšanas darbība. Vairākām darbībām izmantojiet jebkuru pieņemamu bitkartes vērtību summu kā bitmasku (piemēram, 34 apstiprināšanai un notikuma slāpēšanai).
action_userids	ID/array	Atgriež tikai problēmas ar norādītajiem lietotāju ID, kuri veica problēmas notikuma atjaunināšanas darbības.
suppressed	boolean	Ja iestatīts uz `true`, atgriež tikai slāpētas problēmas.
symptom	boolean	Ja iestatīts uz `true`, atgriež tikai simptomu problēmu notikumus.
severities	integer/array	Atgriež tikai problēmas ar norādītajām notikumu smaguma pakāpēm. Attiecas tikai tad, ja object ir triggeris.
evaltype	integer	Tagu novērtēšanas metode. Iespējamās vērtības: 0 - *(noklusējums)* Un/Or; 2 - Or.
tags	array	Atgriež tikai problēmas ar norādītajiem tagiem. Formāts: `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`. Tukšs masīvs atgriež visas problēmas. Iespējamās operatora vērtības: 0 - *(noklusējums)* Satur; 1 - Vienāds ar; 2 - Nesatur; 3 - Nav vienāds ar; 4 - Eksistē; 5 - Neeksistē.
recent	boolean	Ja iestatīts uz `true`, atgriež nesenās problēmas, tostarp tās, kas atrisinātas `ok_period` laikā, kā arī visas neatrisinātās problēmas. Ja iestatīts uz `false`, atgriež tikai neatrisinātās problēmas.
eventid_from	string	Atgriež tikai problēmas ar ID, kas ir lielāks vai vienāds ar norādīto ID.
eventid_till	string	Atgriež tikai problēmas ar ID, kas ir mazāks vai vienāds ar norādīto ID.
time_from	timestamp	Atgriež tikai problēmas, kas izveidotas pēc norādītā laika vai tā laikā.
time_till	timestamp	Atgriež tikai problēmas, kas izveidotas pirms norādītā laika vai tā laikā.
selectAcknowledges	query	Atgriež `acknowledges` īpašību ar problēmas atjauninājumiem. Problēmas atjauninājumi ir sakārtoti dilstošā 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 smaguma pakāpe pirms šīs atjaunināšanas darbības; `new_severity` - `(integer)` notikuma smaguma pakāpe pēc šīs atjaunināšanas darbības; `suppress_until` - `(timestamp)` laiks, līdz kuram notikums tiks slāpēts; `taskid` - `(ID)` uzdevuma ID, ja pašreizējais notikums tiek pakļauts prioritātes maiņai; `maintenanceid` - `(ID)` apkopes ID. Atbalsta `count`.
selectTags	query	Atgriež `tags` īpašību ar problēmas tagiem. Izvades formāts: `[{"tag": "<tag>", "value": "<value>"}, ...]`.

Parameter	Type	Description
selectSuppressionData	query	Atgriež `suppression_data` īpašību ar aktīvo apkopju un manuālo slāpēšanu sarakstu: `maintenanceid` - `(ID)` apkopes ID; `userid` - `(ID)` lietotāja ID, kurš slāpēja problēmu; `suppress_until` - `(integer)` laiks, līdz kuram problēma ir slāpēta.
filter	object	Atgriež 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, pret ko salīdzināt. Neatbalsta `text` datu tipa īpašības.
sortfield	string/array	Kārto rezultātu pēc norādītajām īpašībām. Iespējamās vērtības: `eventid`.
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 šiem:

- objektu masīvu;
- izgūto objektu skaitu, ja ir izmantots parametrs `countOutput`.

Piemēri

Trigeru problēmu notikumu izgūšana

Izgūstiet 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",
            "suppressed": "1",
            "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",
                    "maintenanceid": "0"
                }
            ],
            "opdata": "",
            "suppression_data": [
                {
                    "maintenanceid": "15",
                    "suppress_until": "1472511600",
                    "userid": "0"
                }
            ],
            "tags": [
                {
                    "tag": "test-tag",
                    "value": "test-value"
                }
            ],
            "urls": []
        }
    ],
    "id": 1
}
```


To retrieve problems acknowledged by specified user

Problēmu izgūšana, ko apstiprinājis lietotājs ar ID=10

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 agent on localhost is unreachable for 5 minutes",
            "acknowledged": "1",
            "severity": "3",
            "cause_eventid": "0",
            "suppressed": "0",
            "acknowledges": [
                {
                    "acknowledgeid": "9",
                    "userid": "10",
                    "action": "2"
                }
            ],
            "opdata": "",
            "urls": []
        }
    ],
    "id": 1
}
```

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.

Pārskats

Pārskata objektam ir šādi rekvizīti:

Property	Type	Description
reportid	ID	Pārskata ID. Property behavior: - *tikai lasāms* - *obligāts* atjaunināšanas darbībām
userid	ID	Lietotāja ID, kurš izveidoja pārskatu. Property behavior: - *obligāts* izveides darbībām
name	string	Unikāls pārskata nosaukums. Property behavior: - *obligāts* izveides darbībām
dashboardid	ID	Informācijas paneļa ID, uz kura pamata tiek veidots pārskats. Property behavior: - *obligāts* izveides darbībām
period	integer	Periods, par kuru tiks sagatavots pārskats. Iespējamās vērtības: 0 - *(noklusējums)* iepriekšējā diena; 1 - iepriekšējā nedēļa; 2 - iepriekšējais mēnesis; 3 - iepriekšējais gads.
cycle	integer	Periodisks atkārtošanas grafiks. Iespējamās vērtības: 0 - *(noklusējums)* katru dienu; 1 - katru nedēļu; 2 - katru mēnesi; 3 - katru gadu.
start_time	integer	Dienas laiks sekundēs, kad pārskats tiks sagatavots nosūtīšanai. Noklusējums: 0.

Property	Type	Description
weekdays	integer	Nedēļas dienas, kurās nosūtīt pārskatu. Iespējamās bitkartes vērtības: 1 - pirmdiena; 2 - otrdiena; 4 - trešdiena; 8 - ceturtdiena; 16 - piektdiena; 32 - sestdiena; 64 - svētdiena. Šis ir bitmaskas lauks; ir pieņemama jebkura iespējamo bitkartes vērtību summa (piemēram, 21 pirmdienai, trešdienai un piektdienai). Noklusējums: 0. Property behavior: - *obligāts*, ja `cycle` ir iestatīts uz ”weekly”.
active_since	string	Datums, no kura sākt. Iespējamās vērtības: tukša virkne - *(noklusējums)* nav norādīts (tiek saglabāts kā 0); konkrēts datums formātā YYYY-MM-DD (tiek saglabāts kā dienas sākuma laika zīmogs (00:00:00)).
active_till	string	Datums, kurā beigt. Iespējamās vērtības: tukša virkne - *(noklusējums)* nav norādīts (tiek saglabāts kā 0); konkrēts datums formātā YYYY-MM-DD (tiek saglabāts kā dienas beigu laika zīmogs (23:59:59)).
subject	string	Pārskata ziņojuma temats.
message	string	Pārskata ziņojuma teksts.
status	integer	Norāda, vai pārskats ir iespējots vai atspējots. Iespējamās vērtības: 0 - *(noklusējums)* iespējots; 1 - atspējots.
description	text	Pārskata apraksts.
state	integer	Pārskata stāvoklis. Iespējamās vērtības: 0 - *(noklusējums)* pārskats vēl nav apstrādāts; 1 - pārskats ir ģenerēts un veiksmīgi nosūtīts visiem adresātiem; 2 - pārskata ģenerēšana neizdevās; laukā ”info” ir ietverta informācija par kļūdu; 3 - pārskats ir ģenerēts, taču nosūtīšana dažiem (vai visiem) adresātiem neizdevās; laukā ”info” ir ietverta informācija par kļūdu. Property behavior: - *tikai lasāms*
lastsent	timestamp	Pēdējā veiksmīgi nosūtītā pārskata Unix laika zīmogs. Property behavior: - *tikai lasāms*
info	string	Kļūdas apraksts vai papildu informācija. Property behavior: - *tikai lasāms*

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 ieplānotos pārskatus.

Note:
Šī metode ir pieejama tikai *Admin* un *Super admin* lietotāju tipiem. Tiesības izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā 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* lietotāja tipam. Tiesības izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā Lietotāja 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)`

Šī metode ļauj izgūt ieplānotos pārskatus atbilstoši norādītajiem parametriem.

> **Note:**
> Šī metode ir pieejama jebkura tipa lietotājiem. Tiesības 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
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

Pārskatu 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 plānotos pārskatus.

> **Note:**
> Šī metode ir pieejama tikai *Admin* un *Super admin* lietotāja tipam. Tiesības izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā Lietotāja 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.

Property	Type	Description
regexpid	ID	Regulārās izteiksmes ID. Property behavior: - *tikai lasāms* - *obligāts* atjaunināšanas operācijām.
name	string	Regulārās izteiksmes nosaukums. Property behavior: - *obligāts* izveides operācijām.
description	text	Regulārās izteiksmes apraksts.
test_string	string	Testa virkne.

Izteiksmes

Izteiksmju objekts satur šādus rekvizītus.

Property	Type	Description
expression	string	Regulārā izteiksme. Property behavior: - *required*

Property	Type	Description
expression_type	integer	Regulārās izteiksmes tips. Iespējamās vērtības: 0 - Satur virkni; 1 - Satur jebkuru apakšvirkni no saraksta; 2 - Nesatur virkni; 3 - Atbilst regulārajai izteiksmei; 4 - Neatbilst regulārajai izteiksmei. Property behavior: - *required*
exp_delimiter	string	Izteiksmes atdalītājs. Noklusējuma vērtība: `","`. Iespējamās vērtības: `","` vai `"."`, vai `"/"`. Property behavior: - *supported* if `expression_type` is set to *Contains any substring from list*.
case_sensitive	integer	Reģistrjutība. Noklusējuma vērtība: 0. Iespējamās vērtības: 0 - Nav reģistrjutīgs; 1 - Reģistrjutīgs.

regexp.create

Apraksts

`object regexp.create(object/array regularExpressions)`

Šī metode ļauj izveidot jaunus globālos regulāros izteicienus.

> **Note:**
> Šī metode ir pieejama tikai *Super admin* lietotāju tipiem. Atļaujas izsaukt šo metodi var atsaukt lietotāju lomu iestatījumos. Plašāku informāciju skatiet User roles.

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 globālas regulārās izteiksmes izveide.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "regexp.create",
    "params": {
        "name": "Storage devices for SNMP discovery",
        "description": "Globāla regulārā izteiksme, ko izmanto, lai atklātu un identificētu atmiņas resurs
        "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": [
            "9"
        ]
    },
    "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. Tiesības izsaukt šo metodi var atsaukt lietotāju lomu iestatījumos. Plašāku informāciju skatiet User roles.

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)`

Šī metode ļauj izgūt globālās regulārās izteiksmes atbilstoši norādītajiem parametriem.

Note:
Šī metode ir pieejama tikai *Super Admin*. Tiesības izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet User roles.

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","description"],
        "selectExpressions": ["expression","expression_type"],
        "regexpids": [1,2],
        "preservekeys": true
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "1": {
            "regexpid": "1",
            "name": "Failu sistēmas atklāšanai",
            "description": "",
            "expressions": [
                {
                    "expression": "^(btrfs|ext2|ext3|ext4|reiser|xfs|ffs|ufs|jfs|jfs2|vxfs|hfs|apfs|refs|n
                    "expression_type": "3"
                }
            ]
        },
        "2": {
            "regexpid": "2",
            "name": "Tīkla interfeisi atklāšanai",
            "description": "",
            "expressions": [
                {
                    "expression": "^Software Loopback Interface",
                    "expression_type": "4"
                },
                {
                    "expression": "^(In)?[Ll]oop[Bb]ack[0-9._]*$",
                    "expression_type": "4"
                },
                {
                    "expression": "^NULL[0-9.]*$",
                    "expression_type": "4"
                },
                {
                    "expression": "^[Ll]o[0-9.]*$",
                    "expression_type": "4"
                },
                {
                    "expression": "^[Ss]ystem$",
                    "expression_type": "4"
                },
                {
                    "expression": "^Nu[0-9.]*$",
                    "expression_type": "4"
                }
            ]
        }
```

```
    },
    "id": 1
}
```

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 atsaukt lietotāju lomu iestatījumos. Plašāku informāciju skatiet User roles.

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": "Failu sistēmas atklāšanai",
        "description": "Globāla regulārā izteiksme, ko izmanto, lai atklātu un identificētu failu sistēmas
        "test_string": "",
        "expressions": [
            {
                "expression": "^(btrfs|ext2|ext3|ext4|reiser|xfs|ffs|ufs|jfs|jfs2|vxfs|hfs|apfs|refs|zfs)$
                "expression_type": "3",
                "case_sensitive": "0"
            },
            {
                "expression": "^(ntfs|fat32|fat16)$",
                "expression_type": "3",
                "case_sensitive": "0"
            }
        ]
    },
    "id": 1
}
```

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.

Property	Type	Description
connectorid	ID	Savienotāja ID. Īpašības darbība: - *tikai lasāms* - *obligāts* atjaunināšanas darbībām
name	string	Savienotāja nosaukums. Īpašības darbība: - *obligāts* izveides darbībām
url	string	Gala punkta URL, tas ir, saņēmēja URL. Tiek atbalstīti lietotāja makro. Īpašības darbība: - *obligāts* izveides darbībām
protocol	integer	Saziņas protokols. Iespējamās vērtības: 0 - *(noklusējums)* Zabbix Streaming Protocol v1.0.
data_type	integer	Datu tips. Iespējamās vērtības: 0 - *(noklusējums)* Vienumu vērtības; 1 - Notikumi.

Property	Type	Description
item_value_type	integer	Nosūtāmo vienumu vērtību tipu summa. Iespējamās vērtības: 1 - Skaitlisks (peldošais komats); 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	Nosūtītāja procesu skaits, kas jāpalaiž šim savienotājam. Iespējamās vērtības: 1-100. Noklusējums: 1.
max_attempts	integer	Mēģinājumu skaits. Iespējamās vērtības: 1-5. Noklusējums: 1.
attempt_interval	string	Intervāls starp atkārtotiem mēģinājumiem. Tiek pieņemtas 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 makro. Iespējamās vērtības: 1s-60s. Noklusējums: 5s.
http_proxy	string	HTTP(S) starpniekservera savienojuma virkne formātā *[protocol]://[username[:password]@]proxy.example.com[:port]*. Tiek atbalstīti lietotāja makro.
authtype	integer	HTTP autentifikācijas metode. Iespējamās vērtības: 0 - *(noklusējums)* Nav; 1 - Basic; 2 - NTLM; 3 - Kerberos; 4 - Digest; 5 - Bearer.

Property	Type	Description
username	string	Lietotājvārds. Tiek atbalstīti lietotāja makro. Īpašības darbība: - *atbalstīts*, ja `authtype` ir iestatīts uz “Basic”, “NTLM”, “Kerberos” vai “Digest”
password	string	Parole. Tiek atbalstīti lietotāja makro. Īpašības darbība: - *atbalstīts*, ja `authtype` ir iestatīts uz “Basic”, “NTLM”, “Kerberos” vai “Digest”
token	string	Bearer marķieris. Tiek atbalstīti lietotāja makro. Īpašības darbība: - *obligāts*, ja `authtype` ir iestatīts uz “Bearer”
verify_peer	integer	Vai pārbaudīt, ka hosts sertifikāts ir autentisks. Iespējamās vērtības: 0 - Nepārbaudīt; 1 - *(noklusējums)* Pārbaudīt.
verify_host	integer	Vai pārbaudīt, ka savienojuma hosts nosaukums atbilst tam, kas norādīts hosts sertifikātā. Iespējamās vērtības: 0 - Nepārbaudīt; 1 - *(noklusējums)* Pārbaudīt.
ssl_cert_file	string	Publiskās SSL atslēgas faila ceļš. Tiek atbalstīti lietotāja makro.
ssl_key_file	string	Privātās SSL atslēgas faila ceļš. Tiek atbalstīti lietotāja makro.
ssl_key_password	string	SSL atslēgas faila parole. Tiek atbalstīti lietotāja makro.
description	text	Savienotāja apraksts.
status	integer	Vai savienotājs ir iespējots. Iespējamās vērtības: 0 - Atspējots; 1 - *(noklusējums)* Iespējots.
tags_evaltype	integer	Tagu novērtēšanas metode. Iespējamās vērtības: 0 - *(noklusējums)* Un/Vai; 2 - Vai.

Tagu filtrs

Tagu filtrs ļauj eksportēt tikai atbilstošas vienumu vērtības vai notikumus. Ja tas nav iestatīts, tiks eksportēts viss. Tagu filtra objektam ir šādas īpašības.

Property	Type	Description
tag	string	Taga nosaukums. Property behavior: - *required*

Property	Type	Description
operator	integer	Nosacījuma operators. Iespējamās vērtības: 0 - *(default)* Vienāds ar; 1 - Nav vienāds ar; 2 - Satur; 3 - Nesatur; 12 - Eksistē; 13 - Nepastāv.
value	string	Taga 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. Tiesības izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet 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 savienotāja ierakstus.

> **Note:**
> Šī metode ir pieejama tikai *Super admin* lietotāja tipam. Tiesības izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Plašāka informācija ir pieejama sadaļā 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)`

Šī metode ļauj izgūt connector objektus atbilstoši norādītajiem parametriem.

> **Note:**
> Šī metode ir pieejama tikai *Super admin* lietotāja tipam. Tiesības izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet User roles.

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. Tiesības izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāja 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 šādi rekvizīti.

Property	Type	Description
scriptid	ID	Skripta ID. Rekvizīta darbība: - *tikai lasāms* - *obligāts* atjaunināšanas operācijām
name	string	Skripta nosaukums. Rekvizīta 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. Iespējamā vērtība, ja `scope` ir iestatīts uz ”manual host action” vai ”manual event action”: 6 - URL. Rekvizīta darbība: - *obligāts* izveides operācijām
command	string	Izpildāmā komanda. Rekvizīta darbība: - *obligāts*, ja `type` ir iestatīts uz ”Script”, ”IPMI”, ”SSH”, ”TELNET” vai ”Webhook”
scope	integer	Skripta tvērums. Iespējamās vērtības: 1 - action operation; 2 - manual host action; 4 - manual event action. Rekvizīta darbība: - *obligāts* izveides operācijām

Property	Type	Description
execute_on	integer	Vieta, kur izpildīt skriptu. Iespējamās vērtības: 0 - izpildīt Zabbix aģentā; 1 - izpildīt Zabbix serverī. Tas ir *atbalstīts* tikai tad, ja Zabbix serverī ir iespējota globālo skriptu izpilde; 2 - *(noklusējums)* izpildīt Zabbix serverī vai starpniekserverī. Rekvizīta darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”Script”
menu_path	string	Ar slīpsvītru atdalītas mapes, kas, noklikšķinot uz hosta vai notikuma, lietotāja saskarnē veido izvēlnei līdzīgu navigāciju. Rekvizīta darbība: - *atbalstīts*, ja `scope` ir iestatīts uz ”manual host action” vai ”manual event action”
authtype	integer	SSH skripta tipam izmantotā autentifikācijas metode. Iespējamās vērtības: 0 - parole; 1 - publiskā atslēga. Rekvizīta darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”SSH”
username	string	Autentifikācijai izmantotais lietotājvārds. Rekvizīta darbība: - *obligāts*, ja `type` ir iestatīts uz ”SSH” vai ”TELNET”
password	string	Parole, ko izmanto SSH skriptiem ar paroles autentifikāciju un TELNET skriptiem. Rekvizīta 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, ko izmanto SSH skriptiem ar publiskās atslēgas autentifikāciju. Rekvizīta darbība: - *obligāts*, ja `type` ir iestatīts uz ”SSH” un `authtype` ir iestatīts uz ”public key”
privatekey	string	Privātās atslēgas faila nosaukums, ko izmanto SSH skriptiem ar publiskās atslēgas autentifikāciju. Rekvizīta darbība: - *obligāts*, ja `type` ir iestatīts uz ”SSH” un `authtype` ir iestatīts uz ”public key”
port	string	SSH un TELNET skriptiem izmantotais porta numurs. Rekvizīta darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”SSH” vai ”TELNET”
groupid	ID	Hostu grupas ID, kurā skriptu var izpildīt. Ja iestatīts uz ”0”, skripts būs pieejams visās hostu grupās. Noklusējums: 0.

Property	Type	Description
usrgrpid	ID	Lietotāju grupas ID, kurai būs atļauts izpildīt skriptu. Ja iestatīts uz ”0”, skripts būs pieejams visām lietotāju grupām. Noklusējums: 0. Rekvizīta 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. Rekvizīta darbība: - *atbalstīts*, ja `scope` ir iestatīts uz ”manual host action” vai ”manual event action”
confirmation	string	Apstiprinājuma uznirstošā loga teksts. Uznirstošais logs tiks parādīts, mēģinot izpildīt skriptu no Zabbix lietotāja saskarnes. Rekvizīta 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. Rekvizīta darbība: - *obligāts*, ja `type` ir iestatīts uz ”Webhook”
parameters	array	Webhook ievades parametru masīvs. Rekvizīta darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”Webhook”
description	string	Skripta apraksts.
url	string	Lietotāja definēts URL. Rekvizīta darbība: - *obligāts*, ja `type` ir iestatīts uz ”URL”
new_window	integer	Atvērt URL jaunā logā. Iespējamās vērtības: 0 - Nē; 1 - *(noklusējums)* Jā. Rekvizīta darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”URL”
manualinput	integer	Norāda, vai skripts pieņem lietotāja ievadi. Iespējamās vērtības: 0 - *(noklusējums)* Atspējots; 1 - Iespējots; Rekvizīta darbība: - *atbalstīts*, ja `scope` ir iestatīts uz ”manual host action” vai ”manual event action”

Property	Type	Description
manualinput_prompt	string	Manuālās ievades uzvednes teksts. Rekvizīta darbība: - *obligāts*, ja `manualinput` ir iestatīts uz ”Enabled”
manualinput_validator	string	Rakstzīmju virknes lauks, ko izmanto lietotāja ievades validēšanai. Virkne sastāv no regulārās izteiksmes vai ar komatiem atdalītu vērtību kopas. Rekvizīta darbība: - *obligāts*, ja `manualinput` ir iestatīts uz ”Enabled”
manualinput_validator_type	integer	Nosaka sagaidāmās lietotāja ievades tipu. Iespējamās vērtības: 0 - *(noklusējums)* String. Norāda, ka manualinput_validator jāapstrādā kā regulārā izteiksme; 1 - List. Norāda, ka manualinput_validator jāapstrādā kā ar komatiem atdalīts iespējamo ievades vērtību saraksts. Rekvizīta darbība: - *atbalstīts*, ja `manualinput` ir iestatīts uz ”Enabled”
manualinput_default_value	string	Noklusējuma vērtība lietotāja ievades automātiskai aizpildīšanai. Rekvizīta 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 atsaukt lietotāja lomas iestatījumos. Papildinformāciju skatiet sadaļā User roles.

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 atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā User roles.

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 palaist skriptu uz hosta vai notikuma. Izņēmums ir URL tipa skripti; tos nevar izpildīt.

Note:
Šī metode ir pieejama jebkura tipa lietotājiem. Tiesības izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā Lietotāja 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. Tiesības izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā User roles.

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, tā aizstāj {MANUALINPUT} makro ar norādīto vērtību.

Note:
Šī metode ir pieejama jebkura tipa lietotājiem. Tiesības izsaukt metodi var atsaukt lietotāja lomu iestatījumos. Plašāku informāciju skatiet sadaļā User roles.

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 izgūt visus pieejamos skriptus norādītajā hostā vai konkrētu skriptu, ja ir norādīts skripta ID. Ja ir norādīts manualinput, tā aizstāj {MANUALINPUT} makro ar norādīto vērtību.

Note:
Šī metode ir pieejama jebkura tipa lietotājiem. Tiesības izsaukt metodi var atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā User roles.

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. Tiesības izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā User roles.

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 jums izveidot jaunus SLA objektus.

Note:
Šī metode ir pieejama tikai lietotāju tipiem *Admin* un *Super admin*. Atļaujas izsaukt šo metodi var atsaukt lietotāju lomu iestatījumos. Plašāku informāciju skatiet User roles.

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 jauninājanas 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. Tiesības izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā Lietotāja 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)`

Šī metode ļauj izgūt SLA objektus atbilstoši norādītajiem parametriem.

> **Note:**
> Šī metode ir pieejama jebkura tipa lietotājiem. Tiesības izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāja 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 Pakalpojuma līmeņa indikatora (SLI) datus Pakalpojuma līmeņa līgumam (SLA).

Note:
Šī metode ir pieejama jebkura tipa lietotājiem. Tiesības izsaukt metodi var atsaukt 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. Tiesības izsaukt šo metodi var atsaukt 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 šādi rekvizīti.

Property	Type	Description
proxyid	ID	Starpniekservera ID. Rekvizīta darbība: - *tikai lasāms* - *obligāts* atjaunināšanas darbībām
name	string	Starpniekservera nosaukums. Rekvizīta darbība: - *obligāts* izveides darbībām
proxy_groupid	ID	Starpniekserveru grupas ID. 0 - starpniekserveris nav piešķirts nevienai starpniekserveru grupai.
local_address	string	Adrese aktīvajiem aģentiem. IP adrese vai DNS nosaukums, ar kuru izveidot savienojumu. Rekvizīta darbība: - *obligāts*, ja `proxy_groupid` nav 0

Property	Type	Description
local_port	string	Vietējais starpniekservera porta numurs, ar kuru izveidot savienojumu. Tiek atbalstīti lietotāja makro. Noklusējums: 10051. Rekvizīta darbība: - *atbalstīts*, ja `proxy_groupid` nav 0
operating_mode	integer	Starpniekservera tips. Iespējamās vērtības: 0 - aktīvs starpniekserveris; 1 - pasīvs starpniekserveris. Rekvizīta darbība: - *obligāts* izveides darbībām
description	text	Starpniekservera apraksts.
lastaccess	timestamp	Laiks, kad starpniekserveris pēdējo reizi izveidoja savienojumu ar serveri. Rekvizīta darbība: - *tikai lasāms*
address	string	IP adrese vai DNS nosaukums, ar kuru izveidot savienojumu. Tiek atbalstīti lietotāja makro. Rekvizīta darbība: - *obligāts*, ja Zabbix starpniekservera darbības režīms ir pasīvs
port	string	Savienojuma porta numurs. Tiek atbalstīti lietotāja makro. Noklusējums: 10051. Rekvizīta 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 starpniekserveri. Iespējamās vērtības: 1 - *(noklusējums)* bez šifrēšanas; 2 - PSK; 4 - sertifikāts.
tls_accept	integer	Savienojumi no starpniekservera. 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.

Property	Type	Description
tls_psk_identity	string	PSK identitāte; tai jābūt saistītai tikai ar vienu PSK (starp automātisko reģistrāciju, hostiem un starpniekserveriem). Neiekļaujiet PSK identitātē sensitīvu informāciju, jo tā tiek nosūtīta nešifrētā veidā tīklā, lai informētu saņēmēju, kuru PSK izmantot. Rekvizīta darbība: - *tikai rakstāms* - *obligāts*, ja `tls_connect` ir iestatīts uz ”PSK” vai `tls_accept` satur bitu ”PSK”
tls_psk	string	Iepriekš koplietota atslēga (PSK); tai jāsatur vismaz 32 heksadecimālie cipari. Rekvizīta darbība: - *tikai rakstāms* - *obligāts*, ja `tls_connect` ir iestatīts uz ”PSK” vai `tls_accept` satur bitu ”PSK”
custom_timeouts	integer	Vai ignorēt globālos vienumu taimautus starpniekservera līmenī. Iespējamās vērtības: 0 - *(noklusējums)* izmantot globālos iestatījumus; 1 - ignorēt taimautus.
timeout_zabbix_agent	string	Apstrādei izmantot ne vairāk kā `timeout_*` sekundes. Pieņem sekunžu skaitu (piemēram, 1, 60, 300) vai vērtību ar laika sufiksu (piemēram, 30s, 1m). Tiek pieņemti arī lietotāja makro. Izmanto tikai tad, ja `custom_timeouts` ir iestatīts uz ”override timeouts”; pretējā gadījumā vērtība tiek ignorēta. Iespējamais vērtību diapazons: 1-600s. Rekvizīta darbība: - *obligāts*, ja `custom_timeouts` ir iestatīts uz ”override timeouts”
timeout_simple_check		
timeout_snmp_agent		
timeout_external_check		
timeout_db_monitor		
timeout_http_agent		
timeout_ssh_agent		
timeout_telnet_agent		
timeout_script		
timeout_browser		
timeout_telemetry_query		
version	integer	Starpniekservera versija. Trīsdaļīgs Zabbix versijas numurs, kur katrai daļai tiek izmantoti divi decimālie cipari, piemēram, 60401 versijai 6.4.1, 70002 versijai 7.0.2 utt. 0 - nezināma starpniekservera versija. Rekvizīta darbība: - *tikai lasāms*

Property	Type	Description
compatibility	integer	Starpniekservera versija salīdzinājumā ar Zabbix servera versiju. Iespējamās vērtības: 0 - nenoteikts; 1 - pašreizējā versija (starpniekserverim un serverim ir viena un tā pati galvenā versija); 2 - novecojusi versija (starpniekservera versija ir vecāka par servera versiju, taču tiek daļēji atbalstīta); 3 - neatbalstīta versija (starpniekservera versija ir vecāka par servera iepriekšējo LTS laidiena versiju vai servera galvenā versija ir vecāka par starpniekservera galveno versiju). Rekvizīta darbība: - *tikai lasāms*
state	integer	Starpniekservera stāvoklis. Iespējamās vērtības: 0 - nezināms; 1 - bezsaistē; 2 - tiešsaistē. Rekvizīta darbība: - *tikai lasāms*

Lietojumprogrammu veiktspējas uzraudzība (APM)

APM objektam ir šādas īpašības.

Property	Type	Description
data_collection_status	integer	Vai telemetrijas datu apkopošana ir iespējota starpniekserverī. Iespējamās vērtības: 0 - *(noklusējums)* Atspējota; 1 - Iespējota.
max_messages_per_second	integer	Maksimālais ziņojumu skaits sekundē, ko starpniekserveris pieņem. Iespējamo vērtību diapazons: 0 (neierobežots) – 2147483647. Noklusējums: 0. Īpašības darbība: - *supported*, ja `data_collection_status` ir iestatīts uz "Enabled"
additional_resource_attributes	array	Resursu atribūtu masīvs, kas jāpievieno ienākošajiem telemetrijas datiem. Īpašības darbība: - *supported*, ja `data_collection_status` ir iestatīts uz "Enabled"

Resursa atribūti

Resursa atribūtu objektam ir šādi rekvizīti.

Property	Type	Description
key	string	Resursa atribūta nosaukums. Norādītajam `signal_type` tam ir jābūt unikālam. Rekvizīta darbība: - *obligāts*

Property	Type	Description
value	string	Resursa atribūta vērtība. Var būt tukša.
signal_type	integer	Telemetrijas datu veids, uz kuru attiecas atribūts. Iespējamās vērtības: 0 - Traces; 1 - Metrics; 2 - Logs. Rekvizīta darbība: - *obligāts*

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. Tiesības izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāju lomas.

Parametri

`(object/array)` Izveidojamie starpniekserveri.

Papildus standarta starpniekservera rekvizītiem metode pieņem šādus parametrus.

Parametrs	Tips	Apraksts
hosts	array	Hosts, kurus uzraudzīs starpniekserveris. Ja hosts jau uzrauga cits starpniekserveris, tas tiks piešķirts pašreizējam starpniekserverim. Hosts drīkst būt definēts tikai `hostid` rekvizīts.
apm	object	Starpniekservera lietojumprogrammu veiktspējas uzraudzības (APM) iestatījumi.

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
}
```

Starpniekservera izveide ar sertifikātu balstītu šifrēšanu

Izveidojiet aktīvu starpniekserveri ar sertifikātu balstītu šifrēšanu. Starpniekserveris ir konfigurēts, lai uzraudzītu hostu "10785".

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "proxy.create",
    "params": {
        "name": "Active proxy",
        "operating_mode": "0",
        "hosts": [
            {
                "hostid": "10785"
            }
        ],
        "tls_accept": "4",
        "tls_issuer": "CN=Signing CA,OU=Development group,O=Zabbix SIA,DC=zabbix,DC=com",
        "tls_subject": "CN=Zabbix server,OU=Development group,O=Zabbix SIA,DC=zabbix,DC=com"
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "proxyids": [
            "2"
        ]
    },
    "id": 1
}
```

Starpniekservera izveide ar iespējotu APM

Izveidojiet aktīvu starpniekserveri "Active proxy" ar iespējotu APM.
Starpniekserveris pieņem līdz 1000 ziņojumiem sekundē, pievieno resursa atribūtu `deployment.environment` trasēm un resursa atribūtu `cloud.region` metrikām.

Request:

```
{
    "jsonrpc": "2.0",
    "method": "proxy.create",
    "params": {
        "name": "Active proxy",
        "operating_mode": "0",
        "apm": {
            "data_collection_status": "1",
            "max_messages_per_second": "1000",
```

```
            "additional_resource_attributes": [
                {
                    "key": "deployment.environment",
                    "value": "production",
                    "signal_type": "0"
                },
                {
                    "key": "cloud.region",
                    "value": "eu-central-1",
                    "signal_type": "1"
                }
            ]
        }
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "proxyids": [
            "3"
        ]
    },
    "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. Tiesības izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet User roles.

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)`

Šī metode ļauj izgūt starpniekserverus atbilstoši norādītajiem parametriem.

Note:
Šī metode ir pieejama visu veidu lietotājiem. Atļaujas izsaukt šo metodi var atsaukt lietotāja lomu iestatījumos. Plašāku informāciju skatiet sadaļā Lietotāju lomas.
 Tiek atgriezti tikai tie starpniekserveri, kurus atļauj lietotāja lietotāju grupu *Proxy access list*.

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.
selectApm	query	Atgriezt `apm` rekvizītu ar starpniekservera lietojumprogrammu veiktspējas pārraudzības (APM) iestatījumiem.
selectAssignedHosts	query	Atgriezt `assignedHosts` rekvizītu ar hosts, kas piešķirti starpniekserverim. Atbalsta `count`.
selectHosts	query	Atgriezt `hosts` rekvizītu ar hosts, kurus pārrauga starpniekserveris. Atbalsta `count`.
selectProxyGroup	query	Atgriezt `proxyGroup` rekvizītu ar starpniekserveru grupas objektu.
sortfield	string/array	Kārtot rezultātu pēc norādītajiem rekvizītiem. Iespējamās vērtības: `proxyid`, `name`, `operating_mode`.
countOutput	boolean	Šie parametri ir aprakstīti atsauces komentārā.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	

Parametrs	Tips	Apraksts
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 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. Tiesības izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet User roles.

Parametri

`(object/array)` Atjaunināmie starpniekservera rekvizīti.

Katram starpniekserverim ir jābūt definētam rekvizītam `proxyid`, visi pārējie rekvizīti nav obligāti. Tiks atjaunināti tikai nodotie rekvizīti, bet visi pārējie paliks nemainīti.

Papildus standarta starpniekservera rekvizītiem metode pieņem šādus parametrus.

Parameter	Type	Description
hosts	array	Hosts, kurus jāuzrauga starpniekserverim. Ja hosts jau uzrauga cits starpniekserveris, tas tiks piešķirts pašreizējam starpniekserverim. Hosts drīkst būt definēts tikai rekvizīts `hostid`.
apm	object	Starpniekservera lietojumprogrammu veiktspējas uzraudzības (APM) iestatījumi.

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.

Property	Type	Description
proxy_groupid	ID	Starpniekserveru grupas ID. Property behavior: - *tikai lasāms* - *obligāts* atjaunināšanas darbībām
name	string	Starpniekserveru grupas nosaukums. Property behavior: - *obligāts* izveides darbībām
description	text	Starpniekserveru grupas apraksts.
failover_delay	string	Laika periods, kurā starpniekserverim starpniekserveru grupā ir jāsazinās ar Zabbix serveri, lai to uzskatītu par tiešsaistē esošu. <brTiek atbalstīti laika sufiksi (piemēram, 30s, 1m). Tiek atbalstīti lietotāja makro. Iespējamās vērtības: 10s-15m. Noklusējums: 1m.
min_online	string	Minimālais tiešsaistē esošo starpniekserveru skaits, kas nepieciešams, lai starpniekserveru grupa paliktu tiešsaistē. Tiek atbalstīti lietotāja makro. Iespējamo vērtību diapazons: 1-1000. Noklusējums: 1.
state	integer	Starpniekserveru grupas stāvoklis. Iespējamās vērtības: 0 - Nezināms; 1 - Bezsaistē; 2 - Atkopjas; 3 - Tiešsaistē; 4 - Pasliktinās. Property behavior: - *tikai lasāms*

proxygroup.create

Apraksts

`object proxygroup.create(object/array proxyGroups)`

Šī metode ļauj izveidot jaunus starpniekserveru grupas.

> **Note:**
> Šī metode ir pieejama tikai *Super admin* lietotāja tipam. Tiesības izsaukt šo metodi var atsaukt 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. Tiesības izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāja 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 visu veidu lietotājiem. Atļaujas izsaukt šo metodi var atsaukt lietotāja lomu iestatījumos. Plašāku informāciju skatiet sadaļā Lietotāju lomas. Tiek atgrieztas tikai tās starpniekserveru grupas, kuras ir atļautas lietotāja lietotāju grupu *starpniekserveru piekļuves sarakstā*.

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. Tiesības izsaukt šo metodi var atsaukt 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:**
> Lietotāja lomai ir nepieciešama *Pārvaldīt API tokenus* atļauja, lai pārvaldītu tokenus citiem lietotājiem.

> **Attention:**
> Ar šo metodi izveidotais tokens pirms lietošanas arī ir jābūt ģenerētam.

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:**
> Lietotāja lomai ir nepieciešama *Pārvaldīt API tokenus* atļauja, lai pārvaldītu tokenus citiem lietotājiem.

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 *Pārvaldīt API tokenus* atļauja, lai pārvaldītu tokenus citiem lietotājiem.

> **Attention:**
> Tokenu ar šo metodi var ģenerēt tikai tad, ja tas ir ticis 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)`

Šī metode ļauj izgūt tokenus atbilstoši norādītajiem parametriem.

> **Note:**
> Tikai *Super admin* lietotāja tips drīkst 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:**
> Lietotāja lomai ir nepieciešama *Pārvaldīt API tokenus* atļauja, lai pārvaldītu citu lietotāju tokenus.

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 atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet 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. Tiesības izsaukt šo metodi var atsaukt 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. Tiesības izsaukt šo metodi var atsaukt 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)`

Šī metode ļauj izgūt trigeru prototipus atbilstoši norādītajiem parametriem.

> **Note:**
> Šī metode ir pieejama jebkura tipa lietotājiem. Tiesības izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet User roles.

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. Tiesības izsaukt šo metodi var atsaukt lietotāju lomu iestatījumos. Plašāku informāciju 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. Tiesības izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā 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. Tiesības izsaukt šo metodi var atsaukt 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. Tiesības izsaukt šo metodi var atsaukt lietotāju lomu iestatījumos. Plašāku informāciju skatiet sadaļā 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)`

Šī metode ļauj izgūt trigerus atbilstoši norādītajiem parametriem.

> **Note:**
> Šī metode ir pieejama jebkura tipa lietotājiem. Tiesības izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā Lietotāja lomas.

Parametri

`(object)` Parametri, kas nosaka vēlamo izvadi.

Metode atbalsta šādus parametrus.

Parametrs	Tips	Apraksts
triggerids	ID/array	Atgriež tikai trigerus ar norādītajiem ID.
groupids	ID/array	Atgriež tikai trigerus, kas pieder hostiem vai veidnēm no norādītajām hostu grupām vai veidņu grupām.
templateids	ID/array	Atgriež tikai trigerus, kas pieder norādītajām veidnēm.
hostids	ID/array	Atgriež tikai trigerus, kas pieder norādītajiem hostiem.
itemids	ID/array	Atgriež tikai trigerus, kas satur norādītos vienumus.
maintenanceids	ID/array	Atgriež tikai trigerus, kurus ietekmē norādītās uzturēšanas darbības.
functions	string/array	Atgriež tikai trigerus, kuros tiek izmantotas norādītās funkcijas. Atbalstīto funkciju sarakstu skatiet lapā Atbalstītās funkcijas.
group	string	Atgriež tikai trigerus, kas pieder hostiem vai veidnēm no hostu grupas vai veidņu grupas ar norādīto nosaukumu.
host	string	Atgriež tikai trigerus, kas pieder hostam ar norādīto tehnisko nosaukumu.
inherited	boolean	Ja iestatīts uz `true`, atgriež tikai no veidnes mantotos trigerus.
templated	boolean	Ja iestatīts uz `true`, atgriež tikai trigerus, kas pieder veidnēm.
dependent	boolean	Ja iestatīts uz `true`, atgriež tikai trigerus, kuriem ir atkarības. Ja iestatīts uz `false`, atgriež tikai trigerus, kuriem nav atkarību.

Parametrs	Tips	Apraksts
monitored	flag	Atgriež tikai iespējotus trigerus, kas pieder uzraudzītiem hostiem un satur tikai iespējotus vienumus.
active	flag	Atgriež tikai iespējotus trigerus, kas pieder uzraudzītiem hostiem.
maintenance	boolean	Ja iestatīts uz `true`, atgriež tikai iespējotus trigerus, kas pieder uzturēšanā esošiem hostiem.
withUnacknowledgedEvents	flag	Atgriež tikai trigerus, kuriem ir neapstiprināti notikumi.
withAcknowledgedEvents	flag	Atgriež tikai trigerus, kuriem visi notikumi ir apstiprināti.
withLastEventUnacknowledged	flag	Atgriež tikai trigerus, kuru pēdējais notikums nav apstiprināts.
skipDependent	flag	Izlaiž problēmas stāvoklī esošus trigerus, kas ir atkarīgi no citiem trigeriem. Ņemiet vērā, ka pārējie trigeri tiek ignorēti, ja tie ir atspējoti, tiem ir atspējoti vienumi vai tie pieder hostiem ar atspējotiem vienumiem.
lastChangeSince	timestamp	Atgriež tikai trigerus, kuru stāvoklis ir mainījies pēc norādītā laika.
lastChangeTill	timestamp	Atgriež tikai trigerus, kuru stāvoklis ir mainījies pirms norādītā laika.
only_true	flag	Atgriež tikai trigerus, kas nesen ir bijuši problēmas stāvoklī (pēdējā `ok_period` laikā).
min_severity	integer	Atgriež tikai trigerus, kuru nopietnības līmenis ir lielāks vai vienāds ar norādīto līmeni.
evaltype	integer	Tagu novērtēšanas metode. Iespējamās vērtības: 0 - *(noklusējums)* And/Or; 2 - Or.
tags	array	Atgriež tikai trigerus ar norādītajiem tagiem. Formāts: `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`. Tukšs masīvs atgriež visus trigerus. Iespējamās operatora vērtības: 0 - *(noklusējums)* Satur; 1 - Ir vienāds ar; 2 - Nesatur; 3 - Nav vienāds ar; 4 - Pastāv; 5 - Nepastāv.
expandComment	flag	Izvērš makro trigeru aprakstā.
expandDescription	flag	Izvērš makro trigeru nosaukumā.
expandExpression	flag	Izvērš funkcijas un makro trigeru izteiksmē.
selectHostGroups	query	Atgriež hostu grupas, kurām trigeris pieder, rekvizītā `hostgroups`.
selectHosts	query	Atgriež hostus, kuriem trigeris pieder, rekvizītā `hosts`.
selectItems	query	Atgriež trigerī ietvertos vienumus rekvizītā `items`.
selectFunctions	query	Atgriež trigerī izmantotās funkcijas rekvizītā `functions`. Funkciju objekti attēlo trigeru izteiksmē izmantotās funkcijas, un tiem ir šādi rekvizīti: `functionid` - `(ID)` funkcijas ID; `itemid` - `(ID)` funkcijā izmantotā vienuma ID; `function` - `(string)` funkcijas nosaukums; `parameter` - `(string)` funkcijai nodotais parametrs. Atgrieztajā virknē vaicājuma parametrs tiek aizstāts ar simbolu $.
selectDependencies	query	Atgriež trigerus, no kuriem trigeris ir atkarīgs, rekvizītā `dependencies`.

Parametrs	Tips	Apraksts
selectDiscoveryData	query	Atgriež rekvizītu `discoveryData` ar trigeru atklāšanas objekta datiem. Trigeru atklāšanas objekts sasaista atklāto trigeri ar trigeru prototipu, no kura tas tika atklāts. Tam ir šādi rekvizīti: `parent_triggerid` - `(ID)` trigeru prototipa ID, no kura tika izveidots trigeris; `status` - `(int)` trigeru atklāšanas statuss: 0 - *(noklusējums)* trigeris ir atklāts, 1 - trigeris vairs nav atklāts; `ts_delete` - `(timestamp)` laiks, kad vairs neatklātais trigeris tiks dzēsts; `ts_disable` - `(timestamp)` laiks, kad vairs neatklātais trigeris tiks atspējots; `disable_source` - `(int)` indikators, kas norāda, vai trigeri atspējoja LLD noteikums vai manuāli: 0 - *(noklusējums)* atspējots automātiski, 1 - atspējojis LLD noteikums.
selectDiscoveryRule	query	Atgriež zemā līmeņa atklāšanas noteikumu, kas izveidoja trigeri, rekvizītā `discoveryRule`.
selectLastEvent	query	Atgriež pēdējo nozīmīgo trigeru notikumu rekvizītā `lastEvent`.
selectTags	query	Atgriež trigeru tagus rekvizītā `tags`.
inheritedTags	boolean	Atgriež trigerus, kuriem norādītie `tags` ir arī veidnē, hostā vai saistītajās veidnēs. Iespējamās vērtības: `true` - veidnēm, hostam vai saistītajām veidnēm arī jābūt norādītajiem tagiem; `false` - *(noklusējums)* veidņu, hosta vai saistīto veidņu tagi tiek ignorēti.
selectInheritedTags	query	Atgriež `inheritedTags` rekvizītu ar tagiem, kas mantoti no veidnēm, saistītajām veidnēm un hostiem, kā arī no vienumiem, uz kuriem ir atsauces trigeru vai atkopšanas izteiksmēs.
selectTemplateGroups	query	Atgriež veidņu grupas, kurām trigeris pieder, rekvizītā `templategroups`.
filter	object	Atgriež tikai tos rezultātus, kas precīzi atbilst norādītajam filtram. Pieņem objektu, kurā atslēgas ir rekvizītu nosaukumi, bet vērtības ir viena vērtība vai vērtību masīvs, ar ko veikt salīdzināšanu. Neatbalsta `text` datu tipa rekvizītus. Atbalsta papildu rekvizītus: `host` - hosta, kuram pieder trigeris, tehniskais nosaukums; `hostid` - hosta, kuram pieder trigeris, ID.
limitSelects	integer	Ierobežo apakšatlases atgriezto ierakstu skaitu. Attiecas uz šādām apakšatlasēm: `selectHosts` - rezultāti tiks kārtoti pēc `host`.
sortfield	string/array	Kārto rezultātu pēc norādītajiem rekvizītiem. 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	Atgriež trigeru atklāšanas objektu rekvizītā `triggerDiscovery`. Trigeru atklāšanas objekti sasaista trigeri ar trigeru prototipu, no kura tas tika izveidots. Šis vaicājums ir **novecojis**; 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 šādi rekvizīti.

Property	Type	Description
httptestid	ID	Tīmekļa scenārija ID. Rekvizīta darbība: - *tikai lasāms* - *obligāts* atjaunināšanas darbībām
hostid	ID	hosts ID, kuram pieder tīmekļa scenārijs. Rekvizīta darbība: - *konstants* - *obligāts* izveides darbībām
name	string	Tīmekļa scenārija nosaukums. Rekvizīta 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 sekunžu skaitu (piemēram, 1, 60, 300), vērtību ar laika sufiksu (piemēram, 30s, 1m, 2h, 1d) vai lietotāja makro. Noklusējums: 1m.
headers	array	HTTP galvenes, kas tiks nosūtītas, veicot pieprasījumu.
http_password	string	Parole, ko izmanto pamata HTTP, NTLM, Kerberos vai Digest autentifikācijai.
http_proxy	string	Starpniekserveris, ko izmantos tīmekļa scenārijs, norādīts formātā *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 darbību pirms kļūmes. Noklusējums: 1.
ssl_cert_file	string	SSL sertifikāta faila nosaukums, ko izmanto klienta autentifikācijai (tam jābūt PEM formātā).
ssl_key_file	string	SSL privātās atslēgas faila nosaukums, ko izmanto klienta autentifikācijai (tam 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.

Property	Type	Description
templateid	ID	Vecākās veidnes tīmekļa scenārija ID. Rekvizīta darbība: - *tikai lasāms*
variables	array	Tīmekļa scenārija mainīgie.
verify_host	integer	Vai pārbaudīt, ka savienojuma hosts nosaukums atbilst hosts sertifikātā norādītajam nosaukumam. Iespējamās vērtības: 0 - *(noklusējums)* izlaist hosts pārbaudi; 1 - pārbaudīt hosts.
verify_peer	integer	Vai pārbaudīt, ka hosts sertifikāts ir autentisks. Iespējamās vērtības: 0 - *(noklusējums)* izlaist partnera pārbaudi; 1 - pārbaudīt partneri.
uuid	string	Globāli unikāls identifikators, ko izmanto importēto tīmekļa scenāriju sasaistīšanai ar jau esošajiem scenārijiem. Ja nav norādīts, tiek ģenerēts automātiski. Rekvizīta darbība: - *atbalstīts*, ja tīmekļa scenārijs pieder veidnei

Scenārija solis

Scenārija soļa objekts definē konkrētu tīmekļa scenārija pārbaudi.
Tam ir tālāk norādītās īpašības.

Property	Type	Description
name	string	Scenārija soļa nosaukums. Property behavior: - *required*
no	integer	Soļa kārtas numurs tīmekļa scenārijā. Property behavior: - *required*
url	string	Pārbaudāmais URL. Property behavior: - *required*
follow_redirects	integer	Vai sekot HTTP novirzēm. Iespējamās vērtības: 0 - nesekot novirzēm; 1 - *(noklusējums)* sekot novirzēm.
headers	array	HTTP galvenes, kas tiks nosūtītas, izpildot pieprasījumu. Scenārija soļa galvenes aizstās tīmekļa scenārijam norādītās galvenes.
posts	string/array	HTTP POST mainīgie kā virkne (neapstrādāti POST dati) vai masīvs ar HTTP laukiem (veidlapas lauku dati).
required	string	Teksts, kam jābūt ietvertam atbildē.
retrieve_mode	integer	HTTP atbildes daļa, kas jāiegūst scenārija solim. Iespējamās vērtības: 0 - *(noklusējums)* tikai pamatteksts; 1 - tikai galvenes; 2 - galvenes un pamatteksts.
status_codes	string	Nepieciešamo HTTP statusa kodu diapazoni, atdalīti ar komatiem.

Property	Type	Description
timeout	string	Pieprasījuma taimauts sekundēs. Pieņem sekunžu skaitu (piemēram, 1, 60, 300), vērtību ar laika sufiksu (piemēram, 30s, 1m, 1h) vai lietotāja makro. Noklusējums: 15s. Maksimums: 1h. Minimums: 1s.
variables	array	Scenārija soļa mainīgie.
query_fields	array	Vaicājuma lauki - masīvs ar HTTP laukiem, kas tiks pievienoti 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, automātiski tiks izveidots tīmekļa uzraudzības vienumu kopums.

Note:
Šī metode ir pieejama tikai *Admin* un *Super admin* lietotāju tipiem. Tiesības izsaukt šo metodi var atsaukt lietotāju lomu iestatījumos. Plašāku informāciju 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. Tiesības izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā Lietotāja 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)`

Šī metode ļauj izgūt tīmekļa scenārijus atbilstoši norādītajiem parametriem.

> **Note:**
> Šī metode ir pieejama jebkura tipa lietotājiem. Tiesības 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
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.

Parametrs	Tips	Apraksts
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	
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. Tiesības izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā 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"

Īpašība	Tips	Apraksts
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*
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. Tiesības izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāja 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)`

Šī metode ļauj izgūt uzdevumus atbilstoši norādītajiem parametriem.

> **Note:**
> Šī metode ir pieejama tikai *Super admin* lietotāja tipam. Atļaujas izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet User roles.

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 objekta īpašības ir šādas.

Property	Type	Description
maintenanceid	ID	Uzturēšanas ID. Property behavior: - *read-only* - *required* for update operations
name	string	Uzturēšanas nosaukums. Property behavior: - *required* for create operations

Property	Type	Description
active_since	timestamp	Laiks, kad uzturēšana kļūst aktīva (ieskaitot). Norādītā vērtība tiks noapaļota uz leju līdz minūtēm. Property behavior: - *required* for create operations
active_till	timestamp	Laiks, kad uzturēšana pārstāj būt aktīva (neieskaitot). Norādītā vērtība tiks noapaļota uz leju līdz minūtēm. Property behavior: - *required* for create operations
description	string	Uzturēšanas apraksts.
maintenance_type	integer	Uzturēšanas veids. Iespējamās vērtības: 0 - *(noklusējums)* ar datu vākšanu; 1 - bez datu vākšanas.
tags_evaltype	integer	Problēmu tagu izvērtēšanas 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”

Īpašība	Tips	Apraksts
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”
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

Īpašība	Tips	Apraksts
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”

Notikuma nosaukums

Notikuma nosaukuma objekts tiek izmantots, lai definētu, kuras problēmas ir jāapslāpē, kad stājas spēkā uzturēšana. Notikumu nosaukumus var norādīt tikai tad, ja Maintenance objekta `maintenance_type` ir iestatīts uz ”with data collection”. Tam ir šādas īpašības.

Property	Type	Description
operator	integer	Nosacījuma operators. Iespējamās vērtības: 2 - *(noklusējums)* Satur; 3 - Nesatur.
value	string	Notikuma nosaukuma vērtība. Īpašības darbība: - *obligāts*

Problēmas tags

Problēmas taga objekts tiek izmantots, lai definētu, kuras problēmas ir jāapslāpē, kad stājas spēkā uzturēšana. Tagus var norādīt tikai tad, ja Uzturēšanas objekta `maintenance_type` ir iestatīts uz ”with data collection”. Tam ir šādi rekvizīti.

Property	Type	Description
tag	string	Problēmas taga nosaukums. Rekvizīta darbība: - *obligāts*
operator	integer	Nosacījuma operators. Iespējamās vērtības: 0 - Vienāds; 1 - Nav vienāds; 2 - *(noklusējums)* Satur; 3 - Nesatur.
value	string	Problēmas taga vērtība.

maintenance.create

Apraksts

`object maintenance.create(object/array maintenances)`

Šī metode ļauj izveidot jaunus uzturēšanas darbus.

> **Note:**
> Šī metode ir pieejama tikai *Admin* un *Super admin* lietotāju tipiem. Tiesības izsaukt šo metodi var atsaukt lietotāju lomu iestatījumos. Plašāku informāciju skatiet sadaļā Lietotāju lomas.

Parametri

`(object/array)` Izveidojamās uzturēšanas.

Papildus standarta uzturēšanas rekvizītiem metode pieņem šādus parametrus.

Parameter	Type	Description
groups	object/array	Hosts grupas, kuras jāiekļauj uzturēšanā. Hosts grupām drīkst būt definēts tikai rekvizīts `groupid`. Parametra darbība: - *obligāts*, ja `hosts` nav iestatīts
hosts	object/array	Hosts, kuri jāiekļauj uzturēšanā. Hosts drīkst būt definēts tikai rekvizīts `hostid`. Parametra darbība: - *obligāts*, ja `groups` nav iestatīts
triggers	object/array	Trigeri, kuri jāiekļauj uzturēšanā. Trigeriem drīkst būt definēts tikai rekvizīts `triggerid`. Parametra darbība: - *obligāts*, ja `groups` un `hosts` nav iestatīti - *atbalstīts*, ja maintenance_type=0 (ar datu apkopošanu)
event_names	object/array	Notikumu nosaukumi, kas nosaka problēmas, kuras jāiekļauj uzturēšanā. Parametra darbība: - *atbalstīts*, ja maintenance_type=0 (ar datu apkopošanu)
tags	object/array	Problēmu tagi, kas nosaka problēmas, kuras jāiekļauj uzturēšanā. Parametra darbība: - *atbalstīts*, ja Maintenance object rekvizītam `maintenance_type` ir iestatīta vērtība "ar datu apkopošanu"
timeperiods	object/array	Uzturēšanas laika periodi. Parametra darbība: - *obligāts*

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur izveidoto uzturēšanu ID īpašībā `maintenanceids`. Atgriezto ID secība atbilst nodoto uzturēšanu secībai.

Piemēri

Apkopes izveide

Izveidojiet apkopi ar datu apkopošanu hostu grupai ar ID "20" un 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 jāilgst vienu stundu.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "maintenance.create",
    "params": {
        "name": "Sunday maintenance",
        "active_since": 1773720240,
        "active_till": 1805256240,
        "tags_evaltype": 0,
        "groups": [
            {"groupid": "20"}
        ],
        "timeperiods": [
            {
                "period": 3600,
                "timeperiod_type": 3,
                "start_time": 64800,
                "every": 1,
                "dayofweek": 64
            }
        ],
        "tags": [
            {
                "tag": "service",
                "operator": "0",
                "value": "mysqld"
            },
            {
                "tag": "error",
                "operator": "2",
                "value": ""
            }
        ]
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "maintenanceids": [
            "3"
        ]
    },
    "id": 1
}
```

Ad-hoc uzturēšanas izveide

Izveidojiet vienu stundu ilgu ad-hoc uzturēšanu ar datu apkopošanu hosts grupai ar ID ”4”, trigerim ar ID ”12345” un notikuma nosaukumam **Printer: ink low**.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "maintenance.create",
    "params": {
        "name": "Ad-hoc maintenance",
        "active_since": 1792057740,
        "active_till": 1792061340,
        "tags_evaltype": 0,
        "groups": [
            {"groupid": "4"}
```

```
        ],
        "triggers": [
            {"triggerid": "12345"}
        ],
        "timeperiods": [
            {
                "period": 3600,
                "timeperiod_type": 0,
                "start_date": 1792057740
            }
        ],
        "event_names": [
            {
                "operator": "2",
                "value": "Printer: ink low"
            }
        ]
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "maintenanceids": [
            "4"
        ]
    },
    "id": 1
}
```

Skatiet arī

- Hosts
- Hostu grupa
- Trigeris
- Notikuma nosaukums
- Problēmas tags
- 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. Tiesības izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā User roles.

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)`

Šī metode ļauj izgūt uzturēšanas darbus atbilstoši norādītajiem parametriem.

> **Note:**
> Šī metode ir pieejama jebkura tipa lietotājiem. Tiesības izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā Lietotāja lomas.

Parametri

`(object)` Parametri, kas nosaka vēlamo izvadi.

Metode atbalsta šādus parametrus.

Parametrs	Tips	Apraksts
groupids	ID/array	Atgriezt tikai uzturēšanas periodus, kas piešķirti norādītajām hostu grupām.
hostids	ID/array	Atgriezt tikai uzturēšanas periodus, kas piešķirti norādītajiem hostiem.
triggerids	ID/array	Atgriezt tikai uzturēšanas periodus, kas piešķirti norādītajiem trigeriem.
maintenanceids	ID/array	Atgriezt tikai uzturēšanas periodus ar norādītajiem ID.
selectHostGroups	query	Atgriezt `hostgroups` rekvizītu ar uzturēšanas periodam piešķirtajām hostu grupām.
selectHosts	query	Atgriezt `hosts` rekvizītu ar uzturēšanas periodam piešķirtajiem hostiem.
selectTriggers	query	Atgriezt `triggers` rekvizītu ar uzturēšanas periodam piešķirtajiem trigeriem.
selectEventNames	query	Atgriezt `event_names` rekvizītu ar uzturēšanas periodam piešķirtajiem notikumu nosaukumiem.
selectTags	query	Atgriezt `tags` rekvizītu ar uzturēšanas perioda problēmu tagiem.

Parametrs	Tips	Apraksts
selectTimeperiods	query	Atgriezt `timeperiods` rekvizītu ar uzturēšanas perioda laika periodiem.
sortfield	string/array	Kārtot rezultātu pēc norādītajiem rekvizītiem. 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, datus par piešķirto uzturēšanas tvērumu (hostu grupa, hosts, trigeris, notikuma nosaukums, birkas) un definētajiem laika periodiem.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "maintenance.get",
    "params": {
        "output": "extend",
        "selectHostGroups": "extend",
        "selectHosts": "extend",
        "selectTriggers": "extend",
        "selectEventNames": "extend",
        "selectTags": "extend",
        "selectTimeperiods": "extend"
    },
    "id": 1
}
```

Atbilde:

```
{
  "jsonrpc": "2.0",
  "result": [
    {
      "maintenanceid": "1",
      "name": "Regulāra DB uzturēšana",
      "maintenance_type": "0",
      "description": "",
      "active_since": "1791936000",
      "active_till": "2107555200",
      "tags_evaltype": "0",
      "hostgroups": [
        {
```

```
        "groupid": "20",
        "name": "Datubāzes",
        "flags": "0",
        "uuid": "748ad4d098d447d492bb935c907f652f"
      }
    ],
    "triggers": [],
    "event_names": [],
    "hosts": [],
    "tags": [],
    "timeperiods": [
      {
        "timeperiod_type": "4",
        "every": "1",
        "month": "4095",
        "dayofweek": "0",
        "day": "1",
        "start_time": "64800",
        "period": "3600",
        "start_date": "0"
      }
    ]
  },
  {
    "maintenanceid": "2",
    "name": "Ad hoc: Printer: tinte BEIGUSIES",
    "maintenance_type": "0",
    "description": "",
    "active_since": "1792057740",
    "active_till": "1792061340",
    "tags_evaltype": "0",
    "hostgroups": [
      {
        "groupid": "4",
        "name": "Zabbix serveri",
        "flags": "0",
        "uuid": "6f6799aa69e844b4b3918f779f2abf08"
      }
    ],
    "triggers": [],
    "event_names": [
      {
        "operator": "2",
        "value": "Printer: tinte BEIGUSIES"
      }
    ],
    "hosts": [],
    "tags": [],
    "timeperiods": [
      {
        "timeperiod_type": "0",
        "every": "1",
        "month": "0",
        "dayofweek": "0",
        "day": "0",
        "start_time": "0",
        "period": "3600",
        "start_date": "1792057740"
      }
    ]
  }
],
```

```
  "id": 1
}
```

Skatiet arī

- Hosts
- Hostu grupa
- Trigeris
- Notikuma nosaukums
- Problēmas tags
- 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 apkalpošanas.

Note:
Šī metode ir pieejama tikai *Admin* un *Super admin* lietotāju tipiem. Tiesības izsaukt šo metodi var atsaukt lietotāju lomu iestatījumos. Plašāku informāciju skatiet sadaļā Lietotāju lomas.

Parametri

`(object/array)` Uzturēšanas rekvizīti, kas jāatjaunina.

Katram uzturēšanas periodam ir jābūt definētam rekvizītam `maintenanceid`, visi pārējie rekvizīti nav obligāti. Tiks atjaunināti tikai nodotie rekvizīti, bet visi pārējie paliks nemainīti.

Papildus standarta uzturēšanas rekvizītiem metode pieņem šādus parametrus.

Parameter	Type	Description
groups	object/array	Hosts grupas, ar kurām aizstāt pašreizējās grupas. Hosts grupām drīkst būt definēts tikai rekvizīts `groupid`. Parametra darbība: - *obligāts*, ja `hosts` nav iestatīts
hosts	object/array	Hosts, ar kuriem aizstāt pašreizējos hosts. Hosts drīkst būt definēts tikai rekvizīts `hostid`. Parametra darbība: - *obligāts*, ja `groups` nav iestatīts
triggers	object/array	Trigeri, ar kuriem aizstāt pašreizējos trigerus. Trigeriem drīkst būt definēts tikai rekvizīts `triggerid`. Parametra darbība: - *obligāts*, ja `groups` un `hosts` nav iestatīti - *atbalstīts*, ja maintenance_type=0 (ar datu apkopošanu)
event_names	object/array	Notikumu nosaukumi, ar kuriem aizstāt pašreizējos notikumu nosaukumus. Parametra darbība: - *atbalstīts*, ja maintenance_type=0 (ar datu apkopošanu)

Parameter	Type	Description
tags	object/array	Problēmu tagi, ar kuriem aizstāt pašreizējos tagus. Parametra darbība: - *atbalstīts*, ja Maintenance object rekvizīts `maintenance_type` ir iestatīts uz ”ar datu apkopošanu”
timeperiods	object/array	Uzturēšanas laikposmi, ar kuriem aizstāt pašreizējos laikposmus.

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*.

Īpašība	Tips	Apraksts
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. Tiesības izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā User roles.

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, tiek noņemtas arī visas tās entītijas (vienumi, trigeri, grafiki u. c.) no jebkuriem hostiem vai veidnēm, ar kurām tā ir saistīta. Lai saglabātu šīs entītijas, 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 lietotāju tipiem *Admin* un *Super admin*. Tiesības izsaukt šo metodi var atsaukt 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)`

Šī metode ļauj izgūt veidnes atbilstoši norādītajiem parametriem.

Note:
Šī metode ir pieejama jebkura tipa lietotājiem. Tiesības izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā Lietotāja 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.
evaltype	integer	Birkas novērtēšanas metode. Iespējamās vērtības: 0 - *(noklusējums)* Un/Vai; 2 - Vai.

Parametrs	Tips	Apraksts
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. Tiesības izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā User roles.

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. Tiesības izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā 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. Tiesības izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā 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 jaunus veidņu grupas.

Note:
Šī metode ir pieejama tikai *Super admin* lietotāja tipam. Tiesības izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet User roles.

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. Tiesības izsaukt metodi var atsaukt lietotāju lomu iestatījumos. Plašāku informāciju skatiet 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)`

Šī metode ļauj izgūt veidņu grupas atbilstoši norādītajiem parametriem.

Note:
Šī metode ir pieejama jebkura tipa lietotājiem. Tiesības 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
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 saistītus objektus visām norādītajām veidņu grupām.

> **Note:**
> Šī metode ir pieejama tikai *Admin* un *Super admin* lietotāju tipiem. Tiesības izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Papildinformāciju skatiet sadaļā Lietotāja 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. Tiesības izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet 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 atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā Lietotāju lomas.

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. Tiesības izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā 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 informācijas panelis

Veidnes informācijas paneļa objektam ir šādas īpašības.

Property	Type	Description
dashboardid	ID	Veidnes informācijas paneļa ID. Īpašības uzvedība: - *tikai lasāms* - *obligāts* atjaunināšanas operācijām
name	string	Veidnes informācijas paneļa nosaukums. Īpašības uzvedība: - *obligāts* izveides operācijām

Property	Type	Description
templateid	ID	Veidnes ID, kurai pieder informācijas panelis. Īpašības uzvedība: - *konstants* - *obligāts* izveides operācijām
display_period	integer	Noklusējuma slaidrādes intervāls (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 veidnes informācijas paneļu sasaistīšanai ar jau esošajiem. Ja nav norādīts, tiek ģenerēts automātiski.

Veidnes informācijas paneļa lapa

Veidnes informācijas paneļa lapas objektam ir šādas īpašības.

Property	Type	Description
dashboard_pageid	ID	Informācijas paneļa lapas ID. Property behavior: - *read-only*
name	string	Informācijas paneļa lapas nosaukums. Noklusējums: tukša virkne.
display_period	integer	Informācijas paneļa slaidrādes intervāls (sekundēs). Iespējamās vērtības: 0, 10, 30, 60, 120, 600, 1800, 3600. Noklusējums: 0 (tiks izmantots noklusējuma slaidrādes intervāls).
widgets	array	veidnes informācijas 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 veidnes informācijas paneļus.

Note:
Šī metode ir pieejama tikai *Admin* un *Super admin* lietotāju tipiem. Tiesības izsaukt šo metodi var atsaukt 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 informācijas paneļus.

> **Note:**
> Šī metode ir pieejama tikai *Admin* un *Super admin* lietotāju tipiem. Tiesības izsaukt šo metodi var atsaukt lietotāju lomu 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)`

Šī metode ļauj izgūt veidnes paneļus atbilstoši norādītajiem parametriem.

Note:
Šī metode ir pieejama jebkura tipa lietotājiem. Tiesības izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā Lietotāja 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 veidnes informācijas paneļus.

> **Note:**
> Šī metode ir pieejama tikai *Admin* un *Super admin* lietotāju tipiem. Tiesības izsaukt šo metodi var atsaukt 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 šādi rekvizīti.

Rekvizīts	Tips	Apraksts
itemid	ID	Vienuma prototipa ID. Rekvizīta darbība: - *tikai lasāms* - *obligāts* atjaunināšanas darbībām
delay	string	Vienuma prototipa atjaunināšanas intervāls. Pieņem sekunžu skaitu (piemēram, 1, 60, 300), vērtību ar laika sufiksu (piemēram, 30s, 1m, 2h, 1d) vai, pēc izvēles, vienu vai vairākus pielāgotus intervālus, kas atdalīti ar semikoliem. Pielāgotie intervāli var būt elastīgo un plānošanas intervālu kombinācija. Pieņem lietotāja makro un LLD makro. Ja tiek izmantots makro, vērtībai jābūt vienam makro. Vairāki makro vai makro kombinācijā ar tekstu netiek atbalstīti. Elastīgos intervālus var norādīt kā divus makro, 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 Rekvizīta 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), ”Telemetry query” (24) vai ja type ir iestatīts uz ”Zabbix agent (active)” (7) un key_ nesatur ”mqtt.get”
hostid	ID	Tā hosta ID, kuram pieder vienuma prototips. Rekvizīta darbība: - *konstants* - *obligāts* izveides darbībām
interfaceid	ID	Vienuma prototipa hosta saskarnes ID. Rekvizīta 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. Rekvizīta darbība: - *obligāts* izveides darbībām - *tikai lasāms* mantotiem objektiem
name	string	Vienuma prototipa nosaukums. Atbalsta lietotāja makro. Rekvizīta darbība: - *obligāts* izveides darbībām - *tikai lasāms* mantotiem objektiem

Rekvizīts	Tips	Apraksts
type	integer	Vienuma prototipa tips. Iespējamās vērtības: 0 - Zabbix agent; 2 - Zabbix trapper; 3 - Simple check; 5 - Zabbix internal; 7 - Zabbix agent (active); 10 - External check; 11 - Database monitor; 12 - IPMI agent; 13 - SSH agent; 14 - TELNET agent; 15 - Calculated; 16 - JMX agent; 17 - SNMP trap; 18 - Dependent item; 19 - HTTP agent; 20 - SNMP agent; 21 - Script; 22 - Browser; 24 - Telemetry query. Rekvizīta darbība: - *obligāts* izveides darbībām - *tikai lasāms* mantotiem objektiem
url	string	URL virkne. Atbalsta LLD makro, lietotāja makro, {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {HOST.PORT}, {ITEM.ID}, {ITEM.KEY}. Rekvizīta darbība: - *obligāts*, ja `type` ir iestatīts uz ”HTTP agent” - *tikai lasāms* mantotiem objektiem
value_type	integer	Vienuma prototipa informācijas tips. Iespējamās vērtības: 0 - numeric float; 1 - character; 2 - log; 3 - numeric unsigned; 4 - text; 5 - binary; 6 - JSON. Rekvizīta darbība: - *obligāts* izveides darbībām - *tikai lasāms* mantotiem objektiem
allow_traps	integer	Atļaut vērtību aizpildīt līdzīgi kā trapper vienumam. 0 - *(noklusējums)* nepieņemt ienākošos datus; 1 - pieņemt ienākošos datus. Rekvizīta darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”HTTP agent”

Rekvizīts	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)* none; 1 - basic; 2 - NTLM; 3 - Kerberos; 4 - Digest. Rekvizīta darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”SSH agent” vai ”HTTP agent” - *tikai lasāms* mantotiem 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 Rekvizīta darbība: - *tikai lasāms*
follow_redirects	integer	Sekot atbildes novirzēm datu aptaujas laikā. Iespējamās vērtības: 0 - nesekot novirzēm; 1 - *(noklusējums)* sekot novirzēm. Rekvizīta darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”HTTP agent” - *tikai lasāms* mantotiem objektiem
headers	array	Galveņu masīvs, kas tiks nosūtīts, veicot HTTP pieprasījumu. Rekvizīta darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”HTTP agent” - *tikai lasāms* mantotiem objektiem
history	string	Laika vienība, kas nosaka, cik ilgi jāglabā vēstures dati. Pieņem arī lietotāja makro un LLD makro. Noklusējums: 31d.
http_proxy	string	HTTP(S) starpniekservera savienojuma virkne. Rekvizīta darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”HTTP agent” - *tikai lasāms* mantotiem objektiem
ipmi_sensor	string	IPMI sensors. Rekvizīta 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* mantotiem objektiem

Rekvizīts	Tips	Apraksts
jmx_endpoint	string	JMX aģenta pielāgotā savienojuma virkne. Noklusējums: service:jmx:rmi:///jndi/rmi://{HOST.CONN}:{HOST.PORT}/jmxrmi Rekvizīta darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”JMX agent”
logtimefmt	string	Laika formāts žurnāla ierakstos. Rekvizīta darbība: - *atbalstīts*, ja `value_type` ir iestatīts uz ”log” - *tikai lasāms* mantotiem objektiem
master_itemid	ID	Galvenā vienuma ID. Rekvizīta darbība: - *obligāts*, ja `type` ir iestatīts uz ”Dependent item” - *tikai lasāms* mantotiem objektiem
output_format	integer	Vai atbilde jāpārveido par JSON. Iespējamās vērtības: 0 - *(noklusējums)* saglabāt neapstrādātus datus; 1 - pārveidot par JSON. Rekvizīta darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”HTTP agent” - *tikai lasāms* mantotiem objektiem
params	string	Papildu parametri atkarībā no vienuma prototipa tipa: - izpildāmais skripts SSH agent un TELNET agent vienumu prototipiem; - SQL vaicājums datubāzes pārraudzības vienumu prototipiem; - formula aprēķināto vienumu prototipiem; - skripts skripta un pārlūkprogrammas vienumu prototipiem. Rekvizīta darbība: - *obligāts*, ja `type` ir iestatīts uz ”Database monitor”, ”SSH agent”, ”TELNET agent”, ”Calculated”, ”Script” vai ”Browser” - *tikai lasāms* mantotiem 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` rekvizītiem, kur `name` jābūt unikālam. Rekvizīta darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”Script” vai ”Browser” - *tikai lasāms* mantotiem objektiem
password	string	Autentifikācijas parole. Rekvizīta 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* mantotiem objektiem (ja `type` ir iestatīts uz ”HTTP agent”)
post_type	integer	`posts` rekvizītā saglabāto POST datu pamatteksta tips. Iespējamās vērtības: 0 - *(noklusējums)* neapstrādāti dati. 2 - JSON dati. 3 - XML dati. Rekvizīta darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”HTTP agent” - *tikai lasāms* mantotiem objektiem

Rekvizīts	Tips	Apraksts
posts	string	HTTP(S) pieprasījuma pamatteksta dati. Rekvizīta 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* mantotiem objektiem
privatekey	string	Privātās atslēgas faila nosaukums. Rekvizīta darbība: - *obligāts*, ja `type` ir iestatīts uz ”SSH agent” un `authtype` ir iestatīts uz ”public key”
publickey	string	Publiskās atslēgas faila nosaukums. Rekvizīta 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, veicot HTTP pieprasījumu. Rekvizīta darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”HTTP agent” - *tikai lasāms* mantotiem objektiem
request_method	integer	Pieprasījuma metodes tips. Iespējamās vērtības: 0 - *(noklusējums)* GET; 1 - POST; 2 - PUT; 3 - HEAD. Rekvizīta darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”HTTP agent” - *tikai lasāms* mantotiem objektiem
retrieve_mode	integer	Kura atbildes daļa jāglabā. Iespējamās vērtības, ja `request_method` ir iestatīts uz ”GET”, ”POST” vai ”PUT”: 0 - *(noklusējums)* pamatteksts; 1 - galvenes; 2 - tiks glabāts gan pamatteksts, gan galvenes. Iespējamā vērtība, ja `request_method` ir iestatīts uz ”HEAD”: 1 - galvenes. Rekvizīta darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”HTTP agent” - *tikai lasāms* mantotiem objektiem
snmp_oid	string	SNMP OID. Rekvizīta darbība: - *obligāts*, ja `type` ir iestatīts uz ”SNMP agent” - *tikai lasāms* mantotiem objektiem
ssl_cert_file	string	Publiskās SSL atslēgas faila ceļš. Rekvizīta darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”HTTP agent” - *tikai lasāms* mantotiem objektiem

Rekvizīts	Tips	Apraksts
ssl_key_file	string	Privātās SSL atslēgas faila ceļš. Rekvizīta darbība: - *atbalstīts*, ja `type` ir iestatīts uz "HTTP agent" - *tikai lasāms* mantotiem objektiem
ssl_key_password	string	SSL atslēgas faila parole. Rekvizīta darbība: - *atbalstīts*, ja `type` ir iestatīts uz "HTTP agent" - *tikai lasāms* mantotiem 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 makro vai LLD makro kā daļu no komatiem atdalīta saraksta. Piemērs: 200,200-{$M},{$M},200-400 Rekvizīta darbība: - *atbalstīts*, ja `type` ir iestatīts uz "HTTP agent" - *tikai lasāms* mantotiem objektiem
templateid	ID	Vecākās veidnes vienuma prototipa ID. Rekvizīta darbība: - *tikai lasāms*
timeout	string	Vienuma datu aptaujas pieprasījuma taimauts. Pieņem sekunžu skaitu (piemēram, 1, 60, 300) vai vērtību ar laika sufiksu (piemēram, 30s, 1m). Pieņem arī lietotāja makro un LLD makro. Iespējamo vērtību diapazons: 1-600s. Noklusējums: "" - izmantot starpniekservera/globālos iestatījumus. Rekvizīta 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* mantotiem objektiem
trapper_hosts	string	Atļautie hosti. Rekvizīta darbība: - *atbalstīts*, ja `type` ir iestatīts uz "Zabbix trapper" vai ja `type` ir iestatīts uz "HTTP agent" un `allow_traps` ir iestatīts uz "Allow incoming data". Ja rekvizīts nav norādīts, tam automātiski tiek iestatīts `{$TRAPPER.ALLOWED_HOSTS}` lietotāja makro (noklusējuma vērtība). Pēc noklusējuma `{$TRAPPER.ALLOWED_HOSTS}` ir `127.0.0.1,::1`. Ja rekvizīts ir iestatīts uz tukšu virkni, ienākošie savienojumi no visiem hostiem tiks noraidīti.

Rekvizīts	Tips	Apraksts
trends	string	Laika vienība, kas nosaka, cik ilgi jāglabā tendenču dati. Pieņem arī lietotāja makro un LLD makro. Noklusējums: 365d. Rekvizīta darbība: - *atbalstīts*, ja `value_type` ir iestatīts uz ”numeric float” vai ”numeric unsigned”
units	string	Vērtības mērvienības. Rekvizīta darbība: - *atbalstīts*, ja `value_type` ir iestatīts uz ”numeric float” vai ”numeric unsigned” - *tikai lasāms* mantotiem objektiem
username	string	Lietotājvārds autentifikācijai. Rekvizīta 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* mantotiem objektiem (ja `type` ir iestatīts uz ”HTTP agent”)
uuid	string	Universāls unikālais identifikators, ko izmanto importēto vienumu prototipu sasaistīšanai ar jau esošajiem. Ja nav norādīts, tiek ģenerēts automātiski. Rekvizīta darbība: - *atbalstīts*, ja vienuma prototips pieder veidnei
valuemapid	ID	Saistītās vērtību kartes ID. Rekvizīta darbība: - *atbalstīts*, ja `value_type` ir iestatīts uz ”numeric float”, ”character” vai ”numeric unsigned” - *tikai lasāms* mantotiem objektiem
verify_host	integer	Vai pārbaudīt, ka savienojuma hosta nosaukums atbilst hosta sertifikātā norādītajam. Iespējamās vērtības: 0 - *(noklusējums)* nepārbaudīt; 1 - pārbaudīt. Rekvizīta darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”HTTP agent” - *tikai lasāms* mantotiem objektiem
verify_peer	integer	Vai pārbaudīt, ka hosta sertifikāts ir autentisks. Iespējamās vērtības: 0 - *(noklusējums)* nepārbaudīt; 1 - pārbaudīt. Rekvizīta darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”HTTP agent” - *tikai lasāms* mantotiem 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.

Rekvizīts	Tips	Apraksts
query	object	Telemetrijas vaicājuma konfigurācija. Nosaka, ko vienums vaicā telemetrijas datu avotam, kā tas filtrē datus un kā apkopo rezultātu. Rekvizīta darbība: - *obligāts*, ja `type` ir iestatīts uz ”Telemetry query” - *tikai lasāms* mantotiem vai atklātiem objektiem
time_shift	string	Cik tālu pagātnē (pirms pašreizējā laika) beidzas laika diapazons, par kuru vienums veic vaicājumu. Pieņem sekunžu skaitu (piemēram, 1, 60, 300) vai vērtību ar laika sufiksu (piemēram, `30s`, `1m`, `2h`, `1d`). Pieņem arī lietotāja makro un LLD makro. Iespējamo vērtību diapazons: 0-86400s. Noklusējums: 15s. Rekvizīta darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”Telemetry query”
lookback_limit	string	Cik tālu pagātnē (pirms pašreizējā laika) vienums meklē datus, ja nav iepriekšēja vaicājuma, no kura turpināt (piemēram, vienuma pirmajā izpildes reizē), vai ja iepriekšējais vaicājums ir vecāks par `delay` (piemēram, tāpēc, ka Zabbix serveris nedarbojās). Pieņem sekunžu skaitu (piemēram, 1, 60, 300) vai vērtību ar laika sufiksu (piemēram, `30s`, `1m`, `2h`, `1d`). Pieņem arī lietotāja makro un LLD makro. Iespējamo vērtību diapazons: 1-259200s. Noklusējums: 10m. Rekvizīta darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”Telemetry query”
granularity	string	Vienas daļas garums, kurā tiek sadalīts viss vaicājuma laika diapazons. Pieņem sekunžu skaitu (piemēram, 1, 60, 300) vai vērtību ar laika sufiksu (piemēram, `30s`, `1m`, `2h`, `1d`). Pieņem arī lietotāja makro un LLD makro. Iespējamo vērtību diapazons: 1-86400s. Noklusējums: 15s. Rekvizīta darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”Telemetry query”

HTTP galvene

Galvenes objektam ir šādas īpašības:

Īpašība	Tips	Apraksts
name	string	HTTP galvenes nosaukums. Īpašības darbība: - *obligāta*

Īpašība	Tips	Apraksts
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*

Telemetrijas vaicājums

Telemetrijas vaicājuma objektam ir šādi rekvizīti.

Property	Type	Description
signal_type	integer	Signāla tips, ko vaicā vienums. Iespējamās vērtības: 0 - Traces; 1 - Metrics; 2 - Logs. Rekvizīta darbība: - *obligāts*
metric_point_type	integer	Metrikas punkta tips, ko vaicā vienums. Iespējamās vērtības: 0 - Sum; 1 - Gauge; 2 - Histogram; 3 - Exponential histogram. Rekvizīta darbība: - *obligāts*, ja `signal_type` ir iestatīts uz ”Metrics”
columns	array	Telemetrijas datu kolonnu masīvs, ko vaicā vienums. Var būt tukšs. Rekvizīta darbība: - *obligāts*
aggregated_columns	array	Telemetrijas datu apkopoto kolonnu masīvs, ko vaicā vienums, un veids, kā tas aprēķina vērtību no kolonnu ierakstiem. Rekvizīta darbība: - *obligāts*
filter	object	Telemetrijas vaicājuma filtrs vaicājuma kolonnu ierakstiem. Rekvizīta darbība: - *obligāts*

Telemetrijas datu kolonnas

Telemetrijas datu kolonnu objektam ir šādas īpašības.

Property	Type	Description
column	string	Telemetrijas datu kolonnas nosaukums, ko vaicā vienums. Skatiet atbalstītās telemetrijas datu kolonnas. Vairākām vienas vērtības kolonnām katrai `column` vērtībai ir jābūt unikālai. Vairākām kolonnām, kurās tiek glabāti atslēgu un vērtību pāri, katram `column` un `attribute_key` pārim ir jābūt unikālam.
attribute_key	string	Atslēgas nosaukums, ko vaicā vienums kolonnā, kurā tiek glabāti atslēgu un vērtību pāri. Vairākām kolonnām, kurās tiek glabāti atslēgu un vērtību pāri, katram `column` un `attribute_key` pārim ir jābūt unikālam. Īpašības darbība: - *obligāts*, ja `column` ir iestatīts uz kolonnu, kurā tiek glabāti atslēgu un vērtību pāri

Telemetrijas datu apkopoto kolonnu objekts

Telemetrijas datu apkopoto kolonnu objektam ir šādi rekvizīti.

Property	Type	Description
column	string	Telemetrijas datu kolonnas nosaukums, ko vaicā vienums. Skatiet atbalstītās telemetrijas datu kolonnas. Ja tiek izmantotas vairākas kolonnas, katrai `column` vērtībai ir jābūt unikālai. Šis rekvizīts netiek atbalstīts, ja `function` ir iestatīts uz ”Count”. Rekvizīta darbība: - *obligāts*, ja `function` ir iestatīts uz ”Min”, ”Max”, ”Average”, ”Sum” vai ”Percentile”
function	integer	Funkcija, ko vienums lieto kolonnas ierakstiem. Iespējamās vērtības: 1 - Min; 2 - Max; 3 - Average; 4 - Count; 5 - Sum; 8 - Percentile. Rekvizīta darbība: - *obligāts*
parameters	array	Procentile, ko vienums aprēķina no kolonnas ierakstiem. Iespējamā vērtība: skaitlis no 0 līdz 100, kas norādīts kā virkne. Rekvizīta darbība: - *obligāts*, ja `function` ir iestatīts uz ”Percentile”
alias	string	Unikāls aprēķinātās vērtības nosaukums, ko izmanto, lai to identificētu vienuma vērtībā. Katrai `alias` vērtībai ir jābūt unikālai. Rekvizīta darbība: - *obligāts*

Telemetrijas datu filtrs

Telemetrijas datu filtra objektam ir šādas īpašības.

Property	Type	Description
conditions	array	Telemetrijas datu filtra nosacījumiem jāatbilst ierakstam telemetrijas datu kolonnā. Īpašības darbība: - *obligāts*
evaltype	integer	Filtra nosacījumu novērtēšanas metode. Iespējamās vērtības: 0 - And/Or; 1 - And; 2 - Or; 3 - Pielāgota izteiksme. Īpašības darbība: - *obligāts*
eval_formula	string	Ģenerēta izteiksme, kas tiks izmantota filtra nosacījumu novērtēšanai. Izteiksme satur ID, kas atsaucas uz konkrētiem filtra nosacījumiem, izmantojot to `formulaid`. `eval_formula` vērtība ir vienāda ar `formula` vērtību filtriem ar pielāgotu izteiksmi. Īpašības darbība: - *tikai lasāms*
formula	string	Lietotāja definēta izteiksme, kas jāizmanto filtru ar pielāgotu izteiksmi nosacījumu novērtēšanai. Izteiksmei jāsatur ID, kas atsaucas uz konkrētiem filtra nosacījumiem, izmantojot to `formulaid`. Izteiksmē izmantotajiem ID ir precīzi jāatbilst filtra nosacījumos definētajiem ID: neviens nosacījums nedrīkst palikt neizmantots vai izlaists. Īpašības darbība: - *obligāts*, ja `evaltype` ir iestatīts uz “Pielāgota izteiksme”

Telemetrijas datu filtra nosacījumi

Telemetrijas datu filtra nosacījumu objektam ir šādas īpašības.

Property	Type	Description
column	string	Telemetrijas datu kolonnas nosaukums, uz kuru attiecas filtra nosacījums. Skatiet atbalstītās telemetrijas datu kolonnas. Īpašības darbība: - *obligāta*
attribute_key	string	Atslēgas nosaukums, uz kuru attiecas nosacījums, kolonnā, kurā ir atslēgu un vērtību pāri. Īpašības darbība: - *obligāta*, ja `column` ir iestatīta uz kolonnu, kurā ir atslēgu un vērtību pāri

Property	Type	Description
operator	integer	Filtra nosacījuma operators. Iespējamās vērtības kolonnām ar vienu vērtību: 0 - Vienāds; 1 - Nav vienāds; 2 - Satur; 3 - Nesatur. Iespējamās vērtības kolonnām, kurās ir atslēgu un vērtību pāri: 0 - Vienāds; 1 - Nav vienāds; 12 - Pastāv. Īpašības darbība: - *obligāta*
value	string	Vērtība, ar kuru salīdzināt. Īpašības darbība: - *obligāta*, ja `operator` ir iestatīts uz ”Vienāds”, ”Nav vienāds”, ”Satur” vai ”Nesatur”
formulaid	string	Patvaļīgs unikāls ID, kas tiek izmantots, lai atsauktos uz nosacījumu pielāgotā izteiksmē. Var saturēt tikai lielos burtus. ID lietotājam ir jādefinē, modificējot filtra nosacījumus, taču, tos vēlāk pieprasot, tas tiks ģenerēts no jauna. Īpašības darbība: - *obligāta*, ja Telemetrijas vaicājuma filtra objekta `evaltype` ir iestatīts uz ”Pielāgota izteiksme”

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 šādi rekvizīti.

Rekvizīts	Tips	Apraksts
type	integer	Priekšapstrādes opcijas tips. Iespējamās vērtības: 1 - Pielāgots reizinātājs; 2 - Labās puses apgriešana; 3 - Kreisās puses apgriešana; 4 - Apgriešana; 5 - Regulārā izteiksme; 6 - Būla vērtības pārveidošana decimālskaitlī; 7 - Oktālās sistēmas pārveidošana decimālajā sistēmā; 8 - Heksadecimālās sistēmas pārveidošana decimālajā sistēmā; 9 - Vienkāršas izmaiņas; 10 - Izmaiņas sekundē; 11 - XML XPath; 12 - JSONPath; 13 - Diapazonā; 14 - Atbilst regulārajai izteiksmei; 15 - Neatbilst regulārajai izteiksmei; 16 - JSON kļūdas pārbaude; 17 - XML kļūdas pārbaude; 18 - Kļūdas pārbaude, izmantojot regulāro izteiksmi; 19 - Neizmainītā atmešana; 20 - Neizmainītā atmešana ar sirdspukstu; 21 - JavaScript; 22 - Prometheus raksts; 23 - Prometheus pārveidošana JSON; 24 - CSV pārveidošana JSON; 25 - Aizstāšana; 26 - Neatbalstīta vienuma pārbaude; 27 - XML pārveidošana JSON; 28 - SNMP walk vērtība; 29 - SNMP walk pārveidošana JSON; 30 - SNMP get vērtība. Rekvizīta darbība: - *obligāts*
params	string	Papildu parametri, ko izmanto priekšapstrādes opcija. Vairāki parametri tiek atdalīti ar jaunas rindas (\n) rakstzīmi. Ja `type` ir iestatīts uz “Neatbalstīta vienuma pārbaude”, parametri atbilst sintaksei `<scope>[\n<pattern>]`, kur *pattern* ir regulāra izteiksme, bet *scope* ir viena no šīm vērtībām: -1 - atbilst jebkurai kļūdai; 0 - pārbaudīt, vai kļūdas ziņojums atbilst *pattern*; 1 - pārbaudīt, vai kļūdas ziņojums neatbilst *pattern*. Rekvizīta darbība: - *obligāts*, ja `type` ir iestatīts uz “Pielāgots reizinātājs” (1), “Labās puses apgriešana” (2), “Kreisās puses apgriešana” (3), “Apgriešana” (4), “Regulārā izteiksme” (5), “XML XPath” (11), “JSONPath” (12), “Diapazonā” (13), “Atbilst regulārajai izteiksmei” (14), “Neatbilst regulārajai izteiksmei” (15), “JSON kļūdas pārbaude” (16), “XML kļūdas pārbaude” (17), “Kļūdas pārbaude, izmantojot regulāro izteiksmi” (18), “Neizmainītā atmešana ar sirdspukstu” (20), “JavaScript” (21), “Prometheus raksts” (22), “Prometheus pārveidošana JSON” (23), “CSV pārveidošana JSON” (24), “Aizstāšana” (25), “Neatbalstīta vienuma pārbaude” (26), “SNMP walk vērtība” (28), “SNMP walk pārveidošana JSON” (29) vai “SNMP get vērtība” (30)

Rekvizīts	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 “Neatbalstīta vienuma pārbaude”: 1 - Atmest vērtību; 2 - Iestatīt pielāgotu vērtību; 3 - Iestatīt pielāgotu kļūdas ziņojumu. Rekvizīta darbība: - *obligāts*, ja `type` ir iestatīts uz “Pielāgots reizinātājs” (1), “Regulārā izteiksme” (5), “Būla vērtības pārveidošana decimālskaitlī” (6), “Oktālās sistēmas pārveidošana decimālajā sistēmā” (7), “Heksadecimālās sistēmas pārveidošana decimālajā sistēmā” (8), “Vienkāršas izmaiņas” (9), “Izmaiņas sekundē” (10), “XML XPath” (11), “JSONPath” (12), “Diapazonā” (13), “Atbilst regulārajai izteiksmei” (14), “Neatbilst regulārajai izteiksmei” (15), “JSON kļūdas pārbaude” (16), “XML kļūdas pārbaude” (17), “Kļūdas pārbaude, izmantojot regulāro izteiksmi” (18), “Prometheus raksts” (22), “Prometheus pārveidošana JSON” (23), “CSV pārveidošana JSON” (24), “Neatbalstīta vienuma pārbaude” (26), “XML pārveidošana JSON” (27), “SNMP walk vērtība” (28), “SNMP walk pārveidošana JSON” (29) vai “SNMP get vērtība” (30)
error_handler_params	string	Kļūdu apstrādātāja parametri. Rekvizīta darbība: - *obligāts*, 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	number^1, 6^			0, 1, 2, 3
2	Labās puses apgriešana	rakstzīmju saraksts[2]			
3	Kreisās puses apgriešana	rakstzīmju saraksts[2]			
4	Apgriešana	rakstzīmju saraksts[2]			
5	Regulārā izteiksme	pattern[3]	output[2]		0, 1, 2, 3
6	Būla vērtības pārveidošana decimālskaitlī				0, 1, 2, 3

Priekšapstrādes tips	Nosaukums	1. parametrs	2. parametrs	3. parametrs	Atbalstītie kļūdu apstrādātāji
7	Oktālās sistēmas pārveidošana decimālajā sistēmā				0, 1, 2, 3
8	Heksadecimālās sistēmas pārveidošana decimālajā sistēmā				0, 1, 2, 3
9	Vienkāršas izmaiņas				0, 1, 2, 3
10	Izmaiņas sekundē				0, 1, 2, 3
11	XML XPath	path[4]			0, 1, 2, 3
12	JSONPath	path[4]			0, 1, 2, 3
13	Diapazona	min^1, 6^	max^1, 6^		0, 1, 2, 3
14	Atbilst regulārajai izteiksmei	pattern[3]			0, 1, 2, 3
15	Neatbilst regulārajai izteiksmei	pattern[3]			0, 1, 2, 3
16	JSON kļūdas pārbaude	path[4]			0, 1, 2, 3
17	XML kļūdas pārbaude	path[4]			0, 1, 2, 3
18	Kļūdas pārbaude, izmantojot regulāro izteiksmi	pattern[3]	output[2]		0, 1, 2, 3
19	Neizmainītā atmešana				

Priekšapstrādes tips	Nosaukums	1. parametrs	2. parametrs	3. parametrs	Atbalstītie kļūdu apstrādātāji
20	Neizmainītā atmešana ar sirdspukstu	sekonds^5, 6^			
21	JavaScript	script[2]			
22	Prometheus raksts	pattern^6, 7^	`value`, `label`, `function`	output^8, 9^	0, 1, 2, 3
23	Prometheus pārveidošana JSON	pattern^6, 7^			0, 1, 2, 3
24	CSV pārveidošana JSON	character[2]	character[2]	0,1	0, 1, 2, 3
25	Aizstāšana	search string[2]	replacement[2]		
26	Neatbalstītā vienuma pārbaude	stāpe[1]	pattern^3, 6^		1, 2, 3
27	XML pārveidošana 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 - Integer no BITS		0, 1, 2, 3
29	SNMP walk pārveidošana JSON[10]	Field name[2]	OID prefix[2]	Formāts: 0 - Nemainīts 1 - UTF-8 no Hex-STRING 2 - MAC no Hex-STRING 3 - Integer no BITS	0, 1, 2, 3
30	SNMP get vērtība	Formāts: 1 - UTF-8 no Hex-STRING 2 - MAC no Hex-STRING 3 - Integer no BITS			0, 1, 2, 3

[1] vesels vai peldošā komata skaitlis
 [2] virkne
 [3] regulāra 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 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 vai LLD makross.
 [8] Prometheus izvade atbilstoši sintaksei: `<label name>` (var būt lietotāja makross vai LLD makross), ja kā otrais parametrs ir atlasīts `label`.
 [9] Viena no apkopošanas funkcijām: `sum`, `min`, `max`, `avg`, `count`, ja kā otrais parametrs ir atlasīts `function`.
 [10] Atbalsta vairākus “Field name,OID prefix,Format” ierakstus, kas atdalīti ar jaunas 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. Tiesības izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā 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. Tiesības izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā Lietotāju lomas.

Parametri

`(object/array)` Atjaunināmo 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. Tiesības izsaukt šo metodi var atsaukt lietotāju lomu iestatījumos. Plašāku informāciju skatiet sadaļā 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)`

Šī metode ļauj izgūt vienumu prototipus atbilstoši norādītajiem parametriem.

Note:
Šī metode ir pieejama jebkura tipa lietotājiem. Tiesības izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā User roles.

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.

Parametrs	Tips	Apraksts
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.
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	

Parametrs	Tips	Apraksts
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 šādi rekvizīti.

Rekvizīts	Tips	Apraksts
itemid	ID	Vienuma ID. Rekvizīta darbība: - *tikai lasāms* - *obligāts* atjaunināšanas darbībām

Rekvizīts	Tips	Apraksts
delay	string	Vienuma atjaunināšanas intervāls. Pieņem sekunžu skaitu (piemēram, 1, 60, 300), vērtību ar laika sufiksu (piemēram, 30s, 1m, 2h, 1d) vai, pēc izvēles, vienu vai vairākus pielāgotus intervālus, kas atdalīti ar semikoliem. Pielāgotie intervāli var būt elastīgo un plānošanas intervālu kombinācija. Pieņem lietotāja makro. Ja tiek izmantots makro, vērtībai jābūt vienam makro. Vairāki makro vai makro kopā ar tekstu netiek atbalstīti. Elastīgos intervālus var norādīt kā divus makro, 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 Rekvizīta 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), "Telemetry query" (24) vai ja type ir iestatīts uz "Zabbix agent (active)" (7) un key_ nesatur "mqtt.get"
hostid	ID	Hosts vai veidnes, kurai pieder vienums, ID. Rekvizīta darbība: - *konstants* - *obligāts* izveides darbībām
interfaceid	ID	Vienuma hosts interfeisa ID. Rekvizīta 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. Rekvizīta darbība: - *obligāta* izveides darbībām - *tikai lasāma* mantotajiem vai atklātajiem objektiem
name	string	Vienuma nosaukums. Atbalsta lietotāja makro. Rekvizīta darbība: - *obligāts* izveides darbībām - *tikai lasāms* mantotajiem vai atklātajiem objektiem
name_resolved	string	Vienuma nosaukums ar aizstātiem lietotāja makro. Rekvizīta darbība: - *tikai lasāms*

Rekvizīts	Tips	Apraksts
type	integer	Vienuma tips. Iespējamās vērtības: 0 - Zabbix agent; 2 - Zabbix trapper; 3 - Simple check; 5 - Zabbix internal; 7 - Zabbix agent (active); 9 - Web item; 10 - External check; 11 - Database monitor; 12 - IPMI agent; 13 - SSH agent; 14 - TELNET agent; 15 - Calculated; 16 - JMX agent; 17 - SNMP trap; 18 - Dependent item; 19 - HTTP agent; 20 - SNMP agent; 21 - Script; 22 - Browser; 24 - Telemetry query. Rekvizīta darbība: - *obligāts* izveides darbībām - *tikai lasāms* mantotajiem vai atklātajiem objektiem
url	string	URL virkne. Atbalsta lietotāja makro, {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {HOST.PORT}, {ITEM.ID}, {ITEM.KEY}. Rekvizīta darbība: - *obligāts*, ja `type` ir iestatīts uz "HTTP agent" - *tikai lasāms* mantotajiem vai atklātajiem objektiem
value_type	integer	Vienuma informācijas tips. Iespējamās vērtības: 0 - numeric float; 1 - character; 2 - log; 3 - numeric unsigned; 4 - text; 5 - binary; 6 - JSON. Rekvizīta darbība: - *obligāts* izveides darbībām - *tikai lasāms* mantotajiem vai atklātajiem objektiem
allow_traps	integer	Atļaut vērtības aizpildīšanu tāpat kā trapper vienumam. 0 - *(noklusējums)* nepieņemt ienākošos datus; 1 - pieņemt ienākošos datus. Rekvizīta darbība: - *atbalstīts*, ja `type` ir iestatīts uz "HTTP agent" - *tikai lasāms* atklātajiem objektiem

Rekvizīts	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. Rekvizīta 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. Rekvizīta darbība: - *tikai lasāms* atklātajiem objektiem
error	string	Kļūdas teksts, ja rodas problēmas ar vienuma vērtības atjaunināšanu. Rekvizīta darbība: - *tikai lasāms*
flags	integer	Vienuma izcelsme. Iespējamās vērtības: 0 - parasts vienums; 4 - no prototipa pārveidots vienums. Rekvizīta darbība: - *tikai lasāms*
follow_redirects	integer	Sekot atbildes novirzēm datu aptaujas laikā. Iespējamās vērtības: 0 - nesekot novirzēm; 1 - *(noklusējums)* sekot novirzēm. Rekvizīta darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”HTTP agent” - *tikai lasāms* mantotajiem vai atklātajiem objektiem
headers	array	Galveņu masīvs, kas tiks nosūtīts, veicot HTTP pieprasījumu. Rekvizīta darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”HTTP agent” - *tikai lasāms* mantotajiem vai atklātajiem objektiem
history	string	Laika vienība, kas nosaka, cik ilgi jāglabā vēstures dati. Pieņem arī lietotāja makro. Noklusējums: 31d. Rekvizīta darbība: - *tikai lasāms* atklātajiem objektiem
http_proxy	string	HTTP(S) starpniekservera savienojuma virkne. Rekvizīta darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”HTTP agent” - *tikai lasāms* mantotajiem vai atklātajiem objektiem

Rekvizīts	Tips	Apraksts
inventory_link	integer	Hosts inventarizācijas lauka ID, kuru aizpilda vienums. Atbalstīto hosts inventarizācijas lauku un to ID sarakstu skatiet hosts inventarizācijas lapā. Noklusējums: 0. Rekvizīta darbība: - *atbalstīts*, ja `value_type` ir iestatīts uz ”numeric float”, ”character”, ”numeric unsigned” vai ”text” - *tikai lasāms* atklātajiem objektiem
ipmi_sensor	string	IPMI sensors. Rekvizīta 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 vai atklātajiem objektiem
jmx_endpoint	string	JMX aģenta pielāgotā savienojuma virkne. Noklusējuma vērtība: service:jmx:rmi:///jndi/rmi://{HOST.CONN}:{HOST.PORT}/jmxrmi Rekvizīta darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”JMX agent” - *tikai lasāms* atklātajiem objektiem
lastclock	timestamp	Laiks, kad vienuma vērtība tika pēdējo reizi atjaunināta. Pēc noklusējuma tiek rādītas tikai vērtības, kas ietilpst pēdējās 24 stundās. Šo periodu var pagarināt, mainot parametra *Max history display period* vērtību izvēlnes sadaļā *Administration > General*. Rekvizīta darbība: - *tikai lasāms*
lastns	integer	Nanosekundes, kad vienuma vērtība tika pēdējo reizi atjaunināta. Pēc noklusējuma tiek rādītas tikai vērtības, kas ietilpst pēdējās 24 stundās. Šo periodu var pagarināt, mainot parametra *Max history display period* vērtību izvēlnes sadaļā *Administration > General*. Rekvizīta darbība: - *tikai lasāms*
lastvalue	string	Pēdējā vienuma vērtība. Pēc noklusējuma tiek rādītas tikai vērtības, kas ietilpst pēdējās 24 stundās. Šo periodu var pagarināt, mainot parametra *Max history display period* vērtību izvēlnes sadaļā *Administration > General*. Rekvizīta darbība: - *tikai lasāms*
logtimefmt	string	Laika formāts žurnāla ierakstos. Rekvizīta darbība: - *atbalstīts*, ja `value_type` ir iestatīts uz ”log” - *tikai lasāms* mantotajiem vai atklātajiem objektiem
master_itemid	ID	Galvenā vienuma ID. Rekvizīta darbība: - *obligāts*, ja `type` ir iestatīts uz ”Dependent item” - *tikai lasāms* mantotajiem vai atklātajiem objektiem

Rekvizīts	Tips	Apraksts
output_format	integer	Vai atbilde jāpārveido par JSON. 0 - *(noklusējums)* saglabāt neapstrādātus datus; 1 - pārveidot par JSON. Rekvizīta darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”HTTP agent” - *tikai lasāms* mantotajiem vai atklātajiem objektiem
params	string	Papildu parametri atkarībā no vienuma tipa: - izpildāmais skripts SSH agent un TELNET agent vienumiem; - SQL vaicājums database monitor vienumiem; - formula calculated vienumiem; - skripts script un browser vienumiem. Rekvizīta 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` rekvizītiem, kur `name` jābūt unikālam. Rekvizīta darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”Script” vai ”Browser” - *tikai lasāms* mantotajiem vai atklātajiem objektiem
password	string	Parole autentifikācijai. Rekvizīta darbība: - *obligāta*, ja `type` ir iestatīts uz ”JMX agent” un `username` ir iestatīts - *atbalstīta*, ja `type` ir iestatīts uz ”Simple check”, ”SSH agent”, ”TELNET agent”, ”Database monitor” vai ”HTTP agent” - *tikai lasāma* mantotajiem objektiem (ja `type` ir iestatīts uz ”HTTP agent”) vai atklātajiem objektiem
post_type	integer	`posts` rekvizītā saglabāto POST datu pamatteksta tips. Iespējamās vērtības: 0 - *(noklusējums)* neapstrādāti dati; 2 - JSON dati; 3 - XML dati. Rekvizīta darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”HTTP agent” - *tikai lasāms* mantotajiem vai atklātajiem objektiem
posts	string	HTTP(S) pieprasījuma pamatteksta dati. Rekvizīta darbība: - *obligāti*, ja `type` ir iestatīts uz ”HTTP agent” un `post_type` ir iestatīts uz ”JSON data” vai ”XML data” - *atbalstīti*, ja `type` ir iestatīts uz ”HTTP agent” un `post_type` ir iestatīts uz ”Raw data” - *tikai lasāmi* mantotajiem vai atklātajiem objektiem
prevvalue	string	Iepriekšējā vienuma vērtība. Pēc noklusējuma tiek rādītas tikai vērtības, kas ietilpst pēdējās 24 stundās. Šo periodu var pagarināt, mainot parametra *Max history display period* vērtību izvēlnes sadaļā *Administration > General*. Rekvizīta darbība: - *tikai lasāms*

Rekvizīts	Tips	Apraksts
privatekey	string	Privātās atslēgas faila nosaukums. Rekvizīta 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. Rekvizīta 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, veicot HTTP pieprasījumu. Rekvizīta darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”HTTP agent” - *tikai lasāms* mantotajiem vai atklātajiem objektiem
request_method	integer	Pieprasījuma metodes tips. Iespējamās vērtības: 0 - *(noklusējums)* GET; 1 - POST; 2 - PUT; 3 - HEAD. Rekvizīta darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”HTTP agent” - *tikai lasāms* mantotajiem vai atklātajiem objektiem
retrieve_mode	integer	Kura atbildes daļa jāglabā. Iespējamās vērtības, ja `request_method` ir iestatīts uz ”GET”, ”POST” vai ”PUT”: 0 - *(noklusējums)* pamatteksts; 1 - galvenes; 2 - tiks saglabāts gan pamatteksts, gan galvenes. Iespējamā vērtība, ja `request_method` ir iestatīts uz ”HEAD”: 1 - galvenes. Rekvizīta darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”HTTP agent” - *tikai lasāms* mantotajiem vai atklātajiem objektiem
snmp_oid	string	SNMP OID. Rekvizīta darbība: - *obligāts*, ja `type` ir iestatīts uz ”SNMP agent” - *tikai lasāms* mantotajiem vai atklātajiem objektiem
ssl_cert_file	string	Publiskās SSL atslēgas faila ceļš. Rekvizīta darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”HTTP agent” - *tikai lasāms* mantotajiem vai atklātajiem objektiem
ssl_key_file	string	Privātās SSL atslēgas faila ceļš. Rekvizīta darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”HTTP agent” - *tikai lasāms* mantotajiem vai atklātajiem objektiem
ssl_key_password	string	Parole SSL atslēgas failam. Rekvizīta darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”HTTP agent” - *tikai lasāms* mantotajiem vai atklātajiem objektiem

Rekvizīts	Tips	Apraksts
state	integer	Vienuma stāvoklis. Iespējamās vērtības: 0 - *(noklusējums)* normāls; 1 - netiek atbalstīts. Rekvizīta 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 makro kā daļu no ar komatiem atdalīta saraksta. Piemērs: 200,200-{$M},{$M},200-400 Rekvizīta darbība: - *atbalstīts*, ja `type` ir iestatīts uz "HTTP agent" - *tikai lasāms* mantotajiem vai atklātajiem objektiem
templateid	ID	Vecākās veidnes vienuma ID. *Padoms*: izmantojiet `hostid` rekvizītu, lai norādītu veidni, kurai pieder vienums. Rekvizīta darbība: - *tikai lasāms*
timeout	string	Vienuma datu aptaujas pieprasījuma noildze. Pieņem sekunžu skaitu (piemēram, 1, 60, 300) vai vērtību ar laika sufiksu (piemēram, 300s, 1m). Pieņem arī lietotāja makro. Iespējamo vērtību diapazons: 1-600s. Noklusējums: "" - izmantot starpniekservera/globālos iestatījumus. Rekvizīta 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, - ja tiek lietota noklusējuma vērtība un vienums pieder hostam, kuru uzrauga nepieejams starpniekserveris vai starpniekserveru grupa, tiek izmantota globālā noildze.
trapper_hosts	string	Atļautie hosti. Rekvizīta 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 incoming data". Ja rekvizīts nav norādīts, tam automātiski tiek iestatīts `{$TRAPPER.ALLOWED_HOSTS}` lietotāja makro (noklusējuma vērtība). Pēc noklusējuma `{$TRAPPER.ALLOWED_HOSTS}` ir `127.0.0.1,::1`. Ja rekvizīts ir iestatīts uz tukšu virkni, ienākošie savienojumi no visiem hostiem tiks noraidīti.

Rekvizīts	Tips	Apraksts
trends	string	Laika vienība, kas nosaka, cik ilgi jāglabā tendenču dati. Pieņem arī lietotāja makro. Noklusējums: 365d. Rekvizīta darbība: - *atbalstīts*, ja `value_type` ir iestatīts uz ”numeric float” vai ”numeric unsigned” - *tikai lasāms* atklātajiem objektiem
units	string	Vērtības mērvienības. Rekvizīta darbība: - *atbalstīts*, ja `value_type` ir iestatīts uz ”numeric float” vai ”numeric unsigned” - *tikai lasāms* mantotajiem vai atklātajiem objektiem
username	string	Lietotājvārds autentifikācijai. Rekvizīta 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 importētā vienuma sasaistīšanai ar jau esošiem vienumiem. Ja nav norādīts, tiek ģenerēts automātiski. Rekvizīta darbība: - *atbalstīts*, ja vienums pieder veidnei
valuemapid	ID	Saistītās vērtību kartes ID. Rekvizīta darbība: - *atbalstīts*, ja `value_type` ir iestatīts uz ”numeric float”, ”character” vai ”numeric unsigned” - *tikai lasāms* mantotajiem vai atklātajiem objektiem
verify_host	integer	Vai pārbaudīt, ka savienojuma hosts nosaukums atbilst hosts sertifikātā norādītajam. Iespējamās vērtības: 0 - *(noklusējums)* nepārbaudīt; 1 - pārbaudīt. Rekvizīta darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”HTTP agent” - *tikai lasāms* mantotajiem vai atklātajiem objektiem
verify_peer	integer	Vai pārbaudīt, ka hosts sertifikāts ir autentisks. Iespējamās vērtības: 0 - *(noklusējums)* nepārbaudīt; 1 - pārbaudīt. Rekvizīta darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”HTTP agent” - *tikai lasāms* mantotajiem vai atklātajiem objektiem
query	object	Telemetrijas vaicājuma konfigurācija. Nosaka, ko vienums vaicā telemetrijas datu avotā, kā tas filtrē datus un kā apkopo rezultātu. Rekvizīta darbība: - *obligāts*, ja `type` ir iestatīts uz ”Telemetry query” - *tikai lasāms* mantotajiem vai atklātajiem objektiem

Rekvizīts	Tips	Apraksts
time_shift	string	Cik tālu pagātnē (pirms pašreizējā laika) beidzas laika diapazons, par kuru vienums veic vaicājumu. Pieņem sekunžu skaitu (piemēram, 1, 60, 300) vai vērtību ar laika sufiksu (piemēram, 30s, 1m, 2h, 1d). Pieņem arī lietotāja makro. Iespējamo vērtību diapazons: 0-86400s. Noklusējums: 15s. Rekvizīta darbība: - *atbalstīts*, ja `type` ir iestatīts uz "Telemetry query"
lookback_limit	string	Cik tālu pagātnē (pirms pašreizējā laika) vienums veic datu vaicājumu, ja nav iepriekšēja vaicājuma, no kura turpināt (piemēram, vienuma pirmās palaišanas reizē), vai ja iepriekšējais vaicājums ir vecāks par `delay` (piemēram, tāpēc, ka Zabbix serveris nedarbojās). Pieņem sekunžu skaitu (piemēram, 1, 60, 300) vai vērtību ar laika sufiksu (piemēram, 30s, 1m, 2h, 1d). Pieņem arī lietotāja makro. Iespējamo vērtību diapazons: 1-259200s. Noklusējums: 10m. Rekvizīta darbība: - *atbalstīts*, ja `type` ir iestatīts uz "Telemetry query"
granularity	string	Tā gabala ilgums, kurā tiek sadalīts viss vaicājuma laika diapazons. Pieņem sekunžu skaitu (piemēram, 1, 60, 300) vai vērtību ar laika sufiksu (piemēram, 30s, 1m, 2h, 1d). Pieņem arī lietotāja makro. Iespējamo vērtību diapazons: 1-86400s. Noklusējums: 15s. Rekvizīta darbība: - *atbalstīts*, ja `type` ir iestatīts uz "Telemetry query"

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*

Telemetrijas vaicājums

Telemetrijas vaicājuma objektam ir šādi rekvizīti.

Rekvizīts	Tips	Apraksts
signal_type	integer	Signāla tips, ko vaicā vienums. Iespējamās vērtības: 0 - Traces; 1 - Metrics; 2 - Logs. Rekvizīta darbība: - *obligāts*
metric_point_type	integer	Metrikas punkta tips, ko vaicā vienums. Iespējamās vērtības: 0 - Sum; 1 - Gauge; 2 - Histogram; 3 - Exponential histogram. Rekvizīta darbība: - *obligāts*, ja `signal_type` ir iestatīts uz ”Metrics”
columns	array	Telemetrijas datu kolonnu masīvs, ko vaicā vienums. Var būt tukšs. Rekvizīta darbība: - *obligāts*
aggregated_columns	array	Telemetrijas datu apkopoto kolonnu masīvs, ko vaicā vienums, un veids, kā tas aprēķina vērtību no kolonnu ierakstiem. Rekvizīta darbība: - *obligāts*
filter	object	Telemetrijas vaicājuma filtrs vaicājuma kolonnu ierakstiem. Rekvizīta darbība: - *obligāts*

Telemetrijas datu kolonnas

Telemetrijas datu kolonnu objektam ir šādas īpašības.

Property	Type	Description
column	string	Telemetrijas datu kolonnas nosaukums, ko vaicā vienums. Skatiet atbalstītās telemetrijas datu kolonnas. Vairākām vienas vērtības kolonnām katrai `column` vērtībai ir jābūt unikālai. Vairākām kolonnām, kurās tiek glabāti atslēgu un vērtību pāri, katram `column` un `attribute_key` pārim ir jābūt unikālam.

Property	Type	Description
attribute_key	string	Atslēgas nosaukums, ko vaicā vienums, kolonnā, kurā tiek glabāti atslēgu un vērtību pāri. Vairākām kolonnām, kurās tiek glabāti atslēgu un vērtību pāri, katram `column` un `attribute_key` pārim ir jābūt unikālam. Īpašības darbība: - *obligāts*, ja `column` ir iestatīts uz kolonnu, kurā tiek glabāti atslēgu un vērtību pāri

Telemetrijas datu apkopoto kolonnu objekts

Telemetrijas datu apkopoto kolonnu objektam ir šādas īpašības.

Property	Type	Description
column	string	Telemetrijas datu kolonnas nosaukums, ko vaicā vienums. Skatiet atbalstītās telemetrijas datu kolonnas. Ja ir vairākas kolonnas, katrai `column` vērtībai ir jābūt unikālai. Šī īpašība netiek atbalstīta, ja `function` ir iestatīts uz "Count". Īpašības darbība: - *obligāta*, ja `function` ir iestatīts uz "Min", "Max", "Average", "Sum" vai "Percentile"
function	integer	Funkcija, ko vienums lieto kolonnas ierakstiem. Iespējamās vērtības: 1 - Min; 2 - Max; 3 - Average; 4 - Count; 5 - Sum; 8 - Percentile. Īpašības darbība: - *obligāta*
parameters	array	Procentile, ko vienums aprēķina no kolonnas ierakstiem. Iespējamā vērtība: skaitlis no 0 līdz 100, kas norādīts kā virkne. Īpašības darbība: - *obligāta*, ja `function` ir iestatīts uz "Percentile"
alias	string	Unikāls aprēķinātās vērtības nosaukums, ko izmanto tās identificēšanai vienuma vērtībā. Katrai `alias` vērtībai ir jābūt unikālai. Īpašības darbība: - *obligāta*

Telemetrijas datu filtrs

Telemetrijas datu filtra objektam ir šādas īpašības.

Property	Type	Description
conditions	array	Telemetrijas datu filtra nosacījumi, kuriem jāatbilst ierakstam telemetrijas datu kolonnā. Īpašības darbība: - *obligāts*
evaltype	integer	Filtra nosacījumu novērtēšanas metode. Iespējamās vērtības: 0 - And/Or; 1 - And; 2 - Or; 3 - Pielāgota izteiksme. Īpašības darbība: - *obligāts*
eval_formula	string	Ģenerēta izteiksme, kas tiks izmantota filtra nosacījumu novērtēšanai. Izteiksme satur ID, kas atsaucas uz konkrētiem filtra nosacījumiem, izmantojot to `formulaid`. `eval_formula` vērtība ir vienāda ar `formula` vērtību filtriem ar pielāgotu izteiksmi. Īpašības darbība: - *tikai lasāms*
formula	string	Lietotāja definēta izteiksme, kas jāizmanto filtru ar pielāgotu izteiksmi nosacījumu novērtēšanai. Izteiksmē jābūt ID, kas atsaucas uz konkrētiem filtra nosacījumiem, izmantojot to `formulaid`. Izteiksmē izmantotajiem ID ir precīzi jāatbilst filtra nosacījumos definētajiem ID: neviens nosacījums nedrīkst palikt neizmantots vai izlaists. Īpašības darbība: - *obligāts*, ja `evaltype` ir iestatīts uz “Pielāgota izteiksme”

Telemetrijas datu filtra nosacījumi

Telemetrijas datu filtra nosacījumu objektam ir šādas īpašības.

Property	Type	Description
column	string	Telemetrijas datu kolonnas nosaukums, uz kuru attiecas filtra nosacījums. Skatiet atbalstītās telemetrijas datu kolonnas. Property behavior: - *required*
attribute_key	string	Atslēgas nosaukums, uz kuru attiecas nosacījums, kolonnā, kurā ir atslēgu un vērtību pāri. Property behavior: - *required* if `column` is set to a column that holds key-value pairs

Property	Type	Description
operator	integer	Filtra nosacījuma operators. Iespējamās vērtības kolonnām ar vienu vērtību: 0 - Vienāds; 1 - Nav vienāds; 2 - Satur; 3 - Nesatur. Iespējamās vērtības kolonnām, kurās ir atslēgu un vērtību pāri: 0 - Vienāds; 1 - Nav vienāds; 12 - Pastāv. Property behavior: - *required*
value	string	Vērtība, ar kuru salīdzināt. Property behavior: - *required* if `operator` is set to ”Equals”, ”Does not equal”, ”Contains” or ”Does not contain”
formulaid	string	Patvaļīgs unikāls ID, kas tiek izmantots, lai atsauktos uz nosacījumu no pielāgotas izteiksmes. Tajā var būt tikai lielie burti. ID lietotājam ir jādefinē, modificējot filtra nosacījumus, taču, tos vēlāk pieprasot, tas tiks ģenerēts no jauna. Property behavior: - *required* if `evaltype` of Telemetry query filter object is set to ”Custom expression”

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 šādi rekvizīti.

Rekvizīts	Tips	Apraksts
type	integer	Priekšapstrādes opcijas tips. Iespējamās vērtības: 1 - Pielāgots reizinātājs; 2 - Labās puses atstarpju un rakstzīmju noņemšana; 3 - Kreisās puses atstarpju un rakstzīmju noņemšana; 4 - Atstarpju un rakstzīmju noņemšana; 5 - Regulārā izteiksme; 6 - Būla vērtības pārveidošana decimālskaitlī; 7 - Oktālā skaitļa pārveidošana decimālskaitlī; 8 - Heksadecimālā skaitļa pārveidošana decimālskaitlī; 9 - Vienkāršas izmaiņas; 10 - Izmaiņas sekundē; 11 - XML XPath; 12 - JSONPath; 13 - Diapazonā; 14 - Atbilst regulārajai izteiksmei; 15 - Neatbilst regulārajai izteiksmei; 16 - JSON kļūdas pārbaude; 17 - XML kļūdas pārbaude; 18 - Kļūdas pārbaude, izmantojot regulāro izteiksmi; 19 - Nemainītu vērtību atmešana; 20 - Nemainītu vērtību atmešana ar sirdspukstu; 21 - JavaScript; 22 - Prometheus raksts; 23 - Prometheus pārveidošana JSON formātā; 24 - CSV pārveidošana JSON formātā; 25 - Aizstāšana; 26 - Neatbalstīta vienuma pārbaude; 27 - XML pārveidošana JSON formātā; 28 - SNMP walk vērtība; 29 - SNMP walk pārveidošana JSON formātā; 30 - SNMP get vērtība. Rekvizīta darbība: - *obligāts*
params	string	Papildu parametri, ko izmanto priekšapstrādes opcija. Vairāki parametri tiek atdalīti ar jaunas rindas (\n) rakstzīmi. Ja `type` ir iestatīts uz ”Neatbalstīta vienuma pārbaude”, parametri atbilst sintaksei `<scope>[\n<pattern>]`, kur *pattern* ir regulārā izteiksme, bet *scope* ir viena no šīm vērtībām: -1 - atbilst jebkurai kļūdai; 0 - pārbaudīt, vai kļūdas ziņojums atbilst *pattern*; 1 - pārbaudīt, vai kļūdas ziņojums neatbilst *pattern*. Rekvizīta darbība: - *obligāts*, ja `type` ir iestatīts uz ”Pielāgots reizinātājs” (1), ”Labās puses atstarpju un rakstzīmju noņemšana” (2), ”Kreisās puses atstarpju un rakstzīmju noņemšana” (3), ”Atstarpju un rakstzīmju noņemšana” (4), ”Regulārā izteiksme” (5), ”XML XPath” (11), ”JSONPath” (12), ”Diapazonā” (13), ”Atbilst regulārajai izteiksmei” (14), ”Neatbilst regulārajai izteiksmei” (15), ”JSON kļūdas pārbaude” (16), ”XML kļūdas pārbaude” (17), ”Kļūdas pārbaude, izmantojot regulāro izteiksmi” (18), ”Nemainītu vērtību atmešana ar sirdspukstu” (20), ”JavaScript” (21), ”Prometheus raksts” (22), ”Prometheus pārveidošana JSON formātā” (23), ”CSV pārveidošana JSON formātā” (24), ”Aizstāšana” (25), Neatbalstīta vienuma pārbaude (26), ”SNMP walk vērtība” (28), ”SNMP walk pārveidošana JSON formātā” (29) vai ”SNMP get vērtība” (30)

Rekvizīts	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 ”Neatbalstīta vienuma pārbaude”: 1 - Atmest vērtību; 2 - Iestatīt pielāgotu vērtību; 3 - Iestatīt pielāgotu kļūdas ziņojumu. Rekvizīta darbība: - *obligāts*, ja `type` ir iestatīts uz ”Pielāgots reizinātājs” (1), ”Regulārā izteiksme” (5), ”Būla vērtības pārveidošana decimālskaitlī” (6), ”Oktālā skaitļa pārveidošana decimālskaitlī” (7), ”Heksadecimālā skaitļa pārveidošana decimālskaitlī” (8), ”Vienkāršas izmaiņas” (9), ”Izmaiņas sekundē” (10), ”XML XPath” (11), ”JSONPath” (12), ”Diapazonā” (13), ”Atbilst regulārajai izteiksmei” (14), ”Neatbilst regulārajai izteiksmei” (15), ”JSON kļūdas pārbaude” (16), ”XML kļūdas pārbaude” (17), ”Kļūdas pārbaude, izmantojot regulāro izteiksmi” (18), ”Prometheus raksts” (22), ”Prometheus pārveidošana JSON formātā” (23), ”CSV pārveidošana JSON formātā” (24), ”Neatbalstīta vienuma pārbaude” (26), ”XML pārveidošana JSON formātā” (27), ”SNMP walk vērtība” (28), ”SNMP walk pārveidošana JSON formātā” (29) vai ”SNMP get vērtība” (30)
error_handler_params	string	Kļūdu apstrādes parametri. Rekvizīta darbība: - *obligāts*, ja `error_handler` ir iestatīts uz ”Iestatīt pielāgotu vērtību” vai ”Iestatīt pielāgotu kļūdas ziņojumu”

Tālāk norādītie parametri un kļūdu apstrādes veidi tiek atbalstīti katram priekšapstrādes tipam.

Priekšapstrādes tips	Nosaukums	1. parametrs	2. parametrs	3. parametrs	Atbalstītie kļūdu apstrādes veidi
1	Pielāgots reizinātājs	number^1, 6^			0, 1, 2, 3
2	Labās puses atstarpju un rakstzīmju noņemšana	rakstzīmju saraksts[2]			
3	Kreisās puses atstarpju un rakstzīmju noņemšana	rakstzīmju saraksts[2]			

Priekšapstrādes tips	Nosaukums	1. parametrs	2. parametrs	3. parametrs	Atbalstītie kļūdu apstrādes veidi
4	Atstarpju un rakstzīmju noņemšana	rakstzīmju saraksts[2]			
5	Regulārā izteiksme	pattern[3]	output[2]		0, 1, 2, 3
6	Būla vērtības pārveidošana decimālskaitlī				0, 1, 2, 3
7	Oktālā skaitļa pārveidošana decimālskaitlī				0, 1, 2, 3
8	Heksadecimālā skaitļa pārveidošana decimālskaitlī				0, 1, 2, 3
9	Vienkāršas izmaiņas				0, 1, 2, 3
10	Izmaiņas sekundē				0, 1, 2, 3
11	XML XPath	path[4]			0, 1, 2, 3
12	JSONPath	path[4]			0, 1, 2, 3
13	Diapazona	min^1, 6^	max^1, 6^		0, 1, 2, 3
14	Atbilst regulārajai izteiksmei	pattern[3]			0, 1, 2, 3
15	Neatbilst regulārajai izteiksmei	pattern[3]			0, 1, 2, 3
16	JSON kļūdas pārbaude	path[4]			0, 1, 2, 3
17	XML kļūdas pārbaude	path[4]			0, 1, 2, 3

Priekšapstrādes tips	Nosaukums	1. parametrs	2. parametrs	3. parametrs	Atbalstītie kļūdu apstrādes veidi
18	Kļūdas pārbaude, izmantojot regulāro izteiksmi	pattern[3]	output[2]		0, 1, 2, 3
19	Nemainītu vērtību atmešana				
20	Nemainītu vērtību atmešana ar sirdspukstu	seconds^5, 6^			
21	JavaScript	script[2]			
22	Prometheus raksts	pattern^6, 7^	`value`, `label`, `function`	output^8, 9^	0, 1, 2, 3
23	Prometheus pārveidošana JSON formātā	pattern^6, 7^			0, 1, 2, 3
24	CSV pārveidošana JSON formātā	character[2]	character[2]	0,1	0, 1, 2, 3
25	Aizstāšana	search string[2]	replacement[2]		
26	Neatbalstītu vienuma pārbaude	steps[1]	pattern^3, 6^		1, 2, 3
27	XML pārveidošana JSON formātā				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 - Integer no BITS		0, 1, 2, 3

Priekšapstrādes tips	Nosaukums	1. parametrs	2. parametrs	3. parametrs	Atbalstītie kļūdu apstrādes veidi
29	SNMP walk pārveidošana JSON formātā[10]	Field name[2]	OID prefix[2]	Formāts: 0 - Nemainīts 1 - UTF-8 no Hex-STRING 2 - MAC no Hex-STRING 3 - Integer no BITS	0, 1, 2, 3
30	SNMP get vērtība	Formāts: 1 - UTF-8 no Hex-STRING 2 - MAC no Hex-STRING 3 - Integer no BITS			0, 1, 2, 3

[1] peldošā komata skaitlis (veseli skaitļi tiek netieši pārveidoti par peldošā komata 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 apkopošanas funkcijām: `sum`, `min`, `max`, `avg`, `count`, ja kā otrais parametrs ir atlasīts `function`.
 [10] Atbalsta vairākus "Field name,OID prefix,Format records" ierakstus, kas atdalīti ar jaunas rindas rakstzīmi.

item.get

Apraksts

`integer/array item.get(object parameters)`

Šī metode ļauj izgūt vienumus atbilstoši norādītajiem parametriem.

> **Note:**
> Šī metode ir pieejama jebkura tipa lietotājiem. Tiesības 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 vienumus ar norādītajiem ID.
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.

Parametrs	Tips	Apraksts
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`.
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.

Parametrs	Tips	Apraksts
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. Tiesības 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. Tiesības izsaukt šo metodi var atsaukt lietotāju lomu iestatījumos. Plašāku informāciju skatiet sadaļā 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. Tiesības izsaukt šo metodi var atsaukt lietotāju lomu iestatījumos. Plašāku informāciju skatiet sadaļā 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
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 jaunus vērtību kartējumus.

> **Note:**
> Šī metode ir pieejama tikai *Super admin* lietotāja tipam. Tiesības izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā User roles.

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. Tiesības izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet User roles.

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. Tiesības izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā Lietotāja 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.
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. Tiesības izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā User roles.

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. Tiesības izsaukt šo metodi var atsaukt 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)`

Šī metode ļauj izgūt vēstures datus atbilstoši norādītajiem parametriem.

Attention:
Šī metode var atgriezt dzēstas entītijas vēsturiskos datus, ja šie dati vēl nav noņemti ar housekeeper.

Note:
Šī metode ir pieejama jebkura tipa lietotājiem. Tiesības izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā Lietotāja lomas.

Parametri

`(object)` Parametri, kas definē vēlamo izvadi.

Metode atbalsta šādus parametrus.

Parametrs	Tips	Apraksts
history	integer	Atgriež vēstures objektu tipus. Iespējamās vērtības: 0 - skaitlis ar peldošo komatu; 1 - rakstzīme; 2 - žurnāls; 3 - *(noklusējums)* neparakstīts skaitlis ar peldošo komatu; 4 - teksts; 5 - binārs; 6 - JSON.
hostids	ID/array	Atgriež vēsturi tikai no norādītajiem hosts.

Parametrs	Tips	Apraksts
itemids	ID/array	Atgriež vēsturi tikai no norādītajiem vienumiem.
maxValueSize	integer	Maksimālais baitu skaits, ko atgriezt `value` īpašībā. Iespējamais vērtību diapazons: 1-134217728 (1 baits līdz 128 MiB). Ja iestatīts uz `null`, ierobežojums netiek piemērots. Noklusējums: 65536 (64 KiB). Parametra darbība: - *atbalstīts*, ja `history` ir iestatīts uz "binary" vai "JSON"
time_from	timestamp	Atgriež tikai tās vērtības, kas saņemtas pēc norādītā laika vai tā brīdī.
time_till	timestamp	Atgriež tikai tās vērtības, kas saņemtas pirms norādītā laika vai tā brīdī.
sortfield	string/array	Kārto rezultātu pēc norādītajām īpašībām. Iespējamās vērtības: `itemid`, `clock`, `ns`. Neatbalsta kārtošanu pēc `ns`, ja vēstures dati tiek glabāti ClickHouse.
search	object	Atgriež rezultātus, kas atbilst norādītajam paraugam (nav reģistrjutīgs). Pieņem objektu, kur atslēgas ir īpašību nosaukumi, bet vērtības ir meklējamās virknes. Ja papildu opcijas nav norādītas, 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
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 Metodes paraksts

Katrā API metodes lapā ir norādīts metodes paraksts.

Piemērs: `object action.create(object/array actions)`

Paraksts	Apraksts
`object`	To datu tips, ko metode atgriež.
`action.create`	Pilns metodes nosaukums.
`object/array`	To datu tips, ko metode pieņem.
`actions`	Īss kopsavilkums par to, ko metode pieņem.

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).
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.

Parametrs	Tips	Apraksts
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 64 KiB.

hosts

ZBXNEXT-9185 Pievienots rekvizīta `monitored_by` darbības apraksts objekta host rekvizītam. ZBXNEXT-10387 `host.massupdate`: metode noņemta. Tā vietā izmantojiet `host.update`.

hostgroup

ZBXNEXT-10387 `hostgroup.massupdate`: Noņemta metode. Lūdzu, tā vietā izmantojiet `hostgroup.update`, `hostgroup.massadd` vai `hostgroup.massremove`.

hostinterface

ZBXNEXT-10814 `hostinterface.create`, `hostinterface.update`, `hostinterface.delete`, `hostinterface.massadd` un `hostinterface.massremove` metodes ir noņemtas no `hostinterface` API. Lūdzu, tā vietā izmantojiet host objekta metodes.

loma

ZBXNEXT-9185 Lomas action objekta rekvizītam `name` pievienota jauna vērtība `select_server_for_monitoring`.

veidne

ZBXNEXT-10387 `template.massupdate`: Noņemta metode. Lūdzu, izmantojiet `template.update`, `template.massadd` vai `template_massremove` vietā.

veidņu grupa

ZBXNEXT-10387 `templategroup.massupdate`: metode noņemta. Lūdzu, tā vietā izmantojiet `templategroup.update`, `templategroup.massadd` vai `templategroup.massremove`.

lietotāju grupa

ZBXNEXT-9185 Metodēm `usergroup.create` un `usergroup.update` pievienoti jauni parametri `proxies` un `proxy_groups`.
 Metodei `usergroup.get` pievienoti jauni parametri `proxyids`, `proxy_groupids`, `selectProxies` un `selectProxyGroups`.
 Lietotāju grupas objektam pievienotas jaunas īpašības `proxy_mode` un `proxy_group_mode`.

Citas izmaiņas un kļūdu labojumi darbība

ZBXNEXT-9399 Parametram `conditiontype` darbība filtra parametrā pievienota jauna vērtība 29 (starpniekserveru grupa), ja notikumu avots ir atklāšana/automātiskā reģistrācija.

apmlog

ZBXNEXT-10083 Pievienots jauns APM žurnāla API.

apmmetric

ZBXNEXT-10083 Pievienots jauns APM metrikas API.

apmspan

ZBXNEXT-10083 Pievienots jauns APM span API.

apmtrace

ZBXNEXT-10083 Pievienots jauns APM trace API.

konfigurācija

ZBXNEXT-5419 Metodei configuration.importcompare pievienots jauns parametrs `returnMissingObjects`.
 ZBXNEXT-5419 Objektam rules (`configuration.importcompare` metode) pievienots jauns parametrs `dashboards`.
 ZBXNEXT-5419 Metodei configuration.import pievienots jauns parametrs `returnMissingObjects`.
 ZBXNEXT-5419 Objektam rules (`configuration.import` metode) pievienots jauns parametrs `dashboards`. ZBXNEXT-10390 Metodes `configuration.export` objektam `options` pievienots jauns parametrs `global_regexes`. ZBXNEXT-10390 Metožu `configuration.importcompare` un `configuration.import` objektam `rules` pievienots jauns parametrs `global_regexes`.

savienotājs

ZBXNEXT-10069 Objektam connector pievienota jauna item_value_type vērtība (64 - JSON).

informācijas panelis

ZBXNEXT-4769 graph logrīkam pievienots jauns informācijas paneļa logrīka lauks show_hostnames.
 ZBXNEXT-9546 svg-graph un piechart logrīkiem pievienoti jauni informācijas 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.
 ZBXNEXT-10158 topitems logrīkam pievienoti jauni informācijas paneļa logrīka lauki columns.0.aggregate_columns, columns.0.combined_column_name un columns.0.column_aggregat ZBXNEXT-9424, ZBXNEXT-4769 Pievienots jauns informācijas paneļa logrīka tips scatterplot.
 ZBXNEXT-8160 geomap logrīkam pievienoti jauni informācijas paneļa logrīka lauki clustering_mode un clustering_zoom_level.
 ZBXNEXT-943 graph logrīkam pievienots jauns informācijas paneļa logrīka lauks ds.0.invert_values.
 ZBXNEXT-10657 honeycomb logrīka parametriem Color un Background color pievienotas noklusējuma vērtības jaunajai "dark-blue-theme" tēmai.

drule

ZBXNEXT-9185 Atjaunināta atklāšanas noteikuma objekta timeout rekvizīta darbība.

ierīce

ZBXNEXT-10357 Pievienots jauns Device API ar metodēm device.get un device.offboard.

notikums

ZBXNEXT-1150 Metodei event.get parametram selectAcknowledges tika pievienots īpašums maintenanceid.

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-10249 Objektam vienums pievienota jauna type vērtība (24 - Telemetry query).
 ZBXNEXT-10249 Objektam vienums pievienotas jaunas īpašības query, time_shift, lookback_limit un granularity.
 ZBXNEXT-9185 Atjaunināta objekta vienums īpašības timeout darbība.
 ZBXNEXT-10069 Objektam vienums pievienota jauna value_type vērtība (6 - JSON).
 ZBXNEXT-9546 Metodei item.get pievienoti jauni parametri inheritedTags un selectInheritedTags.

vienuma prototips

ZBXNEXT-10249 Objektam vienuma prototips pievienota jauna type vērtība (24 - Telemetry query).
 ZBXNEXT-10249 Objektam vienuma prototips pievienotas jaunas īpašības query, time_shift, lookback_limit un granularity.
 ZBXNEXT-10069 Objektam vienumaprototips pievienota jauna value_type vērtība (6 - JSON).
 ZBXNEXT-9546 Metodei itemprototype.get pievienots jauns parametrs selectInheritedTags.

uzturēšana

ZBXNEXT-8339 maintenance.create un maintenance.update metodēm pievienoti triggers un event_names rekvizīti. Rekvizītam problem_tags pievienoti operatori "Nav vienāds" un "Nesatur".
 ZBXNEXT-8339 maintenance.get metodei pievienoti parametri triggerids, selectTriggers un selectEventNames.

mediatype

ZBXNEXT-10357 media type objektam pievienota jauna type vērtība (5 - Push).

problēma

ZBXNEXT-1150 Metodei problem.get parametram selectAcknowledges pievienots īpašums maintenanceid.

starpniekserveris

ZBXNEXT-10084 Metodei proxy.get pievienots jauns parametrs selectApm.
 ZBXNEXT-10084 Metodēm proxy.create un proxy.update pievienots jauns parametrs apm.
 ZBXNEXT-10084 Pievienots jauns objekts lietojumprogrammu veiktspējas uzraudzība (APM).
 ZBXNEXT-10249 Objektam proxy pievienots jauns rekvizīts timeout_telemetry_query.

regexp

ZBXNEXT-10390 Pievienots jauns description īpašums regular expression objektam.

loma

ZBXNEXT-10469 Mainīta rekvizīta `api.access` noklusējuma vērtība objektā lomu noteikumi. Tagad noklusējuma vērtība ir 0 (atspējots).

serverinfo

ZBXNEXT-10357 Pievienots jauns Server info API ar metodi `serverinfo.get`.

iestatījumi

ZBXNEXT-10249 Objektam settings pievienots jauns rekvizīts `timeout_telemetry_query`.
 ZBXNEXT-10680 Objektam settings pievienots jauns rekvizīts `apm_global_db`.
 ZBXNEXT-10357 Objektam settings pievienots jauns rekvizīts `device_link_timeout`.
 ZBXNEXT-10657 Objekta settings rekvizītam `default_theme` pievienota vērtība `dark-blue-theme`.

veidne

ZBXNEXT-9546 Pievienoti jauni parametri `inheritedTags` un `selectInheritedTags` metodei template.get.

veidnes panelis

ZBXNEXT-9424 Pievienots jauns veidnes paneļa logrīka tips izkliedes diagramma.

trigeris

ZBXNEXT-9546 Metodei trigger.get pievienoti jauni parametri `inheritedTags` un `selectInheritedTags`.
 ZBXNEXT-8339 Metodei trigger.get pievienota jauna īpašība `maintenanceids`.

trigera prototips

ZBXNEXT-9546 Metodei triggerprototype.get pievienots jauns parametrs `selectInheritedTags`.

lietotājs

ZBXNEXT-10657 Objekta user rekvizītam `theme` pievienota vērtība `dark-blue-theme`.
 ZBXNEXT-8339 Metodēm user.create, user.get un user.update pievienots jauns rekvizīts `default_maintenance_period`.

userdirectory

ZBXNEXT-9834 lietotāja direktorija objektam pievienotas īpašības `idp_certificate`, `sp_private_key`, `sp_certificate`, `idp_certificate_hash`, `sp_private_key_hash` un `sp_certificate_hash`.
 DEV-5022 lietotāja direktorija objektā mainīta īpašību `sign_assertions`, `sign_authn_requests`, `sign_messages`, `sign_logout_requests`, `sign_logout_responses` noklusējuma vērtība. Tagad noklusējuma vērtība ir 1 (iespējots).

3. pielikums. Izmaiņas 8.0 versijā

8.0.1 Šī versija vēl nav izlaista.

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 pieprasa tos no avota. 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 Zabbix sender utilītu vai Zabbix sender protokolu. Datu nosūtīšana uz Zabbix serveri ir iespējama arī, izmantojot `history.push` API metodi.

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 hosts, kurā darbojas Zabbix aģents.

Note:
`system.run[]` vienumi pēc noklusējuma ir atspējoti, un, ja tos izmanto, tie ir jāiespējo (jāatļauj) un jādefinē Zabbix aģenta vai Zabbix aģenta 2 konfigurācijas failā (`AllowKey` vai `AllowKeyRegexp` konfigurācijas parametrā).

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 nosūtīšanai uz HTTP galapunktiem, lai izgūtu datus no tādiem pakalpojumiem kā *Elasticsearch* un *OpenWeatherMap*, pārbaudītu Zabbix API statusu vai Apache vai NGINX tīmekļa servera statusu u. c. HTTP aģenta vienumi (ar iespējotu datu saņemšanu) 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.

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 ir jāsakrīt ar Zabbix atbalstīto moduļa API versiju. Zabbix atbalstītā moduļa API versija ir `ZBX_MODULE_API_VERSION`. Tāpēc šai funkcijai ir 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 avota savietojamību, taču tās lietošana nav ieteicama.

Neobligātā saskarne

Neobligātās funkcijas ir `zbx_module_init()`, `zbx_module_item_list()`, `zbx_module_item_timeout()`, `zbx_module_history_wri` un `zbx_module_uninit()`:

```
int zbx_module_init(void);
```

Šai funkcijai jāveic nepieciešamā moduļa inicializācija, ja tāda ir nepieciešama. Ja izpilde ir veiksmīga, tai jāatgriež `ZBX_MODULE_OK`. Pretējā gadījumā tai jāatgriež `ZBX_MODULE_FAIL`. Pēdējā gadījumā Zabbix netiks palaists.

```
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 tiek noslēgts ar `ZBX_METRIC` struktūru, kuras lauks "key" ir NULL.

```
void    zbx_module_item_timeout(int timeout);
```

Ja modulis eksportē `zbx_module_item_list()`, tad Zabbix izmanto šo funkciju, lai norādītu taimauta iestatījumus Zabbix konfigurācijas failā, kuri jāievēro moduļa ieviestajām vienumu pārbaudēm. Šeit parametrs "timeout" ir sekundēs.

```
ZBX_HISTORY_WRITE_CBS   zbx_module_history_write_cbs(void);
```

Šai funkcijai jāatgriež atpakaļizsaukuma funkcijas, ko Zabbix serveris izmantos dažādu datu tipu vēstures eksportēšanai. Atpakaļizsaukuma funkcijas tiek nodrošinātas kā `ZBX_HISTORY_WRITE_CBS` struktūras lauki; lauki var būt NULL, ja modulim nav nepieciešama noteikta tipa vēsture.

```
int zbx_module_uninit(void);
```

Šai funkcijai jāveic nepieciešamā deinicializācija, ja tāda ir nepieciešama, piemēram, piešķirto resursu atbrīvošana, failu deskriptoru aizvēršana 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 ir 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), bet **test_param** ir parametru saraksts, kas jāizmanto, kad Zabbix aģents tiek palaists ar -p karodziņu (piemēram, ”1,1000”, var būt NULL).

Piemēra definīcija 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 ir tipa AGENT_REQUEST, bet otrais ir tipa 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 ir veiksmīgi iegūta. Pretējā gadījumā tām jāatgriež SYSINFO_RET_FAIL. Skatiet tālāk esošo piemēru ”dummy” modulim, lai uzzinātu, kā iegūt informāciju no AGENT_REQUEST un kā iestatīt informāciju AGENT_RESULT.

Vēstures eksportēšanas atzvanu nodrošināšana

Attention:
Vēstures eksportēšana, izmantojot moduli, vairs netiek atbalstīta ar Zabbix starpniekserveri.

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 jāsaņem history masīvs ar history_num elementiem. Atkarībā no eksportējamā vēstures datu tipa history attiecīgi ir viens no šādiem struktūru masīviem:

```
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;
```

Atzvani tiks izmantoti Zabbix servera vēstures sinhronizācijas procesu beigās pēc tam, kad dati ir ierakstīti Zabbix datubāzē un saglabāti vērtību kešatmiņā.

Attention:
Iekšējas kļūdas gadījumā vēstures eksportēšanas modulī ieteicams moduli izstrādāt tā, lai tas nebloķētu visu uzraudzību līdz atkopšanai, bet gan atmestu datus un ļautu Zabbix serverim turpināt darbu.

Moduļu būvēšana

Pašlaik moduļi ir paredzēti būvēšanai Zabbix avota 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 būvēšanai. Modulim jāiekļauj `stdlib.h`, `stdint.h` un `module.h`, un būvēšanas skriptam jānodrošina, ka šie faili ir iekļaušanas ceļā. Sīkāku informāciju skatiet tālāk esošajā moduļa "dummy" piemērā.

Vēl viena noderīga galvene ir `include/zbxcommon.h`, kas definē `zabbix_log()` funkciju, ko var izmantot reģistrēšanai un atkļūdošanai.

Konfigurācijas parametri

Zabbix aģents, serveris un starpniekserveris atbalsta divus parametrus, lai darbotos ar moduļiem:

- `LoadModulePath` - pilns ceļš uz ielādējamo moduļu atrašanās vietu
- `LoadModule` - modulis(-ļi), kas jāielādē startēšanas laikā.

Moduļiem jāatrodas direktorijā, kas norādīta ar LoadModulePath, vai arī ceļam jābūt norādītam pirms moduļa nosaukuma. Ja norādītais ceļš ir absolūts (sākas ar /), LoadModulePath tiek ignorēts. Var izmantot vairākus LoadModule parametrus.

Piemēram, lai paplašinātu Zabbix aģentu, varam 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
```

Startējot aģentu, tas ielādēs mariadb.so, apache.so un kernel.so moduļus no direktorijas /usr/local/lib/zabbix/agent, savukārt dummy.so tiks ielādēts no /usr/local/lib/zabbix. Aģentu neizdosies startēt, ja trūkst kāda moduļa, ir nepietiekamas piekļuves tiesības vai koplietojamā bibliotēka nav Zabbix modulis.

Lietotāja saskarnes konfigurācija

Ielādējamie moduļi tiek atbalstīti Zabbix aģents, serveris un starpniekserveris. Tāpēc vienuma tips Zabbix lietotāja saskarnē ir atkarīgs no tā, kur modulis tiek ielādēts. Ja modulis tiek ielādēts aģentā, tad vienuma tipam jābūt "Zabbix agent" vai "Zabbix agent (active)". Ja modulis tiek ielādēts serverī vai starpniekserverī, tad vienuma tipam jābūt "Simple check".

Vēstures eksportam, izmantojot Zabbix moduļus, nav nepieciešama nekāda lietotāja saskarnes konfigurācija. Ja moduli veiksmīgi ielādē serveris un tas nodrošina zbx_module_history_write_cbs() funkciju, kas atgriež vismaz vienu ne-NULL atpakaļizsaukuma funkciju, tad vēstures eksports tiks iespējots automātiski.

Dummy modulis

Zabbix ietver parauga moduli, kas rakstīts C valodā. Modulis atrodas 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, un to var izmantot kā veidni saviem moduļiem.

Pēc tam, kad saknes Zabbix avota kokā ir palaists ./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;
```

```c
/* module SHOULD define internal functions as static and use a naming pattern different from Zabbix intern
/* symbols (zbx_*) and loadable module API functions (zbx_module_*) to avoid conflicts
static int  dummy_ping(AGENT_REQUEST *request, AGENT_RESULT *result);
static int  dummy_echo(AGENT_REQUEST *request, AGENT_RESULT *result);
static int  dummy_random(AGENT_REQUEST *request, AGENT_RESULT *result);

static ZBX_METRIC keys[] =
/*  KEY         FLAG        FUNCTION    TEST PARAMETERS */
{
    {"dummy.ping",      0,      dummy_ping, NULL},
    {"dummy.echo",      CF_HAVEPARAMS,  dummy_echo, "a message"},
    {"dummy.random",    CF_HAVEPARAMS,  dummy_random,   "1,1000"},
    {NULL}
};

/******************************************************************************
 *                                                                            *
 * Function: zbx_module_api_version                                           *
 *                                                                            *
 * Purpose: returns version number of the module interface                    *
 *                                                                            *
 * Return value: ZBX_MODULE_API_VERSION - version of module.h module is       *
 *               compiled with, in order to load module successfully Zabbix   *
 *               MUST be compiled with the same version of this header file   *
 *                                                                            *
 ******************************************************************************/
int zbx_module_api_version(void)
{
    return ZBX_MODULE_API_VERSION;
}

/******************************************************************************
 *                                                                            *
 * Function: zbx_module_item_timeout                                          *
 *                                                                            *
 * Purpose: set timeout value for processing of items                         *
 *                                                                            *
 * Parameters: timeout - timeout in seconds, 0 - no timeout set               *
 *                                                                            *
 ******************************************************************************/
void    zbx_module_item_timeout(int timeout)
{
    item_timeout = timeout;
}

/******************************************************************************
 *                                                                            *
 * Function: zbx_module_item_list                                             *
 *                                                                            *
 * Purpose: returns list of item keys supported by the module                 *
 *                                                                            *
 * Return value: list of item keys                                            *
 *                                                                            *
 ******************************************************************************/
ZBX_METRIC  *zbx_module_item_list(void)
{
    return keys;
}

static int  dummy_ping(AGENT_REQUEST *request, AGENT_RESULT *result)
{
    SET_UI64_RESULT(result, 1);
```

```
    return SYSINFO_RET_OK;
}

static int  dummy_echo(AGENT_REQUEST *request, AGENT_RESULT *result)
{
    char    *param;

    if (1 != request->nparam)
    {
        /* set optional error message */
        SET_MSG_RESULT(result, strdup("Invalid number of parameters."));
        return SYSINFO_RET_FAIL;
    }

    param = get_rparam(request, 0);

    SET_STR_RESULT(result, strdup(param));

    return SYSINFO_RET_OK;
}

/******************************************************************************
 *                                                                            *
 * Function: dummy_random                                                     *
 *                                                                            *
 * Purpose: a main entry point for processing of an item                      *
 *                                                                            *
 * Parameters: request - structure that contains item key and parameters      *
 *              request->key - item key without parameters                    *
 *              request->nparam - number of parameters                        *
 *              request->params[N-1] - pointers to item key parameters        *
 *              request->types[N-1] - item key parameters types:              *
 *                  REQUEST_PARAMETER_TYPE_UNDEFINED (key parameter is empty) *
 *                  REQUEST_PARAMETER_TYPE_ARRAY (array)                      *
 *                  REQUEST_PARAMETER_TYPE_STRING (quoted or unquoted string) *
 *                                                                            *
 *             result - structure that will contain result                    *
 *                                                                            *
 * Return value: SYSINFO_RET_FAIL - function failed, item will be marked      *
 *                                  as not supported by zabbix                *
 *               SYSINFO_RET_OK - success                                     *
 *                                                                            *
 * Comment: get_rparam(request, N-1) can be used to get a pointer to the Nth  *
 *          parameter starting from 0 (first parameter). Make sure it exists  *
 *          by checking value of request->nparam.                             *
 *          In the same manner get_rparam_type(request, N-1) can be used to   *
 *          get a parameter type.                                             *
 *                                                                            *
 ******************************************************************************/
static int  dummy_random(AGENT_REQUEST *request, AGENT_RESULT *result)
{
    char    *param1, *param2;
    int from, to;

    if (2 != request->nparam)
    {
        /* set optional error message */
        SET_MSG_RESULT(result, strdup("Invalid number of parameters."));
        return SYSINFO_RET_FAIL;
    }

    param1 = get_rparam(request, 0);
```

```
    param2 = get_rparam(request, 1);

    /* there is no strict validation of parameters and types for simplicity sake */
    from = atoi(param1);
    to = atoi(param2);

    if (from > to)
    {
        SET_MSG_RESULT(result, strdup("Invalid range specified."));
        return SYSINFO_RET_FAIL;
    }

    SET_UI64_RESULT(result, from + rand() % (to - from + 1));

    return SYSINFO_RET_OK;
}

/******************************************************************************
 *                                                                            *
 * Function: zbx_module_init                                                  *
 *                                                                            *
 * Purpose: the function is called on agent startup                           *
 *          It should be used to call any initialization routines             *
 *                                                                            *
 * Return value: ZBX_MODULE_OK - success                                      *
 *               ZBX_MODULE_FAIL - module initialization failed               *
 *                                                                            *
 * Comment: the module won't be loaded in case of ZBX_MODULE_FAIL             *
 *                                                                            *
 ******************************************************************************/
int zbx_module_init(void)
{
    /* initialization for dummy.random */
    srand(time(NULL));

    return ZBX_MODULE_OK;
}

/******************************************************************************
 *                                                                            *
 * Function: zbx_module_uninit                                                *
 *                                                                            *
 * Purpose: the function is called on agent shutdown                          *
 *          It should be used to cleanup used resources if there are any      *
 *                                                                            *
 * Return value: ZBX_MODULE_OK - success                                      *
 *               ZBX_MODULE_FAIL - function failed                            *
 *                                                                            *
 ******************************************************************************/
int zbx_module_uninit(void)
{
    return ZBX_MODULE_OK;
}

/******************************************************************************
 *                                                                            *
 * Functions: dummy_history_float_cb                                          *
 *            dummy_history_integer_cb                                        *
 *            dummy_history_string_cb                                         *
 *            dummy_history_text_cb                                           *
 *            dummy_history_log_cb                                            *
 *                                                                            *
```

```
 * Purpose: callback functions for storing historical data of types float,    *
 *          integer, string, text and log respectively in external storage    *
 *                                                                            *
 * Parameters: history     - array of historical data                         *
 *             history_num - number of elements in history array              *
 *                                                                            *
 ******************************************************************************/
static void dummy_history_float_cb(const ZBX_HISTORY_FLOAT *history, int history_num)
{
    int i;

    for (i = 0; i < history_num; i++)
    {
        /* do something with history[i].itemid, history[i].clock, history[i].ns, history[i].value, ... */
    }
}

static void dummy_history_integer_cb(const ZBX_HISTORY_INTEGER *history, int history_num)
{
    int i;

    for (i = 0; i < history_num; i++)
    {
        /* do something with history[i].itemid, history[i].clock, history[i].ns, history[i].value, ... */
    }
}

static void dummy_history_string_cb(const ZBX_HISTORY_STRING *history, int history_num)
{
    int i;

    for (i = 0; i < history_num; i++)
    {
        /* do something with history[i].itemid, history[i].clock, history[i].ns, history[i].value, ... */
    }
}

static void dummy_history_text_cb(const ZBX_HISTORY_TEXT *history, int history_num)
{
    int i;

    for (i = 0; i < history_num; i++)
    {
        /* do something with history[i].itemid, history[i].clock, history[i].ns, history[i].value, ... */
    }
}

static void dummy_history_log_cb(const ZBX_HISTORY_LOG *history, int history_num)
{
    int i;

    for (i = 0; i < history_num; i++)
    {
        /* do something with history[i].itemid, history[i].clock, history[i].ns, history[i].value, ... */
    }
}

/******************************************************************************
 *                                                                            *
 * Function: zbx_module_history_write_cbs                                     *
 *                                                                            *
 * Purpose: returns a set of module functions Zabbix will call to export      *
```

```
 *          different types of historical data                                *
 *                                                                            *
 * Return value: structure with callback function pointers (can be NULL if    *
 *               module is not interested in data of certain types)           *
 *                                                                            *
 ******************************************************************************/
ZBX_HISTORY_WRITE_CBS   zbx_module_history_write_cbs(void)
{
    static ZBX_HISTORY_WRITE_CBS    dummy_callbacks =
    {
        dummy_history_float_cb,
        dummy_history_integer_cb,
        dummy_history_string_cb,
        dummy_history_text_cb,
        dummy_history_log_cb,
    };

    return dummy_callbacks;
}
```

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 diapazonā no param1 līdz param2, piemēram, `dummy.random[1,100000`

Ierobežojumi

Ielādējamo moduļu atbalsts ir ieviests tikai Unix platformai. Tas nozīmē, ka tas nedarbojas Windows aģentos.

Dažos gadījumos modulim var būt nepieciešams nolasīt ar moduli saistītos konfigurācijas parametrus no `zabbix_agentd.conf`. Pašlaik tas netiek atbalstīts. Ja jūsu modulim ir jāizmanto kādi konfigurācijas parametri, jums, iespējams, vajadzētu ieviest modulim specifiska konfigurācijas faila parsēšanu.

Lietotāja saskarnes moduļi

Pārskats

Ir iespējams paplašināt Zabbix lietotāja saskarnes funkcionalitāti, pievienojot trešo pušu moduļus vai izstrādājot savus moduļus bez nepieciešamības mainīt 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, ko instalējat.
- Kļūdas trešās puses moduļa kodā var izraisīt lietotāja saskarnes avāriju. Ja tas notiek, vienkārši noņemiet moduļa kodu no lietotāja saskarnes. Tiklīdz atkārtoti ielādēsiet Zabbix lietotāja saskarni, redzēsiet piezīmi, ka daži moduļi nav pieejami. Dodieties uz Module administration (sadaļā *Administration* > *General* > *Modules*) un vēlreiz noklikšķiniet *Scan directory*, lai no datubāzes noņemtu neesošos moduļus.

Instalēšana

Lūdzu, vienmēr izlasiet konkrētā moduļa instalēšanas rokasgrāmatu. Ieteicams jaunus moduļus instalēt pa vienam, lai kļūmes varētu viegli pamanīt.

Tieši pirms moduļa instalēšanas:

- Pārliecinieties, ka esat lejupielādējis moduli no uzticama avota. Kaitīga koda instalēšana var radīt sekas, piemēram, datu zudumu.
- Viena un tā paša moduļa dažādas versijas (ar to pašu ID) var instalēt paralēli, taču vienlaikus var būt iespējota tikai viena versija.

Moduļa instalēšanas soļi:

- Izpakojiet moduli tā paša nosaukuma mapē Zabbix lietotāja saskarnes instalācijas `modules` direktorijā (piemēram, `zabbix/ui/modules`).
- Pārliecinieties, ka moduļa mapē ir vismaz `manifest.json` fails.
- Dodieties uz Moduļu administrēšana un noklikšķiniet uz pogas *Skenēt direktoriju*.

- Jaunais modulis parādīsies sarakstā kopā ar tā versiju, autoru, aprakstu un statusu.
- Iespējojiet moduli, noklikšķinot uz tā statusa.

Problēmu novēršana:

Problēma	Risinājums
Modulis neparādījās sarakstā	Ja jūsu modulis neparādījās sarakstā, pārliecinieties, ka `manifest.json` pastāv `modules/your-module/`. Ja tā nav, iespējams, moduli esat izpakojis nepareizā direktorijā. Ja tā ir, modulis, iespējams, nav saderīgs ar jūsu Zabbix versiju. Tāpat pārbaudiet, vai tīmekļa servera lietotājam ir vismaz lasīšanas un meklēšanas piekļuve (`r-x`) moduļa mapei un visām apakšmapēm, kā arī lasīšanas piekļuve (`r--`) visiem tajās esošajiem failiem.
Lietotāja saskarne avarēja	Moduļa kods nav saderīgs ar pašreizējo Zabbix versiju vai servera konfigurāciju. Lūdzu, dzēsiet moduļa failus un pārlādējiet lietotāja saskarni. Jūs redzēsiet paziņojumu, ka daži moduļi nav pieejami. Dodieties uz Moduļu administrēšana un vēlreiz noklikšķiniet uz *Skenēt direktoriju*, lai no datubāzes noņemtu neesošos moduļus.
Parādās kļūdas ziņojums par identisku namespace, ID vai actions	Jaunais modulis mēģināja reģistrēt namespace, ID vai actions, kas jau ir reģistrēti citos iespējotajos moduļos. Pirms jaunā moduļa iespējošanas atspējojiet konfliktējošo moduli (minēto kļūdas ziņojumā).
Parādās tehniski kļūdu ziņojumi	Ziņojiet par kļūdām moduļa izstrādātājam.

Moduļu izstrāde

Informāciju par pielāgotu moduļu izstrādi skatiet Izstrādātāju centrā.

Pielikumi

Pārskats Šajā sadaļā ir iekļauti atsauces materiāli, konfigurācijas parametri, tehniskās specifikācijas un procedūru ceļveži, kas papildina galveno rokasgrāmatu.

Būtiska atsauce

- Instalēšana un iestatīšana - datubāzes izveide, servera un aģenta instalēšana un konfigurēšana.
- Procesu konfigurācija - Zabbix servera, starpniekservera, aģenta un citu komponentu konfigurācijas parametri.
- Vienumi - vienumu atslēgu atsauce un atbalstītie datu tipi.
- Makrosi - iebūvēto un lietotāja makrosu sintakse un lietošana.

Instalēšana un iestatīšana

Šajā sadaļā ir aprakstīta datubāzes izveide, drošības konfigurācija, platformai specifiskas iestatīšanas procedūras un funkciju konfigurācija.

Lūdzu, izmantojiet sānjoslu, lai atrastu savai videi atbilstošo procedūru.

Datu bāzes izveide

Pārskats

Zabbix datubāze ir jāizveido Zabbix serveris vai starpniekserveris instalēšanas laikā.

Šajā sadaļā ir sniegtas instrukcijas Zabbix datubāzes izveidei. Katram atbalstītajam datubāzes veidam ir pieejams atsevišķs instrukciju kopums.

Note:
Lai uzlabotu datubāzes drošību, izveidojot datubāzes lomas/lietotājus ar minimālām tiesībām, skatiet datubāzes izveides labāko praksi katram atbalstītajam datubāzes veidam:

- MySQL/MariaDB
- PostgreSQL/TimescaleDB

Drošu TLS savienojumu konfigurēšanai skatiet Drošs savienojums ar datubāzi.

UTF-8 ir vienīgais Zabbix atbalstītais kodējums. Ir zināms, ka tas darbojas bez drošības trūkumiem. Lietotājiem jāapzinās, ka, izmantojot dažus citus kodējumus, pastāv zināmas drošības problēmas. Lai pārslēgtos uz UTF-8, skatiet Zabbix datubāzes rakstzīmju kopas un salīdzināšanas labošanas. Skatiet arī Filtrēšanas ierobežojumi ar utf8mb4 salīdzināšanām.

Note:
Ja instalējat no Zabbix Git repository, pirms pārejas uz nākamajām darbībām ir jāizpilda šāda komanda:

 `make dbschema`

MySQL/MariaDB

Zabbix servera/starpniekservera pareizai darbībai ar MySQL/MariaDB datubāzēm tiek atbalstīta rakstzīmju kopa utf8mb4 ar utf8mb4_bin kolāciju.
 Lai izvairītos no problēmām nākotnē, gan jaunām, gan esošām instalācijām ir stingri ieteicams izmantot rakstzīmju kopu utf8mb4 ar utf8mb4_bin kolāciju. Lai datubāzes, kurās tiek izmantota rakstzīmju kopa utf8mb3 un utf8mb3_bin kolācija, pārveidotu uz rakstzīmju kopu utf8mb4 un utf8mb4_bin kolāciju, izpildiet norādījumus sadaļā Zabbix datubāzes rakstzīmju kopas un kolācijas labošana.

Deterministiskie trigeri ir jāizveido shēmas importēšanas laikā. MySQL un MariaDB gadījumā, ja ir iespējota binārā reģistrēšana, nav superlietotāja privilēģiju un MySQL konfigurācijas failā nav iestatīts `log_bin_trust_function_creators = 1`, ir jāiestata `GLOBAL log_bin_trust_function_creators = 1`.

Ja instalējat Zabbix no pakotnēm vai konteineriem, skatiet attiecīgos norādījumus.

Ja instalējat Zabbix no pirmkoda:

- Izveidojiet un konfigurējiet datubāzi un lietotāju. Drošības apsvērumu dēļ Zabbix datubāzes lietotājam izmantojiet drošu, unikālu paroli.

```
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 rakstzīmju kopu (pieņemot, ka atrodaties Zabbix pirmkoda saknes direktorijā). Zabbix starpniekservera datubāzei jāimportē tikai `schema.sql` (nevis `images.sql` un `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
```

`log_bin_trust_function_creators` var atspējot pēc tam, kad shēma ir veiksmīgi importēta:

```
mysql -uroot -p<password>

mysql> SET GLOBAL log_bin_trust_function_creators = 0;
mysql> quit;
```

PostgreSQL

Jums ir nepieciešams datubāzes lietotājs ar tiesībām izveidot datubāzes objektus.

Ja instalējat Zabbix no pakotnēm vai konteineriem, skatiet attiecīgās instrukcijas.

Ja instalējat Zabbix no avota koda:

- Izveidojiet datubāzes lietotāju.

Tālāk norādītā čaulas komanda izveidos lietotāju zabbix. Kad tiek prasīts, norādiet paroli un atkārtojiet to (ņemiet vērā, ka vispirms var tikt prasīta sudo parole):

```
sudo -u postgres createuser --pwprompt zabbix
```

- Izveidojiet datubāzi.

Tālāk norādītā čaulas komanda izveidos datubāzi zabbix (pēdējais parametrs), par kuras īpašnieku būs iepriekš izveidotais lietotājs (-O zabbix).

```
sudo -u postgres createdb -O zabbix -E Unicode -T template0 zabbix
```

- Importējiet sākotnējo shēmu un datus (pieņemot, ka atrodaties Zabbix avota koda saknes direktorijā). Zabbix starpniekservera datubāzei jāimportē tikai schema.sql (nevis images.sql un data.sql).

```
cd database/postgresql
cat schema.sql | sudo -u zabbix psql zabbix
#### apstājieties šeit, ja veidojat datubāzi Zabbix starpniekserverim
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 citas komandas, piemēram:

 psql -U <username>

 Ja rodas problēmas ar datubāzes iestatīšanu, lūdzu, sazinieties ar savu datubāzes administratoru.

TimescaleDB

Norādījumi TimescaleDB izveidei un konfigurēšanai ir sniegti atsevišķā sadaļā.

Elasticsearch un ClickHouse

Zabbix var arī glabāt vēstures datus Elasticsearch vai ClickHouse. Iestatīšanas norādījumus skatiet saistītajās lapās.

SQLite

SQLite izmantošana tiek atbalstīta tikai **Zabbix starpniekserverim**!

Datubāze tiks automātiski izveidota, ja tā nepastāv.

Atgriezieties uz instalēšanas sadaļu.

Zabbix datubāzes rakstzīmju kopas un salīdzināšanas kārtības labošana

MySQL/MariaDB

Vēsturiski MySQL un tā atvasinājumi izmantoja 'utf8' kā aizstājvārdu utf8mb3 - MySQL paša 3 baitu standarta UTF8 ieviešanai, kas 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 noņemts, kamēr 'utf8' kļūs par atsauci uz 'utf8mb4'. Kopš Zabbix 6.0 tiek atbalstīts 'utf8mb4'. Lai izvairītos no turpmākām problēmām, ir ļoti ieteicams izmantot 'utf8mb4'. Vēl viena pārejas uz 'utf8mb4' priekšrocība ir papildu Unicode rakstzīmju atbalsts.

> **Warning:**
> Tā kā versijas pirms Zabbix 6.0 neatbalsta utf8mb4, pirms utf8mb4 konvertēšanas noteikti vispirms atjauniniet Zabbix serveri un DB shēmu uz 6.0.x vai jaunāku versiju.

1. Pārbaudiet datubāzes rakstzīmju kopu un salīdzināšanas kārtulu.

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 redzam, rakstzīmju kopa šeit nav 'utf8mb4' un salīdzināšanas kārtula nav 'utf8mb4_bin', tāpēc tās ir jālabo.

2. Apturiet Zabbix.

3. Izveidojiet datubāzes rezerves kopiju!

4. Labo rakstzīmju kopu un salīdzināšanas kārtulu datubāzes līmenī:

```
alter database <your DB name> character set utf8mb4 collate utf8mb4_bin;
```

Labotās vērtības:

```
mysql> SELECT @@character_set_database, @@collation_database;
+--------------------------+----------------------+
| @@character_set_database | @@collation_database |
+--------------------------+----------------------+
| utf8mb4                  | utf8mb4_bin          |
+--------------------------+----------------------+
```

5. Ielādējiet skriptu, lai labotu rakstzīmju kopu un salīdzināšanas kārtulu 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' paredzēts izmantot nedaudz vairāk diska vietas.

7. Ja nav kļūdu - varat izveidot datubāzes rezerves kopiju ar izlaboto datubāzi.

8. Startējiet Zabbix.

Datu bāzes jaunināšana uz primārajām atslēgām

Pārskats

Šajā sadaļā ir sniegtas instrukcijas, kā manuāli jaunināt esošajās instalācijās esošās tabulas 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 klasterizētās konfigurācijās, palīdzot mērogošanā un nodrošinot sistēmas uzticamību pat tad, ja daži serveri neizdodas.

> **Attention:**
> Šajā lapā sniegtās instrukcijas ir paredzētas pieredzējušiem lietotājiem, un tās var būt jāpielāgo jūsu konkrētajai konfigurācijai. Jaunināšana uz primārajām atslēgām var būt laikietilpīga un resursietilpīga. Pārliecinieties, ka ir pieejams pietiekami daudz brīvas diska vietas; atkarībā no datubāzes lieluma un saglabātajiem datiem process var prasīt līdz pat 2.5 reizēm vairāk vietas, nekā pašlaik aizņem vēstures tabulas.

Primārās atslēgas tiek izmantotas visām tabulām jaunās instalācijās kopš Zabbix 6.0.

Nav automātiskas datubāzes jaunināšanas uz primārajām atslēgām; tomēr esošās instalācijas var jaunināt manuāli **pēc** Zabbix servera jaunināšanas uz 6.0 vai jaunāku versiju.

> **Attention:**
> Kopš Zabbix 7.0 tabulu jaunināšana uz primārajām atslēgām arī jaunina tabulas, lai izmantotu dubultprecizitātes datu tipus.

 Ja jums ir Zabbix 7.0 (vai jaunāka versija), tabulas jau izmanto dubultprecizitāti. Tomēr šajā lapā sniegtās instrukcijas joprojām var izmantot, lai jauninātu tabulas uz primārajām atslēgām, neietekmējot tabulas, kas jau izmanto dubultprecizitāti.

 Ja jums ir Zabbix 6.4 (vai vecāka versija), apsveriet iespēju vispirms jaunināt tabulas uz dubultprecizitāti. Papildu informāciju skatiet Jaunināšana uz paplašināta diapazona skaitliskajām vērtībām Zabbix 7.0 dokumentācijā.

Instrukcijas ir pieejamas šādām platformām:

- MariaDB/MySQL
- PostgreSQL
- TimescaleDB

Svarīgas piezīmes

Lai veiktu datubāzes jaunināšanu:

1. Apturiet Zabbix serveri.

Ir stingri ieteicams apturēt Zabbix serveri uz jaunināšanas laiku. Tomēr, ja tas ir absolūti nepieciešams, jaunināšanu 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 jaunināšanas negūs labumu.

Ja datubāzē tiek izmantotas partīcijas, sazinieties ar DB administratoru vai Zabbix atbalsta dienestu, lai saņemtu palīdzību.

CSV failus pēc veiksmīgas jaunināšanas uz primārajām atslēgām var noņemt.

Pēc izvēles Zabbix lietotāja saskarne var tikt pārslēgta uzturēšanas režīmā.

MariaDB/MySQL

Eksportēšana un importēšana jāveic tmux/screen, lai nodrošinātu, ka sesija netiek pārtraukta.

Skatiet arī: Svarīgas piezīmes

MySQL 8.4+ ar mysqlsh

Šo metodi var izmantot ar darbojošos Zabbix serveri, taču jaunināšanas laikā ieteicams serveri apturēt. MySQL Shell (*mysqlsh*) ir jābūt instalētam un spējīgam izveidot savienojumu ar DB.

- Piesakieties MySQL konsolē kā root (ieteicams) vai kā jebkurš lietotājs ar FILE privilēģijām.
- Startējiet MySQL ar iespējotu local_infile mainīgo.
- Pārdēvējiet vecās tabulas un izveidojiet jaunas tabulas, izpildot `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 tiek izmantots socket savienojums, var būt nepieciešams norādīt ceļu.

```
sudo mysqlsh -uroot -S /run/mysqld/mysqld.sock --no-password -Dzabbix
```

Pārslēdzieties uz JavaScript režīmu, izmantojot:

```
\js
```

Pēc tam izpildiet zemāk redzamo kodu (CSVPATH var mainīt pēc vajadzības):

```
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 pēc migrācijas 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 jebkurš lietotājs ar FILE privilēģijām.
- Ja migrāciju veicat ar darbojošos Zabbix serveri, palaidiet MySQL ar iespējotu mainīgo local_infile.
- Pārdēvējiet vecās tabulas un izveidojiet jaunas tabulas, izpildot `history_upgrade_prepare.sql`:

```
mysql -uzabbix -p<password> zabbix < /usr/share/zabbix/sql-scripts/mysql/option-patches/history_upgrade_pr
```

Migrācija ar apturētu serveri

`max_execution_time` (MySQL) vai `max_statement_time` (MariaDB) pirms datu migrēšanas ir jāatspējo, lai migrācijas laikā nerastos noildze.

MySQL:

```
SET @@max_execution_time=0;
```

MariaDB:

```
SET @@max_statement_time=0;
```

```
INSERT IGNORE INTO history SELECT * FROM history_old;
INSERT IGNORE INTO history_uint SELECT * FROM history_uint_old;
INSERT IGNORE INTO history_str SELECT * FROM history_str_old;
INSERT IGNORE INTO history_log SELECT * FROM history_log_old;
INSERT IGNORE INTO history_text SELECT * FROM history_text_old;
```

Lai dzēstu vecās tabulas, izpildiet pēc migrācijas 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 šajā direktorijā esošajiem failiem. Šādā gadījumā attiecīgi labojiet failu ceļus vaicājumos vai jaunināšanas laikā iestatiet *secure_file_priv* vērtību kā tukšu virkni.

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 (iepriekšējā piemērā - '/var/lib/mysql-files/').

Papildinformāciju skatiet MySQL dokumentācijā vai MariaDB dokumentācijā.

`max_execution_time` (MySQL) vai `max_statement_time` (MariaDB) pirms datu eksportēšanas ir jāatspējo, lai eksportēšanas laikā nerastos noildze.

MySQL:

```
SET @@max_execution_time=0;
```

MariaDB:

```
SET @@max_statement_time=0;
```

```
SELECT * INTO OUTFILE '/var/lib/mysql-files/history.csv' FIELDS TERMINATED BY ',' ESCAPED BY '"' LINES TER
LOAD DATA INFILE '/var/lib/mysql-files/history.csv' IGNORE INTO TABLE history FIELDS TERMINATED BY ',' ESC

SELECT * INTO OUTFILE '/var/lib/mysql-files/history_uint.csv' FIELDS TERMINATED BY ',' ESCAPED BY '"' LINE
LOAD DATA INFILE '/var/lib/mysql-files/history_uint.csv' IGNORE INTO TABLE history_uint FIELDS TERMINATED

SELECT * INTO OUTFILE '/var/lib/mysql-files/history_str.csv' FIELDS TERMINATED BY ',' ESCAPED BY '"' LINES
LOAD DATA INFILE '/var/lib/mysql-files/history_str.csv' IGNORE INTO TABLE history_str FIELDS TERMINATED BY

SELECT * INTO OUTFILE '/var/lib/mysql-files/history_log.csv' FIELDS TERMINATED BY ',' ESCAPED BY '"' LINES
LOAD DATA INFILE '/var/lib/mysql-files/history_log.csv' IGNORE INTO TABLE history_log FIELDS TERMINATED BY

SELECT * INTO OUTFILE '/var/lib/mysql-files/history_text.csv' FIELDS TERMINATED BY ',' ESCAPED BY '"' LINE
LOAD DATA INFILE '/var/lib/mysql-files/history_text.csv' IGNORE INTO TABLE history_text FIELDS TERMINATED
```

Lai dzēstu vecās tabulas, izpildiet pēc migrācijas norādījumus.

PostgreSQL

Eksportēšana un importēšana jāveic tmux/screen, lai nodrošinātu, ka sesija netiek pārtraukta. Instalācijām ar TimescaleDB izlaidiet šo sadaļu un turpiniet ar 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
```

Migrācija ar apturētu serveri

- Eksportējiet pašreizējo vēsturi, importējiet to pagaidu tabulā, 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 padomus INSERT veiktspējas uzlabošanai: PostgreSQL: Bulk Loading Huge Amounts of Data, Checkpoint Distance and Amount of WAL.

- Izpildiet pēc-migrācijas norādījumus, lai dzēstu vecās tabulas.

Migrācija ar darbojošos serveri

- Eksportējiet pašreizējo vēsturi, importējiet to pagaidu tabulā, 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 pēc-migrācijas norādījumus, lai dzēstu vecās tabulas.

PostgreSQL + TimescaleDB

Eksportēšana un importēšana jāveic tmux/screen, lai nodrošinātu, ka sesija netiek pārtraukta. Zabbix serverim jaunināšanas laikā 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 pēc-migrācijas 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

Auditlog tabulas sagatavošana sadalīšanai pa nodalījumiem

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` operācijas laikā. Tomēr ir ļoti ieteicams indeksus pārbūvēt arī manuāli, izmantojot `OPTIMIZE TABLE` priekšrakstu, lai nodrošinātu optimālu datubāzes veiktspēju.

Indeksu pārbūvei uz laiku var būt nepieciešama papildu diska vieta tikpat lielā apjomā, cik izmanto pati tabula. Lai iegūtu pašreizējo datu un indeksu apjomu, varat izpildīt šādus priekšrakstus:

```
ANALYZE TABLE auditlog;
SHOW TABLE STATUS LIKE 'auditlog';
```

Ja pieejamā diska vieta ir ierobežota, izpildiet norādījumus sadaļā Primārās atslēgas maiņa bez dīkstāves. Pieejamas arī citas iespējas:

- MySQL parametra `sort_buffer_size` palielināšana var palīdzēt samazināt diska vietas patēriņu, manuāli pārbūvējot indeksus. Tomēr šī mainīgā modificēšana var ietekmēt datubāzes kopējo atmiņas patēriņu.
- Apsveriet iespēju atbrīvot vietu, dzēšot, iespējams, nevajadzīgus datus.
- Apsveriet iespēju samazināt *Datu glabāšanas perioda* housekeeper parametru pirms housekeeper palaišanas.

Primārās atslēgas maiņa 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 (pēc izvēles, bet ļoti ieteicams; skatiet Svarīgas piezīmes par indeksu pārbūvi).

```
OPTIMIZE TABLE auditlog;
```

Primārās atslēgas maiņa bez dīkstāves

Šeit ir aprakstīta primārās atslēgas manuāla maiņas metode. Alternatīvi varat izmantot Percona rīkkopu pt-online-schema-change. Šī rīkkopa automātiski veic tālāk norādītās darbības, vienlaikus samazinot vietu, kas nepieciešama `auditlog` tabulas maiņai.

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. Apmainiet 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 veikt pa daļām (vairāki `INSERT INTO` priekšraksti ar `WHERE clock` nosacījumiem pēc vajadzības), lai izvairītos no pārmērīga resursu patēriņa.

4. Dzēsiet veco tabulu.

```
DROP TABLE auditlog_old;
```

PostgreSQL

Svarīgas piezīmes par indeksu pārbūvi

PostgreSQL automātiski pārbūvē primārās atslēgas indeksus `ALTER TABLE` operācijas laikā. Tomēr ir ļoti ieteicams indeksus pārbūvēt arī manuāli, izmantojot `REINDEX TABLE CONCURRENTLY` paziņojumu, lai nodrošinātu optimālu datubāzes veiktspēju.

Indeksu pārbūvei uz laiku var būt nepieciešama līdz pat trīsreiz lielāka diska vieta nekā pašlaik aizņem indeksi. Lai uzzinātu pašreizējo indeksu izmēru, varat izpildīt šādu vaicājumu:

```
SELECT pg_size_pretty(pg_indexes_size('auditlog'));
```

Ja pieejamā diska vieta ir ierobežota, izpildiet norādījumus sadaļā Primārās atslēgas maiņa bez dīkstāves. Ir pieejamas arī citas iespējas:

- PostgreSQL parametra `maintenance_work_mem` palielināšana var palīdzēt samazināt diska vietas patēriņu, manuāli pārbūvējot indeksus. Tomēr šī mainīgā modificēšana var ietekmēt datubāzes kopējo atmiņas patēriņu.
- Ja jums ir cits disks vai tablespace ar vairāk brīvas vietas, varat apsvērt pagaidu krātuves atrašanās vietas maiņu indeksu pārbūvei. Varat iestatīt PostgreSQL parametru `temp_tablespaces`, lai norādītu citu tablespace pagaidu objektiem.
- Apsveriet iespēju atbrīvot vietu, dzēšot potenciāli nevajadzīgus datus.
- Apsveriet iespēju samazināt *Datu glabāšanas perioda* housekeeper parametru pirms housekeeper palaišanas.

Primārās atslēgas maiņa 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 (pēc izvēles, bet ļoti ieteicams; skatiet Svarīgas piezīmes par indeksu pārbūvi).

```
REINDEX TABLE CONCURRENTLY auditlog;
```

Primārās atslēgas maiņa bez dīkstāves

Šeit ir aprakstīta primārās atslēgas maiņas manuālā metode. Alternatīvi var apsvērt `pg_repack` paplašinājumu jaunas tabulas izveidei, datu kopēšanai un tabulu apmaiņai.

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. Apmainiet tabulas.

```
ALTER TABLE auditlog RENAME TO auditlog_old;
ALTER TABLE auditlog_new RENAME TO auditlog;
```

3. Nokopējiet datus no vecās tabulas uz jauno tabulu.

```
INSERT INTO auditlog SELECT * FROM auditlog_old;
```

To var veikt pa daļām (vairāki `INSERT INTO` priekšraksti ar `WHERE clock` nosacījumiem pēc vajadzības), lai izvairītos no pārmērīga resursu patēriņa.

4. Dzēsiet veco tabulu.

```
DROP TABLE auditlog_old;
```

Skatīt arī

- Datubāzes jaunināšana uz primārajām atslēgām

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 ir instalēts atbalstītas versijas TimescaleDB paplašinājums supported version. Instalēšanas norādījumus skatiet TimescaleDB dokumentācijā.

Warning:
Pirms TimescaleDB instalēšanas instalējiet atbalstītu PostgreSQL laidienu no oficiālās 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 citu datubāzes shēmu, nevis '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 ir 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 migrācija 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

TimescaleDB iebūvētā saspiešana tiek atbalstīta visām Zabbix tabulām, kas ir TimescaleDB hypertables. Jauninašanas vai migrācijas uz TimescaleDB laikā lielo tabulu sākotnējā saspiešana var aizņemt daudz laika.

Ņemiet vērā, ka saspiešana tiek atbalstīta ar "timescale" Timescale Community licenci un netiek atbalstīta ar "apache" Apache 2.0 licenci. Ja Zabbix konstatē, ka saspiešana netiek atbalstīta, Zabbix serveris žurnālā tiek ierakstīts brīdinājuma ziņojums, un lietotāji nevar iespējot saspiešanu lietotāja saskarnē.

Note:
Pirms saspiešanas izmantošanas lietotājiem ieteicams iepazīties ar saspiešanu TimescaleDB dokumentācijā.

Ņemiet vērā, ka saspiešana nosaka noteiktus ierobežojumus, konkrēti:

- Saspiestu chunk modifikācijas (ievietošana, dzēšana, atjaunināšana) nav atļautas
- Saspiestām tabulām shēmas izmaiņas nav atļautas.

Saspiešanas iestatījumus var mainīt blokā *Vēstures, tendenču un audita žurnāla saspiešana* sadaļā *Administrēšana* > *Apkope* Zabbix lietotāja saskarnē.

Parametrs	Noklusējums	Komentāri
Iespējot saspiešanu	Iespējots	Atzīmējot vai noņemot atzīmi izvēles rūtiņā, saspiešana netiek aktivizēta/deaktivizēta nekavējoties. Tā kā saspiešanu apstrādā Housekeeper, izmaiņas stāsies spēkā līdz pat 2 reizēm `HousekeepingFrequency` stundās (iestatīts `zabbix_server.conf`) Pēc saspiešanas atspējošanas jauni chunk, kas nonāk saspiešanas periodā, netiks saspiesti. Tomēr visi iepriekš saspiestie dati paliks saspiesti. Lai atspiestu iepriekš saspiestus chunk, izpildiet norādījumus TimescaleDB dokumentācijā.
Saspiest ierakstus, kas vecāki par	7d	Ja tiek jaunināts no vecākām Zabbix versijām ar TimescaleDB atbalstu, saspiešana pēc noklusējuma nebūs iespējota. Šis parametrs nevar būt mazāks par 7 dienām. Neizmaināmu saspiestu chunk dēļ visi novēlotie dati (piem., dati, ko aizkavējis starpniekserveris), kas ir vecāki par šo vērtību, tiks noraidīti.

Lai uzlabotu tendenču atjaunināšanas veiktspēju, varat samazināt "chunk_time_interval" parametru 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 chunk lielums paliek sistēmas pieejamo resursu robežās.

Elasticsearch iestatīšana

Zabbix var glabāt vēstures datus Elasticsearch kā alternatīvu relāciju datubāzei.

Attention:
Elasticsearch atbalsts pašlaik ir eksperimentāls.

Šajā rokasgrāmatā ir aprakstīta atbalstīto versiju Elasticsearch iestatīšana. Ja izmantojat citu versiju, dažas funkcijas var nedarboties, kā paredzēts.

Elasticsearch var glabāt šādus vērtību tipus:

Vienuma vērtības tips	Datubāzes tabula	Elasticsearch tips
Skaitlisks (neparakstīts)	history_uint	uint
Skaitlisks (peldošs)	history	dbl
Rakstzīme	history_str	str
Žurnāls	history_log	log
Teksts	history_text	text
Binārs	history_bin	nav atbalstīts ar Zabbix
JSON	history_json	json

Note:
Elasticsearch nepieņem JSON masīvus. JSON vērtībai jābūt vai nu vienam objektam, vai objektu kopai.

Svarīgas piezīmes

- Elasticsearch nepieciešams libcurl. Skatiet prasības, lai iegūtu detalizētu informāciju.
- housekeeper nedzēš datus no Elasticsearch. Lai kontrolētu, cik ilgi dati tiek saglabāti, skatiet Index Lifecycle Management.
- Zabbix programmā Elasticsearch netiek aprēķinātas vai saglabātas tendences. Apsveriet iespēju pagarināt vēstures glabāšanas periodu, lai saglabātu vecākus datus.
- Ja tiek izmantots Elasticsearch, diapazona vaicājumi, kas izgūst vērtības no datubāzes, ir ierobežoti ar datu glabāšanas perioda laikspiedolu.
- Elasticsearch nav atbalstīts Zabbix starpniekserveriem.

Ja Elasticsearch vēl nav instalēts, pirms turpināšanas 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 uz datumiem balstītos indeksos

Tā vietā, lai visus vēstures datus rakstītu vienā indeksā (piemēram, `uint`), Elasticsearch var sadalīt šos datus vairākos uz datumiem balstītos indeksos (piemēram, `uint-2026-01-01`, `uint-2026-01-02`). Tas atvieglo datu apjoma un glabāšanas termiņa pārvaldību laika gaitā.

Lai to iespējotu, jums ir jāveic šādas darbības:

- Izveidojiet indeksa veidni katram vērtības tipam, ko vēlaties glabāt — tā nosaka Elasticsearch, kādu kartējumu piemērot, kad tas automātiski izveido jaunu uz datumu balstītu indeksu.
- Izveidojiet ingest pipeline katram vērtības tipam — tas apstrādā katru ienākošo vērtību un novirza to uz pareizo uz datumu balstīto indeksu.

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

Savā Zabbix servera konfigurācijas failā (`zabbix_server.conf`) iestatiet parametru `HistoryProvider`.

Piemēram, lai Elasticsearch glabātu *Character*, *Log*, *Text* un *JSON* tipa vērtības (vienlaikus *Numeric* vērtības saglabājot datubāzē):

```
HistoryProvider=elasticsearch;value_types="str,log,text,json",url=http://localhost:9200
```

Ja izmantojat uz datumu balstītus indeksus, pievienojiet parametram `date_index=1`:

```
HistoryProvider=elasticsearch;value_types="str,log,text,json",url=http://localhost:9200,date_index=1
```

Pēc izmaiņu veikšanas restartējiet Zabbix serveri:

```
systemctl restart zabbix-server
```

Zabbix lietotāja saskarnes konfigurēšana

Zabbix lietotāja saskarnes konfigurācijas failā (`zabbix.conf.php`) iestatiet mainīgo `$HISTORY_PROVIDERS`, lai tas atbilstu servera konfigurācijai:

```
$HISTORY_PROVIDERS[] = [
  'types' => ['str','log','text','json'],
  'provider' => 'elasticsearch',
  'url' => 'http://localhost:9200'
];
```

Problēmu novēršana

Tālāk norādītās darbības var palīdzēt novērst problēmas ar jūsu Elasticsearch konfigurāciju:

1. Pārbaudiet Elasticsearch vai Zabbix servera žurnālus, lai atrastu kļūdas.
2. Lai identificētu lēnus vaicājumus, izmantojiet `log_slow_queries` opciju Zabbix servera konfigurācijas parametrā `HistoryProvider`.
3. Pārliecinieties, ka Elasticsearch atļauj piekļuvi no Zabbix servera un Zabbix lietotāja saskarne.
4. Pārliecinieties, ka `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": {} }
```

5. Pārbaudiet, vai mappingi, veidnes un ingest pipeline 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.

6. Pārbaudiet, vai kādi shards nav kļūmes stāvoklī; Elasticsearch restartēšana var šo problēmu atrisināt.

7. Veiciet vaicājumu Elasticsearch, lai pārbaudītu, vai Zabbix savāktie dati ir saglabāti, piemēram:

```
curl 'http://localhost:9200/json/_search' \
 -H 'Content-Type: application/json' \
 -d '{
   "query": {
     "term": {
       "itemid": 42269
     }
   }
 }'
```

8. Ja nepieciešams atiestatīt Elasticsearch konfigurāciju un sākt no jauna, varat dzēst visus indeksus, veidnes un ingest pipeline:

```
curl -X DELETE "http://localhost:9200/_all"
curl -X DELETE "http://localhost:9200/_template/*"
curl -X DELETE "http://localhost:9200/_ingest/pipeline/*"
```

ClickHouse iestatīšana

Zabbix var glabāt vēstures datus ClickHouse kā alternatīvā relāciju datubāzē.

> **Note:**
> ClickHouse var izmantot arī kā telemetrijas datu avotu lietojumprogrammu veiktspējas uzraudzībai (APM); plašāku informāciju skatiet *Administrēšana* > *Datu avots*.

Šajā rokasgrāmatā aprakstīta atbalstīto versiju ClickHouse iestatīšana. Ja izmantojat citu versiju, dažas funkcijas var nedarboties, kā paredzēts.

ClickHouse var glabāt šādus vērtību tipus:

Vienuma vērtības tips	Datubāzes tabula	ClickHouse tips
Skaitlis (bez zīmes)	history_uint	uint
Skaitlis (peldošais komats)	history	dbl
Rakstzīme	history_str	str
Žurnāls	history_log	log
Teksts	history_text	text
Binārs	history_bin	Zabbix neatbalsta
JSON	history_json	json

Note:
ClickHouse nepieņem JSON masīvus. JSON vērtībai jābūt vai nu vienam objektam, vai objektu kopai. Turklāt ClickHouse apstrādā JSON atslēgas ar NULL tāpat kā trūkstošas atslēgas.

Svarīgas piezīmes

- Housekeeper nedzēš datus no ClickHouse. Lai kontrolētu, cik ilgi dati tiek glabāti, konfigurējiet ClickHouse datu glabāšanas periodu programmā ClickHouse.
- Zabbix **neaprēķina un nesaglabā trendus** programmā ClickHouse. Apsveriet iespēju pagarināt vēstures glabāšanas periodu, lai saglabātu vecākus datus.
- Ja vēlaties migrēt vēstures datus no esošas Zabbix datubāzes (MySQL vai PostgreSQL) uz ClickHouse, skatiet ClickHouse shēmas un vēstures migrācijas skriptus.
- ClickHouse netiek atbalstīts kā vēstures datu glabātuve Zabbix starpniekserveriem, taču to var izmantot kā telemetrijas datu avotu Zabbix starpniekserverim ar iespējotu APM.
- Ja jums ir ClickHouse klasteris, skatiet sadaļu ClickHouse klasteru konfigurēšana.

ClickHouse konfigurēšana

Jums ir jāizveido un jākonfigurē Zabbix datubāze un lietotājs, kā arī jāimportē datubāzes shēma.

Šajā rokasgrāmatā ir sniegtas instrukcijas par ClickHouse Docker vai pakotņu instalācijām.

Docker

1. Izveidojiet un konfigurējiet Zabbix datubāzi un lietotāju, palaižot ClickHouse konteineru:

```
sudo docker run -d \
  --name clickhouse \
  -e CLICKHOUSE_DB=zabbix \
  -e CLICKHOUSE_USER=zabbix \
  -e CLICKHOUSE_DEFAULT_ACCESS_MANAGEMENT=1 \
  -e CLICKHOUSE_PASSWORD=<password> \
  -p 8123:8123/tcp \
  -p 9000:9000/tcp \
  --cap-add=SYS_NICE \
  --cap-add=NET_ADMIN \
  --cap-add=IPC_LOCK \
  --ulimit nofile=262144:262144 \
  -v clickhouse_data:/var/lib/clickhouse \
  clickhouse/clickhouse-server:26.4
```

2. Pārliecinieties, ka ClickHouse darbojas un ka varat tam pieslēgties:

```
sudo docker exec -it clickhouse \
  clickhouse-client \
  --query "SELECT version()"

#### 26.4.4.38
```

3. Importējiet datubāzes shēmu, izmantojot `history_all.sh` skriptu no sava Zabbix direktorija:

```
./usr/share/zabbix/sql-scripts/clickhouse/history_all.sh \
  --user zabbix \
  --password <password> \
  --db zabbix \
  --server http://localhost:8123
```

Note:
ClickHouse datu glabāšanas periods (Time-To-Live jeb TTL) pēc noklusējuma ir 31 diena. Lai to mainītu, importējot datubāzes shēmu izmantojiet opciju `--ttl` (piemēram, `--ttl 604800` 7 dienām), vai arī konfigurējiet to vēlāk.

4. Pārbaudiet, vai tabulas ir izveidotas:

```
sudo docker exec -it clickhouse \
  clickhouse-client \
  --user zabbix \
  --password <password> \
```

```
  --query "SHOW TABLES FROM zabbix"

#### history
#### history_json
#### history_log
#### history_str
#### history_text
#### history_uint
```

Pakotnes

1. Startējiet ClickHouse:

```
sudo service clickhouse-server start
```

2. Izveidojiet un konfigurējiet Zabbix datubāzi un lietotāju:

```
clickhouse-client --user default

localhost :) CREATE DATABASE IF NOT EXISTS zabbix
localhost :) CREATE USER IF NOT EXISTS zabbix IDENTIFIED WITH sha256_password BY '<password>'
localhost :) GRANT CREATE, ALTER, DROP, INSERT, SELECT, UPDATE, OPTIMIZE ON zabbix.* TO zabbix
localhost :) quit
```

3. Importējiet datubāzes shēmu, izmantojot `history_all.sh` skriptu no jūsu Zabbix direktorijas:

```
./usr/share/zabbix/sql-scripts/clickhouse/history_all.sh \
  --user zabbix \
  --password <password> \
  --db zabbix \
  --server http://localhost:8123
```

> **Note:**
> ClickHouse datu glabāšanas periods (Time-To-Live jeb TTL) pēc noklusējuma ir 31 diena. Lai to mainītu, importējot datubāzes shēmu izmantojiet opciju `--ttl` (piemēram, `--ttl 604800` 7 dienām), vai arī konfigurējiet to vēlāk.

4. Pārbaudiet, vai tabulas ir izveidotas:

```
clickhouse-client \
  --user zabbix \
  --password <password> \
  --query "SHOW TABLES FROM zabbix"

#### history
#### history_json
#### history_log
#### history_str
#### history_text
#### history_uint
```

Zabbix servera konfigurēšana

Savā Zabbix servera konfigurācijas failā (`zabbix_server.conf`) iestatiet parametru `HistoryProvider`.

Piemēram, lai glabātu visus atbalstītos vērtību tipus ClickHouse:

```
HistoryProvider=clickhouse;value_types="uint,dbl,str,log,text,json",url=http://localhost:8123,db=zabbix,us
```

Pēc izmaiņu veikšanas restartējiet Zabbix serveri:

```
systemctl restart zabbix-server
```

Zabbix lietotāja saskarnes konfigurēšana

Zabbix lietotāja saskarnes konfigurācijas failā (`zabbix.conf.php`) iestatiet mainīgo `$HISTORY_PROVIDERS`, lai tas atbilstu servera konfigurācijai:

```
$HISTORY_PROVIDERS[] = [
  'types' => ['uint','dbl','str','log','text','json'],
```

```
  'provider' => 'clickhouse',
  'url' => 'http://localhost:8123',
  'db' => 'zabbix',
  'username' => 'zabbix',
  'password' => '<password>'
];
```

Papildu konfigurācija

Tālāk norādītās darbības ir izvēles. Tās nav nepieciešamas pamata iestatīšanai.

ClickHouse datu glabāšanas perioda konfigurēšana

ClickHouse datu glabāšanas periods (Time-To-Live jeb TTL) pēc noklusējuma ir 31 diena. Lai to mainītu, izpildiet tālāk norādītās komandas.

Tālāk sniegtajos piemēros tiek izmantots Docker. Ja ClickHouse ir instalēts, izmantojot pakotnes, vaicājumus izpildiet tieši ClickHouse klientā.

1. Mainiet tabulu (aizstājiet `history_json` un 3600 ar nepieciešamajām vērtībām):

```
sudo docker exec -it clickhouse \
  clickhouse-client \
  --user zabbix \
  --password <password> \
  --query "ALTER TABLE zabbix.history_json MODIFY TTL clock_ns + toIntervalSecond(3600)"
```

2. Nekavējoties lietojiet izmaiņas:

```
sudo docker exec -it clickhouse \
  clickhouse-client \
  --user zabbix \
  --password <password> \
  --query "OPTIMIZE TABLE zabbix.history_json MATERIALIZE TTL"
```

3. Pārbaudiet, vai datu glabāšanas periods ir mainījies:

```
sudo docker exec -it clickhouse \
  clickhouse-client \
  --user zabbix \
  --password <password> \
  --query "SHOW CREATE TABLE zabbix.history_json"

#### CREATE TABLE zabbix.history_json
#### (
####     `itemid` UInt64,
####     `clock_ns` DateTime64(9),
####     `value` JSON,
####     `value_str` String
#### )
#### ENGINE = MergeTree
#### PARTITION BY toDate(clock_ns)
#### PRIMARY KEY (itemid, clock_ns)
#### ORDER BY (itemid, clock_ns)
#### TTL clock_ns + toIntervalSecond(3600)
#### SETTINGS index_granularity = 8192
```

4. Restartējiet Zabbix serveri, lai atsvaidzinātu datu glabāšanas periodu sadaļā *Administrēšana* > *Mājturība*:

```
systemctl restart zabbix-server
```

ClickHouse klasteru konfigurēšana

Šī sadaļa sniedz ClickHouse klasteru konfigurēšanas darbības ar replikāciju un shardēšanu.

Tālāk norādītajās darbībās tiek izmantots Docker un ClickHouse `cluster_2S_2R` no ClickHouse dokumentācijas (darbinot ClickHouse 26.4.4.38 ar ClickHouse Keeper).

Pirms sākat, Zabbix vajadzībām ieteicams ClickHouse klastera priekšā izvietot slodzes balansētāju. Zabbix servera un lietotāja saskarnes konfigurācija atbalsta tikai vienu ClickHouse URL, tāpēc tā var norādīt tikai uz vienu mezglu. Ja šis mezgls pārstāj

darboties, slodzes balansētājs, izmantojot to pašu URL, pāradresēs Zabbix uz veselu mezglu.

1. Izveidojiet Zabbix datubāzi:

```
sudo docker exec -it clickhouse-01 \
  clickhouse-client \
  --user default \
  --query "CREATE DATABASE IF NOT EXISTS zabbix ON CLUSTER cluster_2S_2R ENGINE = Replicated('/clickhouse/
```

Attention:
`ON CLUSTER` klauzula sasniedz tikai tos mezglus, kas komandas izpildes brīdī ir klastera daļa. Ja vēlāk klasterim pievienojat mezglus, palaidiet šo komandu vēlreiz.

2. Izveidojiet un konfigurējiet Zabbix lietotāju:

```
sudo docker exec -it clickhouse-01 \
  clickhouse-client \
  --user default \
  --query "CREATE USER IF NOT EXISTS zabbix ON CLUSTER cluster_2S_2R IDENTIFIED WITH sha256_password BY '<

sudo docker exec -it clickhouse-01 \
  clickhouse-client \
  --user default \
  --query "GRANT ON CLUSTER cluster_2S_2R CREATE, ALTER, DROP, INSERT, SELECT, UPDATE, OPTIMIZE ON zabbix.
```

3. Importējiet datubāzes shēmu, izmantojot `history_all.sh` skriptu no jūsu Zabbix direktorijas:

```
./usr/share/zabbix/sql-scripts/clickhouse/history_all.sh \
  --user zabbix \
  --password <password> \
  --db zabbix \
  --server http://localhost:8123 \
  --engine "ReplicatedMergeTree()"
```

Note:
ClickHouse datu glabāšanas periods (Time-To-Live jeb TTL) pēc noklusējuma ir 31 diena. Lai to mainītu, importējot datubāzes shēmu izmantojiet `--ttl` opciju (piemēram, `--ttl 604800` 7 dienām) vai konfigurējiet to vēlāk.

Note:
Ja jūsu ClickHouse klasterim ir nepieciešami skaidri norādīti replica ceļi (piemēram, ja jūsu ClickHouse servera konfigurācijā nav iestatīti noklusējuma ceļi replicētajām tabulām), palaidiet shēmas skriptus (`history_schema.sh`, `history_uint_schema.sh` u. c.) atsevišķi un katrai tabulai iestatiet atšķirīgu ceļu:

```
./usr/share/zabbix/sql-scripts/clickhouse/history_schema.sh \
  --user zabbix \
  --password <password> \
  --db zabbix \
  --server http://localhost:8123 \
  --engine "ReplicatedMergeTree('/clickhouse/tables/{shard}/history', '{replica}')"
```

Plašāku informāciju par ClickHouse replikāciju skatiet sadaļās Replicating data un Replicated* table engines ClickHouse dokumentācijā.

4. Konfigurējiet shardēšanu katram tabulas tipam. Pārdēvējiet tabulas un izveidojiet `Distributed` tabulu ar sākotnējo nosaukumu, norādot uz pārdēvēto tabulu. Šī `Distributed` tabula ir tā, no kuras Zabbix faktiski lasa un kurā raksta; tabula automātiski sadala datus pa shardiem. Piemēram, tabulai `history`:

```
sudo docker exec -it clickhouse-01 \
  clickhouse-client \
  --user default \
  --query "RENAME TABLE zabbix.history TO zabbix.history_local"

sudo docker exec -it clickhouse-01 \
  clickhouse-client \
  --user default \
```

```
  --query "CREATE TABLE zabbix.history AS zabbix.history_local ENGINE = Distributed('cluster_2S_2R', 'zabb
```

Note:
Plašāku informāciju par ClickHouse shardēšanu skatiet Distributed table engine ClickHouse dokumentācijā.

5. Konfigurējiet Zabbix serveri.

6. Konfigurējiet Zabbix lietotāja saskarni.

7. Pārbaudiet, vai Zabbix dati ir sadalīti pa shardiem. Rindu skaits `zabbix.history_local` var mainīties starp atkārtotām pārbaudēm, jo Zabbix serveris nepārtraukti ievieto jaunus datus. Palaidiet šo komandu katrā mezglā:

```
sudo docker exec -it clickhouse-01 \
  clickhouse-client \
  --user default \
  --query "SELECT count(*) FROM zabbix.history_local"

#### 2446 (clickhouse-01)
#### 2471 (clickhouse-02)
#### 2494 (clickhouse-03)
```

8. Pārbaudiet, vai replikācija ir sinhronizēta (`active_replicas` jāatbilst `total_replicas`):

```
sudo docker exec -it clickhouse-01 \
  clickhouse-client \
  --user default \
  --query "SELECT database, table, is_leader, total_replicas, active_replicas FROM system.replicas WHERE d

#### zabbix  history_json_local  1 2 2
#### zabbix  history_local       1 2 2
#### zabbix  history_log_local   1 2 2
#### zabbix  history_str_local   1 2 2
#### zabbix  history_text_local  1 2 2
#### zabbix  history_uint_local  1 2 2
```

Problēmu novēršana

Tālāk norādītās darbības var palīdzēt novērst problēmas ar jūsu ClickHouse konfigurāciju:

1. Pārbaudiet ClickHouse vai Zabbix servera žurnālus, lai atrastu kļūdas.
2. Lai identificētu lēnus vaicājumus, izmantojiet `log_slow_queries` opciju Zabbix servera konfigurācijas parametrā `HistoryProvider`.
3. Pārliecinieties, ka ClickHouse atļauj piekļuvi no Zabbix servera un Zabbix lietotāja saskarnes.
4. Veiciet vaicājumu ClickHouse, lai pārbaudītu, vai Zabbix savāktie dati tiek saglabāti, piemēram:

```
sudo docker exec -it clickhouse \
  clickhouse-client \
  --user zabbix \
  --password <password> \
  --query "SELECT * FROM zabbix.history_uint WHERE itemid = 42269"
```

Ar konkrētiem izplatījumiem saistītas piezīmes par NGINX iestatīšanu Zabbix vajadzībām

RHEL

RHEL 8, 9 un 10 versijā NGINX ir iekļauts AppStream, tāpēc papildu repozitoriji nav nepieciešami. Vienkārši iespējojiet un instalējiet to, izmantojot:

```
dnf module list nginx
dnf -y install nginx
```

RHEL 7 versijā NGINX var instalēt no CentOS 7 Extras repozitorija (ja izmantojat CentOS), izmantojot:

```
yum -y install nginx
```

Ja dodat priekšroku jaunākajām augšupējās izstrādes versijām, konfigurējiet oficiālo NGINX repozitoriju failā `/etc/yum.repos.d/nginx.rep`

```
cat > /etc/yum.repos.d/nginx.repo << 'EOF'
[nginx-stable]
name=nginx stable repo
baseurl=https://nginx.org/packages/rhel/$releasever/$basearch/
gpgcheck=1
enabled=1
gpgkey=https://nginx.org/keys/nginx_signing.key
EOF
dnf -y install nginx
```

SLES 15

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
```

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.

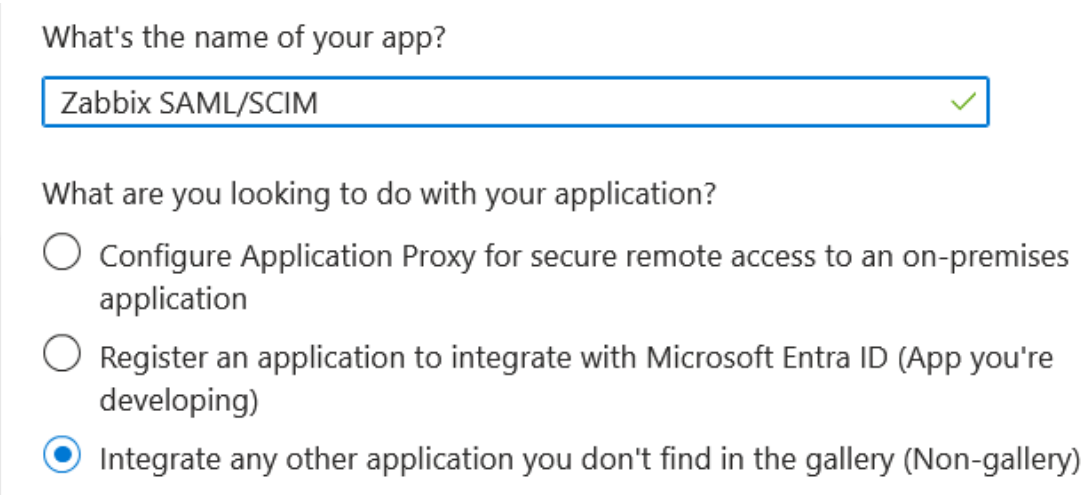
Microsoft Entra ID konfigurācija

Lietotnes 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 lietotnes nosaukumu un atlasiet opciju *Integrate any other application...*. Pēc tam noklikšķiniet uz *Create*.

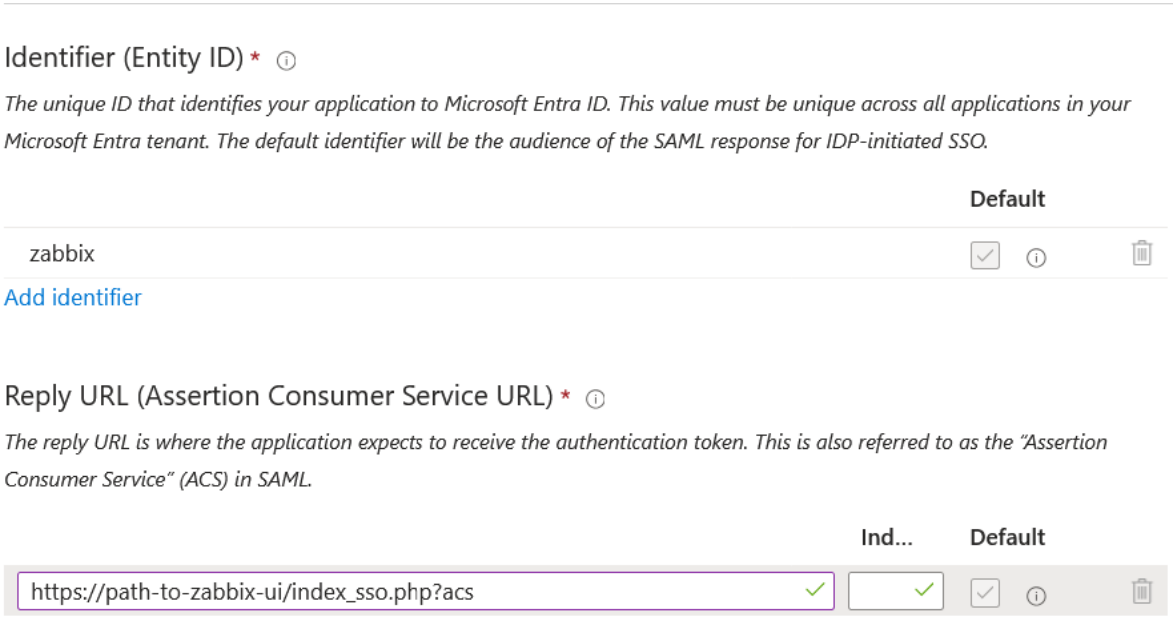


Vienotās pierakstīšanās iestatīšana

1. Lietotnes lapā atveriet *Set up single sign on* un noklikšķiniet uz *Get started*. Pēc tam atlasiet *SAML*.

2. Rediģējiet *Basic SAML Configuration*:

- Sadaļā *Identifier (Entity ID)* iestatiet unikālu nosaukumu, lai identificētu savu lietotni Microsoft Entra ID, piemēram, `zabbix`;
- Sadaļā *Reply URL (Assertion Consumer Service URL)* iestatiet Zabbix vienotās pierakstīšanās galapunktu: `https://<path-to-zabbix`



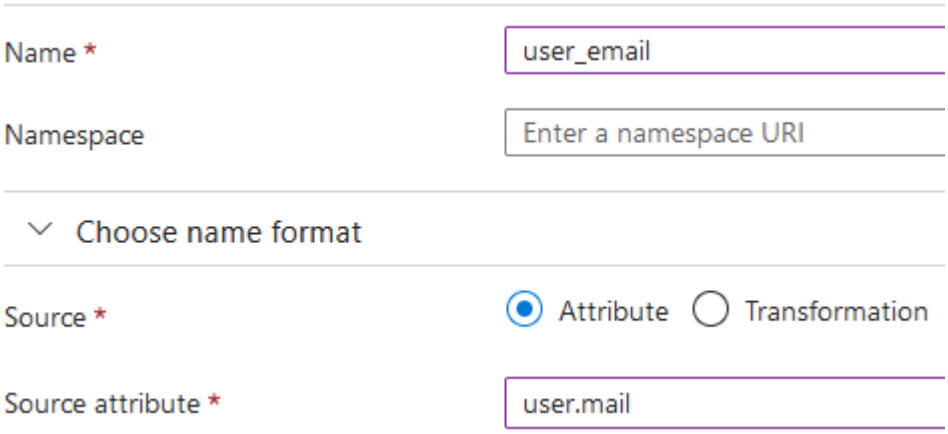
Ņemiet vērā, ka ir nepieciešams `https`. Lai to izmantotu ar Zabbix, pievienojiet tālāk norādīto rindu lietotāja saskarnes konfigurācijas failam:

```
$SSO['SETTINGS'] = ['use_proxy_headers' => true];
```

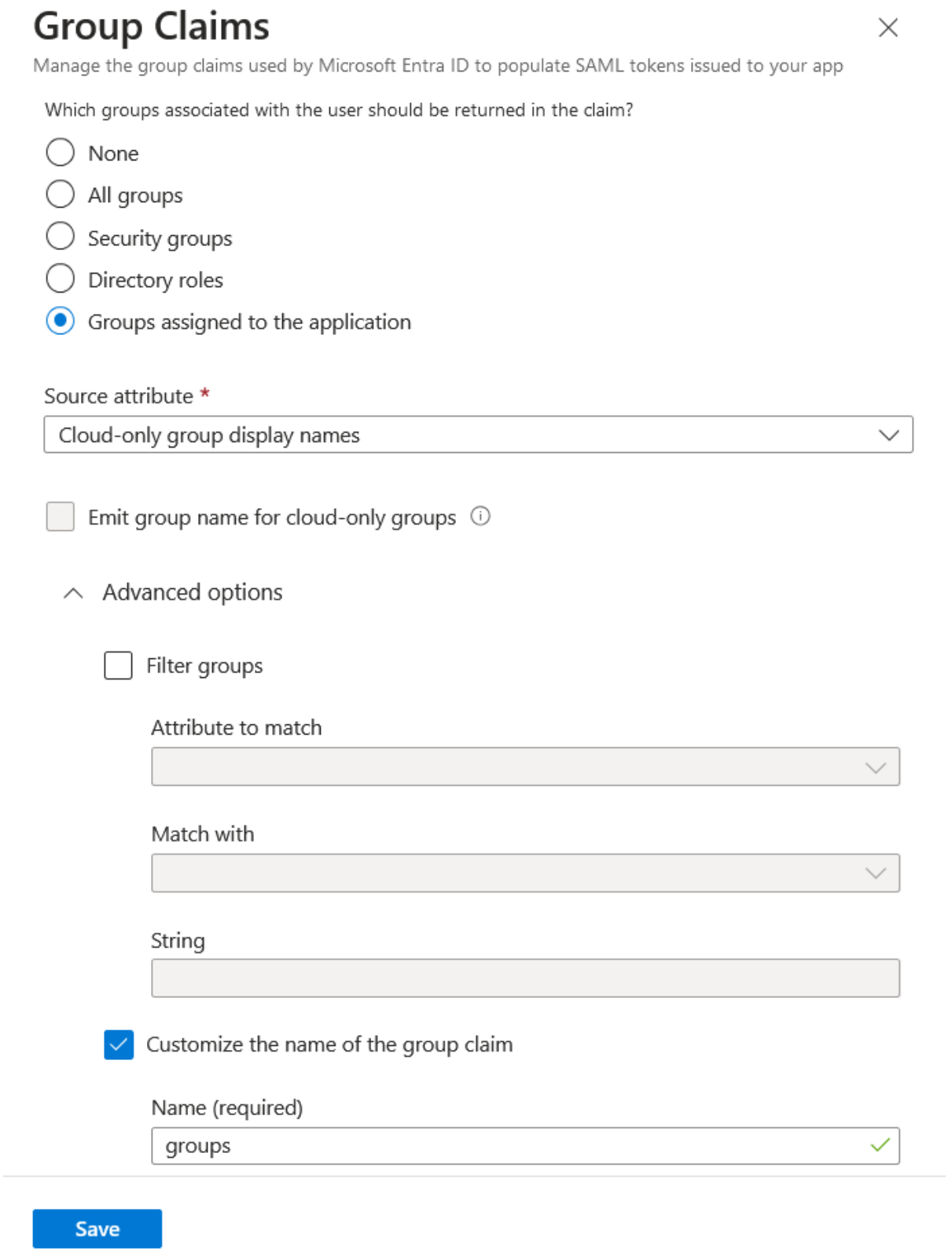
3. Rediģējiet *Attributes & Claims*. Jums jāpievieno visi atribūti, kurus vēlaties nodot Zabbix (user_name, user_lastname, user_email, user_mobile, groups).

Atribūtu nosaukumi ir patvaļīgi. Var izmantot citus atribūtu nosaukumus, tomēr tiem ir jāatbilst attiecīgajai 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:



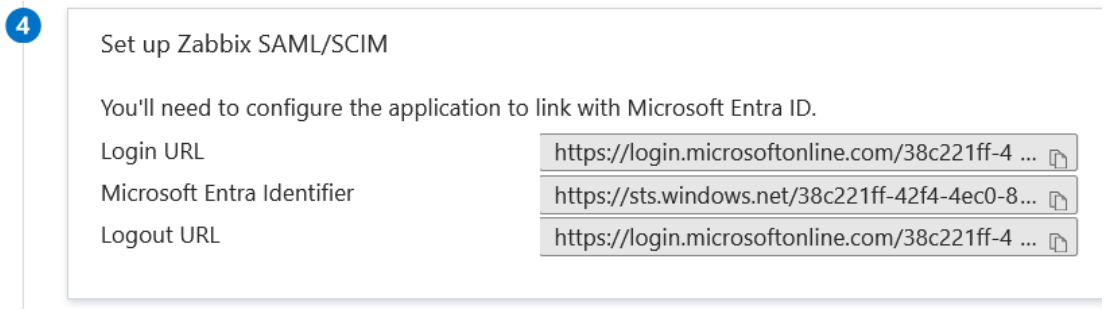
Šajā pieprasījumā ir svarīgi, lai atlasītais *Source attribute* nodotu Zabbix grupu nosaukumus, nevis grupu ID. Pretējā gadījumā JIT lietotāju nodrošināšana nedarbosies pareizi.

4. Sadaļā *SAML Certificates* lejupielādējiet Entra ID nodrošināto Base64 sertifikātu un ievietojiet to Zabbix lietotāja saskarnes instalācijas direktorijā `conf/certs` (atrašanās vieta var atšķirties atkarībā no Zabbix instalācijas operētājsistēmas un avota).

Iestatiet tam 644 atļaujas, izpildot:

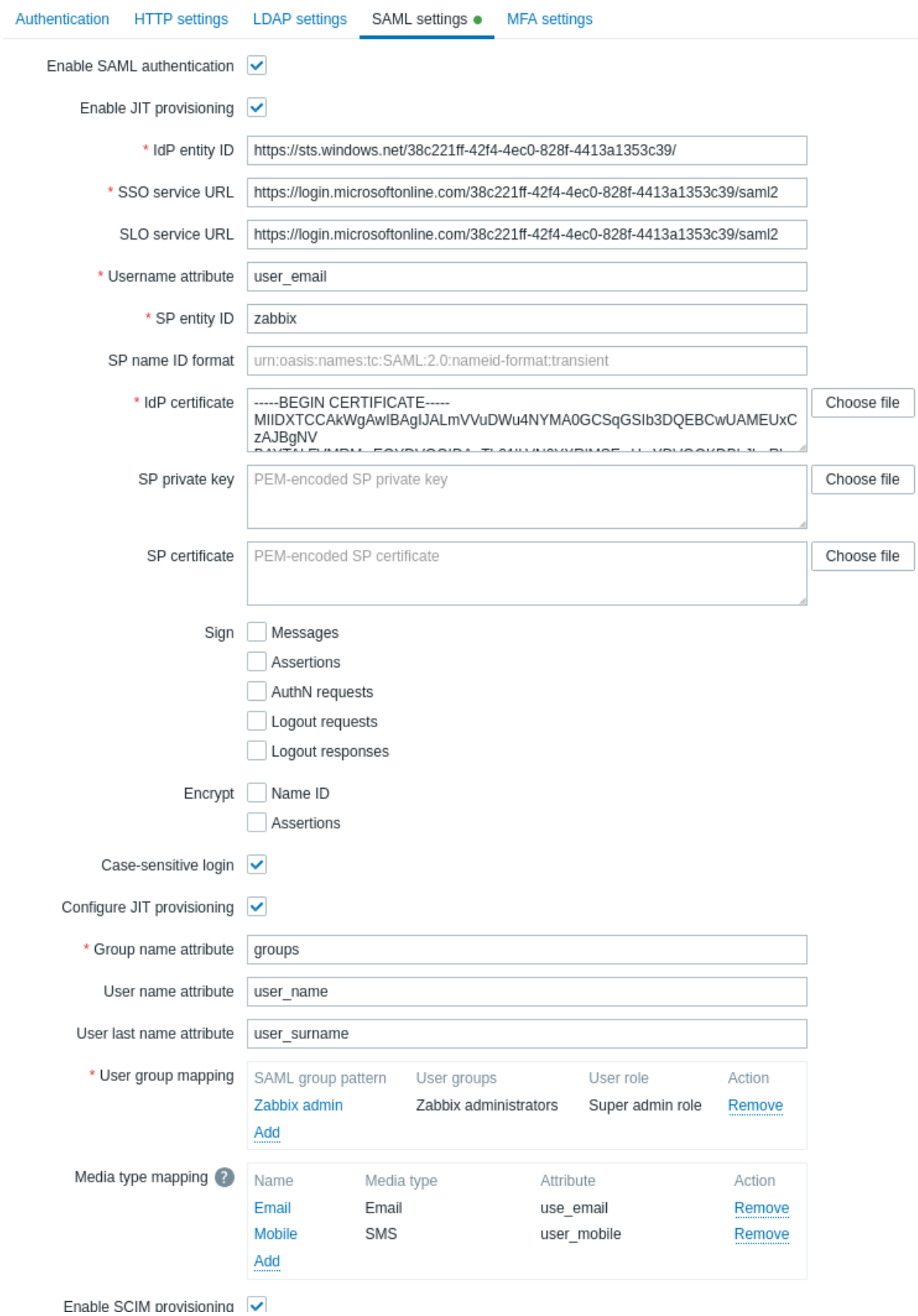
```
chmod 644 entra.cer
```

5. Izmantojiet Entra ID sadaļā *Set up <your app name>* norādītās vērtības, lai konfigurētu Zabbix SAML autentifikāciju (skatiet nākamo sadaļu):



Zabbix konfigurācija

1. Programmā Zabbix atveriet SAML iestatījumus un aizpildiet konfigurācijas opcijas, pamatojoties uz Entra ID konfigurāciju:



Zabbix lauks	Iestatīšanas lauks programmā Entra ID	Piemēra vērtība
IdP entītijas ID	Microsoft Entra identifikators	
SSO pakalpojuma URL	Pieteikšanās URL	
SLO pakalpojuma URL	Izrakstīšanās URL	
SP entītijas ID	Identifikators (entītijas ID)	
Lietotājvārda atribūts	Pielāgots atribūts (apgalvojums)	`user_email`
Grupas nosaukuma atribūts	Pielāgots atribūts (apgalvojums)	`groups`
Lietotāja vārda atribūts	Pielāgots atribūts (apgalvojums)	`user_name`

Zabbix lauks	Iestatīšanas lauks programmā Entra ID	Piemēra vērtība
Lietotāja uzvārda atribūts	Pielāgots atribūts (apgalvojums)	`user_lastname`

2. Pievienojiet Entra ID nodrošināto Base64 sertifikātu.

Ja lietotāja saskarnes konfigurācijas failā (`zabbix.conf.php`) ir iestatīts `$SSO['CERT_STORAGE'] = 'database'`, SAML konfigurēšanas laikā varat ielīmēt sertifikāta tekstu vai augšupielādēt sertifikāta failu lietotāja saskarnē — failiem failu sistēmā nav jābūt pieejamiem.

Ja konfigurācijas failā ir iestatīts `$SSO['CERT_STORAGE'] = 'file'`, sertifikātam jābūt pieejamam failu sistēmā (pēc noklusējuma mapē `ui/conf/certs` vai konfigurācijas failā norādītajā ceļā), un lietotāja saskarne sertifikātus datubāzē nesaglabās. Ņemiet vērā: ja `$SSO['CERT_STORAGE']` nav iestatīts vai ir komentēts, tiek pieņemta glabāšana failā un sertifikāti tiek nolasīti no `ui/conf/certs`.

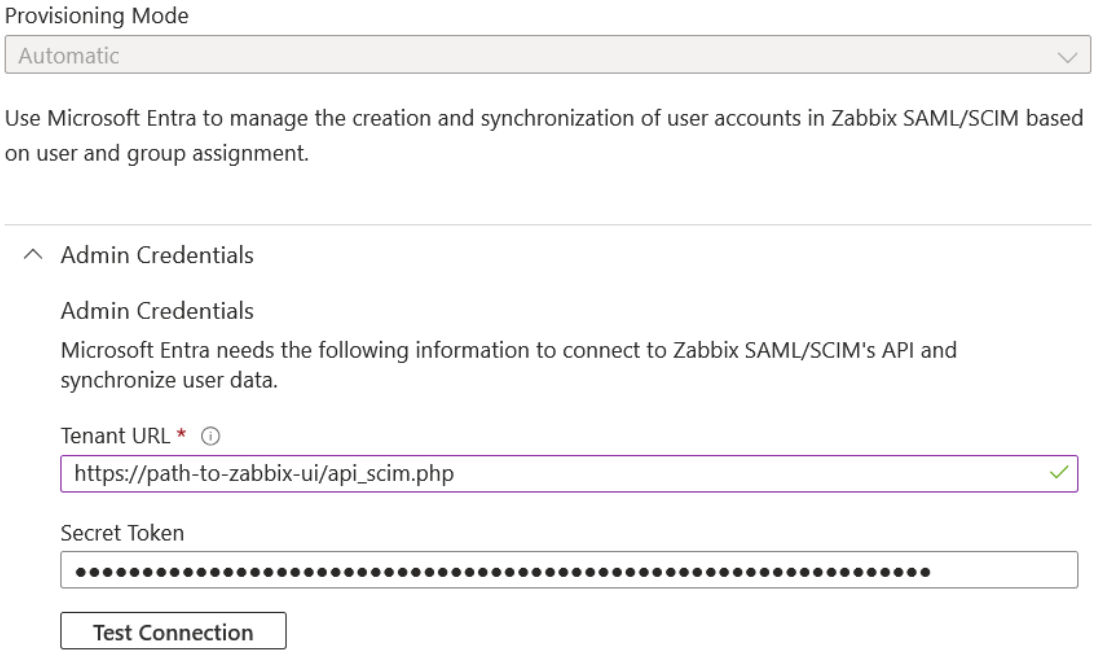
Ir arī jākonfigurē lietotāju grupu kartēšana. Multivides kartēšana nav obligāta.

3. Noklikšķiniet uz pogas *Atjaunināt*, lai saglabātu šos iestatījumus.

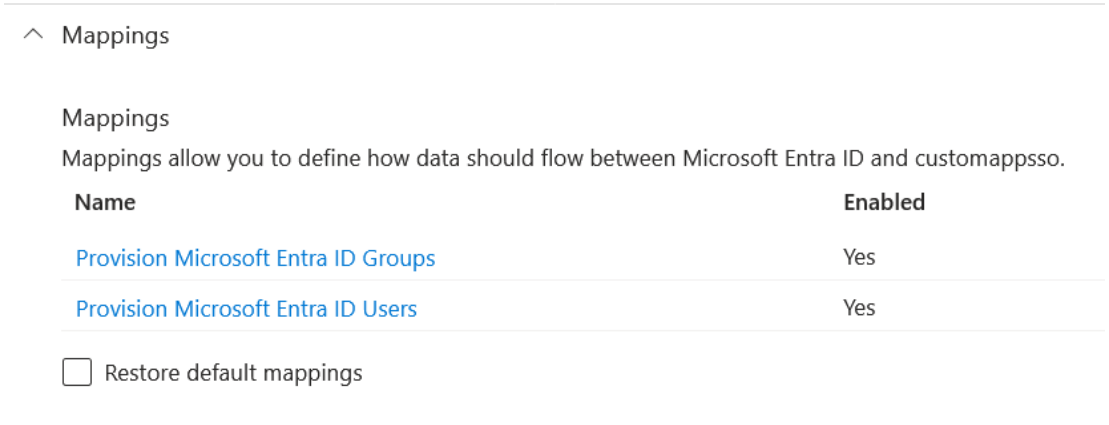
SCIM lietotāju nodrošināšana

1. Savā Entra ID lietotnes lapā galvenajā izvēlnē atveriet lapu *Provisioning*. Noklikšķiniet uz *Get started* un pēc tam atlasiet automātiskās nodrošināšanas režīmu:

- Laukā *Tenant URL* norādiet šādu vērtību: `https://<path-to-zabbix-ui>/api_scim.php`
- Laukā *Secret token* ievadiet Zabbix API tokenu ar Super admin tiesībām.
- Noklikšķiniet uz *Test connection*, lai pārbaudītu, vai savienojums ir izveidots.

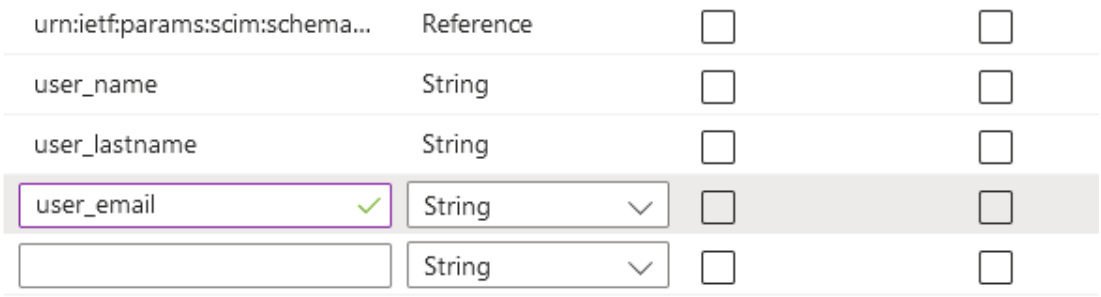


2. Tagad varat pievienot visus atribūtus, kas tiks nodoti ar SCIM uz Zabbix. Lai to izdarītu, noklikšķiniet uz *Mappings* un pēc tam uz *Provision Microsoft Entra ID Users*.



Atribūtu kartējuma saraksta apakšā iespējojiet *Show advanced options* un pēc tam noklikšķiniet uz *Edit attribute list for custom-appsso*.

Atribūtu saraksta apakšā pievienojiet savus atribūtus ar tipu 'String':



Saglabājiet sarakstu.

3. Tagad varat pievienot kartējumus pievienotajiem atribūtiem. Atribūtu kartējuma saraksta apakšā noklikšķiniet uz *Add New Mapping* un izveidojiet kartējumus, kā parādīts tālāk:



Kad visi kartējumi ir pievienoti, saglabājiet kartējumu sarakstu.



4. Kā priekšnoteikums lietotāju nodrošināšanai Zabbix, jums Entra ID ir jākonfigurē lietotāji un grupas.

Lai to izdarītu, dodieties uz *Microsoft Entra admin center* un pēc tam pievienojiet lietotājus/grupas attiecīgajās lapās *Users* un *Groups*.

5. Kad lietotāji un grupas ir izveidoti Entra ID, varat doties uz savas lietotnes izvēlni *Users and groups* un pievienot tos lietotnei.

6. Dodieties uz savas lietotnes izvēlni *Provisioning* un noklikšķiniet uz *Start provisioning*, lai lietotāji tiktu nodrošināti Zabbix.

Ņemiet vērā, ka Entra ID lietotāju PATCH pieprasījums neatbalsta izmaiņas vidē.

Autentifikācijas pieprasījumu parakstīšana

Entra ID var konfigurēt, lai validētu parakstīto autentifikācijas pieprasījumu parakstu.

Lai tas darbotos, izveidojiet publiskās un privātās atslēgas:

```
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 saskarnes konfigurāciju, pievienojot:

```
$SSO['SP_KEY'] = 'conf/certs/request-sign.key';
$SSO['SP_CERT'] = 'conf/certs/request-sign.crt';
```

Problēmu novēršana

Autentifikācijas problēmas var rasties pārlūkprogrammā Microsoft Edge, ja lietotājs, mēģinot pieteikties Zabbix, izmantojot SAML, jau ir pieteicies ar Microsoft Edge profilu. Par šādu problēmu var liecināt tas, ka lietotājs var pieteikties Zabbix, izmantojot Microsoft Edge privātajā režīmā.

Lai šādā gadījumā izvairītos no autentifikācijas problēmām, Zabbix lietotāja saskarnes konfigurācijas failā var būt nepieciešams iestatīt `requestedAuthnContext` vērtību uz `false`.

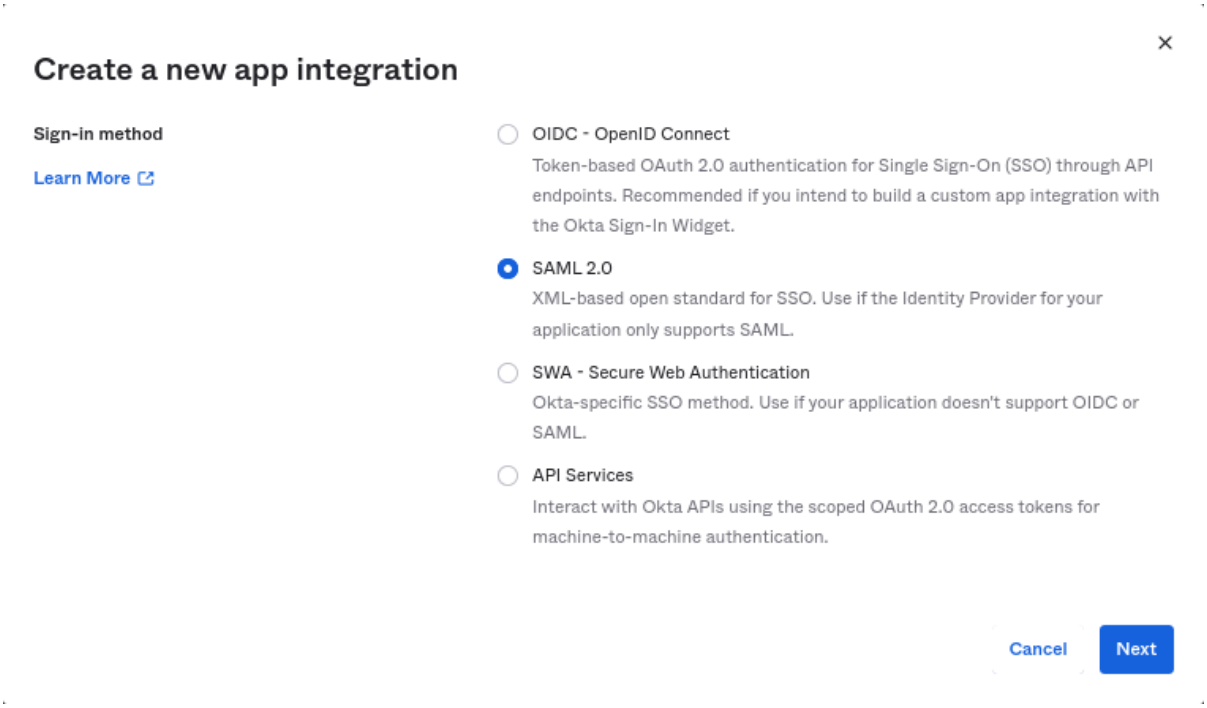
```
$SSO['SETTINGS'] = [
    'security' => [
        'requestedAuthnContext' => false
    ]
];
```

SAML iestatīšana ar Okta

Šī sadaļa sniedz norādījumus par Okta konfigurēšanu, lai iespējotu SAML 2.0 autentifikāciju un lietotāju nodrošināšanu Zabbix.

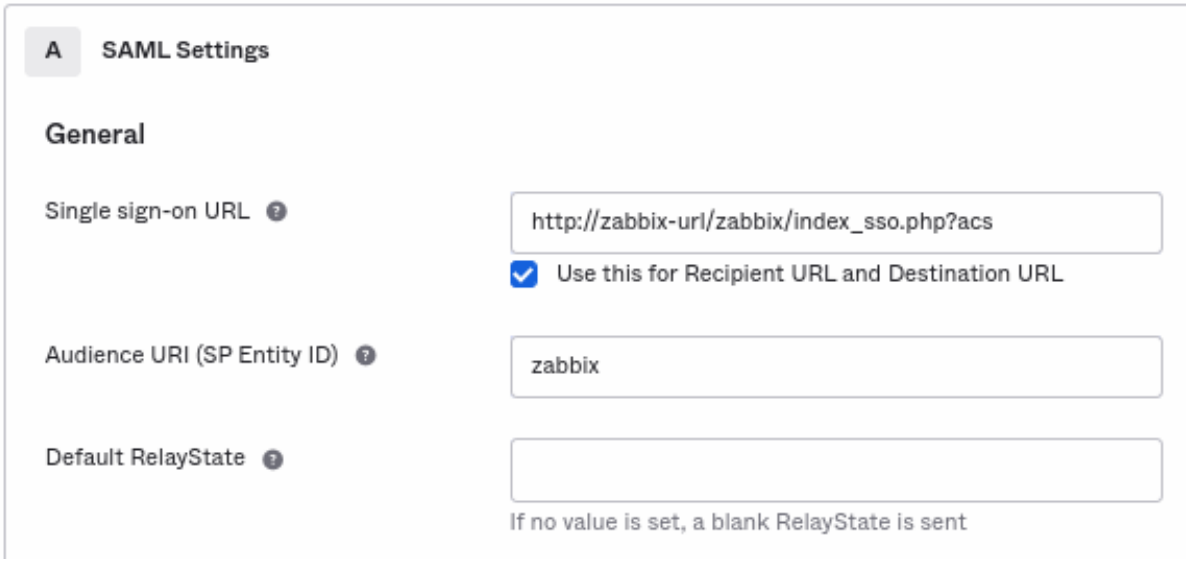
Okta konfigurācija

1. Dodieties uz https://developer.okta.com/signup/ un reģistrējieties/piesakieties savā kontā.
2. Okta lietotāja saskarnē dodieties uz *Applications* > *Applications*.
3. Noklikšķiniet *Create App Integration*.

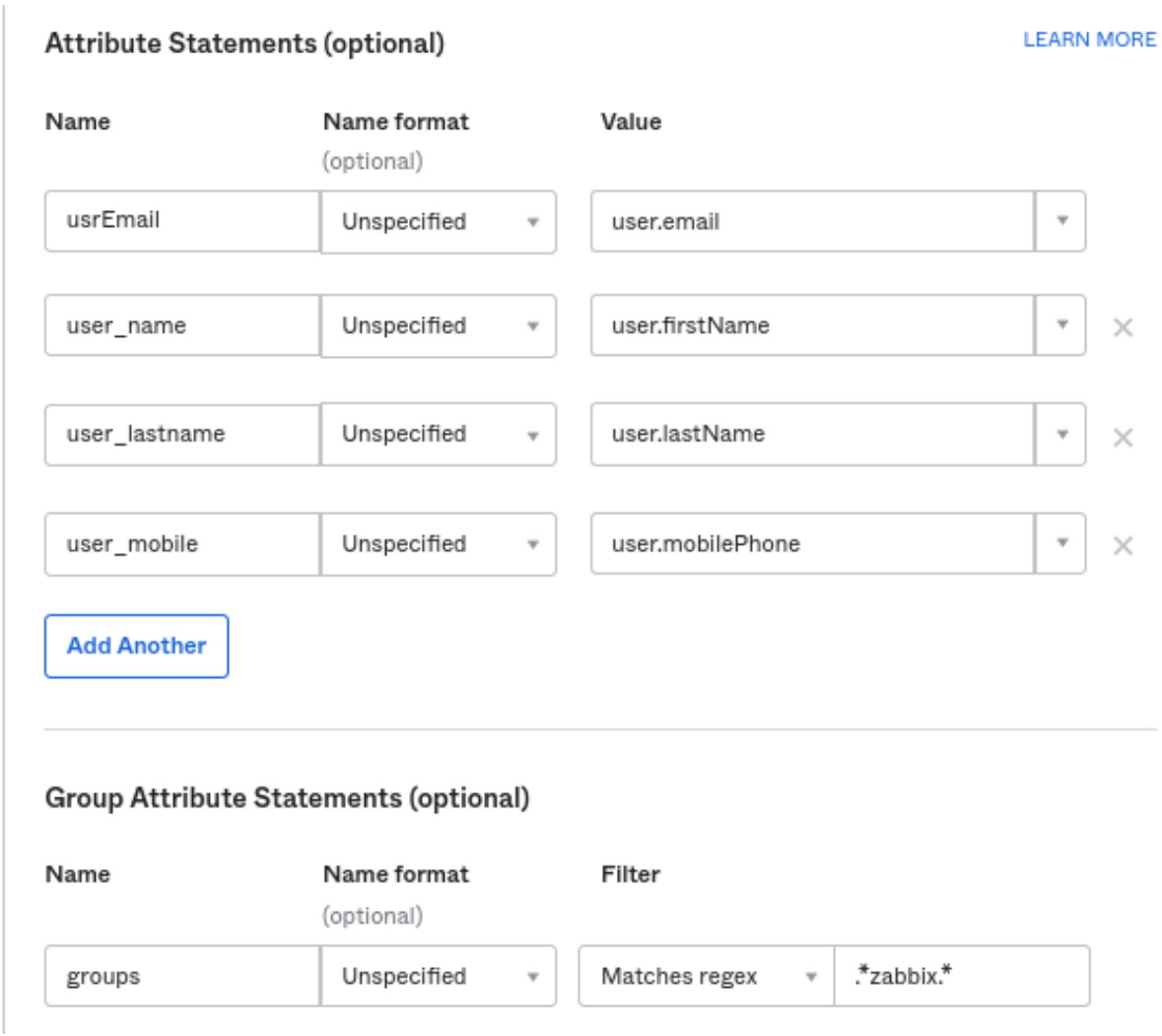


Kā pierakstīšanās metodi atlasiet "SAML 2.0" un noklikšķiniet *Next*.

4. Vispārīgajos iestatījumos ievadiet lietotnes nosaukumu un noklikšķiniet *Next*.
5. SAML konfigurācijā ievadiet tālāk norādītās vērtības, pēc tam noklikšķiniet *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 nogriezts parametrs `acs`. Jābūt atzīmētai arī izvēles rūtiņai *Use this for Recipient URL and Destination URL*.
  - *Audience URI (SP Entity ID)*: `zabbix`
 Ņemiet vērā, ka šī vērtība tiks izmantota SAML apgalvojumā kā unikāls pakalpojuma sniedzēja identifikators (ja tā nesakritīs, darbība tiks noraidīta). Šajā laukā ir iespējams norādīt URL vai jebkuru teksta virkni.
  - *Default RelayState*:
 Atstājiet šo lauku tukšu; ja nepieciešama pielāgota pāradresācija, to var pievienot Zabbix sadaļā *Users* > *Users*.
  - Aizpildiet citus laukus atbilstoši savām vēlmēm.
- Sadaļā **Attribute Statements/Group Attribute Statements** pievienojiet:



Šie atribūtu paziņojumi tiek ievietoti SAML apgalvojumos, kas tiek kopīgoti ar Zabbix.

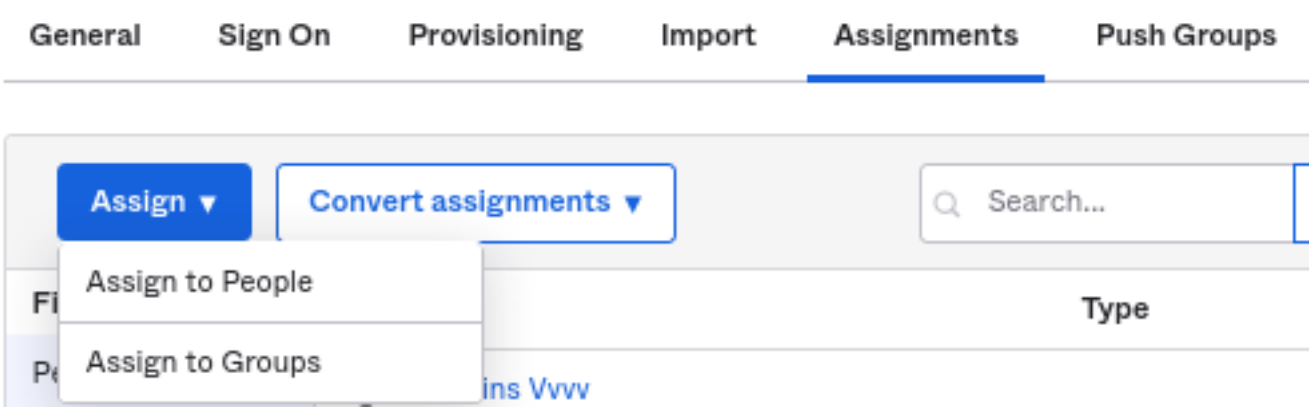
Šeit izmantotie atribūtu nosaukumi ir patvaļīgi piemēri. Varat izmantot citus atribūtu nosaukumus, tomēr tiem ir jāsakrīt ar attiecīgā lauka vērtību Zabbix SAML iestatījumos.

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 Okta. Sertifikāta augšupielādes forma parādās, kad *Assertion Encryption* ir iestatīts uz "Encrypted" (noklikšķiniet *Show Advanced Settings*, lai atrastu šo parametru).

6. Nākamajā cilnē atlasiet "I'm a software vendor. I'd like to integrate my app with Okta" un noklikšķiniet *Finish*.

7. Dodieties uz jaunizveidotās lietotnes cilni "Assignments" un noklikšķiniet pogu *Assign*, pēc tam nolaižamajā izvēlnē atlasiet *Assign to People*.



8. Uznirstošajā logā piešķiriet lietotni cilvēkiem, kuri izmantos SAML 2.0 autentifikācijai ar Zabbix, pēc tam noklikšķiniet *Save and go back*.

9. Lejupielādējiet IdP sertifikātu. Iestatiet tam 644 atļaujas, izpildot:

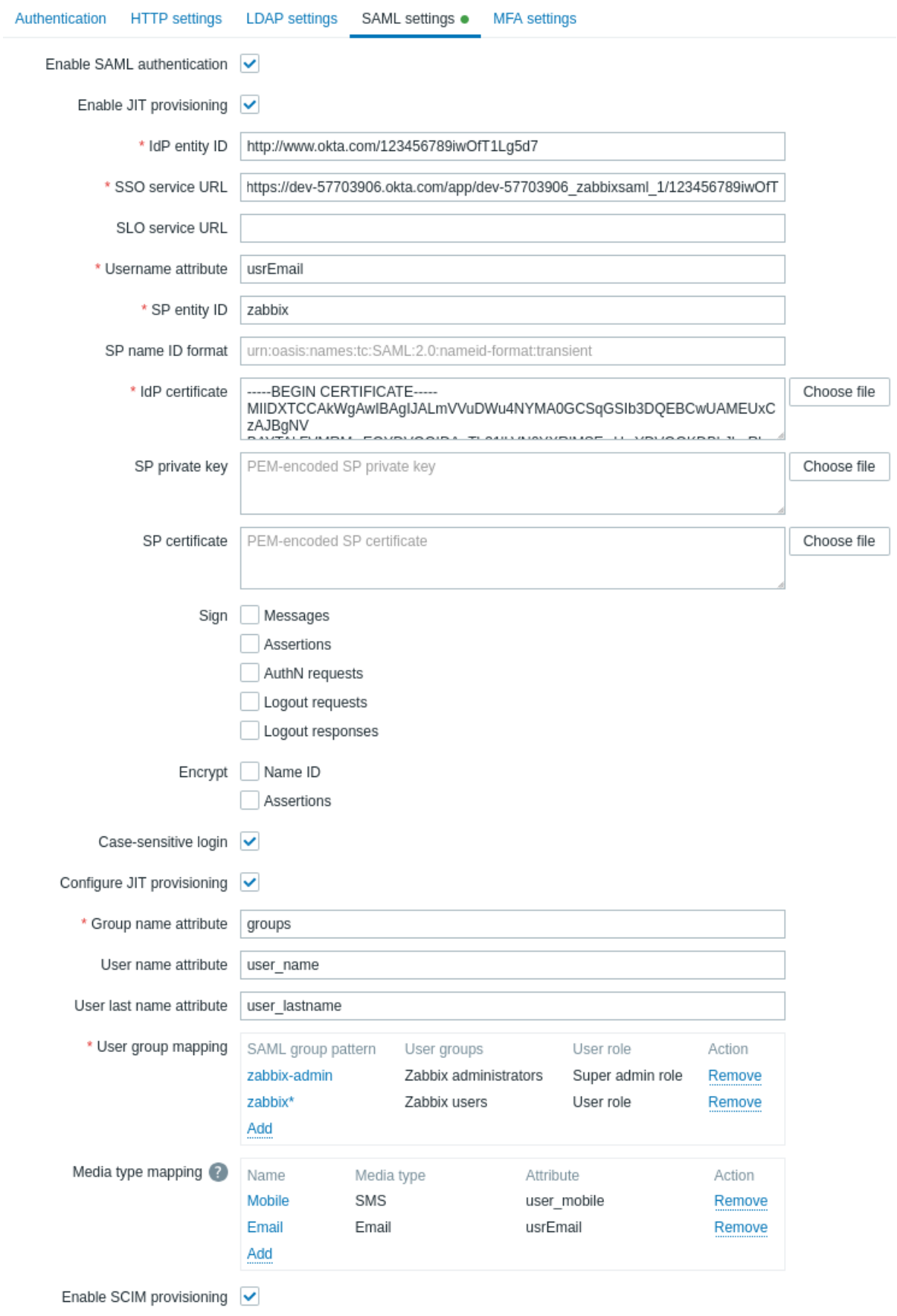
```
chmod 644 idp.crt
```

10. Dodieties uz cilni "Sign On" un noklikšķiniet pogu *View Setup Instructions*.

Iestatīšanas **instrukcijas** tiks atvērtas jaunā cilnē; konfigurējot Zabbix, atstājiet šo cilni atvērtu.

Zabbix konfigurācija

1. Programmā Zabbix atveriet SAML iestatījumus un aizpildiet konfigurācijas opcijas, pamatojoties uz Okta iestatīšanas norādījumiem:



Zabbix lauks	Iestatīšanas lauks programmā Okta	Piemēra vērtība
IdP entītijas ID	Identity Provider Issuer	
SSO pakalpojuma URL	Identity Provider Single Sign-On URL	
Lietotājvārda atribūts	Attribute name	`usrEmail`
SP entītijas ID	Audience URI	`zabbix`
Grupas nosaukuma atribūts	Attribute name	`groups`

Zabbix lauks	Iestatīšanas lauks programmā Okta	Piemēra vērtība
Lietotāja vārda atribūts	Attribute name	`user_name`
Lietotāja uzvārda atribūts	Attribute name	`user_lastname`

Ir arī jākonfigurē lietotāju grupu un datu nesēju kartēšana.

2. Pievienojiet Base64 sertifikātu, kas norādīts Okta SAML iestatīšanas norādījumos.

Ja lietotāja saskarnes konfigurācijas failā (`zabbix.conf.php`) ir iestatīts `$SSO['CERT_STORAGE'] = 'database'`, SAML konfigurēšanas laikā varat ielīmēt sertifikāta tekstu vai augšupielādēt sertifikāta failu lietotāja saskarnē — failu sistēmā faili nav nepieciešami.

Ja konfigurācijas failā ir iestatīts `$SSO['CERT_STORAGE'] = 'file'`, sertifikātam jābūt pieejamam failu sistēmā (pēc noklusējuma mapē `ui/conf/certs` vai konfigurācijas failā norādītajā ceļā), un lietotāja saskarne nesaglabās sertifikātus datubāzē. Ņemiet vērā: ja `$SSO['CERT_STORAGE']` nav iestatīts vai ir komentēts, tiek pieņemta glabāšana failā un sertifikāti tiek nolasīti no `ui/conf/certs`.

3. Ja programmā Okta *Assertion Encryption* ir iestatīts uz "Encrypted", arī programmā Zabbix parametram *Encrypt* ir jāatzīmē izvēles rūtiņa "Assertions".

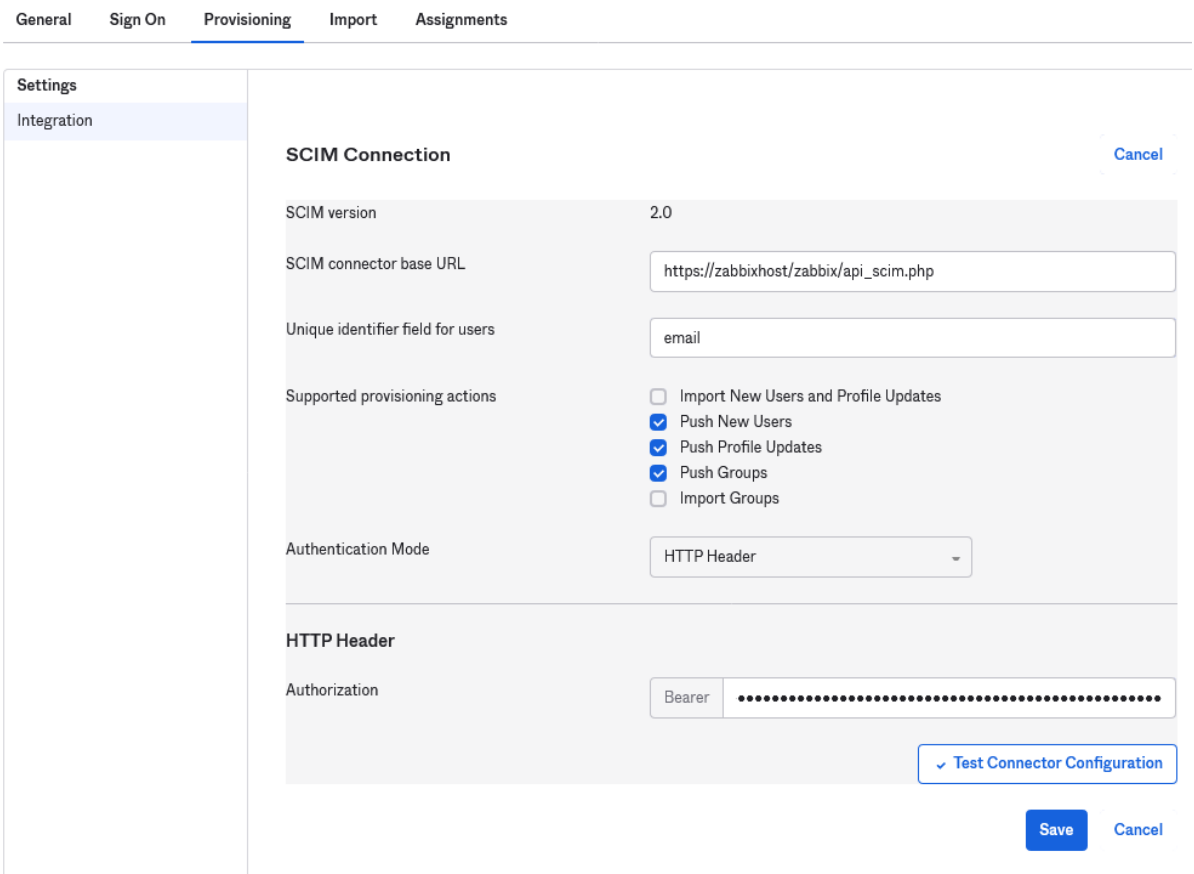
4. Noklikšķiniet uz pogas *Update*, lai saglabātu šos iestatījumus.

SCIM nodrošināšana

1. Lai ieslēgtu SCIM nodrošināšanu, Okta lietojumprogrammai atveriet *General* > *App Settings*.

Atzīmējiet izvēles rūtiņu *Enable SCIM provisioning*. Tā rezultātā tiks parādīta jauna cilne *Provisioning*.

2. Lai iestatītu SCIM savienojumu, atveriet cilni *Provisioning*: - Laukā *SCIM connector base URL* norādiet ceļu uz Zabbix lietotāja saskarni 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.

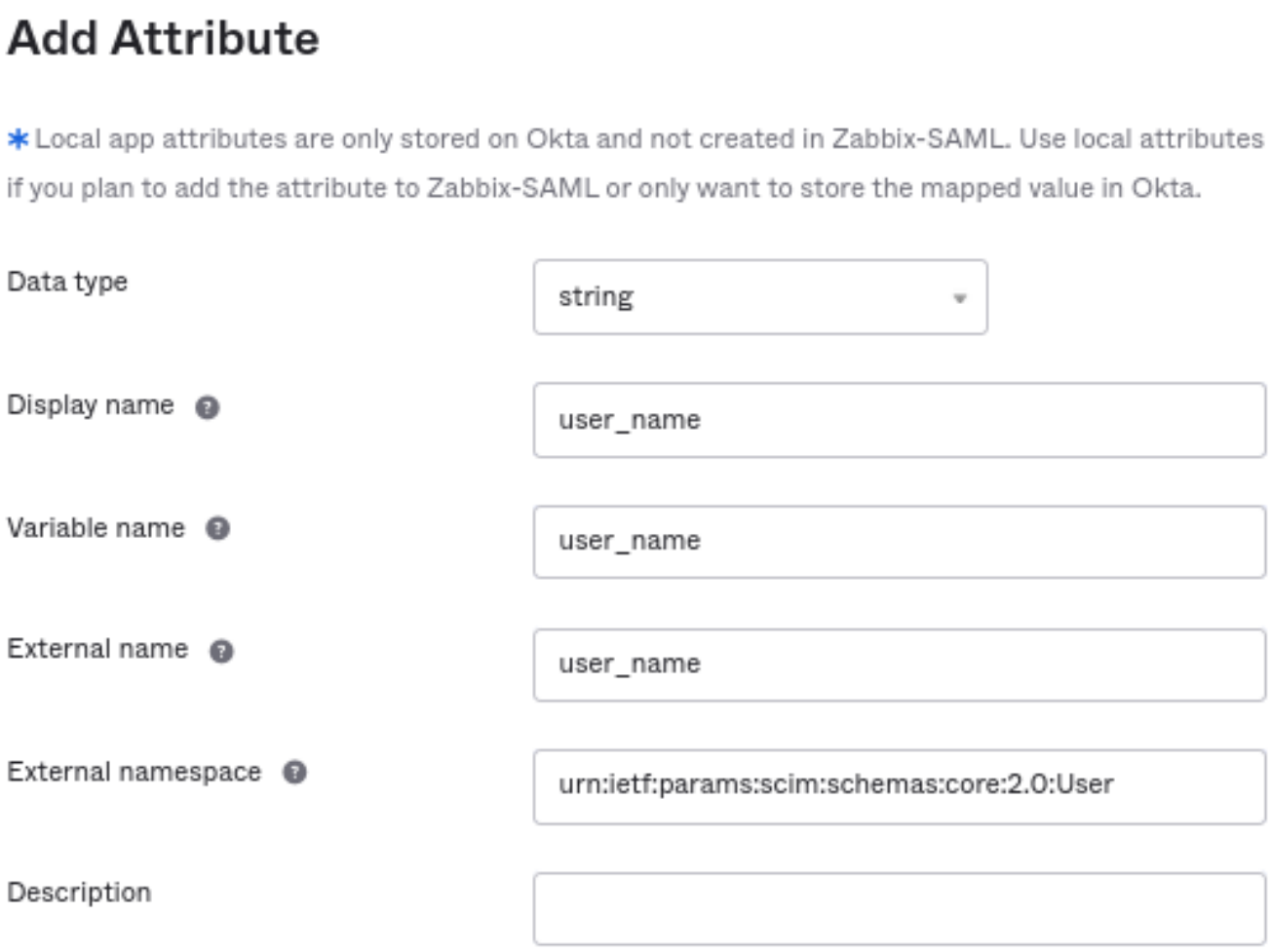
3. Noklikšķiniet uz *Test Connector Configuration*, lai pārbaudītu savienojumu. Ja viss ir pareizi, tiks parādīts ziņojums par veiksmīgu izpildi.

4. Sadaļā *Provisioning* > *To App* noteikti atzīmējiet šādas izvēles rūtiņas: - Create Users - Update User Attributes - Deactivate Users

Tādējādi tiks nodrošināts, ka šie pieprasījumu veidi tiks nosūtīti uz Zabbix.

5. Pārliecinieties, ka visi SAML definētie atribūti ir definēti arī SCIM. Lietojumprogrammas profila redaktoram var piekļūt sadaļā *Provisioning* > *To App*, noklikšķinot uz *Go to Profile Editor*.

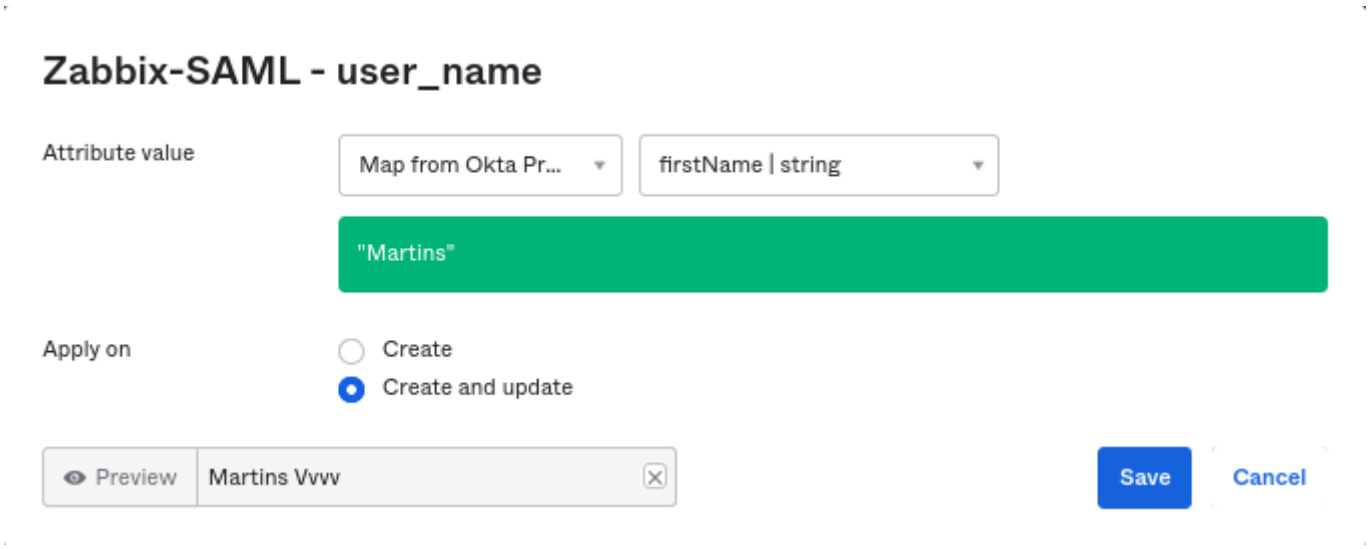
Noklikšķiniet uz *Add Attribute*. Aizpildiet laukus *Display name*, *Variable name* un *External name* ar SAML atribūta nosaukumu, piemēram, `user_name`.



External namespace jābūt tādam pašam kā lietotāja shēmai: `urn:ietf:params:scim:schemas:core:2.0:User`

6. Atveriet savas lietojumprogrammas sadaļu *Provisioning* > *To App* > *Attribute Mappings*. Apakšdaļā noklikšķiniet uz *Show Unmapped Attributes*. Tiks parādīti tikko pievienotie atribūti.

7. Kartējiet katru pievienoto atribūtu.



8. Cilnē "Assignments" pievienojiet lietotājus. Vispirms lietotāji jāpievieno sadaļā *Directory* > *People*. Visi šie piešķīrumi tiks nosūtīti uz Zabbix kā pieprasījumi.

9. Cilnē "Push Groups" pievienojiet grupas. Lietotāju grupu kartēšanas paraugam Zabbix SAML iestatījumos jāatbilst šeit norādītajai grupai. Ja atbilstības nav, lietotāju nevar izveidot Zabbix.

Informācija par grupu dalībniekiem tiek nosūtīta ikreiz, kad tiek veiktas izmaiņas.

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 kontā OneLogin. Testēšanas nolūkos varat izveidot bezmaksas izstrādātāja kontu OneLogin.

2. OneLogin lietotāja saskarnē dodieties uz *Applications* > *Applications*.

3. Noklikšķiniet uz "Add App" un meklējiet atbilstošo lietotni. Šajā lapā sniegtās vadlīnijas ir balstītas uz *SCIM Provisioner with SAML (SCIM v2 Enterprise, full SAML)* lietotnes piemēru.

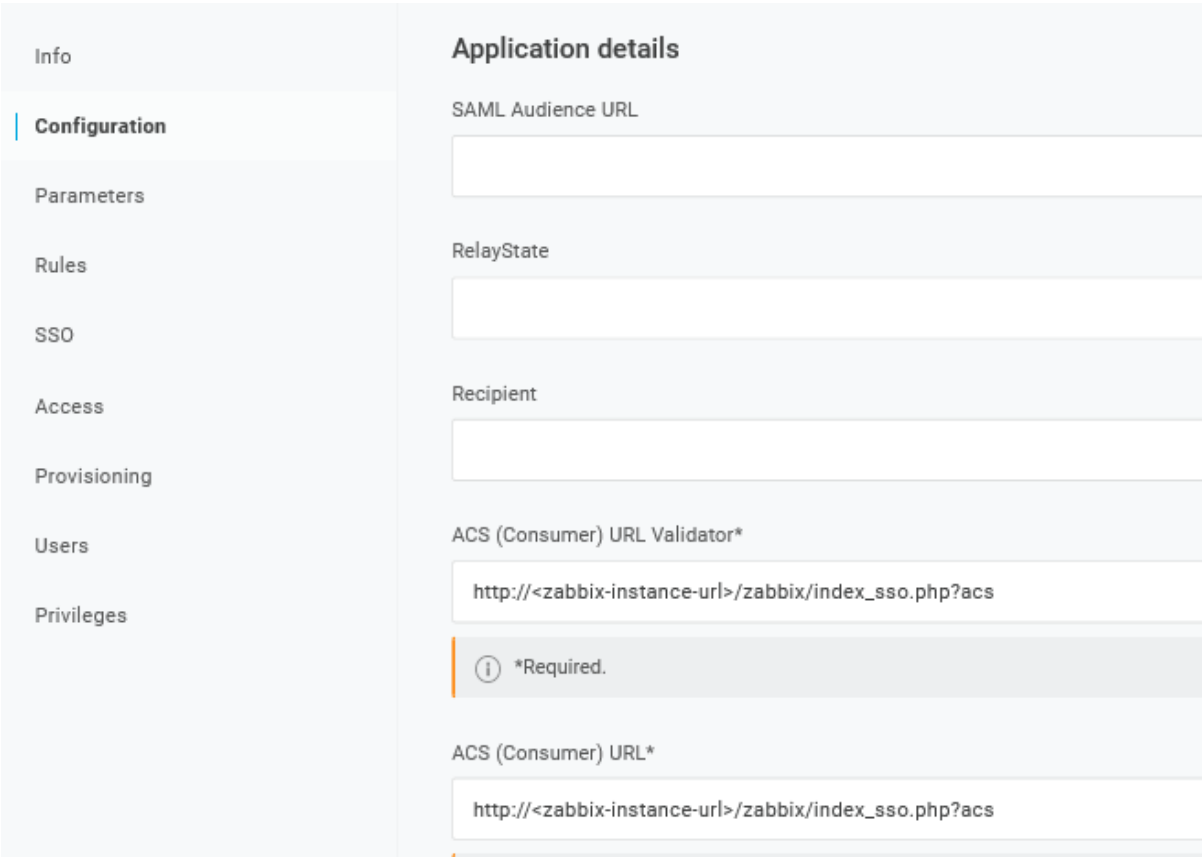
4. Vispirms, iespējams, vēlēsieties pielāgot savas lietotnes attēlojamo nosaukumu. Varat arī pievienot ikonu un lietotnes informāciju. Pēc tam noklikšķiniet uz *Save*.

SSO/SCIM nodrošināšanas iestatīšana

1. Sadaļā *Configuration* > *Application details* iestatiet Zabbix vienotās pierakstīšanās galapunktu `http://<zabbix-instance-url>/zabbi` kā šo lauku vērtību:

- *ACS (Consumer) URL Validator*
- *ACS (Consumer) URL*

Ņemiet vērā, ka tiek izmantots "http", nevis "https", lai pieprasījumā netiktu izņemts parametrs `acs`.



Var izmantot arī "https". Lai to izmantotu ar Zabbix, pievienojiet tālāk norādīto rindu lietotāja saskarnes konfigurācijas failam (`zabbix.conf.php`):

```
$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`:

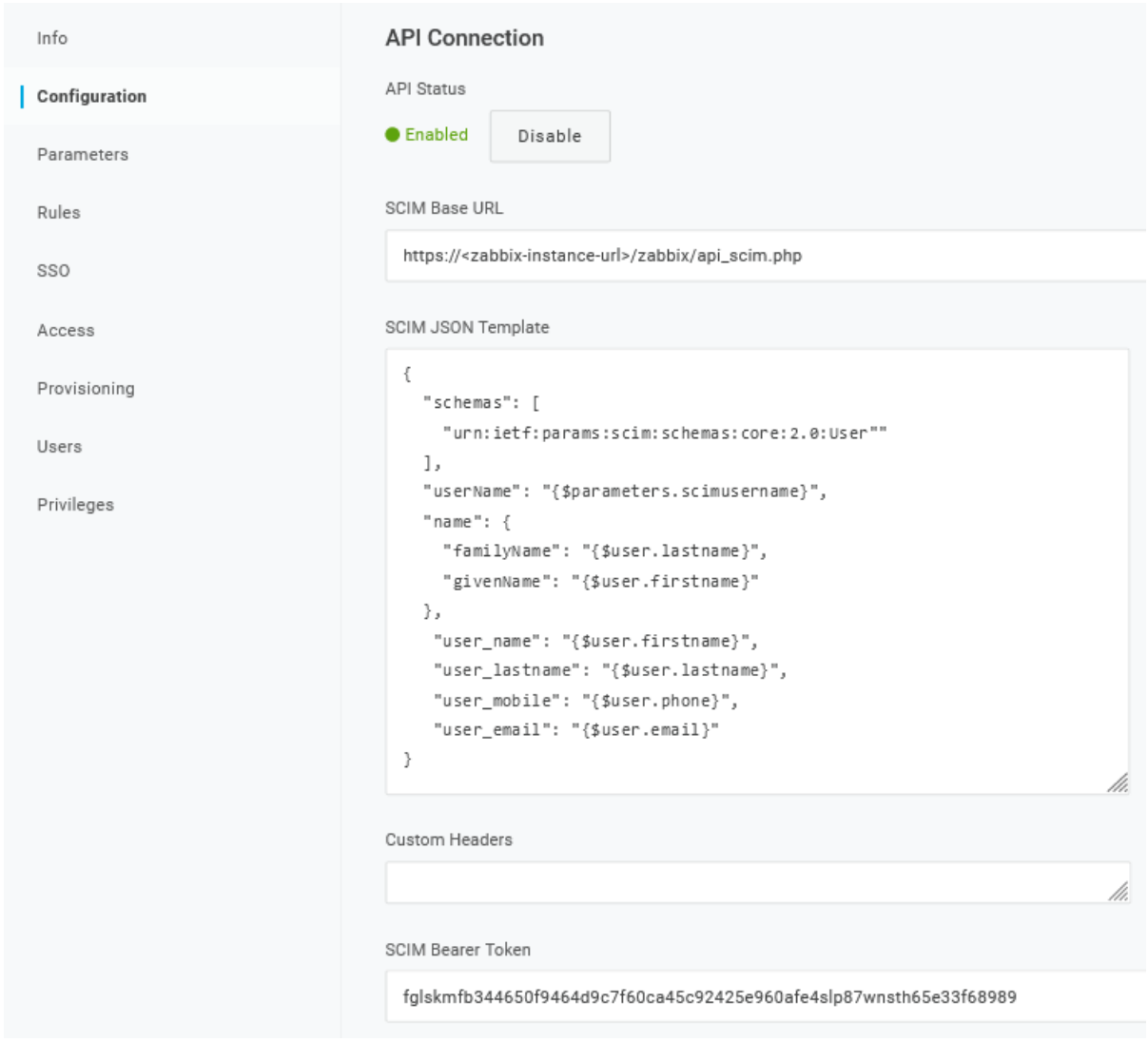
```
{
  "schemas": [
    "urn:ietf:params:scim:schemas:core:2.0:User"
  ],
  "userName": "{$parameters.scimusername}",
  "name": {
    "familyName": "{$user.lastname}",
    "givenName": "{$user.firstname}"
  },
   "user_name": "{$user.firstname}",
   "user_lastname": "{$user.lastname}",
   "user_mobile": "{$user.phone}",
   "user_email": "{$user.email}"
}
```

Atribūtu nosaukumi ir patvaļīgi. Var izmantot citus atribūtu nosaukumus, tomēr tiem ir jāatbilst attiecīgajai lauka vērtībai Zabbix SAML iestatījumos.

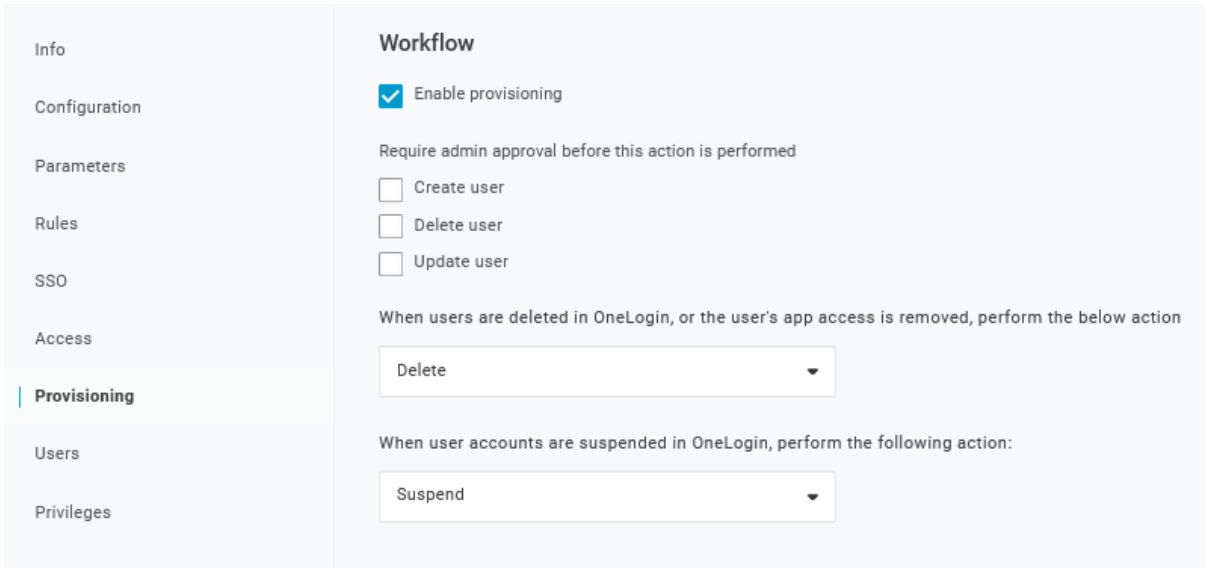
Ņemiet vērā, ka, lai lietotāju nodrošināšana darbotos, OneLogin atbildē ir jāsaņem atribūts 'name' ar 'givenName' un 'familyName', pat ja pakalpojumu sniedzējs to nepieprasīja. Tāpēc tas ir jānorāda shēmā lietojumprogrammas konfigurācijas sadaļā.

- *SCIM Bearer Token*: ievadiet Zabbix API pilnvaru ar Super admin atļaujām.

Noklikšķiniet uz *Enable*, lai aktivizētu savienojumu.

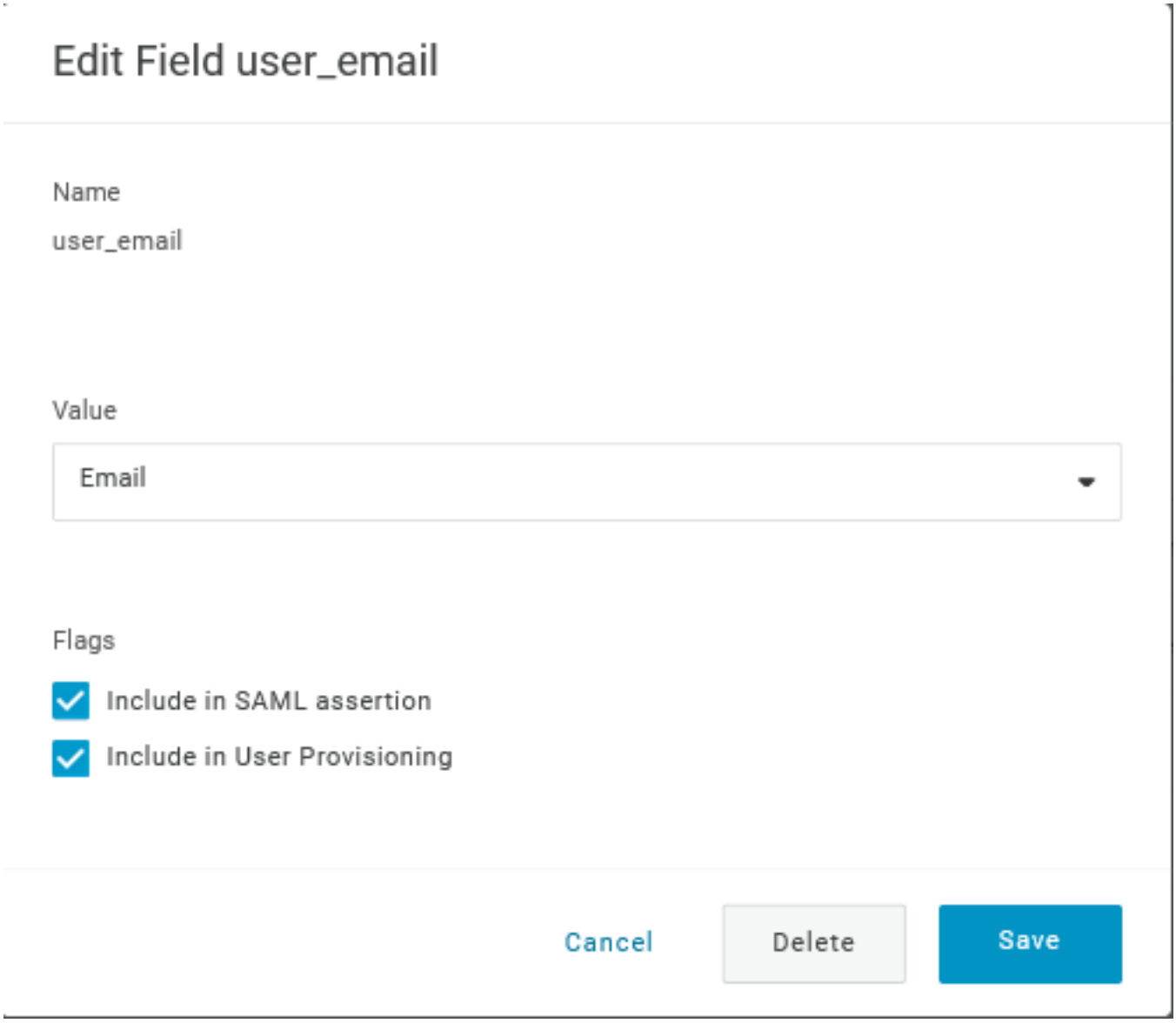


3. Lapā *Provisioning* iespējojiet opciju Provisioning:



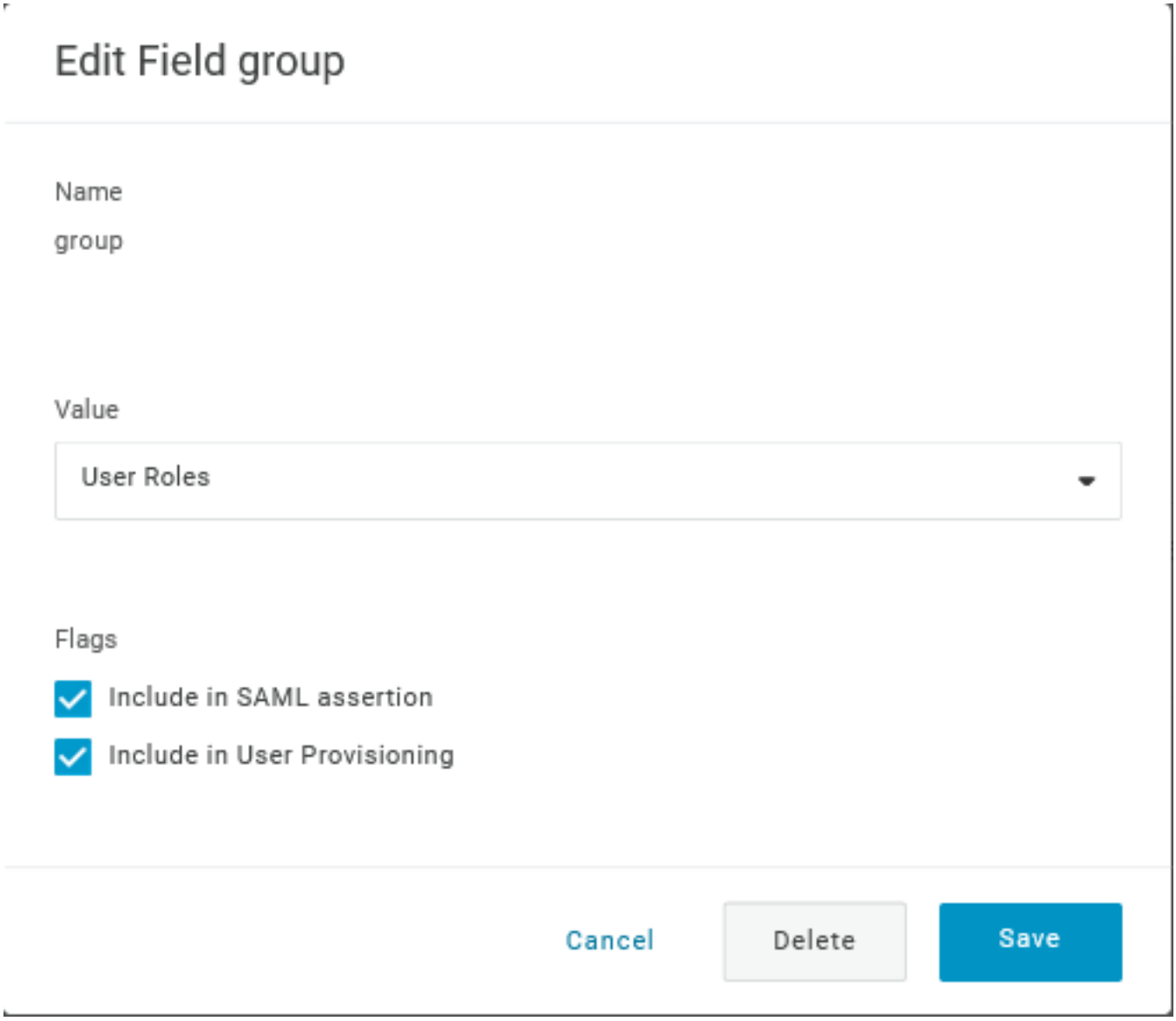
4. Lapā *Parameters* ir ietverts noklusējuma parametru saraksts:

- 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`:

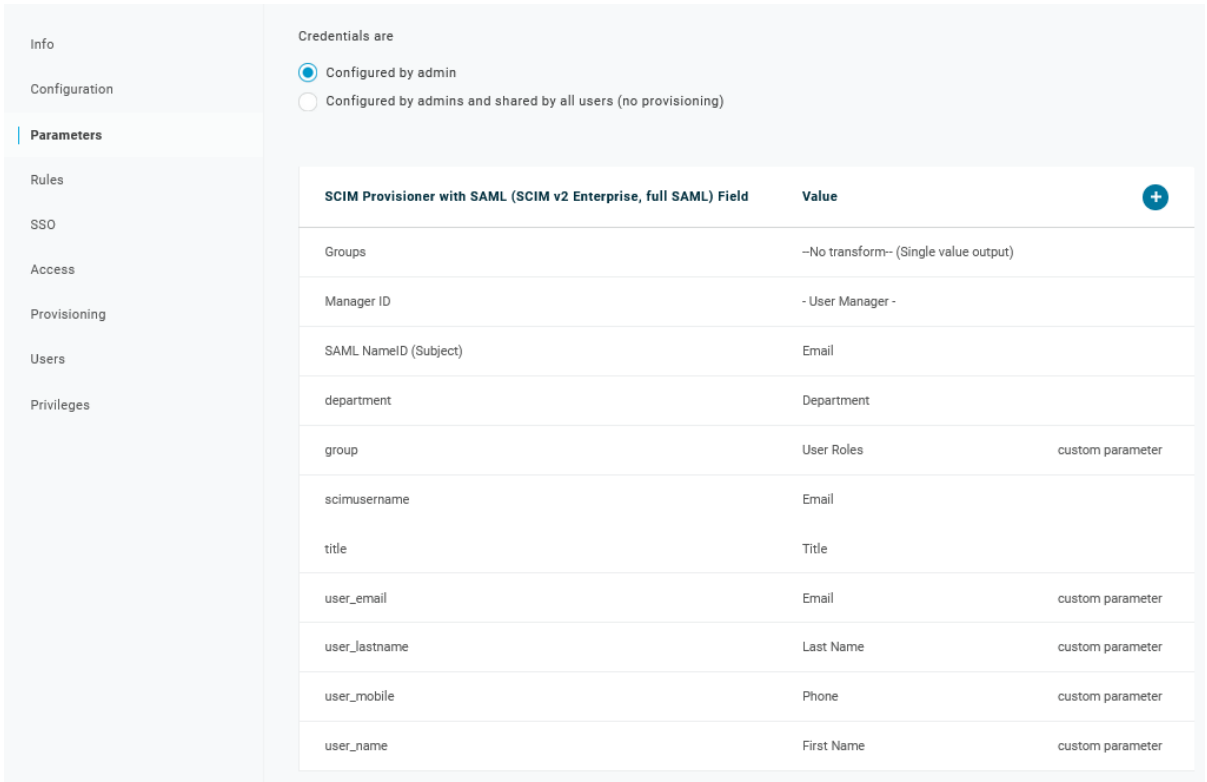


Pievienojot parametru, noteikti atzīmējiet gan opciju *Include in SAML assertion*, gan *Include in User Provisioning*.

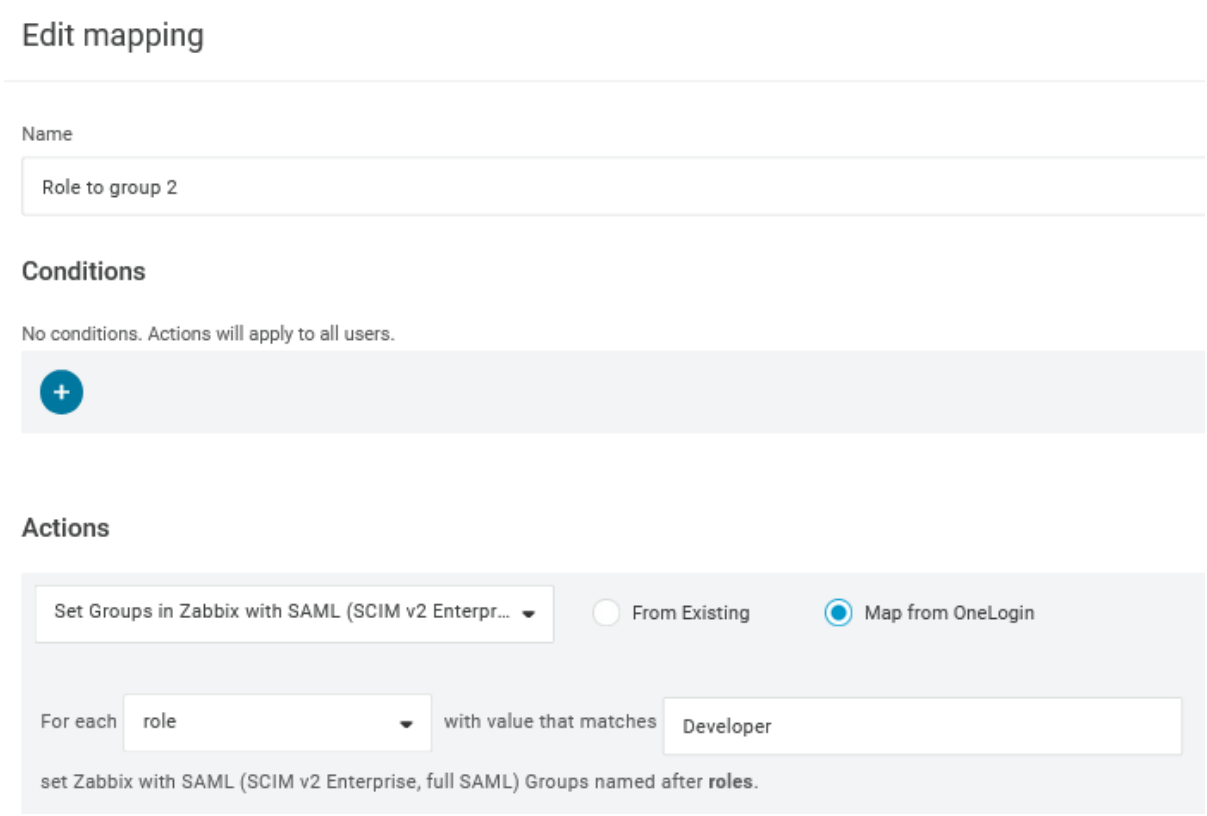
- Pievienojiet parametru 'group', kas atbilst lietotāju lomām OneLogin. Lietotāju lomas tiks nodotas kā virkne, atdalot tās ar semikolu ;. OneLogin lietotāju lomas tiks izmantotas lietotāju grupu izveidei Zabbix:



Pārbaudiet parametru sarakstu:



5. Lapā *Rules* izveidojiet lietotāju lomu kartējumus uz noklusējuma parametru Groups.



Varat izmantot regulāro izteiksmi, lai nodotu konkrētas lomas kā grupas. Lomu nosaukumos nedrīkst būt `;`, jo OneLogin to izmanto kā atdalītāju, nosūtot atribūtu ar vairākām lomām.

6. Lejupielādējiet IdP sertifikātu. Iestatiet tam 644 atļaujas, izpildot:

```
chmod 644 idp.crt
```

Zabbix konfigurēšana

1. Zabbix atveriet SAML iestatījumus un aizpildiet konfigurācijas opcijas, pamatojoties uz OneLogin konfigurāciju:



Zabbix lauks	Iestatīšanas lauks OneLogin	Piemēra vērtība
IdP entītijas ID	Issuer URL (skatiet savas lietojumprogrammas SSO cilni OneLogin)	

Zabbix lauks	Iestatīšanas lauks OneLogin	Piemēra vērtība
SSO pakalpojuma URL	SAML 2.0 Endpoint (HTTP) (skatiet savas lietojumprogrammas SSO cilni OneLogin)	
SLO pakalpojuma URL	SLO Endpoint (HTTP) (skatiet savas lietojumprogrammas SSO cilni OneLogin)	
Lietotājvārda atribūts	Custom parameter	`user_email`
Grupas nosaukuma atribūts	Custom parameter	`group`
Lietotāja vārda atribūts	Custom parameter	`user_name`
Lietotāja uzvārda atribūts	Custom parameter	`user_lastname`

Ir arī jākonfigurē lietotāju grupu kartēšana.
Multivides kartēšana nav obligāta.
Noklikšķiniet uz *Atjaunināt*, lai saglabātu šos iestatījumus.

2. Pievienojiet OneLogin nodrošināto Base64 sertifikātu.

Ja lietotāja saskarnes konfigurācijas failā (`zabbix.conf.php`) ir iestatīts `$SSO['CERT_STORAGE'] = 'database'`, SAML konfigurēšanas laikā varat ielīmēt sertifikāta tekstu vai augšupielādēt sertifikāta failu lietotāja saskarnē — faili failu sistēmā nav nepieciešami.

Ja konfigurācijas failā ir iestatīts `$SSO['CERT_STORAGE'] = 'file'`, sertifikātam jābūt pieejamam failu sistēmā (pēc noklusējuma mapē `ui/conf/certs` vai konfigurācijas failā norādītajā ceļā), un lietotāja saskarne sertifikātus nesaglabās datubāzē. Ņemiet vērā: ja `$SSO['CERT_STORAGE']` nav iestatīts vai ir komentēts, tiek pieņemta glabāšana failā un sertifikāti tiek nolasīti no `ui/conf/certs`.

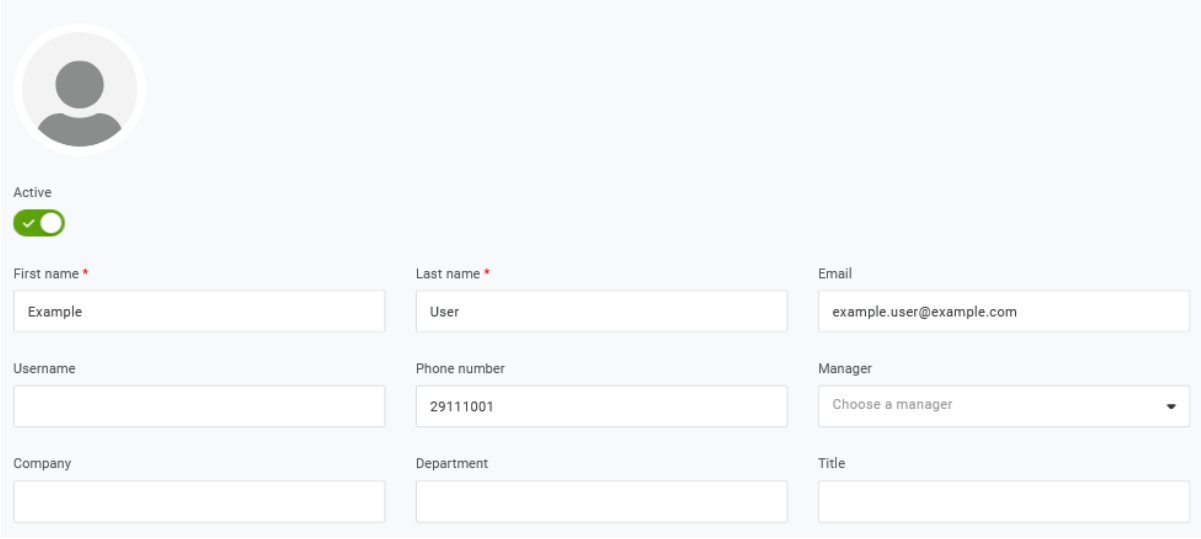
OneLogin sertifikāta lejupielādei varat piekļūt, atverot *Applications* > *SSO* un zem pašreizējā sertifikāta noklikšķinot uz *View details*.

3. Noklikšķiniet uz pogas *Atjaunināt*, lai saglabātu šos iestatījumus.

SCIM lietotāju nodrošināšana

Ja lietotāju nodrošināšana ir iespējota, tagad ir iespējams pievienot/atjaunināt lietotājus un viņu lomas OneLogin un nekavējoties nodrošināt tos Zabbix.

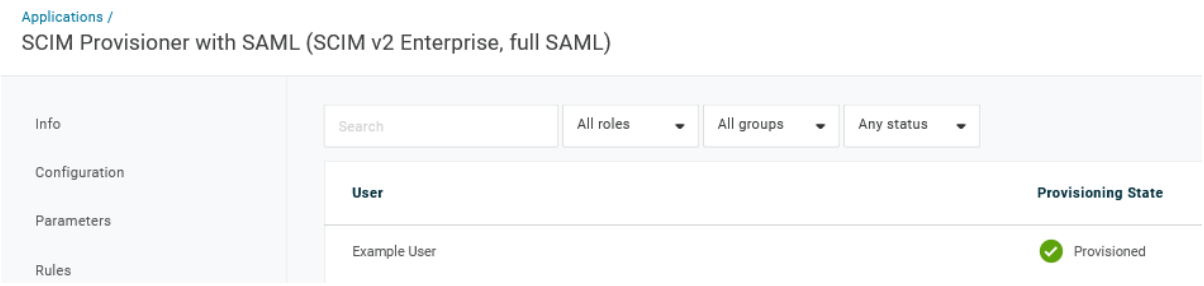
Piemēram, varat izveidot jaunu lietotāju:



Pievienojiet to lietotāja lomai un lietojumprogrammai, kas nodrošinās lietotāju:



Saglabājot lietotāju, tas tiks nodrošināts Zabbix. Sadaļā *Application* > *Users* varat pārbaudīt pašreizējo lietojumprogrammas lietotāju nodrošināšanas statusu:



Ja nodrošināšana ir veiksmīga, lietotāju var redzēt 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

Plānoto 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 pakalpojumu no avota kodiem, skatiet Zabbix tīmekļa pakalpojuma instalēšana.

Pēc instalēšanas palaidiet `zabbix_web_service` tajā datorā, kur ir instalēts tīmekļa pakalpojums:

```
zabbix_web_service
```

Konfigurācija

Lai nodrošinātu pareizu saziņu starp visiem iesaistītajiem elementiem, pārliecinieties, ka servera konfigurācijas 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 servisu.
URL ir jābūt šādā formātā: `http[s]://host:port/report`.

- Pēc noklusējuma tīmekļa serviss klausās 10053. portā.
  Citu portu var norādīt tīmekļa servisa konfigurācijas failā.
- Ceļa `/report` norādīšana ir obligāta (ceļš ir iekodēts un to nevar mainīt).

Piemērs:

```
WebServiceURL=http://localhost:10053/report
```

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.

StartReportWriters

Šis parametrs nosaka, cik atskaišu ģenerēšanas procesiem ir jābūt palaistiem.
Ja tas nav iestatīts vai tā vērtība ir 0, atskaišu ģenerēšana ir atspējota.
Atkarībā no nepieciešamo atskaišu skaita un biežuma var iespējot no 1 līdz 100 atskaišu ģenerēšanas procesiem.

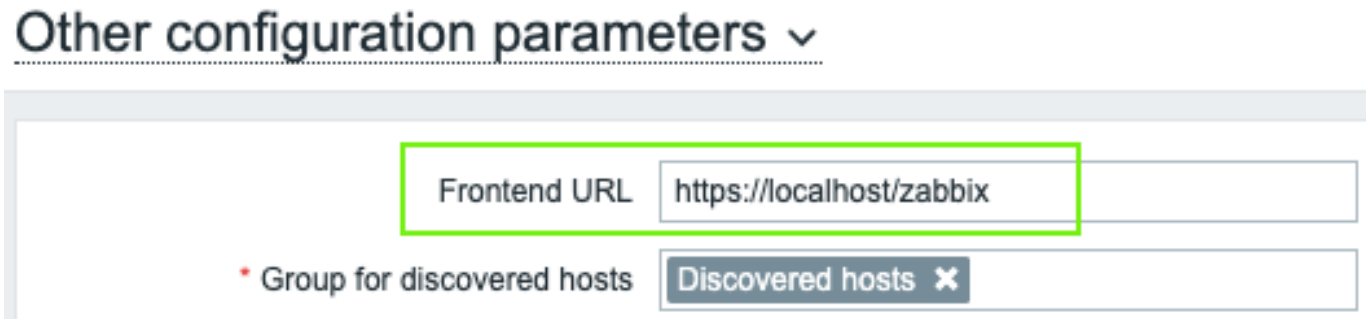
Piemērs:

```
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.

Vienumi

Šajā sadaļā ir ietverta atsauces informācija un tehniskas piezīmes par vienumiem, tostarp parametru informācija konkrētiem vienumu atslēgu tipiem, platformai specifiska informācija un piezīmes par īpašām vienumu konfigurācijām.

Lūdzu, izmantojiet sānjoslu, lai atrastu informāciju, kas attiecas uz jūsu lietošanas gadījumu.

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

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.

Windows atbalstītie biežāk lietotie aģenta vienumi

Vienuma atslēga	Lietotāju grupa	
	Ieteicamais	Minimāli tehniski pieņemamais (funkcionalitāte ir ierobežota)
agent.hostname	Viesi	Viesi
agent.ping	Viesi	Viesi
agent.variant	Viesi	Viesi
agent.version	Viesi	Viesi

Vienuma atslēga	Lietotāju grupa	
log	Administratori	Viesi
log.count	Administratori	Viesi
logrt	Administratori	Viesi
logrt.count	Administratori	Viesi
net.dns	Viesi	Viesi
net.dns.perf	Viesi	Viesi
net.dns.record	Viesi	Viesi
net.if.discovery	Viesi	Viesi
net.if.in	Viesi	Viesi
net.if.out	Viesi	Viesi
net.if.total	Viesi	Viesi
net.tcp.listen	Viesi	Viesi
net.tcp.port	Viesi	Viesi
net.tcp.service	Viesi	Viesi
net.tcp.service.perf	Viesi	Viesi
net.udp.service	Viesi	Viesi
net.udp.service.perf	Viesi	Viesi
proc.num	Administratori	Viesi
system.cpu.discovery	Veiktspējas monitora lietotāji	Veiktspējas monitora lietotāji
system.cpu.load	Veiktspējas monitora lietotāji	Veiktspējas monitora lietotāji
system.cpu.num	Viesi	Viesi
system.cpu.util	Veiktspējas monitora lietotāji	Veiktspējas monitora lietotāji
system.hostname	Viesi	Viesi
system.localtime	Viesi	Viesi
system.run	Administratori	Viesi
system.sw.arch	Viesi	Viesi
system.swap.size	Viesi	Viesi
system.uname	Viesi	Viesi
system.uptime	Veiktspējas monitora lietotāji	Veiktspējas monitora lietotāji
vfs.dir.count	Administratori	Viesi
vfs.dir.get	Administratori	Viesi
vfs.dir.size	Administratori	Viesi
vfs.file.cksum	Administratori	Viesi
vfs.file.contents	Administratori	Viesi
vfs.file.exists	Administratori	Viesi
vfs.file.md5sum	Administratori	Viesi
vfs.file.regexp	Administratori	Viesi
vfs.file.regmatch	Administratori	Viesi
vfs.file.size	Administratori	Viesi
vfs.file.time	Administratori	Viesi
vfs.fs.discovery	Administratori	Viesi
vfs.fs.size	Administratori	Viesi
vfs.fs.get	Administratori	Viesi
vm.memory.size	Viesi	Viesi
web.page.get	Viesi	Viesi
web.page.perf	Viesi	Viesi
web.page.regexp	Viesi	Viesi
zabbix.stats	Viesi	Viesi

Windows specifiskie vienumu atslēgas

Vienuma atslēga	Lietotāju grupa	
	Ieteicamais	Minimāli tehniski pieņemamais (funkcionalitāte ir ierobežota)
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

Vienuma atslēga	Lietotāju grupa	
wmi.get	Administrators	Guests
vm.vmemory.size	Guests	Guests

Atgriezto vērtību kodējums

Zabbix serveris sagaida, ka katra atgrieztā teksta vērtība ir UTF8 kodējumā. Tas attiecas uz jebkura veida pārbaudēm: Zabbix aģents, SSH, Telnet u. c.

Dažādas uzraudzītās sistēmas/ierīces un pārbaudes vērtībā var atgriezt ne-ASCII rakstzīmes. Šādos gadījumos gandrīz visos iespējamos **zabbix** atslēgu gadījumos ir papildu vienuma atslēgas parametrs - **<encoding>**. Šis atslēgas parametrs nav obligāts, taču tas jānorāda, ja atgrieztā vērtība nav UTF8 kodējumā un tajā ir ne-ASCII rakstzīmes. Pretējā gadījumā rezultāts var būt negaidīts un neparedzams.

Tālāk ir sniegts apraksts par darbību ar dažādiem datubāzu aizmugursistēmas risinājumiem šādos gadījumos.

MySQL

Ja vērtība satur ne-ASCII rakstzīmi, kas nav UTF8 kodējumā, 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 ne UTF8 kodējumā, 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.

Lielu failu atbalsts

Lielu failu atbalsts, ko bieži saīsina kā LFS, ir termins, kas apzīmē iespēju strādāt ar failiem, kas ir lielāki par 2 GB 32 bitu operētājsistēmās. Lielu failu atbalsts ietekmē vismaz žurnāla failu uzraudzību un visus vfs.file.* vienumus. Lielu failu atbalsts ir atkarīgs no sistēmas iespējām Zabbix kompilēšanas laikā, taču 32 bitu Solaris tas ir pilnībā atspējots, jo nav saderīgs ar procfs un swapctl.

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].

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. Turklāt dažās platformās tiek atbalstītas arī platformai specifiskas memtype vērtības.

AIX

Tabulā skatiet AIX parametram memtype atbalstītās vērtības.

Atbalstītā vērtība	Apraksts	Avots procentry64 struktūrā	Mēģina nodrošināt saderību ar
vsize [1]	Virtuālās atmiņas lielums	pi_size	
pmem	Reālās atmiņas procentuālā daļa	pi_prm	ps -o pmem
rss	Rezidentās kopas lielums	pi_trss + pi_drss	ps -o rssize
size	Procesa lielums (kods + dati)	pi_dvm	ps gvw kolonna SIZE
dsize	Datu lielums	pi_dsize	
tsize	Teksta (koda) lielums	pi_tsize	ps gvw kolonna TSIZ
sdsize	Koplietojamās bibliotēkas datu lielums	pi_sdsize	
drss	Rezidentās datu kopas lielums	pi_drss	
trss	Rezidentās teksta kopas lielums	pi_trss	

Piezīmes par AIX:

1. Izvēloties parametrus proc.mem[] vienuma atslēgai AIX sistēmā, mēģiniet norādīt pēc iespējas precīzākus procesu atlases kritērijus. Pretējā gadījumā pastāv risks, ka proc.mem[] rezultātā tiks ieskaitīti nevēlami procesi.

Piemērs:

```
$ zabbix_agentd -t proc.mem[,,,NonExistingProcess,rss]
proc.mem[,,,NonExistingProcess,rss]           [u|2879488]
```

Šis piemērs parāda, ka, norādot tikai komandrindas parametru (regulāro izteiksmi, pēc kuras veikt atbilstības meklēšanu), Zabbix aģents uzskaita pats sevi, kas, visticamāk, nav tas, ko vēlaties.

2. Neizmantojiet ps -ef, lai pārlūkotu procesus — tas parāda tikai procesus, kas nav kodola procesi. Izmantojiet ps -Af, lai skatītu visus procesus, kurus redzēs Zabbix aģents.
3. Apskatīsim topasrec piemēru, lai noskaidrotu, 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 šādi argumenti:

proc.mem[<name>,<user>,<mode>,<cmdline>,<memtype>]

kur: - Pirmais kritērijs ir procesa nosaukums (arguments <name>). Mūsu piemērā Zabbix aģents to redzēs kā topasrec. Lai panāktu atbilstību, jānorāda topasrec vai arī lauks jāatstāj tukšs. - Otrais kritērijs ir lietotāja vārds (arguments <user>). Lai panāktu atbilstību, jānorāda root vai arī lauks jāatstāj tukšs. - Trešais kritērijs, ko izmanto procesu atlasē, ir arguments <cmdline>. Zabbix aģents tā vērtību redzēs 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 panāktu atbilstību, jānorāda regulāra izteiksme, kas atbilst šai virknei, vai arī lauks jāatstāj tukšs.

Argumenti <mode> un <memtype> tiek lietoti pēc iepriekš minēto trīs kritēriju izmantošanas.

FreeBSD

Tabulā skatiet FreeBSD atbalstītās parametra memtype vērtības.

Atbalstītā vērtība	Apraksts	Avots kinfo_proc struktūrā	Mēģina nodrošināt saderību ar
vsize	Virtuālās atmiņas lielums	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 lielums	kp_eproc.e_vm.vm_rssize vai ki_rssize	ps -o rss
size [1]	Procesa lielums (kods + dati + steks)	tsize + dsize + ssize	
tsize	Teksta (koda) lielums	kp_eproc.e_vm.vm_tsize vai ki_tsize	ps -o tsiz
dsize	Datu lielums	kp_eproc.e_vm.vm_dsize vai ki_dsize	ps -o dsiz
ssize	Steka lielums	kp_eproc.e_vm.vm_ssize vai ki_ssize	ps -o ssiz

Linux

Tabulā skatiet Linux atbalstītās memtype parametra 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	Fiksēto 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 mijmaiņas vietas lielums	VmSwap

Piezīmes par Linux:

1. Vecāki Linux kodoli neatbalsta visas memtype vērtības. Piemēram, Linux 2.4 kodoli neatbalsta hwm, pin, peak, pte un swap vērtības.
2. Esam novērojuši, ka Zabbix aģenta aktīvās pārbaudes procesa pašuzraudzība ar proc.mem[...,...,...,...,data] uzrāda vērtību, kas ir par 4 kB lielāka nekā VmData rindā norādītā vērtība aģenta failā /proc/<pid>/status. Pašmērījuma laikā aģenta datu segments palielinās par 4 kB un pēc tam atgriežas iepriekšējā lielumā.

Solaris

Tabulā skatiet Solaris atbalstītās parametra memtype vērtības.

Atbalstītā vērtība	Apraksts	Avots psinfo struktūrā	Mēģina nodrošināt saderību ar
vsize [1]	Procesa attēla lielums	pr_size	ps -o vsz
pmem	Reālās atmiņas procentuālā daļa	pr_pctmem	ps -o pmem
rss	Rezidentās kopas lielums Tas var būt novērtēts par zemu (skatiet rss aprakstu sadaļā man ps).	pr_rssize	ps -o rss

Piezīmes

[1] Noklusējuma vērtība.

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`.

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.

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

Linux operētājsistēmā katram režīmam tiek atgriezti šādi procesa parametri:

mode=process	mode=thread	mode=summary
pid: procesa PID	pid: procesa PID	name: procesa nosaukums
ppid: vecākprocesa PID	ppid: vecākprocesa PID	processes: procesu skaits
name: procesa nosaukums	name: procesa nosaukums	vsize: virtuālās atmiņas lielums
cmdline: komanda ar argumentiem	user: lietotājs (reālais), kura vārdā darbojas process	pmem: reālās atmiņas procentuālā daļa
user: lietotājs (reālais), kura vārdā darbojas process	group: grupa (reālā), kuras vārdā darbojas process	rss: rezidentās kopas lielums
group: grupa (reālā), kuras vārdā darbojas process	uid: lietotāja ID	data: datu segmenta lielums
uid: lietotāja ID	gid: grupas, kuras vārdā darbojas process, ID	exe: koda segmenta lielums
gid: grupas, kuras vārdā darbojas process, ID	tid: pavediena ID	lib: koplietojamo bibliotēku lielums
vsize: virtuālās atmiņas lielums	tname: pavediena nosaukums	lck: bloķētās atmiņas lielums
pmem: reālās atmiņas procentuālā daļa	cputime_user: kopējais CPU laiks, kas pavadīts lietotāja režīmā (vērtība izteikta takts impulsos — lai iegūtu sekundes, daliet ar `getconf CLK_TCK` vai `sysconf(_SC_CLK_TCK)`)	pin: piesaistīto lapu lielums
rss: rezidentās kopas lielums	cputime_system: kopējais CPU laiks, kas pavadīts sistēmas režīmā (vērtība izteikta takts impulsos — lai iegūtu sekundes, daliet ar `getconf CLK_TCK` vai `sysconf(_SC_CLK_TCK)`)	pte: lapu tabulas ierakstu lielums

mode=process	mode=thread	mode=summary
data: datu segmenta lielums	state: pavediena stāvoklis	size: procesa koda, datu un steka segmentu kopējais lielums
exe: koda segmenta lielums	ctx_switches: konteksta pārslēgšanu skaits	stk: steka segmenta lielums
hwm: maksimālais rezidentās kopas lielums	page_faults: lapu kļūmju skaits	swap: izmantotās mijmaiņas vietas lielums
lck: bloķētās atmiņas lielums		cputime_user: kopējais CPU laiks, kas pavadīts lietotāja režīmā (vērtība izteikta takts impulsos — lai iegūtu sekundes, daliet ar `getconf CLK_TCK` vai `sysconf(_SC_CLK_TCK)`)
lib: koplietojamo bibliotēku lielums		cputime_system: kopējais CPU laiks, kas pavadīts sistēmas režīmā (vērtība izteikta takts impulsos — lai iegūtu sekundes, daliet ar `getconf CLK_TCK` vai `sysconf(_SC_CLK_TCK)`)
peak: maksimālais virtuālās atmiņas lielums		ctx_switches: konteksta pārslēgšanu skaits
pin: piesaistīto lapu lielums		threads: pavedienu skaits
pte: lapu tabulas ierakstu lielums		page_faults: lapu kļūmju skaits
size: procesa koda, datu un steka segmentu kopējais lielums		pss: proporcionālās kopas atmiņas lielums
stk: steka segmenta lielums		
swap: izmantotās mijmaiņas vietas lielums		
cputime_user: kopējais CPU laiks, kas pavadīts lietotāja režīmā (vērtība izteikta takts impulsos — lai iegūtu sekundes, daliet ar `getconf CLK_TCK` vai `sysconf(_SC_CLK_TCK)`)		
cputime_system: kopējais CPU laiks, kas pavadīts sistēmas režīmā (vērtība izteikta takts impulsos — lai iegūtu sekundes, daliet ar `getconf CLK_TCK` vai `sysconf(_SC_CLK_TCK)`)		
state: procesa stāvoklis (caurskatāmi iegūts no procfs, pilnā formā)		
ctx_switches: konteksta pārslēgšanu skaits		
threads: pavedienu skaits		
page_faults: lapu kļūmju skaits		
pss: proporcionālās kopas atmiņas lielums		

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
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

mode=process	mode=thread	mode=summary
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
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)

mode=process	mode=thread	mode=summary
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		

Nesasniedzamas/nepieejamas hosta saskarnes iestatījumi

Pārskats

Vairāki konfigurācijas parametri nosaka, kā Zabbix serverim jāuzvedas, ja aģenta pārbaude (Zabbix, SNMP, IPMI, JMX) neizdodas un hosts saskarne kļūst nesasniedzama.

Nesasniedzams interfeiss

Hosts interfeiss tiek uzskatīts par nesasniedzamu pēc neveiksmīgas pārbaudes (tīkla kļūda, noildze), ko veic Zabbix, SNMP, IPMI vai JMX aģenti. Sākot ar Zabbix 6.2.0, aktīvās Zabbix aģenta pārbaudes arī ietekmē interfeisa pieejamību. Kad aktīvās pārbaudes kļūst nepieejamas, tās veicina kopējo aģenta interfeisa pieejamības statusu.

No brīža, kad interfeiss kļūst nesasniedzams, **UnreachableDelay** nosaka, cik bieži tas tiek atkārtoti pārbaudīts, izmantojot vienu no vienumiem (tostarp LLD noteikumus). 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 tādiem ziņojumiem kā:

```
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 ziņojumi norāda precīzu neveiksmīgo vienumu un tā tipu (Zabbix aģents).

Note:
Parametrs *Timeout* ietekmēs arī to, cik agri interfeiss tiek atkārtoti pārbaudīts nesasniedzamības laikā. Ja Timeout ir iestatīts uz 20 sekundēm un UnreachableDelay uz 30 sekundēm, nākamā pārbaude notiks pēc 50 sekundēm pēc 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 interfeiss tiek atkārtoti pārbaudīts vairākas reizes, pirms tas tiek atzīmēts kā nepieejams.

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.

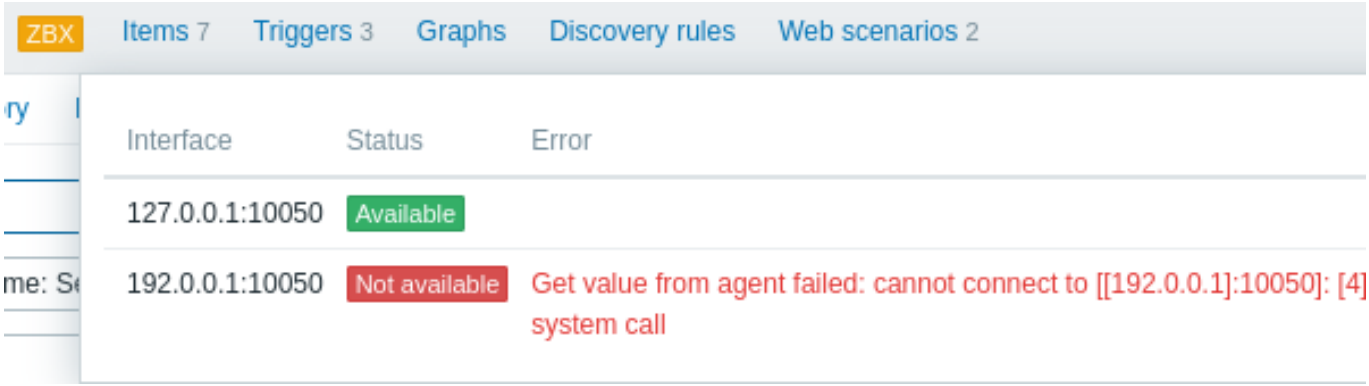
Nepieejama saskarne

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
```

Zabbix statistikas attālināta uzraudzība

Pārskats

Ir iespējams padarīt dažas Zabbix servera un starpniekservera iekšējās metrikas attālināti pieejamas citai Zabbix instancei vai trešās puses rīkam. 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 apvienotās uzraudzības risinājumā.

Zabbix iekšējā statistika tiek eksponēta konfigurējamam adrešu kopumam, kas norādīts jaunajā 'StatsAllowedIP' serveris/starpniekserveris parametrā. Pieprasījumi tiks pieņemti tikai no šīm adresēm.

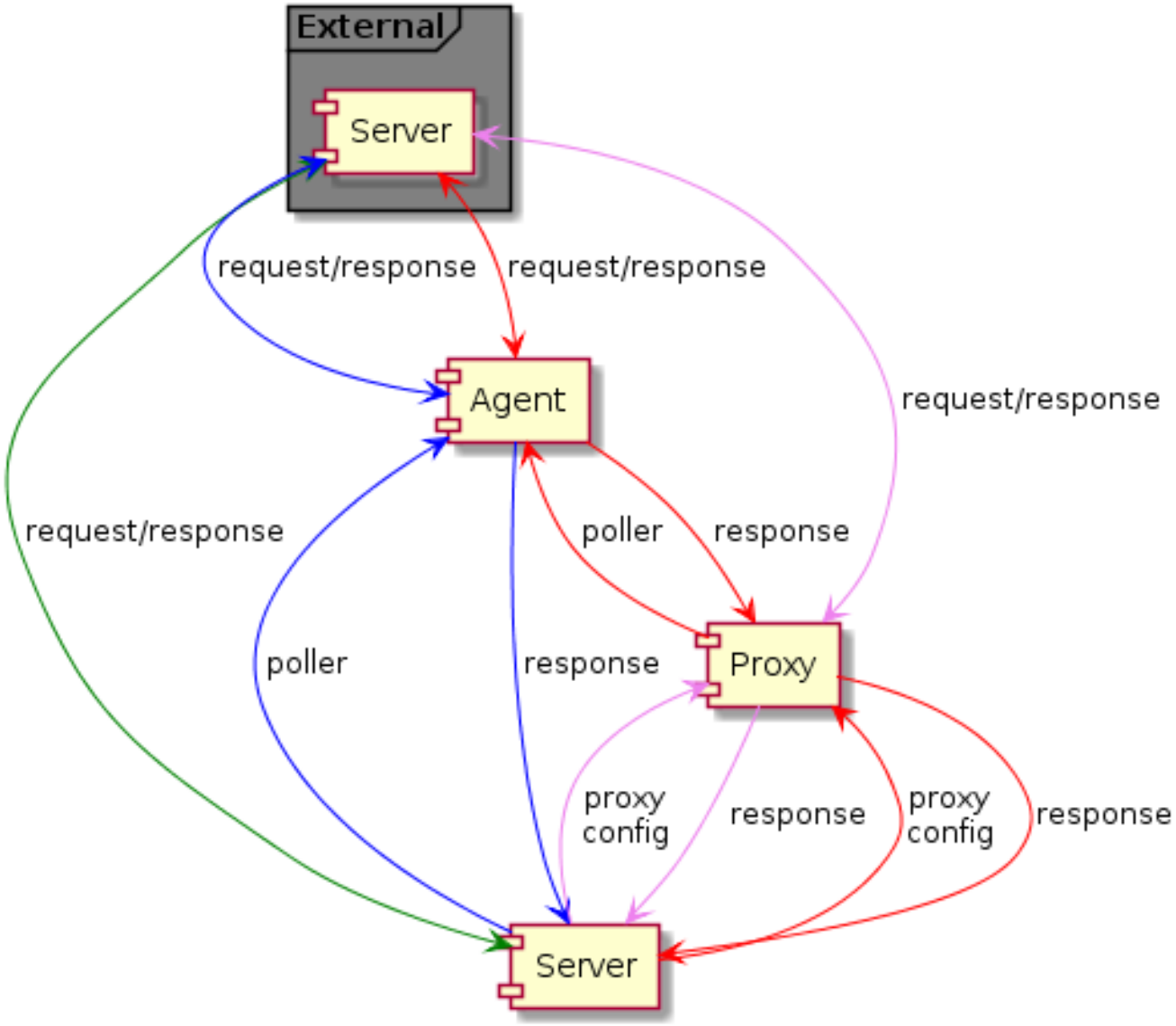
Vienumi

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
      }
    }
  }
}
```


Iekšējie rindas vienumi

Ir pieejami arī divi citi vienumi, kas īpaši paredzēti iekšējās rindas statistikas attālai vaicāšanai citā Zabbix instancē:

- `zabbix[stats,<ip>,<port>,queue,<from>,<to>]` iekšējais vienums — tiešiem iekšējās rindas vaicājumiem attālajā

Zabbix serverī/starpniekserverī
- `zabbix.stats[<ip>,<port>,queue,<from>,<to>]` aģenta vienums — uz aģentu balstītiem iekšējās rindas vaicājumiem attālajā Zabbix serverī/starpniekserverī

Skatiet arī: Iekšējie vienumi, Zabbix aģenta vienumi

Zabbix konfigurēšana ar Kerberos

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 utilītprogrammas:

```
sudo apt update
sudo apt install krb5-kdc krb5-admin-server krb5-user
```

Pakotnes 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. Kartējiet viegli iegaumējamu 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
```

Iestatījumu piemērs:

```
[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 domēna→realm kartēšanas, lai darbotos resursdatora nosaukuma→realm kartēšana. Neatbilstības šeit izraisa kļūdas `Server not found in Kerberos database`.

4. Inicializējiet Kerberos datubāzi (vienreiz, KDC hosts). Kad tiek prasīts, iestatiet drošu galveno paroli:

```
sudo krb5_newrealm
```

5. Izveidojiet `HTTP/host.fqdn@REALM` principālu, izmantojot precīzu resursdatora nosaukumu, ko izmantos klienti; ieteicams izmantot mazos burtus (piemēram, `HTTP/web.example.com@EXAMPLE.COM`). Reģistra vai nosaukuma neatbilstība izraisa kļūdu `Server not found in Kerberos database`.

```
sudo kadmin.local
```

Programmā `kadmin.local`:

```
addprinc kerb-admin@EXAMPLE.COM      # administratīvais principāls
addprinc -randkey HTTP/dc01.example.com@EXAMPLE.COM
ktadd -k /etc/apache2/http.keytab HTTP/dc01.example.com@EXAMPLE.COM
quit
```

Pārvietojiet keytab failu uz tīmekļa serveri (vai atstājiet to lokāli, ja tā ir tā pati iekārta) un iestatiet Apache izmantojamas atļaujas:

```
chown www-data:www-data /etc/apache2/http.keytab
chmod 600 /etc/apache2/http.keytab
#### pārbaude
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 kļūdu `Invalid command 'GssapiCredStore'`, noņemiet neatbalstīto direktīvu vai jauniniet moduli.

7. Konfigurējiet VirtualHost (pielāgojiet `DocumentRoot` / ceļu uz savu Zabbix lietotāja saskarni):

```
sudo vi /etc/apache2/sites-available/zabbix.conf
```

Failā `zabbix.conf`:

```
<VirtualHost *:80>
    ServerName dc01.example.com
    DocumentRoot /usr/share/zabbix/ui
    <Directory /usr/share/zabbix/ui>
        Options FollowSymLinks
        AllowOverride None
        Require all granted
        AuthType GSSAPI
        AuthName "Kerberos Login"
        GssapiCredStore keytab:/etc/apache2/http.keytab
        GssapiLocalName On
        Require valid-user
    </Directory>
    RequestHeader set X-Remote-User %{REMOTE_USER}s env=REMOTE_USER
    RequestHeader unset Authorization
</VirtualHost>
```

Restartējiet Apache:

```
sudo systemctl restart apache2
```

8. Iespējojiet un palaidiet KDC pakalpojumus, kā arī pārbaudiet, vai porti tiek izmantoti (KDC hosts):

```
sudo systemctl enable --now krb5-kdc krb5-admin-server
ss -tnlp | grep :80    # vai: sudo netstat -tnlp | grep :80
```

9. Iegūstiet TGT testēšanai (izpildiet kā lietotājs, kurš izmantos biļeti).

Biļešu sarakstā jābūt redzamam `krbtgt/EXAMPLE.COM@EXAMPLE.COM`. Palaidiet `kinit` kā tas pats OS lietotājs, kuram nepieciešama biļete (piemēram, `zabbix` tīmekļa pārbaužu veikšanai vai `www-data`/Apache interaktīvās pārlūkprogrammas SSO testiem). Citai OS lietotāja identitātei 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). Atbilde 200 OK (vai novirzīšana uz lietotni) norāda, ka SPNEGO ir veiksmīgs:

```
curl -v --negotiate -u : http://dc01.example.com/
```

11. Ja nepieciešams, iespējojiet HTTP autentifikāciju Zabbix lietotāja saskarnē, lai tā pieņemtu ar HTTP autentificētus pieteikumus.

lietotāja saskarnes konfigurācijas failā (`zabbix.conf.php`) iestatiet:

```
$ZBX_FEATURE_FLAGS['http_auth_enabled'] = true;
```

Pēc tam konfigurējiet HTTP autentifikāciju tīmekļa saskarnē.

12. Pārlūkprogrammas konfigurācija (kā piemērs izmantots Firefox): iestatiet `network.negotiate-auth.trusted-uris` uz hosts, kas veic Negotiate autentifikāciju (`dc01.example.com`), lai pārlūkprogramma automātiski nosūtītu Kerberos marķierus.

Sadaļā `about:config`:

```
network.negotiate-auth.trusted-uris = dc01.example.com
```

Tagad, apmeklējot `http://dc01.example.com`, jums vajadzētu tikt tieši pieteiktam Zabbix bez pieteikšanās formas.

13. Uzturiet atslēgas un biļetes aktuālas. Kerberos biļetes noklusējuma darbības laiks ir aptuveni 10 stundas. Pievienojiet cron/systemd taimeri, lai novērstu to derīguma termiņa beigas:

```
#### tīmekļa pakalpojumam
kinit -kt /etc/apache2/http.keytab HTTP/dc01.example.com@EXAMPLE.COM
#### uzraudzības lietotājam
kinit -kt /var/lib/zabbix/kerb.keytab kerb-admin@EXAMPLE.COM
```

14. Regulārās pārbaudes:

- `klist -k /etc/apache2/http.keytab` - pārbaudiet, vai keytab failā ir servisa principāls.
- `sudo tail -f /var/log/apache2/error.log` - pārraugiet GSSAPI kļūdas (`gss_acquire_cred[_from]() failed to get server creds` nozīmē problēmu ar keytab failu vai atļaujām, vai arī trūkstošu principālu).
- Ja `curl --negotiate` atgriež 401/403, tas bieži nozīmē nepareizu principālu, trūkstošu biļeti, neatbilstību hosts galvenē vai failu sistēmas atļauju problēmu; pārbaudiet žurnālus un domēnu kartējumus failā `/etc/krb5.conf`.

Piezīmes par drošību un failu atļaujām

Keytab failiem jābūt lasāmiem tikai kontam, kuram tie ir nepieciešami. Atļauju piemēri: 0400, īpašnieks `zabbix:zabbix`, ja tas ir zabbix lietotāja keytab fails, vai 0440 un `root:www-data`, ja tas ir Apache keytab fails.

Izvairieties no ilgstoši derīgu nešifrētu paroļu glabāšanas uz hosts. Ja iespējams, izmantojiet keytab failus vai domēnam pievienotu datoru principāļus.

Palaižot testus vai skriptus, kas iestata `KRB5CCNAME` vai kopē keytab failus, pēc darbības vēlreiz pārbaudiet īpašnieku un atļaujas — tīmekļa serveris, kas noraida akreditācijas datus, bieži norāda uz faila atļauju problēmu.

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

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 rollup tips, piemēram, *average*

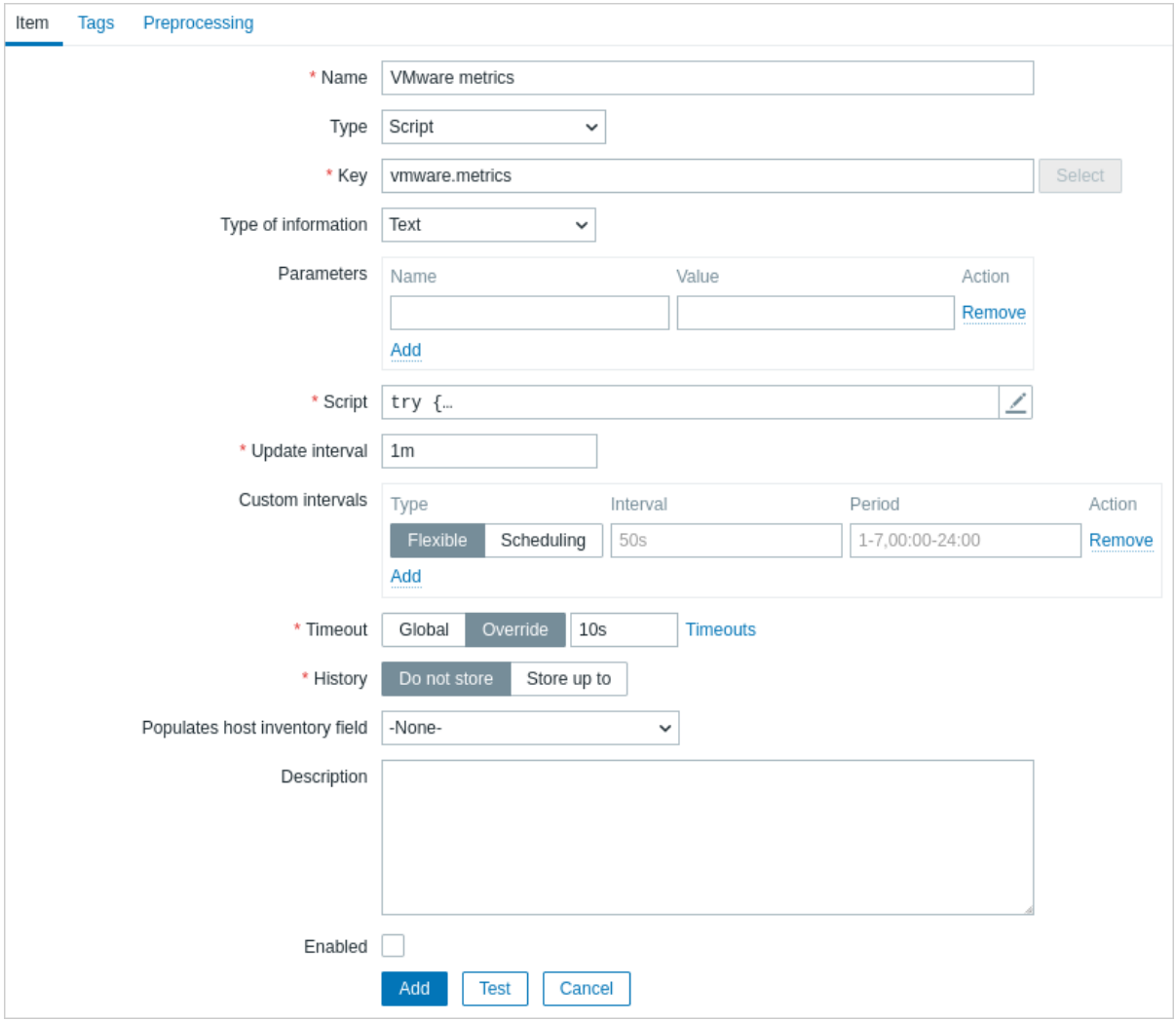
Tātad iepriekš minētais piemērs dotu šādu skaitītāja ceļu: `cpu/usagemhz[average]`

Veiktspējas skaitītāju grupu aprakstus, skaitītāju nosaukumus un rollup tipus var atrast VMware dokumentācijā.

Ir iespējams iegūt iekšējos nosaukumus un izveidot pielāgotus veiktspējas skaitītāju nosaukumus, izmantojot skripta vienumu Zabbix.

Konfigurācija

1. Izveidojiet atspējotu Script vienumu galvenajā VMware hostā (kur ir pieejams **eventlog[]** vienums) ar šādiem parametriem:



- *Nosaukums*: VMware metrics
- *Tips*: Script
- *Atslēga*: vmware.metrics
- *Informācijas tips*: Text
- *Script*: nokopējiet un ielīmējiet zemāk norādīto skriptu
- *Noildze*: 10
- *Vēsture*: Nesaglabāt
- *Ieslēgts*: 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, noklikšķiniet uz pogas *Test*, pēc tam uz *Get value*.



Nokopējiet saņemto XML jebkurā XML formatētājā un atrodiet vajadzīgo metriku.

XML piemērs vienai metrikai:

```
<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 izgūtu skaitītāja ceļu. Iepriekš minētajam piemēram XPath būs:

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]`

Sistēmas system.sw.packages.get atgrieztās vērtības

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 īpašības:

- **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, iestatiet uz 0)
- **arch** - pakotnes arhitektūra
- **buildtime** - objekts ar 2 ierakstiem:
  - **timestamp** - UNIX laika zīmogs, kad pakotne tika izveidota (ja nav pieejams, iestatiet uz 0)
  - **value** - cilvēkam lasāms datums un laiks, kad pakotne tika izveidota (ja nav pieejams, iestatiet uz tukšu virkni)
- **installtime** - objekts ar 2 ierakstiem:
  - **timestamp** - UNIX laika zīmogs, kad pakotne tika instalēta (ja nav pieejams, iestatiet uz 0)
  - **value** - cilvēkam lasāms datums un laiks, kad pakotne tika instalēta (ja nav pieejams, iestatiet 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": ""
        }
    }
]
```

Atgrieztās vērtības net.dns.get funkcijai

Pārskats

Šajā sadaļā ir sniegta informācija par `net.dns.get` Zabbix aģents 2 vienuma atgrieztajām vērtībām.

Detaļas

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 atteiktas 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[192.0.2.1,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[192.0.2.1,example.com]` var atgriezt kļūdu:

```
{
    "zbx_error_code": -1,
    "zbx_error_msg": "Communication error: read udp 192.0.2.1:12345->192.0.2.1:53: i/o timeout"
}
```

Iespējami šādi kļūdu kodu veidi:

Scenārijs	"zbx_error_code"	"zbx_error_msg"
Nav kļūdu, un DNS atbilde tika saņemta un parsēta.	0	
DNS nedarbojas.	-1	"Communication error"
Kļūda rodas JSON parsēšanas laikā	-2	"Received unexpected response"

Ar papildu parametriem vienums `net.dns.get[192.0.2.1,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": "192.0.2.1"
```

```
            },
            "rdlength": 4,
            "ttl": 19308,
            "type": "A"
        },
        {
            "class": "IN",
            "name": "example.com.",
            "rdata": {
                "algorithm": 13,
                "expiration": 1704715951,
                "inception": 1702910624,
                "key_tag": 21021,
                "labels": 2,
                "orig_ttl": 86400,
                "signature": "HVBOBcJJQy0S08J3f8kviPj8UkEUj7wmyiMyQqPSWgQIY9SCEJ5plq6KuxJmtAek1txZWXDo+6tp
                "signer_name": "example.com.",
                "type_covered": "A"
            },
            "rdlength": 95,
            "ttl": 19308,
            "type": "RRSIG"
        }
    ],
    "flags": [
        "RD",
        "RA",
        "AD",
        "CD"
    ],
    "query_time": "0.058221",
    "question_section": [
        {
            "qclass": "IN",
            "qname": "example.com.",
            "qtype": "ANY"
        }
    ],
    "response_code": "NOERROR",
    "zbx_error_code": 0
}
```


Skatiet arī

Lai iegūtu vairāk informācijas par DNS ierakstiem, skatiet:

- Domēna vārdi — ieviešana un specifikācija
- Domēna vārdu sistēmas (DNS) parametri

Atgrieztās vērtības net.if.get funkcijai

Pārskats

Šajā sadaļā ir sniegta informācija par `net.if.get` vienuma atgrieztajām vērtībām. Šo vienumu Linux vidē atbalsta Zabbix aģents un Zabbix aģents 2.

Detalizēta informācija

Šis vienums atgriež detalizētu informāciju par tīkla saskarnēm, kuru nosaukumi atbilst norādītajai regulārajai izteiksmei. Izvade ir strukturēta kā JSON objekts, kas satur `config` un `values` masīvus.

Izvade satur šādas īpašības:

- **name** - saskarnes nosaukums, nolasīts no `/proc/net/dev`;
- **ifalias** - saskarnes aizstājvārds, nolasīts no `/sys/class/net/<name>/ifalias`;
- **mac** - saskarnes MAC adrese, nolasīta no `/sys/class/net/<name>/address`;

- **type** - saskarnes tips (`physical`, `virtual`, `loopback`);
- **speed** - savienojuma ātrums Mbit/s, specifisks Ethernet, nolasīts no `/sys/class/net/<name>/speed`;
- **duplex** - saskarnes pēdējā vai pašreizējā dupleksa vērtība (`half`, `full`), specifiska Ethernet, nolasīta no `/sys/class/net/<name>/du`
- **administrative_state** - konfigurētais stāvoklis (`down`, `up`), nolasīts no `/sys/class/net/<name>/flags`;
- **operational_state** - saskarnes RFC2863 darbības stāvoklis (`unknown`, `notpresent`, `down`, `lowerlayerdown`, `testing`, `dormant`, `up`), nolasīts no `sys/class/net/<name>/operstate`;
- **carrier** - fiziskā savienojuma esamība (0 - fiziskais savienojums ir izslēgts, 1 - fiziskais savienojums ir ieslēgts), nolasīta no `/sys/class/net/<name>/carrier`;
- **carrier_changes** - kopējais reižu skaits, kad savienojuma stāvoklis mainījās no ieslēgta uz izslēgtu un otrādi, nolasīts no `/sys/class/net/<name>/carrier_changes`;
- **carrier_up_count** - reižu skaits, kad savienojums kļuva ieslēgts, nolasīts no `/sys/class/net/<name>/carrier_up_count`;
- **carrier_down_count** - reižu skaits, kad savienojums kļuva izslēgts, nolasīts no `/sys/class/net/<name>/carrier_down_count`;
- **in** - ienākošā datplūsma, nolasīta no `/proc/net/dev`. Satur saskarnes skaitītājus:
  - **bytes** - korekto baitu skaits;
  - **packets** - korekto pakešu skaits;
  - **errors** - kopējais bojāto pakešu skaits;
  - **dropped** - saņemto, bet neapstrādāto pakešu skaits + hosta izlaisto pakešu skaits;
  - **overruns** - pārpildes notikumu skaits;
  - **frame** - pakešu skaits ar kadrēšanas kļūdām;
  - **compressed** - korekti saņemto saspiesto pakešu skaits;
  - **multicast** - saņemtās multiraides paketes.
- **out** - izejošā datplūsma, nolasīta no `/proc/net/dev`. Satur saskarnes skaitītājus:
  - **bytes** - korekto baitu skaits;
  - **packets** - korekto pakešu skaits;
  - **errors** - kopējais bojāto pakešu skaits;
  - **dropped** - saņemto, bet neapstrādāto pakešu skaits + hosta izlaisto pakešu skaits;
  - **overruns** - pārpildes notikumu skaits;
  - **collisions** - sadursmju skaits pakešu pārraides laikā;
  - **carrier** - kadru pārraides kļūdu skaits nesēja zuduma dēļ pārraides laikā;
  - **compressed** - pārraidīto saspiesto pakešu skaits.

JSON izvadē nav iekļauti nepieejamie mērījumi. Tālāk redzamajā piemērā saskarnei "lo" nav konfigurēts aizstājvārds, tāpēc īpašība `ifalias` nav iekļauta.

Piemērs:


```
{
    "config": [
        {
            "name": "lo",
            "mac": "00:00:00:00:00:00",
            "type": "loopback",
            "administrative_state": "up",
            "operational_state": "unknown",
            "carrier": 1
        },
        {
            "name": "enp3s0",
            "ifalias": "Wired Ethernet connection",
            "mac": "8c:16:45:e3:1d:58",
            "type": "physical",
            "duplex": "unknown",
            "administrative_state": "up",
            "operational_state": "down",
            "carrier": 0
        },
        {
            "name": "wlp5s0",
            "ifalias": "Wi-Fi network adapter",
            "mac": "34:e1:2d:72:67:c6",
            "type": "physical",
            "administrative_state": "up",
            "operational_state": "up",
            "carrier": 1
```

```
        }
    ],
    "values": [
        {
            "name": "lo",
            "mac": "00:00:00:00:00:00",
            "carrier": 1,
            "carrier_changes": 0,
            "carrier_up_count": 0,
            "carrier_down_count": 0,
            "in": {
                "bytes": 1040047,
                "packets": 10784,
                "errors": 0,
                "dropped": 0,
                "overruns": 0,
                "frame": 0,
                "compressed": 0,
                "multicast": 0
            },
            "out": {
                "bytes": 1040047,
                "packets": 10784,
                "errors": 0,
                "dropped": 0,
                "overruns": 0,
                "collisions": 0,
                "carrier": 0,
                "compressed": 0
            }
        },
        {
            "name": "enp3s0",
            "ifalias": "Wired Ethernet connection",
            "mac": "8c:16:45:e3:1d:58",
            "carrier": 0,
            "carrier_changes": 1,
            "carrier_up_count": 0,
            "carrier_down_count": 1,
            "in": {
                "bytes": 0,
                "packets": 0,
                "errors": 0,
                "dropped": 0,
                "overruns": 0,
                "frame": 0,
                "compressed": 0,
                "multicast": 0
            },
            "out": {
                "bytes": 0,
                "packets": 0,
                "errors": 0,
                "dropped": 0,
                "overruns": 0,
                "collisions": 0,
                "carrier": 0,
                "compressed": 0
            }
        },
        {
            "name": "wlp5s0",
```

```
            "ifalias": "Wi-Fi network adapter",
            "mac": "34:e1:2d:72:67:c6",
            "carrier": 1,
            "carrier_changes": 6,
            "carrier_up_count": 3,
            "carrier_down_count": 3,
            "in": {
                "bytes": 165537298,
                "packets": 216182,
                "errors": 0,
                "dropped": 291,
                "overruns": 0,
                "frame": 0,
                "compressed": 0,
                "multicast": 0
            },
            "out": {
                "bytes": 252574347,
                "packets": 246607,
                "errors": 0,
                "dropped": 0,
                "overruns": 0,
                "collisions": 0,
                "carrier": 0,
                "compressed": 0
            }
        }
    ]
}
```


Skatiet arī

Plašāku informāciju par sysfs ABI tīkla ierīcēm skatiet:

- Linux kodola dokumentācija

Atgrieztās vērtības funkcijai vfs.dev.get

Pārskats

Šajā sadaļā ir sniegta atgrieztās vērtības informācija par `vfs.dev.get` vienumu. Šis vienums tiek atbalstīts Linux vidē ar Zabbix aģentu un Zabbix aģentu 2, kā arī AIX vidē ar Zabbix aģentu.

Detaļas

Šis vienums atgriež detalizētu informāciju par bloku ierīcēm, kuru nosaukumi atbilst norādītajai regulārajai izteiksmei.

Izvade ir strukturēta kā JSON objekts, kas satur `config` un, atkarībā no <mode> parametra, `values` masīvus.

`config` masīvs satur īpašības, kas ir pieejamas atlasītajam režīmam. `values` masīvs tiek atgriezts tikai tiem režīmiem, kas nodrošina bloku ierīču statistiku.

Nepieejamie metriķi tiek izslēgti no JSON izvades.

Tālākajās sadaļās ir aprakstītas īpašības, kas tiek atgrieztas katram režīmam.

disku režīms

`vfs.dev.get[disks,.*]` atgriež bloka ierīču konfigurācijas datus.

Īpašības:

- **name** - bloka ierīces nosaukums;
- **devid** - bloka ierīces identifikators, piemēram, uz WWN balstīts vai kodola ģenerēts ierīces ID;
- **type** - bloka ierīces tips (`disk`, `partition`, `rom`);
- **path** - ceļš uz bloka ierīces mezglu;
- **model** - ierīces modelis;
- **serial** - ierīces sērijas numurs;
- **wwn** - ierīces globālais nosaukums WWN (World Wide Name);

- **size_bytes** - ierīces izmērs baitos;
- **logical_block_size** - loģiskā bloka izmērs baitos;
- **physical_block_size** - fiziskā bloka izmērs baitos.

Piemērs Linux sistēmai:


```
{
  "config": [
    {
      "name": "sda",
      "devid": "ata-ST2000LM015-2E8174_ZDZ3YHZY",
      "type": "disk",
      "path": "/dev/sda",
      "model": " ST2000LM015-2E81",
      "wwn": "naa.5000c500b2791423",
      "size_bytes": 2000398934016,
      "logical_block_size": 512,
      "physical_block_size": 4096
    },
    {
      "name": "sdb",
      "devid": "usb-Kingston_DataTraveler_3.0_E0D55E6C382C15A158991CBD-0:0",
      "type": "disk",
      "path": "/dev/sdb",
      "model": "DataTraveler 3.0",
      "size_bytes": 61991813632,
      "logical_block_size": 512,
      "physical_block_size": 512
    },
    {
      "name": "nvme0n1",
      "devid": "nvme-SAMSUNG_MZVLW256HEHP-000L7_S35ENX1K753633",
      "type": "disk",
      "path": "/dev/nvme0n1",
      "model": "SAMSUNG MZVLW256HEHP-000L7",
      "serial": "S35ENX1K753633",
      "wwn": "eui.002538b781b7dc55",
      "size_bytes": 256060514304,
      "logical_block_size": 512,
      "physical_block_size": 512
    }
  ]
}
```


Piemērs AIX sistēmai:


```
{
  "config": [
    {
      "name": "cd0",
      "type": "disk",
      "path": "/dev/cd0",
      "size_bytes": 0,
      "logical_block_size": 512
    },
    {
      "name": "hdisk1",
      "type": "disk",
      "devid": "48333321360050768 0C818022F000000000015EA404214503IBMfcp05VDASD03AIXvscsi",
      "path": "/dev/hdisk1",
      "size_bytes": 51136000000,
      "logical_block_size": 512
    },
    {
      "name": "hdisk0",
```

```
      "type": "disk",
      "devid": "48333321360050768OC818022F000000000015EA304214503IBMfcp05VDASD03AIXvscsi",
      "path": "/dev/hdisk0",
      "size_bytes": 0,
      "logical_block_size": 512
    }
  ]
}
```

ierīču režīms

`vfs.dev.get[devices,.*]` atgriež bloka ierīču konfigurācijas datus, tostarp nodalījumus.

Īpašības:

- **name** - bloka ierīces nosaukums.
- **devid** - bloka ierīces identifikators.
- **type** - bloka ierīces tips (`disk`, `partition`, `rom`).
- **partitions** - nodalījumi ierīcē. Satur nodalījumu nosaukumus un to izmērus baitos.

Linux piemērs:

```
{
  "config": [
    {
      "name": "sda",
      "devid": "ata-ST2000LM015-2E8174_ZDZ3YHZY",
      "type": "disk",
      "partitions": {
        "sda1": 16759808,
        "sda2": 2000381018112
      }
    },
    {
      "name": "sdb",
      "devid": "usb-Kingston_DataTraveler_3.0_E0D55E6C382C15A158991CBD-0:0",
      "type": "disk",
      "partitions": {
        "sdb1": 27631878144,
        "sdb2": 34359768576,
        "sdb3": 32256
      }
    },
    {
      "name": "nvme0n1",
      "devid": "nvme-SAMSUNG_MZVLW256HEHP-000L7_S35ENX1K753633",
      "type": "disk",
      "partitions": {
        "nvme0n1p1": 209715200,
        "nvme0n1p2": 16777216,
        "nvme0n1p3": 255060869120,
        "nvme0n1p4": 770703360
      }
    }
  ]
}
```

disk_stats režīms

`vfs.dev.get[disk_stats,.*]` atgriež bloka ierīču konfigurācijas datus un diska statistiku.

Īpašības:

- **name** - bloka ierīces nosaukums.
- **devid** - bloka ierīces identifikators.
- **type** - bloka ierīces tips (`disk`, `partition`, `rom`).
- **size_bytes** - ierīces izmērs baitos.

- **logical_block_size** - loģiskā bloka izmērs baitos.
- **stats** - bloka ierīces I/O statistika. Satur:
  - **reads_completed** - pabeigto lasīšanas operāciju skaits;
  - **writes_completed** - pabeigto rakstīšanas operāciju skaits;
  - **bytes_read** - nolasīto baitu skaits;
  - **bytes_written** - ierakstīto baitu skaits;
  - **io_time_ms** - kopējais I/O veikšanai pavadītais laiks milisekundēs.

Linux piemērs


```
{
  "config": [
    {
      "name": "sda",
      "devid": "ata-ST2000LM015-2E8174_ZDZ3YHZY",
      "type": "disk",
      "size_bytes": 61991813632
    },
    {
      "name": "sdb",
      "devid": "usb-Kingston_DataTraveler_3.0_E0D55E6C382C15A158991CBD-0:0",
      "type": "disk",
      "size_bytes": 61991813632
    },
    {
      "name": "nvme0n1",
      "devid": "nvme-SAMSUNG_MZVLW256HEHP-000L7_S35ENX1K753633",
      "type": "disk",
      "size_bytes": 256060514304
    }
  ],
  "values": [
    {
      "name": "sda",
      "stats": {
        "reads_completed": 871,
        "writes_completed": 0,
        "bytes_read": 13045760,
        "bytes_written": 0,
        "io_time_ms": 1256
      }
    },
    {
      "name": "sdb",
      "stats": {
        "reads_completed": 51017,
        "writes_completed": 10682,
        "bytes_read": 10514627,
        "bytes_written": 1530472,
        "io_time_ms": 284988
      }
    },
    {
      "name": "nvme0n1",
      "stats": {
        "reads_completed": 1259,
        "writes_completed": 7,
        "bytes_read": 49698,
        "bytes_written": 6,
        "io_time_ms": 243
      }
    }
  ]
}
```

AIX piemērs:


```
{
  "config": [
    {
      "name": "cd0",
      "type": "disk",
      "path": "/dev/cd0",
      "size_bytes": 0,
      "logical_block_size": 2048
    },
    {
      "name": "hdisk1",
      "type": "disk",
      "devid": "48333321360050768OC818022F000000000015EA404214503IBMfcp05VDASD03AIXvscsi",
      "path": "/dev/hdisk1",
      "size_bytes": 51136000000,
      "logical_block_size": 512
    },
    {
      "name": "hdisk0",
      "type": "disk",
      "devid": "48333321360050768OC818022F000000000015EA304214503IBMfcp05VDASD03AIXvscsi",
      "path": "/dev/hdisk0",
      "size_bytes": 0,
      "logical_block_size": 512
    }
  ],
  "values": [
    {
      "name": "cd0",
      "stats": {
        "reads_completed": 21,
        "writes_completed": 0,
        "bytes_read": 75776,
        "bytes_written": 0,
        "io_time_ms": 0
      }
    },
    {
      "name": "hdisk1",
      "stats": {
        "reads_completed": 35888,
        "writes_completed": 1165147,
        "bytes_read": 593699840,
        "bytes_written": 10009646592,
        "io_time_ms": 69
      }
    },
    {
      "name": "hdisk0",
      "stats": {
        "reads_completed": 3,
        "writes_completed": 0,
        "bytes_read": 12800,
        "bytes_written": 0,
        "io_time_ms": 0
      }
    }
  ]
}
```


device_stats režīms

`vfs.dev.get[device_stats,.*]` atgriež bloka ierīču konfigurācijas datus un bloka ierīču statistiku.

Īpašības:

- **name** - bloka ierīces nosaukums.
- **devid** - bloka ierīces identifikators.
- **type** - bloka ierīces tips (`disk`, `partition`, `rom`).
- **size_bytes** - ierīces izmērs baitos.
- **stats** - bloka ierīces I/O statistika. Satur:
  - **reads_completed** - pabeigto lasīšanas operāciju skaits;
  - **writes_completed** - pabeigto rakstīšanas operāciju skaits;
  - **bytes_read** - nolasīto baitu skaits;
  - **bytes_written** - ierakstīto baitu skaits;
  - **io_time_ms** - kopējais I/O veikšanai pavadītais laiks milisekundēs.

Piemērs Linux sistēmai:


```
{
  "config": [
    {
      "name": "sda",
      "devid": "ata-ST2000LM015-2E8174_ZDZ3YHZY",
      "type": "disk",
      "size_bytes": 61991813632
    },
    {
      "name": "sda1",
      "type": "partition",
      "devid": "ata-ST2000LM015-2E8174_ZDZ3YHZY-part1",
      "size_bytes": 1234567890
    },
    {
      "name": "sda2",
      "type": "partition",
      "devid": "ata-ST2000LM015-2E8174_ZDZ3YHZY-part2",
      "size_bytes": 1234567890
    },
    {
      "name": "sdb",
      "devid": "usb-Kingston_DataTraveler_3.0_E0D55E6C382C15A158991CBD-0:0",
      "type": "disk",
      "size_bytes": 61991813632
    },
    {
      "name": "sdb1",
      "type": "partition",
      "devid": "usb-Kingston_DataTraveler_3.0_E0D55E6C382C15A158991CBD-0:0-part1",
      "size_bytes": 1234567890
    },
    {
      "name": "sdb2",
      "type": "partition",
      "devid": "usb-Kingston_DataTraveler_3.0_E0D55E6C382C15A158991CBD-0:0-part2",
      "size_bytes": 1234567890
    },
    {
      "name": "sdb3",
      "type": "partition",
      "devid": "usb-Kingston_DataTraveler_3.0_E0D55E6C382C15A158991CBD-0:0-part3",
      "size_bytes": 1234567890
    },
    {
      "name": "nvme0n1",
      "devid": "nvme-SAMSUNG_MZVLW256HEHP-000L7_S35ENX1K753633",
      "type": "disk",
```

```
      "size_bytes": 256060514304
    },
    {
      "name": "nvme0n1p1",
      "type": "partition",
      "devid": "nvme-SAMSUNG_MZVLW256HEHP-000L7_S35ENX1K753633-part1",
      "size_bytes": 1234567890
    },
    {
      "name": "nvme0n1p2",
      "type": "partition",
      "devid": "nvme-SAMSUNG_MZVLW256HEHP-000L7_S35ENX1K753633-part2",
      "size_bytes": 1234567890
    },
    {
      "name": "nvme0n1p3",
      "type": "partition",
      "devid": "nvme-SAMSUNG_MZVLW256HEHP-000L7_S35ENX1K753633-part3",
      "size_bytes": 1234567890
    },
    {
      "name": "nvme0n1p4",
      "type": "partition",
      "devid": "nvme-SAMSUNG_MZVLW256HEHP-000L7_S35ENX1K753633-part4",
      "size_bytes": 1234567890
    }
  ],
  "values": [
    {
      "name": "sda",
      "stats": {
        "reads_completed": 871,
        "writes_completed": 0,
        "bytes_read": 13045760,
        "bytes_written": 0,
        "io_time_ms": 1256
      }
    },
    {
      "name": "sda1",
      "stats": {
        "reads_completed": 123,
        "writes_completed": 123,
        "bytes_read": 123,
        "bytes_written": 123,
        "io_time_ms": 123
      }
    },
    {
      "name": "sda2",
      "stats": {
        "reads_completed": 123,
        "writes_completed": 123,
        "bytes_read": 123,
        "bytes_written": 123,
        "io_time_ms": 123
      }
    },
    {
      "name": "sdb",
      "stats": {
        "reads_completed": 51017,
```

```
      "writes_completed": 10682,
      "bytes_read": 10514627,
      "bytes_written": 1530472,
      "io_time_ms": 284988
    }
  },
  {
    "name": "sdb1",
    "stats": {
      "reads_completed": 123,
      "writes_completed": 123,
      "bytes_read": 123,
      "bytes_written": 123,
      "io_time_ms": 123
    }
  },
  {
    "name": "sdb2",
    "stats": {
      "reads_completed": 123,
      "writes_completed": 123,
      "bytes_read": 123,
      "bytes_written": 123,
      "io_time_ms": 123
    }
  },
  {
    "name": "sdb3",
    "stats": {
      "reads_completed": 123,
      "writes_completed": 123,
      "bytes_read": 123,
      "bytes_written": 123,
      "io_time_ms": 123
    }
  },
  {
    "name": "nvme0n1",
    "stats": {
      "reads_completed": 1259,
      "writes_completed": 7,
      "bytes_read": 49698,
      "bytes_written": 6,
      "io_time_ms": 243
    }
  },
  {
    "name": "nvme0n1p1",
    "stats": {
      "reads_completed": 123,
      "writes_completed": 123,
      "bytes_read": 123,
      "bytes_written": 123,
      "io_time_ms": 123
    }
  },
  {
    "name": "nvme0n1p2",
    "stats": {
      "reads_completed": 123,
      "writes_completed": 123,
      "bytes_read": 123,
```

```
        "bytes_written": 123,
        "io_time_ms": 123
      }
    },
    {
      "name": "nvme0n1p3",
      "stats": {
        "reads_completed": 123,
        "writes_completed": 123,
        "bytes_read": 123,
        "bytes_written": 123,
        "io_time_ms": 123
      }
    },
    {
      "name": "nvme0n1p4",
      "stats": {
        "reads_completed": 123,
        "writes_completed": 123,
        "bytes_read": 123,
        "bytes_written": 123,
        "io_time_ms": 123
      }
    }
  ]
}
```


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
```

Liela JSON vērtību atbalsts

Lai nodrošinātu uzticamu lielu vērtību apstrādi ar JSON datu tipu (1 MiB vai vairāk), pārskatiet un, ja nepieciešams, piemērojiet šādas konfigurācijas izmaiņas:

1. Ja izmantojat MySQL vai MariaDB, palieliniet maksimāli atļautā paketes lielumu failā `/etc/mysql/my.cnf` (dažās sistēmās noklusējuma vērtība var būt 1 MB):

```
[mysqld]
max_allowed_packet=128M

[mysqldump]
max_allowed_packet=1G
```

2. Pielāgojiet papildu MySQL/MariaDB servera sistēmas mainīgos un InnoDB sistēmas mainīgos pēc nepieciešamības. 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 saziņas noildzes starp Zabbix serveri/starpniekserveri un aģentu (piemēram, vienuma noildzes vai `Timeout` parametra vērtību Zabbix servera/starpniekservera un aģenta konfigurācijas failos). Pretējā gadījumā ziņojumi starp tiem var netikt pabeigti, un var tikt parādītas kļūdas, piemēram, *"message length does not match expected length"*. Tāpat pēc nepieciešamības pārskatiet arī citas noildzes, piemēram, skripta noildzi vai vienuma testa noildzi.

Atbalstītās funkcijas

Skatiet Trigeru izteiksme un funkcijas.

Makrokomandas

Ir iespējams izmantot gatavās atbalstītās makrokomandas un lietotāja makrokomandas, ko atbalsta atrašanās vieta.

Atrašanās vietā atbalstītie makrosi

Pārskats

Tabulā makro, ko atbalsta atrašanās vieta ir iekļauta visu makro (iebūvēto, izteiksmju un lietotāja definējamo), kas tiek atbalstīti konkrētajā atrašanās vietā, atsauce.

Ja tiek atbalstīti visi makro no saistītās grupas, tiek norādīts tikai grupas nosaukums. Noklikšķiniet uz katra makro nosaukuma vai grupas nosaukuma, lai skatītu pilnu informāciju par makro.

Sarakstā makro informācija ir detalizēts visu **iebūvēto** makro saraksts, kas sagrupēts pēc lietojuma jomas. Ņemiet vērā, ka, lai pielāgotu makro vērtības (piemēram, saīsinātu vai izvilktu noteiktas apakšvirknes), varat izmantot makro funkcijas.

Makro, ko atbalsta atrašanās vieta

Atrašanās vieta	Atbalstītie makro	Komentāri
Darbība		
Laika perioda nosacījums, Noklusējuma darbības soļa ilgums, Soļa ilgums	Lietotāja definējami makro (tikai globāli)	Vienam makro jāaizpilda viss lauks; vairāki makro vai to apvienošana ar tekstu netiek atbalstīta
Paziņojumi un komandas		
Balstīti uz trigeri	Darbības makro Datuma/laika makro {ESC.HISTORY} Notikuma makro, Cēloņa/simptoma notikuma makro Notikuma atjaunināšanas makro: {EVENT.UPDATE.HISTORY}, {EVENT.UPDATE.STATUS} Izteiksmes makro: {?EXPRESSION} {FUNCTION.VALUE} Hosta makro: {HOST.CONN}, {HOST.DESCRIPTION}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Mērķa hosta makro (tikai komandām) Hosta inventāra makro Vienuma makro (izņemot {ITEM.STATE}, {ITEM.STATE.ERROR}) Starpniekservera makro Trigeru makro: {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 definējami makro	Skatiet arī: Makro izmantošana ziņojumos

Atrašanās vieta	Atbalstītie makro	Komentāri
Problēmas atjaunināšana	Darbības makro Datuma/laika makro {ESC.HISTORY} Notikuma makro, Cēloņa/simptoma notikuma makro, Atjaunošanas notikuma makro Notikuma atjaunināšanas makro: visi (izņemot {EVENT.UPDATE.NSEVERITY}, {EVENT.UPDATE.SEVERITY}) Izteiksmes makro: {?EXPRESSION} Funkciju makro Hosta makro: {HOST.CONN}, {HOST.DESCRIPTION}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Mērķa hosta makro (tikai komandām) Hosta inventāra makro Vienuma makro (izņemot {ITEM.STATE}, {ITEM.STATE.ERROR}) Starpniekservera makro Trigeru makro: {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 definējami makro	
Problēmas atjaunošana	Atjaunošanas notikuma makro Notikuma makro {FUNCTION.RECOVERY.VALUE} {TRIGGER.EXPRESSION.RECOVERY.EXPLAIN}	
Atklāšana	Darbības makro Datuma/laika makro Atklāšanas makro Notikuma makro: {EVENT.AGE}, {EVENT.DATE}, {EVENT.ID}, {EVENT.OBJECT}, {EVENT.SOURCE}, {EVENT.TIME}, {EVENT.TIMESTAMP} Mērķa hosta makro (tikai komandām) Starpniekservera makro	
Automātiska reģistrācija	Darbības makro Datuma/laika makro Notikuma makro: {EVENT.AGE}, {EVENT.DATE}, {EVENT.ID}, {EVENT.OBJECT}, {EVENT.SOURCE}, {EVENT.TIME}, {EVENT.TIMESTAMP} Hosta makro: {HOST.HOST}, {HOST.IP}, {HOST.PORT}, {HOST.METADATA} Mērķa hosta makro (tikai komandām) Starpniekservera makro	
Balstīta uz pakalpojumu	Darbības makro Datuma/laika makro {ESC.HISTORY} Notikuma makro: visi, izņemot {EVENT.ACK.STATUS}, {EVENT.OPDATA} Pakalpojuma makro Lietotāja definējami makro	
Pakalpojuma atjaunošana	Notikuma makro (izņemot {EVENT.ACK.STATUS}, {EVENT.OPDATA}), Atjaunošanas notikuma makro	

Atrašanās vieta	Atbalstītie makro	Komentāri
Pakalpojuma atjaunināšana	Darbības makro Datuma/laika makro {ESC.HISTORY} Notikuma makro (izņemot {EVENT.ACK.STATUS}, {EVENT.OPDATA}) Notikuma atjaunināšanas makro: {EVENT.UPDATE.DATE}, {EVENT.UPDATE.NSEVERITY}, {EVENT.UPDATE.SEVERITY}, {EVENT.UPDATE.STATUS}, {EVENT.UPDATE.TIME}, {EVENT.UPDATE.TIMESTAMP} Pakalpojuma makro Lietotāja definējami makro	
Iekšējie paziņojumi	Darbības makro Datuma/laika makro {ESC.HISTORY} Notikuma makro (izņemot {EVENT.NSEVERITY}, {EVENT.SEVERITY}), Atjaunošanas notikuma makro Hosta makro: {HOST.CONN}, {HOST.DESCRIPTION}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Hosta inventāra makro Vienuma makro: {ITEM.DESCRIPTION}, {ITEM.DESCRIPTION.ORIG}, {ITEM.ID}, {ITEM.KEY}, {ITEM.KEY.ORIG}, {ITEM.NAME}, {ITEM.NAME.ORIG}, {ITEM.VALUETYPE} Starpniekservera makro	
Balstīti uz vienumu	Vienuma makro: {ITEM.STATE}, {ITEM.STATE.ERROR}	
Balstīti uz LLD noteikumu	Zema līmeņa atklāšanas paziņojuma makro	
Balstīti uz trigeri	Trigeru makro: {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 definējami makro	
Brīdinājuma skripta parametri	Brīdinājuma makro Lietotāja definējami makro (tikai globāli)	
Savienotājs		
URL, Lietotājvārds, Parole, Bearer token, HTTP starpniekserveris, SSL sertifikāts, SSL atslēgas fails, SSL atslēgas parole lauki	Lietotāja definējami makro (tikai globāli)	
Noildzes lauks	Lietotāja definējami makro (tikai globāli)	Vienam makro jāaizpilda viss lauks; vairāki makro vai to apvienošana ar tekstu netiek atbalstīta.

Atrašanās vieta	Atbalstītie makro	Komentāri
Grafika nosaukums	Izteiksmes makro: {?EXPRESSION}	Šajā atrašanās vietā šajā makro tiek atbalstīta tikai viena **avg**, **last**, **max**, **min** funkcija ar sekundēm kā parametru (var izmantot laika sufiksus). Makro iekšpusē kā hostu var izmantot {HOST.HOST<1-9>} makro (skatiet arī indeksētos makro). Lietošanas piemērs: {?avg(/{HOST.HOST}/item.key,1h)} Neatbalstīts lietojums: {?last(/host/item1)/last(/host/item2)} {?last(/host/item1)*10} {?count(/host/item1,5m)}
Hosts, hosta prototips		
Saskarnes IP/DNS	Hosta makro: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME} Lietotāja definējami makro [1]	[1] IP adresei vienam makro jāaizpilda viss lauks; vairāki makro vai to apvienošana ar tekstu netiek atbalstīta
Saskarnes ports	Lietotāja definējami makro	Vienam makro jāaizpilda viss lauks; vairāki makro vai to apvienošana ar tekstu netiek atbalstīta.
SNMP kopiena *SNMPv3 konteksta nosaukums, drošības nosaukums, autentifikācijas frāze, privātuma frāze* *IPMI lietotājvārds, parole*	Lietotāja definējami makro	
Vienums, vienuma prototips, LLD noteikums		
Nosaukums	Lietotāja definējami makro	Nav atbalstīts LLD noteikumu nosaukumos.
Apraksts	Lietotāja definējami makro	

Atrašanās vieta	Atbalstītie makro	Komentāri
Atslēgas parametri	Hosta makro: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Lietotāja definējami makro	Atbalstītie {HOST.*} makro tiks atrisināti uz saskarni, kas ir izvēlēta vienumam. Ja tos izmanto vienumos bez saskarnēm, tie tiks atrisināti uz Zabbix aģenta, SNMP, JMX vai IPMI saskarni hostā šādā prioritātes secībā vai uz 'UNKNOWN', ja hostam nav nevienas saskarnes. Makro netiek atbalstīti Zabbix trapper vienumu atslēgā.
Atjaunināšanas intervāls, Pielāgotie intervāli, Noildze (pieejama atbalstītiem vienumu tipiem), Saglabāt līdz (vēsturei/tendencēm) lauki	Lietotāja definējami makro	Vienam makro jāaizpilda viss lauks; vairāki makro vai to apvienošana ar tekstu netiek atbalstīta.
Dzēst zaudētos resursus, Atspējot zaudētos resursus lauki	Lietotāja definējami makro	Šie lauki tiek atbalstīti tikai LLD noteikumiem. Vienam makro jāaizpilda viss lauks; vairāki makro vai to apvienošana ar tekstu netiek atbalstīta.
Pirmapstrādes soļa parametri (tostarp pielāgoti skripti), *pielāgotie kļūdu apstrādes parametri* (*Iestatīt vērtību uz* un *Iestatīt kļūdu uz* lauki)	Lietotāja definējami makro	
Filtra regulārās izteiksmes	Hosta makro: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Lietotāja definējami makro	Šie lauki tiek atbalstīti tikai LLD noteikumiem.
Pārrakstīšanas filtra regulārās izteiksmes	Lietotāja definējami makro	Šie lauki tiek atbalstīti tikai LLD noteikumiem.
Pārrakstīšanas darbības: *Atjaunināšanas intervāls*, *Vēstures glabāšanas periods*, *Tendenču glabāšanas periods* (vienuma prototipiem)	Lietotāja definējami makro	Šie lauki tiek atbalstīti tikai LLD noteikumiem. Vienam makro jāaizpilda viss lauks; vairāki makro vai to apvienošana ar tekstu netiek atbalstīta.
Vienumu tipi		
Pārlūka vienums *parametru nosaukumi un vērtības*	Hosta makro: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Hosta inventāra makro Vienuma makro: {ITEM.ID}, {ITEM.KEY}, {ITEM.KEY.ORIG} Lietotāja definējami makro	

Atrašanās vieta	Atbalstītie makro	Komentāri
Skripta vienums *parametru nosaukumi un vērtības*		
Aprēķinātā/apkopotā vienuma formula (*izteiksmes konstantes*, *funkciju parametri*, *vienuma atslēgas parametri*, (tikai apkopotajam vienumam) *filtra nosacījumi* (hosta grupas nosaukums un taga nosaukums))	Lietotāja definējami makro	
Datubāzes monitora vienums *Lietotājvārds, parole, SQL vaicājums* lauki	Lietotāja definējami makro	
HTTP aģenta vienums *URL, vaicājuma lauki, pieprasījuma pamatteksts, galvenes (nosaukumi un vērtības), SSL sertifikāta fails, SSL atslēgas fails* lauki	Hosta makro: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Vienuma makro: {ITEM.ID}, {ITEM.KEY}, {ITEM.KEY.ORIG} Lietotāja definējami makro	URL, kas satur slepeno makro, nedarbosies, jo tajā esošais makro tiks aizstāts ar "******"
HTTP aģenta vienums *HTTP starpniekserveris* lauks	Vienuma makro: {ITEM.ID}, {ITEM.KEY}, {ITEM.KEY.ORIG} Lietotāja definējami makro	
HTTP aģenta vienums *Nepieciešamie statusa kodi, HTTP autentifikācijas lietotājvārds/parole, SSL atslēgas parole* lauki	Lietotāja definējami makro	
HTTP aģenta vienums *Atļautie hosti* lauks	Hosta makro: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME} Lietotāja definējami makro	
JMX aģenta vienums *Galamērķa* lauks	Hosta makro: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.PORT} Lietotāja definējami makro	
Trapper vienums *Atļautie hosti* lauks	Hosta makro: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP} Lietotāja definējami makro	
SNMP aģenta vienums *SNMP OID* lauks	Lietotāja definējami makro	
SSH un Telnet vienums *skripts*	Hosta makro: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Lietotāja definējami makro	
SSH aģenta vienums *Lietotājvārds, publiskās atslēgas fails, privātās atslēgas fails, parole* lauki	Lietotāja definējami makro	

Atrašanās vieta	Atbalstītie makro	Komentāri
Telnet aģenta vienums *Lietotājvārds, parole* lauki	Lietotāja definējami makro	
Karte		
URL, URL nosaukums	Hosta makro: {HOST.CONN}, {HOST.DESCRIPTION}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} {HOSTGROUP.ID} Hosta inventāra makro Kartes makro {TRIGGER.ID}	
Elementa etiķete	Izteiksmes makro: {?EXPRESSION} [1] Hosta makro: {HOST.CONN}, {HOST.DESCRIPTION}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} {HOSTGROUP.ID} Hosta inventāra makro Kartes makro Trigeru makro: {TRIGGER.EVENTS.*}, {TRIGGER.ID}, {TRIGGER.PROBLEM.EVENTS.*}, {TRIGGERS.ACK}, {TRIGGERS.PROBLEM.ACK}, {TRIGGERS.PROBLEM.UNACK}, {TRIGGERS.UNACK}	[1] Šajā atrašanās vietā šajā makro tiek atbalstīta tikai viena **avg**, **last**, **max**, **min** funkcija ar sekundēm kā parametru (var izmantot laika sufiksus). Makro iekšpusē kā hostu var izmantot {HOST.HOST<1-9>} makro. Lietošanas piemērs: {?avg(/{HOST.HOST}/item.key,1h)} Neatbalstīts lietojums: {?last(/host/item1)/last(/host/item2)} {?last(/host/item1)*10} {?count(/host/item1,5m)}
Saites etiķete	Izteiksmes makro: {?EXPRESSION} [1]	
Formas etiķete	Izteiksmes makro: {?EXPRESSION} [1]	
Teksta lauks kartes formās	{MAP.NAME}	
Tīkla atklāšana		
Atjaunināšanas intervāls lauks	Lietotāja definējami makro (tikai globāli)	Vienam makro jāaizpilda viss lauks; vairāki makro vai to apvienošana ar tekstu netiek atbalstīta.
SNMP kopiena, SNMP OID lauki *SNMPv3 konteksta nosaukums, drošības nosaukums, autentifikācijas frāze, privātuma frāze* lauki	Lietotāja definējami makro (tikai globāli)	
Starpniekserveris		

Atrašanās vieta	Atbalstītie makro	Komentāri
Adrese aktīvajiem aģentiem > *Ports* (ja starpniekserveris pieder grupai) Pasīvā starpniekservera režīmā: saskarnes *Adrese* un *Ports* lauki *Noildzes vienumu tipu* lauki pārrakstīšanas režīmā	Lietotāja definējami makro (tikai globāli)	Vienam makro jāaizpilda viss lauks; vairāki makro vai to apvienošana ar tekstu netiek atbalstīta.
Starpniekserveru grupa		
Pārslēgšanās perioda, Minimālā starpniekserveru skaita lauki	Lietotāja definējami makro (tikai globāli)	Vienam makro jāaizpilda viss lauks; vairāki makro vai to apvienošana ar tekstu netiek atbalstīta.
Skripti		
Komandas, apstiprinājuma teksts (Tips: skripts, manuāla **hosta darbība**)	Hosta makro: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.ID} [1], {HOST.IP}, {HOST.NAME}, {HOST.PORT} Hosta inventāra makro [2] {MANUALINPUT} Lietotājvārda makro Lietotāja definējami makro	[1] Tikai apstiprinājuma tekstam [2] Atbalstīts Zabbix serverim un Zabbix starpniekserverim
Komandas (Tips: skripts, manuāla **notikuma darbība**)	Datuma/laika makro Notikuma makro, Cēloņa/simptoma notikuma makro, Atjaunošanas notikuma makro (ja notikuma atjaunošana ir notikusi) Notikuma atjaunināšanas makro: {EVENT.UPDATE.HISTORY}, {EVENT.UPDATE.STATUS} {FUNCTION.VALUE} Hosta makro: {HOST.CONN}, {HOST.DESCRIPTION}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Hosta inventāra makro Vienuma makro (izņemot {ITEM.STATE}, {ITEM.STATE.ERROR}) {MANUALINPUT} Starpniekservera makro Trigeru makro: {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 makro Lietotāja definējami makro	
Apstiprinājuma teksts (Tips: skripts, manuāla **notikuma darbība**)	Notikuma makro: {EVENT.ID}, {EVENT.NAME}, {EVENT.NSEVERITY}, {EVENT.SEVERITY}, {EVENT.STATUS}, {EVENT.VALUE} Hosta makro: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Hosta inventāra makro {MANUALINPUT} Lietotājvārda makro Lietotāja definējami makro	

Atrašanās vieta	Atbalstītie makro	Komentāri
URL, apstiprinājuma teksts (Tips: URL, manuāla **hosta/notikuma darbība**)		
Taga nosaukums un vērtība	Hosta makro: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Hosta inventāra makro Vienuma makro: {ITEM.LASTVALUE*}, {ITEM.LOG.*}, {ITEM.VALUE*} Trigeru makro: {TRIGGER.ID} (tikai trigeru taga vērtībai) Lietotāja definējami makro	Taga nosaukuma un vērtības makro tiek atrisināti tikai notikuma ģenerēšanas procesa laikā.
Trigeris		
Nosaukums	Hosta makro: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Vienuma makro: {ITEM.LASTVALUE*}, {ITEM.LOG.*}, {ITEM.VALUE*} Pozicionālie makro/atsauces: $1...$9 Lietotāja definējami makro	
Notikuma nosaukums	Datuma/laika makro: {TIME}, {TIMESTAMP} Izteiksmes makro: {?EXPRESSION} Funkciju makro Hosta makro: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Vienuma makro: {ITEM.LASTVALUE*}, {ITEM.LOG.*}, {ITEM.VALUE*} Trigeru makro: {TRIGGER.EXPRESSION.EXPLAIN}	
Operatīvie dati	Hosta makro: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Vienuma makro: {ITEM.LASTVALUE*}, {ITEM.LOG.*}, {ITEM.VALUE*} Lietotāja definējami makro	
Izteiksme	{TRIGGER.VALUE} Lietotāja definējami makro [1]	[1] Tikai konstantēs un funkciju parametros; slepenie makro netiek atbalstīti
Taga atbilstības, izvēlnes ieraksta nosaukuma, izvēlnes ieraksta URL lauki	Lietotāja definējami makro	URL, kas satur slepeno makro, nedarbosies, jo tajā esošais makro tiks aizstāts ar "******".
Apraksts	Hosta makro: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Vienuma makro: {ITEM.LASTVALUE*}, {ITEM.LOG.*}, {ITEM.VALUE*} Lietotāja definējami makro	
URL	{EVENT.ID} Hosta makro: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Vienuma makro: {ITEM.LASTVALUE*}, {ITEM.LOG.*}, {ITEM.VALUE*} {TRIGGER.ID}	
Lietotāja parametru komandas **Tīmekļa scenārijs**	Pozicionālie makro/atsauces: $1...$9	

Atrašanās vieta	Atbalstītie makro	Komentāri
Nosaukums, Aģents, HTTP starpniekserveris lauki	Lietotāja definējami makro	
Atjaunināšanas intervāls lauks	Lietotāja definējami makro	Vienam makro jāaizpilda viss lauks; vairāki makro vai to apvienošana ar tekstu netiek atbalstīta.
Mainīgie, galvenes, SSL sertifikāts, SSL atslēgas fails lauki	Hosta makro: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Lietotāja definējami makro	
Lietotājs, parole, SSL atslēgas parole lauki	Lietotāja definējami makro	
Tīmekļa scenārija solis *Nosaukums, Mainīgie (tikai vērtības)* lauki	Lietotāja definējami makro	
Tīmekļa scenārija solis *Noildze, Nepieciešamie statusa kodi* lauki	Lietotāja definējami makro	Vienam makro jāaizpilda viss lauks; vairāki makro vai to apvienošana ar tekstu netiek atbalstīta.
Tīmekļa scenārija solis *URL, galvenes (nosaukumi un vērtības), nepieciešamā virkne* lauki	Hosta makro: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Lietotāja definējami makro	
Tīmekļa scenārija solis *Post* lauks	Hosta makro: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT}	
Webhook multivides tips		
Parametru vērtības	Brīdinājuma makro Visi makro, kas atbalstīti uz trigeri balstītos problēmu paziņojumos (skatīt iepriekš)	Vienam makro jāaizpilda viss lauks; vairāki makro vai to apvienošana ar tekstu netiek atbalstīta.
JavaScript skripts, *JavaScript* skripta parametra nosaukums	Lietotāja definējami makro (tikai globāli)	
Izvēlnes ieraksta nosaukums, izvēlnes ieraksta URL	{EVENT.TAGS.<tag name>}	
Logrīki		
Mēraparāta logrīks *Apraksts* lauks	Hosta makro: {HOST.CONN}, {HOST.DESCRIPTION}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Hosta inventāra makro Vienuma makro (izņemot {ITEM.STATE.ERROR}) Lietotāja definējami makro (tikai globāli)	
Honeycomb logrīks *Primārās/sekundārās etiķetes* lauks		
Vienuma vērtības logrīks *Apraksts* lauks		

Atrašanās vieta	Atbalstītie makro	Komentāri
Top hosts logrīks *Teksta dati* kolonna	Hosta makro: {HOST.CONN}, {HOST.DESCRIPTION}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Hosta inventāra makro Lietotāja definējami makro (tikai globāli)	
URL (dinamisks) logrīks *URL* lauks	Hosta makro: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Lietotāja definējami makro	
***Pielāgota teksta* kolonna**		
Uzraudzība > Problēmas	Hosta makro (izņemot {HOST.METADATA}) Hosta inventāra makro Vienuma makro (izņemot {ITEM.STATE.ERROR}) Trigeru makro: {TRIGGER.ID} Notikuma makro: {EVENT.ID} Lietotāja definējami makro	
Uzraudzība > Hosti	Hosta makro (izņemot {HOST.METADATA}) Hosta inventāra makro Lietotāja definējami makro	
Uzraudzība > Jaunākie dati	Hosta makro (izņemot {HOST.METADATA}) Hosta inventāra makro Vienuma makro (izņemot {ITEM.STATE.ERROR}) Lietotāja definējami makro	
Datu vākšana > Veidnes	Hosta makro: {HOST.DESCRIPTION}, {HOST.HOST}, {HOST.ID}, {HOST.NAME}	
Datu vākšana > Hosti	Hosta makro (izņemot {HOST.METADATA}) Hosta inventāra makro Lietotāja definējami makro	
Citas atrašanās vietas		
Darba laiks lauks	Lietotāja definējami makro (tikai globāli)	Vienam makro jāaizpilda viss lauks; vairāki makro vai to apvienošana ar tekstu netiek atbalstīta.
Noildzes vienumu tipiem lauks (globāls iestatījums)		
Lietotāja multivide *Kad aktīvs* lauks	Lietotāja definējami makro (tikai globāli)	
E-pasta multivides tips *Lietotājvārds, parole* lauki	Lietotāja definējami makro (tikai globāli)	Ieteicami slepenie makro.
Plānotie pārskati *Tēma, ziņojums* lauki	Datuma/laika makro: {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 taga vērtība, uz kuru norāda taga nosaukums.
 Taga nosaukums, kas satur nealfanumeriskus simbolus (tostarp neangļu daudzbaitu UTF rakstzīmes), ir jāievieto dubultpēdiņās. Pēdiņas un slīpsvītras citētā taga nosaukumā ir jāekranē ar 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>}

Pamata notikuma taga vērtība, uz kuru atsaucas taga nosaukums.
 Taga nosaukums, kas satur nealfanumeriskus simbolus (tostarp neangļu daudzbaitu UTF rakstzīmes), ir jāievieto dubultpēdiņās. Pēdiņas un slīpsvītras citātā taga nosaukumā ir jāekranē ar slīpsvītru.
 Tiek izmantots 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 nosaukumu, notikuma nosaukumu, nopietnību, vecumu, pakalpojuma tagus un vērtības.
 Šis makro tiek izmantots 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 atdalītām komatiem atdalītu atkopšanas notikuma tagu saraksts. Tiek izvērsts tukšā virknē, ja tagi neeksistē.

{EVENT.RECOVERY.TAGSJSON}

JSON masīvs, kas satur atkopšanas notikuma taga objektus. Ja tagu nav, tiek izvērsts tukšā masīvā.

{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}

Cilvēklasāms veikto darbību nosaukums(-i) problēmas atjaunināšanas laikā.
 Atrisinās uz šādām vērtībām: *acknowledged*, *unacknowledged*, *commented*, *changed severity from (original severity) to (updated severity)* un *closed* (atkarībā no tā, cik darbību tiek veiktas vienā atjaunināšanā).

{EVENT.UPDATE.ACTIONJSON}

JSON masīvs, kas satur informāciju par darbību(-ām), kas veiktas problēmas atjaunināšanas laikā.
 Iespējamās JSON īpašību vērtības:
- true (īpašībām `acknowledge`, `unacknowledge`, `close`, `unsuppress`, `cause` un `symptom`);
- <ziņojuma virkne> (īpašībai `message`);
- laikspiedols (īpašībām `suppress_until` un `timestamp`) vai 0 (īpašībai `suppress_until`, ja apspiešana ir beztermiņa);
- 0, 1, 2, 3, 4, 5 (īpašībām `old` un `new` severity).

 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-tā uz vienuma balstītās funkcijas vērtība trigeris izteiksmē notikuma laikā.
 Tiek skaitītas tikai tās funkcijas, kuru pirmais parametrs ir `/host/key`.

 Šo makro var izmantot ar skaitlisku indeksu, piemēram, {FUNCTION.VALUE<1-9>}, lai norādītu uz pirmo, otro, trešo utt. funkciju trigeris izteiksmē. Skatiet indexed macros.

{FUNCTION.RECOVERY.VALUE}

N-tā uz vienuma balstītās funkcijas vērtība atkopšanas izteiksmē notikuma brīdī.*
 Tiek skaitītas tikai funkcijas, kuru pirmais parametrs ir `/host/key`.

 Šo makro var izmantot ar skaitlisku indeksu, piemēram, {FUNCTION.RECOVERY.VALUE<1-9>}, lai norādītu uz pirmo, otro, trešo utt. funkciju trigeris izteiksmē. Skatiet indeksētos makro.

Hosta makrosi

{HOST.CONN}

Hosts IP adrese vai DNS nosaukums atkarībā no hosts iestatījumiem.
 Globālajos skriptos, interfeisa IP/DNS laukos un tīmekļa scenārijos makro tiks atrisināts uz galveno aģenta interfeisu. Ja aģenta interfeiss nav definēts, tiks izmantots galvenais SNMP interfeiss. Ja arī SNMP interfeiss nav definēts, tiks izmantots galvenais JMX interfeiss. Ja JMX interfeiss nav definēts, tiks izmantots galvenais IPMI interfeiss. Ja hostam nav neviena interfeisa, makro tiks atrisināts uz 'UNKNOWN'.

 Šo makro var izmantot ar skaitlisku indeksu kā {HOST.CONN<1-9>}, lai trigeris izteiksmē norādītu uz pirmo, otro, trešo utt. hostu. Skatiet indeksētos makro.

{HOST.DESCRIPTION}

Hosta apraksts.

 Šo makro var izmantot ar skaitlisku indeksu kā {HOST.DESCRIPTION<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigera izteiksmē. Skatiet indeksētos makro.

{HOST.DNS}

Hosts DNS nosaukums.
 Globālajos skriptos, saskarnes IP/DNS laukos un tīmekļa scenārijos makro 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 JMX saskarne arī nav definēta, tiks izmantota galvenā IPMI saskarne. Ja hostam nav nevienas saskarnes, makro tiek atrisināts uz 'UNKNOWN'.

 Šo makro var izmantot ar skaitlisku indeksu kā {HOST.DNS<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigeris izteiksmē. Skatiet indeksētos makro.

{HOST.HOST}

Tehniskais hosta nosaukums.
 Makro funkcijas netiek atbalstītas šim makro, ja tas tiek izmantots kā vietturis vēstures funkcijas pirmajā параметrā, 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 trigeru izteiksmē. Skatiet indeksētos makro.

{HOST.ID}

Hosts ID.

 Šo makro var izmantot ar skaitlisku indeksu kā {HOST.ID<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigeru izteiksmē. Skatiet indeksētos makro.

{HOST.IP}

Hosts IP adrese.
 Globālajos skriptos, interfeisa IP/DNS laukos un tīmekļa scenārijos makro tiks atrisināts uz galveno aģenta interfeisu. Ja aģenta interfeiss nav definēts, tiks izmantots galvenais SNMP interfeiss. Ja arī SNMP interfeiss nav definēts, tiks izmantots galvenais JMX interfeiss. Ja JMX interfeiss nav definēts, tiks izmantots galvenais IPMI interfeiss. Ja hostam nav neviena interfeisa, makro tiks atrisināts uz 'UNKNOWN'.

 Šo makro var izmantot ar skaitlisku indeksu kā {HOST.IP<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigerī. Skatiet indeksētos makro.

{HOST.METADATA}

Hosta metadati.

 Izmanto tikai aktīvā aģenta automātiskajai reģistrācijai.

{HOST.NAME}

Redzamais hosta nosaukums.

 Šo makro var izmantot ar skaitlisku indeksu kā {HOST.NAME<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigera izteiksmē. Skatiet indeksētos makro.

{HOST.PORT}

Hosts (aģents) ports.
 Globālajos skriptos, interfeisa IP/DNS laukos un tīmekļa scenārijos makro tiks atrisināts uz galveno aģenta interfeisu. Ja aģenta interfeiss nav definēts, tiks izmantots galvenais SNMP interfeiss. Ja arī SNMP interfeiss nav definēts, tiks izmantots galvenais JMX interfeiss. Ja JMX interfeiss arī nav definēts, tiks izmantots galvenais IPMI interfeiss. Ja hostam nav neviena interfeisa, makro tiks atrisināts uz 'UNKNOWN'.

 Šo makro var izmantot ar skaitlisku indeksu kā {HOST.PORT<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigerī. Skatiet indeksētos makro.

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}

Alias lauks hosta inventarizācijā.

 Šo makro var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.ALIAS<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigeru izteiksmē. Skatiet indeksētos makro.

{INVENTORY.ASSET.TAG}

Lauks *Asset tag* hosta inventarizācijā.

 Šo makro var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.ASSET.TAG<1-9>}, lai trigera izteiksmē norādītu uz pirmo, otro, trešo utt. hostu. Skatiet indeksētos makro.

{INVENTORY.CHASSIS}

Šasijas lauks hosta inventarizācijā.

 Šo makro var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.CHASSIS<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigeru izteiksmē. Skatiet indeksētos makro.

{INVENTORY.CONTACT}

Contact lauks hosta inventarizācijā.

 Šo makro var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.CONTACT<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigeru izteiksmē. Skatiet indeksētos makro.

{INVENTORY.CONTRACT.NUMBER}

Lauks *Līguma numurs* hosta inventarizācijā.

 Šo makro var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.CONTRACT.NUMBER<1-9>}, lai trigeris izteiksmē norādītu uz pirmo, otro, trešo utt. hostu. Skatiet indeksētos makro.

{INVENTORY.DEPLOYMENT.STATUS}

Lauks *Deployment status* hosta inventārā.

 Šo makro var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.DEPLOYMENT.STATUS<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigeris izteiksmē. Skatiet indexed macros.

{INVENTORY.HARDWARE}

Hardware lauks hosta inventarizācijā.

 Šo makro var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.HARDWARE<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigera izteiksmē. Skatiet indeksētos makro.

{INVENTORY.HARDWARE.FULL}

Hardware (pilnīga informācija) lauks hosta inventarizācijā.

 Šo makro var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.HARDWARE.FULL<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigeru izteiksmē. Skatiet indeksētos makro.

{INVENTORY.HOST.NETMASK}

Lauks *Host subnet mask* resursdatora inventarizācijā.

 Šo makro var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.HOST.NETMASK<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigera izteiksmē. Skatiet indexed macros.

{INVENTORY.HOST.NETWORKS}

Lauks *Host networks* resursdatora inventārā.

 Šo makro var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.HOST.NETWORKS<1-9>}, lai trigera izteiksmē norādītu uz pirmo, otro, trešo utt. hostu. Skatiet indeksētos makro.

{INVENTORY.HOST.ROUTER}

Host router lauks hosta inventarizācijā.

 Šo makro var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.HOST.ROUTER<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigeris izteiksmē. Skatiet indeksētos makro.

{INVENTORY.HW.ARCH}

Lauks *Aparatūras arhitektūra* hosta inventarizācijā.

 Šo makro var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.HW.ARCH<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigeru izteiksmē. Skatiet indeksētos makro.

{INVENTORY.HW.DATE.DECOMM}

Lauks *Date hardware decommissioned* hosta inventārā.

 Šo makro var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.HW.DATE.DECOMM<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigera izteiksmē. Skatiet indexed macros.

{INVENTORY.HW.DATE.EXPIRY}

Lauks *Date hardware maintenance expires* hosta inventarizācijā.

 Šo makro var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.HW.DATE.EXPIRY<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigeru izteiksmē. Skatiet indexed macros.

{INVENTORY.HW.DATE.INSTALL}

Lauks *Date hardware installed* hosta inventarizācijā.

 Šo makro var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.HW.DATE.INSTALL<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigeris izteiksmē. Skatiet indexed macros.

{INVENTORY.HW.DATE.PURCHASE}

Datums, kad aparatūra iegādāta lauks hosta inventārā.

 Šo makro var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.HW.DATE.PURCHASE<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigera izteiksmē. Skatiet indeksētos makro.

{INVENTORY.INSTALLER.NAME}

Lauks *Installer name* hosta inventarizācijā.

 Šo makro var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.INSTALLER.NAME<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigeru izteiksmē. Skatiet indexed macros.

{INVENTORY.LOCATION}

Location lauks host inventarizācijā.

 Šo makro var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.LOCATION<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigeru izteiksmē. Skatiet indeksētos makro.

{INVENTORY.LOCATION.LAT}

Atrašanās vietas platuma lauks hosta inventarizācijā.

 Šo makro var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.LOCATION.LAT<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigera izteiksmē. Skatiet indeksētos makro.

{INVENTORY.LOCATION.LON}

Lauks *Atrašanās vietas garums* hosta inventarizācijā.

 Šo makro var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.LOCATION.LON<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigeru izteiksmē. Skatiet indeksētos makro.

{INVENTORY.MACADDRESS.A}

MAC adreses A lauks hosta inventārā.

 Šo makro var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.MACADDRESS.A<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigeru izteiksmē. Skatiet indeksētos makro.

{INVENTORY.MACADDRESS.B}

Lauks *MAC address B* hosta inventārā.

 Šo makro var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.MACADDRESS.B<1-9>}, lai trigeru izteiksmē norādītu uz pirmo, otro, trešo utt. hostu. Skatiet indeksētos makro.

{INVENTORY.MODEL}

Modelis lauks hosta inventarizācijā.

 Šo makro var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.MODEL<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigeru izteiksmē. Skatiet indeksētos makro.

{INVENTORY.NAME}

Nosaukums lauks hosta inventarizācijā.

 Šo makro var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.NAME<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigeru izteiksmē. Skatiet indeksētos makro.

{INVENTORY.NOTES}

Piezīmes lauks hosta inventarizācijā.

 Šo makro var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.NOTES<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigera izteiksmē. Skatiet indeksētos makro.

{INVENTORY.OOB.IP}

OOB IP address lauks hosta inventārā.

 Šo makro var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.OOB.IP<1-9>}, lai trigera izteiksmē norādītu uz pirmo, otro, trešo utt. hostu. Skatiet indeksētos makro.

{INVENTORY.OOB.NETMASK}

OOB apakštīkla maska lauks hosta inventarizācijā.

 Šo makro var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.OOB.NETMASK<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigeru izteiksmē. Skatiet indeksētos makro.

{INVENTORY.OOB.ROUTER}

OOB maršrutētāja lauks hosta inventārā.

 Šo makro var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.OOB.ROUTER<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigeru izteiksmē. Skatiet indeksētos makro.

{INVENTORY.OS}

OS lauks hosta inventarizācijā.

 Šo makro var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.OS<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigeru izteiksmē. Skatiet indeksētos makro.

{INVENTORY.OS.FULL}

Lauks *OS (pilna informācija)* hosta inventarizācijā.

 Šo makro var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.OS.FULL<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigeru izteiksmē. Skatiet indeksētos makro.

{INVENTORY.OS.SHORT}

Lauks *OS (īss)* hosta inventarizācijā.

 Šo makro var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.OS.SHORT<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigera izteiksmē. Skatiet indeksētos makro.

{INVENTORY.POC.PRIMARY.CELL}

Lauks *Primārā POC šūna* hosta inventarizācijā.

 Šo makro var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.POC.PRIMARY.CELL<1-9>}, lai trigeris izteiksmē norādītu uz pirmo, otro, trešo utt. hostu. Skatiet indeksētos makro.

{INVENTORY.POC.PRIMARY.EMAIL}

Primārā POC e-pasta lauks hosta inventarizācijā.

 Šo makro var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.POC.PRIMARY.EMAIL<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigeru izteiksmē. Skatiet indeksētos makro.

{INVENTORY.POC.PRIMARY.NAME}

Lauks *Primary POC name* hosta inventārā.

 Šo makro var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.POC.PRIMARY.NAME<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigeru izteiksmē. Skatiet indexed macros.

{INVENTORY.POC.PRIMARY.NOTES}

Primārā POC piezīmes lauks hosta inventarizācijā.

 Šo makro var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.POC.PRIMARY.NOTES<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigeris izteiksmē. Skatiet indeksētos makro.

{INVENTORY.POC.PRIMARY.PHONE.A}

Primārā POC tālruņa A lauks hosta inventarizācijā.

 Šo makro var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.POC.PRIMARY.PHONE.A<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigeru izteiksmē. Skatiet indeksētos makro.

{INVENTORY.POC.PRIMARY.PHONE.B}

Primārā POC tālruņa B lauks hosta inventarizācijā.

 Šo makro var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.POC.PRIMARY.PHONE.B<1-9>}, lai trigeru izteiksmē norādītu pirmo, otro, trešo utt. hostu. Skatiet indeksētos makro.

{INVENTORY.POC.PRIMARY.SCREEN}

Lauks *Primary POC screen name* resursdatora inventārā.

 Šo makro var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.POC.PRIMARY.SCREEN<1-9>}, lai trigeru izteiksmē norādītu uz pirmo, otro, trešo utt. hostu. Skatiet indexed macros.

{INVENTORY.POC.SECONDARY.CELL}

Lauks *Secondary POC cell* hosta inventārā.

 Šo makro var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.POC.SECONDARY.CELL<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigeru izteiksmē. Skatiet indexed macros.

{INVENTORY.POC.SECONDARY.EMAIL}

Lauks *Secondary POC email* hosta inventārā.

 Šo makro var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.POC.SECONDARY.EMAIL<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigeru izteiksmē. Skatiet indexed macros.

{INVENTORY.POC.SECONDARY.NAME}

Lauks *Secondary POC name* hosta inventarizācijā.

 Šo makro var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.POC.SECONDARY.NAME<1-9>}, lai trigeru izteiksmē norādītu uz pirmo, otro, trešo utt. hostu. Skatiet indexed macros.

{INVENTORY.POC.SECONDARY.NOTES}

Sekundārā POC piezīmes lauks hosta inventārā.

 Šo makro var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.POC.SECONDARY.NOTES<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigeru izteiksmē. Skatiet indeksētos makro.

{INVENTORY.POC.SECONDARY.PHONE.A}

Sekundārā POC tālruņa A lauks hosta inventārā.

 Šo makro var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.POC.SECONDARY.PHONE.A<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigeris izteiksmē. Skatiet indeksētos makro.

{INVENTORY.POC.SECONDARY.PHONE.B}

Sekundārā POC tālruņa B lauks hosta inventārā.

 Šo makro 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 makro.

{INVENTORY.POC.SECONDARY.SCREEN}

Lauks *Secondary POC screen name* host inventarizācijā.

 Šo makro var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.POC.SECONDARY.SCREEN<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigeru izteiksmē. Skatiet indexed macros.

{INVENTORY.SERIALNO.A}

Lauks *Sērijas numurs A* hosta inventārā.

 Šo makro var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.SERIALNO.A<1-9>}, lai trigeru izteiksmē norādītu uz pirmo, otro, trešo utt. hostu. Skatiet indeksētos makro.

{INVENTORY.SERIALNO.B}

Serial number B lauks hosta inventarizācijā.

 Šo makro var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.SERIALNO.B<1-9>}, lai trigeru izteiksmē norādītu uz pirmo, otro, trešo utt. hostu. Skatiet indeksētos makro.

{INVENTORY.SITE.ADDRESS.A}

Lauks *Site address A* hosta inventarizācijā.

 Šo makro var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.SITE.ADDRESS.A<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigeru izteiksmē. Skatiet indeksētos makro.

{INVENTORY.SITE.ADDRESS.B}

Site address B lauks hosta inventarizācijā.

 Šo makro var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.SITE.ADDRESS.B<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigeru izteiksmē. Skatiet indeksētos makro.

{INVENTORY.SITE.ADDRESS.C}

Site address C lauks hosta inventarizācijā.

 Šo makro var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.SITE.ADDRESS.C<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigeru izteiksmē. Skatiet indeksētos makro.

{INVENTORY.SITE.CITY}

Lauks *Site city* hosta inventarizācijā.

 Šo makro var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.SITE.CITY<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigeru izteiksmē. Skatiet indexed macros.

{INVENTORY.SITE.COUNTRY}

Lauks *Site country* hosta inventārā.

 Šo makro var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.SITE.COUNTRY<1-9>}, lai trigeru izteiksmē norādītu uz pirmo, otro, trešo utt. hostu. Skatiet indexed macros.

{INVENTORY.SITE.NOTES}

Vietnes piezīmes lauks hosta inventarizācijā.

 Šo makro var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.SITE.NOTES<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigeru izteiksmē. Skatiet indeksētos makro.

{INVENTORY.SITE.RACK}

Lauks *Vietas statīva atrašanās vieta* hosta inventarizācijā.

 Šo makro var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.SITE.RACK<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigeru izteiksmē. Skatiet indeksētos makro.

{INVENTORY.SITE.STATE}

Lauks *Vietas štats/province* hosta inventārā.

 Šo makro var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.SITE.STATE<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigeru izteiksmē. Skatiet indeksētos makro.

{INVENTORY.SITE.ZIP}

Vietnes pasta indeksa lauks hosta inventārā.

 Šo makro var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.SITE.ZIP<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigeru izteiksmē. Skatiet indeksētos makro.

{INVENTORY.SOFTWARE}

Programmatūra lauks hosta inventarizācijā.

 Šo makro var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.SOFTWARE<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigeru izteiksmē. Skatiet indeksētos makro.

{INVENTORY.SOFTWARE.APP.A}

Programmatūras lietojumprogrammas A lauks hosta inventarizācijā.

 Šo makro var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.SOFTWARE.APP.A<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigeru izteiksmē. Skatiet indeksētos makro.

{INVENTORY.SOFTWARE.APP.B}

Software application B lauks hosta inventārā.

 Šo makro var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.SOFTWARE.APP.B<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigeru izteiksmē. Skatiet indeksētos makro.

{INVENTORY.SOFTWARE.APP.C}

Programmatūras lietojumprogrammas C lauks hosta inventarizācijā.

 Šo makro var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.SOFTWARE.APP.C<1-9>}, lai trigeru izteiksmē norādītu uz pirmo, otro, trešo utt. hostu. Skatiet indeksētos makro.

{INVENTORY.SOFTWARE.APP.D}

Programmatūras lietojumprogrammas D lauks hosta inventarizācijā.

 Šo makro var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.SOFTWARE.APP.D<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigeru izteiksmē. Skatiet indeksētos makro.

{INVENTORY.SOFTWARE.APP.E}

Software application E lauks hosta inventarizācijā.

 Šo makro var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.SOFTWARE.APP.E<1-9>}, lai trigeru izteiksmē norādītu uz pirmo, otro, trešo utt. hostu. Skatiet indeksētos makro.

{INVENTORY.SOFTWARE.FULL}

Lauks *Programmatūra (pilna informācija)* hosta inventarizācijā.

 Šo makro var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.SOFTWARE.FULL<1-9>}, lai trigera izteiksmē norādītu uz pirmo, otro, trešo utt. hostu. Skatiet indeksētos makro.

{INVENTORY.TAG}

Tag lauks hosta inventarizācijā.

 Šo makro var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.TAG<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigeru izteiksmē. Skatiet indeksētos makro.

{INVENTORY.TYPE}

Tips lauks hosta inventarizācijā.

 Šo makro var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.TYPE<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigeru izteiksmē. Skatiet indeksētos makro.

{INVENTORY.TYPE.FULL}

Lauks *Tips (pilna informācija)* hosta inventarizācijā.

 Šo makro var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.TYPE.FULL<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigera izteiksmē. Skatiet indeksētos makro.

{INVENTORY.URL.A}

URL A lauks hosta inventarizācijā.

 Šo makro var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.URL.A<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigeru izteiksmē. Skatiet indeksētos makro.

{INVENTORY.URL.B}

URL B lauks hosta inventarizācijā.

 Šo makro var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.URL.B<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigeru izteiksmē. Skatiet indeksētos makro.

{INVENTORY.URL.C}

URL C lauks hosta inventarizācijā.

 Šo makro var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.URL.C<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigeru izteiksmē. Skatiet indeksētos makro.

{INVENTORY.VENDOR}

Ražotājs lauks hosta inventārā.

 Šo makro var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.VENDOR<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigera izteiksmē. Skatiet indeksētos makro.

Vienumu makrosi

{ITEM.DESCRIPTION}

Trigeri izteiksmē esošā N-tā vienuma apraksts, kas izraisīja paziņojumu.

 Šo makro var izmantot ar skaitlisku indeksu kā {ITEM.DESCRIPTION<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigeri izteiksmē. Skatiet indeksētos makro.

{ITEM.DESCRIPTION.ORIG}

Trigeri izteiksmē esošā N-tā vienuma sākotnējais apraksts (ar neatrisinātiem makro), kas izraisīja paziņojumu.

 Šo makro var izmantot ar skaitlisku indeksu kā {ITEM.DESCRIPTION.ORIG<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hosts trigeri izteiksmē. Skatiet indeksētos makro.

{ITEM.ID}

Trigeri izraisījušās paziņojuma izteiksmes N-tā vienuma skaitliskais ID.

 Šo makro var izmantot ar skaitlisku indeksu kā {ITEM.ID<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hosts trigera izteiksmē. Skatiet indexed macros.

{ITEM.KEY}

Trigerī esošā N-tā vienuma atslēga, kas izraisīja paziņojumu.
 Makro funkcijas šim makro netiek atbalstītas, ja tas tiek izmantots kā vietturis vēstures funkcijas pirmajā параметrā, piemēram, `last(/{HOST.HOST}/{ITEM.KEY})`.

 Šo makro var izmantot ar skaitlisku indeksu kā {ITEM.KEY<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigerī. Skatiet indeksētos makro.

{ITEM.KEY.ORIG}

Trigeri izteiksmē esošā N-tā vienuma sākotnējā atslēga (ar neatrisinātiem makro), kas izraisīja paziņojumu.

 Šo makro var izmantot ar skaitlisku indeksu kā {ITEM.KEY.ORIG<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hosts trigeri izteiksmē. Skatiet indeksētos makro.

{ITEM.LASTVALUE}

Jaunākā N-tā vienuma vērtība trigeris izteiksmē, kas izraisīja paziņojumu.
 Tā tiks atrisināta uz *UNKNOWN* lietotāja saskarne, ja jaunākā vēstures vērtība ir savākta vairāk nekā pirms *Max history display period* laika (iestatīts izvēlnes sadaļā *Administration > General*).
 Ja to izmanto problēmas nosaukumā, makro, skatot problēmas notikumus, netiks atrisināts uz jaunāko vienuma vērtību; tā vietā tas saglabās vienuma vērtību no brīža, kad problēma notika.
Ja to izmanto paziņojumos, dažos gadījumos makro var netikt atrisināts uz jaunāko vienuma vērtību brīdī, kad trigeris nostrādāja. Piemēram, ja vienums ātri saņem divas vērtības, ”A” un ”B”, un trigeris nostrādā pie ”A”, paziņojumos kā jaunākā vērtība var tikt rādīta ”B” nelielas apstrādes aizkaves dēļ - jaunākā vienuma vērtība ir mainījusies starp brīdi, kad trigeris nostrādāja, un brīdi, kad paziņojums tika izveidots. Lai no tā izvairītos, varat izmantot {ITEM.VALUE} makro, kas tiek atrisināts uz vērtību brīdī, kad trigeris nostrādā, nodrošinot, ka paziņojumā tiek izmantota pareizā vērtība.
 Tas ir `last(/{HOST.HOST}/{ITEM.KEY})` aizstājvārds.
 Atrisinātā teksta/logu vienumu vērtība lietotāja saskarne tiek saīsināta līdz 20 rakstzīmēm šādās vietās:
- Operatīvie dati;
- Trigeris apraksts;
- Trigeris URL;
- Trigeris URL etiķetes;
- Vienuma vērtības logrīka apraksts.
Lai atrisinātu pilnu vērtību, varat izmantot makro funkcijas, jo serveris nevienu vērtību nesaīsina. Piemēram: `{{ITEM.LASTVALUE}.regsub("(.*)", \1)}`.

Šo makro var izmantot ar skaitlisku indeksu, piemēram, {ITEM.LASTVALUE<1-9>}, lai norādītu uz pirmo, otro, trešo utt. vienumu trigeris izteiksmē. Skatiet indeksētos makro.

{ITEM.LASTVALUE.AGE}

Laiks, kas pagājis starp jaunākā vienuma vērtības savākšanu un makro novērtēšanu.
 Tiek attēlots cilvēkam saprotamā formātā (piem., 1m 45s).

 Šo makro var izmantot ar skaitlisku indeksu, piem., {ITEM.LASTVALUE.AGE<1-9>}, lai norādītu uz pirmo, otro, trešo utt. vienumu trigera izteiksmē. Skatiet indeksētos makro.

{ITEM.LASTVALUE.DATE}

Datums, kad tika iegūta jaunākā vienuma vērtība.
 Tiek attēlots `YYYYMMDD` formātā.
 Servera laika josla tiek izmantota trigeru nosaukumos (*Monitoring* > *Problems* sarakstā), notikumu nosaukumos, kā arī tagu nosaukumos un vērtībās. Visos citos 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 trigeru 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 (sarakstā *Monitoring* > *Problems*), notikumu nosaukumos, kā arī tagu nosaukumos un vērtībās. Visos citos gadījumos tiek izmantota lietotāja laika josla.

 Šo makro var izmantot ar skaitlisku indeksu, piemēram, {ITEM.LASTVALUE.TIME<1-9>}, lai norādītu uz pirmo, otro, trešo utt. vienumu trigeru izteiksmē. Skatiet indeksētos makro.

{ITEM.LASTVALUE.TIMESTAMP}

UNIX laikspiedols, kad tika iegūta jaunākā vienuma vērtība.

 Šo makro var izmantot ar skaitlisku indeksu, piemēram, {ITEM.LASTVALUE.TIMESTAMP<1-9>}, lai norādītu uz pirmo, otro, trešo utt. vienumu trigera izteiksmē. Skatiet indexed macros.

{ITEM.LOG.AGE}

Žurnāla notikuma vecums.
Ar precizitāti līdz sekundei.

 Šo makro var izmantot ar skaitlisku indeksu, piemēram, {ITEM.LOG.AGE<1-9>}, lai norādītu uz pirmo, otro, trešo utt. vienumu trigeris izteiksmē. Skatiet indeksētos makro.

{ITEM.LOG.DATE}

Datums, kad žurnāla ieraksts tika ierakstīts žurnālā.

 Šo makro var izmantot ar skaitlisku indeksu, piemēram, {ITEM.LOG.DATE<1-9>}, lai norādītu uz pirmo, otro, trešo utt. vienumu trigera izteiksmē. Skatiet indeksētos makro.

{ITEM.LOG.EVENTID}

Notikuma ID notikumu žurnālā.
 Tikai Windows notikumu žurnāla uzraudzībai.

 Šo makro var izmantot ar skaitlisku indeksu, piemēram, {ITEM.LOG.EVENTID<1-9>}, lai norādītu uz pirmo, otro, trešo utt. vienumu trigeris izteiksmē. Skatiet indeksētos makro.

{ITEM.LOG.NSEVERITY}

Notikuma skaitliskais smaguma līmenis notikumu žurnālā.
 Tikai Windows notikumu žurnāla uzraudzībai.

 Šo makro var izmantot ar skaitlisku indeksu, piemēram, {ITEM.LOG.NSEVERITY<1-9>}, lai norādītu uz pirmo, otro, trešo utt. vienumu trigeris izteiksmē. Skatiet indeksētos makro.

{ITEM.LOG.SEVERITY}

Notikuma verbālais smagums notikumu žurnālā.
 Tikai Windows notikumu žurnāla uzraudzībai.

 Šo makro var izmantot ar skaitlisku indeksu, piemēram, {ITEM.LOG.SEVERITY<1-9>}, lai norādītu uz pirmo, otro, trešo utt. vienumu trigera izteiksmē. Skatiet indeksētos makro.

{ITEM.LOG.SOURCE}

Notikuma avots notikumu žurnālā.
 Tikai Windows notikumu žurnāla uzraudzībai.

 Šo makro var izmantot ar skaitlisku indeksu, piemēram, {ITEM.LOG.SOURCE<1-9>}, lai norādītu uz pirmo, otro, trešo utt. vienumu trigera izteiksmē. Skatiet indeksētos makro.

{ITEM.LOG.TIME}

Laiks, kad žurnāla ieraksts tika ierakstīts žurnālā.

 Šo makro var izmantot ar skaitlisku indeksu, piemēram, {ITEM.LOG.TIME<1-9>}, lai norādītu uz pirmo, otro, trešo utt. vienumu trigera izteiksmē. Skatiet indeksētos makro.

{ITEM.LOG.TIMESTAMP}

UNIX laikspiedols, kad žurnāla ieraksts tika ierakstīts žurnālā.

 Šo makro var izmantot ar skaitlisku indeksu, piemēram, {ITEM.LOG.TIMESTAMP<1-9>}, lai norādītu uz pirmo, otro, trešo utt. vienumu trigera izteiksmē. Skatiet indeksētos makro.

{ITEM.NAME}

Vienuma nosaukums ar visām atrisinātajām makro vērtībām.

 Šo makro var izmantot ar skaitlisku indeksu, piemēram, {ITEM.NAME<1-9>}, lai norādītu uz pirmo, otro, trešo utt. vienumu trigera izteiksmē. Skatiet indeksētās makro.

{ITEM.NAME.ORIG}

Vienuma sākotnējais nosaukums (ar neatrisinātiem makro).

 Šo makro var izmantot ar skaitlisku indeksu, piemēram, {ITEM.NAME.ORIG<1-9>}, lai norādītu uz pirmo, otro, trešo utt. vienumu trigera izteiksmē. Skatiet indeksētos makro.

{ITEM.STATE}

Pēdējais N-tā vienuma stāvoklis trigera izteiksmē, kas izraisīja paziņojumu.
 Iespējamās vērtības: *Not supported*, *Normal*.

 Šo makro var izmantot ar skaitlisku indeksu, piemēram, {ITEM.STATE<1-9>}, lai norādītu uz pirmo, otro, trešo utt. vienumu trigera izteiksmē. Skatiet indexed macros.

{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 uz vienu no šiem:

- Nth vienuma vēsturiskā (notikuma brīdī) vērtība trigeris izteiksmē, ja to izmanto trigeris stāvokļa maiņas kontekstā, piemēram, attēlojot notikumus vai sūtot paziņojumus;
- Nth vienuma jaunākā vērtība trigeris izteiksmē, ja to izmanto bez trigeris stāvokļa maiņas konteksta, piemēram, attēlojot trigeru sarakstu uznirstošajā atlases logā. Šajā gadījumā tas darbojas tāpat kā {ITEM.LASTVALUE}.

Pirmajā gadījumā tas tiks atrisināts uz *UNKNOWN*, ja vēstures vērtība jau ir dzēsta vai nekad nav tikusi saglabāta.
Otrajā gadījumā, un tikai lietotāja saskarne, tas tiks atrisināts uz *UNKNOWN*, ja jaunākā vēstures vērtība ir savākta vairāk nekā pirms *Max history display period* laika (iestatīts izvēlnes sadaļā *Administration > General*).
Teksta/logu vienumu atrisinātā vērtība tiek saīsināta līdz 20 rakstzīmēm lietotāja saskarne šādās vietās:
- Operatīvie dati;
- Trigeris apraksts;
- Trigeris URL;
- Trigeris URL etiķetes;
- Vienuma vērtības logrīka apraksts.
Lai atrisinātu pilnu vērtību, varat izmantot makro funkcijas, jo serveris vērtības netiek saīsinātas. 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 trigeris izteiksmē. Skatiet indeksētie makro.

{ITEM.VALUE.AGE}

Laiks, kas pagājis starp vienuma vērtības savākšanu un makro novērtēšanu.
 Tiek attēlots cilvēkam saprotamā formātā (piem., 1m 45s).

 Šo makro var izmantot ar skaitlisku indeksu, piem., {ITEM.VALUE.AGE<1-9>}, lai norādītu uz pirmo, otro, trešo utt. vienumu trigeris izteiksmē. Skatiet indexed macros.

{ITEM.VALUE.DATE}

Datums, kad tika savākta vienuma vērtība.
 Tiek attēlots `YYYYMMDD` formātā.
 Servera laika josla tiek izmantota trigeru nosaukumos (sarakstā *Monitoring* > *Problems*), notikumu nosaukumos, kā arī tagu nosaukumos un vērtībās. Visos citos gadījumos tiek izmantota lietotāja laika josla.

Šo makro var izmantot ar skaitlisku indeksu, piemēram, {ITEM.VALUE.DATE<1-9>}, lai norādītu uz pirmo, otro, trešo utt. vienumu trigera izteiksmē. Skatiet indeksētos makro.

{ITEM.VALUE.TIME}

Laiks, kad tika savākta vienuma vērtība.
 Tiek attēlots `HHMMSS` formātā.
 Servera laika josla tiek izmantota trigeru nosaukumos (*Monitoring* > *Problems* sarakstā), notikumu nosaukumos, kā arī tagu nosaukumos un vērtībās. Visos citos gadījumos tiek izmantota lietotāja laika josla.

Šo makro var izmantot ar skaitlisku indeksu, piemēram, {ITEM.VALUE.TIME<1-9>}, lai norādītu uz pirmo, otro, trešo utt. vienumu trigeru izteiksmē. Skatiet indeksētos makro.

{ITEM.VALUE.TIMESTAMP}

UNIX laikspiedogs, kad tika savākta vienuma vērtība.

 Šo makro var izmantot ar skaitlisku indeksu, piemēram, {ITEM.VALUE.TIMESTAMP<1-9>}, lai norādītu uz pirmo, otro, trešo utt. vienumu trigera izteiksmē. Skatiet indeksētos makro.

{ITEM.VALUETYPE}

Trigerī esošā N-tā vienuma vērtības tips, kas izraisīja paziņojumu.
 Iespējamās vērtības: *0* - skaitlisks peldošais punkts, *1* - rakstzīme, *2* - žurnāls, *3* - skaitlisks bez zīmes, *4* - teksts.

 Šo makro var izmantot ar skaitlisku indeksu, piemēram, {ITEM.VALUETYPE<1-9>}, lai norādītu uz pirmo, otro, trešo utt. vienumu trigerī. Skatiet indeksētos makro.

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}

Zemākā līmeņa atklāšanas noteikuma jaunākais stāvoklis.
 Iespējamās vērtības: *Not supported*, *Normal*.

 Šo makro var izmantot ar skaitlisku indeksu, piemēram, {LLDRULE.STATE<1-9>}, lai norādītu uz pirmo, otro, trešo utt. vienumu trigeris izteiksmē. Skatiet indexed macros.

{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 atrisināts kā viens no šiem:

- N-tā vienuma starpniekserveris trigeru izteiksmē (paziņojumos, kas balstīti uz trigeriem). Šeit var izmantot indeksētos makro;
- Starpniekserveris, kas izpildīja atklāšanu (atklāšanas paziņojumos). Šeit izmantojiet {PROXY.DESCRIPTION} bez indeksēšanas;
- Starpniekserveris, pie kura reģistrējās aktīvais aģents (autoregistrācijas paziņojumos). Šeit izmantojiet {PROXY.DESCRIPTION} bez indeksēšanas.

Šo makro var izmantot ar skaitlisku indeksu, piemēram, {PROXY.DESCRIPTION<1-9>}, lai norādītu uz pirmo, otro, trešo utt. vienumu trigeru izteiksmē. Skatiet indeksētos makro.

{PROXY.NAME}

Starpniekservera nosaukums.
 Atrisinās uz vienu no šiem:

- N-tā vienuma starpniekserveris trigeru izteiksmē (paziņojumos, kas balstīti uz trigeriem). Šeit var izmantot indeksētos makro;
- Starpniekserveris, kas izpildīja atklāšanu (atklāšanas paziņojumos). Šeit izmantojiet {PROXY.NAME}, bez indeksēšanas;
- Starpniekserveris, pie kura reģistrējās aktīvais aģents (autoreģistrācijas paziņojumos). Šeit izmantojiet {PROXY.NAME}, bez indeksēšanas.

Šo makro var izmantot ar skaitlisku indeksu, piemēram, {PROXY.NAME<1-9>}, lai norādītu uz pirmo, otro, trešo utt. vienumu trigeru izteiksmē. Skatiet indeksētos makro.

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 servisa notikumu tagu saraksts.
 Servisa notikumu tagus var definēt servisa konfigurācijas tagu sadaļā. Ja tagi nepastāv, tiek izvērsts tukšā virknē.

{SERVICE.TAGSJSON}

JSON masīvs, kas satur pakalpojuma notikumu tagu objektus.
 Pakalpojuma notikumu tagus var definēt pakalpojuma konfigurācijas tagu sadaļā. Ja tagu nav, tiek izvērsts tukšā masīvā.

{SERVICE.TAGS.<tag name>}

Pakalpojuma notikuma taga vērtība, uz kuru atsaucas taga nosaukums.
 Pakalpojuma notikuma tagus var definēt pakalpojuma konfigurācijas tagu sadaļā.
 Taga nosaukums, kas satur nealfanumeriskus simbolus (tostarp neangļu daudzbaitu UTF rakstzīmes), ir jāievieto dubultpēdiņās. Pēdiņas un slīpsvītras citētā taga nosaukumā ir jāekranē ar 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 novērtēts trigeris 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 trigeris 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 izvērtēts trigeris atkopšanas izteiksme.
 Uz vienumiem balstītās funkcijas tiek izvērtētas un aizstātas ar rezultātiem notikuma ģenerēšanas brīdī, savukārt visas pārējās funkcijas tiek attēlotas tā, kā tās ir uzrakstītas izteiksmē. Var izmantot trigeris 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

Indeksēto makrosu sintakse **{MACRO<1-9>}** var atsaukties tikai uz trigeris izteiksmes lauka *Expression* N-to vienumu, funkciju vai hostu:

- **{HOST.IP1}, {HOST.IP2}, {HOST.IP3}** tiek aizstāti ar pirmā, otrā un trešā hosta IP adresēm trigeris izteiksmē (ja tādas ir).
- **{ITEM.VALUE1}, {ITEM.VALUE2}, {ITEM.VALUE3}** tiek aizstāti ar pirmā, otrā un trešā vienuma vērtībām trigeris izteiksmē notikuma brīdī (ja tādas ir).
- **{FUNCTION.VALUE1}, {FUNCTION.VALUE2}, {FUNCTION.VALUE3}** tiek aizstāti ar pirmās, otrās un trešās uz vienumu balstītās funkcijas vērtībām notikuma brīdī (ja tādas ir).

Trigeru kontekstā indeksētie makrosi vienmēr attiecas uz trigeris konfigurācijas lauku *Expression*, nevis uz *Recovery expression*. Piemēram, atkopšanas notikumā **{ITEM.VALUE2}** tiks aizstāts ar otrā vienuma vērtību no problēmas izteiksmes atkopšanas brīdī.

Makross **{HOST.HOST<1-9>}** ir atbalstīts arī `{?func(/host/key,param)}` izteiksmes makrosā **grafiku nosaukumos**. Piemēram, `{?func(/{HOST.HOST2}/key,param)}` grafika nosaukumā tiks aizstāts ar otrā vienuma hostu grafikā.

Warning:
Indeksētie makrosi netiks aizstāti nevienā citā kontekstā, izņemot šeit minētos gadījumus. Citos kontekstos izmantojiet makrosus bez indeksa (**{HOST.HOST}, {HOST.IP}** utt.).

Atrašanās vietas atbalstītie lietotāja makrosi

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
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 makro var izmantot šādos laukos:

Atrašanās vieta	Vairāki makro/jaukšana ar tekstu[1]
Vienuma nosaukums	jā
Vienuma atslēgas parametri	jā[4]
Atjaunināšanas intervāls	nē
Pielāgotie intervāli	nē
Noildze (piejama atbalstītajiem vienumu tipiem)	nē
Saglabāt līdz (vēsturei un tendencēm)	nē
Apraksts *Aprēķinātais/apkopotais vienums*	jā

Atrašanās vieta	Vairāki makro/jaukšana ar tekstu[1]
Formula (izteiksmes konstantes un funkciju parametri; vienuma atslēgas parametri; (*tikai apkopotajam vienumam*) filtra nosacījumi (hosts grupas nosaukums un taga nosaukums))	jā
Datu bāzes pārraugs	
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 hosts	jā
JMX aģents	
JMX galapunkts	jā
Skripta vienums	
Parametru nosaukumi un vērtības	jā
Pārlūkprogrammas vienums	
Parametru nosaukumi un vērtības	jā
Telemetrijas vaicājuma vienums	
Laika nobīde	nē
Atpakaļskatīšanās ierobežojums	nē
Detalizācijas līmenis	nē
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 hosts	jā
Tagi[2]	
Tagu nosaukumi	jā

Atrašanās vieta	Vairāki makro/jaukšana ar tekstu[1]
Tagu vērtības	jā
Iepriekšējas apstrādes darbības	
Parametri (tostarp pielāgotie skripti)	jā
Pielāgotie kļūdu apstrādes parametri (lauki *Iestatīt vērtību* un *Iestatīt kļūdu*)	jā

Zema līmeņa atklāšana

Zema līmeņa atklāšanas kārtulā lietotāja makrosus var izmantot šādos laukos:

Atrašanās vieta		Vairāki makrosi/sajauk
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 uztvērējs		
	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

starpniekserveris konfigurācijā lietotāja makro var izmantot šādos laukos:

Location	Multiple macros/mix with text[1]
Address for active agents > *Port* (if the starpniekserveris belongs to a group)	no
Interface address and port (for passive starpniekserveris)	no
Timeouts for vienums types	no

Starpniekserveru grupas

starpniekserveru grupas konfigurācijā lietotāja makro var izmantot šajos laukos:

Location	Multiple macros/mix with text[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 ieraksta nosaukums	jā
Izvēlnes ieraksta 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

Atrašanās vieta		Vairāki makrosi/sajaukums ar tekstu[1]
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		
	Lietotājs	jā
	Parole	jā
	SSL sertifikāts	jā
	SSL atslēgas fails	jā
	SSL atslēgas parole	jā
Birkas[2]		
	Birku nosaukumi	jā
	Birku vērtības	jā

Citas vietas

Papildus šeit uzskaitītajām vietām lietotāja makro var izmantot šādos laukos:

Vieta	Vairāki makro/jaukšana ar tekstu[1]
Globālie skripti (URL, skripts, SSH, Telnet, IPMI), tostarp apstiprinājuma teksts	yes
Webhook	
JavaScript skripts	no
JavaScript skripta parametra nosaukums	no
JavaScript skripta parametra vērtība	yes
Informācijas paneļi	
Datu tipa *Teksts* kolonna *Top hosts* informācijas paneļa logrīkā	yes
Apraksts parametru *Vienuma vērtība* un *Mērītājs* informācijas paneļa logrīkā	yes
Primārās/sekundārās etiķetes *Teksts* parametru *Honeycomb* informācijas paneļa logrīkā	yes
URL[3] parametru *URL* informācijas paneļa logrīkā	yes
Uzraudzība > *Problēmas*, *Hosts*, *Jaunākie dati*	

Vieta	Vairāki makro/jaukšana ar tekstu[1]
Pielāgots teksts kolonna *Datu vākšana* > *Vei- dnes*, *Hosts*	yes
Pielāgots teksts kolonna *Brīdinājumi* > *Mediju tipi* > *Ziņo- jumu vei- dnes*	yes
Tēma	yes
Ziņojums *Brīdinājumi* > *Mediju tipi* > *Skripts*	yes
Skripta parametri *Brīdinājumi* > *Mediju tipi* > *Mediju tips*	yes
Lietotājvārda un *Paroles* lauki *E-pasts* mediju tipam (ja *Autentifikācija* ir iestatīta uz ”Lietotājvārds un parole”; ieteicami slepenie makro) *Lietotāji* > *Li- etotāji* > *Mediji*	yes
Kad aktīvs *Administrēšana* > *Vis- pārīgi* > *GUI*	yes
Darba laiks *Administrēšana* > *Vis- pārīgi* > *Noildzes*	no
Vienumu tipu noildzes	no

Vieta	Vairāki makro/jaukšana ar tekstu[1]
Administrēšana > *Vispārīgi* > *Savienotāji*	
URL	yes
Lietotājvārds	yes
Parole	yes
Bearer token	yes
Noildze	no
HTTP starpniekserveris	yes
SSL sertifikāta fails	yes
SSL atslēgas fails	yes
SSL atslēgas parole	yes

Pilnu visu Zabbix atbalstīto makro sarakstu skatiet atbalstītie makro.

Piezīmes

[1] Ja konkrētajā vietā netiek atbalstīti vairāki makrosi vienā laukā vai makrosi kopā ar tekstu, vienam makrosam ir jāaizpilda viss lauks.

[2] Makrosi, kas izmantoti tagu nosaukumos un vērtībās, tiek atrisināti tikai notikuma ģenerēšanas procesa laikā.

[3] URL, kas satur slepeno makrosu, nedarbosies, jo tajos esošais makross tiks atrisināts kā "******".

[4] Makrosi netiek atbalstīti Zabbix trapper vienumu item key.

Versijas saderība

Atbalstītie aģenti

Lai būtu saderīgs ar Zabbix 8.0, Zabbix aģents nedrīkst būt vecāks par 1.6. versiju un nedrīkst būt jaunāks par 8.0. versiju.

Iespējams, būs jāpārskata vecāku aģentu konfigurācija, jo daži parametri ir mainīti, piemēram, parametri, kas saistīti ar reģistrēšanu versijās pirms 3.0.

Lai pilnībā izmantotu jaunākās funkcionalitātes, metriku, uzlabotas veiktspējas un samazināta atmiņas lietojuma priekšrocības, 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. Jaunākajās veidnēs tiek izmantoti veiktspējas skaitītāji angļu valodā, 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	Attālās komandas (piem., čaulas skripti); Tūlītējas vienuma vērtības pārbaudes (t.i., *Izpildīt tagad*); Piezīme: Priekšapstrādes testi ar reālu vērtību netiek atbalstīti.
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.

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
```

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 entītiju konfigurācijā atbalsta šādus laika sufiksus:
 s - sekundes *(ja tiek izmantots, darbojas tāpat kā neapstrādāta vērtība)*;
 m - minūtes;
 h - stundas;
 d - dienas;
 w - nedēļas;
 M - mēneši *(tikai tendences funkcijām)*;
 y - gadi *(tikai tendences funkcijām)*.

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 vērtības sufiksi

Sufiksus var izmantot arī, lai attēlotu skaitliskas vienuma vērtības cilvēkam saprotamā formātā.

Lai to iespējotu, konfigurējot vienumu, laukā *Units* izmantojiet vienu no šiem sufiksiem:
 B - baiti;
 Bps - baiti sekundē;
 s - sekundes, attēlotas, izmantojot līdz trim lielākajām nenulles laika vienībām;
 uptime - pagājušais laiks formātā hh:mm:ss vai N dienas, hh:mm:ss;
 unixtime - Unix laikspiedols, formatēts kā yyyy.mm.dd hh:mm:ss.

Tālāk minētie papildu noteikumi attiecas uz to, kā šie sufiksi tiek interpretēti un attēloti:

- B un Bps gadījumā Zabbix izmanto bāzes-2 pārveidošanu (1K = 1024B), ievērojot JEDEC standartu.
- Citiem vienumiem (piemēram, Hz, W u. c.) Zabbix izmanto bāzes-10 pārveidošanu (1K = 1000).
- s (sekundes) gadījumā:
  - Formāts ietver yyy mmm ddd hhh mmm sss ms; tiek attēlotas tikai līdz trim lielākajām nenulles laika vienībām (piemēram, 1M 10d 4h).
  - Ja kāda 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, exa, zetta, yotta *(tie tiek piemēroti tikai lietotāja saskarnē).*

Lai novērstu vienību pārveidošanu, izmantojiet prefiksu ! (piemēram, !B vai !s).

Tālāk redzamie piemēri parāda, kā saņemtās vienuma vērtības tiek pārveidotas, pamatojoties uz 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 cieti kodēts vienību stoplists, kas sastāvēja no `ms`, `rpm`, `RPM`, `%`. Šis stoplists ir novecojis, tāpēc pareizais veids, kā novērst šādu vienību pārveidošanu, ir `!ms`, `!rpm`, `!RPM`, `!%`.

Komandas izpilde

Zabbix izmanto kopīgu funkcionalitāti ārējām pārbaudēm, lietotāja parametriem, system.run vienumiem, pielāgotiem brīdinājumu skriptiem, attālajām 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.

Iziešanas koda pārbaude

Iziešanas kodi tiek pārbaudīti, izmantojot šādus nosacījumus:

- Tikai pielāgotiem brīdinājumu skriptiem, attāliem komandām un lietotāja skriptiem, kas tiek izpildīti Zabbix serverī un Zabbix starpniekserverī.
- Jebkurš iziešanas kods, kas nav 0, tiek uzskatīts par izpildes kļūmi.
- Standarta kļūdu izvades un standarta izvades saturs neveiksmīgām izpildēm tiek savākts un ir pieejams lietotāja saskarnē (kur tiek parādīts izpildes rezultāts).
- Papildu žurnāla ieraksts var tikt izveidots attālām komandām, kas izpildītas Zabbix aģentā/starpniekserverī, iespējojot parametru LogRemoteCommands aģents/starpniekserveris.

Iespējamie lietotāja saskarnes ziņojumi un žurnāla ieraksti neveiksmīgām komandām/skriptiem:

- Standarta kļūdu izvades un standarta izvades saturs neveiksmīgām izpildēm (ja tāds ir).
- "Process exited with code: N." (ja izvade ir tukša un iziešanas kods nav 0).
- "Process killed by signal: N." (ja process tika pārtraukts ar signālu, tikai Linux).
- "Process terminated unexpectedly." (ja process tika pārtraukts nezināmu iemeslu dēļ).

Skatiet arī

- Ārējās pārbaudes
- Lietotāja parametri
- system.run vienumi
- Pielāgoti brīdinājumu skripti
- Attālās komandas
- Globālie skripti

Atkāpšanās 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.

Kas ir Zabbix

Zabbix izveidoja Alexei Vladishev, un pašlaik to aktīvi izstrādā un atbalsta Zabbix SIA.

Zabbix ir uzņēmumu līmeņa atvērtā pirmkoda izplatīts uzraudzības risinājums.

Zabbix ir programmatūra, kas uzrauga daudzus tīkla parametrus, kā arī serveru, virtuālo mašīnu, lietojumprogrammu, pakalpojumu, datubāzu, tīmekļa vietņu, mākoņa un citu komponentu stāvokli un integritāti. Zabbix izmanto elastīgu paziņojumu mehānismu, kas ļauj lietotājiem konfigurēt brīdinājumus, kuru pamatā ir e-pasts, praktiski jebkuram notikumam. Tas nodrošina ātru reakciju uz servera problēmām. Zabbix piedāvā lieliskas atskaišu veidošanas un datu vizualizācijas iespējas, kuru pamatā ir saglabātie dati. Tādēļ Zabbix ir ideāli piemērots kapacitātes plānošanai.

Zabbix atbalsta gan aptaujas, gan slazdošanu. Visām Zabbix atskaitēm un statistikai, kā arī konfigurācijas parametriem var piekļūt, izmantojot tīmekļa lietotāja saskarni. Tīmekļa lietotāja saskarne nodrošina iespēju novērtēt tīkla stāvokli un serveru darbspēju no jebkuras atrašanās vietas. Pareizi konfigurēts Zabbix var būtiski palīdzēt IT infrastruktūras uzraudzībā. Tas vienlīdz attiecas gan uz mazām organizācijām ar dažiem serveriem, gan uz lieliem uzņēmumiem ar daudziem serveriem.

Zabbix ir pieejams bez maksas. Zabbix ir izstrādāts un tiek izplatīts saskaņā ar AGPL-3.0 licenci. Tas nozīmē, ka tā pirmkods tiek brīvi izplatīts un ir pieejams plašai sabiedrībai.

Komercatbalsts ir pieejams, un to nodrošina Zabbix uzņēmums un tā partneri visā pasaulē.

Uzziniet vairāk par Zabbix funkcijām.

Zabbix lietotāji Daudzas dažāda lieluma organizācijas visā pasaulē paļaujas uz Zabbix kā uz galveno uzraudzības platformu.

Arhitektūra Zabbix sastāv no vairākiem galvenajiem programmatūras komponentiem. To pienākumi ir izklāstīti tālāk.

Serveris

Zabbix serveris ir centrālais komponents, kuram aģenti ziņo par pieejamības un integritātes informāciju, kā arī statistiku. Serveris ir centrālā krātuve, kurā tiek glabāti visi konfigurācijas, statistikas un darbības dati.

Datu bāzes krātuve

Visa konfigurācijas informācija, kā arī Zabbix savāktie dati tiek glabāti datu bāzē.

Lietotāja saskarne

Lai nodrošinātu ērtu piekļuvi Zabbix no jebkuras vietas un jebkuras platformas, tiek nodrošināta tīmekļa lietotāja saskarne. Šī saskarne ir daļa no Zabbix serveris, un parasti (bet ne obligāti) tā darbojas tajā pašā fiziskajā mašīnā, kurā darbojas serveris.

Starpniekserveris

Zabbix starpniekserveris var vākt veiktspējas un pieejamības datus Zabbix servera vārdā. Starpniekserveris ir neobligāta Zabbix izvietošanas sastāvdaļa; tomēr tas var būt ļoti noderīgs, lai sadalītu viena Zabbix servera slodzi.

Aģents

Zabbix aģenti tiek izvietoti uz uzraudzības mērķiem, lai aktīvi uzraudzītu lokālos resursus un lietojumprogrammas un nosūtītu savāktos datus uz Zabbix serveri. Kopš Zabbix 4.4 ir pieejami divu veidu aģenti: Zabbix aģents (viegls, atbalstīts daudzās platformās, rakstīts C valodā) un Zabbix aģents 2 (ļoti 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 ir jāizveido hosts. Savukārt, lai izveidotu trigeri, vispirms ir nepieciešams vienums. Lai izveidotu darbību, ir nepieciešams trigeris. Tādējādi, ja vēlaties saņemt brīdinājumu, ka jūsu CPU noslodze uz *Server X* ir pārāk augsta, vispirms ir jāizveido hosts ieraksts *Server X*, pēc tam vienums tā CPU uzraudzībai, tad trigeris, kas aktivizējas, ja CPU noslodze ir pārāk augsta, un visbeidzot darbība, kas nosūta jums e-pastu. Lai gan tas var šķist kā daudz soļu, izmantojot veidnes, tā patiesībā nav. Tomēr šādas uzbūves dēļ ir iespējams izveidot ļoti elastīgu konfigurāciju.

Funkcijas Zabbix ir ļoti integrēts tīkla uzraudzības risinājums, kas vienā paketē piedāvā daudzveidīgas funkcijas.

Datu vākšana

- pieejamības un veiktspējas pārbaudes
- atbalsts SNMP (gan trapping, gan polling), IPMI, JMX, VMware uzraudzībai
- pielāgotas pārbaudes
- vēlamo datu vākšana pielāgotos intervālos
- veic serveris/starpniekserveris un aģenti

Elastīgas sliekšņa definīcijas

- varat definēt ļoti elastīgus problēmu sliekšņus, ko sauc par trigeriem, atsaucoties uz vērtībām no aizmugursistēmas datubāzes

Ļoti konfigurējama brīdināšana

- paziņojumu nosūtīšanu var pielāgot eskalācijas grafikam, saņēmējam, multivides tipam
- paziņojumus var padarīt jēgpilnus un noderīgus, izmantojot makro mainīgos
- automātiskās darbības ietver attālinātas komandas

Reāllaika grafiku veidošana

- uzraudzītie vienumi tiek nekavējoties attēloti grafikā, 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, kas 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 uzturēšanas procedūra

Vienkārša konfigurēšana

- pievienojiet uzraudzītās ierīces kā hosts
- hosts tiek iekļauti uzraudzībā, tiklīdz tie ir datubāzē
- lietojiet veidnes uzraudzītajām ierīcēm

Veidņu izmantošana

- pārbaužu grupēšana veidnēs
- veidnes var mantot citas veidnes

Tīkla atklāšana

- tīkla ierīču automātiska atklāšana
- aģenta automātiska reģistrācija
- failu sistēmu, tīkla saskarņu un SNMP OID atklāšana

Ātra lietotāja saskarne

- tīmekļa lietotāja saskarne PHP
- pieejama no jebkuras vietas
- varat pārvietoties pa to, klikšķinot
- audita žurnāls

Zabbix API

- Zabbix API nodrošina programmējamu saskarni ar Zabbix masveida manipulācijām, trešo pušu programmatūras integrācijai un citiem mērķiem.

Atļauju sistēma

- droša lietotāju autentifikācija
- noteiktus lietotājus var ierobežot līdz noteiktiem skatiem

Pilnvērtīgs un viegli paplašināms aģents

- izvietots uzraudzības mērķos
- var tikt izvietots gan Linux, gan Windows

Binārie dēmoni

- rakstīti C valodā, lai nodrošinātu veiktspēju un nelielu atmiņas patēriņu
- viegli pārnesami

Sagatavots sarežģītām vidēm

- attālināta uzraudzība ir vienkārša, izmantojot Zabbix starpniekserveri

Glosārijs Šajā sadaļā varat uzzināt dažu Zabbix bieži izmantotu terminu nozīmi.

hosts

- jebkura fiziska vai virtuāla ierīce, lietojumprogramma, pakalpojums vai jebkura cita loģiski saistīta uzraudzīto parametru kopa.

hosts grupa

- loģiska hostu grupa. Hostu grupas tiek izmantotas, piešķirot piekļuves tiesības hostiem dažādām lietotāju grupām.

vienums

- konkrēts datu elements, 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 vienumos saņemto datu "novērtēšanai".

Kad saņemtie dati pārsniedz slieksni, trigeri no stāvokļa *Ok* pāriet stāvoklī *Problem*. Kad saņemtie dati ir zem sliekšņa, trigeri paliek stāvoklī *Ok* vai atgriežas tajā.

veidne

- entītiju kopa (vienumi, trigeri, grafiki, zema līmeņa atklāšanas noteikumi, tīmekļa scenāriji), kas ir gatava lietošanai vienā vai vairākos hostos.

Veidņu uzdevums ir paātrināt uzraudzības uzdevumu ieviešanu hostā, kā arī atvieglot masveida izmaiņu piemērošanu uzraudzības uzdevumiem. Veidnes tiek tieši piesaistītas atsevišķiem hostiem.

veidņu grupa

- loģiska veidņu grupa. Veidņu grupas tiek izmantotas, piešķirot piekļuves tiesības veidnēm dažādām lietotāju grupām.

notikums

- vienreizēja parādība, kurai jāpievērš uzmanība, piemēram, trigera stāvokļa maiņa vai atklāšanas/aģenta automātiskās reģistrācijas veikšana.

notikuma tags

- iepriekš definēts notikuma marķieris. To var izmantot notikumu korelācijā, atļauju detalizētā pārvaldībā u. c.

notikumu korelācija

- metode problēmu elastīgai un precīzai sasaistīšanai ar to novēršanu.

Piemēram, varat definēt, ka viena trigera ziņoto problēmu var novērst cits trigeris, kas var izmantot pat citu datu apkopošanas metodi.

problēma

- trigeris, kas atrodas stāvoklī "Problem".

problēmas atjauninājums

- Zabbix nodrošinātās problēmu pārvaldības iespējas, piemēram, komentāra pievienošana, apstiprināšana, nopietnības maiņa vai manuāla aizvēršana.

darbība

- iepriekš definēts veids, kā reaģēt uz notikumu.

Darbība sastāv no operācijām (piemēram, paziņojuma nosūtīšanas) un nosacījumiem (*kad* operācija tiek veikta).

eskalācija

- pielāgots scenārijs operāciju izpildei darbības ietvaros; paziņojumu nosūtīšanas/attālo komandu izpildes secība.

makross

- mainīga vietturis, kas atkarībā no konteksta tiek aizstāts ar konkrētu vērtību.

multivide

- paziņojumu piegādes līdzeklis; piegādes kanāls.

paziņojums

- ziņojums par kādu notikumu, kas lietotājam nosūtīts, izmantojot izvēlēto multivides kanālu.

attālā komanda

- iepriekš definēta komanda, kas noteiktos apstākļos tiek automātiski izpildīta uzraudzītajā hostā.

tīmekļa scenārijs

- viens vai vairāki HTTP pieprasījumi tīmekļa vietnes pieejamības pārbaudei.

lietotāja saskarne

- Zabbix nodrošinātā tīmekļa saskarne.

informācijas panelis

- pielāgojama tīmekļa saskarnes sadaļa, kurā svarīgas informācijas kopsavilkumi un vizualizācijas tiek attēlotas 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 informācijas 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 citus pielāgotus uzdevumus.

Zabbix serveris

- Zabbix programmatūras centrālais process, kas veic uzraudzību, mijiedarbojas ar Zabbix starpniekserveriem un aģentiem, aprēķina trigerus, nosūta paziņojumus un kalpo kā centrālā datu krātuve.

Zabbix starpniekserveris

- process, kas var vākt datus Zabbix servera vārdā, samazinot servera apstrādes slodzi.

Zabbix aģents

- process, kas izvietots uzraudzības mērķos, lai aktīvi uzraudzītu lokālos resursus un lietojumprogrammas.

Zabbix aģents 2

- jaunās paaudzes Zabbix aģents lokālo resursu un lietojumprogrammu aktīvai uzraudzībai, kas ļauj izmantot pielāgotus spraudņus uzraudzībai.

> **Attention:**
> Tā kā Zabbix aģentam 2 ir daudz kopīgas funkcionalitātes ar Zabbix aģentu, dokumentācijā termins "Zabbix aģents" apzīmē gan Zabbix aģentu, gan Zabbix aģentu 2, ja funkcionālā darbība ir vienāda. Zabbix aģents 2 tiek īpaši nosaukts tikai tad, ja tā funkcionalitāte atšķiras.

šifrēšana

- šifrētu sakaru atbalsts starp Zabbix komponentiem (serveri, starpniekserveri, aģentu, zabbix_sender un zabbix_get utilītprogrammām), izmantojot Transport Layer Security (TLS) protokolu.

aģenta automātiskā reģistrācija

- automatizēts process, kurā pats Zabbix aģents tiek reģistrēts kā hosts un sāk uzraudzību.

tīkla atklāšana

- tīkla ierīču automatizēta atklāšana.

zema līmeņa atklāšana

- zema līmeņa entītiju automatizēta atklāšana konkrētā ierīcē (piemēram, failu sistēmu, tīkla saskarņu u. c.).

zema līmeņa atklāšanas noteikums

- definīciju kopa zema līmeņa entītiju automatizētai atklāšanai ierīcē.

vienuma prototips

- *metrika ar noteiktiem parametriem kā mainīgajiem, kas ir 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, kas ir gatavs zema līmeņa atklāšanai.* Pēc zema līmeņa atklāšanas mainīgie tiek automātiski aizstāti ar reāli atklātajiem parametriem, un trigeris automātiski sāk novērtēt datus.

Zema līmeņa atklāšanā tiek izmantoti arī citu Zabbix entītiju prototipi — grafiku prototipi, hostu prototipi un 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.

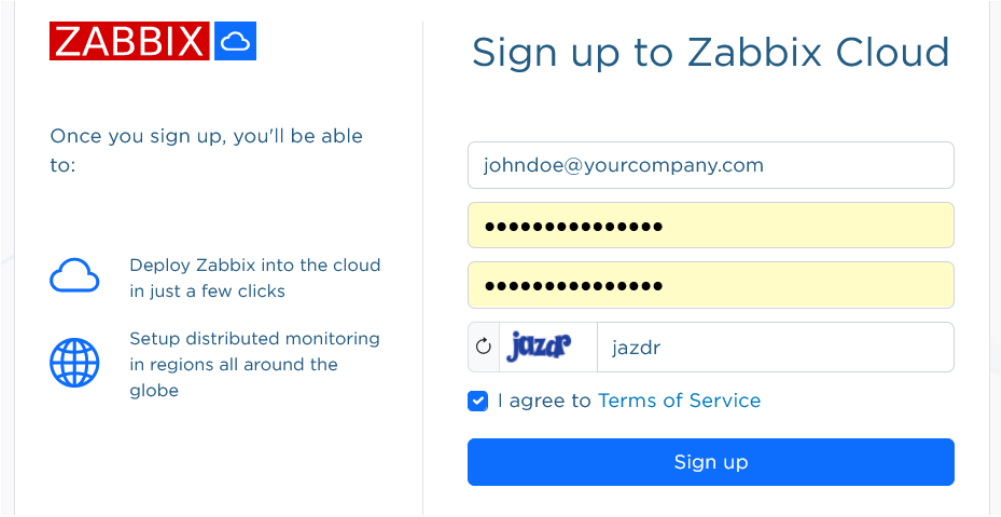
Zabbix izvietošana mākonī

Lai sāktu izmantot Zabbix mākonī, izpildiet tālāk norādītās darbības:

1. Piesakieties vai reģistrējieties vietnē cloud.zabbix.com.
2. Izveidojiet mākoņa mezglu.
3. Sāciet izmantot Zabbix mākonī.

Pieteikšanās vai reģistrēšanās Varat pieteikties vietnē cloud.zabbix.com, izmantojot esošu Google, GitHub vai Microsoft kontu.

Varat arī doties uz reģistrēšanās lapu, lai reģistrētos.



Pēc reģistrēšanās e-pastā saņemsiet jaunā konta apstiprinājumu.

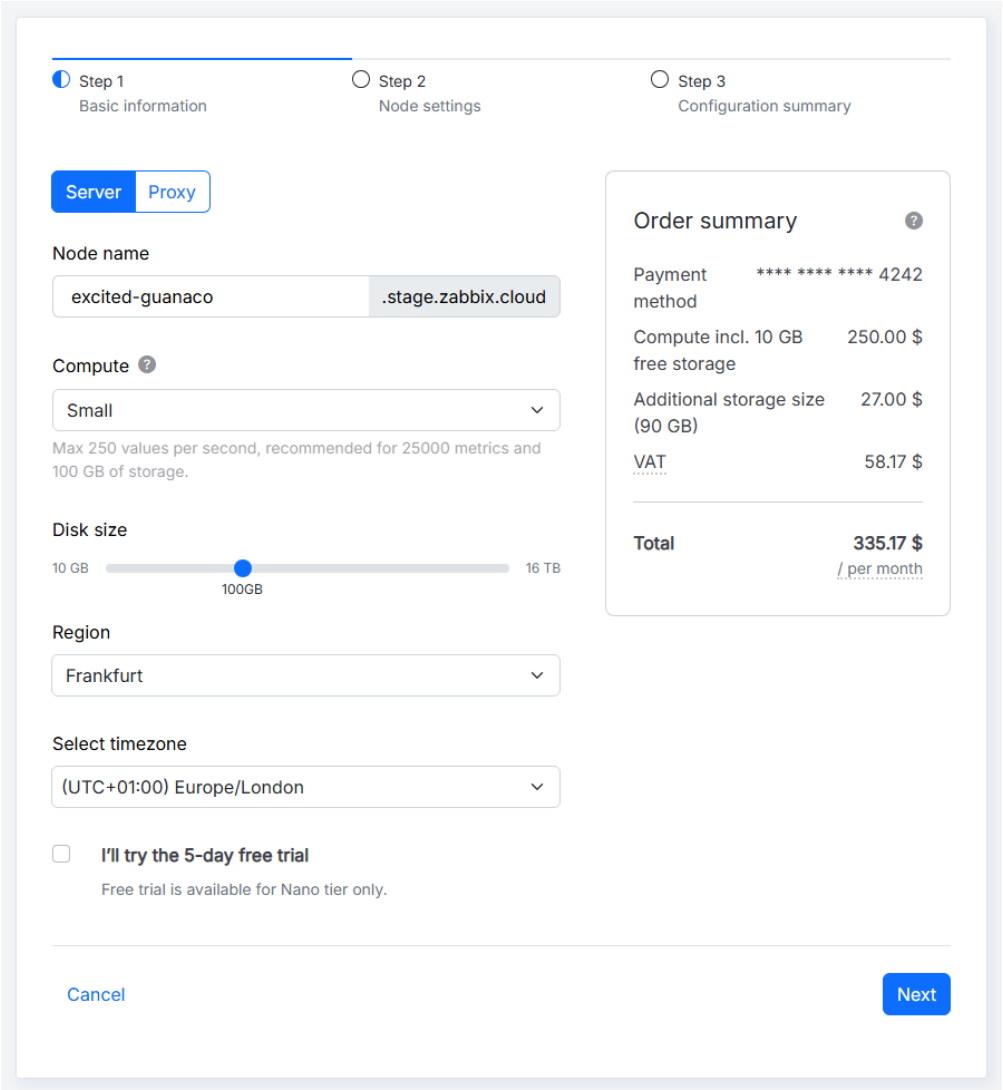
Lai pieteiktos, dodieties uz cloud.zabbix.com. Pieteikšanās laikā jums tiks lūgts ievadīt vienreizēju paroli. Šī parole jums tiks nosūtīta e-pastā.

Mākoņa mezgla izveide Mākoņa mezgls ir Zabbix instance, kas izvietota Zabbix Cloud. To var izvietot kā servera mezglu vai starpniekservera mezglu.

Pēc pieteikšanās redzēsiet, ka jūsu kontam pēc noklusējuma jau pastāv *Private* organizācija. Noklusējuma organizācijas nosaukumu vēlāk varēsiet mainīt.

Noklikšķiniet uz *Create New Node*. Tiks atvērta mezgla konfigurēšanas veidlapa, kas sastāv no trim darbībām.

Pirmajā darbībā atlasīsiet pamatinformāciju.



Parameter	Description
Server/Proxy	Atlasiet, vai izvietot Zabbix servera instanci vai Zabbix starpniekservera instanci.
Node name	Mezgla nosaukums veido unikālu apakšdomēnu formātā `<your-name>.zabbix.cloud`, kas tiek izmantots, lai piekļūtu mezgla pakalpojumiem.
Compute	Atlasiet mezglam nepieciešamo skaitļošanas jaudu. Lai noteiktu nepieciešamo resursu līmeni, skatiet cenu informāciju. Ja būs nepieciešama lielāka jauda, vēlāk vienmēr varēsiet veikt jauninášanu. Pirms netestēšanas mezgla izveides būs nepieciešams norēķinu veids. Norēķinu informāciju pievienojiet izvēlnē *Billing*. Lai izmēģinātu *Nano* līmeņa mezglu, ir pieejams **5 dienu bezmaksas izmēģinājuma periods**. Izmēģinājuma periodā maksa netiks iekasēta, taču pēc tā beigām mezgls tiks noņemts, ja netiks pārveidots par maksas mezglu.
Disk size	Slīdnis pēc noklusējuma ir iestatīts uz ieteicamo minimālo vērtību atbilstoši līmenim. Palieliniet to, ja glabājat lielu vēstures vai tendenču datu apjomu. Ņemiet vērā, ka lielāki diski rada lielākas ikmēneša izmaksas un ka diska izmēru pats samazināt nevarēsiet — lai to samazinātu, būs jāsazinās ar atbalsta dienestu.
Region	Atlasiet reģionu, kurā tiks izvietots mezgls.
Select timezone	Atlasiet Zabbix servera laika joslu. Šo laika joslu izmanto Zabbix serveris, un tā nav atkarīga no uzturēšanas un lietotāja profila laika joslām.

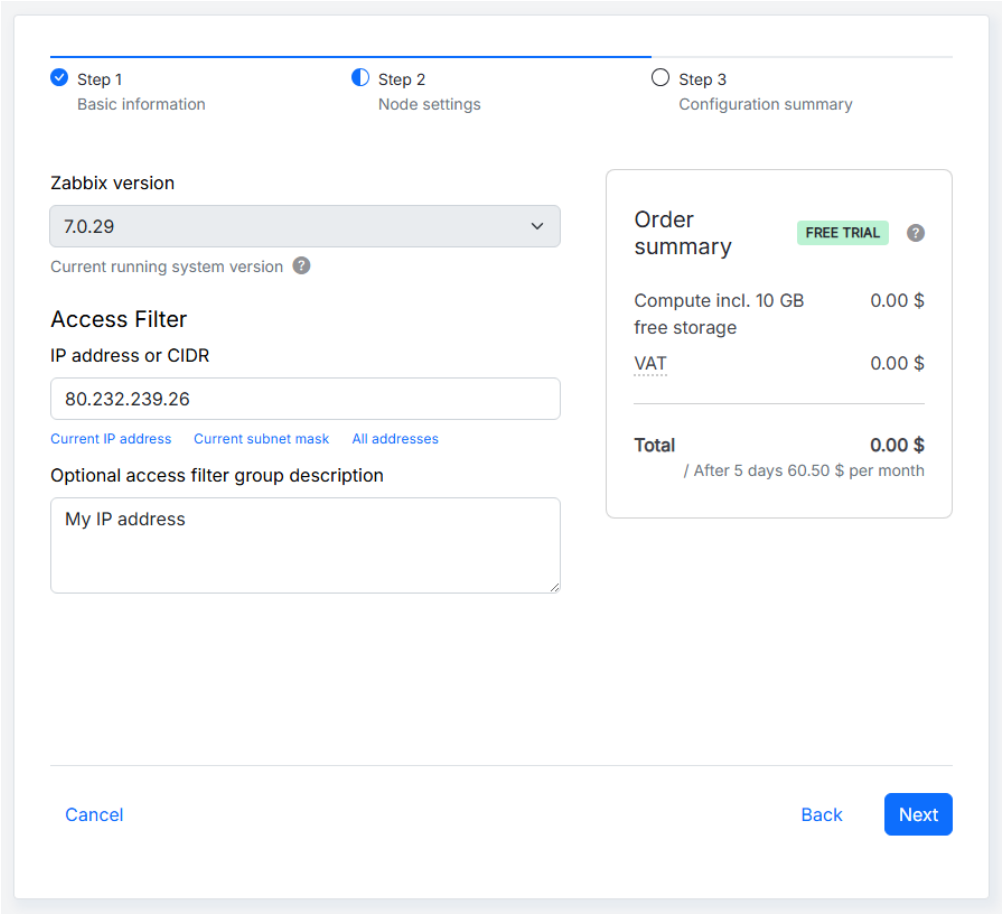
Pēc konfigurācijas pabeigšanas noklikšķiniet uz *Next*.

Otrajā darbībā atlasīsiet mezgla iestatījumus.

Otrās darbības konfigurācijas opcijas ir atkarīgas no tā, vai veidojat servera vai starpniekservera mezglu.

Servera mezgla konfigurēšana

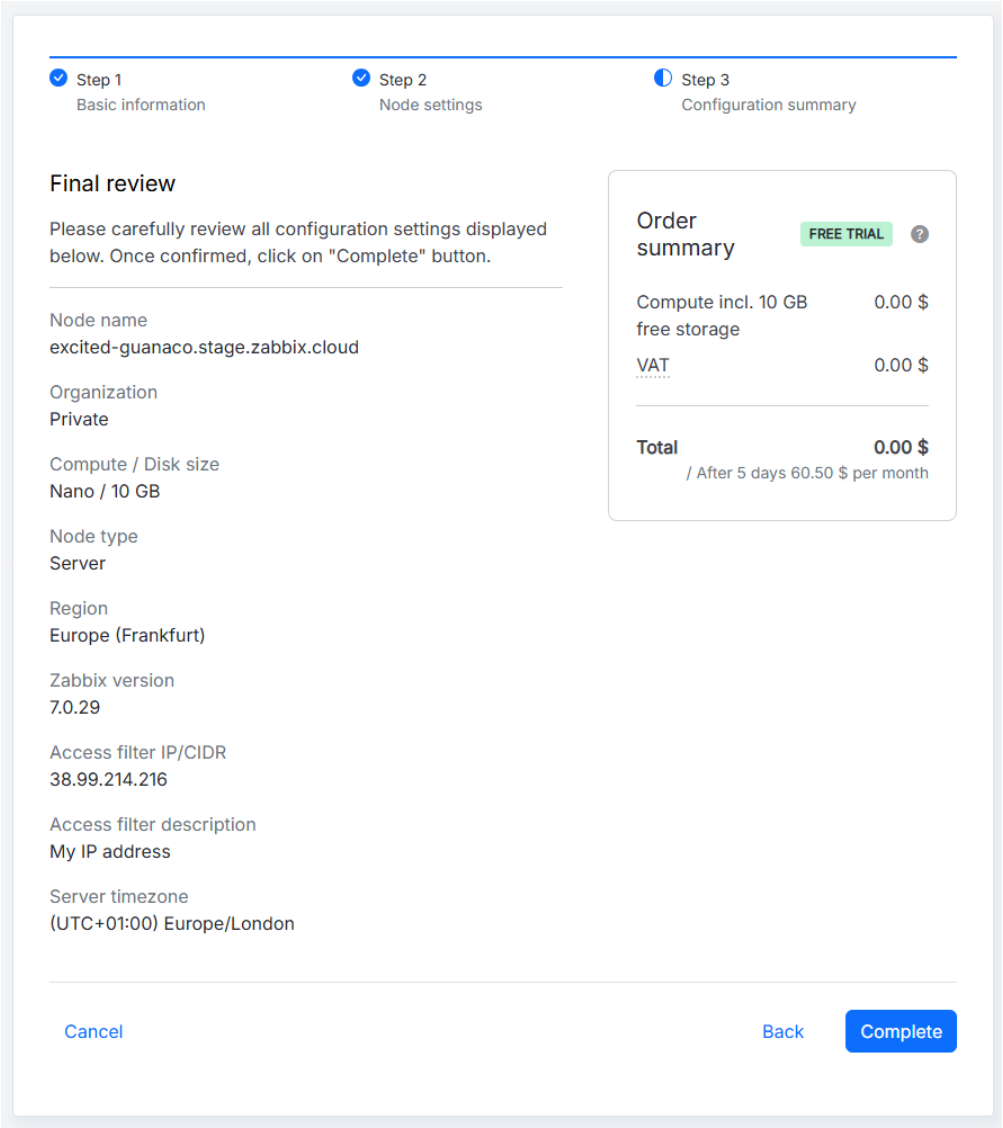
Servera mezgla izveides laikā ir pieejami šādi parametri:



Parameter	Description
Zabbix version	Atlasiet Zabbix versiju. Zabbix Cloud atbalsta jaunāko LTS versiju un automātiski lieto mazāko versiju atjauninājumus.
IP address or CIDR	Norādiet mezgla sākotnējo piekļuves filtru. Ievadiet IP adresi vai CIDR manuāli vai automātiski aizpildiet lauku, atlasot atbilstošo opciju. Servera mezgliem šis filtrs attiecas uz savienojumiem ar serveri, API un lietotāja saskarni. Filtru vēlāk var mainīt cilnē Access filters.
Optional access filter group description	Pievienojiet neobligātu piekļuves filtru grupas aprakstu, kurā var būt vairākas IP adreses vai CIDR.

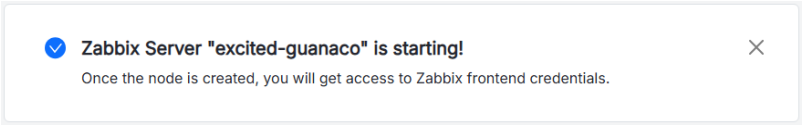
Warning:
Mezgla izveides laikā lauks *IP address or CIDR* tiek iepriekš aizpildīts ar IP adresi, kas izmantota piekļuvei Zabbix Cloud. Šo vērtību varat mainīt, lai norādītu citu sākotnējo piekļuves filtru. Pēc mezgla izveides piekļuves filtrus varat pievienot, rediģēt vai dzēst cilnē *Access filters*.

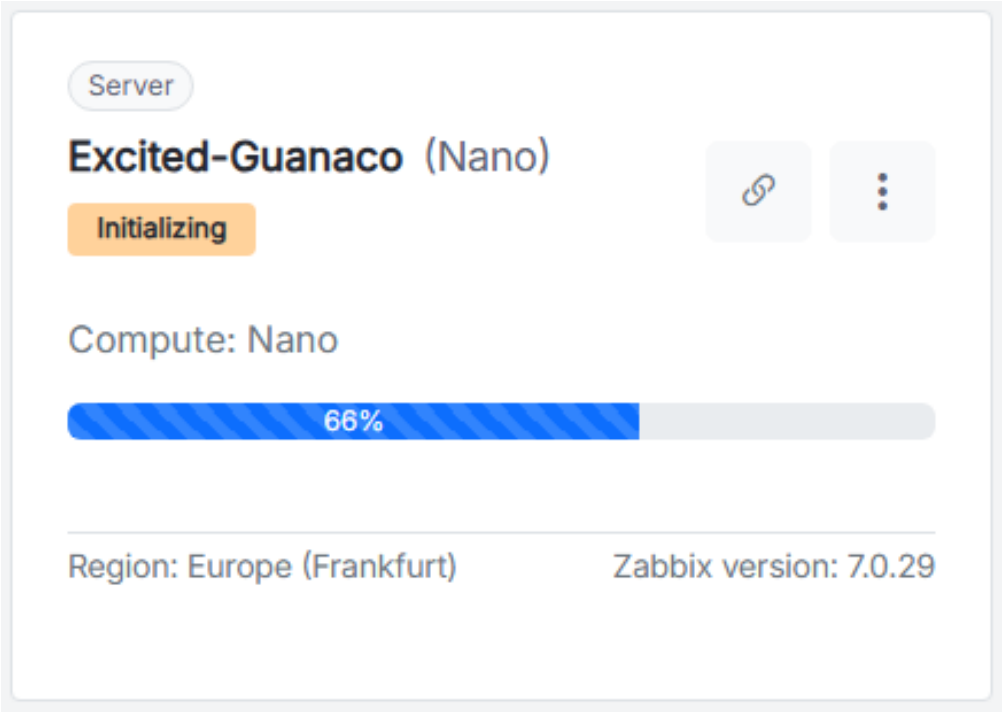
Trešajā darbībā tiek parādīts mezgla konfigurācijas kopsavilkums.



Pēc konfigurācijas iestatījumu pārskatīšanas un pārliecināšanās, ka viss ir pareizi, izmēģinājuma mezglam noklikšķiniet uz *Complete*, bet maksas mezglam — uz *Proceed to payment*. Pretējā gadījumā atgriezieties iepriekšējā darbībā, noklikšķinot uz pogas *Back*, vai noklikšķiniet uz *Cancel*, lai sāktu no sākuma.

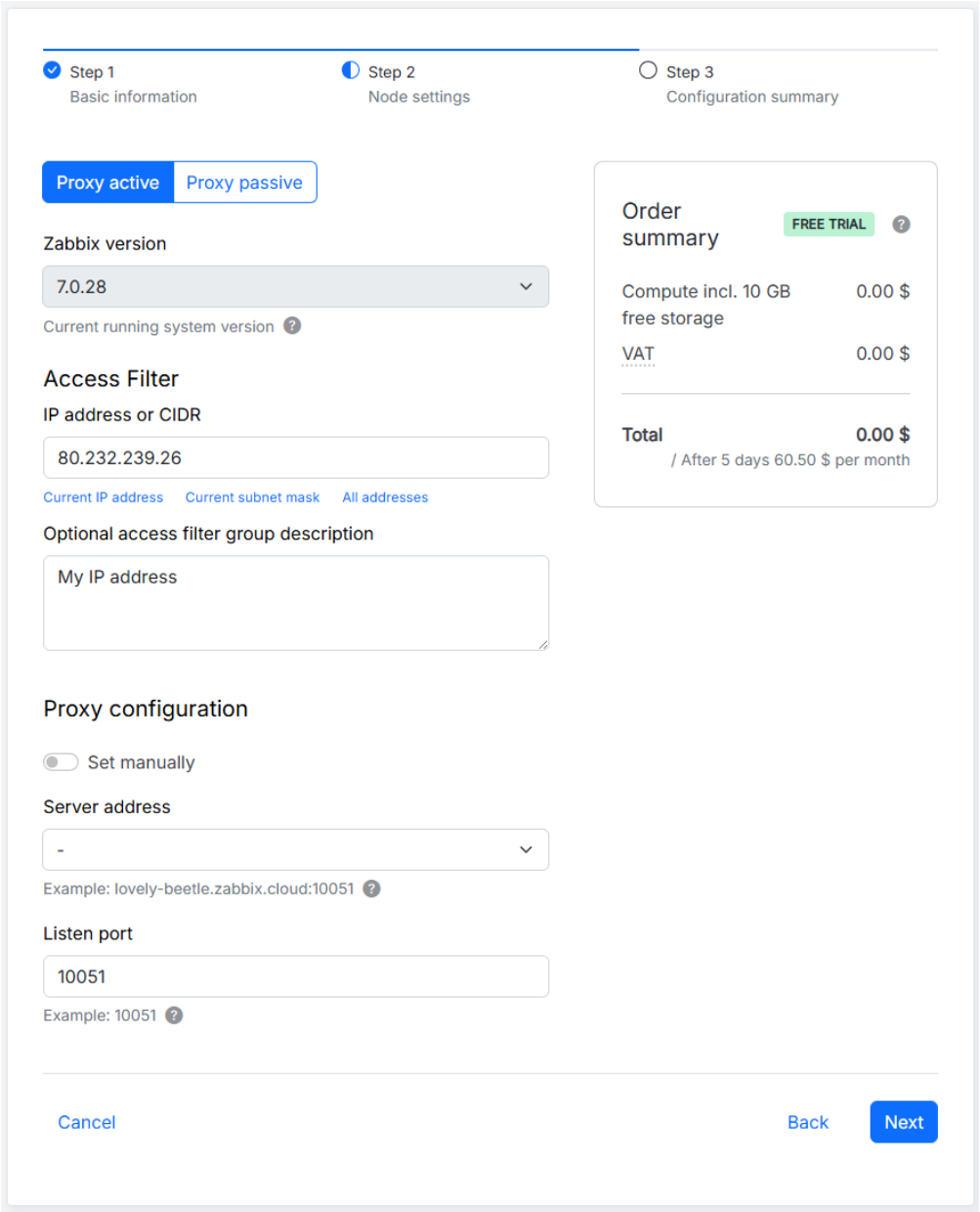
Kamēr mezgls tiek inicializēts, tiek parādīts atbilstošs ziņojums, un mezgls jau ir redzams sadaļā *Organizations*.





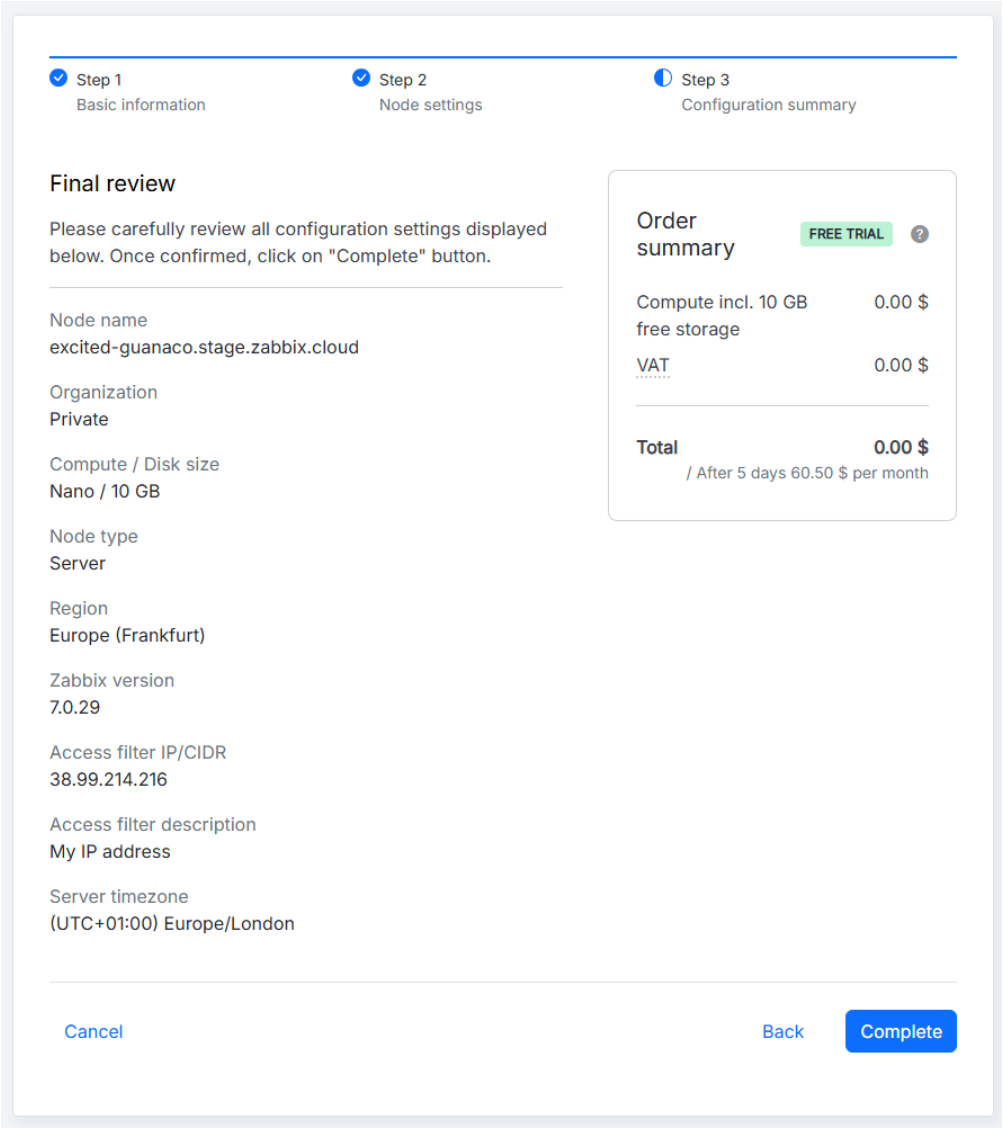
Starpniekservera mezgla konfigurēšana

Starpniekservera mezgla izveides laikā ir pieejami šādi parametri:



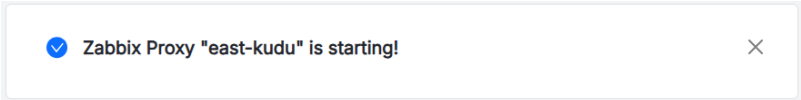
Parameter	Description
Proxy active/Proxy passive	Atlasiet starpniekservera režīmu. Aktīvajā režīmā starpniekserveris iniciē savienojumus ar Zabbix serveri; pasīvajā režīmā Zabbix serveris iniciē savienojumus ar starpniekserveri.
Zabbix version	Atlasiet Zabbix versiju. Zabbix Cloud atbalsta jaunāko LTS versiju un automātiski lieto mazāko versiju atjauninājumus.
IP address or CIDR	Norādiet sākotnējo ienākošās IP piekļuves filtru starpniekserverim. Ievadiet IP adresi vai CIDR manuāli vai automātiski aizpildiet lauku, atlasot atbilstošo opciju. Filtrs nosaka, kuriem tīkla pieprasījumiem ir atļauts sasniegt starpniekserveri, un to vēlāk var mainīt cilnē Access filters.
Optional access filter group description	Pievienojiet neobligātu piekļuves filtru grupas aprakstu, kurā var būt vairākas IP adreses vai CIDR.
Set manually	Atlasiet, kā norādīt Zabbix serveri. Ja opcija ir atspējota, atlasiet servera mezglu no jums pieejamo servera mezglu saraksta. Ja opcija ir iespējota, norādiet servera adreses manuāli.
Server address	Norādiet Zabbix servera adresi. Ja *Set manually* ir atspējota, atlasiet servera mezglu no saraksta. Ja opcija ir iespējota, norādiet līdz 10 servera adresēm. Aktīvam starpniekserverim norādiet Zabbix servera IP adresi vai DNS nosaukumu. Pasīvam starpniekserverim norādiet Zabbix servera IP adresi, DNS nosaukumu vai CIDR. Piekļuves filtri tiek konfigurēti atsevišķi un netiek automātiski atjaunināti, kad tiek norādīta servera adrese.
Listen port	Norādiet portu, kurā starpniekserveris klausās ienākošos savienojumus. Noklusējuma ports ir 10051. Klausīšanās portu pēc mezgla izveides nevar mainīt.

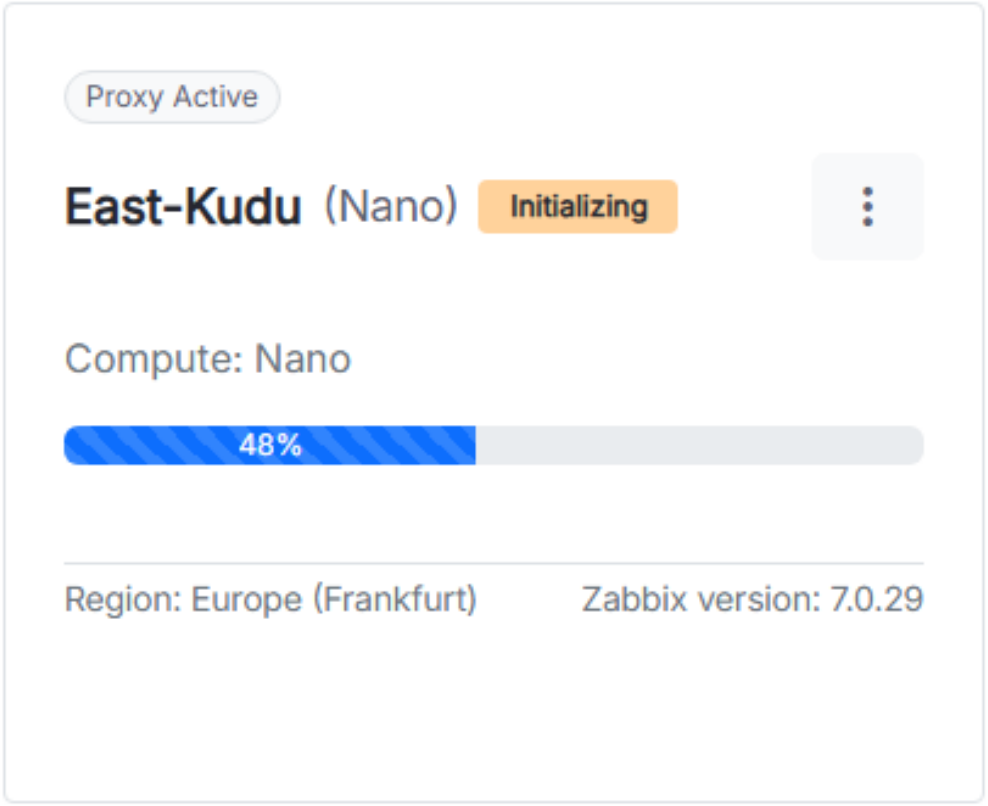
Trešajā darbībā redzēsiet mezgla konfigurācijas kopsavilkumu.



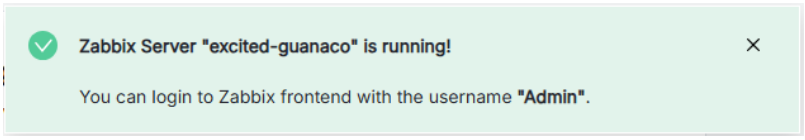
Pēc konfigurācijas iestatījumu pārskatīšanas un pārliecināšanās, ka viss ir pareizi, izmēģinājuma mezglam noklikšķiniet uz *Complete*, bet maksas mezglam — uz *Proceed to payment*. Pretējā gadījumā atgriezieties iepriekšējās darbībās, noklikšķinot uz pogas *Back*, vai noklikšķiniet uz *Cancel*, lai sāktu no sākuma.

Kamēr mezgls tiek inicializēts, tiek parādīts atbilstošs ziņojums, un mezgls jau ir redzams sadaļā *Organizations*.





Lai konfigurētu Zabbix Cloud starpniekserveri, skatiet Zabbix Cloud starpniekservera iestatīšana.

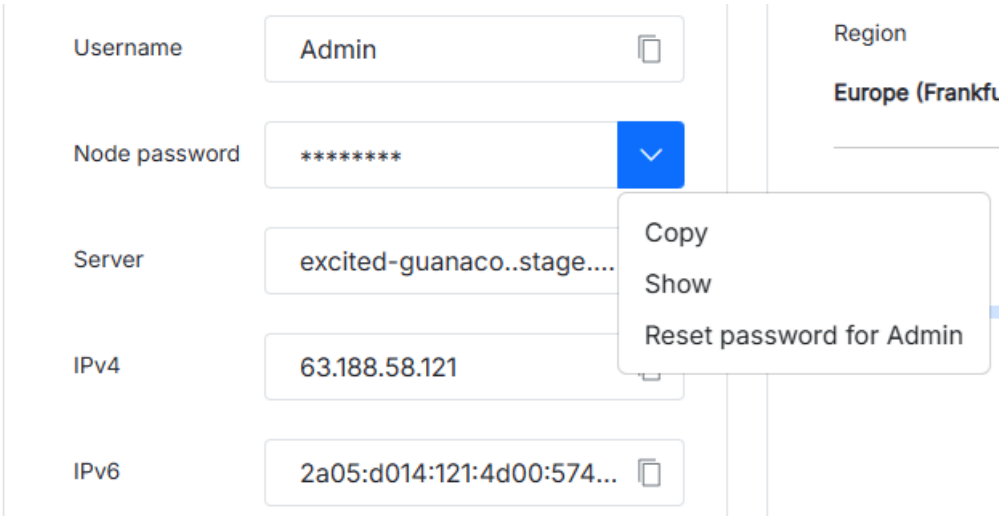


Sāciet izmantot Zabbix

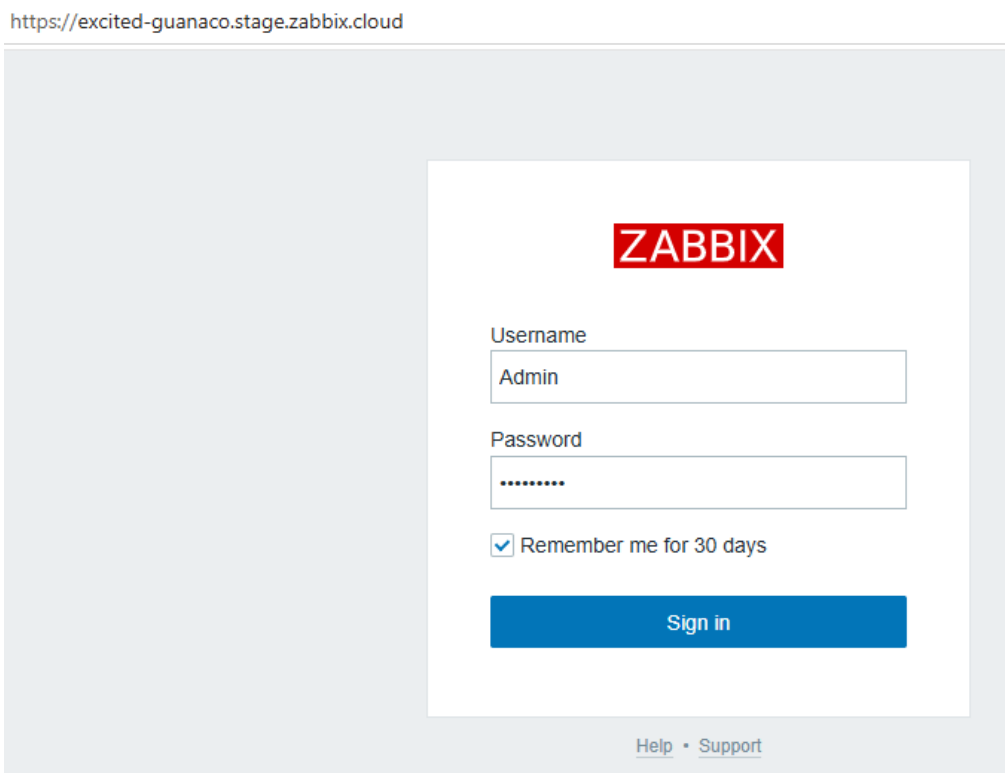
Kad servera mezgla statuss mainās no *Initializing* uz *Running*, varat pieteikties Zabbix lietotāja saskarnē.

Lai piekļūtu sākotnējai lietotāja saskarnes parolei, noklikšķiniet uz izveidotā mezgla.

Jūs tiksiet novirzīts uz mezgla iestatījumiem (pārskats, piekļuves filtri, šifrēšana, dublējumkopijas u. c.). Sadaļā **Connection info** atrodiet lauku *Node password* un nolaižamajā sarakstā atlasiet *Reset password for Admin*. Pēc paroles atiestatīšanas noklikšķiniet uz tā paša nolaižamā saraksta un atlasiet *Copy*.



Noklikšķiniet uz lietotāja saskarnes URL, lai atvērtu Zabbix lietotāja saskarni (`excited-guanaco.zabbix.cloud`).



Piesakieties, izmantojot šādus akreditācijas datus:

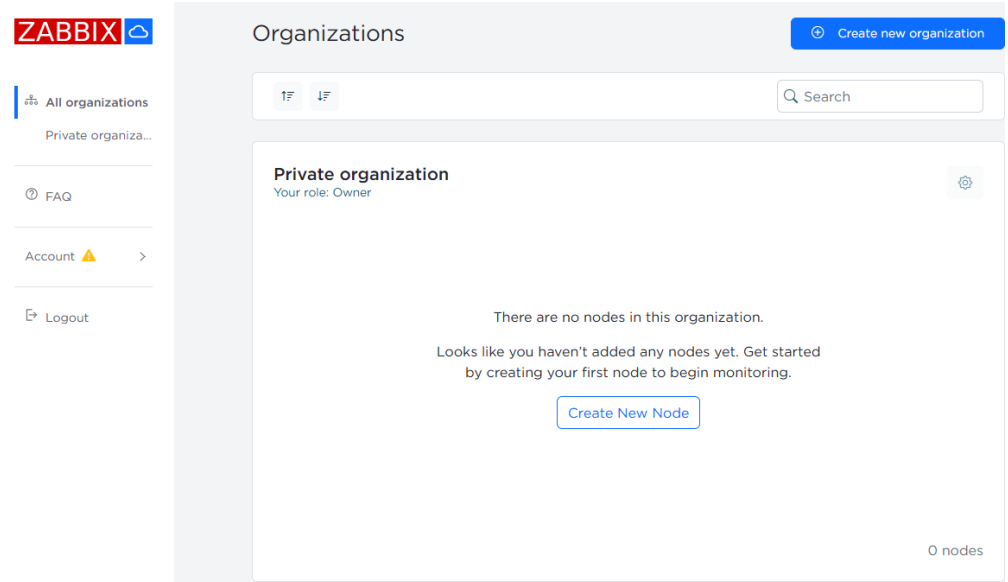
- *Username* - `Admin`
- *Password* - ievadiet nokopēto paroli

Lai sāktu uzraudzību, jādefinē piekļuves filtri - IP adrese vai adreses, kurām ir atļauts izveidot savienojumu ar Zabbix Cloud serveri. Jādefinē vismaz viena adrese.

Organizācijas pārvaldība Organizācija ir konteiners vienam vai vairākiem mezgliem. Organizācijas arī palīdz strukturēt jūsu resursus un deleģēt atbildību, piemēram:

- Mezglu pārvaldība
- Norēķini

Kad piesakāties Zabbix Cloud, tiek parādīta lapa *Visas organizācijas*. Pēc noklusējuma jūsu kontam tiek izveidota privāta organizācija.

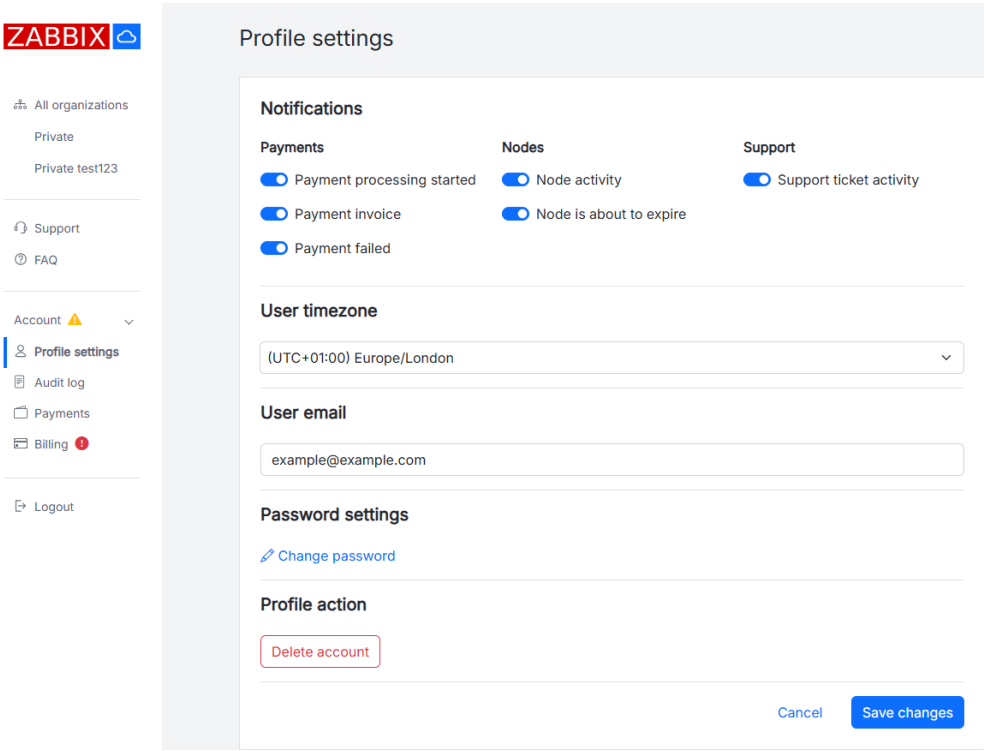


Kā šīs organizācijas īpašnieks jūs varat pārvaldīt lietotājus, administratorus, mezglus un organizācijas nosaukumu. Noklikšķiniet uz iestatījumu pogas, lai piekļūtu pārvaldības opcijām.

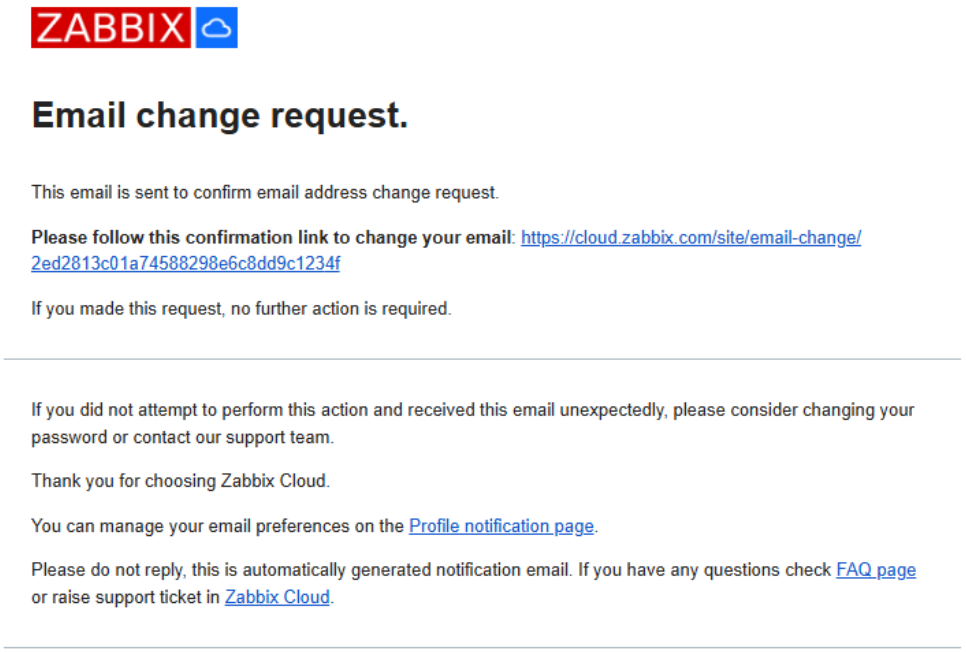
Kontu pārvaldība Konta e-pasta adreses maiņa

Lai mainītu konta e-pasta adresi, dodieties uz *Account* > *Profile settings*.

Laukā *User email* ievadiet jauno e-pasta adresi un noklikšķiniet uz *Save changes*.



Uz jauno e-pasta adresi tiek nosūtīts apstiprinājuma ziņojums. Lai apstiprinātu e-pasta adreses maiņu, sekojiet ziņojumā norādītajai saitei.



Apstiprinājuma saite atver lapu *Profile settings* ar apstiprinājuma ziņojumu un atjaunināto e-pasta adresi laukā User email.

Pēc izmaiņu apstiprināšanas uz jauno un iepriekšējo e-pasta adresi tiek nosūtīts e-pasta ziņojums, kas apstiprina izmaiņas. Uz iepriekšējo adresi nosūtītajā ziņojumā ir norādīta iepriekšējā un jaunā e-pasta adrese.

Mezglu konfigurācija

Pārskats

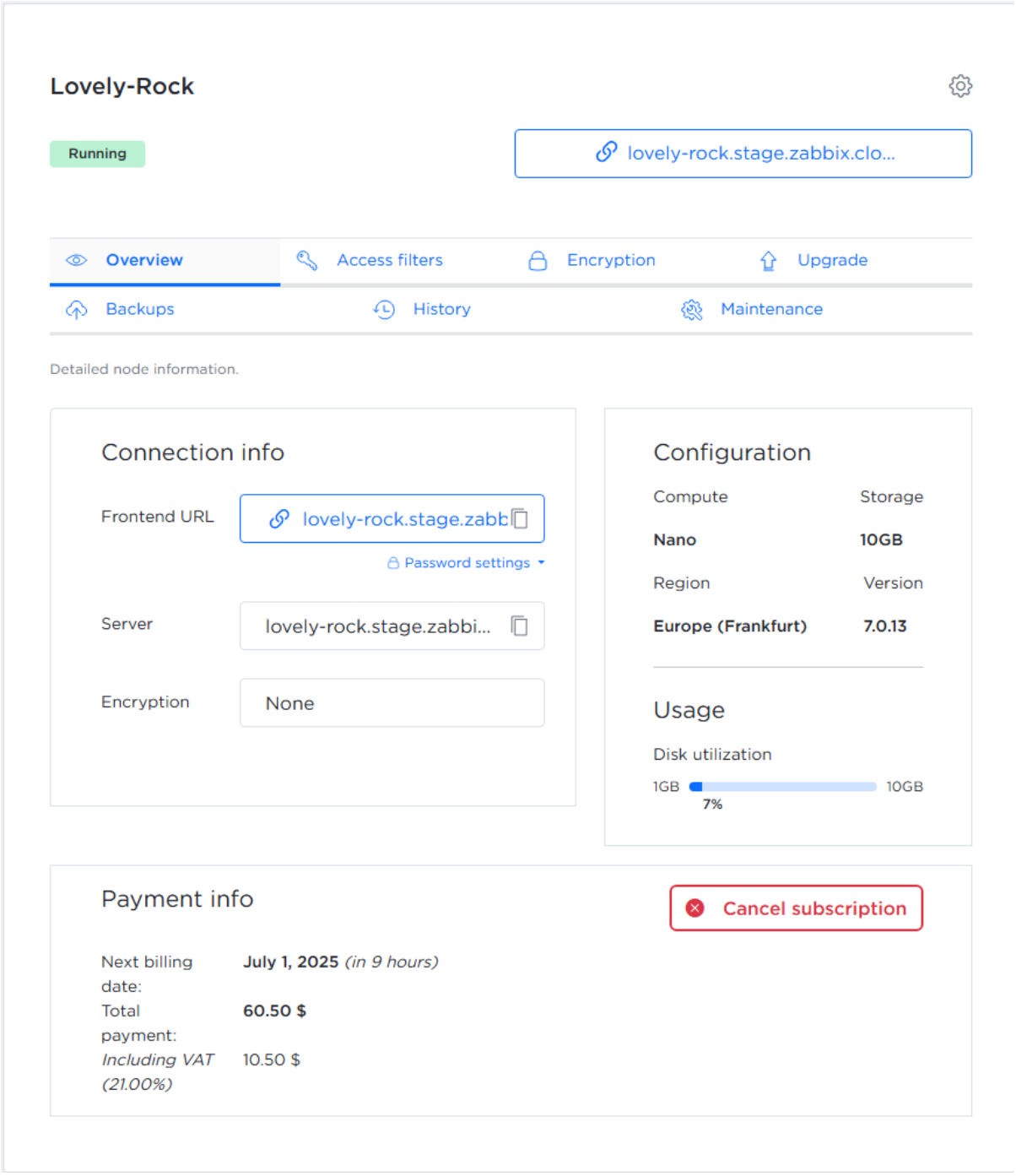
Mākoņa mezgls ir Zabbix komponents, kas izvietots mākonī. Servera mezgli ietver Zabbix serveri, datubāzi un lietotāja saskarni, savukārt starpniekservera mezgli ietver Zabbix starpniekserveri un datubāzi. Mākoņa mezglam ir piešķirti īpaši skaitļošanas resursi (CPU, atmiņa) un diska krātuve.

Lietotāji var izveidot vienu vai vairākus mākoņa mezglus. Mākoņa mezglus var grupēt **organizācijās** (pēc struktūras, reģiona u. c.), izveidojot izkliedētu uzraudzības iestatījumu.

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 uz saites *Mezgla konfigurācijas lapa*, kas tiek parādīta pēc mezgla palaišanas.

Mezgla konfigurācijas skatā katra cilne ļauj pārvaldīt konkrētu 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 neobligātos CRL.
- Dublējumkopijas - izveidot, atjaunot un dzēst mezgla momentuzņēmumus; skatīt nākamo ieplānoto dublējumkopiju.
- Vēsture - iestatīt audita žurnālu, vēstures un tendenču glabāšanas periodus, izmantojot iepriekš iestatītas mērvienības un atjaunināšanas ierobežojumu 3 reizes dienā.
- Apkope - definēt iknedēļas apkopes logus pēc laika joslas, dienas un stundas.
- Jaunināšana - mērogot skaitļošanas/krātuves resursus vai pārvaldīt izmēģinājuma versijas/maksājuma statusu.

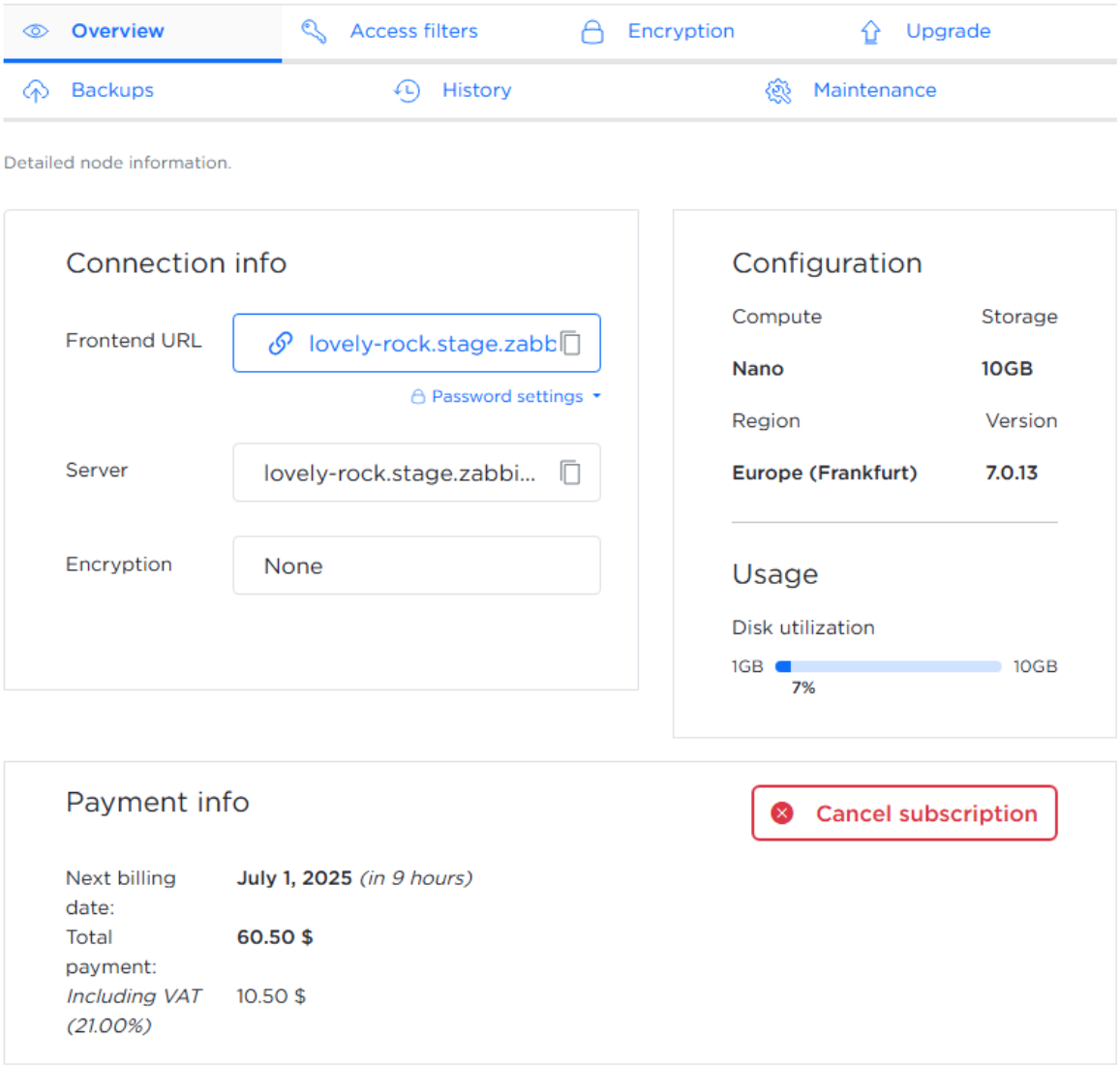


Note:
Dažas cilnes var nebūt pieejamas atkarībā no mezgla tipa (serveris vai starpniekserveris), lietotāja tipa, kurš skata lapu, un mezgla statusa (izmēģinājuma versija/bez izmēģinājuma versijas).

Cilne “Pārskats”

Cilnē *Pārskats* ir sniegta detalizēta informācija par mezglu. Parādītā informācija ir atkarīga no tā, vai mezgls ir serveris vai starpniekserveris.

Servera mezgls Servera mezgla cilnē *Pārskats* ir ietvertas šādas sadaļas:



Savienojuma informācija:

- Noklikšķināms lietotāja saskarnes URL ar iespēju to kopēt starpliktuvē
- Lietotājvārds
- Mezglam paredzētā parole un tās iestatījumi
- Servera resursdatora nosaukums ar iespēju to kopēt starpliktuvē
- IPv4 adrese ar iespēju to kopēt starpliktuvē
- IPv6 adrese ar iespēju to kopēt starpliktuvē
- Pašreizējā šifrēšanas metode

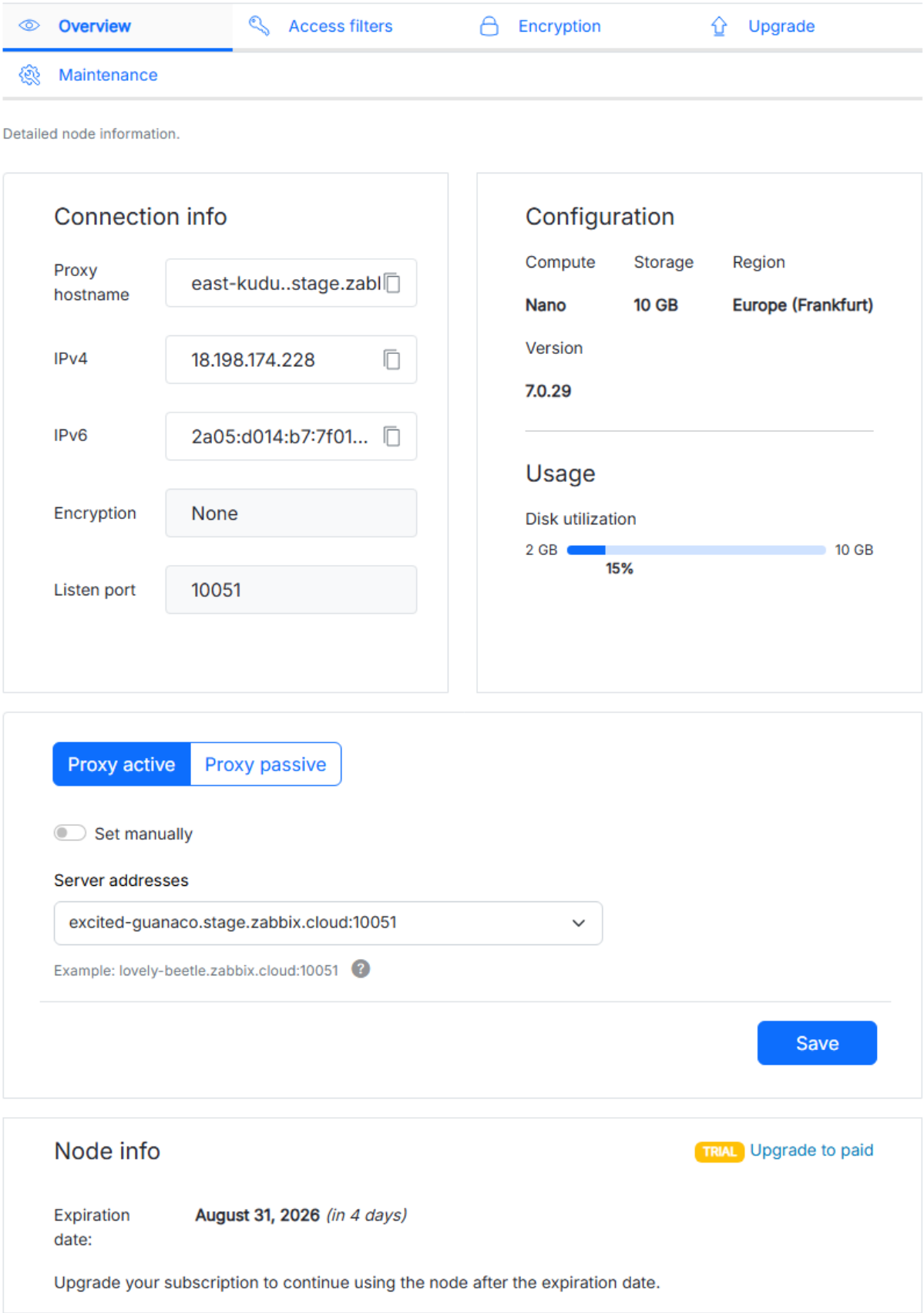
Konfigurācija un lietojums:

- Aprēķinu līmenis, maksimālā krātuve, reģions
- Zabbix komponentu versija
- Diska izmantojums

Mezgla informācija/Maksājumu informācija:

- Derīguma termiņš (izmēģinājuma un priekšapmaksas mezgliem)
- Pēc mezgla inicializācijas pabeigšanas atkarībā no mezgla veida tiek parādīta kontekstam atbilstoša poga: - *Jaunināt uz maksas plānu* - pieejama mezgliem, kuriem ir bezmaksas izmēģinājuma periods - *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

Starpniekservera mezgls Cilnē *Overview* starpniekservera mezglam ir pieejamas šādas sadaļas:



Savienojuma informācija:

- Starpniekservera resursdatora nosaukums ar iespēju to kopēt starpliktuvē
- IPv4 adrese ar iespēju to kopēt starpliktuvē
- IPv6 adrese ar iespēju to kopēt starpliktuvē
- Pašreizējā šifrēšanas metode
- Pašreizējais klausīšanās ports

Konfigurācija un lietojums:

- Skaitļošanas līmenis, maksimālā krātuve, reģions
- Zabbix komponentu versija
- Diska izmantojums

Starpniekservera režīms:

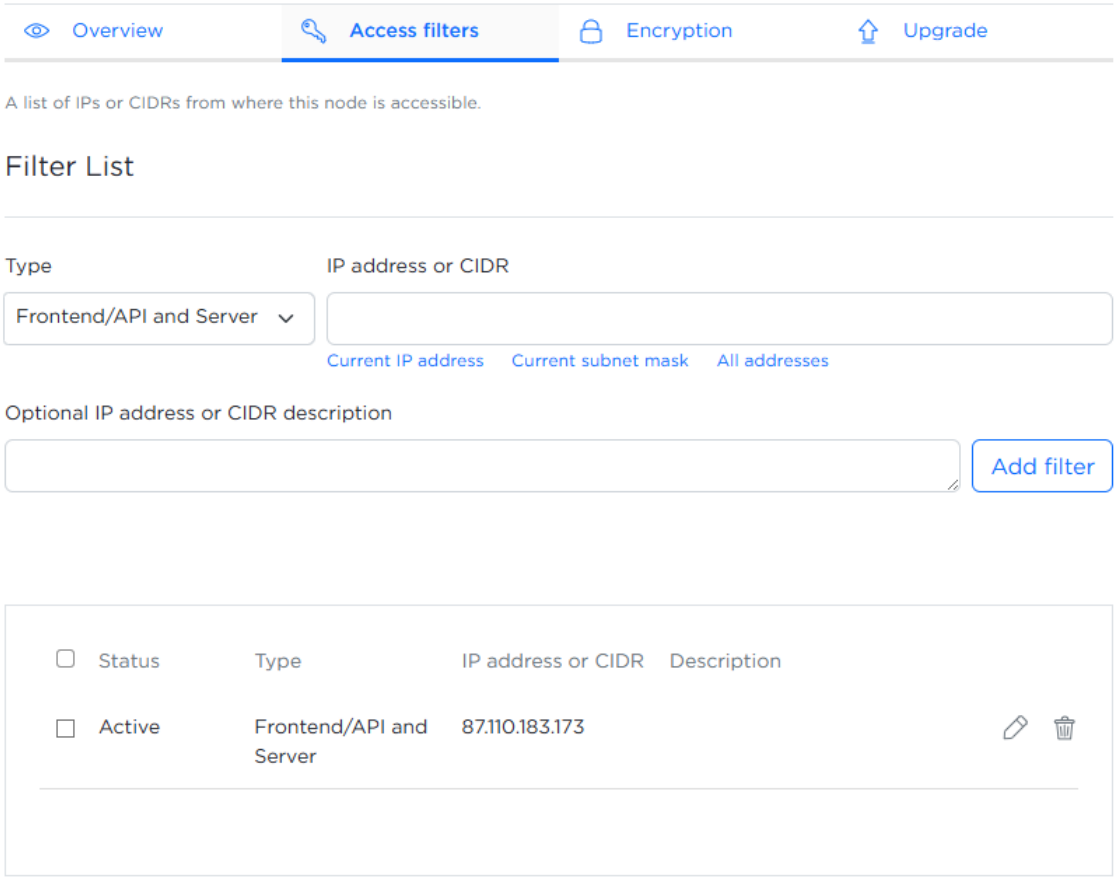
- Pārslēgšanās starp aktīvo un pasīvo starpniekservera režīmu
- Atlasiet servera adresi nolaižamajā sarakstā *Server addresses* vai iespējojiet *Set manually*, lai servera adresi ievadītu manuāli

Mezgla informācija/Maksājumu informācija:

- Derīguma termiņš (izmēģinājuma un priekšapmaksas mezgliem)
- Pēc mezgla inicializācijas pabeigšanas atkarībā no mezgla veida tiek parādīta kontekstam atbilstoša poga: - *Upgrade to paid* - pieejama mezgliem, kuriem ir bezmaksas izmēģinājuma periods - *Cancel subscription* - pieejama maksas mezgliem ar aktīvu abonementu - *Resubscribe to paid* - tiek parādīta, ja abonements ir atcelts - Priekšapmaksas mezgliem, kuriem ir ilgs derīguma termiņš un kurus nevar atcelt, darbības poga netiek parādīta

Piekļuves filtru cilne

Cilnē *Piekļuves filtri* varat ierobežot piekļuvi savam Zabbix mezglam pēc IP adreses vai apakštīkla.



Lai konfigurētu piekļuves filtru, aizpildiet tālāk norādītos laukus:

- *Tips* - servera mezgliem izmantojiet nolaižamo sarakstu, lai izvēlētos, kuru Zabbix komponentu ierobežot: tikai lietotāja saskarni/API, tikai serveri vai abus. Starpniekservera mezgliem ir pieejams tikai *Ienākošais IP*. Tas nosaka, kuriem ienākoša-jiem tīkla pieprasījumiem ir atļauts sasniegt starpniekserveri.
- *IP adrese vai CIDR* - ievadiet IP adresi vai CIDR manuāli vai automātiski aizpildiet lauku, 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 tālāk esošajā tabulā, kurā redzams statuss, tips, IP adrese vai apakštīkls un apraksts.

Warning:
Mezgla inicializācijas laikā IP adresei, kuru izmantojāt, tiek automātiski izveidots piekļuves filtrs. Ja vēlāk izveidojat savienojumu no citas IP adreses vai tīkla, šī adrese ir jāpievieno šeit, pretējā gadījumā piekļuve tiks bloķēta.

Varat:

-  - rediģēt filtru.
-  - dzēst filtru vai vairākus filtrus, izmantojot izvēles rūtiņas.

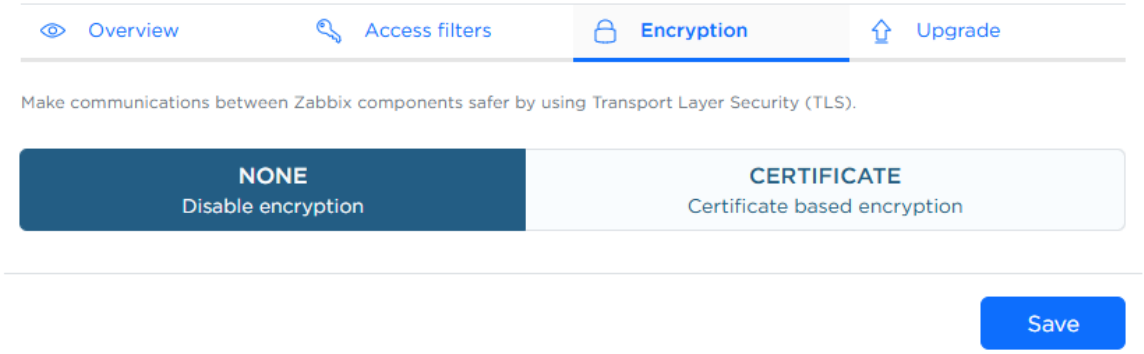
Šifrēšanas cilne

Cilnē *Šifrēšana* varat konfigurēt TLS šifrēšanu saziņai starp Zabbix komponentēm, augšupielādējot nepieciešamos sertifikātu failus.

Lai iespējotu drošu saziņu, izmantojot sertifikātus, atlasiet *CERTIFICATE* un augšupielādējiet no sava datora tālāk norādītos failus:

- *Root CA certificates* - uzticami CA sertifikāti, kas tiek izmantoti partneru sertifikātu pārbaudei.
  Noklikšķiniet uz *Choose file* un atlasiet failu, kurā ir uzticamie Root CA sertifikāti (parasti `.crt` vai `.pem` fails PEM formātā).
- *Certificate chain* - jūsu sertifikāta uzticamības ķēde.
  Noklikšķiniet uz *Choose file* un atlasiet PEM formātā kodētu failu (piemēram, `.crt` vai `.pem`), kurā ir jūsu sertifikāts un visi starpniekservera CA sertifikāti.
- *Private key* - privātā atslēga, kas atbilst jūsu sertifikātam.
  Noklikšķiniet uz *Choose file* un atlasiet PEM formātā kodētu `.key` failu.
- *Revoked certificates* (neobligāti) - ja uzturat sertifikātu atsaukšanas sarakstu (CRL), noklikšķiniet uz *Choose file* un atlasiet CRL failu, lai norādītu sertifikātus, kuriem vairs nevajadzētu uzticēties.

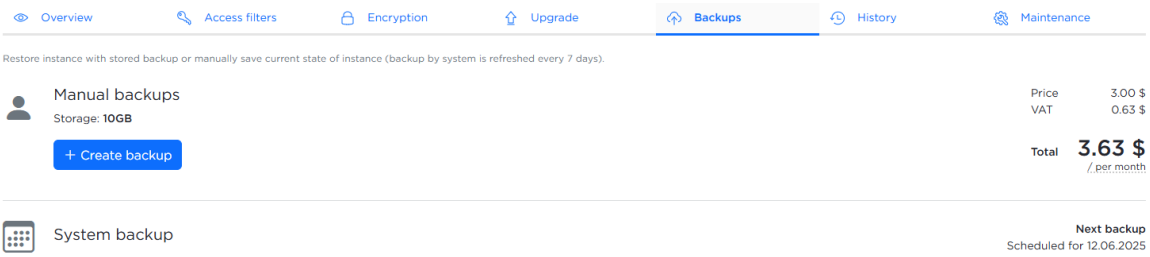
Pēc failu atlasīšanas noklikšķiniet uz *Save*.
Faili tiks augšupielādēti, validēti un izmantoti, lai konfigurētu TLS šifrēšanu starp Zabbix komponentēm.



Dublējumu cilne

Dublējumu cilnē varat skatīt, izveidot, atjaunot un dzēst sava Zabbix mezgla dublējumus.

Varat atjaunot savu instanci no jebkura saglabāta dublējuma vai jebkurā laikā izveidot jaunu manuālu dublējumu.
Cilnē tiek parādīts arī nākamā ieplānotā dublējuma datums.



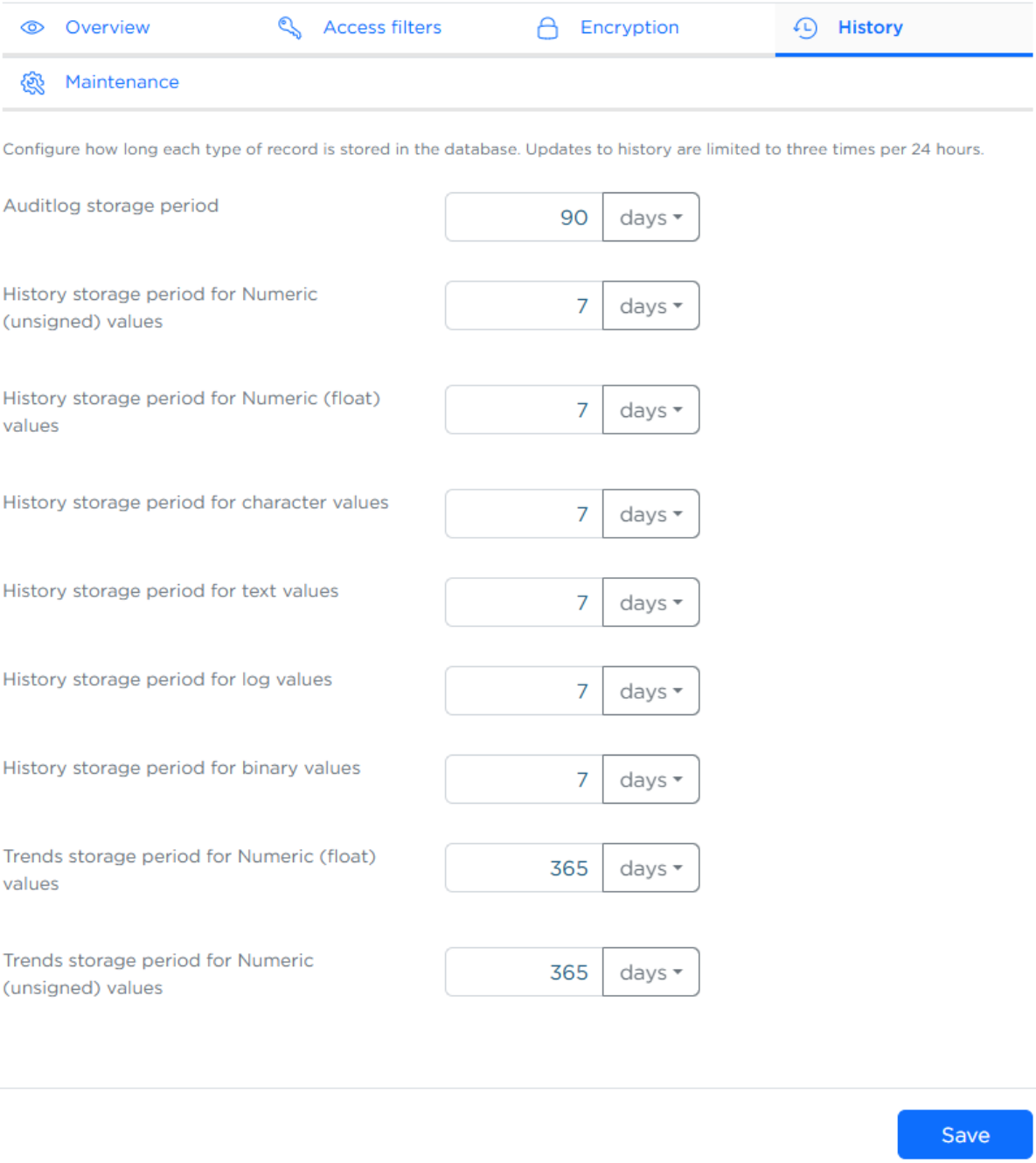
Lai izveidotu dublējumu:

1. Noklikšķiniet uz pogas *Izveidot dublējumu*.
2. Uznirstošajā logā pēc izvēles ievadiet aprakstu.
3. Noklikšķiniet uz *Izveidot*, lai apstiprinātu.

Warning:
Par papildu dublējumiem tiek iekasēta ikmēneša maksa atkarībā no izmantotās krātuves apjoma.

Vēstures cilne

Cilnē *Vēsture* varat konfigurēt, cik ilgi tiek glabāti dažādu veidu vēsturiskie dati.



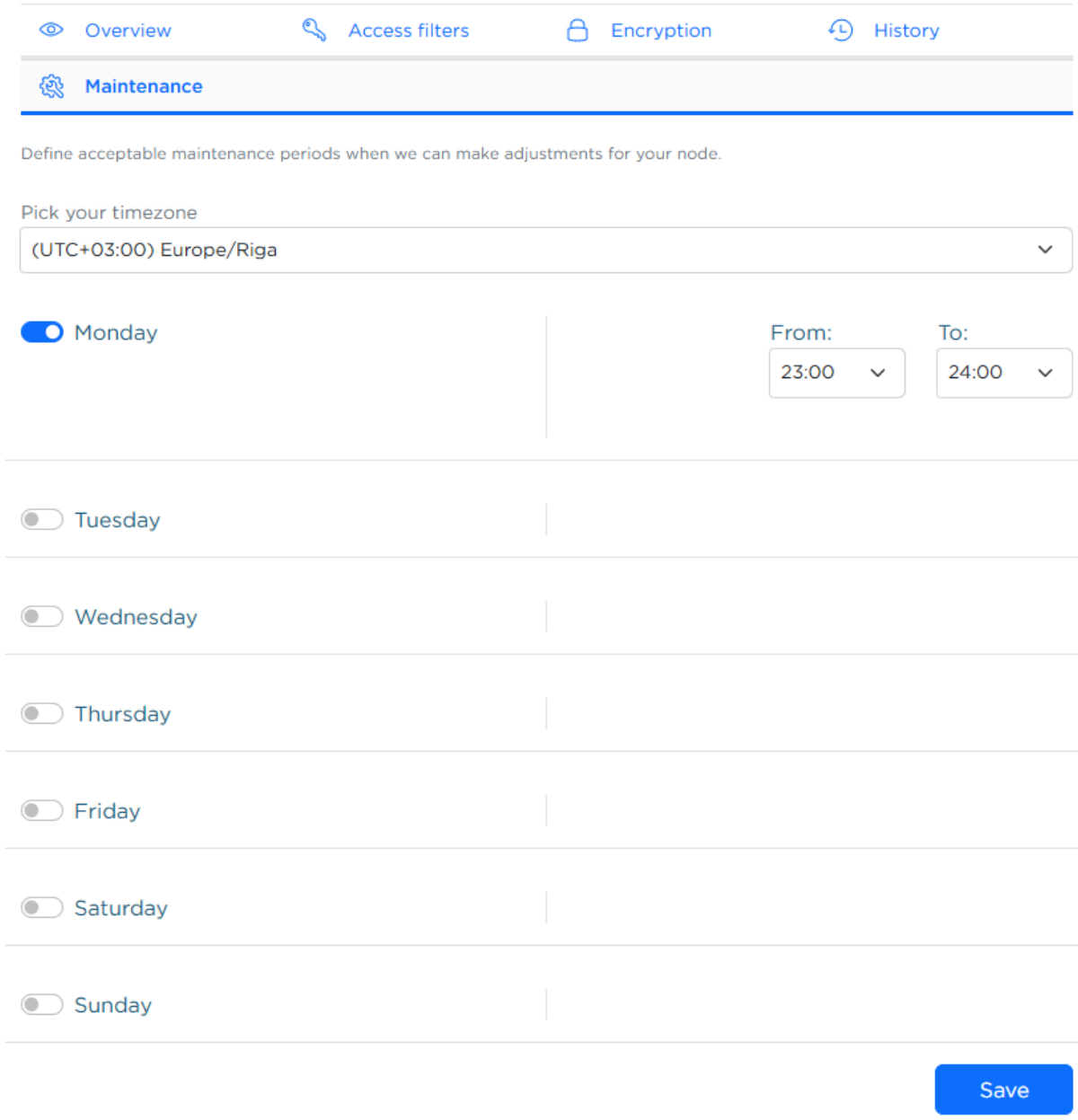
Varat iestatīt glabāšanas periodu katram datu veidam (piemēram, skaitliskajiem datiem, tekstam un žurnāliem) un izvēlēties mērvienību (dienas, mēneši, gadi). Izmaiņas var veikt ne vairāk kā trīs reizes 24 stundu laikā.

Pēc vērtību atjaunināšanas noklikšķiniet uz *Saglabāt*, lai lietotu izmaiņas.

Warning:
Ilgāki glabāšanas periodi var palielināt diska izmantošanu un ietekmēt veiktspēju. Ieteicams konfigurēt glabāšanas periodus, ņemot vērā faktiskos datus un analīzes vajadzības.
 Zabbix Cloud datu glabāšanas pārvaldībai izmanto datubāzes sadalīšanu nodalījumos, nevis tradicionālo Zabbix mājsaimniecības uzkopšanu. Tas nodrošina labāku veiktspēju un paredzamāku glabāšanas periodu pārvaldību.

Uzturēšanas cilne

Cilnē *Uzturēšana* varat definēt pieļaujamos uzturēšanas periodus, kuros jūsu mezglā var veikt atjaunināšanu vai pielāgošanu.

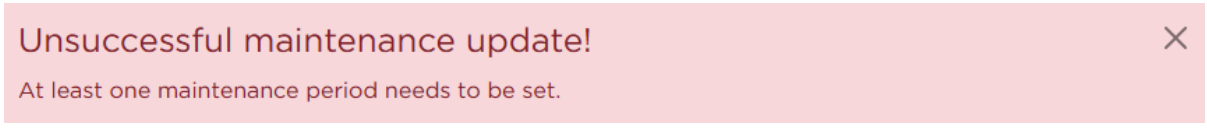


Jums ir jāveic tālāk norādītās darbības:

- Nolaižamajā sarakstā atlasiet savu laika joslu.
- Uzturēšanai atlasiet vismaz vienu dienu un vienu stundu intervālu.

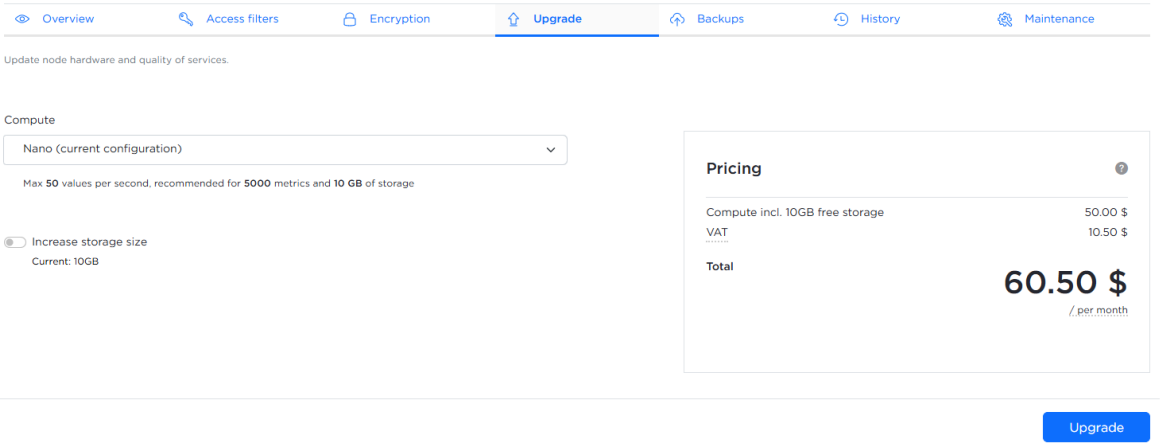
Noklikšķiniet uz *Saglabāt*, lai lietotu izmaiņas.

Ja nav atlasīta neviena diena vai laika intervāls, tiks parādīts kļūdas ziņojums.



Jauninašanas cilne

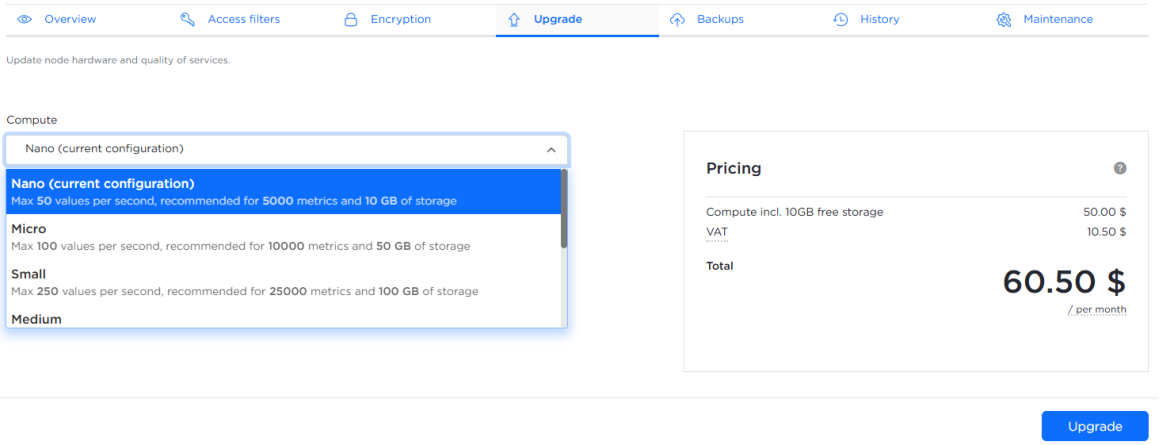
Cilne *Jaunināšana* ļauj palielināt jūsu mezgla resursus vai pārvaldīt tā abonementu.



Varat palielināt skaitļošanas resursu līmeni vai diska lielumu. Ņemiet vērā, ka diska lielumu var palielināt tikai reizi 24 stundās, un to nevar samazināt, izmantojot lietotāja saskarni (ja vēlaties samazināt diska lielumu, sazinieties ar Zabbix atbalsta dienestu). Bezmaksas izmēģinājuma mezglus var pārveidot par maksas abonementiem.

Lai palielinātu skaitļošanas resursu līmeni:

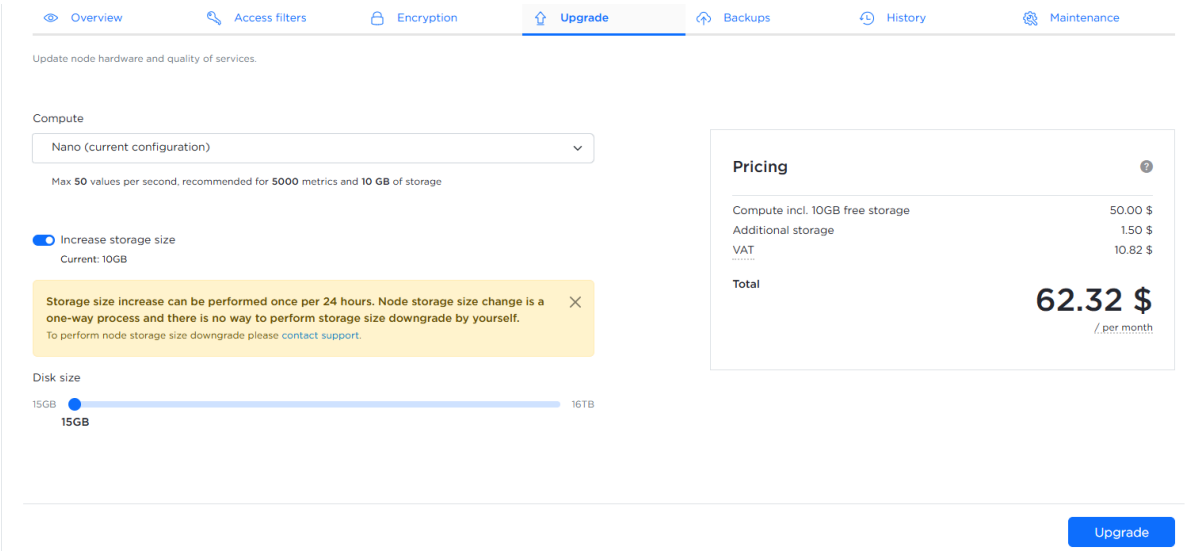
1. Noklikšķiniet uz nolaižamā saraksta *Compute*.



2. Atlasiet vēlamo konfigurācijas līmeni.
3. Noklikšķiniet uz pogas *Upgrade*.

Lai palielinātu diska lielumu:

1. Noklikšķiniet uz radiopogas *Increase storage size*.
2. Pielāgojiet slīdni *Disk size* līdz vēlamajai vērtībai.



3. Noklikšķiniet uz *Upgrade*.

Note:
Atjauninātā kopējā mēneša maksa tiek parādīta tabulā *Pricing* pirms izmaiņu apstiprināšanas.

Lietotāju pievienošana

Ievads

Tālāk sniegtajos piemēros ir aprakstīts, kā pievienot lietotājus jūsu organizācijai un mezgliem.

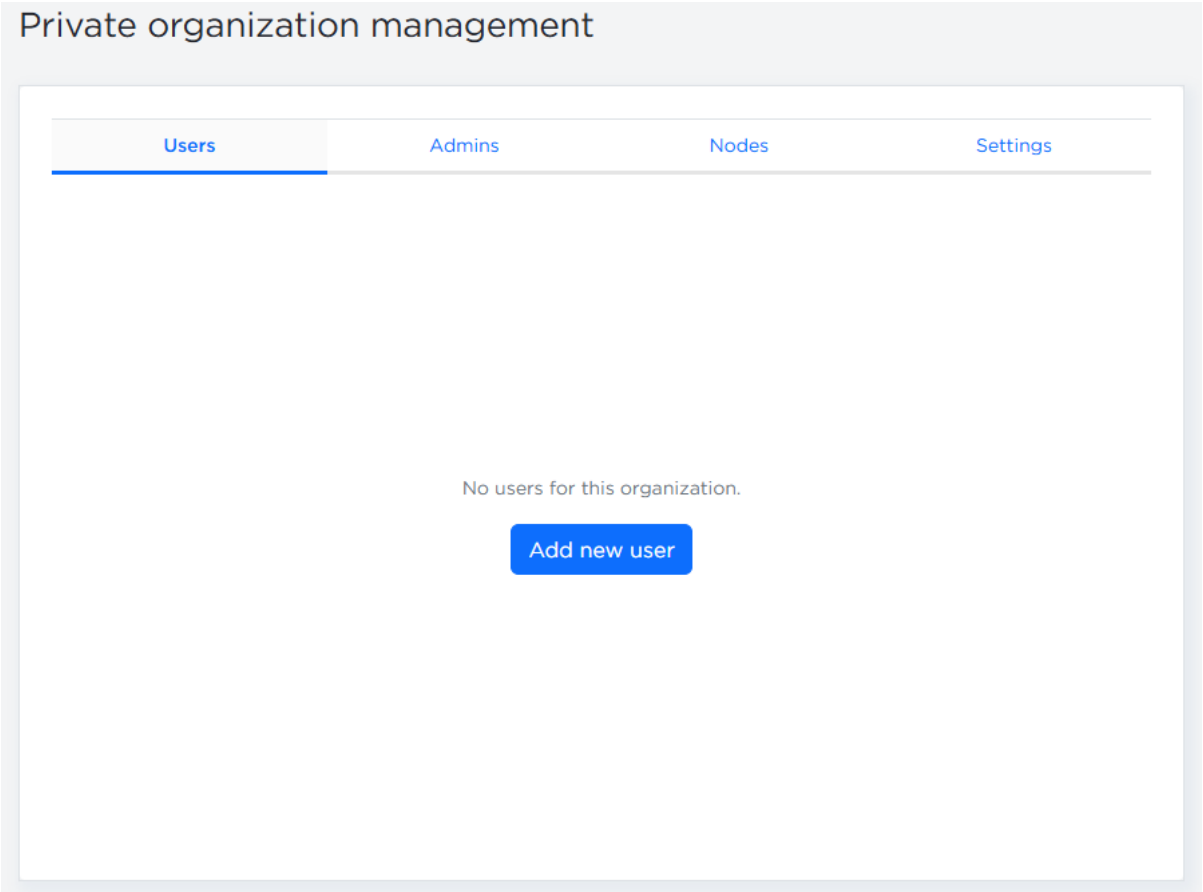
Note:
Lietotāja saskarnes paroles kopēšanu, notīrīšanu un atiestatīšanu var veikt tikai īpašnieks.

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*.

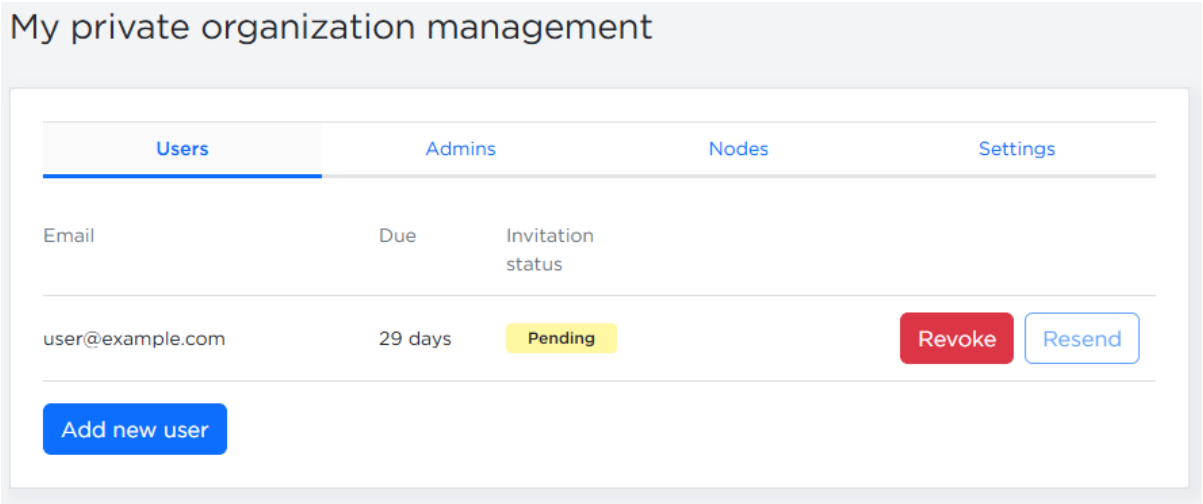
Ņemiet vērā, ka, izvēloties lomu *User*, lietotājs tiek pievienots organizācijai, taču viņam netiek piešķirta piekļuve nevienam mezglam.

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.

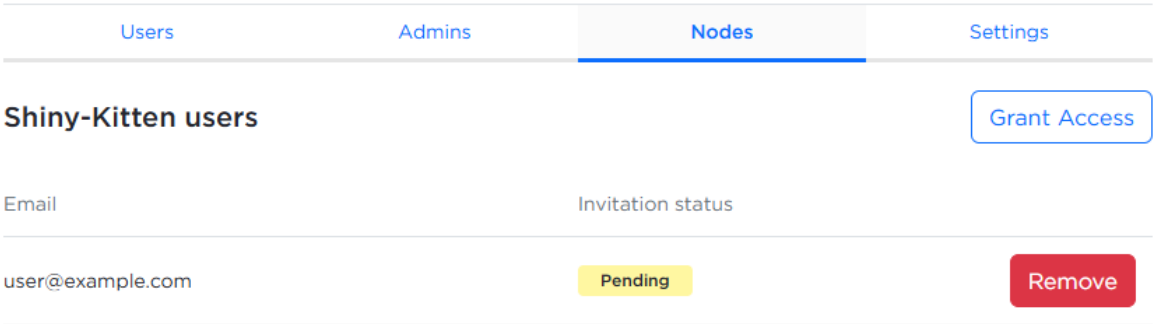
Uzaicinātais lietotājs tiks pievienots ar statusu *Pending*.
Jūs varat atsaukt uzaicinājumu vai nosūtīt to atkārtoti (atkārtota nosūtīšana ir pieejama reizi 10 minūtēs — poga būs atspējota, ja šis ierobežojums būs sasniegts).



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 lokālajās Zabbix instalācijās un mākoņa mezglos pārklājas, pastāv arī atšķirības.
Tālāk ir sniegta kopsavilkuma salīdzinājuma tabula.
Nākamajās sadaļās atsevišķi vienumi ir aprakstīti detalizēti.

Funkcija / aspekts	Zabbix Cloud	Zabbix lokālā instalācija
Laidienu cikls	Saskaņots ar LTS laidieniem	Jūs izvēlaties savu Zabbix versiju (stabilu, beta vai pielāgotu būvējumu)
Lietošanas ērtums	Lietošanai gatava platforma, ko pilnībā pārvalda Zabbix	Nepieciešama instalēšana, konfigurēšana un uzturēšana
Ievainojamību labošana	Automātiska OS/Zabbix labošana	Lietotājs veic labošanu manuāli
HTTPS sertifikāts	Derīgs sertifikāts uzreiz pēc instalēšanas (bez Let's Encrypt)	Pēc noklusējuma pašparakstīts (var konfigurēt Let's Encrypt vai citu CA)
DB veiktspēja (INSERT/UPDATE/SELECT)	Automātiska sadalīšana nodalījumos maksimālam ātrumam; nav ilgstošu DELETE darbību	Manuāla sadalīšana nodalījumos un tīrīšana (iespējamas ilgstošas DELETE transakcijas)
Veiktspējas regulēšana	Zabbix Cloud komanda veic iepriekšēju regulēšanu (bez izmaiņu žurnāliem)	Pilnīga pārskatāmība un kontrole pār regulēšanas parametriem
Ugunsmūra kontrole	To pārvalda Zabbix Cloud komanda; GUI/starpnieka porti nav publiski pieejami	Pilnīga kontrole pār ugunsmūra noteikumiem un portu pieejamību
Izejošā datplūsma 25. portā	Pēc noklusējuma bloķēta, lai novērstu surogātpastu un ļaunprātīgu izmantošanu; tā vietā izmantojiet SMTP 587. vai 465. portu	Zabbix to neierobežo; tas ir atkarīgs no jūsu tīkla un mitināšanas pakalpojumu sniedzēja
DB izolācija un aizsardzība	Augsta izolācijas pakāpe, nav tiešas piekļuves DB/SSH	Atkarīga no jūsu tīkla un hosta drošības
Augsta pieejamība (HA)	Darbojas augstas pieejamības mākoņa platformā, lai uzlabotu datu apkopošanas noturību, var izmantot Zabbix starpniekserverus	HA pārvalda lietotājs
SNMP trap	Tikai ar īpašu starpniekserveri (nav HA/slodzes līdzsvarota trap uztvērēja)	Tieši vai ar starpniekserveri; iespējama HA/slodzes līdzsvarošana
SNMP aptauja	Sarežģīta bez starpniekserveriem (katrai ierīcei nepieciešams NAT/pielāgots ports)	Iebūvēta SNMP aptauja; starpniekserveris nav obligāts
Datu saglabāšanas iestatījumi	Vēsture/Trendi/Revīzija pieejama tikai lietotāja saskarnē (ne API, ne konfigurācijas failā)	Konfigurējama Zabbix servera konfigurācijas failā vai ar API

Funkcija / aspekts	Zabbix Cloud	Zabbix lokālā instalācija
Pielāgoti skripti	`AlertScriptsPath`, `ExternalScripts`, lietotāja saskarnes un kopienas moduļi netiek atbalstīti	Pilnībā atbalstīti (skriptu ceļš, moduļi, integrācijas)
ODBC uzraudzība	Tikai PostgreSQL (draiveris `{postgresql}`); MariaDB spraudnis ir pieejams kā vietturis	ODBC PostgreSQL, MySQL, Oracle u. c. datubāzēm - pilnībā konfigurējams
ODBC izsaukumu ierobežošana	Nav iespējama (pieejams tikai `StartODBCPollers=1`); intensīvi sinhronizācijas vaicājumi bloķē citus darbus	Pilnībā regulējams aptaujas procesu skaits un plānošana
SAML sertifikāta augšupielāde	Netiek atbalstīta	Atbalstīta, izmantojot lietotāja saskarni vai API
Ieplānotie pārskati	Jāizveido savs Email multivides tips (skriptu multivides tips netiek atbalstīts)	Pēc noklusējuma atbalsta gan skriptu, gan Email multivides tipus
Hosta saskarne aktīvajām pārbaudēm	Platformas piešķirta IP adrese (tiek 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.

Izejošā datplūsma 25. portā

Lai novērstu surogātpastu un ļaunprātīgu izmantošanu, izejošie savienojumi no Zabbix Cloud mezgliem uz TCP 25. portu pēc noklusējuma ir bloķēti. Tas ietekmē:

- E-pasta multivides tipus, kas sūta ziņojumus, izmantojot SMTP serveri 25. portā. Tā kā 25. ports ir noklusējuma *SMTP servera ports* jaunam e-pasta multivides tipam, mainiet to uz 587 (iestatot *Savienojuma drošību* uz STARTTLS) vai 465 (SSL/TLS) atkarībā no tā, ko atbalsta jūsu SMTP serveris. E-pasta pakalpojumu sniedzēju *Office365 relay* nevar izmantot, jo tas savienojas ar Microsoft relay galapunktu 25. portā.
- Ieplānotajiem pārskatiem, kuriem nepieciešams e-pasta multivides tips.
- Pārbaudēm, ko pats mākoņa mezgls veic 25. portā, piemēram, `net.tcp.service[smtp]` vienkāršajām pārbaudēm vai SMTP pārbaudēm tīkla atklāšanas noteikumos. Šīs pārbaudes ziņo, ka SMTP pakalpojums nedarbojas, pat ja tas darbojas; tā vietā veiciet šādas pārbaudes, izmantojot Zabbix starpniekserveri vai Zabbix aģentu savā tīklā.

Noklusējuma *Cloud Email* multivides tips netiek ietekmēts.

Aktīvās pārbaudes saskarne

Aktīvās pārbaudes mākonī 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.

Ieplānotie pārskati

Noklusējuma *Cloud Email* multivides tips izmanto skripta transportu, un to nevar izmantot ieplānotajiem pārskatiem. Lai nosūtītu pārskatus pa e-pastu:

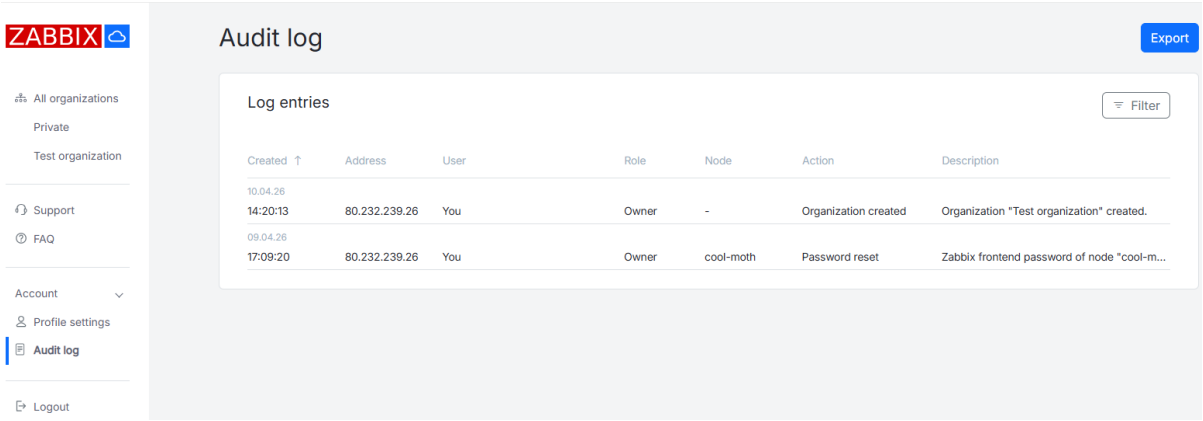
1. Izveidojiet jaunu e-pasta multivides tipu.
2. Piešķiriet e-pasta multividi savam lietotājam sadaļā *Lietotāja iestatījumi* > *Multivide*.
3. Konfigurējot ieplānotos pārskatus, atlasiet šo multivides tipu.

Attention:
Šim multivides tipam neizmantojiet 25. portu kā *SMTP servera portu*; skatiet Izejošā datplūsma 25. portā.

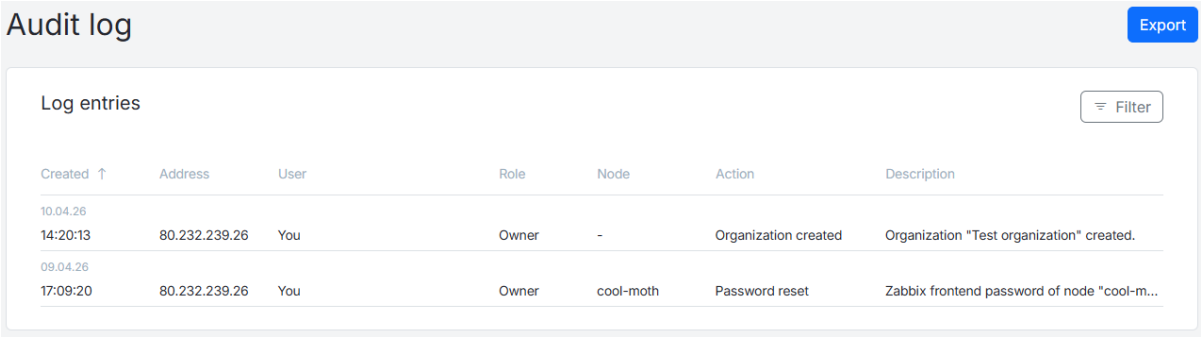
Audita žurnāls

Pārskats Zabbix Cloud audita žurnāls reģistrē lietotāju un sistēmas darbības.

Pēc pieteikšanās Zabbix Cloud kontā dodieties uz *Account* > *Audit log*, lai skatītu audita žurnāla ierakstus. Lietotājam pieejamie audita žurnāla ieraksti ir atkarīgi no lietotāja organizācijas lomas.



Audita žurnāla ierakstu skatīšana Audita žurnāla ieraksti tiek parādīti tabulā.



Tabulā ir iekļauta šāda informācija:

Column	Description
Created	Datums un laiks, kad tika izveidots žurnāla ieraksts. Parādītais laiks atbilst lietotāja laika joslai, kas izvēlēta nolaižamajā sarakstā *User timezone* sadaļā *Account > Profile settings*.
User	Organizācijas lietotāja vārds, kurš veica darbību.
Role	Lietotāja, kurš veica darbību, loma.
Node	Ar darbību saistītā mezgla nosaukums.
Action	Veiktās darbības nosaukums.
Description	Veiktās darbības apraksts.

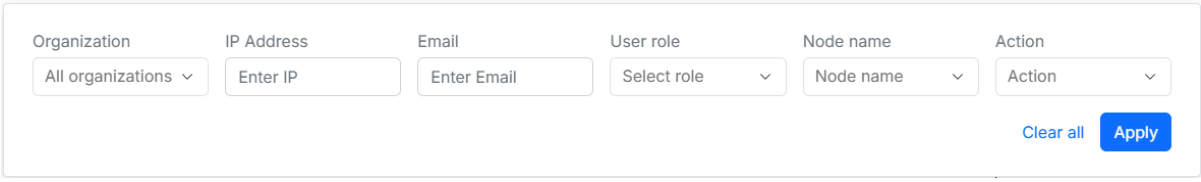
Lai kārtotu ierakstus augošā vai dilstošā secībā, var noklikšķināt uz kolonnu nosaukumiem.

Filtrētus audita žurnāla ierakstus var eksportēt `.csv` formātā, izmantojot pogu *Export*.

Filtru izmantošana

Izmantojiet pogu *Filter*, lai atvērtu filtru.

Lai parādītu filtrētos rezultātus, pēc viena vai vairāku filtrēšanas parametru atlasīšanas noklikšķiniet uz *Apply*. Lai notīrītu filtru, noklikšķiniet uz *Clear all*.



Ierakstus var filtrēt pēc viena vai vairākiem parametriem. Ir pieejami šādi filtrēšanas parametri:

Parameter	Description
Organization	Atlasiet visas organizācijas vai konkrētu organizāciju.
IP Address	Norādiet IP adresi.
Email	Norādiet lietotājam piesaistītu e-pasta adresi.
User role	Atlasiet visas lomas vai konkrētu lomu: Owner, Admin vai User.
Node name	Atlasiet visus mezglus vai konkrētu mezglu.
Action	Atlasiet konkrētu darbības veidu.

Audita žurnālā tiek reģistrētas šādas darbības:

- *Node created*
- *Node deleted*
- *Node upgraded*
- *Node version updated*
- *Password reset*
- *Maintenance updated*
- *History setting updated*
- *Node expiration date extended*
- *Proxy configuration updated*
- *Backup created*
- *Backup deleted*

- *Backup restored*
- *Backup updated*
- *Access filter added*
- *Access filter updated*
- *Access filter deleted*
- *Encryption enabled*
- *Encryption disabled*
- *Subscription cancelled*
- *Subscription renewed*
- *Subscription payment failed*
- *Payment requires confirmation*
- *Organization created*
- *Organization deleted*
- *Admin invited*
- *User invited*
- *Admin joined*
- *User joined*
- *Admin removed from organization*
- *User removed from organization*
- *Node added to organization*
- *Node removed from organization*
- *Node access granted*
- *Node access revoked*
- *User settings updated*
- *Payment created*
- *Payment completed*
- *Payment failed*
- *VAT charge applied*
- *VAT validation failed*
- *Billing info updated*
- *Payment method updated*
- *Payment method removed*
- *Prepaid node created*
- *Prepaid node updated*
- *Incident started*
- *Incident ended*

Zabbix Cloud starpniekservera iestatīšana

Pārskats Šajā lapā aprakstīts, kā konfigurēt Zabbix starpniekserveri, kas izvietots Zabbix Cloud, lai apkopotu uzraudzības datus no attālinātā vidē esošajiem hostiem.

Procedūra galvenokārt paredzēta Zabbix serverim un starpniekserverim, kas izveidoti Zabbix Cloud.

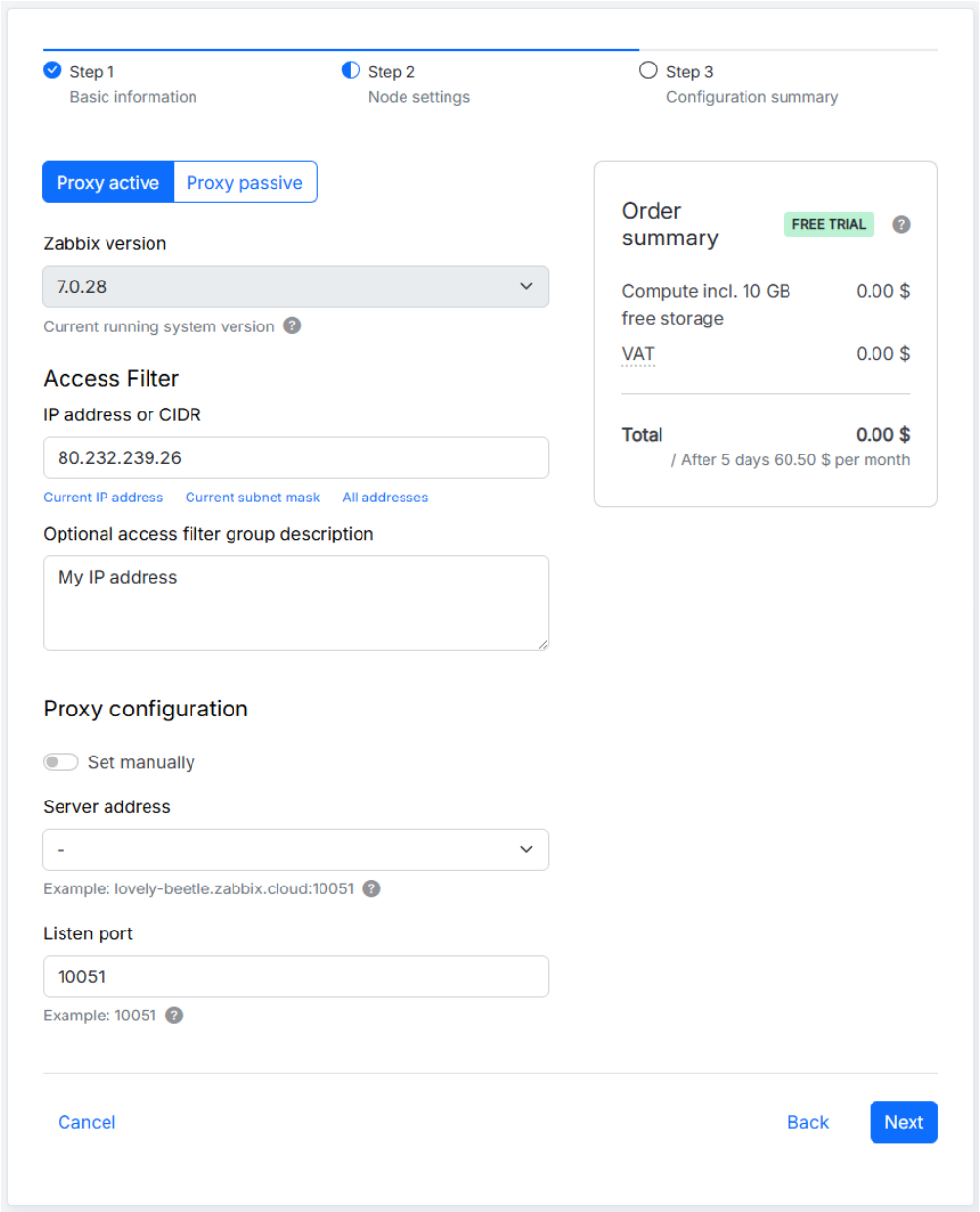
Procedūrā aprakstīta arī nepieciešamā konfigurācija gadījumā, ja Zabbix serveris nav izvietots Zabbix Cloud.

Priekšnosacījumi Zabbix serveris ir instalēts, konfigurēts un darbojas, vai arī Zabbix Cloud darbojas Zabbix servera mezgls.

Izveidot starpniekservera mezglu Izveidojiet Zabbix Cloud starpniekservera mezglu.

Mezgla izveides laikā:

1. Atlasiet *Proxy*.
2. Atlasiet *Proxy active* vai *Proxy passive* atkarībā no tā, kā starpniekserverim jāsazinās ar Zabbix serveri.
3. Konfigurējiet mezgla IP adreses vai CIDR piekļuves filtrus (to var izdarīt arī pēc mezgla izveides).
4. Norādiet Zabbix serveri, ar kuru tiks savienots starpniekserveris.



5. Pabeidziet mezgla izveidi.

Plašāku informāciju par starpniekservera mezgla izveidi skatiet sadaļā Zabbix izvietošana mākonī.

Piekļuves filtru konfigurēšana Zabbix Cloud izmanto piekļuves filtrus, lai kontrolētu, kuras IP adreses vai apakštīkli var izveidot savienojumu ar mākoņa mezglu.

Mezgla inicializācijas laikā piekļuves filtrs tiek automātiski izveidots IP adresei, kas tika izmantota piekļuvei mezglam. Tas ne vienmēr ietver uzraudzīto hostu vai citu Zabbix komponentu adreses. Pievienojiet nepieciešamās adreses manuāli.

Plašāku informāciju skatiet sadaļā Mezgla konfigurēšana — piekļuves filtri.

Starpniekservera mezgls

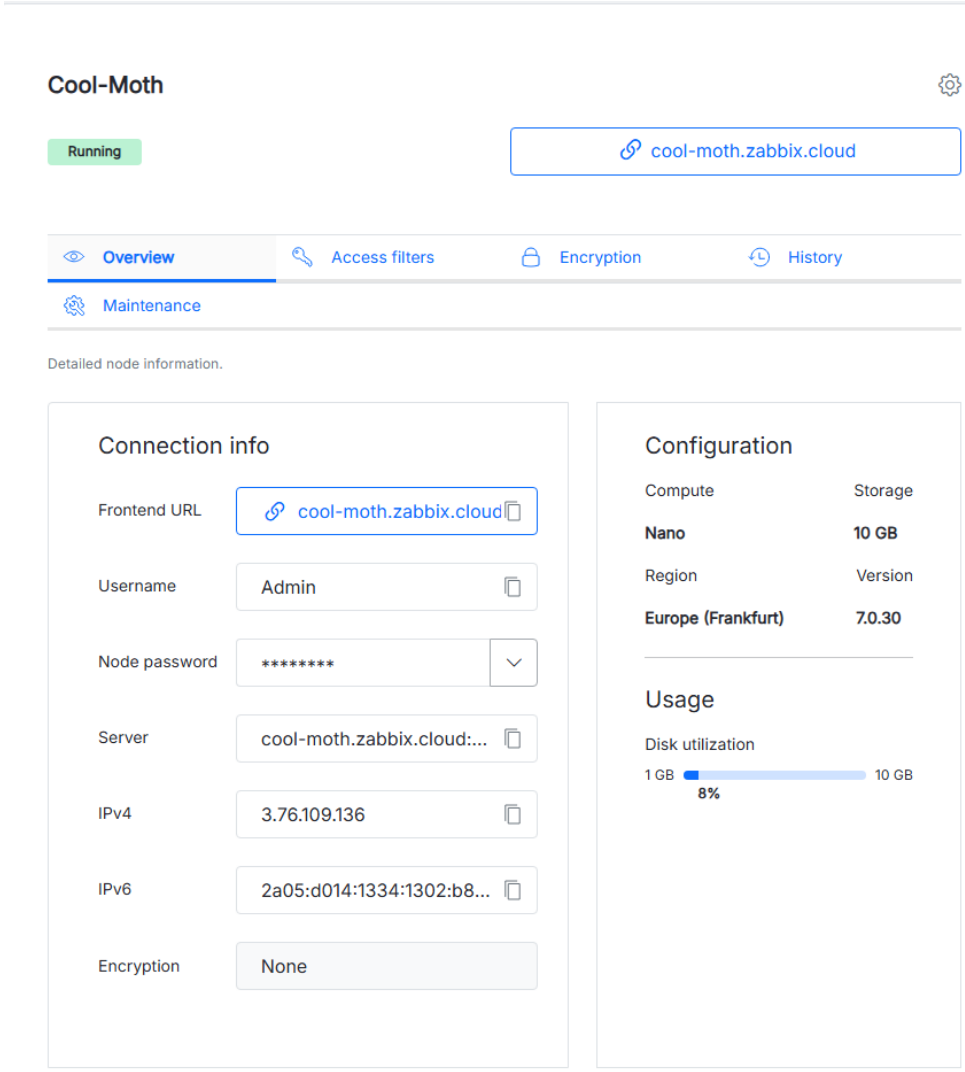
Pievienojiet IP adreses vai apakštīklus, no kuriem ir paredzēti savienojumi ar starpniekserveri.

Ja uzraudzītie hosti izmanto aktīvās aģenta pārbaudes, aģenti uzsāk savienojumus ar starpniekserveri, tāpēc uzraudzīto hostu IP adrese vai apakštīkls ir jāatļauj starpniekservera mezgla piekļuves filtros.

Ja kā starpniekservera režīms tika iestatīts *Proxy Passive*, Zabbix serveris izveido savienojumu ar starpniekserveri, tāpēc Zabbix servera IP adrese vai apakštīkls ir jāatļauj starpniekservera mezgla piekļuves filtros.

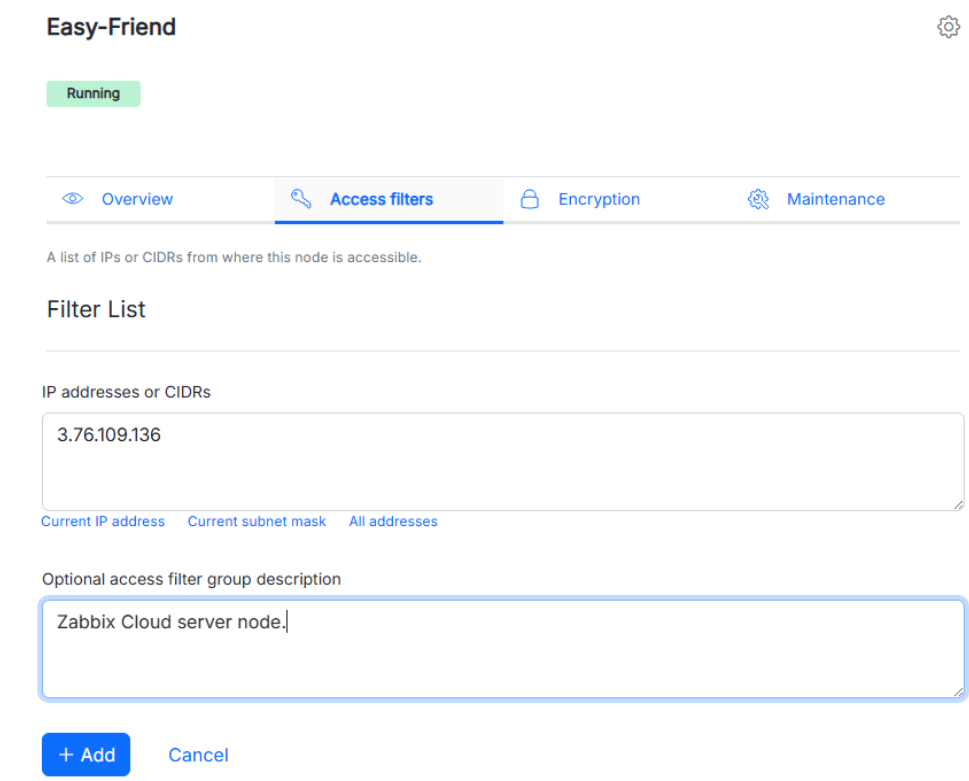
Ja tika atlasīts *Proxy passive* un tiek izmantots servera mezgls:

1. Atveriet servera mezgla cilni *Overview* un nokopējiet mezgla IPv4 vai IPv6 adresi.

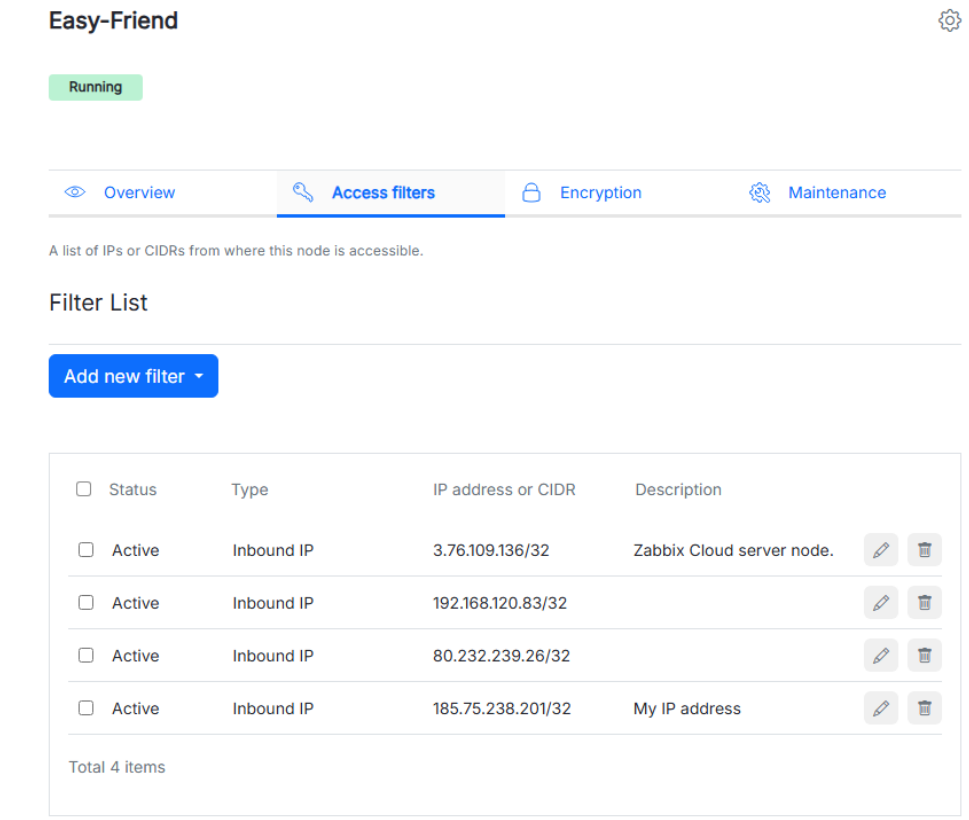


2. Atveriet starpniekservera mezgla cilni *Access filters* un noklikšķiniet uz *Add new filter* > *Add*.

3. Ielīmējiet nokopēto adresi laukā *IP addresses or CIDRs* un laukā *Optional access filter group description* pievienojiet aprakstu.



4. Noklikšķiniet uz *Add*.



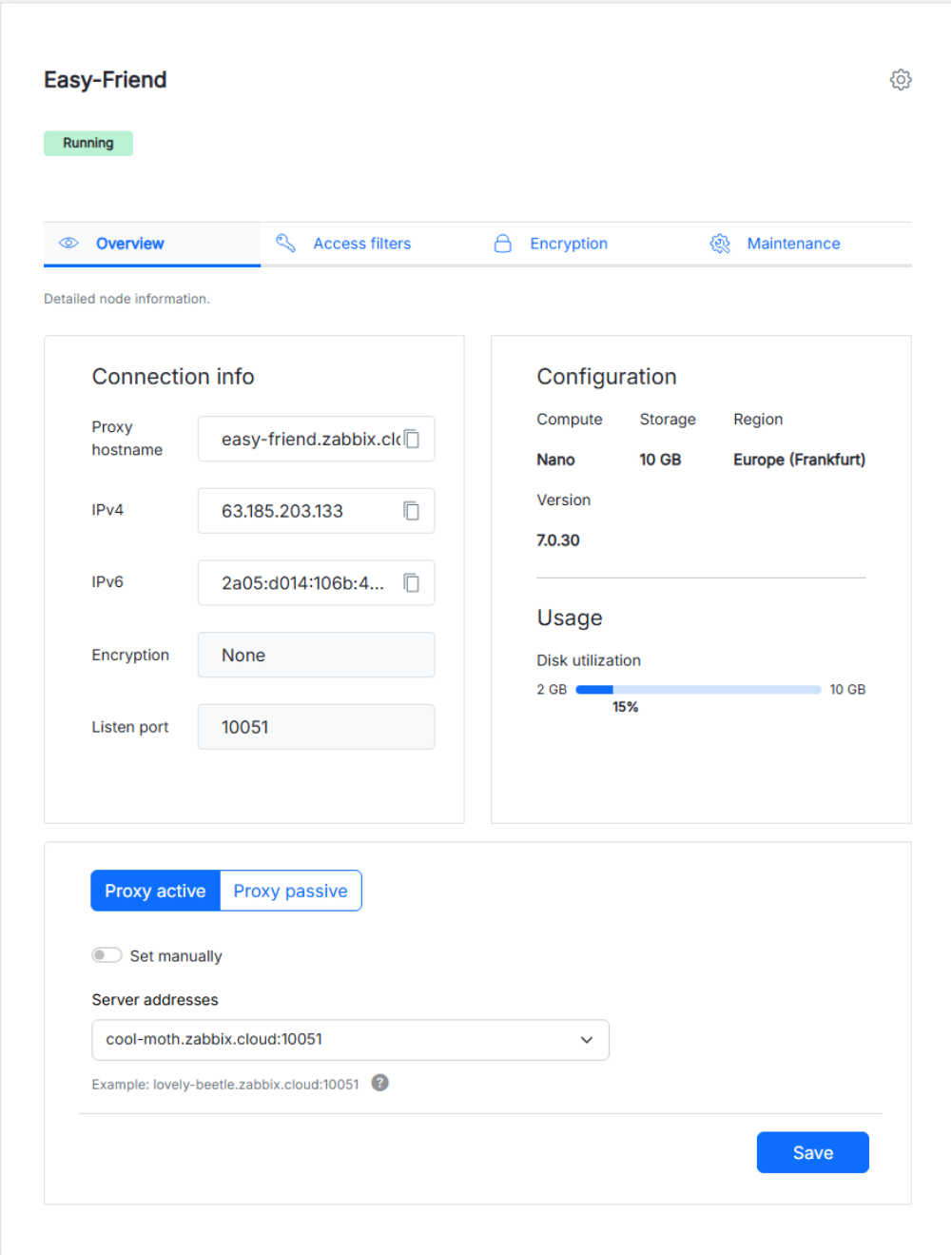
Veiciet 2.–4. darbību Zabbix aģenta, kas izmanto aktīvās pārbaudes, IP adresei.

Servera mezgls

Ja Zabbix serveris tiek mitināts Zabbix Cloud, konfigurējiet piekļuves filtrus arī servera mezglā.

Aktīvā režīma starpniekserverim pievienojiet starpniekservera mezgla IP adresi vai CIDR servera mezgla piekļuves filtriem, jo starpniekserveris uzsāk savienojumu ar serveri.

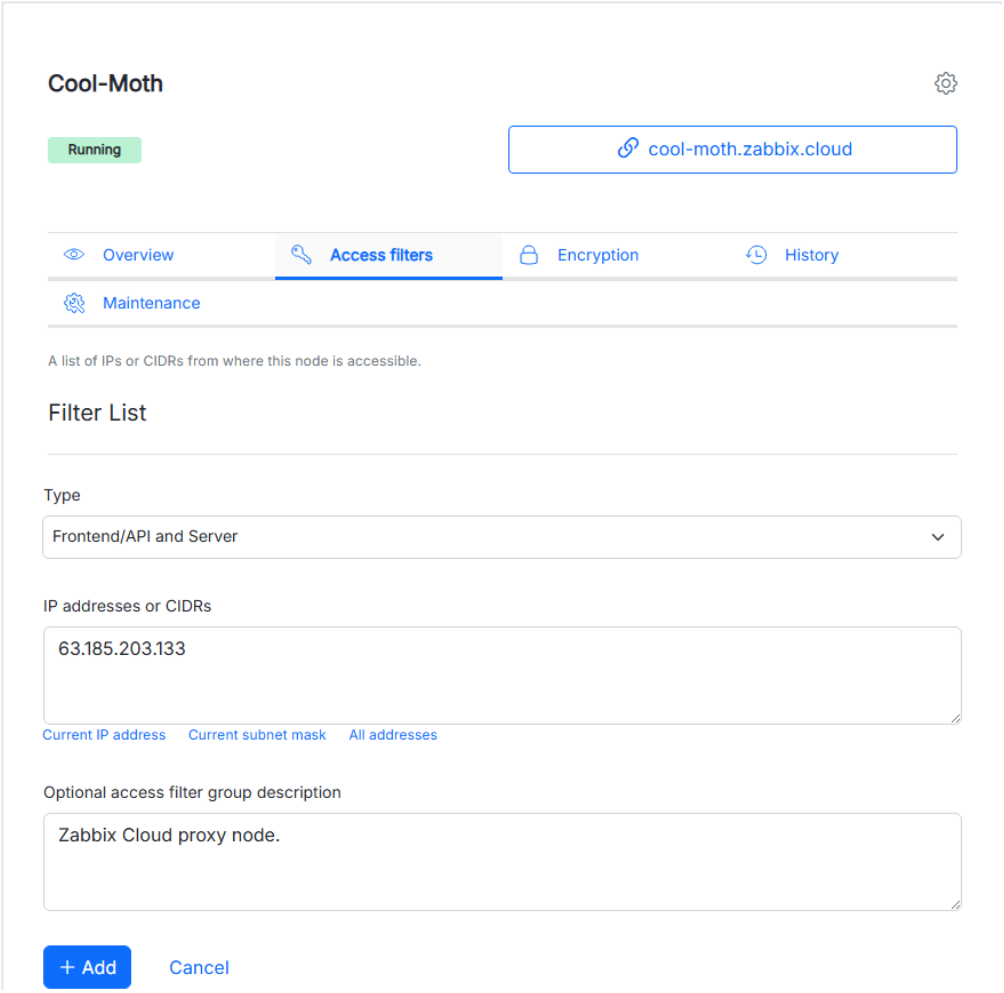
1. Atveriet starpniekservera mezgla cilni *Overview* un nokopējiet mezgla IPv4 vai IPv6 adresi.



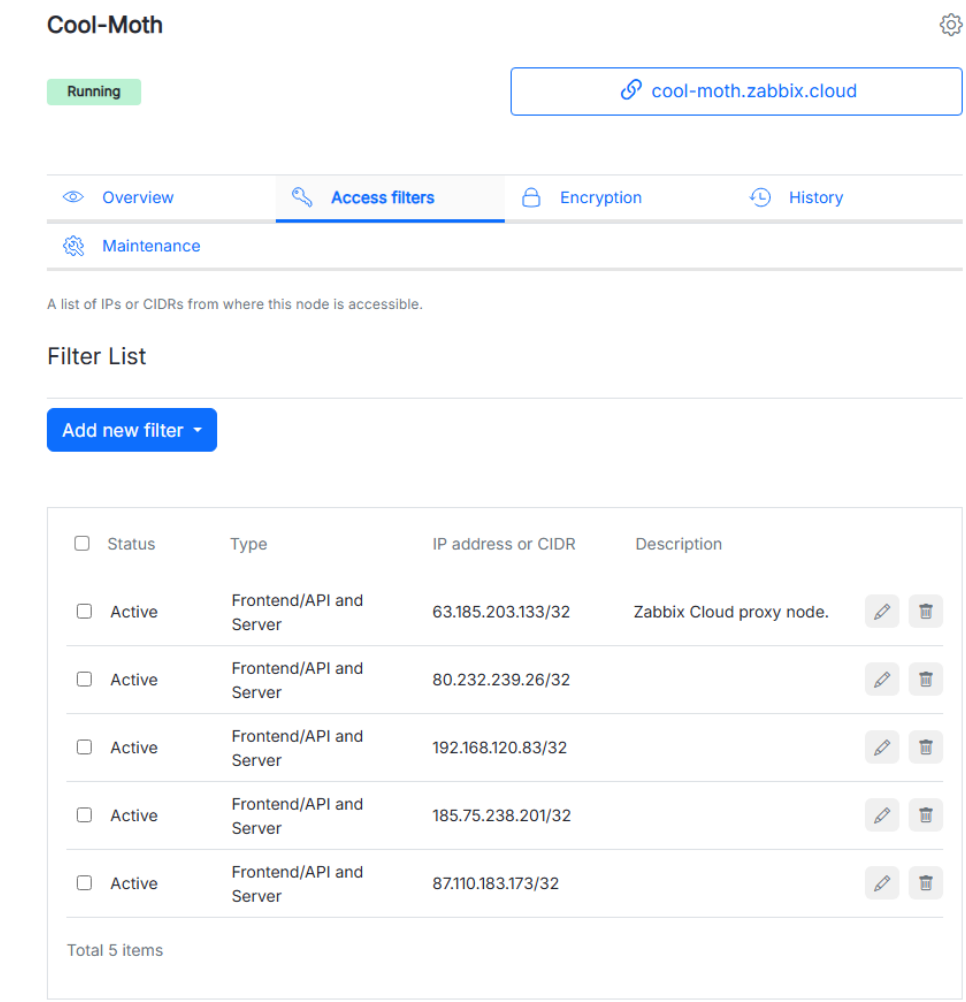
2. Atveriet servera mezgla cilni *Access filters* un noklikšķiniet uz *Add new filter* > *Add*.

3. Ielīmējiet nokopēto adresi laukā *IP addresses or CIDRs* un laukā *Optional access filter group description* pievienojiet aprakstu.

4. Atlasiet atbilstošo filtra tipu.



5. Noklikšķiniet uz *Add*.



Konfigurējiet starpniekserveri Starpniekserveris ir jākonfigurē Zabbix lietotāja saskarnē, lai tas varētu sazināties ar Zabbix serveri.

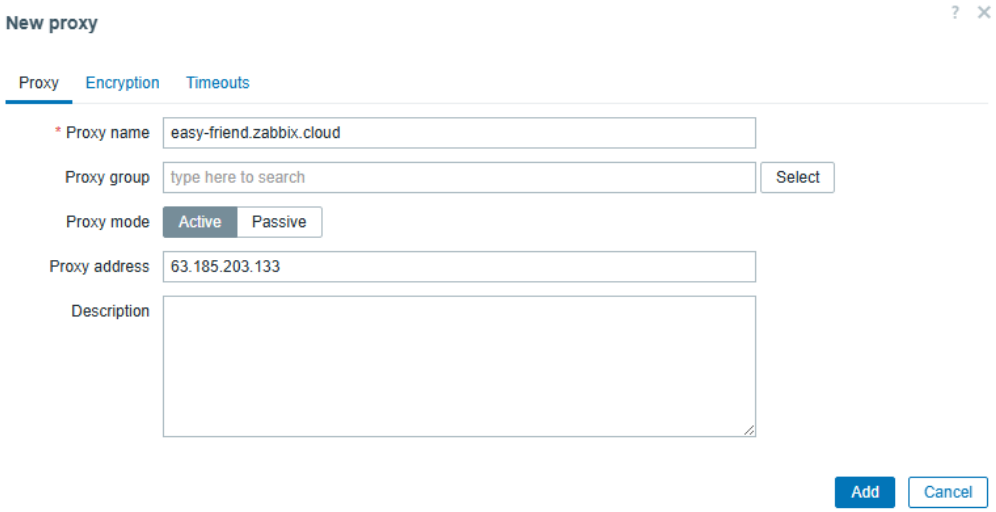
Atveriet *Administration* > *Proxies* Zabbix lietotāja saskarnē, noklikšķiniet uz *Create proxy* un konfigurējiet starpniekserveri atbilstoši nepieciešamajam:

1. Laukā *Proxy name* pievienojiet starpniekservera nosaukumu.
Tam ir jāatbilst starpniekserverim konfigurētajam resursdatora nosaukumam.
Ievadiet to manuāli vai nokopējiet no starpniekservera cilnes *Overview*.

2. Atlasiet Proxy mode, kas atbilst starpniekservera izveides laikā atlasītajam režīmam.

3. Aktīvam starpniekserverim ievadiet starpniekservera IP adresi vai DNS nosaukumu.

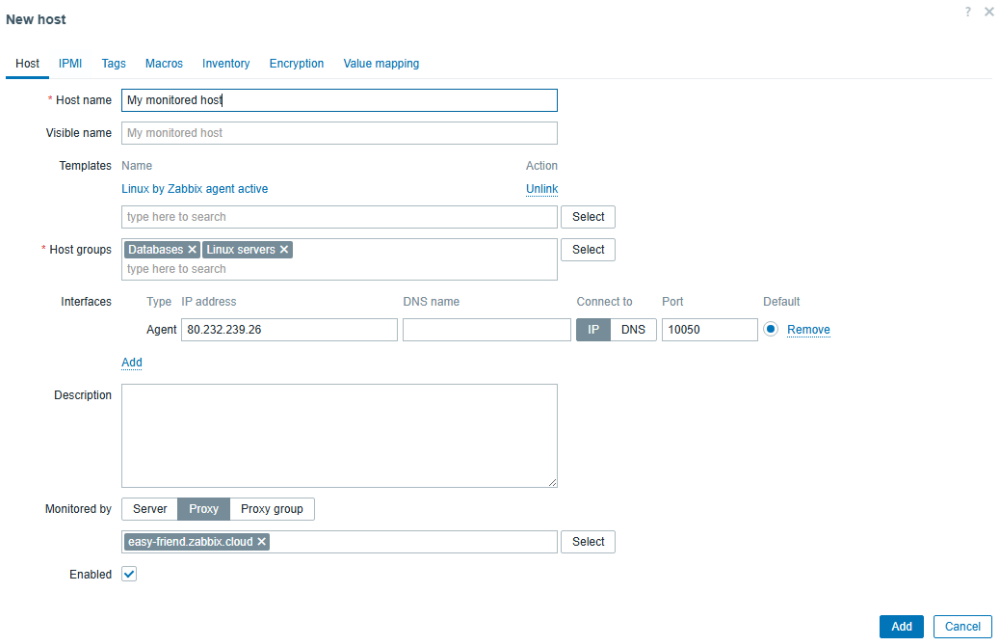
4. Noklikšķiniet uz *Add*.



Papildinformāciju skatiet sadaļā Distributed monitoring using proxies.

Konfigurējiet uzraugāmos hostus Pievienojiet hostus, kas jāuzrauga, izmantojot starpniekserveri.

1. Atveriet *Data collection* > *Hosts*.
2. Izveidojiet hostu vai atveriet esošu hostu.
3. Sadaļā *Monitored by* atlasiet *Proxy* un vai nu ievadiet starpniekservera nosaukumu, lai to meklētu un atlasītu, vai nospiediet pogu *Select*, lai atvērtu atlasāmo starpniekserveru sarakstu.
4. Konfigurējiet hosta saskarni un citus parametrus.
5. Noklikšķiniet uz *Add* vai *Update*.



Hosts, kas konfigurēts uzraudzīšanai, izmantojot starpniekserveri, nosūta vai saņem uzraudzības datus, izmantojot šo starpniekserveri, atkarībā no vienuma tipa un aģenta režīma.

Zabbix aģenta konfigurēšana Ja uzraudzītais hosts izmanto Zabbix aģentu vai Zabbix aģentu 2, konfigurējiet aģentu saziņai ar atbilstošo Zabbix komponenti.

Konfigurācija ir atkarīga no tā, vai aģents izmanto *pasīvās* vai *aktīvās* pārbaudes.

Pasīvās aģenta pārbaudes

Izmantojot pasīvās pārbaudes, Zabbix starpniekserveris izveido savienojumus ar aģentu un pieprasa uzraudzības datus.

Atveriet Zabbix aģenta konfigurācijas failu un atjauniniet parametru `Server`, lai atļautu savienojumus no Zabbix Cloud starpniekservera IP adreses.

Piemēram:

```
Server=<proxy_IP>
```

Pārliecinieties, ka uzraudzītais hosts atļauj ienākošos TCP savienojumus no starpniekservera aģenta portā, kas pēc noklusējuma ir 10050.

Starpniekserverim jāspēj sasniegt uzraudzīto hostu tīklā.

Aktīvās aģenta pārbaudes

Izmantojot aktīvās pārbaudes, aģents izveido savienojumus ar Zabbix starpniekserveri un nosūta apkopotos uzraudzības datus.

Atveriet Zabbix aģenta konfigurācijas failu un atjauniniet parametru `ServerActive`, norādot Zabbix Cloud starpniekservera adresi.

Piemēram:

`ServerActive=<proxy_IP>:<proxy_listen_port>`

Aizstājiet `<proxy_IP>` ar Zabbix Cloud starpniekservera IP adresi un `<proxy_listen_port>` ar klausīšanās portu, kas tika konfigurēts, kad starpniekservera mezgls tika izveidots. Noklusējuma klausīšanās ports ir 10051.

Ja parametrā `ServerActive` ir norādīts starpniekserveris, aģents aktīvo pārbaužu konfigurāciju iegūst no starpniekservera, nevis tieši no Zabbix servera.

Pārliecinieties, ka uzraudzītais hosts var izveidot izejošu TCP savienojumu ar starpniekserveri, izmantojot konfigurēto klausīšanās portu.

Plašāku informāciju par pasīvajām un aktīvajām aģenta pārbaudēm skatiet sadaļā Pasīvās un aktīvās aģenta pārbaudes.

Ārējais Zabbix serveris Zabbix Cloud starpniekserveris var sazināties arī ar Zabbix serveri, kas izvietots ārpus Zabbix Cloud.

Šādā gadījumā ārējam Zabbix serverim ir jāspēj sazināties ar mākoņa starpniekserveri atbilstoši atlasītajam starpniekservera režīmam.

Ja starpniekserveris darbojas aktīvajā režīmā, tas iniciē savienojumu ar ārējo Zabbix serveri. Konfigurējiet ārējo Zabbix serveri, lai tas pieņemtu savienojumus no starpniekservera, un pārliecinieties, ka starpniekserveris tīklā var sasniegt serveri.

Pasīvam starpniekserverim savienojumu ar starpniekserveri iniciē Zabbix serveris. Pievienojiet ārējā Zabbix servera IP adresi vai apakštīklu starpniekservera mezgla piekļuves filtriem un pārliecinieties, ka serveris tīklā var sasniegt starpniekserveri.

Ārējais Zabbix serveris neizmanto Zabbix Cloud piekļuves filtrus. Tā vietā konfigurējiet nepieciešamos ugunsmūra un tīkla noteikumus ārējā vidē.

Uzraudzītie hosti var atrasties tajā pašā vidē, kur ārējais Zabbix serveris, citā tīklā vai tīklā, kas ir pieejams starpniekserverim.

Ja aģenti izmanto aktīvās pārbaudes, konfigurējiet to `ServerActive` parametru, lai tas norādītu uz Zabbix Cloud starpniekserveri. Ja aģenti izmanto pasīvās pārbaudes, konfigurējiet to `Server` parametru, lai atļautu savienojumus no starpniekservera.

Pārbaudiet konfigurāciju Pēc starpniekservera, aģentu, piekļuves filtru un tīkla savienojuma konfigurēšanas pārbaudiet, vai starpniekserveris sazinās ar Zabbix serveri.

1. Atveriet *Administration* > *Proxies*.
2. Pārbaudiet starpniekservera *State*.
3. Atveriet *Monitoring* > *Latest data* un pārbaudiet, vai ir pieejamas jaunākās vienumu vērtības.

Ja starpniekserveris nesazinās ar serveri, pārbaudiet:

- starpniekservera režīmu;
- servera adresi un portu;
- starpniekservera piekļuves filtrus;
- servera piekļuves filtrus, ja serveris tiek mitināts Zabbix Cloud;
- aģenta konfigurāciju;
- ugunsmūra un maršrutēšanas noteikumus;
- adresi, kas konfigurēta sadaļā `Server` vai `ServerActive`.

Mobilā lietotne

Mobilā lietotne

Zabbix mobilā lietotne ļauj pārraudzīt Zabbix vidi no mobilās ierīces. Skatiet problēmas, pārbaudiet to detalizētu informāciju, atjauniniet problēmas un saņemiet pašpiegādes paziņojumus, kad nepieciešama uzmanība.

Lejupielādēt lietotni (drīzumā)

Zabbix iestatīšana lietotnei

Iestatiet Zabbix bridge adapter Instalējiet Zabbix bridge adapter, lai mobilās ierīces varētu izveidot savienojumu ar Zabbix.

Pievienot ierīces Pievienojiet savu mobilo ierīci Zabbix lietotāja saskarnē, lai savienotu to ar savu Zabbix serveri.

Iestatiet push paziņojumus Saņemiet brīdinājumus savā mobilajā ierīcē, kad rodas problēma.

Uzziniet par lietotni

Lietotnes pārskats Uzziniet Zabbix mobilās lietotnes lietošanas pamatus.

Droša mobilā piekļuve Nodrošiniet lietotnes, savu mobilo ierīču un Zabbix bridge adaptera drošību vai atsauciet piekļuvi nozaudētām vai kompromitētām ierīcēm.

Zabbix iestatīšana lietotnei

Lai izmantotu Zabbix mobilo lietotni (drīzumā), vispirms ir jāveic tālāk norādītās darbības:

1. Iestatiet Zabbix tilta adapteri.
2. Iestatiet pašpiegādes paziņojumus.
3. Pievienojiet savu mobilo ierīci Zabbix lietotāja saskarnē.

Tālākajās sadaļās ir aprakstīta katra darbība.

Pēc iestatīšanas skatiet sadaļu Droša mobilā piekļuve, lai aizsargātu lietotni un Zabbix tilta adapteri.

Zabbix bridge adapter iestatīšana Lai jūsu Zabbix serveris varētu sazināties ar mobilo lietotni, jums jāiestata Zabbix bridge adapter.

Bridge adapteri var instalēt tajā pašā sistēmā, kurā atrodas jūsu Zabbix serveris, vai citā sistēmā, izmantojot pakotnes:

> **Attention:**
> Tālāk norādītās darbības ir paredzētas Ubuntu 24.04. Citām operētājsistēmām izmantojiet repozitorija un pakotņu instalēšanas komandas (piemēram, `wget`, `dpkg`, `apt`) no lapas Get Zabbix, kas paredzēta jūsu operētājsistēmai, un izmantojiet tās, lai instalētu pakotni `zabbix-bridge-adapter`.

1. Sāciet jaunu čaulas sesiju ar root privilēģijām:

```
sudo -s
```

2. Instalējiet Zabbix repozitoriju:

```
wget https://repo.zabbix.com/zabbix/8.0/release/ubuntu/pool/main/z/zabbix-release/zabbix-release_latest_8.
dpkg -i zabbix-release_latest_8.0+ubuntu24.04_all.deb
apt update
```

3. Instalējiet Zabbix bridge adapteri:

```
apt install zabbix-bridge-adapter
```

4. Rediģējiet Zabbix servera konfigurācijas failu:

```
EnableMobileDevices=1
BridgeAdapterURL=https://bridge-adapter.example.com:10054/rpc
```

5. Restartējiet Zabbix serveri, piemēram (skatiet citas opcijas):

```
systemctl restart zabbix-server
```

6. Nokopējiet Zabbix servera ID no *Reports* > *System information*; tas būs nepieciešams nākamajā darbībā.

7. Rediģējiet Zabbix bridge adaptera konfigurācijas failu:

```
ZabbixAPIURL=https://example.com/zabbix/api_jsonrpc.php
ServerId=019fb1f9-22bd-74ea-bc03-51efb8b76aba
DataDir=/var/lib/zabbix-bridge-adapter
```

Attention:
Direktorijai, kas norādīta parametrā `DataDir`, jābūt ar 0700 atļaujām, un visiem tajā glabātajiem failiem jābūt ar 0600 atļaujām.

8. Pārskatiet un pieņemiet Zabbix bridge adaptera galalietotāja licences līgumu (EULA):

```
zabbix_bridge_adapter license show
sudo -u zabbix-bridge-adapter zabbix_bridge_adapter -c /etc/zabbix/zabbix_bridge_adapter.conf license acce
```

9. Palaidiet Zabbix bridge adapteri:

```
systemctl start zabbix-bridge-adapter
```

Attention:
Palaišanas laikā Zabbix bridge adapteris izmanto servera ID, lai reģistrētu jūsu Zabbix serveri centralizēti mitinātajā Zabbix Bridge. Reģistrācijas stāvoklis tiek glabāts stāvokļa failā `DataDir`. Ja atkārtoti instalējat bridge adapteri un stāvokļa fails tiek pazaudēts, tā paša Zabbix servera reģistrācija var neizdoties. Lai no tā izvairītos, pirms bridge adaptera atkārtotas instalēšanas izveidojiet stāvokļa faila dublējumkopiju un pēc tam to atjaunojiet.

Iestatiet push paziņojumus Lai saņemtu brīdinājumus savā mobilajā ierīcē, kad jūsu uzraudzītajā vidē ir nepieciešama uzmanība:

1. Izveidojiet push paziņojumu multivides tipu (vai iespējojiet noklusējuma tipu).
2. Izveidojiet darbību (vai iespējojiet noklusējuma trigeru darbību *Ziņot par problēmām Zabbix administratoriem*).
3. Pievienojiet ierīci, lai multivides tips tiktu automātiski pievienots jūsu lietotājam (vai manuāli pievienojiet lietotāja multividi).

Ierīču pievienošana Lai savā ierīcē izmantotu Zabbix mobilo lietotni, šī ierīce ir jāpievieno Zabbix lietotāja saskarnē. Norādījumus skatiet:

- *Lietotāji* > *Ierīces* (kā *Super admin* pievienojiet savas vai cita lietotāja mobilās ierīces)
- *Lietotāja iestatījumi* > *Ierīces* (kā jebkurš lietotājs pievienojiet savas mobilās ierīces)

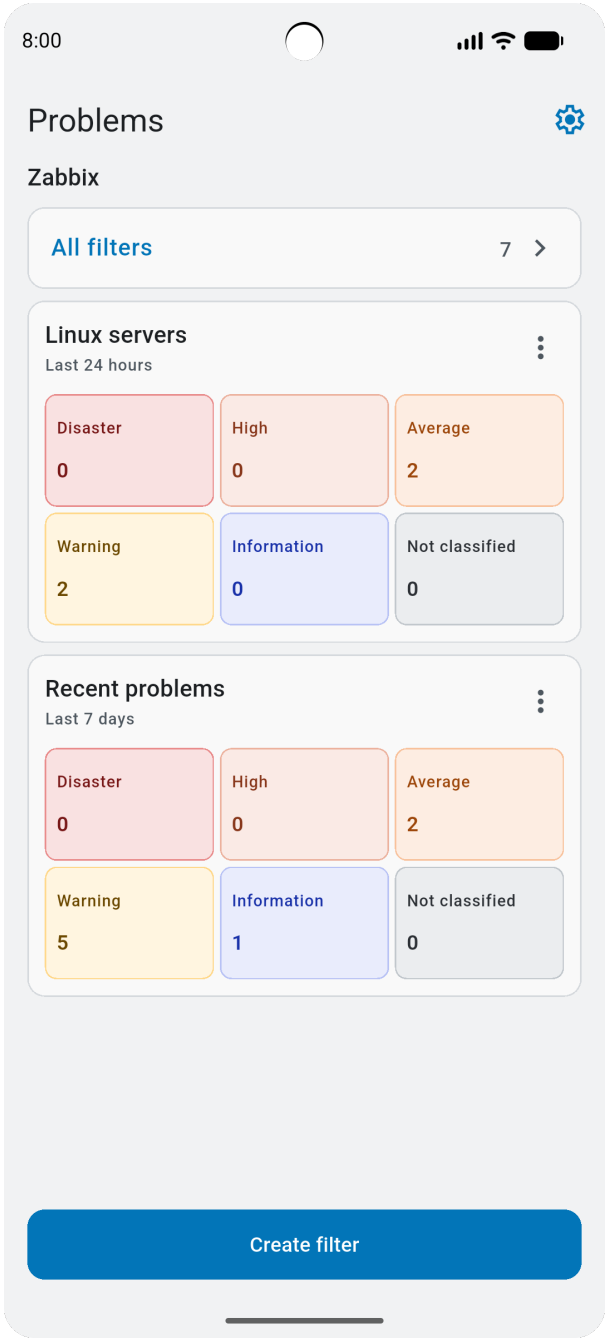
Lietotnes pārskats

Zabbix mobilā lietotne (drīzumā) ļauj pārraudzīt savu Zabbix vidi no mobilās ierīces.
Skatiet problēmas, pārbaudiet to detalizētu informāciju, atjauniniet problēmas un saņemiet pašpiegādes paziņojumus, kad nepieciešama uzmanība.

Šajā lapā sniegts mobilās lietotnes pārskats un informācija par tās lietošanu.

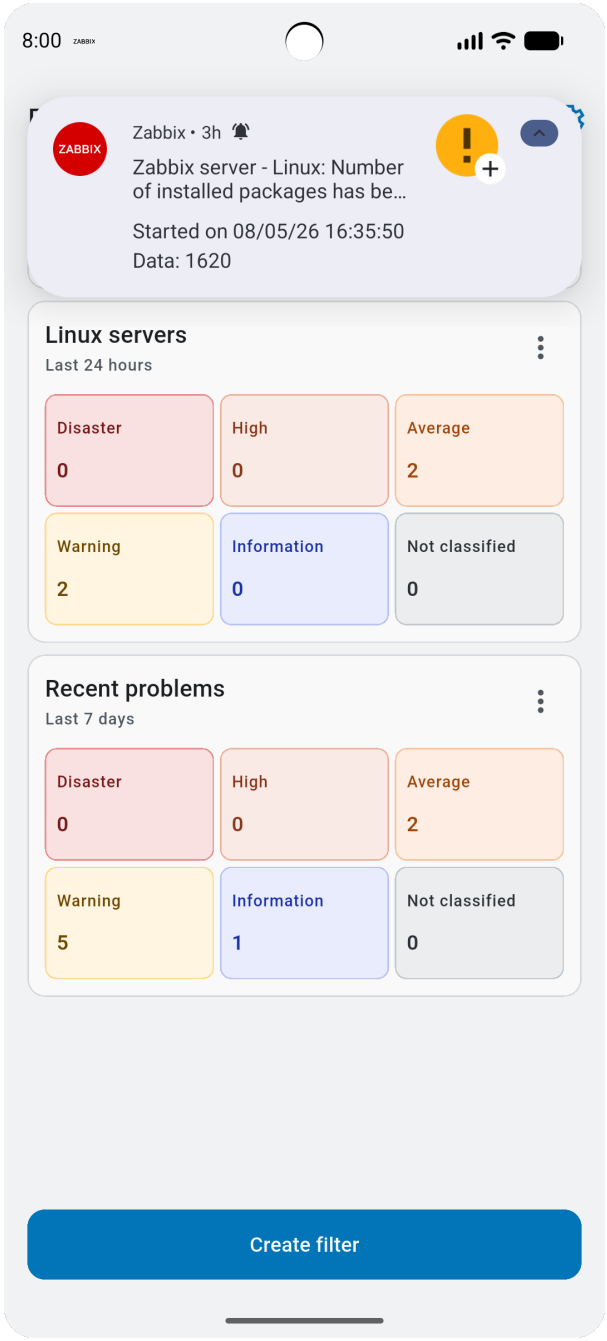
Problēmas Ekrānā *Problēmas* tiek rādīts problēmu pārskats, pamatojoties uz piespraustajiem filtriem. Tajā ir:

- Ikona *Iestatījumi* — pieskarieties tai, lai skatītu lietotnes iestatījumus.
- *Visi filtri* — pieskarieties, lai skatītu un pārvaldītu filtrus.
- Zabbix servera nosaukums (vai *Zabbix*, ja instalēšanas laikā Zabbix lietotāja saskarnē tas netika konfigurēts).
- Piespraustie filtri — pieskarieties to nosaukumam vai nopietnības pakāpei, lai atvērtu ekrānu *Problēmu saraksts*. Pieskarieties trīs punktu izvēlnei, lai pārvaldītu filtru: rediģētu, dublētu, atspraustu vai dzēstu to. Ņemiet vērā, ka piespraustie filtri ir sakārtoti alfabētiskā secībā un filtru *Pēdējās problēmas* nevar rediģēt vai noņemt.
- *Izveidot filtru* — pieskarieties, lai izveidotu jaunu filtru.



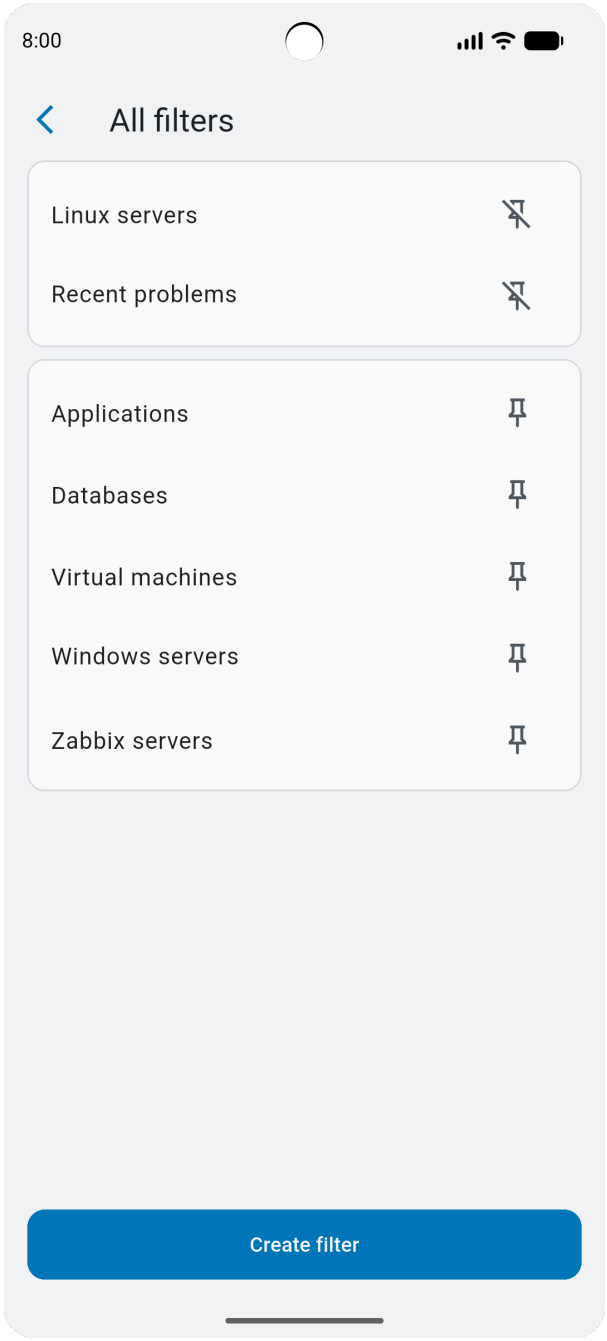
Paziņojumi Kad Zabbix serveris konstatē problēmu, tas var nosūtīt paziņojumu uz jūsu mobilo ierīci. Paziņojumā ir norādīts problēmas hosts, nosaukums un ziņojums, kas ģenerēts no ziņojumu veidnēm, kas konfigurētas push paziņojumu multivides tipam.

Pieskarieties paziņojumam, lai atvērtu ekrānu *Problēmas informācija*.



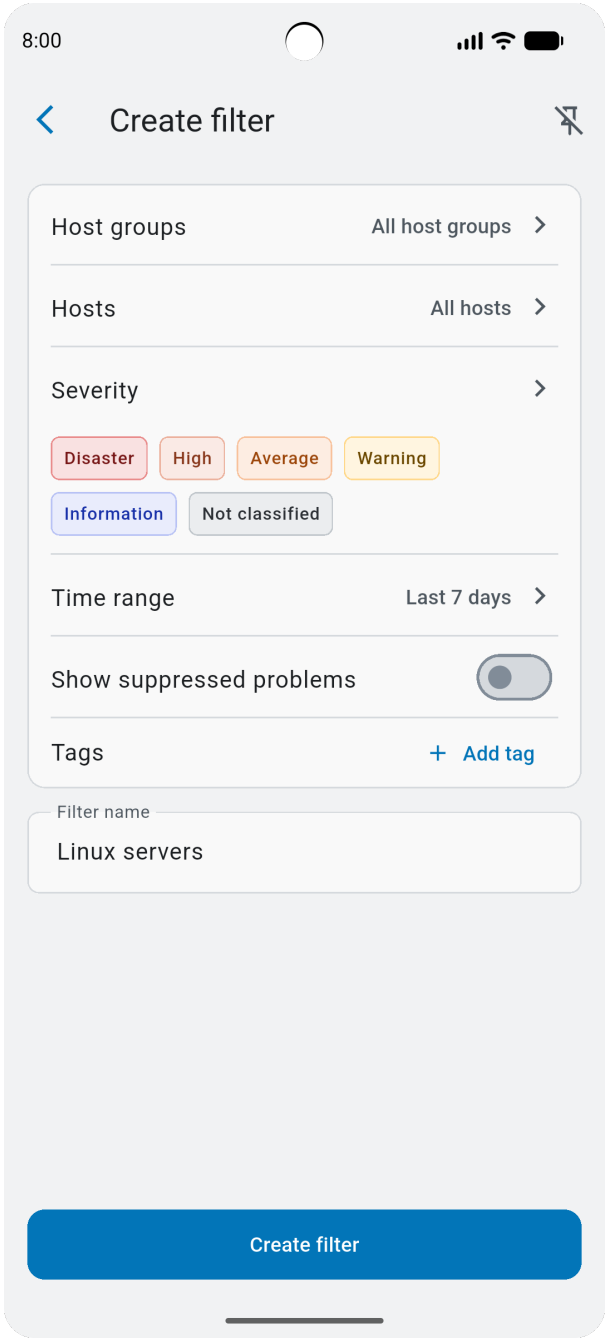
Visi filtri Ekrānā *Visi filtri* varat pārvaldīt filtrus. Tajā ir:

- Filtru nosaukumi - pieskarieties, lai rediģētu filtru.
- Piespraušanas ikonas - pieskarieties, lai piespraustu filtru ekrānam *Problēmas* vai noņemtu tā piespraušanu.
- *Izveidot filtru* - pieskarieties, lai izveidotu jaunu filtru.



Filtra izveide Ekrānā *Filtra izveide* varat izveidot filtru ar opcijām, kas līdzīgas tām, kuras pieejamas sadaļā *Monitoring* > *Problems* (skatiet informāciju). Varat arī iestatīt *Laika diapazonu*, cik ilgi rādīt neatrisinātās problēmas.

Pieskarieties *Izveidot filtru*, lai saglabātu filtru. Jaunie filtri tiek piesprausti ekrānam *Problēmas*; pieskarieties piespraušanas ikonai, lai to mainītu.



Problēmu saraksts Ekrānā *Problēmu saraksts* tiek parādītas problēmas, kas atbilst filtram (piemēram, *Nesenās problēmas*).

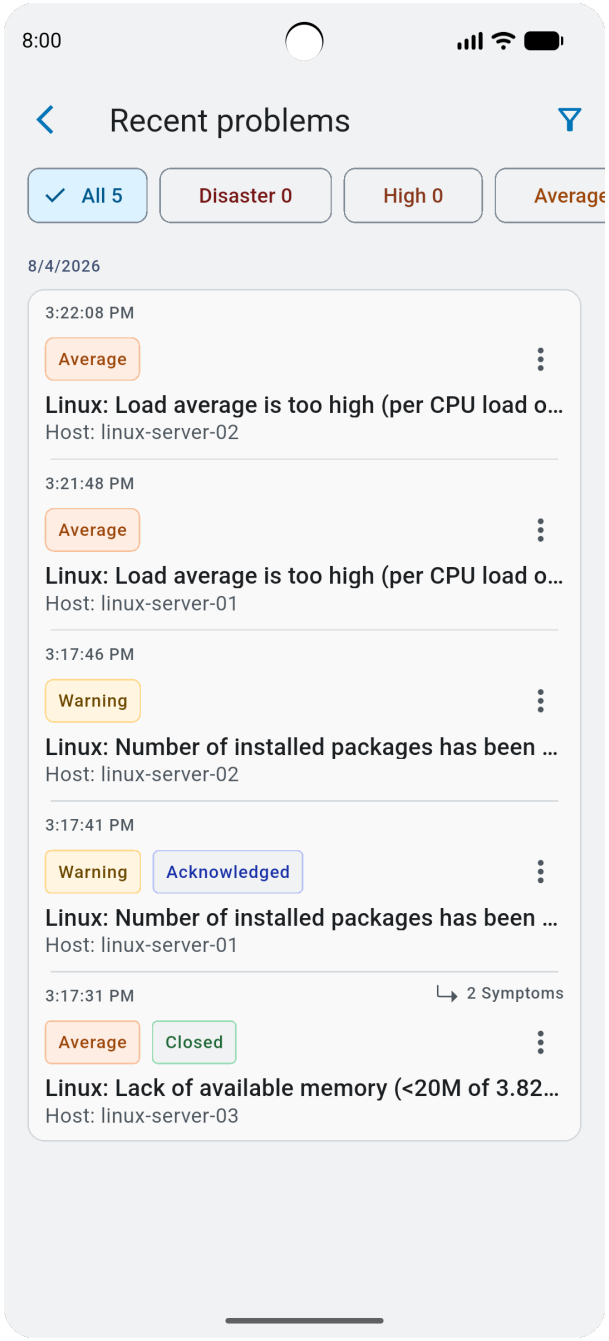
Pieskarieties filtra ikonai vai problēmas nopietnības līmenim, lai vēl vairāk sašaurinātu sarakstu.

Katrai problēmai varat skatīt:

- Sākuma laiku un datumu.
- Nopietnības līmeni, nosaukumu un hostu.
- Vai tā ir apstiprināta vai nesen aizvērta (automātiski atrisināta vai manuāli aizvērta).
- Vai tai ir simptomi — pieskarieties, lai skatītu simptomātisko problēmu sarakstu.

Pieskarieties problēmai, lai atvērtu ekrānu *Problēmas informācija*.

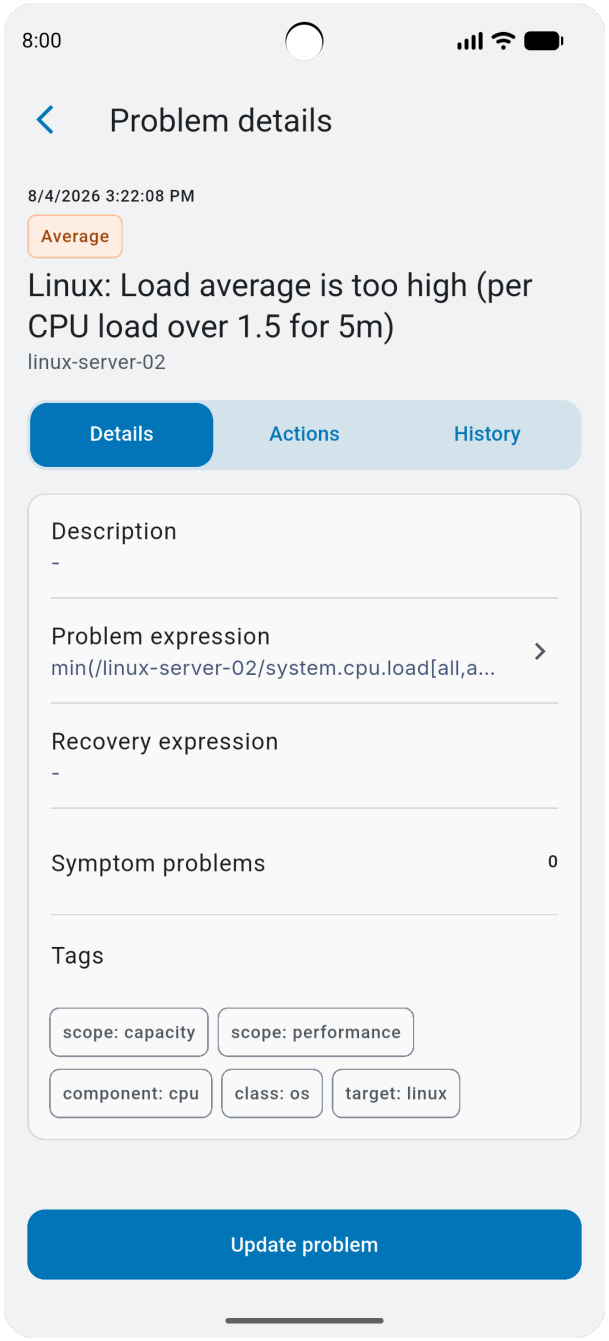
Pieskarieties trīs punktu izvēlnei, lai ātri apstiprinātu problēmu vai apslāpētu to uz jūsu izvēlētu laiku.



Problēmas informācija Ekrānā *Problēmas informācija* tiek rādīta informācija par problēmu, darbības un vēsture.

- Cilne *Informācija* - rāda problēmas aprakstu, izteiksmes, simptomu problēmas (pieskarieties, lai skatītu problēmu sarakstu) un tagus.
- Cilne *Darbības* - rāda problēmas laika grafiku: kad tā sākās vai tika apstiprināta, vai tika nosūtīts paziņojums un citas darbības.
- Cilne *Vēsture* - rāda problēmas ilgumu, ilgumu un atkopšanas laiku (vai –, ja problēma joprojām turpinās).

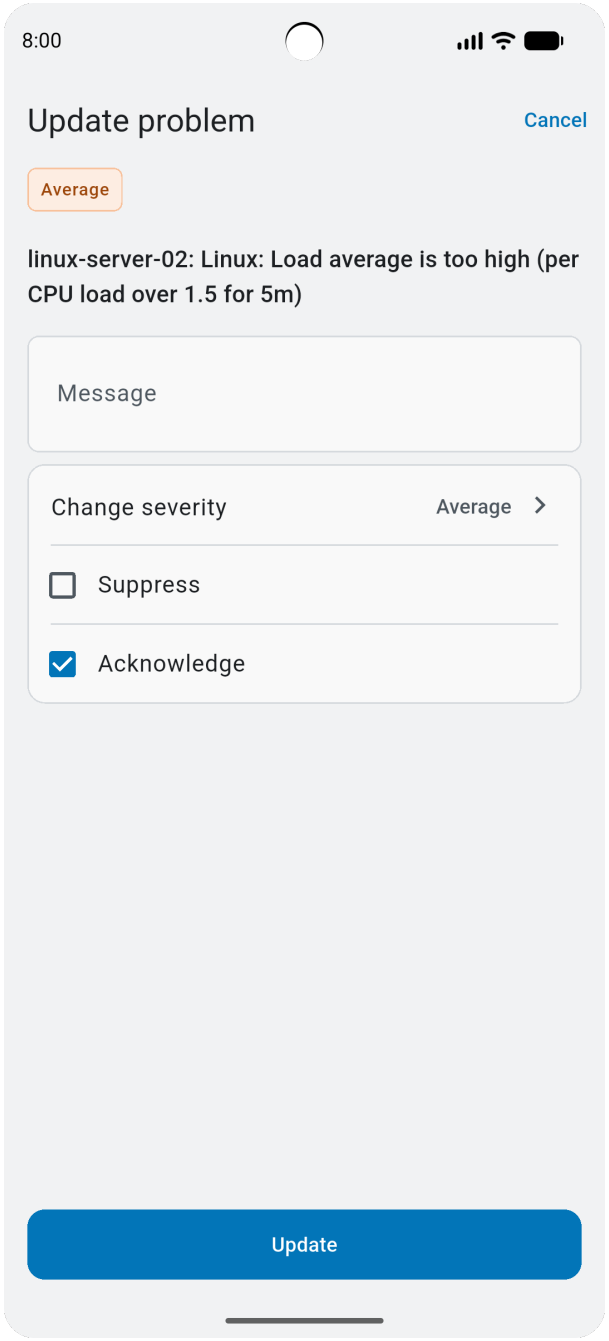
Pieskarieties *Atjaunināt problēmu*, lai atjauninātu problēmu.



Problēmas atjaunināšana Ekrānā *Problēmas atjaunināšana* varat atjaunināt problēmu:

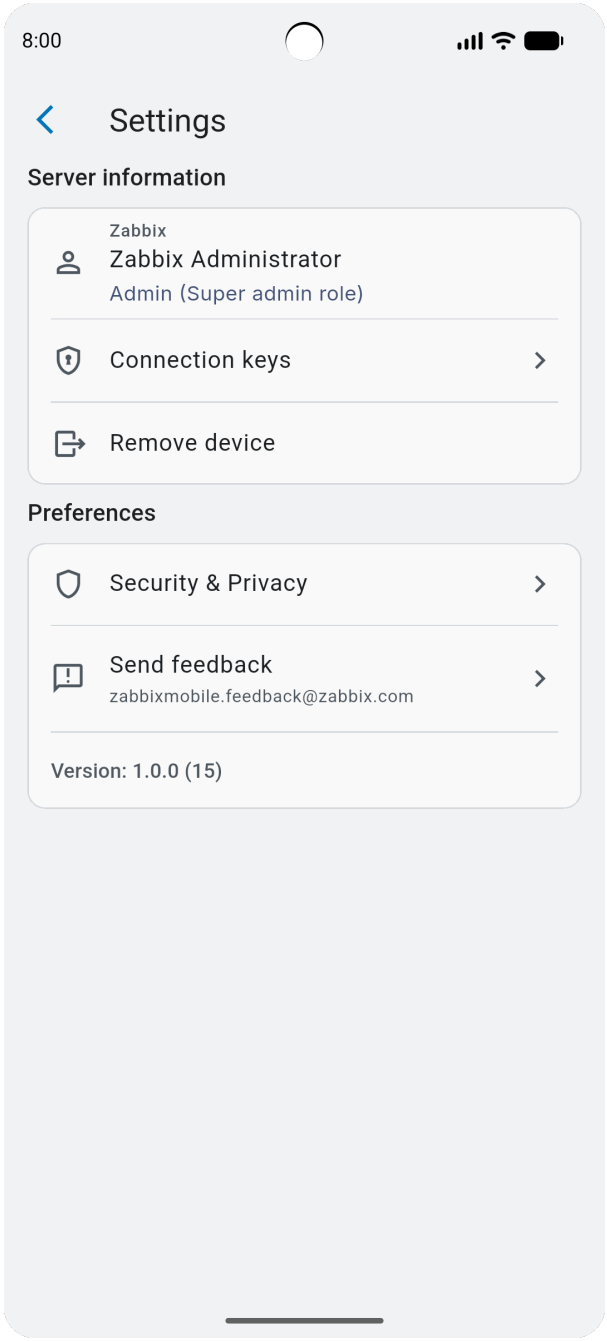
- Pievienot ziņojumu.
- Mainīt problēmas nopietnību.
- Apturēt problēmu uz jūsu izvēlētu laiku.
- Apstiprināt vai atcelt problēmas apstiprināšanu.
- Aizvērt problēmu.

Pieskarieties *Atjaunināt*, lai saglabātu izmaiņas.



Iestatījumi Ekrānā *Iestatījumi* varat piekļūt lietotnes iestatījumiem. Tajā ir ietverta šāda informācija:

- Zabbix servera nosaukums (vai *Zabbix*, ja instalēšanas laikā Zabbix lietotāja saskarne netika konfigurēta).
- Tā lietotāja vārds, tips un loma, kuram ir pievienota mobilā ierīce.
- *Savienojuma atslēgas* - atveriet, lai skatītu un kopētu servera ID, ierīces ID un citu informāciju.
- *Noņemt ierīci* - pieskarieties, lai noņemtu mobilo ierīci no Zabbix.
- *Drošība un konfidencialitāte* - atveriet, lai iespējotu lietotnes bloķēšanu un pārvaldītu diagnostikas datu kopīgošanu.
- *Nosūtīt atsauksmes* - atveriet, lai nosūtītu atsauksmes Zabbix.
- Lietotnes versija.



Droša piekļuve no mobilajām ierīcēm

Šajā lapā ir sniegta informācija par labāko praksi Zabbix mobilās lietotnes un Zabbix bridge adaptera drošības nodrošināšanai.

Mobilo ierīču aizsardzība Lai neļautu citām personām atvērt lietotni jūsu mobilajā ierīcē, iespējojiet lietotnes bloķēšanu lietotnes sadaļā *Iestatījumi*.

Ja ierīce ir pazaudēta vai apdraudēta, noņemiet to Zabbix lietotāja saskarnē. Tādējādi tiek atsaukti un noņemti tās piekļuves akreditācijas dati, un ierīce vairs nevar piekļūt Zabbix. Atkarībā no jūsu lietotāja lomas atļaujām ierīces varat noņemt šādās sadaļās:

- *Lietotāja iestatījumi > Ierīces* - jebkurš lietotājs var noņemt savas ierīces (nepieciešama atļauja *Pārvaldīt savas ierīces*).
- *Lietotāji > Ierīces* - *Super administrators* var noņemt jebkura lietotāja ierīces (nepieciešama atļauja *Pārvaldīt lietotāju ierīces*).

Varat arī ierobežot, kurām Zabbix daļām var piekļūt no mobilās ierīces. Lai to izdarītu, piešķiriet lietotājiem lietotāja lomu, kas nodrošina tikai lietotnei nepieciešamās atļaujas. Minimālās atļaujas ir šādas:

Sadaļa	Konfigurācija
Piekļuve lietotāja saskarnes elementiem	Notīriet visas izvēles rūtiņas.
Piekļuve pakalpojumiem	Iestatiet *Lasīšanas un rakstīšanas piekļuve pakalpojumiem* un *Tikai lasīšanas piekļuve pakalpojumiem* uz `None`.
Piekļuve moduļiem	Notīriet visas izvēles rūtiņas.
Piekļuve API	Notīriet izvēles rūtiņu *Iespējots*.
Piekļuve darbībām	Notīriet visas izvēles rūtiņas. Lai lietotāji lietotnē varētu atjaunināt problēmas, atzīmējiet arī nepieciešamās darbības: - *Pievienot problēmu komentārus* - *Mainīt nopietnību* - *Apstiprināt problēmas* - *Apturēt problēmas* - *Aizvērt problēmas* - *Izveidot un rediģēt savus multivides līdzekļus* Ņemiet vērā, ka problēmas, kuras lietotāji var redzēt lietotnē, ir atkarīgas arī no viņu piekļuves hostiem.
Ierīces	Atzīmējiet izvēles rūtiņu *Iespējots*. Lai lietotāji varētu pievienot un noņemt savas mobilās ierīces, atzīmējiet arī izvēles rūtiņu *Pārvaldīt savas ierīces*; pretējā gadījumā *Super administratoram* savas ierīces būs jāpārvalda sadaļā *Lietotāji > Ierīces*.

Zabbix bridge adaptera aizsardzība Zabbix bridge adapteris glabā savus datus (stāvokļa failu un šifrēšanas atslēgas) direktorijā, kas norādīta Zabbix bridge adaptera konfigurācijas faila parametrā `DataDir`. Pārliecinieties, ka direktorijai ir 0700 atļaujas un visiem tajā glabātajiem failiem ir 0600 atļaujas.

Zabbix bridge adapteris pārsūta pieprasījumus no mobilajām ierīcēm uz Zabbix API. Lai ierobežotu API metodes, kuras var izsaukt, izmantojot Zabbix bridge adapteri, izmantojiet tā parametru `AllowedAPIMethods`. Piemēram, atļaujiet tikai minimālo mobilo lietotni nepieciešamo metožu kopu vai pat mazāku kopu, ja nevēlaties, lai tiktu izmantotas dažas mobilās lietotnes funkcijas (piemēram, neatļaujiet `event.acknowledge`, ja mobilo lietotņu lietotājiem nevajadzētu atjaunināt problēmas).

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 vācēja) izstrāde Zabbix aģentam 2.

Pielāgots spraudnis paplašina Zabbix aģenta 2 funkcionalitāti, pārsniedzot nodrošināto iebūvēto spraudņu un ielādējamo spraudņu iespējas.

Katrs spraudnis ir *Go* pakotne, kas definē struktūru un īsteno vienu vai vairākas saskarnes (*Exporter*, *Configurator*, *Runner*). Papildinformāciju skatiet sadaļās Spraudņu saskarnes un Savienojumu diagramma.

Šī rokasgrāmata palīdzēs izveidot pielāgotu ielādējamu spraudni.

Papildu norādījumus skatiet arī šādās repozitorijās:

- Piemēra spraudnis
- 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 nepieciešamo ietvaru spraudņu izstrādei un nodrošina, ka visiem spraudņiem ir konsekventa struktūra.

1. Iestatiet pamata izpildes plūsmu.

Sāciet, definējot spraudņa galveno izpildes plūsmu. Pievienojiet tālāk norādīto kodu failam `main.go`:

```
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 ietvers spraudņa galveno loģiku.

2. Izpētiet spraudņu saskarnes.

Zabbix aģenta 2 spraudnim ir jābūt attēlotam kā struktūrai, kas implementē saskarnes no pakotnes `golang.zabbix.com/sdk/plugin`:

- *Accessor* - definē būtiskās metodes, kas jāimplementē visiem spraudņiem, piemēram, spraudņa nosaukuma iestatīšanu un vienuma atslēgas taimautu apstrādi.
- Vienu vai vairākas no tālāk norādītajām funkcionālajām spraudņa saskarnēm:
  - *Exporter* - veic aptauju un atgriež vērtību (vai vērtības), neko vai kļūdu; bieži tiek izmantots kopā ar *Collector* saskarni.
  - *Collector* - pārvalda periodisku datu apkopošanu.
  - *Runner* - definē spraudņa palaišanas un apturēšanas procedūras.

- *Watcher* - nodrošina neatkarīgu metrikas aptauju, apejot aģenta iekšējo plānotāju; noder uz slazdiem vai notikumiem balstītai uzraudzībai.
- *Configurator* - definē, kā spraudnis nolasa un piemēro konfigurācijas iestatījumus.

Šīs saskarnes varat implementēt paši vai izmantot Zabbix Go SDK nodrošinātās noklusējuma saskarnes, pēc vajadzības tās modificējot. Šajā pamācībā tiek izmantotas noklusējuma implementācijas.

3. Izveidojiet spraudņa struktūru.

Tagad importējiet pakotni *plugin* un izveidojiet struktūru `myIP`, kas ietver struktūru `plugin.Base`:

```
import "golang.zabbix.com/sdk/plugin"

type myIP struct {
    plugin.Base
}
```

Struktūra `myIP` pašlaik atbilst saskarnei *Accessor*. Metode vienas no funkcionālajām spraudņa saskarnēm, *Exporter*, implementēšanai 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. darbība: izveidojiet un konfigurējiet spraudni

1. Lai izveidotu spraudni, palaidiet:

```
go mod tidy
go build
```

Tam vajadzētu izveidot izpildāmo failu `myip` pašreizējā direktorijā.

2. Konfigurējiet Zabbix aģents 2, lai izmantotu spraudni:

```
echo "Plugins.MyIP.System.Path=$PATH_TO_THE_MYIP_PLUGIN_EXECUTABLE" > /etc/zabbix/zabbix_agent2.d/plugins.
```

Aizstājiet `$PATH_TO_THE_MYIP_PLUGIN_EXECUTABLE` ar ceļu uz `myip`, kas tika izveidots 5. darbībā.

Konfigurācijas parametra nosaukumā norādītajam spraudņa nosaukumam (*MyIP* šajā pamācībā) ir jāsakrīt ar spraudņa nosaukumu, kas definēts funkcijā *plugin.RegisterMetrics()*.

3. Lai pārbaudītu spraudni un tā `myip` vienumu, palaidiet:

```
zabbix_agent2 -c /etc/zabbix/zabbix_agent2.conf -t myip
```

Izvadē jābūt jūsu hosta ārējai IP adresei, un tai vajadzētu izskatīties līdzīgi šādi:

```
myip                                          [s|192.0.2.1]
```

Tādējādi 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.

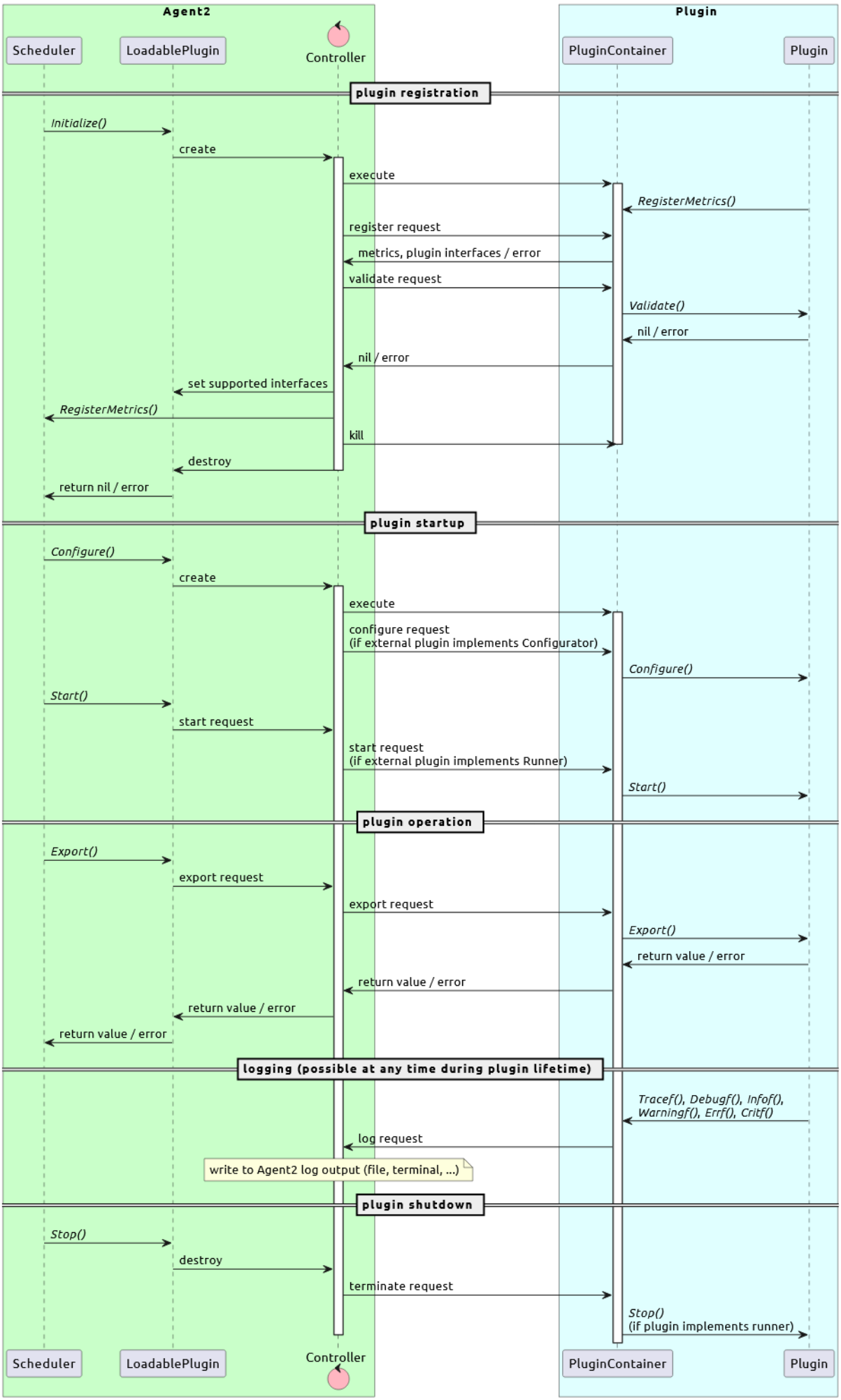
plugin.Exporter *Exporter* ir vienkāršākais interfeiss, kas veic aptauju un atgriež vērtību (vērtības), neko vai kļūdu. Tas pieņem iepriekš parsētu vienuma atslēgu, parametrus un kontekstu. Piekļuve visiem pārējiem spraudņa interfeisiem ir ekskluzīva, un nevienu metodi nevar izsaukt, ja spraudnis jau 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 Zabbix aģenta IP adresi un portu.
2. Izsauciet `get()` metodi `Getter` instancei, norādot vienuma atslēgu, kuru vēlaties izgūt.

Piemēram, lai pieprasītu datus vienumam `system.uname`:

```
agent = Getter(host='192.0.2.1', port=10050)
response = agent.get('system.uname')
```

Izmantojot nestandarta IP

Ja serverim, kurā darbojas jūsu skripts, ir vairākas IP adreses, varat norādīt `source_ip`, ko `Getter` izmantos, veidojot savienojumu ar Zabbix aģentu:

```
agent = Getter(
    host='192.0.2.1',
    port=10050,
    source_ip='10.10.7.1'
)
```

Izmantojot taimautu

Varat iestatīt `Getter` atbildes `timeout`, lai kontrolētu, cik ilgi jūsu skriptam jāgaida Zabbix aģenta atbilde, pirms tas pārtrauc mēģinājumu:

```
agent = Getter(
    host='192.0.2.1',
    port=10050,
    timeout=30
)
```

Šifrēšanas izmantošana

`Getter` neietver iebūvētu šifrēšanas atbalstu, taču to var nodrošināt, izveidojot ietvaru, izmantojot trešo pušu bibliotēkas:

```
def psk_wrapper(sock, tls):
    # ...
    # Implementation of TLS PSK wrapper for the socket
    # ...
```

```
agent = Getter(
    host='192.0.2.1',
    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 savākt vērtības, negaidot, līdz katra no tām tiek saņemta. Tas var padarīt jūsu skriptu efektīvāku, ja tam jāapkopo daudzas vērtības vai ja dažu vērtību savākšana aizņem ilgu laiku.

Izmantojot asinhrono režīmu, pastāv svarīgas atšķirības salīdzinājumā ar sinhrono režīmu:

- Importējiet Python asyncio moduli (vispirms jums jāinstalē nepieciešamās atkarības).
- Importējiet AsyncGetter nevis Getter.
- Rakstiet savu kodu async funkcijā.
- Izmantojiet await, izsaucot get() metodi.

Piemēram, lai savāktu vienu vērtību, izmantojot asinhrono režīmu:

```
### 1. Import asyncio asinhronajam režīmam un AsyncGetter no zabbix_utils:
import asyncio
from zabbix_utils import AsyncGetter

### 2. Definējiet galveno async funkciju, kurā tiks izpildīti visi datu pieprasījumi:
async def main():
    aģents = AsyncGetter(host='192.0.2.1', port=10050)

    # 3. Iegūstiet system.uname vērtību no Zabbix aģents (jāizmanto await):
    response = await aģents.get('system.uname')

    # 4. Izdrukājiet Zabbix aģents atgriezto vērtību:
    print(response.value)

### 5. Palaidiet async main() funkciju, izmantojot asyncio notikumu ciklu:
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 parametrus Server vai ServerActive no lokālā Zabbix aģenta vai aģenta 2 konfigurācijas faila. Šādos gadījumos, veidojot Sender instanci, savienojuma parametri nav jānorāda:

```
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 Server adrese 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 Parauga moduļa izveides process sastāv no šādiem soļiem (ja pieejams, noklikšķiniet uz faila vai mapes nosaukuma, lai skatītu papildu informāciju par šo soli):

1. Izveidojiet jaunu moduļa direktoriju iekš **zabbix/ui/modules/**.
2. Pievienojiet manifest.json failu ar moduļa metadatiem.
3. Izveidojiet *views* mapi un definējiet moduļa skatu(-s).
4. Izveidojiet *actions* mapi un definējiet moduļa darbību(-s).

5. Izveidojiet failu Module.php (vai Widget.php dashboard logrīkiem) un definējiet inicializācijas un notikumu apstrādes noteikumus.
6. Izveidojiet *assets* mapi JavaScript failiem (ievietojiet tos *assets/js*), CSS stiliem (ievietojiet tos *assets/css*) vai 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 uzreiz pēc manifest.json faila izveides. Kad modulis ir iespējots, jūs varat nekavējoties priekšskatīt visas izmaiņas, kas veiktas moduļa failos, vienkārši atsvaidzinot Zabbix lietotāja saskarni.

manifest.json

Jebkuram modulim ir nepieciešams fails manifest.json. Fails jānovieto moduļa primārajā direktorijā (piemēram, *zabbix/ui/modules/module_nam*

Vismaz minimālajā gadījumā manifest.json jānorāda šādi lauki:

```
{
    "manifest_version": 2.0,
    "id": "my_ip_address",
    "name": "My IP Address",
    "namespace": "MyIPAddress",
    "version": "1.0"
}
```

Parametri, kas atbalstīti **manifest.json** (noklikšķiniet uz parametra nosaukuma, lai skatītu detalizētu aprakstu):

Parameter	Description	Required
manifest_version	Moduļa manifestācijas versija.	Yes
id	Unikāls moduļa ID.	
name	Moduļa nosaukums, kas tiks parādīts administrēšanas sadaļā.	
namespace	PHP namespace moduļa klasēm.	
version	Moduļa versija.	
type	Moduļa tips. Widgetam jābūt iestatītam uz *widget*	Yes for widgets, otherwise no
widget	Widget konfigurācija. Tiek izmantota tikai widgetiem.	
actions	Darbības, kas jāreģistrē kopā ar moduli.	
assets	CSS stili un JavaScript faili, kas jāiekļauj.	No
author	Moduļa autors.	
config	Pielāgoto moduļa opciju noklusējuma vērtības.	
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 Informācijas 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 konkrētai lietotāja saskarnes tēmai.

Pieejamās tēmas un to atribūtu vērtības:

- **Zila** - [theme='blue-theme']
- **Zila (klasiskā)** - [theme='blue-classic-theme']
- **Tumša** - [theme='dark-theme']
- **Tumši zila** - [theme='dark-blue-theme']
- **Tumša (klasiskā)** - [theme='dark-classic-theme']
- **Gaiša ar augstu kontrastu** - [theme='hc-light']
- **Tumša ar augstu kontrastu** - [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 *Administration* > *General* > *Modules* lapu un noklikšķiniet uz *Scan directory*.



Atrodiet savu moduli sarakstā un aktivizējiet to.

Lai aktivizētu moduli, noklikšķiniet uz hipersaites *Disabled* - moduļa statuss tiks mainīts uz *Enabled*.

Noklikšķiniet 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
<?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
<?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

- CWidgetFormView* klase ir nepieciešama, lai norādītu *WidgetForm* klasē definēto lauku attēlošanas loģiku, nosakot to izskatu un darbību, kad tie tiek renderēti konfigurācijas skatā.

CWidgetFormView klase atbalsta šādas metodes:

- *addField()* - kā parametru saņem *CWidgetFieldView* klases instanci; katrai *CWidgetField* klasei ir atbilstoša *CWidgetFieldView* klase, ko izmantot logrīka konfigurācijas skatā.
- *addFieldset()* - saņem *CWidgetFieldsGroupView* klases instanci, kas apvieno laukus sakļaujamā konteinerā.
- *addFieldsGroup()* - saņem *CWidgetFormFieldsetCollapsibleView* klases instanci, kas vizuāli (ar apmali) apvieno laukus grupā.
- *includeJsFile()* - ļauj logrīka konfigurācijas skatam pievienot JavaScript failu.
- *addJavaScript()* - ļauj pievienot iekļautu JavaScript kodu, kas tiks izpildīts, tiklīdz logrīka konfigurācijas skats būs ielādēts.

CWidgetFormView klasei jāatrodas direktorijā *views*.

views/widget.edit.php piemērs

```
<?php

/**
 * My custom widget form view.
 *
 * @var CView $this
 * @var array $data
 */

use Modules\MyCustomWidget\Includes\WidgetForm;

(new CWidgetFormView($data))
    ->addField(
        (new CWidgetFieldMultiSelectItemView($data['fields']['itemid']))->setPopupParameter('numeric', tru
    )
    ->addFieldset(
        (new CWidgetFormFieldsetCollapsibleView(_('Advanced configuration')))
            ->addField(
                new CWidgetFieldTextBoxView($data['fields']['description'])
            )
            ->addField(
                new CWidgetFieldColorView($data['fields']['chart_color'])
            )
    )
    ->includeJsFile('widget.edit.js.php')
    ->addJavaScript('my_custom_widget_form.init('.json_encode([
        'color_palette' => WidgetForm::DEFAULT_COLOR_PALETTE
    ]).');')
    ->show();
```

JavaScript

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

- CWidgetField* klase ir bāzes klase, no kuras tiek mantotas visas veidlapas 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 atlases lauks.
CWidgetFieldDatePicker	string	Datuma atlases lauks.
CWidgetFieldIntegerBox	int32	Lauks vesela skaitļa ievadīšanai. Var izmantot minimālo un maksimālo vērtību konfigurēšanai.
CWidgetFieldLatLng	string	Teksta lauks ar komatiem atdalīta platuma, garuma un kartes mērogojuma līmeņa ievadīšanai.
CWidgetFieldMultiSelect	-	Bāzes klase vairākatlases laukiem, ko paplašina visas `CWidgetFieldMultiSelect*` klases.
CWidgetFieldMultiSelectAction	ID	Vairākatlases lauks darbību atlasei no sadaļā *Alerts > Actions* definēto darbību saraksta.
CWidgetFieldMultiSelectGraph	ID	Vairākatlases lauks pielāgoto grafiku atlasei.
CWidgetFieldMultiSelectGraphPrototype	ID	Vairākatlases lauks pielāgoto grafiku prototipu atlasei.
CWidgetFieldMultiSelectGroup	ID	Vairākatlases lauks hostu grupu atlasei.
CWidgetFieldMultiSelectHost	ID	Vairākatlases lauks hostu atlasei.
CWidgetFieldMultiSelectHostInventory	array of int32	Vairākatlases lauks hostu inventāra lauku atlasei.
CWidgetFieldMultiSelectItem	ID	Vairākatlases lauks vienumu atlasei.
CWidgetFieldMultiSelectItemPrototype	ID	Vairākatlases lauks vienumu prototipu atlasei.
CWidgetFieldMultiSelectMap	ID	Vairākatlases lauks karšu atlasei.
CWidgetFieldMultiSelectMediaType	ID	Vairākatlases lauks datu nesēju tipu atlasei.
CWidgetFieldMultiSelectOverrideHost	ID	Vairākatlases lauks hostu datu avota atlasei no informācijas paneļa vai cita logrīka.
CWidgetFieldMultiSelectService	ID	Vairākatlases lauks pakalpojumu atlasei.
CWidgetFieldMultiSelectSla	ID	Vairākatlases lauks SLA atlasei.
CWidgetFieldMultiSelectUser	ID	Vairākatlases lauks lietotāju atlasei.
CWidgetFieldNavTree	array of (multiple mixed)	Saglabā karšu navigācijas koka struktūru izmantošanai logrīkā *Map navigation tree*.
CWidgetFieldNumericBox	string	Lauks peldošā komata skaitļa ievadīšanai.
CWidgetFieldPatternSelect	-	Bāzes klase modeļu atlases laukiem, ko paplašina visas `CWidgetPatternSelect*` klases.

CWidgetField class	Database field type	Description
CWidgetFieldPatternSelectHost	string	Vairākatlases lauks hostu atlasei. Atbalsta hostu nosaukumu modeļus.
CWidgetFieldPatternSelectItem	string	Vairākatlases lauks vienumu atlasei. Atbalsta vienumu nosaukumu modeļus.
CWidgetFieldRadioButtonList	int32	Radiopogu grupa, kas sastāv no vienas vai vairākām radiopogām.
CWidgetFieldRangeControl	int32	Slīdnis vesela skaitļa tipa vērtības atlasei.
CWidgetFieldReference	string	Saglabā unikālu šī logrīka identifikatoru informācijas panelī; tiek izmantots logrīku saziņai.
CWidgetFieldSelect	int32	Nolaižamais atlases lauks.
CWidgetFieldSeverities	array of int32	*CWidgetFieldCheckBoxList* ar iepriekš iestatītiem trigeru nopietnības līmeņiem.
CWidgetFieldSparkline	array of (multiple mixed)	Ļauj konfigurēt dzirksteles diagrammu.
CWidgetFieldTags	array of (string, int32, string)	Ļauj konfigurēt vienu vai vairākas tagu filtra rindas.
CWidgetFieldTextArea	string	Teksta apgabals vairākrindu teksta ievadīšanai.
CWidgetFieldTextBox	string	Teksta lauks vienas rindas teksta ievadīšanai.
CWidgetFieldThresholds	array of (string, string)	Ļauj konfigurēt krāsu un skaitļu pārus.
CWidgetFieldTimePeriod	array of string	Laika perioda atlases lauks.
CWidgetFieldTimeZone	string	Nolaižamais saraksts ar laika joslām.
CWidgetFieldUrl	string	Teksta lauks URL ievadīšanai.

Note:
`int32` lauku maksimālais garums ir 10 rakstzīmes, `ID` lauku garums ir 20 rakstzīmes, bet `string` lauku garums ir 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

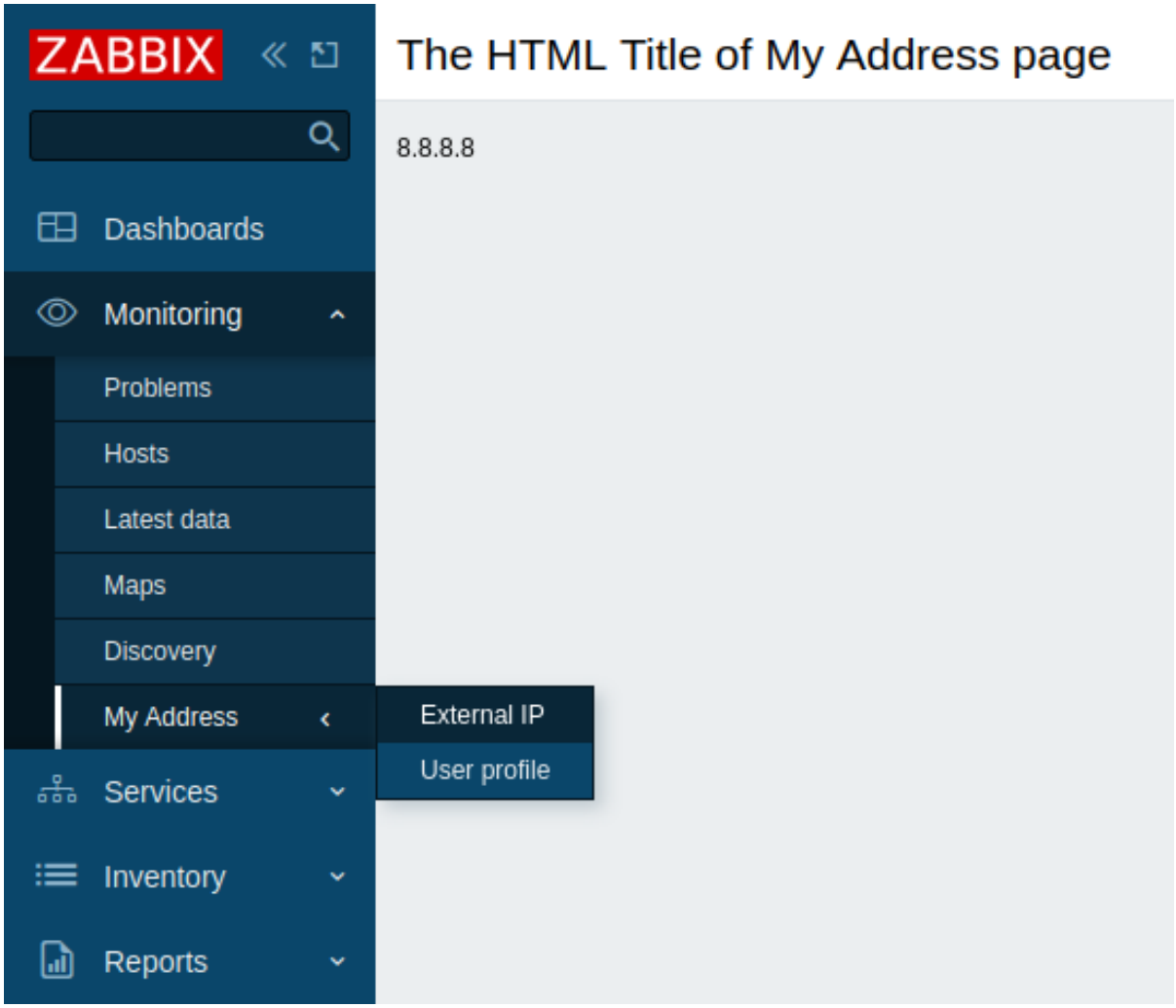
Šajā sadaļā ir iekļautas praktiskas soli pa solim izstrādātas pamācības, lai parādītu, kā izveidot pielāgotu moduli un logrīku Zabbix.

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 parāda atbildi — jūsu datora IP adresi. Lūk, kā izskatīsies pabeigtais modulis:



I daļa — Jauna izvēlnes sadaļa

Pievienot tukšu moduli Zabbix lietotāja saskarnei

1. Izveidojiet direktoriju *MyAddress* direktorijā *modules* savā Zabbix lietotāja saskarnes instalācijā (piemēram, *zabbix/ui/modules*).
2. Izveidojiet failu *manifest.json* ar pamata moduļa metadatiem (skatiet atbalstīto parametru aprakstu).

ui/modules/MyAddress/manifest.json

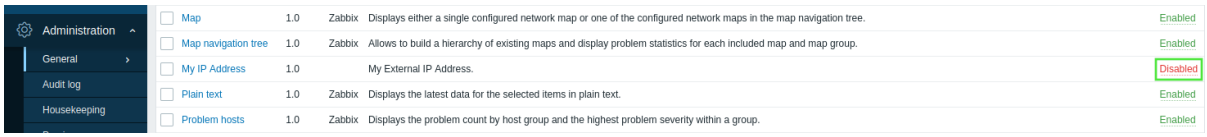
```
{
    "manifest_version": 2.0,
    "id": "my-address",
    "name": "My IP Address",
    "version": "1.0",
```

```
    "namespace": "MyAddress",
    "description": "My External IP Address."
}
```

3. Zabbix lietotāja saskarnē dodieties uz 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 uzskaitīts, skatiet sadaļu troubleshooting).



Modulis tagad ir reģistrēts lietotāja saskarnē. Tomēr tas vēl nav redzams, jo joprojām ir jādefinē moduļa funkcionalitāte. Kad pievienosiet saturu moduļa direktorijai, pēc lapas atsvaidzināšanas Zabbix lietotāja saskarnē uzreiz redzēsiet izmaiņas.

Izveidot izvēlnes sadaļu

1. Izveidojiet *Module.php* failu direktorijā *MyAddress*.

Šis fails implementē jaunu *Module* klasi, kas paplašina noklusējuma *CModule* klasi. *Module* klase ievietos jaunu *My Address* izvēlnes sadaļu galvenajā izvēlnē.

Metode *setAction()* norāda darbību, kas jāizpilda, noklikšķinot uz izvēlnes sadaļas. Sākumā varat izmantot iepriekš definēto darbību *userprofile.edit*, kas atvērs *User profile* lapu. Šīs pamācības III daļā jūs uzzināsiet, kā izveidot pielāgotu darbību.

ui/modules/MyAddress/Module.php

```
<?php

namespace Modules\MyAddress;

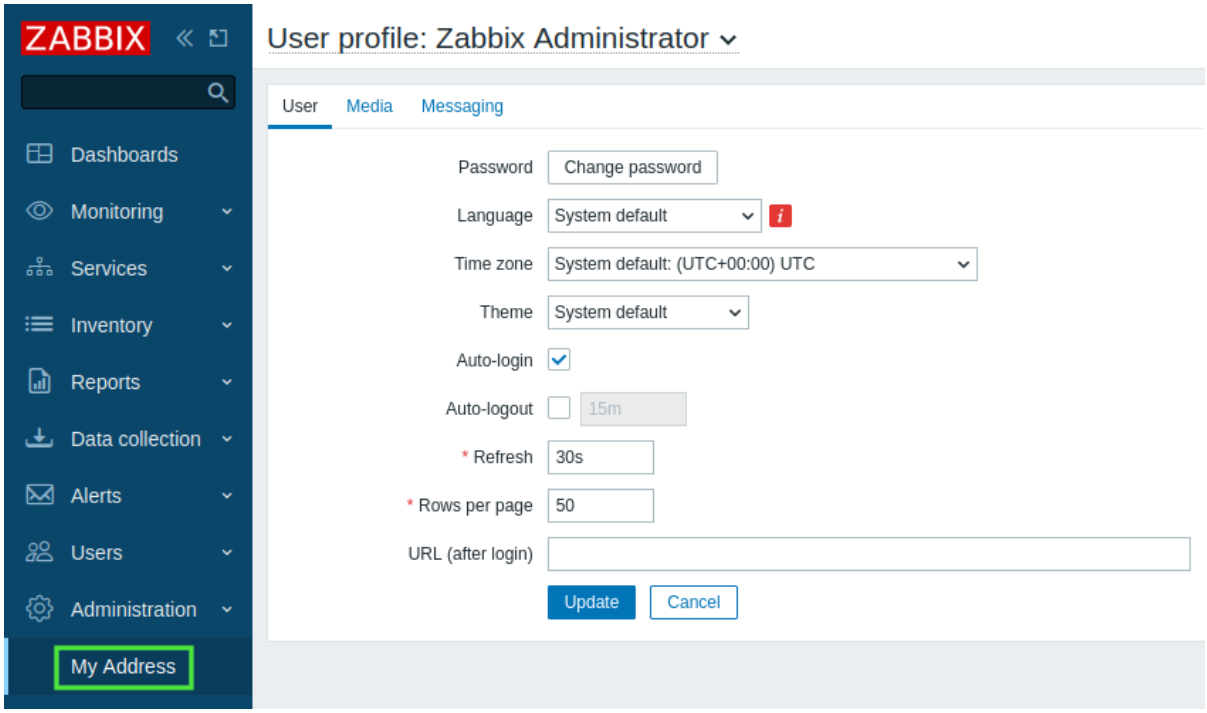
use Zabbix\Core\CModule,
    APP,
    CMenuItem;

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 pielāgotās diagrammas), `'problems.view'` (atver *Monitoring* > *Problems*) vai `'report.status'` (atver *System information* pārskatu).

3. Atsvaidziniet Zabbix lietotāja saskarni. Tagad Zabbix galvenās izvēlnes apakšā ir jauna sadaļa *My Address*. Noklikšķiniet uz *My Address*, lai atvērtu *User profile* 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

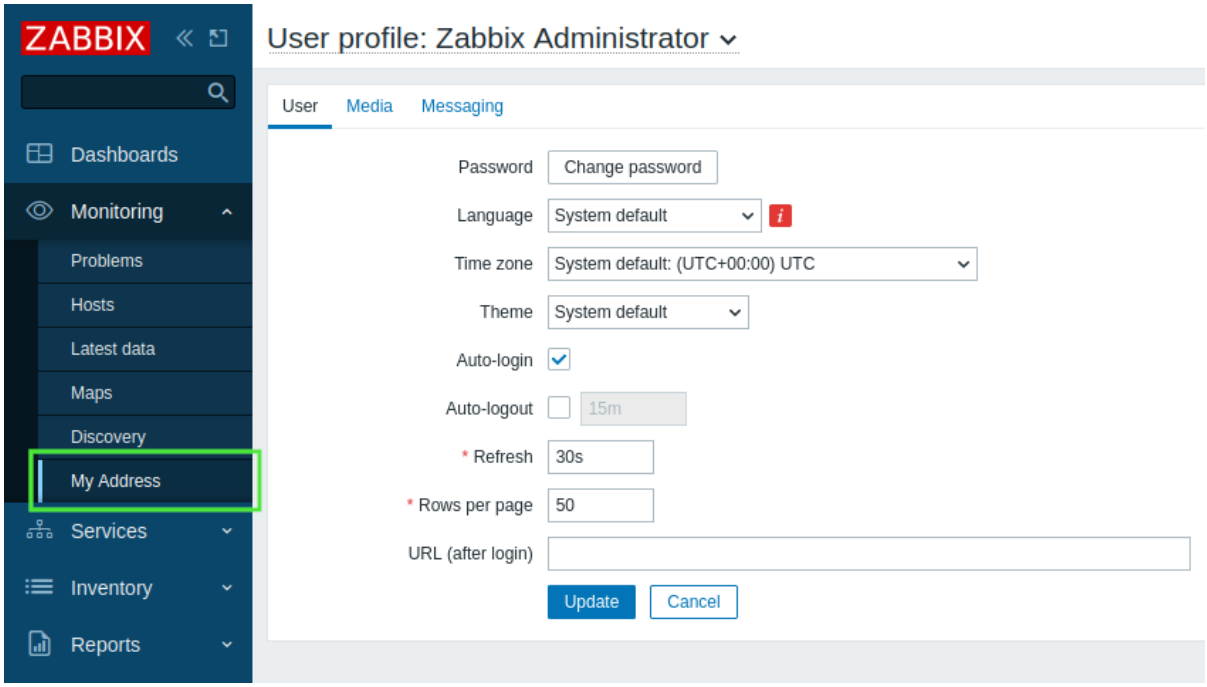
namespace Modules\MyAddress;

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 pievienotu ligzdotas lapas izvēlnes sadaļai *My Address*, vēlreiz atveriet un rediģējiet failu *Module.php*.

Šis solis izveido 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

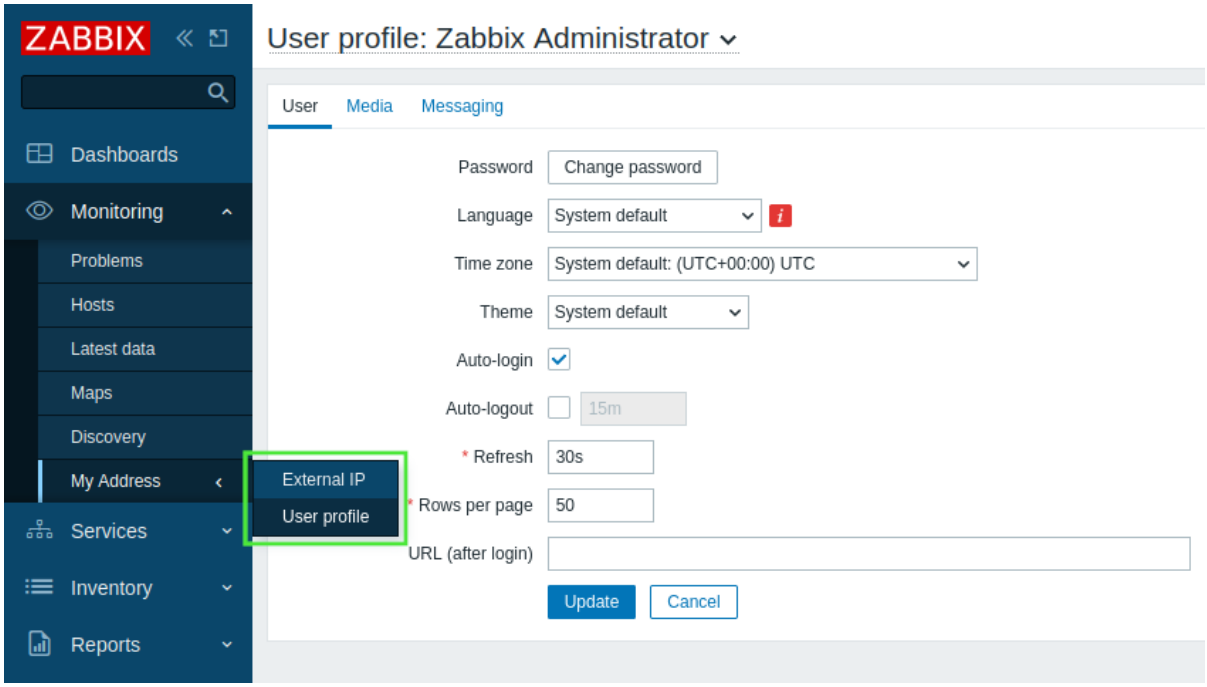
namespace Modules\MyAddress;

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:**
> Datiem jābūt organizētiem kā asociatīvam masīvam. Masīvs var būt daudzdimensionāls un var saturēt jebkurus 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 definēšanas. Sistēma 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();
```

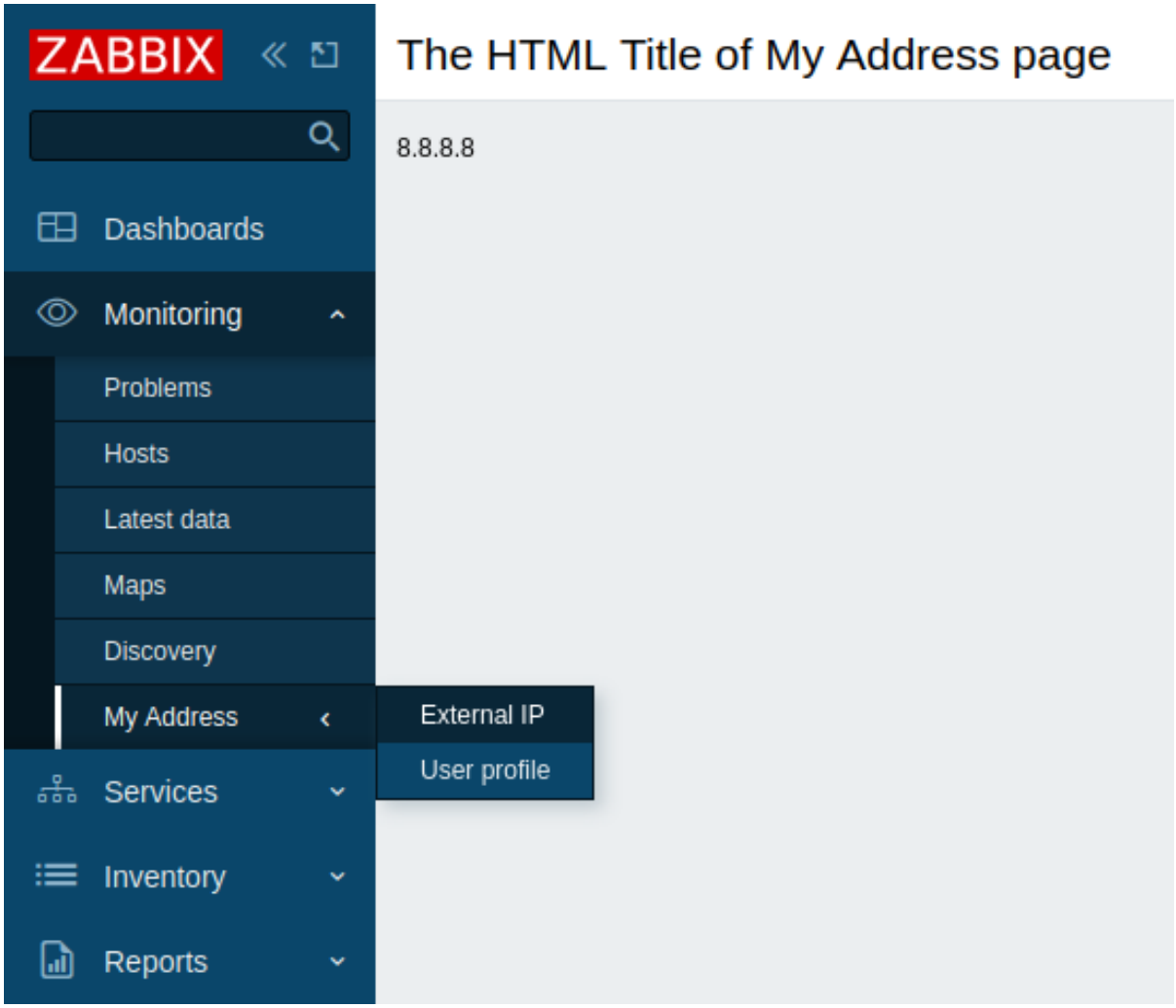
5. Moduļa darbība ir jāreģistrē failā *manifest.json*. Atveriet *manifest.json* un pievienojiet jaunu objektu `actions`, kas satur:

- darbības atslēgu ar darbības nosaukumu, kas rakstīts ar mazajiem burtiem (a-z) un kur vārdi ir atdalīti ar punktiem (piemēram, `my.address`);
- darbības klases nosaukumu (`MyAddress`) kā vērtību atslēgai `class` objektā `my.address`;
- darbības skata nosaukumu (`my.address`) kā vērtību atslēgai `view` objektā `my.address`.

ui/modules/MyAddress/manifest.json

```
{
    "manifest_version": 2.0,
    "id": "my-address",
    "name": "My IP Address",
    "version": "1.0",
    "namespace": "MyAddress",
    "description": "My External IP Address.",
    "actions": {
        "my.address": {
            "class": "MyAddress",
            "view": "my.address"
        }
    }
}
```

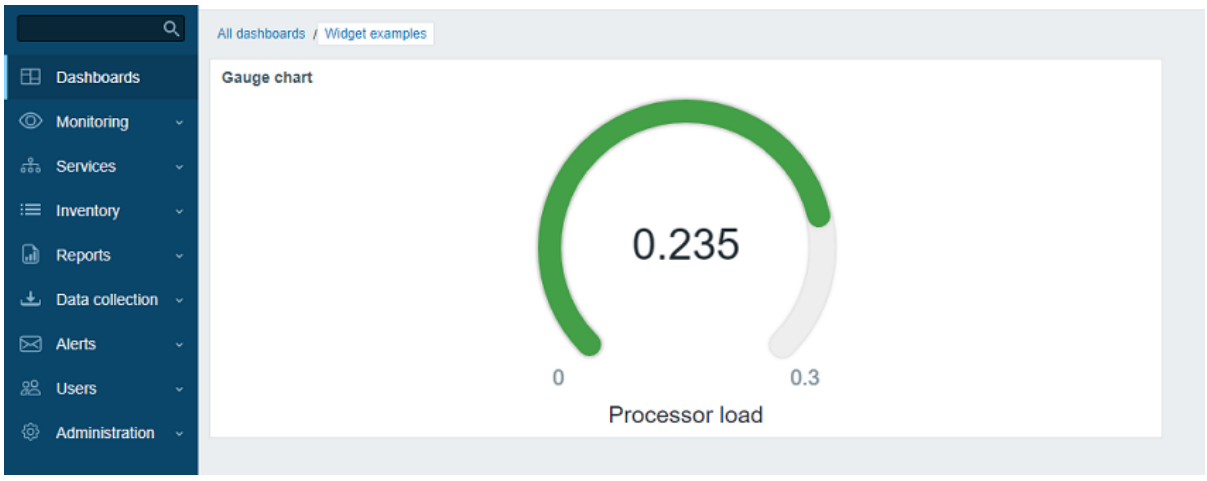
6. Atsvaidziniet Zabbix lietotāja saskarni. Noklikšķiniet *My address* > *External IP*, lai redzētu sava datora IP adresi.



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 pamata "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.

Pievienot tukšu logrīku Zabbix lietotāja saskarnei

1. Izveidojiet direktoriju `lesson_gauge_chart` direktorijā `modules` savā Zabbix lietotāja saskarnes instalācijā (piemēram, *zabbix/ui/modules*).

> **Note:**
> Visi pielāgotie logrīki tiek uzskatīti par ārējiem moduļiem, un tie ir jāpievieno Zabbix lietotāja saskarnes instalācijas direktorijam `modules` (piemēram, *zabbix/ui/modules*). Direktorijs `zabbix/ui/widgets` ir rezervēts Zabbix iebūvētajiem logrīkiem un tiek atjaunināts kopā ar Zabbix UI.

2. Izveidojiet failu `manifest.json` ar pamata logrīka metadatiem (skatiet atbalstīto parametru aprakstu).

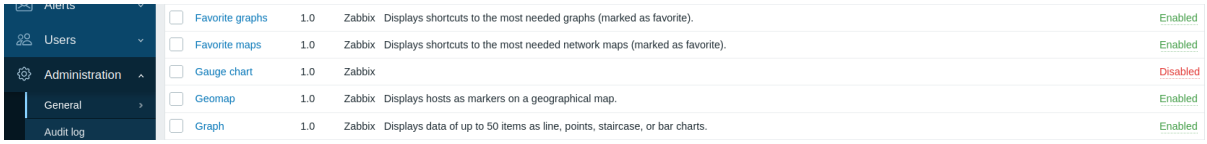
ui/modules/lesson_gauge_chart/manifest.json

```
{
    "manifest_version": 2.0,
    "id": "lesson_gauge_chart",
    "type": "widget",
    "name": "Gauge chart",
    "namespace": "LessonGaugeChart",
    "version": "1.1",
    "author": "Zabbix"
}
```

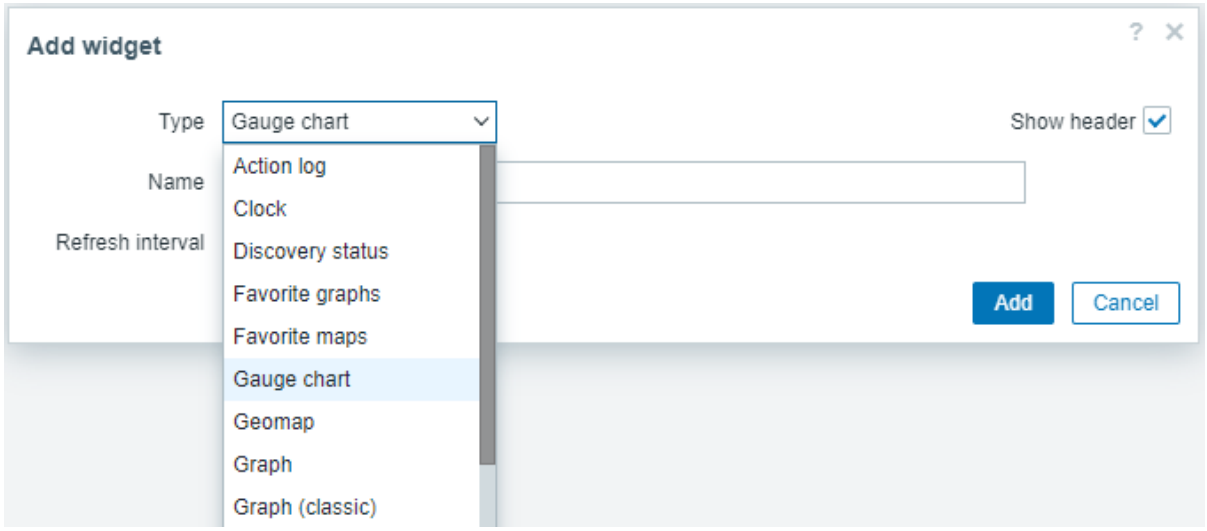
3. Zabbix lietotāja saskarnē dodieties uz sadaļu *Administration* > *General* > *Modules* un noklikšķiniet uz pogas *Scan directory*.



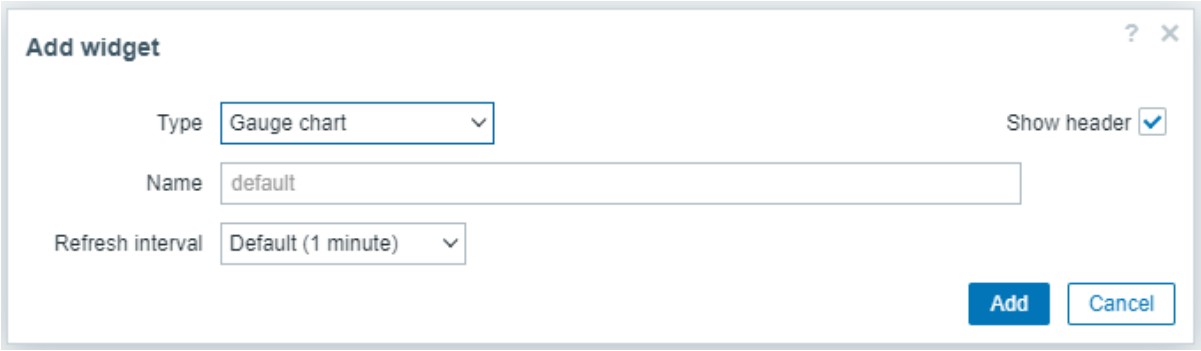
4. Sarakstā atrodiet jauno moduli *Gauge chart* un noklikšķiniet uz hipersaites "Disabled", lai mainītu moduļa statusu no "Disabled" uz "Enabled" (ja modulis nav uzskaitīts, skatiet sadaļu troubleshooting).



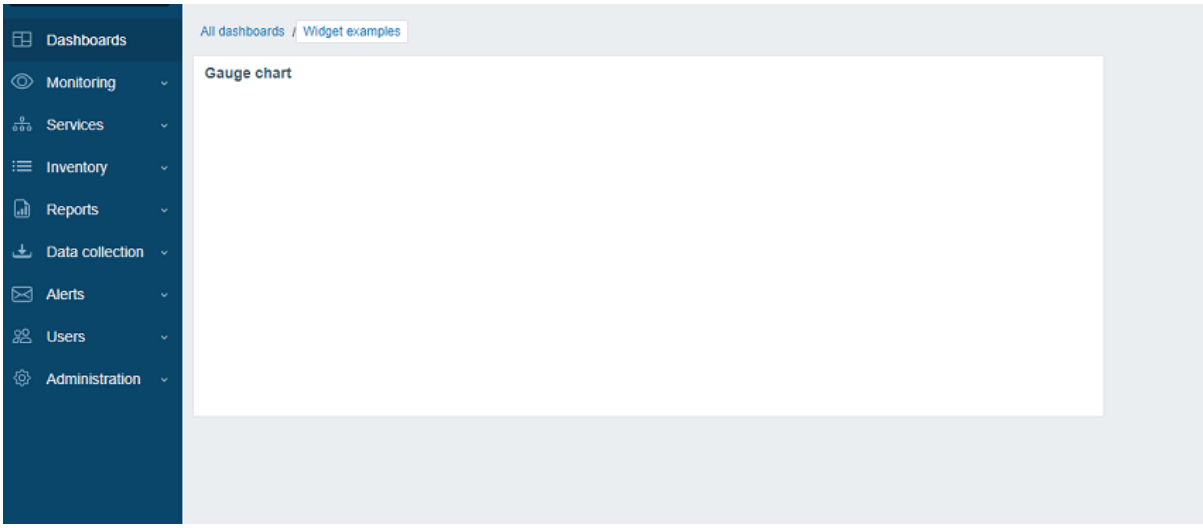
5. Atveriet paneli, pārslēdziet to rediģēšanas režīmā un pievienojiet jaunu logrīku. Laukā *Type* atlasiet "Gauge chart".



6. Šajā brīdī *Gauge chart* logrīka konfigurācijā ir tikai kopīgie logrīka lauki *Name* un *Refresh interval*. Noklikšķiniet uz *Add*, lai pievienotu logrīku panelim.



7. Panelī jāparādās tukšam logrīkam. Noklikšķiniet uz *Save changes* augšējā labajā stūrī, lai saglabātu paneli.



Pievienot widget skatu

Note:
Widgeta **skata** failam jāatrodas `views` direktorijā (šajā pamācībā: `ui/modules/lesson_gauge_chart/views/`). Ja failam ir noklusējuma nosaukums `widget.view.php`, tas nav jāreģistrē `manifest.json` failā. Ja failam ir cits nosaukums, norādiet to `manifest.json` faila sadaļā `actions/widget.lesson_gauge_chart.view`.

1. Izveidojiet direktoriju `views` direktorijā `lesson_gauge_chart`.
2. Izveidojiet failu `widget.view.php` direktorijā `views`.

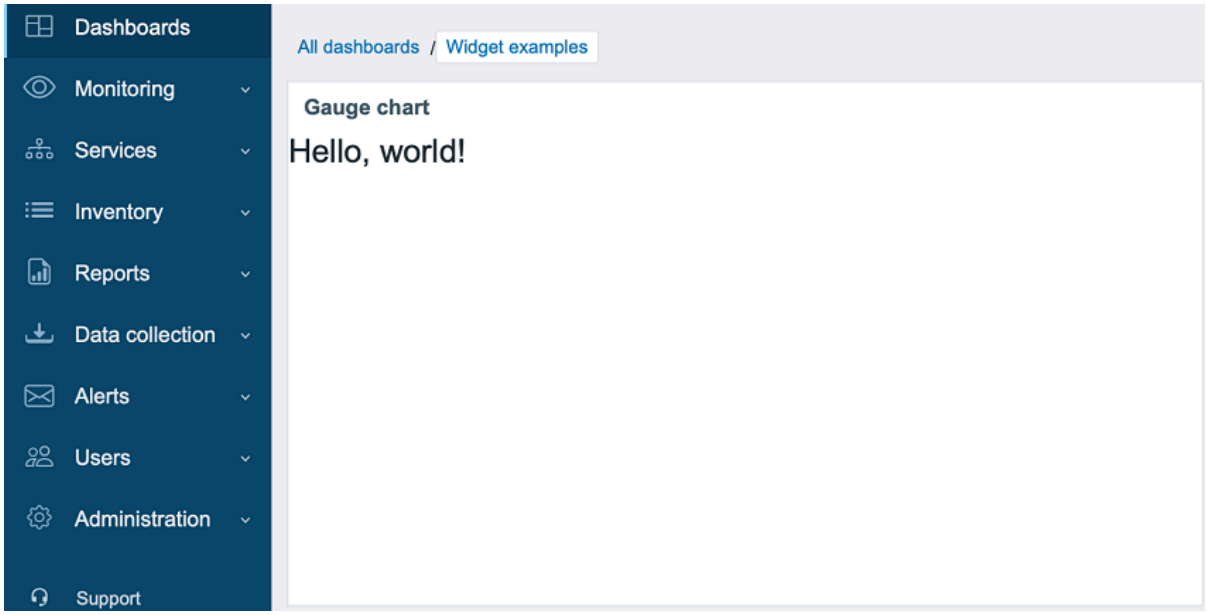
`ui/modules/lesson_gauge_chart/views/widget.view.php`

```
<?php

/**
 * Gauge chart widget view.
 *
 * @var CView $this
 * @var array $data
 */

(new CWidgetView($data))
    ->addItem(
        new CTag('h1', true, 'Hello, world!')
    )
    ->show();
```

3. Atsvaidziniet informācijas paneli. Tagad *Gauge chart* widgetā tiek rādīts "Hello, world!".



Part II - Gauge chart Pievienot iestatījumus konfigurācijas skatam un izmantot 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 ir jāizveido `WidgetForm` klase, kas paplašinās `Zabbix\Widgets\CWidgetFor`

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 pārvērš vērtību vienotā formātā, lai to saglabātu datubāzē.

> **Note:**
> Logrīka **formas** failam jāatrodas `includes` direktorijā (šajā pamācībā: `ui/modules/lesson_gauge_chart/includes/`). Ja failam ir noklusējuma nosaukums `WidgetForm.php`, tas nav jāreģistrē `manifest.json` failā. Ja failam ir cits nosaukums, norādiet to `manifest.json` faila 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*. Klases `CWidgetFieldTextBox` laukam skats ir `CWidgetFieldTextBoxView`.

`ui/modules/lesson_gauge_chart/views/widget.edit.php`

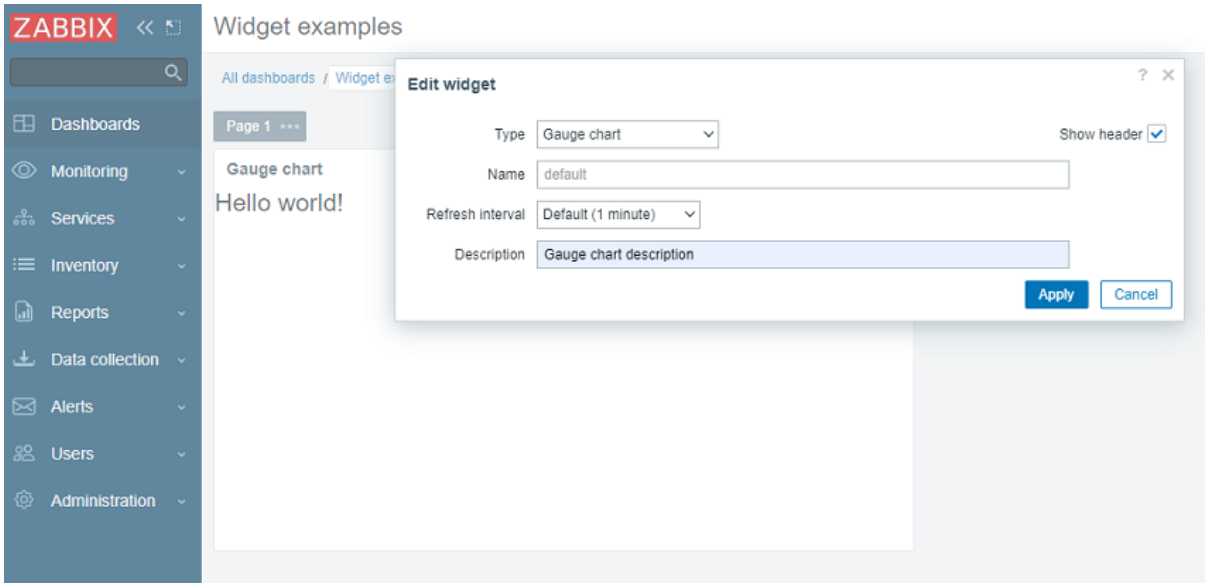
```
<?php

/**
 * 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. Noklikšķiniet uz *Apply* logrīka konfigurācijas formā. Pēc tam augšējā labajā stūrī noklikšķiniet uz *Save changes*, lai saglabātu paneli. Ņemiet vērā, ka jaunais apraksts nekur nav redzams, un logrīks joprojām rāda "Hello, world!".

Lai jaunais apraksts parādītos logrīkā, lauka *Description* vērtība ir jāiegūst no datubāzes un jānodod logrīka skatam. Šim nolūkam jums ir jāizveido darbības klase.

8. Izveidojiet jaunu direktoriju `actions` direktorijā `lesson_gauge_chart`.

9. Izveidojiet failu `WidgetView.php` direktorijā `actions`. Darbības klase `WidgetView` paplašinās klasi `CControllerDashboardWidget`

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']`.

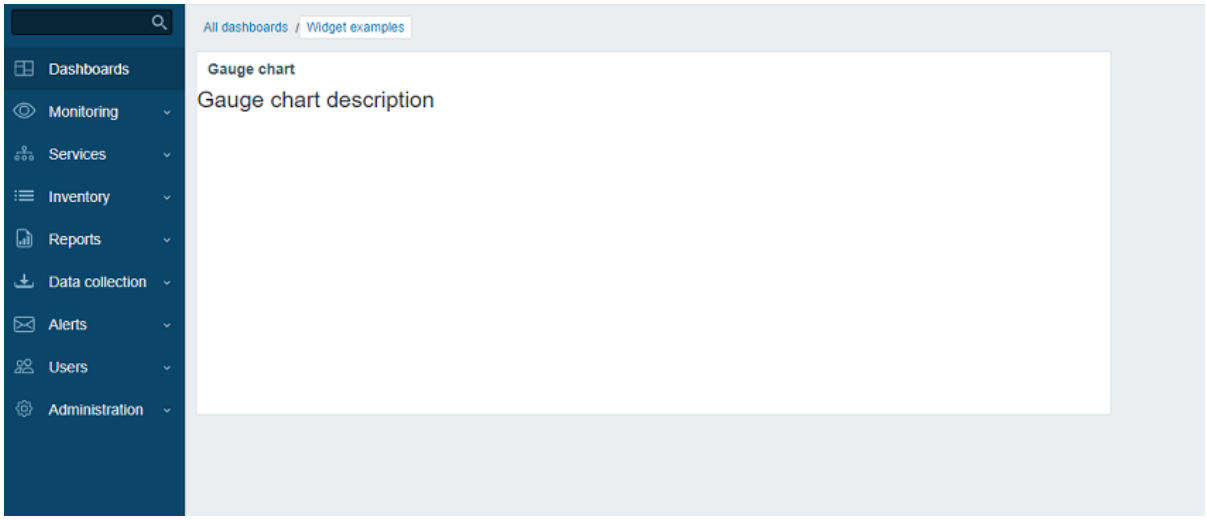
`ui/modules/lesson_gauge_chart/views/widget.view.php`

```
<?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 ir jāparāda lietotāja izvēlēta vienuma pēdējā vērtība. Lai to izdarītu, konfigurācijā ir jāpievieno iespēja atlasīt vienumus.

Šajā sadaļā jūs uzzināsiet, kā pievienot vienuma atlases lauku logrīka formai un kā pievienot šī lauka vizuālo daļu konfigurācijas skatā. Pēc tam logrīka kontrolleris varēs iegūt vienuma datus un tā vērtību, izmantojot API pieprasījumu. Kad vērtība būs saņemta, to varēs parādīt logrīka skatā.

1. Atveriet `includes/WidgetForm.php` un pievienojiet `CWidgetFieldMultiSelectItem` lauku. 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
};

/**
 * Gauge chart widget form.
 */
class WidgetForm extends CWidgetForm {

    public function addFields(): self {
        return $this
            ->addField(
                (new CWidgetFieldMultiSelectItem('itemid', _('Item')))
                    ->setFlags(CWidgetField::FLAG_NOT_EMPTY | CWidgetField::FLAG_LABEL_ASTERISK)
                    ->setMultiple(false)
            )
            ->addField(
                new CWidgetFieldTextBox('description', _('Description'))
            );
    }
}
```

2. Atveriet `views/widget.edit.php` un pievienojiet lauka vizuālo komponentu konfigurācijas skatam.

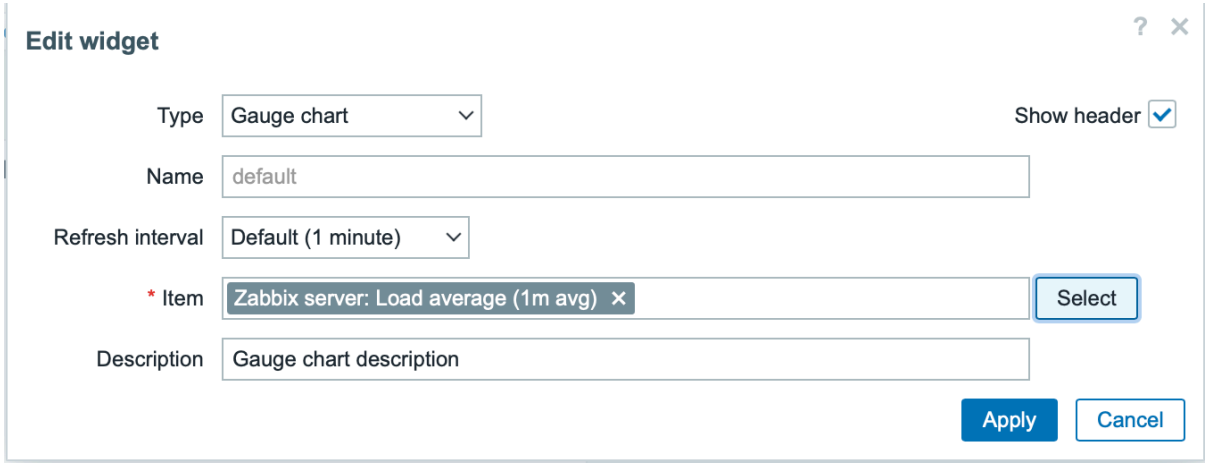
`ui/modules/lesson_gauge_chart/views/widget.edit.php`

```php
<?php

/**
 * Gauge chart widget form view.
 *
 * @var CView $this
 * @var array $data
 */

(new CWidgetFormView($data))
    ->addField(
        new CWidgetFieldMultiSelectItemView($data['fields']['itemid'])
    )
    ->addField(
        new CWidgetFieldTextBoxView($data['fields']['description'])
    )
    ->show();
```

3. Atgriezieties informācijas panelī un noklikšķiniet uz zobrata ikonas logrīkā, lai atvērtu logrīka konfigurācijas formu.
4. Tagad logrīka konfigurācijas formā ir jauns ievades lauks *Item*. Atlasiet hostu "Zabbix server" un vienumu "Load average (1m avg)".



5. Noklikšķiniet uz *Apply* logrīka konfigurācijas formā. Pēc tam augšējā labajā stūrī noklikšķiniet uz *Save changes*, lai saglabātu informācijas paneli.
6. Atveriet un modificējiet `actions/WidgetView.php`.

Turpmāk vienuma ID būs pieejams logrīka kontrolierī `$this->fields_values['itemid']`. Kontrollera `doAction()` metode iegūst vienuma datus (nosaukumu, vērtības tipu, vienības) ar API metodi `item.get` un vienuma pēdējo vērtību ar 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.

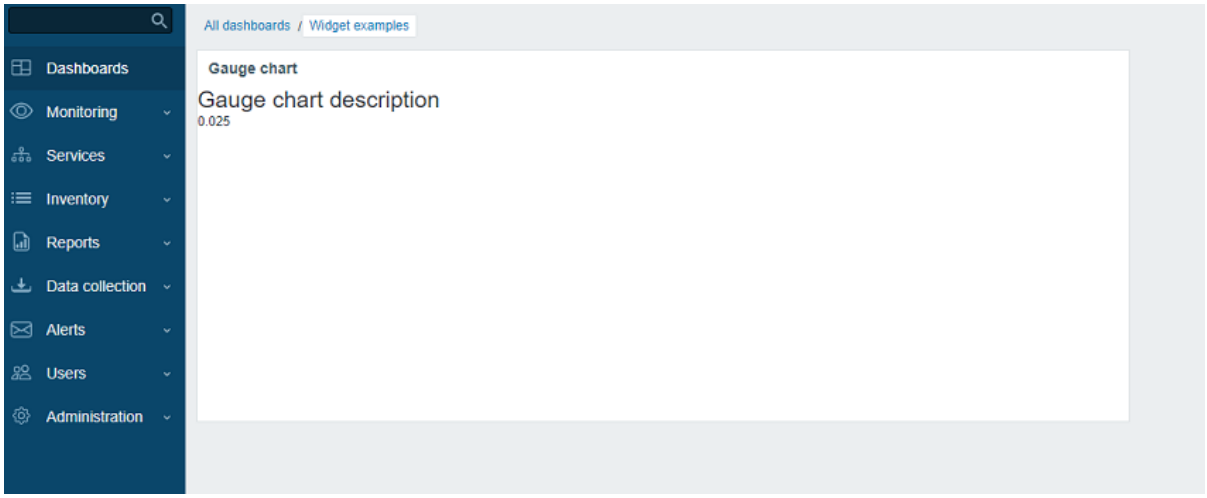
`ui/modules/lesson_gauge_chart/views/widget.view.php`

```
<?php

/**
 * Gauge chart widget view.
 *
 * @var CView $this
 * @var array $data
 */

(new CWidgetView($data))
    ->addItem([
        new CTag('h1', true, $data['description']),
        new CDiv($data['value'] !== null ? $data['value']['value'] : _('No data'))
    ])
    ->show();
```

8. Atsvaidziniet informācijas paneļa lapu. Logrīks parādīs jaunāko vienuma vērtību.



Pievienot papildu konfigurācijas iestatījumus konfigurācijas skatā

Šajā sadaļā jūs uzzināsiet, kā pievienot izvēršamu/sakļaujamu *Papildu konfigurācija* sadaļu ar izvēles parametriem, piemēram, krāsu, minimālajām un maksimālajām vērtībām, vienībām un iepriekš izveidoto *Apraksts* lauku.

1. Izveidojiet `Widget.php` failu galvenajā widget direktorijā `lesson_gauge_chart`, lai izveidotu jaunu klasi `Widget`.

Klase `Widget` paplašinās bāzes klasi `CWidget`, lai pievienotu/pārrakstītu noklusējuma widget iestatījumus (šajā gadījumā - tulkojumus). Tālāk norādītais JavaScript parāda virkni "No data", ja dati nav pieejami. Virkne "No data" ir pieejama Zabbix lietotāja saskarnes tulkojumu failos.

Ja ir kādas widget konstantes, ieteicams tās arī norādīt `Widget` klasē.

`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 izvēlne), *Min* (skaitliskais lauks), *Max* (skaitliskais lauks) un *Units* (izvēlne), kā arī definējiet noklusējuma krāsu paleti krāsu izvēlnei, 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
};

/**
 * Gauge chart widget form.
 */
class WidgetForm extends CWidgetForm {

    public const DEFAULT_COLOR_PALETTE = [
        'FF465C', 'B0AF07', '0EC9AC', '524BBC', 'ED1248', 'D1E754', '2AB5FF', '385CC7', 'EC1594', 'BAE37D',
        '6AC8FF', 'EE2B29', '3CA20D', '6F4BBC', '00A1FF', 'F3601B', '1CAE59', '45CFDB', '894BBC', '6D6D6D'
    ];

    public function addFields(): self {
        return $this
            ->addField(
                (new CWidgetFieldMultiSelectItem('itemid', _('Item')))
                    ->setFlags(CWidgetField::FLAG_NOT_EMPTY | CWidgetField::FLAG_LABEL_ASTERISK)
                    ->setMultiple(false)
            )
            ->addField(
                (new CWidgetFieldColor('chart_color', _('Color')))->setDefault('FF0000')
            )
            ->addField(
                (new CWidgetFieldNumericBox('value_min', _('Min')))
                    ->setDefault(0)
                    ->setFlags(CWidgetField::FLAG_NOT_EMPTY | CWidgetField::FLAG_LABEL_ASTERISK)
            )
            ->addField(
                (new CWidgetFieldNumericBox('value_max', _('Max')))
                    ->setDefault(100)
                    ->setFlags(CWidgetField::FLAG_NOT_EMPTY | CWidgetField::FLAG_LABEL_ASTERISK)
            )
            ->addField(
                (new CWidgetFieldSelect('value_units', _('Units'), [
                    Widget::UNIT_AUTO => _x('Auto', 'history source selection method'),
                    Widget::UNIT_STATIC => _x('Static', 'history source selection method')
                ]))->setDefault(Widget::UNIT_AUTO)
            )
            ->addField(
                (new CWidgetFieldTextBox('value_static_units'))
            )
            ->addField(
                new CWidgetFieldTextBox('description', _('Description'))
            );
    }
}
```

3. Atveriet `views/widget.edit.php` un pievienojiet lauku vizuālos komponentus konfigurācijas skatam.

`ui/modules/lesson_gauge_chart/views/widget.edit.php`

```
<?php

/**
 * Gauge chart widget form view.
 *
 * @var CView $this
```

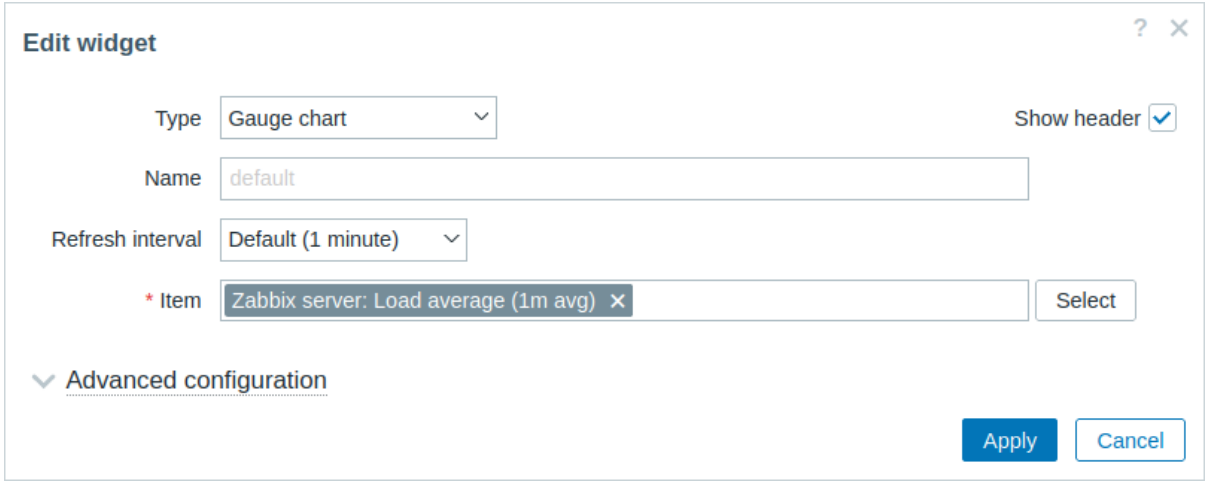
```
 * @var array $data
 */

$lefty_units = new CWidgetFieldSelectView($data['fields']['value_units']);
$lefty_static_units = (new CWidgetFieldTextBoxView($data['fields']['value_static_units']))
    ->setPlaceholder(_('value'))
    ->setWidth(ZBX_TEXTAREA_TINY_WIDTH);

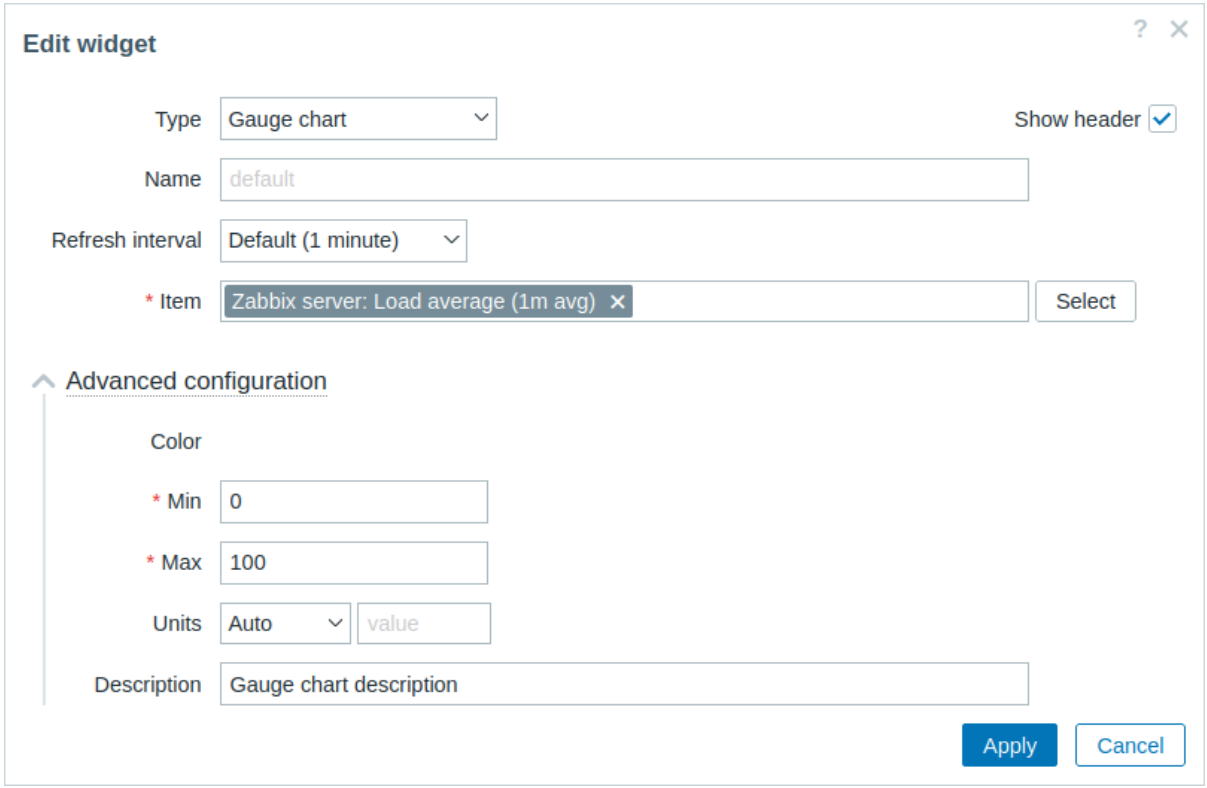
(new CWidgetFormView($data))
    ->addField(
        (new CWidgetFieldMultiSelectItemView($data['fields']['itemid']))
            ->setPopupParameter('numeric', true)
    )
    ->addFieldset(
        (new CWidgetFormFieldsetCollapsibleView(_('Advanced configuration')))
            ->addField(
                new CWidgetFieldColorView($data['fields']['chart_color'])
            )
            ->addField(
                new CWidgetFieldNumericBoxView($data['fields']['value_min'])
            )
            ->addField(
                new CWidgetFieldNumericBoxView($data['fields']['value_max'])
            )
            ->addItem([
                $lefty_units->getLabel(),
                (new CFormField([
                    $lefty_units->getView()->addClass(ZBX_STYLE_FORM_INPUT_MARGIN),
                    $lefty_static_units->getView()
                ]))
            ])
            ->addField(
                new CWidgetFieldTextBoxView($data['fields']['description'])
            )
    )
    ->show();
```

Note:
`CWidgetFormView` klases `addField()` metode kā otro parametru pieņem CSS klases virkni.

4. Atgriezieties informācijas panelī, pārslēdzieties rediģēšanas režīmā un noklikšķiniet uz zobrata ikonas widget, lai atvērtu widget konfigurācijas formu. Tagad widget konfigurācijas forma satur jaunu izvēršamu/sakļaujamu sadaļu *Papildu konfigurācija*.



5. Izvēršiet sadaļu *Papildu konfigurācija*, lai redzētu papildu widget konfigurācijas laukus. Ņemiet vērā, ka laukam *Color* vēl nav krāsu izvēlnes. Tas ir tāpēc, ka krāsu izvēlne ir jāinicializē ar JavaScript, kas tiks pievienots nākamajā sadaļā - *Pievienot JavaScript widget*.



Pievienojiet JavaScript sīkrīkam

Šajā sadaļā jūs uzzināsiet, kā pievienot mēraparāta diagrammu, kas izveidota, izmantojot JavaScript, un kas parāda, vai jaunākā vērtība ir normāla vai pārāk augsta/pārāk zema.

1. Izveidojiet failu `widget.edit.js.php` direktorijā `views`.

JavaScript būs atbildīgs par krāsu izvēlnes 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

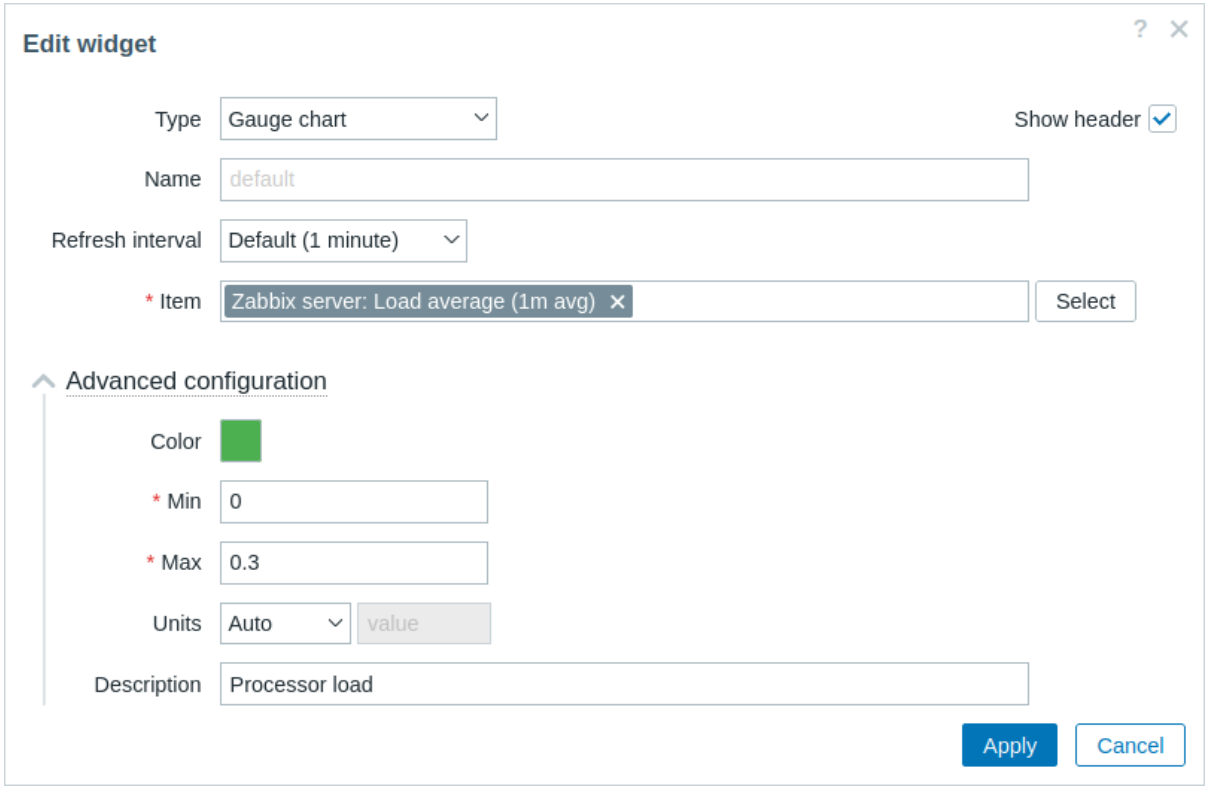
/**
 * Gauge chart widget form view.
 *
 * @var CView $this
 * @var array $data
 */

use Modules\LessonGaugeChart\Includes\WidgetForm;

$lefty_units = new CWidgetFieldSelectView($data['fields']['value_units']);
$lefty_static_units = (new CWidgetFieldTextBoxView($data['fields']['value_static_units']))
    ->setPlaceholder(_('value'))
    ->setWidth(ZBX_TEXTAREA_TINY_WIDTH);

(new CWidgetFormView($data))
    ->addField(
        (new CWidgetFieldMultiSelectItemView($data['fields']['itemid']))
            ->setPopupParameter('numeric', true)
    )
    ->addFieldset(
        (new CWidgetFormFieldsetCollapsibleView(_('Advanced configuration')))
            ->addField(
                new CWidgetFieldColorView($data['fields']['chart_color'])
            )
            ->addField(
                new CWidgetFieldNumericBoxView($data['fields']['value_min'])
            )
            ->addField(
                new CWidgetFieldNumericBoxView($data['fields']['value_max'])
            )
            ->addItem([
                $lefty_units->getLabel(),
                (new CFormField([
                    $lefty_units->getView()->addClass(ZBX_STYLE_FORM_INPUT_MARGIN),
                    $lefty_static_units->getView()
                ]))
            ])
            ->addField(
                new CWidgetFieldTextBoxView($data['fields']['description'])
            )
    )
    ->includeJsFile('widget.edit.js.php')
    ->addJavaScript('widget_lesson_gauge_chart_form.init('.json_encode([
        'color_palette' => WidgetForm::DEFAULT_COLOR_PALETTE
    ], JSON_THROW_ON_ERROR).');')
    ->show();
```

3. Atgriezieties informācijas panelī, noklikšķiniet uz zobrata ikonas sīkrīkā, lai atvērtu sīkrīka konfigurācijas formu. Pēc tam izvērsiet sadaļu *Advanced configuration*, lai redzētu inicializēto krāsu izvēlni. Aizpildiet laukus ar vērtībām un izvēlieties mēraparāta diagrammas krāsu.



4. Sīkrīka konfigurācijas formā noklikšķiniet uz *Apply*. Pēc tam augšējā labajā stūrī noklikšķiniet uz *Save changes*, lai saglabātu informācijas paneli.
5. Atveriet `actions/WidgetView.php` un atjauniniet kontrolieri.

Rekvizīts `$this->fields_values` tagad satur visu *Advanced configuration* lauku vērtības. Pabeidziet kontrolieri, lai konfigurāciju un atlasītā vienuma vērtību varētu nodot sīkrīka 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 ir jāizveido konteiners mēraparāta diagrammai, kuru zīmēsiet nākamajās darbībās, 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 widget view.
 *
 * @var CView $this
 * @var array $data
 */

(new CWidgetView($data))
    ->addItem([
        (new CDiv())->addClass('chart'),
        $data['fields_values']['description']
            ? (new CDiv($data['fields_values']['description']))->addClass('description')
            : null
    ])
    ->setVar('history', $data['history'])
    ->setVar('fields_values', $data['fields_values'])
    ->show();
```

7. Izveidojiet jaunu direktoriju `assets` direktorijā `lesson_gauge_chart`. Šī direktorija tiks izmantota JavaScript, CSS un, iespējams, citu resursu, piemēram, fontu vai attēlu, glabāšanai.
8. Sīkrīka skata JavaScript izveidojiet direktoriju `js` direktorijā `assets`.
9. Izveidojiet failu `class.widget.js` direktorijā `assets/js`.

Šī JavaScript sīkrīka klase paplašinās visu informācijas paneļa sīkrīku bāzes JavaScript klasi - `CWidget`.

Informācijas panelis paļaujas uz pareizu sīkrīka implementāciju un nodod sīkrīkam visu būtisko informāciju, izsaucot attiecīgās JavaScript metodes. Informācijas panelis arī sagaida, ka sīkrīks ģenerēs notikumus, kad notiek kāda mijiedarbība. Tādēļ klase `CWidget` satur metožu kopu ar sīkrīka uzvedības noklusējuma implementāciju, ko var pielāgot, paplašinot klasi.

Šajā gadījumā ir nepieciešama noteikta pielāgošana, tāpēc pielāgota loģika tiks ieviesta šādai sīkrīka uzvedībai:

- sīkrīka inicializācija, kas ir atbildīga par sīkrīka sākotnējā stāvokļa definēšanu (skatiet metodi `onInitialize()`);

- sīkrīka satura attēlošana (tas ir, mēraparāta diagrammas zīmēšana), ja sīkrīka atjaunināšanas process ir bijis veiksmīgs un bez kļūdām (skatiet metodi `processUpdateResponse(response)` un saistītās metodes `_resizeChart()` un `_updatedChart()`)
- sīkrīka izmēra maiņa (skatiet metodi `onResize()` un saistīto metodi `_resizeChart()`)

Pārējiem mēraparāta diagrammas sīkrīka aspektiem tiks izmantota sīkrīka uzvedības noklusējuma implementācija. Lai uzzinātu vairāk par klases `CWidget` JavaScript metodēm, skatiet: JavaScript.

Tā kā šis JavaScript ir nepieciešams sīkrīka skatam, tas ir jāielādē kopā ar informācijas paneļa lapu. Lai iespējotu JavaScript ielādi, jums būs jāatjaunina parametri `assets/js` un `js_class` failā `manifest.json`, kā parādīts 10. darbībā.

`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 `assets/js` sadaļā;
- klases nosaukumu (`WidgetLessonGaugeChart`) parametram `js_class` sadaļā `widget`.

Klase `WidgetLessonGaugeChart` tagad tiks automātiski ielādēta kopā ar informācijas 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 CSS stilus vienumam

Šajā sadaļā jūs uzzināsiet, kā pievienot pielāgotus CSS stilus, lai vienums izskatītos pievilcīgāks.

1. Vienuma stiliem izveidojiet jaunu direktoriju `css` direktorijā `assets`.
2. Izveidojiet failu `widget.css` direktorijā `assets/css`. Lai noformētu vienuma elementus, izmantojiet selektoru `div.dashboard-widget-{widget id}`. Lai konfigurētu CSS visam vienumam, 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;
}
```

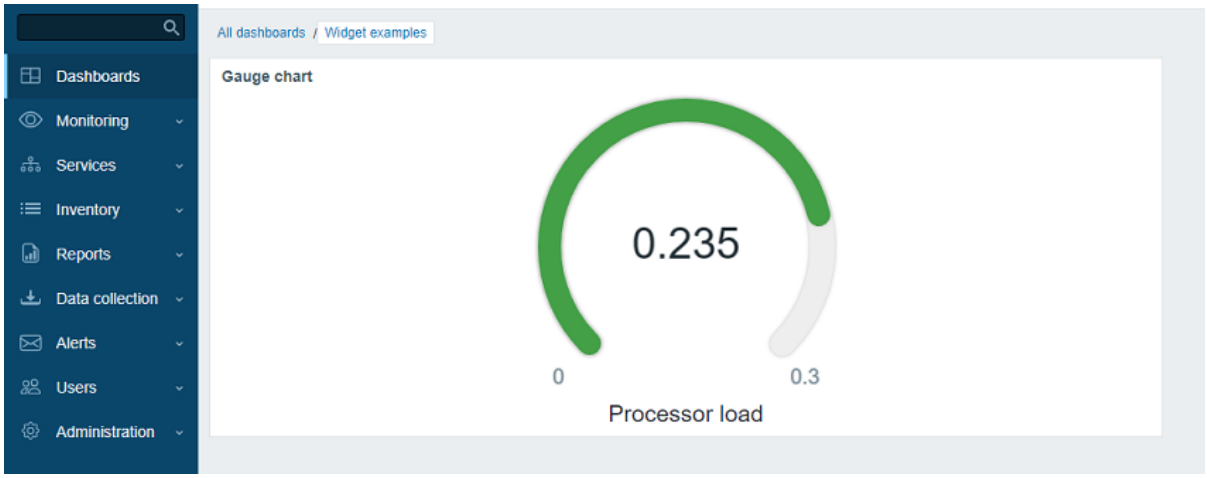
3. Atveriet `manifest.json` un pievienojiet CSS faila nosaukumu (`widget.css`) masīvam sadaļā `assets/css`. Tas ļaus CSS stiliem, kas definēti `widget.css`, ielādēties kopā ar informācijas paneļa lapu.

`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 informācijas paneļa lapu, lai redzētu vienuma pabeigto 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

```
ZABBIX_AGENT2(8)            Sistēmas administratora rokasgrāmata           ZABBIX_AGENT2(8)

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_agent2 ir lietojumprogramma dažādu pakalpojumu parametru
       uzraudzībai.

OPCIJAS
       -c, --config config-file
              Izmantot alternatīvo config-file noklusējuma faila vietā.

       -R, --runtime-control runtime-option
              Veikt administratīvās funkcijas atbilstoši runtime-option.

           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 metrikas
               version
                   Parādīt versiju

       -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 atgrieztās vērtības tips,
              atdalīts ar vertikālu svītru. Lietotāja parametriem tips
              vienmēr ir t, jo aģents nevar noteikt visas iespējamās
              atgrieztās vērtības. Vienumi, kas parādīti kā strādājoši, nav
              garantēti darbam no Zabbix servera 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 par vienumu, kas
               darbojas tikai aktīvajā režīmā, piemēram, žurnāla uzraudzības
               vienums, vai vienumu, kam nepieciešamas vairākas savāktās
               vērtības. Arī atļauju problēmas vai nepareizi lietotāja
               parametri 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
              Testēt vienu vienumu un iziet. Izvades aprakstu skatiet pie
              --print.

       -v, --verbose
              Iespējot detalizētu izvadi, testējot vienumu vai drukājot
              zināmos vienumus.

       -T, --test-config
              Validēt konfigurācijas failu un iziet.

       -h, --help
              Parādīt šo palīdzību un iziet.

       -V, --version
              Izvadīt versijas informāciju un iziet.

FAILI
       /usr/local/etc/zabbix_agent2.conf
              Zabbix aģenta 2 konfigurācijas faila noklusējuma atrašanās vieta
              (ja tā nav mainīta kompilēšanas laikā).

SKATĪT ARĪ
       Dokumentācija

       zabbix_agentd(8),  zabbix_get(1),  zabbix_proxy(8),   zabbix_sender(1),
       zabbix_server(8), zabbix_js(1), zabbix_web_service(8)

Zabbix                            2022-01-06                  ZABBIX_AGENT2(8)
```

zabbix_agentd

```
ZABBIX_AGENTD(8)            Sistēmas pārvaldnieka rokasgrāmata           ZABBIX_AGENTD(8)

NOSAUKUMS
       zabbix_agentd - Zabbix aģenta dēmons

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.

OPCIJAS
       -c, --config config-file
              Izmantot alternatīvu config-file noklusētā faila vietā.

       -f, --foreground
              Palaist Zabbix aģentu priekšplānā.

       -R, --runtime-control runtime-option
              Veikt administratīvās funkcijas atbilstoši 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

           Žurnālošanas 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 target kā "process-type,N"

       -p, --print
              Izdrukāt zināmos vienumus un iziet. Katram vienumam tiek
              izmantotas vai nu vispārīgas noklusētās vērtības, vai arī
              testēšanai paredzētas specifiskas noklusētās vērtības.
              Šīs noklusētās vērtības ir norādītas kvadrātiekavās kā vienuma
              atslēgas parametri. Atgrieztās vērtības ir ietvertas
              kvadrātiekavās un pirms tām ir 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,
              negarantēti darbosies no Zabbix servera vai zabbix_get, veicot
              pieprasījumu darbojošam aģenta dēmonam, jo atļaujas vai vide
              var atšķirties. Atgriezto vērtību tipi ir:

           d   Skaitlis ar decimāldaļu.

           m   Nav atbalstīts. To var izraisīt tāda vienuma vaicāšana, kas
               darbojas tikai aktīvajā režīmā, piemēram, žurnāla uzraudzības
               vienums, vai vienums, 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   Bezzīmes vesels skaitlis.

       -t, --test item-key
              Testēt 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 noklusētā atrašanās vieta
              (ja tā nav mainīta kompilēšanas laikā).

SKATĪT ARĪ
       Dokumentācija

       zabbix_get(1), zabbix_proxy(8), zabbix_sender(1), zabbix_server(8),
       zabbix_js(1), zabbix_agent2(8), zabbix_web_service(8)

Zabbix                           2022-01-06                  ZABBIX_AGENTD(8)
```

zabbix_get

```
ZABBIX_GET(1)               Vispārīgo komandu rokasgrāmata              ZABBIX_GET(1)

NOSAUKUMS
       zabbix_get - Zabbix datu iegūšanas utilīta

KOPSAVILKUMS
       zabbix_get -s host-name-or-IP [-p port-number] [-I IP-address] [-t
       timeout] -k item-key
       zabbix_get -s host-name-or-IP [-p port-number] [-I IP-address] [-t
       timeout] --tls-connect cert --tls-ca-file CA-file [--tls-crl-file
       CRL-file] [--tls-agent-cert-issuer cert-issuer] [--tls-agent-cert-sub-
       ject cert-subject] --tls-cert-file cert-file --tls-key-file key-file
       [--tls-cipher13 cipher-string] [--tls-cipher cipher-string] -k item-key
       zabbix_get -s host-name-or-IP [-p port-number] [-I IP-address] [-t
       timeout] --tls-connect psk --tls-psk-identity PSK-identity
       --tls-psk-file PSK-file [--tls-cipher13 cipher-string] [--tls-cipher
       cipher-string] -k item-key
       zabbix_get -h
       zabbix_get -V

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 porta numuru aģentam, kas darbojas uz hosta. 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, kurai iegūt vērtību.

-P, --protocol value
Protokols, ko izmanto saziņai ar aģentu. Vērtības:

auto	pieslēgties, izmantojot JSON protokolu, un kļūmes gadījumā pārslēgties un mēģināt vēlreiz ar vienkārša teksta protokolu (noklusējums)
json	pieslēgties, izmantojot JSON protokolu
plaintext	pieslēgties, izmantojot vienkārša teksta protokolu, kur tiek nosūtīta tikai vienuma atslēga (6.4.x un vecākās laidienu versijas)

--tls-connect value
Kā pieslēgties aģentam. Vērtības:

unencrypted	pieslēgties bez šifrēšanas (noklusējums)
psk	pieslēgties, izmantojot TLS un iepriekš koplietotu atslēgu
cert	pieslēgties, 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.  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 atla-
              ses kritērijus.

       -h, --help
              Parādīt šo palīdzību un iziet.

       -V, --version
              Izvadīt versijas informāciju un iziet.
IZEJAS STATUSS
       Izejas statuss ir 0, ja zabbix_get veiksmīgi iegūst vērtību no
       Zabbix aģenta. Ja vērtības iegūšana neizdevās, izejas statuss ir 1.

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  op-
       erations,O=Example  Corp,DC=example,DC=com"  --tls-cert-file /home/zab-
       bix/zabbix_get.crt --tls-key-file /home/zabbix/zabbix_get.key
       zabbix_get  -s  127.0.0.1  -p  10050   -k   "system.cpu.load[all,avg1]"
       --tls-connect   psk   --tls-psk-identity   "PSK   ID   Zabbix   agentd"
       --tls-psk-file /home/zabbix/zabbix_agentd.psk

SKATIET ARĪ
       Dokumentācija

       zabbix_agentd(8), zabbix_proxy(8), zabbix_sender(1),  zabbix_server(8),
       zabbix_js(1), zabbix_agent2(8), zabbix_web_service(8)

Zabbix                            2022-01-06                       ZABBIX_GET(1)
```

zabbix_js

```
ZABBIX_JS(1)                Vispārīgo komandu rokasgrāmata                ZABBIX_JS(1)

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 nosau-
              kums ir norādīts '-', skripts tiks nolasīts no stdin.
```

```
       -p, --param input-param
              Norādiet ievades parametru, kas tiek nodots skriptam. Parametrs
              tiek nodots burtiski kā virkne un ir pieejams skriptā, izmanto-
              jot mainīgo 'value'. Savstarpēji izslēdzošs ar -i.

       -i, --input input-file
              Norādiet failu, kas satur skriptam nododamo ievades parametru.
              Faila saturs tiek nodots burtiski kā virkne un ir pieejams skri-
              ptā, izmantojot mainīgo 'value'. Ja kā faila nosaukums ir norā-
              dīts '-', ievade tiks nolasīta no stdin. Savstarpēji izslēdzošs
              ar -p.

       -w, --webdriver url
              Norādiet webdriver URL.

       -l, --loglevel log-level
              Norādiet žurnālošanas līmeni (diapazons: 0-5).

       -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

       zabbix_agentd(8), zabbix_proxy(8), zabbix_sender(1),  zabbix_server(8),
       zabbix_get(1), zabbix_agent2(8), zabbix_web_service(8)

Zabbix                           2022-01-06                       ZABBIX_JS(1)
```

zabbix_proxy

```
ZABBIX_PROXY(8)             Sistēmas administratora rokasgrāmata            ZABBIX_PROXY(8)

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 Zabbix serverim.

OPCIJAS
       -c, --config config-file
              Izmanto alternatīvu config-file, nevis noklusējuma failu.
```

```
-f, --foreground
       Palaist Zabbix starpniekserveri priekšplānā.

-R, --runtime-control runtime-option
       Veikt administratīvās funkcijas atbilstoši runtime-option.

    Runtime control options
        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.
            Tiks izmantots noklusējuma konfigurācijas fails (ja vien nav
            norādīta -c opcija), lai atrastu PID failu, un signāls tiks nosūtīts
            procesam, kas norādīts PID failā.

        snmp_cache_reload
            Pārlādēt SNMP kešatmiņu.

        housekeeper_execute
            Izpildīt housekeeper. Tiek ignorēts, ja housekeeper pašlaik
            tiek izpildīts.

        diaginfo[=section]
            Reģistrēt norādītās sadaļas iekšējo diagnostikas informāciju.
            Sadaļa var būt historycache, preprocessing, locks.
            Pēc noklusējuma tiek reģistrēta visu sadaļu diagnostikas informācija.

        prof_enable[=target]
            Iespējot profilēšanu, ietekmē visus procesus, ja target nav norādīts

        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.

    Log level control targets
        process-type
            Visi norādītā tipa procesi (availability manager, browser poller,
            configuration syncer, data sender, discovery manager, history syncer,
            housekeeper, http poller, icmp pinger, internal poller, ipmi manager,
            ipmi poller, java poller, odbc poller, poller, agent poller,
            http agent poller, snmp poller, preprocessing manager, preprocessing worker,
            self-monitoring, snmp trapper, supervisor, task manager, trapper,
            unreachable poller, vmware collector)

        process-type,N
            Procesa tips un numurs (piem., poller,3)

        pid Procesa identifikators

    Profiling control targets
        process-type
            Visi norādītā tipa procesi (availability manager, browser poller,
            configuration syncer, data sender, discovery manager, history syncer,
            housekeeper, http poller, icmp pinger, internal poller, ipmi manager,
            ipmi poller, java poller, odbc poller, poller, agent poller, http agent poller,
            snmp poller, preprocessing manager, self-monitoring, snmp trapper,
            task manager, trapper, unreachable poller, vmware collector)
```

```
               process-type,N
                   Procesa tips un numurs (piem., history syncer,1)

               pid Procesa identifikators

               scope
                   (rwlock, mutex, processing) var izmantot kopā ar process-type
                   (piem., 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 vien tā nav mainīta kompilēšanas laikā).

SKATĪT ARĪ
       Dokumentācija

       zabbix_agentd(8),  zabbix_get(1),  zabbix_sender(1),  zabbix_server(8),
       zabbix_js(1), zabbix_agent2(8), zabbix_web_service(8)

Zabbix                            2022-01-06                   ZABBIX_PROXY(8)
```

zabbix_sender

```
ZABBIX_SENDER(1)            Vispārīgo komandu rokasgrāmata           ZABBIX_SENDER(1)

NOSAUKUMS
       zabbix_sender - Zabbix sūtītāja utilīta

KOPSAVILKUMS
       zabbix_sender [-v] -z server [-p port] [-I IP-address] [-t timeout] -s
       host -k key -o value
       zabbix_sender [-v] -z server [-p port] [-I IP-address] [-t timeout] [-s
       host] [-T] [-N] [-r] -i input-file
       zabbix_sender [-v] -c config-file [-z server] [-p port] [-I IP-address]
       [-t timeout] [-s host] -k key -o value
       zabbix_sender [-v] -c config-file [-z server] [-p port] [-I IP-address]
       [-t timeout] [-s host] [-T] [-N] [-r] -i input-file
       zabbix_sender [-v] -z server [-p port] [-I IP-address] [-t timeout] -s
       host --tls-connect cert --tls-ca-file CA-file [--tls-crl-file CRL-file]
       [--tls-server-cert-issuer cert-issuer] [--tls-server-cert-subject
       cert-subject] --tls-cert-file cert-file --tls-key-file key-file
       [--tls-cipher13 cipher-string] [--tls-cipher cipher-string] -k key -o
       value
       zabbix_sender [-v] -z server [-p port] [-I IP-address] [-t timeout] [-s
       host] --tls-connect cert --tls-ca-file CA-file [--tls-crl-file
       CRL-file] [--tls-server-cert-issuer cert-issuer]
       [--tls-server-cert-subject cert-subject] --tls-cert-file cert-file
       --tls-key-file key-file [--tls-cipher13 cipher-string] [--tls-cipher
       cipher-string] [-T] [-N] [-r] -i input-file
       zabbix_sender [-v] -c config-file [-z server] [-p port] [-I IP-address]
       [-t timeout] [-s host] --tls-connect cert --tls-ca-file CA-file
```

```
[--tls-crl-file CRL-file] [--tls-server-cert-issuer cert-issuer]
[--tls-server-cert-subject cert-subject] --tls-cert-file cert-file
--tls-key-file key-file [--tls-cipher13 cipher-string] [--tls-cipher
cipher-string] -k key -o value
zabbix_sender [-v] -c config-file [-z server] [-p port] [-I IP-address]
[-t timeout] [-s host] --tls-connect cert --tls-ca-file CA-file
[--tls-crl-file CRL-file] [--tls-server-cert-issuer cert-issuer]
[--tls-server-cert-subject cert-subject] --tls-cert-file cert-file
--tls-key-file key-file [--tls-cipher13 cipher-string] [--tls-cipher
cipher-string] [-T] [-N] [-r] -i input-file
zabbix_sender [-v] -z server [-p port] [-I IP-address] [-t timeout] -s
host --tls-connect psk --tls-psk-identity PSK-identity --tls-psk-file
PSK-file [--tls-cipher13 cipher-string] [--tls-cipher cipher-string] -k
key -o value
zabbix_sender [-v] -z server [-p port] [-I IP-address] [-t timeout] [-s
host] --tls-connect psk --tls-psk-identity PSK-identity --tls-psk-file
PSK-file [--tls-cipher13 cipher-string] [--tls-cipher cipher-string]
[-T] [-N] [-r] -i input-file
zabbix_sender [-v] -c config-file [-z server] [-p port] [-I IP-address]
[-t timeout] [-s host] --tls-connect psk --tls-psk-identity PSK-iden-
tity --tls-psk-file PSK-file [--tls-cipher13 cipher-string] [--tls-ci-
pher cipher-string] -k key -o value
zabbix_sender [-v] -c config-file [-z server] [-p port] [-I IP-address]
[-t timeout] [-s host] --tls-connect psk --tls-psk-identity PSK-iden-
tity --tls-psk-file PSK-file [--tls-cipher13 cipher-string] [--tls-ci-
pher cipher-string] [-T] [-N] [-r] -i input-file
zabbix_sender -h
zabbix_sender -V
```

APRAKSTS

zabbix_sender ir komandrindas utilīta uzraudzības datu nosūtīšanai uz Zabbix serveri vai starpniekserveri. Zabbix serverī jāizveido vienums ar tipu Zabbix trapper un atbilstošo atslēgu. Ņemiet vērā, ka ienākošās vērtības tiks pieņemtas tikai no hostiem, kas šī vienuma laukā Allowed hosts ir norādīti kā atļautie.

OPCIJAS

-c, --config config-file

Izmanto config-file. Zabbix sender nolasa servera informāciju no agentd konfigurācijas faila. Pēc noklusējuma Zabbix sender nelasa nevienu konfigurācijas failu. Tiek atbalstīti tikai parametri Hostname, ServerActive, SourceIP, TLSConnect, TLSCAFile, TLSCRLFile, TLSServerCertIssuer, TLSServerCertSubject, TLSCertFile, TLSKeyFile, TLSPSKIdentity un TLSPSKFile. Hostname, kas definēts, izmantojot parametru HostnameItem, netiks izmantots; šādā gadījumā hosts jānorāda komandrindā (skatiet opciju -s). Nosūtīšanai tiek izmantotas visas adreses, kas definētas aģenta konfigurācijas parametrā ServerActive. Ja pakešu datu nosūtīšana uz kādu adresi neizdodas, nākamās paketes uz šo adresi netiek nosūtītas.

-z, --zabbix-server server

Zabbix servera resursdatora nosaukums vai IP adrese. Ja hosts tiek uzraudzīts, izmantojot starpniekserveri, tā vietā jāizmanto starpniekservera resursdatora nosaukums vai IP adrese. Izmantojot kopā ar --config, šī opcija aizstāj parametrā ServerActive aģenta konfigurācijas failā norādītos ierakstus.

-p, --port port

Norāda Zabbix servera trapper porta numuru. Noklusējuma vērtība ir 10051. Izmantojot kopā ar --config, šī opcija aizstāj aģenta konfigurācijas faila parametrā ServerActive norādītos porta ierakstus.

-I, --source-address IP-address
       Norāda avota IP adresi. Izmantojot kopā ar --config, šī opcija
       aizstāj aģenta konfigurācijas failā norādīto parametru SourceIP.

-t, --timeout seconds
       Norāda taimautu. Derīgais diapazons: 1-300 sekundes (noklusējums:
       60).

-s, --host host
       Norāda tā hosta nosaukumu, kuram pieder vienums (kā reģistrēts
       Zabbix lietotāja saskarnē). Hosta IP adrese un DNS nosaukums
       nedarbosies. Izmantojot kopā ar --config, šī opcija aizstāj aģenta
       konfigurācijas failā norādīto parametru Hostname.

-k, --key key
       Norāda vienuma atslēgu, uz kuru nosūtīt vērtību.

-o, --value value
       Norāda vienuma vērtību.

-i, --input-file input-file
       Ielādē vērtības no ievades faila. Norādiet - kā <input-file>, lai
       nolasītu vērtības no standarta ievades. Katrā faila rindā ir ar
       atstarpēm atdalīti elementi: <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, kas reģistrēts Zabbix lietotāja
       saskarnē, "key" ir mērķa vienuma atslēga, bet "value" ir nosūtāmā
       vērtība. Norādiet - kā <hostname>, lai izmantotu hosta nosaukumu
       no aģenta konfigurācijas faila vai argumenta --host.

       Ievades faila rindas piemērs:

        "Linux DB3" db.connections 43

       Vērtības tipam jābūt pareizi iestatītam Zabbix lietotāja
       saskarnes vienuma konfigurācijā. Zabbix sender vienā savienojumā
       nosūtīs līdz 250 vērtībām. Ievades faila saturam jābūt UTF-8
       kodējumā. Visas ievades faila vērtības tiek nosūtītas secīgi no
       augšas uz leju. Ieraksti jāformatē, ievērojot šādus noteikumus:

              • Tiek atbalstīti ieraksti pēdiņās un bez pēdiņām.

              • Pēdiņu rakstzīme ir dubultpēdiņa.

              • Ieraksti, kuros ir atstarpes, jāiekļauj pēdiņās.

              • Dubultpēdiņas un slīpsvītras rakstzīmes ierakstā, kas
                iekļauts pēdiņās, jāekranē ar slīpsvītru.

              • Ierakstos bez pēdiņām ekrānošana netiek atbalstīta.

              • Rindas pārneses ekrāna secības (\n) tiek atbalstītas
                virknēs, kas iekļautas pēdiņās.

              • Rindas pārneses ekrāna 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>. Laikspiedogs jānorāda Unix laikspiedoga formātā. Ja mērķa vienumam ir uz to attiecināti trigeri, visiem laikspiedogiem jābūt augošā secībā, pretējā gadījumā notikumu aprēķins nebūs pareizs.

Ievades faila rindas piemērs:

```
 "Linux DB3" db.connections 1429533600 43
```

Papildinformāciju skatiet opcijas --input-file aprakstā.

Ja hostam, kas atrodas uzturēšanas periodā ar tipu “no data”, tiek nosūtīta vērtība ar laikspiedogu, šī vērtība tiks atmesta; tomēr vērtību ar laikspiedogu var nosūtīt arī pēc uzturēšanas perioda beigām, 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 opcijas --input-file aprakstā.

-r, --real-time
Nosūta vērtības pa vienai, tiklīdz tās tiek saņemtas. To var izmantot, nolasot datus no standarta ievades.

--tls-connect value
Kā izveidot savienojumu ar serveri vai starpniekserveri. Vērtības:

unencrypted
izveidot savienojumu bez šifrēšanas (noklusējums)

psk
izveidot savienojumu, izmantojot TLS un iepriekš koplietotu atslēgu

cert
izveidot savienojumu, izmantojot TLS un sertifikātu

--tls-ca-file CA-file
Pilns ceļš uz failu, kurā ir augstākā līmeņa CA sertifikāts(-i) partnera sertifikāta pārbaudei.

--tls-crl-file CRL-file
Pilns ceļš uz failu, kurā ir atsauktie sertifikāti.

--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, kurā ir sertifikāts vai sertifikātu ķēde.

--tls-key-file key-file
Pilns ceļš uz failu, kurā ir privātā atslēga.

```
       --tls-psk-identity PSK-identity
              PSK-identity virkne.

       --tls-psk-file PSK-file
              Pilns ceļš uz failu, kurā ir iepriekš koplietotā atslēga.

       --tls-cipher13 cipher-string
              Šifru virkne OpenSSL 1.1.1 vai jaunākai versijai TLS 1.3
              protokolam. Ignorē noklusējuma šifrkopas atlases kritērijus. Šī
              opcija nav pieejama, ja OpenSSL versija ir vecāka 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). Ignorē noklusējuma
              šifrkopas atlases kritērijus.

       -v, --verbose
              Detalizētais režīms; -vv sniedz vēl vairāk informācijas.

       -h, --help
              Parāda šo palīdzību un iziet.

       -V, --version
              Izvada informāciju par versiju un iziet.

IZIETES 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
              Nosūta 342.45 kā mysql.queries vienuma vērtību uzraudzītajam
              hostam. Izmanto uzraudzīto hostu un Zabbix serveri, kas definēti
              aģenta konfigurācijas failā.

       zabbix_sender -c /etc/zabbix/zabbix_agentd.conf -s "Monitored Host"  -k
       mysql.queries -o 342.45
              Nosūta 342.45 kā Monitored Host hosta mysql.queries vienuma
              vērtību, izmantojot Zabbix serveri, kas definēts aģenta
              konfigurācijas failā.

       zabbix_sender -z 192.168.1.113 -i data_values.txt
              Nosūta vērtības no faila data_values.txt uz Zabbix serveri ar IP
              adresi 192.168.1.113. Hosta nosaukumi un atslēgas ir definēti
              failā.

       echo "- hw.serial.number  1287872261  SQ4321ASDF"  |  zabbix_sender  -c
       /usr/local/etc/zabbix_agentd.conf -T -i -
              Nosūta vērtību ar laikspiedogu no komandrindas uz Zabbix serveri,
              kas norādīts aģenta konfigurācijas failā. Defise ievades datos
              norāda, ka hosta nosaukums jāizmanto no tā paša konfigurācijas
              faila.

       echo '"Zabbix server" trapper.item ""' | zabbix_sender -z 192.168.1.113
       -p 10000 -i -
              No komandrindas nosūta tukšu vienuma vērtību uz Zabbix serveri ar
              IP adresi 192.168.1.113 un portu 10000. 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
              Nosūta 342.45 kā mysql.queries vienuma vērtību Monitored Host
              hostam uz serveri ar IP adresi 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
              Nosūta 342.45 kā mysql.queries vienuma vērtību Monitored Host
              hostam uz serveri ar IP adresi 192.168.1.113, izmantojot TLS ar
              iepriekš koplietotu atslēgu (PSK).

SKATIET ARĪ
       Dokumentācija

       zabbix_agentd(8),  zabbix_get(1),  zabbix_proxy(8),   zabbix_server(8),
       zabbix_js(1), zabbix_agent2(8), zabbix_web_service(8)

Zabbix                            2022-01-06                  ZABBIX_SENDER(1)
```

zabbix_server

```
ZABBIX_SERVER(8)            Sistēmas pārvaldnieka rokasgrāmata           ZABBIX_SERVER(8)

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 kodola dēmons.

OPCIJAS
       -c, --config config-file
              Izmantot alternatīvu config-file, nevis noklusējuma failu.

       -f, --foreground
              Palaist Zabbix serveri priekšplānā.

       -R, --runtime-control runtime-option
              Veikt administratīvās funkcijas atbilstoši runtime-option.

           Runtime control options
               config_cache_reload
                   Pārlādēt konfigurācijas kešatmiņu. Tiek ignorēts, ja kešatmiņa
                   pašlaik tiek ielādēta. Lai atrastu PID failu, tiks izmantots
                   noklusējuma konfigurācijas fails (ja vien nav norādīta -c
                   opcija), un signāls tiks nosūtīts procesam, kas norādīts PID
                   failā.

               snmp_cache_reload
                   Pārlādēt SNMP kešatmiņu.

               housekeeper_execute
```

```
        Izpildīt housekeeper. Tiek ignorēts, ja housekeeper pašlaik
        tiek izpildīts.

    trigger_housekeeper_execute
        Izpildīt trigger housekeeper (noņemt problēmas dzēstiem
        trigeriem). Tiek ignorēts, ja trigger housekeeper pašlaik tiek
        izpildīts.

    secrets_reload
        Pārlādēt noslēpumus no Vault. Tiek ignorēts, ja konfigurācijas
        kešatmiņa vai noslēpumi pašlaik tiek ielādēti.

    diaginfo[=section]
        Reģistrēt norādītās sadaļas iekšējo diagnostikas informāciju.
        Sadaļa var būt historycache, preprocessing, alerting, lld,
        valuecache, locks un connector. Pēc noklusējuma tiek reģistrēta
        visu sadaļu diagnostikas informācija.

    prof_enable[=target]
        Iespējot profilēšanu, ietekmē visus procesus, ja target nav
        norādīts

    prof_disable[=target]
        Atspējot profilēšanu, ietekmē visus procesus, ja target nav
        norādīts

    ha_status
        Parādīt augstas pieejamības klastera statusu. Var veikt tikai
        aktīvajā mezglā.

    ha_remove_node
        ha_remove_node[=name|ID] Noņemt augstas pieejamības mezglu pēc
        tā nosaukuma vai ID, kas norādīts ha_status. Var veikt tikai
        aktīvajā mezglā.

    ha_set_failover_delay[=delay]
        Iestatīt augstas pieejamības pārslēgšanās aizkavi. Aizkavi var
        norādīt, izmantojot laika sufiksus. Var veikt tikai aktīvajā
        mezglā.

    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

Log level control targets
    process-type
        Visi norādītā tipa procesi (alerter, alert manager, alert
        syncer, availability manager, browser poller, configuration
        syncer, configuration syncer worker, connector manager,
        connector worker, discovery manager, escalator, ha manager,
        history poller, history syncer, housekeeper, http poller, http
        agent poller, icmp pinger, internal poller, ipmi manager, ipmi
        poller, java poller, lld manager, lld worker, odbc poller,
        poller, agent poller, preprocessing manager, preprocessing
        worker, proxy poller, proxy group manager, report manager,
        report writer, self-monitoring, service manager, snmp poller,
        snmp trapper, supervisor, task manager, timer, trapper,
        trigger housekeeper, unreachable poller, vmware collector)
```

```
                process-type,N
                    Procesa tips un numurs (piemēram, poller,3)

                pid Procesa identifikators

            Profiling control targets
                process-type
                    Visi norādītā tipa procesi (alerter, alert manager, alert
                    syncer, availability manager, browser poller, configuration
                    syncer, configuration syncer worker, connector manager,
                    connector worker, discovery manager, escalator, ha manager,
                    history poller, history syncer, housekeeper, http poller, http
                    agent poller, icmp pinger, internal poller, ipmi manager, ipmi
                    poller, java poller, lld manager, lld worker, odbc poller,
                    poller, agent poller, preprocessing manager, proxy poller,
                    proxy group manager, report manager, report writer,
                    self-monitoring, service manager, snmp poller, snmp trapper,
                    task manager, timer, trapper, trigger housekeeper, unreachable
                    poller, vmware collector)

                process-type,N
                    Procesa tips un numurs (piemēram, history syncer,1)

                pid Procesa identifikators

                scope
                    (rwlock, mutex, processing) var izmantot kopā ar process-type
                    (piemēram, history syncer,1,processing)

        -T, --test-config
               Validēt konfigurācijas failu un iziet.

        -h, --help
               Parādīt šo palīdzību un iziet.

        -V, --version
               Izvadīt versijas informāciju un iziet.

FAILI
        /usr/local/etc/zabbix_server.conf
               Zabbix servera konfigurācijas faila noklusējuma atrašanās vieta
               (ja kompilēšanas laikā nav mainīta).

SKATĪT ARĪ
        Dokumentācija

        zabbix_agentd(8),  zabbix_get(1),  zabbix_proxy(8),   zabbix_sender(1),
        zabbix_js(1), zabbix_agent2(8), zabbix_web_service(8)

Zabbix                            2022-01-06                   ZABBIX_SERVER(8)
```

zabbix_web_service

```
ZABBIX_WEB_SERVICE(8)       Sistēmas administratora rokasgrāmata      ZABBIX_WEB_SERVICE(8)

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 lietojumprogramma, kas nodrošina tīmekļa pakalpo-
       jumus Zabbix komponentiem.

OPCIJAS
       -c, --config config-file
              Izmantot alternatīvu config-file noklusējuma faila vietā.

       -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_web_service.conf
              Zabbix tīmekļa pakalpojuma konfigurācijas faila noklusējuma atra-
              šanās vieta (ja tā nav mainīta kompilēšanas laikā).

SKATĪT ARĪ
       Dokumentācija

       zabbix_agentd(8),  zabbix_get(1),  zabbix_proxy(8),   zabbix_sender(1),
       zabbix_server(8), zabbix_js(1), zabbix_agent2(8)

Zabbix                            2022-01-06             ZABBIX_WEB_SERVICE(8)
```

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.