Documentation 7.0

ZABBIX

02.04.2026

Contents

Manual do usuário **7**

Bem-vindo à Documentação do Zabbix **7**

Primeiros passos com o Zabbix . . . 7
Zabbix Cloud . . . 7
Central do desenvolvedor . . . 7
Comunidade e outros recursos . . . 7
1. Introdução . . . 7
1 Estrutura do manual . . . 8
2 O que é o Zabbix . . . 8
3 Funcionalidades do Zabbix . . . 9
4 Visão geral do Zabbix . . . 10
5 Novidades do Zabbix 7.0.0 . . . 11
6 O que há de novo no Zabbix 7.0.1 . . . 26
7 Novidades do Zabbix 7.0.2 . . . 27
8 Novidades do Zabbix 7.0.3 . . . 27
9 Novidades do Zabbix 7.0.4 . . . 28
10 Novidades do Zabbix 7.0.5 . . . 29
11 Novidades do Zabbix 7.0.6 . . . 29
12 Novidades do Zabbix 7.0.7 . . . 30
13 O que há de novo no Zabbix 7.0.8 . . . 31
14 Novidades do Zabbix 7.0.9 . . . 31
15 Novidades no Zabbix 7.0.10 . . . 31
16 Novidades do Zabbix 7.0.11 . . . 32
17 Novidades do Zabbix 7.0.12 . . . 33
18 Novidades no Zabbix 7.0.13 . . . 33
19 Novidades no Zabbix 7.0.14 . . . 33
20 Novidades do Zabbix 7.0.15 . . . 34
21 O que há de novo no Zabbix 7.0.16 . . . 34
22 Novidades do Zabbix 7.0.17 . . . 34
23 Novidades do Zabbix 7.0.18 . . . 35
24 Novidades do Zabbix 7.0.19 . . . 35
25 Novidades do Zabbix 7.0.20 . . . 35
26 Novidades do Zabbix 7.0.21 . . . 36
27 Novidades do Zabbix 7.0.22 . . . 36
28 Novidades do Zabbix 7.0.23 . . . 36
29 Novidades do Zabbix 7.0.24 . . . 36
30 Novidades no Zabbix 7.0.25 . . . 37
2. Definições . . . 37
3 Processos do Zabbix . . . 39
1 Server . . . 39
2 Agent . . . 47
3 Agent 2 . . . 51
4 Proxy . . . 55
5 Java gateway . . . 59
6 Sender . . . 62
7 Get . . . 63
8 JS . . . 64
9 Serviço da Web . . . 65
4 Instalação . . . 66
1 Obtendo o Zabbix . . . 66

2 Requisitos . . . 66
3 Instalação a partir do código-fonte . . . 76
4 Instalação a partir de pacotes . . . 92
5 Instalação a partir de containers . . . 106
6 Instalação da interface web . . . 115
7 Procedimento de atualização . . . 120
8 Problemas conhecidos . . . 132
10 Alterações em templates . . . 145
11 Notas de atualização para 7.0.0 . . . 148
12 Notas de atualização para 7.0.1 . . . 154
13 Notas de atualização para 7.0.2 . . . 155
14 Notas de atualização para 7.0.3 . . . 155
15 Notas de atualização para 7.0.4 . . . 155
16 Notas de atualização para 7.0.5 . . . 156
17 Notas de atualização para 7.0.6 . . . 157
18 Notas de atualização para 7.0.7 . . . 157
19 Notas de atualização para 7.0.8 . . . 157
20 Notas de atualização para 7.0.9 . . . 157
21 Notas de atualização para 7.0.10 . . . 157
22 Notas de atualização para 7.0.11 . . . 158
23 Notas de atualização para 7.0.12 . . . 158
24 Notas de atualização para 7.0.13 . . . 158
25 Notas de atualização para 7.0.14 . . . 158
26 Notas de atualização para 7.0.15 . . . 159
27 Notas de atualização para 7.0.16 . . . 159
28 Notas de atualização para 7.0.17 . . . 159
29 Notas de atualização para 7.0.18 . . . 159
30 Notas de atualização para 7.0.19 . . . 159
31 Notas de atualização para 7.0.20 . . . 159
32 Notas de atualização para 7.0.21 . . . 160
33 Notas de atualização para 7.0.22 . . . 160
34 Notas de atualização para 7.0.23 . . . 160
35 Notas de atualização para 7.0.24 . . . 160
36 Notas de atualização para 7.0.25 . . . 161
5. Início rápido . . . 161
1 Autenticando e configurando usuário . . . 161
2 Novo host . . . 165
3 Novo item . . . 167
4 Novo trigger . . . 168
5 Recebendo notificação de problema . . . 170
6 Novo modelo (template) . . . 174
6 Zabbix appliance . . . 176
7 Configuração . . . 179
1 Hosts e grupos de hosts . . . 189
2 Items . . . 200
3 Triggers . . . 407
4 Eventos . . . 426
5 Correlação de eventos . . . 430
6 Marcação . . . 437
7 Visualização . . . 439
8 Templates e grupos de templates . . . 465
9 Modelos prontos para uso . . . 465
10 Notificações em eventos . . . 478
11 Macros . . . 523
12 Usuários e grupos de usuários . . . 538
13 Armazenamento de segredos . . . 547
14 Relatórios agendados . . . 554
15 Exportação de dados . . . 557
8. Monitoramento de serviço . . . 563
1 Árvore de serviços . . . 564
2 SLA . . . 567
3 Exemplo de configuração . . . 569
9. Monitoramento web . . . 573

1 Itens de monitoramento web . . . 583
2 Cenário real . . . 584
10. Monitoramento de máquina virtual . . . 590
1 Chaves de item de monitoramento da VMware . . . 592
2 Campos de chave de descoberta de máquina virtual . . . 610
3 exemplos de JSON para items VMware . . . 614
4 Exemplo de configuração de monitoramento do VMware . . . 618
11. Manutenção . . . 623
12. Expressões regulares . . . 628
13 Reconhecimento do Problema . . . 634
1 Supressão de problemas . . . 636
14. Configuração exportação/importação . . . 637
1 Grupos de templates . . . 639
2 Grupos de hosts . . . 639
3 Templates . . . 639
4 Hosts . . . 655
5 Mapas de rede . . . 668
5 Tipos de mídia . . . 676
15. Descoberta . . . 683
1 descoberta de rede . . . 683
2 Registro automático de agente ativo . . . 692
3 Descoberta de baixo nível . . . 696
16 Monitoração distribuída . . . 746
1 Proxies . . . 747
17. Criptografia . . . 755
1 Usando certificados . . . 762
2 Usando chaves pré-compartilhadas . . . 769
3 Solução de problemas . . . 772
18. Interface da Web . . . 774
1 Menu . . . 775
2 Seções de front-end . . . 782
3 Configurações do usuário . . . 997
4 Pesquisa global . . . 1001
5 Modo de manutenção de front-end . . . 1003
6 Parâmetros da página . . . 1004
7 Definições . . . 1005
8 Criando seu próprio tema . . . 1006
9 Modo de depuração . . . 1006
10 Cookies usados pelo Zabbix . . . 1007
11 Fuso horário . . . 1008
12 Redefinindo senha . . . 1009
13 Seletor de período de tempo . . . 1009
19 Melhores práticas . . . 1011
1 Melhores práticas de segurança . . . 1011
2 Melhores práticas de configuração . . . 1020
20 API . . . 1021
referência do método . . . 1026
Apêndice 1. Comentário de referência . . . 1793
Apêndice 2. Alterações de 6.4 para 7.0 . . . 1799
Apêndice 3. Mudanças na versão 7.0 . . . 1803
21 Extensões . . . 1805
1 Módulos carregáveis . . . 1807
2 Plugins . . . 1816
3 Módulos do frontend . . . 1823
22 Apêndices . . . 1824
1 Instalação e configuração . . . 1824
2 Configuração do processo . . . 1886
3 Protocolos . . . 1965
4 Items . . . 1990
5 Funções suportadas . . . 2027
6 Macros . . . 2058
7 Símbolos de unidade . . . 2106
8 Sintaxe do período de tempo . . . 2108

9 Execução de comandos . . . 2109
10 Compatibilidade de versões . . . 2110
12 Biblioteca de vínculo dinâmico do Zabbix sender para Windows . . . 2111
13 Atualização do monitoramento de serviços . . . 2112
14 Outros problemas . . . 2112
15 Comparação entre agent e agent 2 . . . 2113
16 Exemplos de escape . . . 2114
23 Guias de referência rápida . . . 2116
Visão geral . . . 2116
1 Monitorar Linux com o agent do Zabbix . . . 2116
2 Monitor Windows with Zabbix agent . . . 2120
3 Monitorar o Apache via HTTP . . . 2123
4 Monitorar MySQL com o Zabbix agent 2 . . . 2127
5 Monitorar VMware com o Zabbix . . . 2134
6 Monitorar o tráfego de rede com o Zabbix . . . 2138
7 Monitorar o tráfego de rede usando verificações ativas . . . 2141
8 Monitorar sites com itens de Navegador . . . 2144
9 Monitorar certificados de sites com o Zabbix agent 2 (passivo) . . . 2149
10 Monitore um switch ou roteador de rede com o Zabbix . . . 2152
11 Monitorar o log de eventos do Windows usando verificações ativas . . . 2159

Zabbix Cloud **2163**

Zabbix Cloud **2163**
Configurar e gerenciar o Zabbix Cloud . . . 2163
Saiba mais sobre o Zabbix Cloud . . . 2163
1 Implante o Zabbix na nuvem . . . 2163
2 Configuração do nó . . . 2167
Visão Geral . . . 2167
Aba Visão Geral . . . 2168
Aba Filtros de acesso . . . 2169
Aba Criptografia . . . 2170
Aba Backups . . . 2171
Aba Histórico . . . 2171
Aba Manutenção . . . 2172
Aba de Atualização . . . 2173
3 Adicionando usuários . . . 2175
Introdução . . . 2175
Adicionando usuários às organizações . . . 2175
Adicionando usuários aos nós . . . 2176
4 Principais diferenças do Zabbix Cloud . . . 2177

Central do desenvolvedor **2179**

Central do desenvolvedor **2179**
Construa no Zabbix . . . 2179
Módulos . . . 2179
Estrutura do arquivo do módulo . . . 2179
Widgets . . . 2187
Tutoriais . . . 2195
Exemplos . . . 2221
Biblioteca Python para o Zabbix . . . 2221
Instalação . . . 2221
Guia de início rápido . . . 2222
Usar a API do Zabbix . . . 2225
Coletar dados do agent Zabbix . . . 2228
Enviar dados para o Zabbix server ou proxy . . . 2230
Log de depuração . . . 2234
Plugins . . . 2234
Exemplos . . . 2237
Criar um plugin (tutorial) . . . 2237
Interfaces de plug-in . . . 2242
Mudanças no desenvolvimento de extensões . . . 2242

Páginas de manual do Zabbix **2242**

zabbix_agent2 . . . 2242
NOME . . . 2242
SINOPSE . . . 2243
DESCRIÇÃO . . . 2243
OPÇÕES . . . 2243
ARQUIVOS . . . 2244
VEJA TAMBÉM . . . 2244
Índice . . . 2244
NOME . . . 2244
SINOPSE . . . 2244
DESCRIÇÃO . . . 2244
OPÇÕES . . . 2245
ARQUIVOS . . . 2245
VEJA TAMBÉM . . . 2245
Índice . . . 2246
zabbix_get . . . 2246
NOME . . . 2246
SINOPSE . . . 2246
DESCRIÇÃO . . . 2246
OPÇÕES . . . 2246
EXEMPLOS . . . 2247
VEJA TAMBÉM . . . 2247
Índice . . . 2248
zabbix_js . . . 2248
NOME . . . 2248
SINOPSE . . . 2248
DESCRIÇÃO . . . 2248
OPÇÕES . . . 2248
EXEMPLOS . . . 2249
VEJA TAMBÉM . . . 2249
Índice . . . 2249
zabbix_proxy . . . 2249
NOME . . . 2249
SINOPSE . . . 2249
DESCRIÇÃO . . . 2249
OPÇÕES . . . 2249
ARQUIVOS . . . 2251
VEJA TAMBÉM . . . 2251
Índice . . . 2251
zabbix_sender . . . 2251
NOME . . . 2251
SINOPSE . . . 2251
DESCRIÇÃO . . . 2252
OPÇÕES . . . 2252
STATUS DE SAÍDA . . . 2254
EXEMPLOS . . . 2254
VEJA TAMBÉM . . . 2255
Índice . . . 2255
zabbix_server . . . 2255
NOME . . . 2255
SINOPSE . . . 2255
DESCRIÇÃO . . . 2255
OPÇÕES . . . 2255
ARQUIVOS . . . 2257
VEJA TAMBÉM . . . 2257
Índice . . . 2257
zabbix_web_service . . . 2257
NOME . . . 2257
SINOPSE . . . 2257
DESCRIÇÃO . . . 2257
OPÇÕES . . . 2258
ARQUIVOS . . . 2258
VEJA TAMBÉM . . . 2258

Index 2258

Aviso de direitos autorais **2258**

Manual do usuário

Bem-vindo à Documentação do Zabbix

Seu recurso de referência para trabalhar com o monitoramento Zabbix, desde configurações básicas até configurações avançadas. Este manual cobre tudo o que é necessário para instalar, configurar e operar o Zabbix.

Novidades no Zabbix 7.0

Primeiros passos com o Zabbix

Instalação Instruções passo a passo para instalar o Zabbix na sua plataforma preferida, abrangendo vários sistemas operacionais e configurações de banco de dados.

Requisitos Uma lista de plataformas suportadas e pré-requisitos de software para ajudá-lo a preparar seu ambiente para uma implantação bem-sucedida do Zabbix.

Guias de início rápido Tutoriais concisos e orientados a tarefas que o guiam pelo básico — desde os primeiros passos de configuração até o recebimento do primeiro alerta de problema.

Zabbix Cloud

Comece a usar o Zabbix Cloud Instruções guiadas para colocar sua instância Zabbix Cloud em funcionamento rapidamente, incluindo a configuração inicial e a conexão dos seus dispositivos monitorados.

Zabbix Cloud vs. on-premises Compare as diferenças de gerenciamento, escalabilidade e manutenção entre as duas implantações e descubra qual pode suportar seu fluxo de trabalho de forma mais eficiente.

Explore o Zabbix Cloud Obtenha uma visão geral do Zabbix Cloud, incluindo seus principais recursos, vantagens e o que está incluído na assinatura. Para perguntas comuns sobre configuração e uso, consulte o FAQ.

Central do desenvolvedor

Módulos do frontend Guias e referências para criar módulos personalizados que estendem ou modificam o frontend do Zabbix para atender a casos de uso específicos.

Widgets Uma análise da estrutura e lógica dos widgets, com instruções para criar elementos personalizados do dashboard adaptados às suas necessidades.

Plugins Uma visão geral sobre como desenvolver e gerenciar plugins do Zabbix para estender a funcionalidade ou integrar com sistemas externos.

Comunidade e outros recursos

Fóruns Zabbix Um espaço para trocar ideias, encontrar soluções e compartilhar experiências em todos os níveis de conhecimento do Zabbix.

Blog do Zabbix Notícias, tutoriais e estudos de caso selecionados pela equipe do Zabbix e colaboradores da comunidade.

Vídeos tutoriais Uma biblioteca de vídeos com demonstrações, discussões e dicas para ajudá-lo a aproveitar ao máximo o Zabbix.

1. Introdução

Por favor utilize a barra lateral para acessar o conteúdo na seção de Introdução.

1 Estrutura do manual

Estrutura

O conteúdo deste manual é dividido em seções e subseções para fornecer fácil acesso a assuntos de interesse em particular.

Quando você navega até seções específicas, certifique-se de que você expanda as pastas da seção para revelar o conteúdo completo do que está incluso nas subseções e páginas individuais.

Associação cruzada entre páginas de conteúdo relacionado é fornecida tanto quanto possível de modo a garantir que informações relevantes não passem despercebidas pelos usuários.

Seções

Introdução fornece informações gerais sobre o software Zabbix atual. A leitura desta seção deve fornecer bons motivos para escolher o Zabbix.

Definições do Zabbix explicam a terminologia usada no Zabbix, enquanto Processos do Zabbix fornece detalhes sobre os componentes do Zabbix.

As seções Instalação e Início rápido devem ajudá-lo a começar a usar o Zabbix. Zabbix appliance é uma alternativa para obter rapidamente uma amostra de como é usar o Zabbix.

Configuração é uma das maiores e mais importantes seções deste manual. Ela contém muitos conselhos essenciais sobre como configurar o Zabbix para monitorar seu ambiente, desde a configuração de hosts até a coleta de dados essenciais e sua visualização, bem como a configuração de notificações e comandos remotos a serem executados em caso de problemas.

Monitoramento de serviços detalha como usar o Zabbix para uma visão geral de alto nível do seu ambiente de monitoramento.

Monitoramento web deve ajudá-lo a aprender como monitorar a disponibilidade de sites.

Monitoramento de máquinas virtuais apresenta um tutorial para configurar o monitoramento de ambientes VMware.

Manutenção, Expressões regulares, Reconhecimento de problemas e Exportação/importação de configuração são outras seções que mostram como usar esses vários aspectos do software Zabbix.

Descoberta contém instruções para configurar a descoberta automática de dispositivos de rede, agents ativos, sistemas de arquivos, interfaces de rede, etc.

Monitoramento distribuído trata das possibilidades de usar o Zabbix em ambientes maiores e mais complexos.

Criptografia ajuda a explicar as possibilidades de criptografar as comunicações entre os componentes do Zabbix.

Interface web contém informações específicas para o uso da interface web do Zabbix.

Melhores práticas contém dicas úteis para uma experiência do usuário ideal e segura.

A seção API apresenta detalhes sobre o trabalho com a API do Zabbix.

Extensões é uma coleção de maneiras de estender o Zabbix.

Listas detalhadas de informações técnicas estão incluídas nos Apêndices.

Guias de referência rápida contêm guias passo a passo para realizar tarefas com o Zabbix.

2 O que é o Zabbix

Visão geral

O Zabbix foi criado por Alexei Vladishev e atualmente é desenvolvido e suportado ativamente pela Zabbix SIA.

O Zabbix é uma solução de monitoramento distribuído de código aberto de classe empresarial.

O Zabbix é um software que monitora inúmeros parâmetros de uma rede e a integridade e saúde de servidores, máquinas virtuais, aplicações, serviços, bancos de dados, sites, nuvem e muito mais. O Zabbix utiliza um mecanismo de notificação flexível que permite aos usuários configurar alertas por e-mail para praticamente qualquer evento. Isso permite uma reação rápida a problemas nos servidores. O Zabbix oferece excelentes recursos de relatórios e visualização de dados com base nos dados armazenados. Isso torna o Zabbix ideal para planejamento de capacidade.

O Zabbix suporta tanto polling quanto trapping. Todos os relatórios e estatísticas do Zabbix, bem como os parâmetros de configuração, são acessados através de um frontend baseado na web. Um frontend baseado na web garante que o status da sua

rede e a integridade dos seus servidores possam ser avaliados de qualquer local. Quando devidamente configurado, o Zabbix pode desempenhar um papel importante no monitoramento da infraestrutura de TI. Isso é igualmente verdadeiro para pequenas organizações com poucos servidores e para grandes empresas com uma infinidade de servidores.

O Zabbix é gratuito. O Zabbix é escrito e distribuído sob a licença AGPL-3.0. Isso significa que seu código-fonte é distribuído livremente e está disponível para o público em geral.

Suporte comercial está disponível e é fornecido pela Zabbix Company e seus parceiros em todo o mundo.

Saiba mais sobre os recursos do Zabbix.

Usuários do Zabbix

Muitas organizações de diferentes tamanhos ao redor do mundo confiam no Zabbix como a plataforma principal de monitoramento.

3 Funcionalidades do Zabbix

Visão geral

O Zabbix é uma solução de monitoramento de rede altamente integrada, oferecendo uma multiplicidade de recursos em um único pacote.

Coleta de dados

- verificações de disponibilidade e desempenho
- suporte para SNMP (tanto trap quanto polling), IPMI, JMX, monitoramento VMware
- verificações personalizadas
- coleta de dados desejados em intervalos personalizados
- realizado por server/proxy e por agents

Definições de limiares flexíveis

- você pode definir limiares de problema muito flexíveis, chamados triggers, referenciando valores do banco de dados backend

Alertas altamente configuráveis

- o envio de notificações pode ser personalizado para o cronograma de escalonamento, destinatário, tipo de mídia
- as notificações podem ser significativas e úteis usando variáveis de macro
- ações automáticas incluem comandos remotos

Gráficos em tempo real

- os items monitorados são imediatamente exibidos em gráficos usando a funcionalidade de gráficos incorporada

Capacidades de monitoramento web

- o Zabbix pode seguir um caminho de cliques simulados em um site e verificar a funcionalidade e o tempo de resposta

Opções extensas de visualização

- capacidade de criar gráficos personalizados que podem combinar múltiplos items em uma única visualização
- mapas de rede
- apresentações de slides em uma visão geral no estilo dashboard
- relatórios
- visão de alto nível (negócios) dos recursos monitorados

Armazenamento de dados históricos

- dados armazenados em um banco de dados
- histórico configurável
- procedimento de limpeza incorporado

Configuração fácil

- adicione dispositivos monitorados como hosts
- hosts são selecionados para monitoramento, uma vez no banco de dados
- aplique templates aos dispositivos monitorados

Uso de templates

- agrupamento de verificações em templates
- templates podem herdar outros templates

Descoberta de rede

- descoberta automática de dispositivos de rede
- autorregistro de agent
- descoberta de sistemas de arquivos, interfaces de rede e OIDs SNMP

Interface web rápida

- um frontend baseado na web em PHP
- acessível de qualquer lugar
- você pode navegar clicando
- log de auditoria

Zabbix API

- a Zabbix API fornece uma interface programável para o Zabbix para manipulações em massa, integração com software de terceiros e outros propósitos.

Sistema de permissões

- autenticação segura de usuários
- certos usuários podem ser limitados a certas visualizações

Agent completo e facilmente extensível

- implantado nos alvos de monitoramento
- pode ser implantado tanto em Linux quanto em Windows

Daemons binários

- escrito em C, para desempenho e baixo consumo de memória
- facilmente portável

Pronto para ambientes complexos

- monitoramento remoto facilitado pelo uso de um Zabbix proxy

4 Visão geral do Zabbix

Arquitetura

O Zabbix consiste de vários componentes de software importantes. Suas responsabilidades estão resumidas abaixo.

Server

O Zabbix server é o componente central para o qual os agentes reportam informações de disponibilidade e integridade e estatísticas. O Server é o repositório central no qual todas as configurações, estatísticas e dados operacionais são armazenados.

Armazenamento banco de dados

Todas as informações de configuração assim como os dados coletados pelo Zabbix são armazenados em um banco de dados.

Interface web

Para fácil acesso ao Zabbix de qualquer lugar e de qualquer plataforma, a interface baseada em web é oferecida. A interface é parte do Zabbix Server, e usualmente (mas não necessariamente) é executada na mesma máquina física na qual está sendo executado o Server.

Proxy

O Zabbix proxy pode coletar dados de performance e disponibilidade em nome do Zabbix Server. Um proxy é uma parte opcional da implementação do Zabbix; no entanto, ele pode ser de grande benefício para distribuir a carga de um único Zabbix Server.

Agente

Os agentes do Zabbix são implantados em alvos de monitoramento para monitorar ativamente recursos e aplicativos locais e relatar os dados coletados ao Zabbix server. Desde o Zabbix 4.4, há dois tipos de agentes disponíveis: o Zabbix agent (leve, suportado em muitas plataformas, escrito em C) e o Zabbix agent 2 (extra-flexível, facilmente extensível com plug-ins, escrito em Go).

Fluxo de dados

Em adição é importante voltar um passo atrás e dar uma olhada geral no fluxo de dados dentro do Zabbix. De modo a criar um item que colete dados você deve primeiro criar um host. Indo para o outro lado do espectro do Zabbix você deve primeiro ter um

item para criar um gatilho. Você deve ter um gatilho para criar uma ação. Portanto se você quiser receber um alerta de que sua carga de CPU está muito alta no *Servidor X* você deve primeiro criar uma entrada de host para o *Servidor X* seguido por um item para monitoramento de sua CPU, e então um gatilho que é ativado se a CPU estiver muito alta, seguida de uma ação que envia um e-mail para você. Ainda que isto possa parecer um monte de passos, com o uso de modelos realmente não é. No entanto, devido a este desenho é possível criar uma configuração muito flexível.

5 Novidades do Zabbix 7.0.0

Veja as alterações incompatíveis para esta versão.

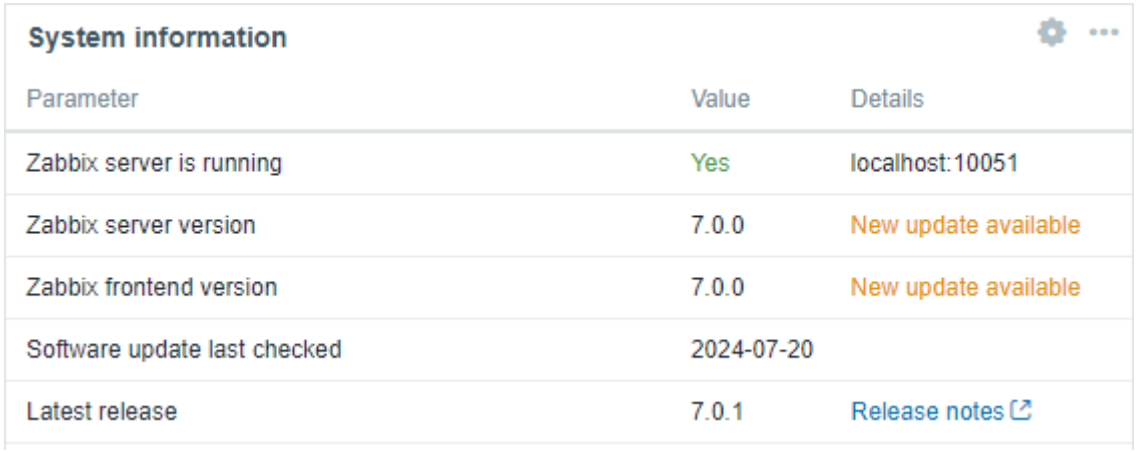
Licença AGPL-3.0

O software Zabbix agora é escrito e distribuído sob a licença AGPL-3.0 (anteriormente licença GPL v2.0).

Verificação de atualização de software

Uma verificação de atualização de software agora é adicionada por padrão a instalações novas e existentes - o frontend do Zabbix se comunicará com o endpoint público do Zabbix para verificar atualizações.

As notícias sobre atualizações de software Zabbix disponíveis são exibidas em *Relatórios* -> *Informações do sistema* e (opcionalmente) no widget *Informações do sistema* do dashboard.



Você pode desabilitar a verificação de atualização de software definindo AllowSoftwareUpdateCheck=0 na configuração do server.

Pollers assíncronos

Novos processos de poller foram adicionados, capazes de executar várias verificações ao mesmo tempo:

- `agent poller`
- `http agent poller`
- `snmp poller` (para itens `walk[OID]` e `get[OID]`)

Esses pollers são assíncronos - eles são capazes de iniciar novas verificações sem a necessidade de esperar por uma resposta, com concorrência configurável de até 1000 verificações simultâneas.

Os pollers assíncronos foram desenvolvidos porque, em comparação, os processos de poller síncronos podem executar apenas uma verificação ao mesmo tempo e a maior parte do tempo é gasta esperando pela resposta. Assim, a eficiência pode ser aumentada iniciando novas verificações paralelas enquanto se aguarda a resposta da rede, e os novos pollers fazem isso.

Você pode iniciar pollers assíncronos de agent modificando o valor de StartAgentPollers - um novo parâmetro do server/proxy. Os pollers de HTTP agent podem ser iniciados modificando StartHTTPAgentPollers respectivamente. Os pollers de SNMP podem ser iniciados modificando StartSNMPPollers respectivamente.

A concorrência máxima para pollers assíncronos (agent, HTTP agent e SNMP) é definida por MaxConcurrentChecksPerPoller.

Observe que após a atualização todas as verificações de agent, HTTP agent e SNMP `walk[OID]` serão movidas para pollers assíncronos.

Como parte do desenvolvimento, o recurso de conexões persistentes do cURL foi adicionado às verificações de HTTP agent.

Monitoramento de navegador

Um novo tipo de item - Item de navegador - foi adicionado ao Zabbix, permitindo o monitoramento de sites e aplicações web complexas usando um navegador. Os itens de navegador permitem a execução de código JavaScript definido pelo usuário para simular ações relacionadas ao navegador, como clicar, inserir texto, navegar por páginas da web, etc.

Este item coleta dados via HTTP/HTTPS e implementa parcialmente o padrão W3C WebDriver com Selenium Server ou um WebDriver simples (por exemplo, ChromeDriver) como endpoint de teste.

Observe que o suporte a itens de navegador é atualmente experimental.

Além disso, este recurso adiciona o template *Website by Browser* e novos elementos para exportação/importação de configuração, arquivos de configuração do Zabbix server/proxy, timeouts e a ferramenta de linha de comando `zabbix_js`. Para mais informações, consulte Notas de atualização para 7.0.0.

Balanceamento de carga e alta disponibilidade do proxy

O balanceamento de carga do proxy é implementado pela introdução de grupos de proxy no Zabbix. Os grupos de proxy fornecem distribuição automática de hosts entre proxies, reequilíbrio de carga do proxy e alta disponibilidade - quando um proxy fica offline, seus hosts são imediatamente distribuídos entre outros proxies no grupo.

Para mais informações, consulte balanceamento de carga e alta disponibilidade do proxy.

Buffer de memória do proxy

Um buffer de memória foi desenvolvido para o Zabbix proxy. O buffer de memória permite armazenar novos dados (valores de item, descoberta de rede, autorregistro de host) no buffer e enviá-los para o Zabbix server sem acessar o banco de dados.

Em instalações anteriores ao Zabbix 7.0, os dados coletados eram armazenados no banco de dados antes de serem enviados para o Zabbix server. Para essas instalações, esse continua sendo o comportamento padrão após a atualização.

Para desempenho otimizado, recomenda-se configurar o uso do buffer de memória no proxy. Isso é possível modificando o valor de ProxyBufferMode de ”disk” (padrão fixo para instalações existentes) para ”hybrid” (recomendado) ou ”memory”. Também é necessário definir o tamanho do buffer de memória (parâmetro ProxyMemoryBufferSize).

No modo híbrido, o buffer é protegido contra perda de dados, liberando dados não enviados para o banco de dados se o proxy for interrompido, o buffer estiver cheio ou os dados forem muito antigos. Quando todos os valores forem liberados para o banco de dados, o proxy volta a usar o buffer de memória.

No modo de memória, o buffer de memória será usado, no entanto, não há proteção contra perda de dados. Se o proxy for interrompido ou a memória ficar sobrecarregada, os dados não enviados serão descartados.

O modo híbrido (ProxyBufferMode=hybrid) é aplicado a todas as novas instalações desde o Zabbix 7.0.

Parâmetros adicionais, como ProxyMemoryBufferSize e ProxyMemoryBufferAge definem o tamanho do buffer de memória e a idade máxima dos dados no buffer, respectivamente.

Novos itens internos foram adicionados para monitorar o buffer de memória do proxy.

Provisionamento de usuário JIT

Anteriormente, os usuários provisionados estavam limitados apenas às mídias criadas durante o provisionamento, sem a flexibilidade de editar propriedades como horários de trabalho ou severidades.

Agora, mais flexibilidade está disponível para usuários provisionados no Zabbix:

- as mídias de usuário provisionadas podem ser desabilitadas/habilitadas;
- os campos de mídia do usuário provisionado, como *Quando ativo*, *Usar se severidade* e *Habilitado*, podem ser editados manualmente;
- mídias de usuário adicionais podem ser adicionadas (por exemplo, endereços de e-mail adicionais) para usuários provisionados manualmente;
- mídias de usuário adicionadas manualmente podem ser excluídas (as mídias de usuário provisionadas não podem).

Além disso, ao configurar o mapeamento de mídia de usuário para provisionamento, campos como *Quando ativo*, *Usar se severidade* e *Habilitado* agora estão disponíveis. Observe que as alterações no formulário de mapeamento de tipo de mídia de usuário terão efeito apenas para novas mídias criadas durante o provisionamento.

Timeouts configuráveis por item

A configuração de timeout por item agora está disponível para mais tipos de item (veja os tipos de item suportados). Além de definir os valores de timeout no nível do item, é possível definir timeouts globais e de proxy para vários tipos de item.

Os timeouts configurados no nível do item têm a maior prioridade. Por padrão, os timeouts globais são aplicados a todos os itens; no entanto, se os timeouts de proxy forem definidos, eles substituirão os globais.

Oracle DB obsoleto

O suporte ao Oracle como banco de dados backend foi descontinuado e espera-se que seja completamente removido em versões futuras.

Protocolo JSON para verificações passivas do agent

Um protocolo baseado em JSON para verificações passivas do agent foi implementado.

Para compatibilidade com agents mais antigos, foi adicionado um failover para o antigo protocolo em texto simples. Se o agent retornar "ZBX_NOTSUPPORTED", o Zabbix irá armazenar em cache a interface como protocolo antigo e tentará novamente a verificação enviando apenas a chave do item em texto simples.

O Zabbix get agora pode ser executado com uma nova opção `-P --protocol <valor>`, onde "valor" pode ser:

- auto - conecta usando o protocolo JSON, faz fallback e tenta novamente com o protocolo em texto simples (padrão);
- json - conecta usando a chave do protocolo JSON;
- plaintext - conecta usando o protocolo em texto simples, onde apenas a chave do item é enviada.

Se uma chave de item não for suportada, o Zabbix get retornará o código de saída 1.

Protocolos unificados do agent/agent2

Os protocolos do agent Zabbix e do agent 2 foram unificados ao alternar o agent Zabbix para o protocolo do agent 2. A diferença entre as solicitações/respostas do agent Zabbix e do agent 2 é expressa pelo valor da tag "variant" ("1" - agent Zabbix, "2" - agent 2).

Veja também: Verificações passivas e ativas do agent.

Suporte a intervalos flexíveis/agendados em checagens ativas

Intervalos flexíveis/agendados agora são suportados em checagens ativas tanto pelo Zabbix agent quanto pelo Zabbix agent 2 (anteriormente apenas pelo Zabbix agent 2).

Desativação automática de recursos perdidos

Recursos que não são mais descobertos pela descoberta de baixo nível agora podem ser desativados automaticamente. Eles podem ser desativados imediatamente, após um período de tempo especificado ou nunca (veja o novo parâmetro *Desativar recursos perdidos* na configuração da regra de descoberta).

Recursos perdidos (hosts, items, triggers) são marcados por um ícone na coluna de informações. O texto da dica de ferramenta fornece detalhes sobre seu status.

No mesmo desenvolvimento, o parâmetro *Manter período de recursos perdidos* foi renomeado para *Excluir recursos perdidos*, com opções para excluir imediatamente, após um período de tempo especificado ou nunca.

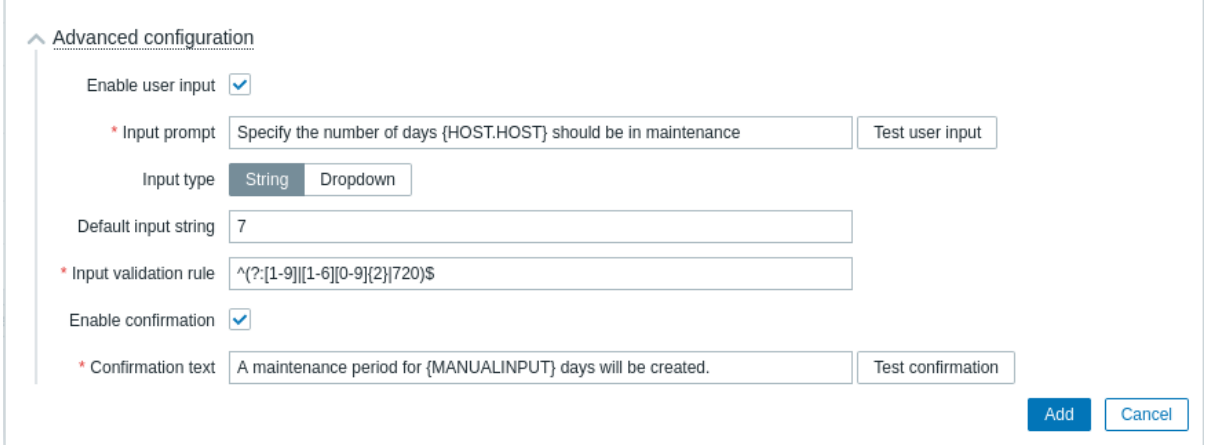
Entrada manual do usuário para scripts

A entrada manual do usuário para scripts no frontend permite fornecer um parâmetro personalizado a cada execução do script. Isso elimina a necessidade de criar vários scripts de usuário semelhantes com apenas uma diferença de parâmetro.

Por exemplo, você pode querer fornecer um inteiro diferente ou um endereço de URL diferente para o script durante a execução.

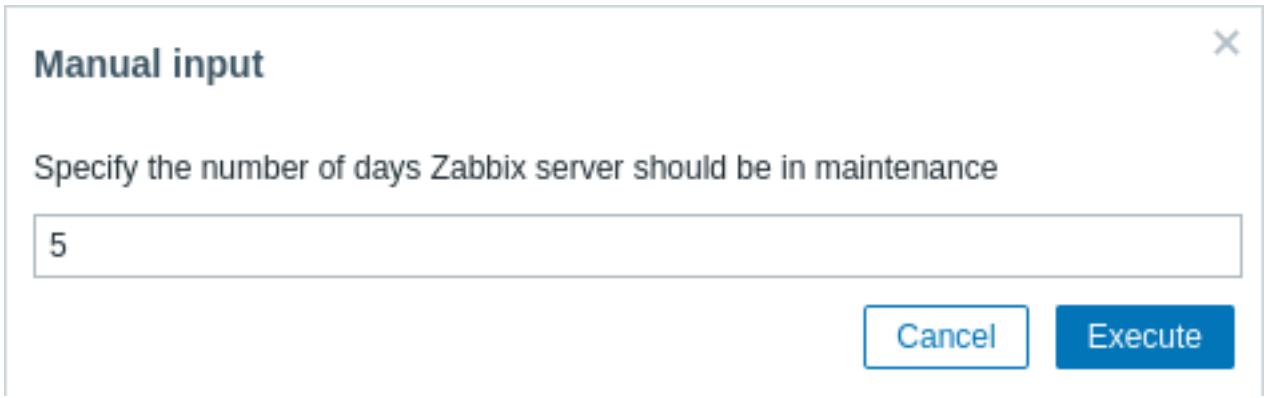
Para habilitar a entrada manual do usuário:

- use a macro {MANUALINPUT} no script (comandos, script, parâmetro do script) onde necessário; ou no campo URL de scripts de URL;
- na configuração avançada do script, habilite a entrada manual do usuário e configure as opções de entrada:



Com a entrada do usuário habilitada, antes da execução do script, um popup de *Entrada manual* aparecerá para o usuário solicitando o fornecimento de um valor personalizado. O valor fornecido substituirá {MANUALINPUT} no script.

Dependendo da configuração, o usuário será solicitado a inserir um valor de string ou selecionar o valor em uma lista suspensa de opções predefinidas.



Enviando dados para o Zabbix server via Zabbix API

Anteriormente, o envio de dados específicos para o Zabbix server era possível usando a ferramenta Zabbix sender ou implementando um protocolo de comunicação personalizado baseado em JSON, semelhante ao utilizado no Zabbix sender.

Agora também é possível enviar dados para o Zabbix server via protocolo HTTP usando o método da API `history.push`. Observe que o recebimento dos dados enviados requer um item trapper configurado ou um item HTTP agent (com trapping habilitado).

Além disso, as operações corretas de `history.push` são registradas em *Relatórios* → *Log de auditoria*, que possui opções adicionais de filtragem (uma nova ação *Push* e recurso *History*), e o método da API `history.push` também está disponível na *Lista de Permitir/Negar* de métodos da API ao configurar uma função de usuário.

Desempenho Reação mais rápida à atualização do período de manutenção do host

Anteriormente, as manutenções eram recalculadas apenas a cada minuto, causando uma possível latência de até 60 segundos para iniciar ou parar um período de manutenção.

Agora, as manutenções ainda são recalculadas a cada minuto ou assim que o cache de configuração é recarregado, caso haja alterações no período de manutenção.

A cada segundo, o processo de temporizador verifica se alguma manutenção deve ser iniciada/parada com base em alterações nos períodos de manutenção após a atualização da configuração. Assim, a velocidade de início/parada dos períodos de manutenção depende do intervalo de atualização da configuração (10 segundos por padrão). Observe que as alterações no período de manutenção não incluem as configurações *Ativo de/Ativo até*. Além disso, se um host/grupo de hosts for adicionado a um período de manutenção ativo existente, as alterações só serão ativadas pelo processo de temporizador no início do próximo minuto.

Verificações de permissão mais rápidas

As verificações de permissão ficaram muito mais rápidas com a introdução de várias tabelas intermediárias para verificar as permissões de usuários não privilegiados.

Essas tabelas mantêm hashes (SHA-256) de conjuntos de grupos de usuários e conjuntos de grupos de hosts para cada usuário/host, respectivamente. Além disso, há uma tabela de permissões que armazena apenas as combinações acessíveis de usuários e hosts, especificadas pelos IDs de hash.

Essa melhoria torna o carregamento de páginas do frontend com muitas permissões (ou seja, hosts, problemas) muito mais rápido. Observe que os hashes e permissões não são calculados para usuários Super-admin.

Execução mais rápida de ações de trigger

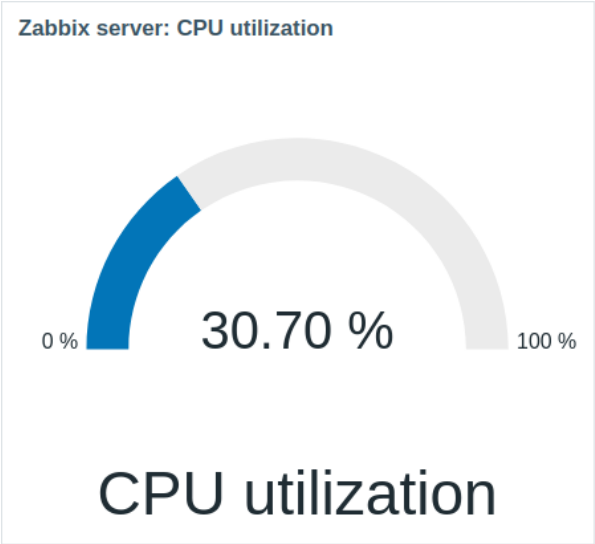
A execução de operações de ação de trigger, operação de recuperação e operação de atualização no Zabbix server agora ocorre imediatamente (menos de 100 milissegundos) após uma alteração no status do trigger, enquanto anteriormente os usuários podiam experimentar até 4 segundos de latência.

A redução da latência é possível graças à implementação de mecanismos de comunicação entre processos (IPC) entre vários processos (escalator e seu módulo escalation-dispatch, escalator e alerter, preprocessing manager e history syncer).

Widgets Vários novos widgets foram adicionados na nova versão, enquanto a funcionalidade disponível em outros foi aprimorada. Além disso, os widgets do dashboard agora podem se conectar e se comunicar entre si, tornando os widgets e dashboards mais dinâmicos.

Medidor

Um widget *Medidor* foi adicionado aos widgets do dashboard, permitindo exibir o valor de um único item como um medidor. Para mais informações, consulte *Medidor*.



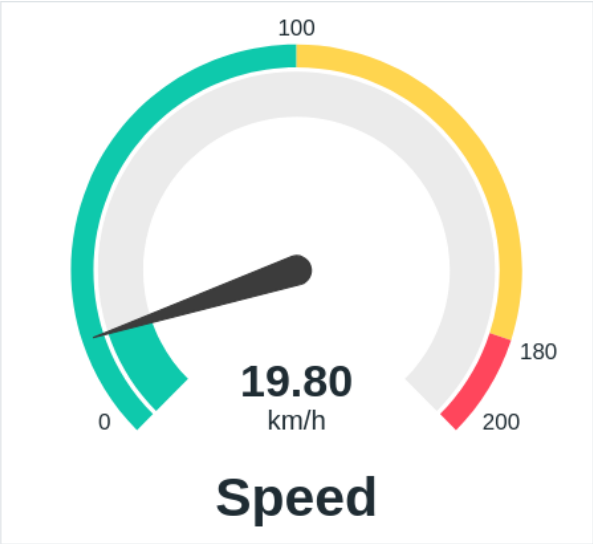


Gráfico de pizza

Um widget de *Gráfico de pizza* foi adicionado aos widgets do dashboard, permitindo exibir valores de itens selecionados como:

- gráfico de **pizza**;
- gráfico de **rosca**.

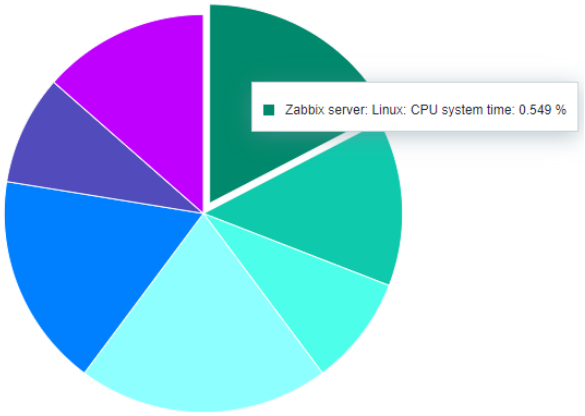


Gráfico de pizza.

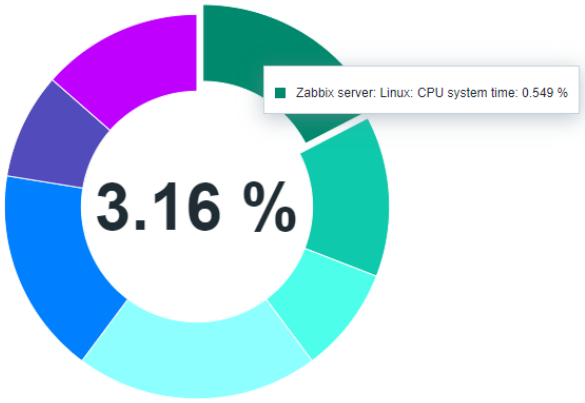


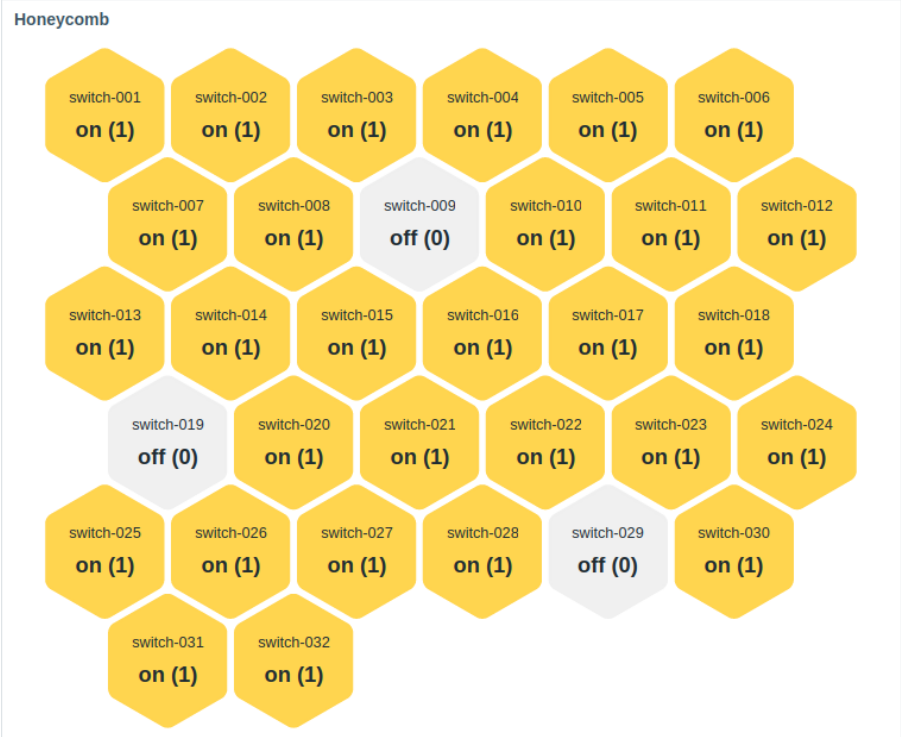
Gráfico de rosca.

Para mais informações, consulte *Gráfico de pizza*.

Como parte deste desenvolvimento, uma caixa de seleção *Exibir função de agregação* foi adicionada à configuração do widget gráfico (na aba *Legenda*).

Honeycomb

Um widget *Honeycomb* foi adicionado aos widgets do dashboard, que oferece uma visão geral dinâmica e vibrante da infraestrutura e dos recursos de rede monitorados, onde grupos de hosts, como máquinas virtuais e dispositivos de rede, juntamente com seus respectivos items, são representados visualmente como células hexagonais interativas. Para mais informações, consulte *Honeycomb*.



Principais triggers

Um widget *Principais triggers* foi adicionado aos widgets do dashboard, que permite visualizar os triggers com o maior número de problemas.

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

Para mais informações, consulte: *Principais triggers*.

Histórico de item e Texto simples

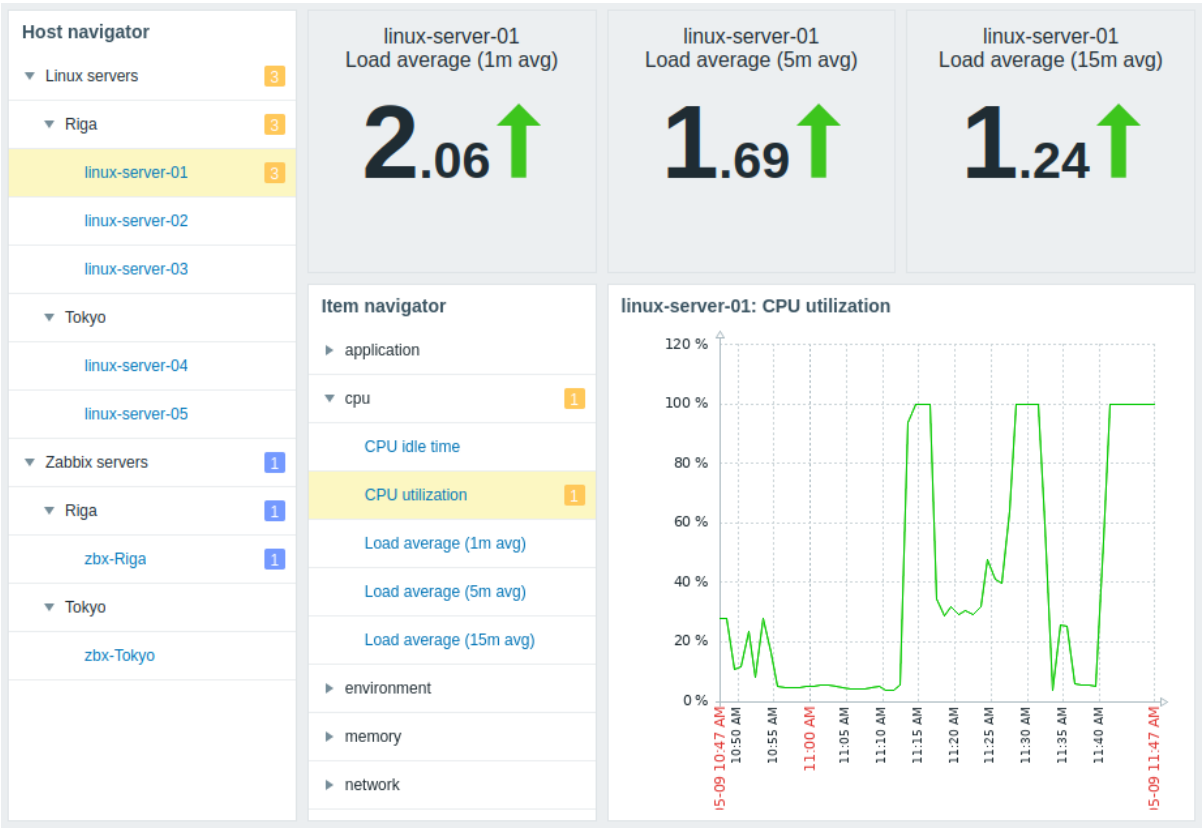
O novo *Histórico de item* widget de dashboard substituiu o widget *Texto simples*, oferecendo várias melhorias.

Ao contrário do widget *Texto simples*, que exibia apenas o dado mais recente do item em texto simples, o widget *Histórico de item* suporta várias opções de exibição para múltiplos tipos de item (numérico, caractere, log, texto e binário). Por exemplo, ele pode mostrar barras de progresso ou indicadores, imagens para tipos de dados binários (útil para itens de navegador), e destacar valores de texto (útil para monitoramento de arquivos de log).

Para mais informações, consulte *Histórico de item*. Para detalhes sobre a substituição do widget *Texto simples*, consulte Notas de atualização para 7.0.0.

Navegador de host e Navegador de item

Os widgets *Navegador de host* e *Navegador de item* foram adicionados aos widgets do dashboard. Esses widgets exibem hosts ou items, respectivamente, com base em várias opções de filtragem e agrupamento, e permitem controlar as informações exibidas em outros widgets com base no host ou item selecionado. Para mais informações, consulte *Navegador de host* e *Navegador de item*.



Estrutura de comunicação para widgets

Os widgets do dashboard agora podem se conectar e se comunicar entre si, tornando os widgets e dashboards mais dinâmicos. Vários widgets possuem parâmetros que permitem compartilhar dados de configuração entre widgets compatíveis ou com o dashboard.

Este recurso introduz as seguintes alterações:

- Os parâmetros *Grupos de hosts*, *Hosts* e *Item* permitem selecionar as respectivas entidades ou uma fonte de dados que as forneça.

- O parâmetro *Habilitar seleção de host* foi substituído pelo parâmetro *Substituir host*, que permite selecionar uma fonte de dados que forneça hosts.
- O parâmetro *Período de tempo* foi adicionado a vários widgets, permitindo selecionar uma fonte de dados que forneça um período de tempo.
- O parâmetro *Mapa* no widget *Mapa* permite selecionar um mapa ou outro widget como fonte de dados para mapas.
- O parâmetro *Gráfico* no widget *Gráfico (clássico)* permite selecionar um gráfico ou outro widget como fonte de dados para gráficos.

Dependendo do widget e de seus parâmetros, a fonte de dados pode ser um widget compatível do mesmo dashboard ou o próprio dashboard. Para mais informações, consulte Widgets do dashboard.

Para alterações nos templates padrão fornecidos com o Zabbix, consulte Alterações nos templates.

Períodos de tempo para agregação em widgets de valor de item/top hosts

Os períodos de tempo agora podem ser configurados nos widgets de Valor do item e Top hosts.

Agora também é possível exibir um valor agregado no widget de valor do item para o período escolhido. O valor agregado pode ser exibido como:

- mínimo
- máximo
- média
- contagem
- soma
- primeiro
- último

Esses recursos adicionais são úteis para criar widgets de comparação de dados. Por exemplo, em um widget você pode exibir o valor mais recente, enquanto em outro o valor médio para um período mais longo. Ou vários widgets podem ser usados para comparação lado a lado de valores agregados de vários períodos no passado.

Disponibilidade expandida de widgets em dashboards de template

Anteriormente, em um dashboard de template, era possível criar apenas os seguintes widgets: *Clock*, *Graph (classic)*, *Graph prototype*, *Item value*, *Plain text*, *URL*.

Agora, os dashboards de template suportam a criação de todos os widgets.

Ordenação estendida no widget Top hosts

Agora, além de ordenar por *Item value*, também é possível definir a coluna *Host name* ou *Text* como a coluna de ordenação no widget *Top hosts*.

Funcionalidade aumentada do widget de Disponibilidade do host

O widget de Disponibilidade do host agora permite exibir os hosts com a interface *Zabbix agent (active checks)*. Mais um status de disponibilidade foi adicionado, ou seja, *Misto*, que corresponde à situação em que pelo menos uma interface está indisponível e pelo menos uma está disponível ou desconhecida. Além disso, foi introduzida a possibilidade de ver apenas o total de hosts, sem detalhamento por interfaces.

Tamanho da legenda variável no widget de Gráfico

O widget *Gráfico* agora suporta a configuração de um número variável de linhas de legenda determinado pelo número de items configurados.

Funções Novas funções

Novas funções foram adicionadas para uso em expressões de trigger e itens calculados:

- **jsonpath()** - retorna o resultado do JSONPath;
- **xmlxpath()** - retorna o resultado do XML XPath.

Veja também: Funções de string

Funções atualizadas

Várias funções foram atualizadas:

- Funções agregadas agora também suportam tipos não numéricos para cálculo. Isso pode ser útil, por exemplo, com as funções count e count_foreach.
- As funções agregadas count e count_foreach suportam os parâmetros opcionais *operator* e *pattern*, que podem ser usados para ajustar a filtragem dos items e contar apenas valores que correspondam a determinados critérios.
- Todas as funções foreach não incluem mais items não suportados na contagem.

- A função **last_foreach**, anteriormente configurada para ignorar o argumento de período de tempo, agora o aceita como parâmetro opcional.
- O intervalo suportado para valores retornados pelas funções de previsão foi expandido para corresponder ao intervalo do tipo de dado double. Agora a função *timeleft()* pode aceitar valores de até 1.7976931348623158E+308 e a função *forecast()* pode aceitar valores de -1.7976931348623158E+308 até 1.7976931348623158E+308.

Items Período padrão consistente de armazenamento do histórico

O período padrão para manter o histórico do item foi padronizado para 31 dias no frontend e no banco de dados. Essa alteração afeta os formulários de configuração de item, item de template e protótipo de item, bem como a substituição do período de armazenamento do histórico na descoberta de baixo nível.

Valores de ponto flutuante truncados para itens inteiros

Agora, se um valor de ponto flutuante for recebido para um item inteiro sem sinal, o valor será truncado da parte decimal e salvo como um inteiro. Anteriormente, um valor de ponto flutuante tornaria um item inteiro não suportado.

Contagem de linhas no log de eventos do Windows

Um novo item `eventlog.count` foi adicionado ao agent/agent 2 do Zabbix no Windows. Este item retorna um valor inteiro com a contagem de linhas no log de eventos do Windows com base nos parâmetros especificados.

Requisições SNMP assíncronas de OID único

Um novo item SNMP `get[OID]` foi adicionado, permitindo consultar o valor de um único OID de forma assíncrona.

Itens internos

As verificações internas agora são tratadas por um novo processo `internal poller` do Zabbix server/proxy.

Itens internos foram adicionados para monitorar o buffer de memória do proxy:

- `zabbix[proxy_buffer,buffer,<mode>]` - retorna as estatísticas de uso do buffer de memória do proxy;
- `zabbix[proxy_buffer,state,changes]` - retorna o número de mudanças de estado entre os modos de buffer de disco/memória desde o início;
- `zabbix[proxy_buffer,state,current]` - retorna o estado de trabalho atual onde os novos dados estão sendo armazenados.

Os seguintes itens internos também foram adicionados:

- `zabbix[discovery_queue]` - permite monitorar o número de verificações de descoberta na fila;
- `zabbix[vps,written]` - permite monitorar o número total de valores de histórico gravados no banco de dados.

Novos e atualizados itens do agent

Novos itens foram adicionados ao agent/agent 2 do Zabbix:

- O item `net.dns.perf` retorna o número de segundos gastos aguardando uma resposta de um serviço, cronometrando a execução do item `net.dns`.
- O item `net.dns.get` do agent 2 do Zabbix retorna informações detalhadas do registro DNS.

Os seguintes itens do agent/agent 2 do Zabbix foram atualizados:

- Os itens `net.dns` e `net.dns.record` agora aceitam o nome DNS nos formatos invertido e não invertido ao realizar pesquisas reversas de DNS;
- Os itens `proc.get` nos modos ”process” e ”summary” agora também retornam a memória PSS (proportional set size) no Linux;
- Os itens `system.sw.packages` e `system.sw.packages.get` agora são suportados no Gentoo Linux;
- O item `system.hostname` agora pode retornar um Fully Qualified Domain Name, se a nova opção *fqdn* for especificada no parâmetro **type**;
- Os itens `wmi.get` e `wmi.getall` usados com o agent 2 do Zabbix agora retornam JSON com valores booleanos representados como strings (por exemplo, `"RealTimeProtectionEnabled": "True"` em vez de `"RealTimeProtectionEnabled": true` retornado anteriormente) para corresponder ao formato de saída desses itens no agent do Zabbix;
- O item `oracle.ts.discovery` do agent 2 do Zabbix agora retorna uma nova macro LLD {#CON_NAME} com o nome do container;
- O item `oracle.ts.stats` do agent 2 do Zabbix possui um novo parâmetro **conname** para especificar o nome do container de destino. O formato JSON dos dados retornados foi atualizado. Quando nenhum **tablespace**, **type** ou **conname** for especificado nos parâmetros da chave, os dados retornados incluirão um nível JSON adicional com o nome do container, permitindo a diferenciação entre containers.

Verificações simples

O item `vmware.eventlog` agora suporta filtragem opcional por severidade no terceiro parâmetro.

O item `vmware.vm.discovery` agora também retorna dados sobre as interfaces de rede da máquina virtual. Esses dados podem ser usados para configurar interfaces de host personalizadas.

O item `vmware.vm.net.if.discovery` agora também retorna um array de endereços de interfaces de rede.

Um novo parâmetro **options** foi adicionado aos seguintes itens:

- `icmpping`
- `icmppingloss`
- `icmppingsec`

Este parâmetro pode ser usado para especificar se as respostas redirecionadas devem ser tratadas como host de destino ativo ou host de destino inativo. Veja verificações simples para mais detalhes.

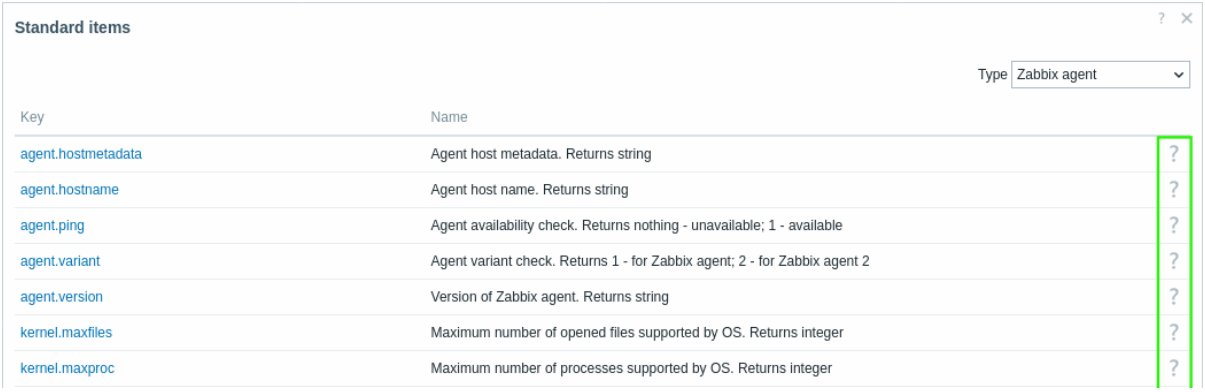
Registro de IDs de Engine SNMPv3 duplicados

Os IDs de Engine no SNMPv3 são usados como identificadores exclusivos do dispositivo. Às vezes, os IDs de Engine são os mesmos em vários dispositivos devido a uma configuração incorreta ou configurações de fábrica. Como os padrões SNMP exigem que os IDs de Engine sejam exclusivos, os items que compartilham o mesmo ID de Engine tornam-se não suportados no Zabbix, levando a problemas de disponibilidade com esses dispositivos.

Para ajudar a solucionar esses problemas, as informações sobre dispositivos SNMPv3 que compartilham o mesmo ID de Engine agora serão registradas periodicamente pelo Zabbix server. Observe que a detecção de IDs de Engine duplicados funciona em cada SNMP poller separadamente.

Link de documentação para cada item padrão

Cada item padrão agora possui um link direto do frontend para sua página de documentação.



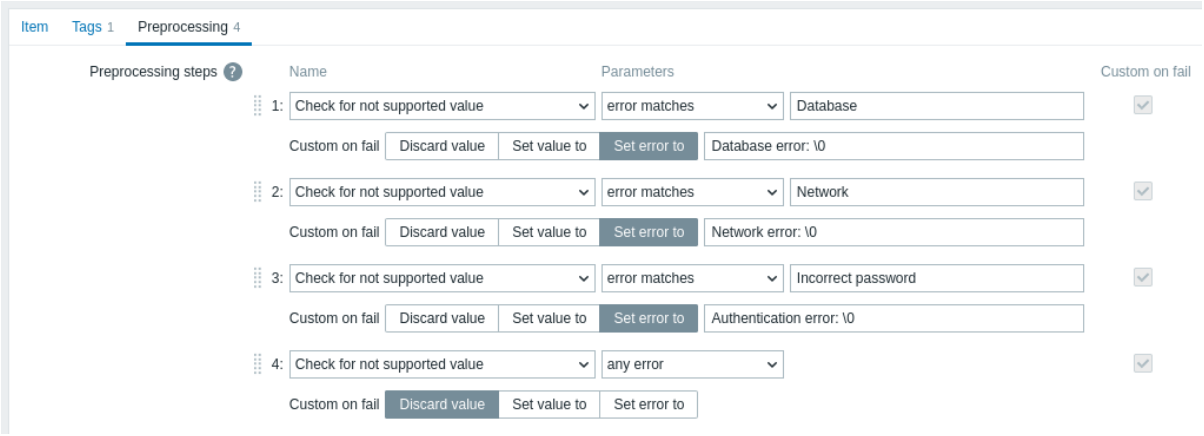
Os links são colocados sob o ícone de ponto de interrogação, ao abrir a janela de ajuda do item a partir do formulário de configuração do item (clique em *Selecionar* ao lado do campo de chave do item).

Pré-processamento Manipulação avançada de causa raiz para estado de item não suportado

O tratamento de erros em caso de falha ao recuperar o valor de um item (e, portanto, ele se tornar não suportado) anteriormente não permitia distinguir o motivo ou o estágio de execução em que o processo falhou. Todos os erros tinham que ser tratados usando uma única opção de tratamento de erro - descartar o valor, definir um valor especificado ou definir uma mensagem de erro especificada.

Agora é possível comparar a mensagem de erro com uma expressão regular. Se o erro corresponder (ou não corresponder), é possível especificar como o caso de erro deve ser processado. Por exemplo, uma mensagem de erro específica pode ser ”mapeada” para um caso mais geral para ser correspondida e tratada por uma etapa de pré-processamento adicional, ou algum problema intermitente (por exemplo, conectividade de rede) pode ser tratado de forma diferente de uma falha definitiva ao adquirir o valor do item.

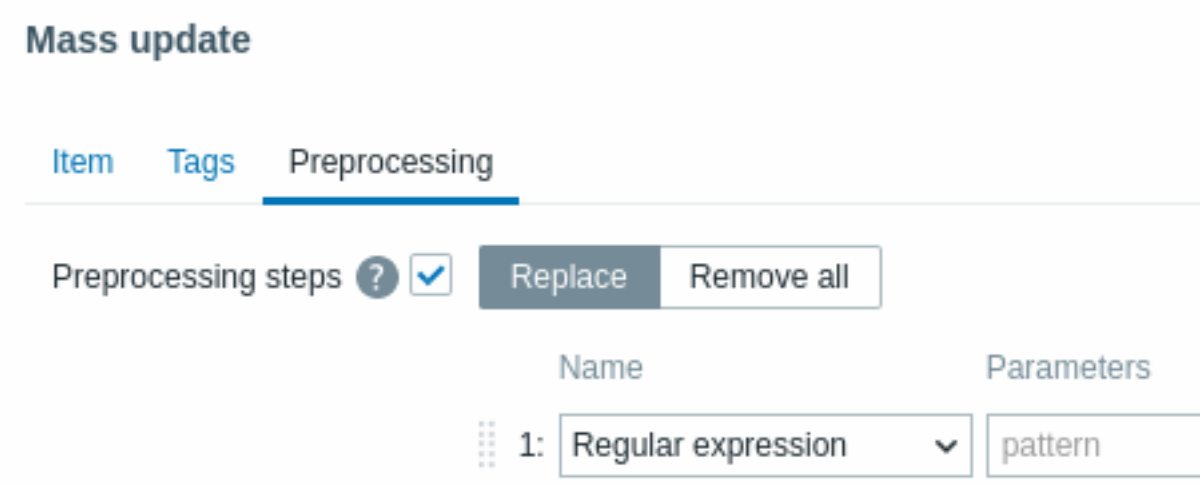
Agora é possível adicionar várias etapas de pré-processamento *Verificar valor não suportado*. Observe que só pode haver uma etapa de correspondência ”qualquer erro” no final do pipeline, verificando o estado não suportado do item. Se presente, ela é ativada se nenhuma das verificações específicas corresponder (ou não corresponder) ao padrão correspondente, ou se uma mensagem de erro (modificada) foi mantida - ou seja, nenhuma substituição de ”Descartar valor” ou ”Definir valor como” entrou em vigor.



Veja também: Verificar valor não suportado

Melhor usabilidade para atualização em massa de etapas de pré-processamento

O design anterior do formulário de atualização em massa de item não era suficientemente claro se a atualização da etapa de pré-processamento adicionaria ou substituiria as etapas de pré-processamento. No novo design, os botões de opção *Substituir* e *Remover tudo* foram adicionados, tornando claro para os usuários o que esperar como resultado da atualização em massa das etapas de pré-processamento:



Macros Macros de usuário suportadas em nomes de item e protótipo de item

Macros de usuário agora são suportadas em nomes de item e nomes de protótipo de item.

Observe que o suporte a macros de usuário foi removido dos nomes de item/protótipo de item no Zabbix 6.0. Agora ele foi restaurado. Também é possível pesquisar pelo nome do item com as macros resolvidas, o que anteriormente não era suportado.

O nome do item com macros resolvidas é armazenado em uma tabela separada no banco de dados (`item_rtname`), que é uma extensão da tabela de itens. Para cada registro na tabela de itens, um registro correspondente em `item_rtname` é criado (exceto para protótipos de item, itens de regras de descoberta e itens de template). O nome com macros resolvidas é limitado a 2048 caracteres.

O nome do item com macros resolvidas é exibido em todos os locais do frontend, exceto na seção *Coleta de dados*.

Um novo processo do server chamado `configuration syncer worker` foi adicionado, responsável por resolver e sincronizar os valores das macros de usuário nos nomes dos itens.

Suporte estendido para funções de macro

Funções de macro agora são suportadas com todos os tipos de macros:

- Macros internas
- Macros de usuário
- Macros de descoberta de baixo nível
- Macros de expressão

Funções de macro podem ser usadas em todos os locais que suportam as macros listadas. Isso se aplica, a menos que seja explicitamente declarado que apenas uma macro é esperada (por exemplo, ao configurar macros de host ou filtros de regras de descoberta de baixo nível).

Relatórios agendados A funcionalidade de relatórios agendados não é mais experimental.

Relatórios de várias páginas

Para dashboards com várias páginas, os relatórios agora são retornados com todas as páginas do dashboard, sendo que cada página do PDF corresponde a uma página do dashboard. Anteriormente, essa funcionalidade era limitada a retornar apenas a primeira página do dashboard.

Notificações Comandos remotos em agents ativos

Agora é possível executar comandos remotos em um agent da versão 7.0 que está operando no modo ativo. Assim que a execução de um comando remoto for acionada por uma operação de ação ou execução manual de um script, o comando será incluído na configuração de verificações ativas e executado assim que o agent ativo recebê-lo.

Para scripts que são executados apenas em um agent ativo, é provável que o timeout padrão do server ou proxy precise ser aumentado. O timeout do server/proxy deve ser maior que a frequência de atualização da verificação ativa, caso contrário, o timeout do server/proxy pode ser excedido antes mesmo que o agent ativo receba o script.

Observe que agents ativos mais antigos ignorarão quaisquer comandos remotos incluídos na configuração de verificações ativas. Para mais informações, consulte *Verificações passivas e ativas do agent*.

Suporte ao processamento de tags para eventos internos

O processamento de tags retornadas pelo script webhook agora também é suportado para eventos internos.

Além disso, os macros {EVENT.TAGS.<tag name>}, {EVENT.TAGS}, {EVENT.TAGSJSON}, {EVENT.RECOVERY.TAGS}, {EVENT.RECOVERY.TAGSJS agora são suportados para notificações de eventos internos.

Essas alterações permitem o uso de webhooks para atualizar ou fechar um chamado externo/suporte por meio de uma notificação de recuperação de evento interno.

Bancos de dados Auditlog convertido em hypertable no TimescaleDB

A tabela `auditlog` foi convertida em hypertable no TimescaleDB em novas instalações para se beneficiar do particionamento automático por tempo (7 dias por padrão) e melhor desempenho.

Para atualizar instalações existentes com sucesso, consulte Atualizando o schema do TimescaleDB.

Veja também: Versões do TimescaleDB suportadas

Tabela de banco de dados separada para proxies

Os registros de proxy foram movidos da tabela `hosts` e agora são armazenados na nova tabela `proxy`.

Além disso, os dados operacionais dos proxies (como último acesso, versão, compatibilidade) foram movidos da tabela `host_rtdata` e agora são armazenados na nova tabela `proxy_rtdata`.

Processos Multithreading

Várias alterações foram feitas como parte da transição para uma arquitetura multithreaded:

- Um novo parâmetro de configuração foi adicionado: `--with-stacksize`. Este parâmetro permite substituir o tamanho padrão da pilha de threads usado pelo sistema (em kilobytes).
- A resolução de macros de usuário foi movida do gerenciador de pré-processamento para os workers de pré-processamento.

Reforço do ambiente do servidor

Agora é possível restringir algumas funções do Zabbix para reforçar a segurança do ambiente do servidor:

- a execução de scripts globais no Zabbix server pode ser desabilitada configurando EnableGlobalScripts=0 na configuração do server. Para novas instalações, a execução de scripts globais no Zabbix server é desabilitada por padrão.
- a autenticação HTTP do usuário pode ser desabilitada configurando `$ALLOW_HTTP_AUTH=false` no arquivo de configuração do frontend (zabbix.conf.php).
- o modem GSM para notificações SMS agora pode ser especificado no novo parâmetro SMSDevices, limitando a possibilidade de configurar incorretamente o caminho do modem GSM a partir do frontend.

Validação do arquivo de configuração

A possibilidade de validação do arquivo de configuração foi adicionada aos comandos de manutenção do Zabbix server, proxy, agent, agent 2 e web service. A validação pode ser feita usando a opção *-T --test-config*. Em caso de validação bem-sucedida, o código de saída será "0"; caso contrário, o componente sairá com um código de saída diferente de zero e uma mensagem de erro correspondente. Avisos (por exemplo, no caso de um parâmetro obsoleto) não afetarão o código de saída bem-sucedido.

Detectando recursos da biblioteca cURL em tempo de execução

Anteriormente, os recursos da biblioteca cURL eram detectados no momento da compilação do Zabbix server, proxy ou agent. Se os recursos do cURL fossem atualizados, para utilizá-los, o respectivo componente do Zabbix precisava ser recompilado.

Agora, apenas uma reinicialização é necessária para que os recursos atualizados da biblioteca cURL fiquem disponíveis no Zabbix. A recompilação não é mais necessária. Isso é válido para o Zabbix server, proxy ou agent.

Veja também as notas de atualização.

Configuração do Agent 2

Tamanho do buffer

O valor padrão do parâmetro de configuração BufferSize para o agent 2 do Zabbix foi aumentado de 100 para 1000.

Valores vazios permitidos

Agora são permitidos valores vazios em parâmetros de configuração relacionados a plugins no agent 2 do Zabbix.

Definindo o tipo de inicialização do serviço do agent do Windows

A opção para definir o tipo de inicialização do serviço do agent/agent 2 do Zabbix no Windows (`-S --startup-type`) foi adicionada. Esta opção permite configurar o serviço do agent/agent 2 para iniciar automaticamente na inicialização do Windows (`automatic`), após os serviços iniciados automaticamente terem concluído a inicialização (`delayed`), quando iniciado manualmente por um usuário ou aplicativo (`manual`) ou para desabilitar o serviço completamente (`disabled`).

Ao realizar a instalação do agent do Windows a partir do MSI, o tipo de inicialização padrão no Windows Server 2008/Vista e versões posteriores agora é `delayed` se não for especificado de outra forma no parâmetro de linha de comando `STARTUPTYPE` parâmetro de linha de comando. Isso melhora a confiabilidade e o desempenho do serviço do agent/agent 2 do Zabbix no Windows, especialmente durante reinicializações do sistema.

Suporte ao antigo tipo numérico removido

O antigo estilo de valores de ponto flutuante, anteriormente obsoleto, não é mais suportado, pois valores numéricos de faixa estendida são usados.

Parâmetro de prefixo do Vault adicionado aos arquivos de configuração

Os arquivos de configuração *zabbix_server.conf* e *zabbix_proxy.conf* foram complementados com um novo parâmetro opcional `Vault Prefix`; *zabbix.conf.php* foi complementado com o parâmetro opcional `$DB['VAULT_PREFIX']`, e *setup.php* foi atualizado de acordo.

Os caminhos do vault para CyberArk e HashiCorp não são mais codificados, permitindo implantações do vault com caminhos não padrão.

Descoberta Concorrência na descoberta de rede

Anteriormente, cada regra de descoberta de rede era processada por um processo discoverer. Assim, todas as verificações de serviço dentro da regra só podiam ser realizadas sequencialmente.

Na nova versão, o processo de descoberta de rede foi reformulado para permitir a concorrência entre as verificações de serviço. Um novo processo discovery manager foi adicionado juntamente com um número configurável de discovery workers (ou threads).

O discovery manager processa as regras de descoberta e cria um discovery job para cada regra com tarefas (verificações de serviço). As verificações de serviço são coletadas e realizadas pelos discovery workers. Apenas aquelas verificações que possuem o mesmo IP e porta são agendadas sequencialmente, pois alguns dispositivos podem não permitir conexões simultâneas na mesma porta.

Um novo item interno `zabbix[discovery_queue]` permite monitorar o número de verificações de descoberta na fila.

O parâmetro StartDiscoverers agora determina o número total de discovery workers disponíveis para descoberta. O número padrão de StartDiscoverers foi aumentado de 1 para 5, e o intervalo de 0-250 para 0-1000. Os processos `discoverer` das versões anteriores do Zabbix foram removidos.

Adicionalmente:

- Todas as verificações de serviço agora são realizadas assíncronamente, exceto as verificações LDAP;
- O número de verificações assíncronas simultâneas por cada tipo de verificação de serviço (ou o número de workers disponíveis para todas as verificações de serviço síncronas) agora é configurável no frontend (veja *Máximo de verificações simultâneas por tipo*). Este parâmetro é opcional.
- A verificação de serviço HTTP anteriormente era igual à verificação TCP. Agora, a verificação HTTP/HTTPs é feita via libcurl. Se o Zabbix server/proxy for compilado sem libcurl, as verificações HTTP funcionarão como anteriormente (ou seja, como verificações TCP), mas as verificações HTTPS não funcionarão.
- Erros no processo de descoberta de rede agora serão exibidos no frontend (em *Coleta de dados -> Descoberta*), por exemplo:
  - erros de fping;

- OID SNMP incorreto;
- macro incorreta para timeout do item;
- erros de intervalo de endereço.

Adicionando tags de host durante descoberta/autoregistro

Operações adicionais agora estão disponíveis para eventos de descoberta e autoregistro:

- Adicionar tags de host
- Remover tags de host

Compartilhamento de grupos de hosts descobertos

As regras de descoberta de baixo nível agora podem vincular grupos de hosts já descobertos e existentes aos hosts criados pelas mesmas regras de descoberta de baixo nível. Isso afeta os grupos de hosts descobertos e criados anteriormente por outras regras de descoberta de baixo nível com base nos protótipos de grupo especificados.

Conectores A funcionalidade de data streaming não é mais experimental.

Transmitir dados seletivos e configurar intervalos de tentativa

Ao transmitir valores de item do Zabbix para sistemas externos, agora você pode configurar quais valores de item o conector deve transmitir com base em seu tipo de informação (numérico (sem sinal), numérico (float), caractere, etc.).

Além disso, para evitar tentativas malsucedidas de transmitir valores de item ou eventos (por exemplo, se o endpoint HTTP estiver ocupado ou com limite de taxa), agora você também pode configurar o intervalo de tentativa - quanto tempo o conector deve esperar após uma tentativa malsucedida de transmitir dados.

Os códigos de resposta HTTP 201, 202, 203 e 204 agora também são aceitos pelos conectores como sucesso (anteriormente apenas 200).

Transmitir dados para o Apache Kafka

Uma nova ferramenta para transmitir dados para sistemas externos - o conector Kafka para o Zabbix server - já está disponível. O conector Kafka é um servidor leve escrito em Go, projetado para encaminhar valores de item e eventos de um Zabbix server para um broker Kafka.

Templates Para novos templates e alterações em templates existentes, consulte Alterações em templates.

Frontend Autenticação multifator

A autenticação multifator (MFA) com senha única baseada em tempo (TOTP) ou método de autenticação Duo Universal Prompt agora pode ser usada para fazer login no Zabbix, fornecendo uma camada adicional de segurança além do nome de usuário e senha.

Formato de hora dos EUA

As exibições de hora e data no frontend agora estão em conformidade com o padrão de exibição de hora/data dos EUA quando o idioma padrão (en_US) do frontend é usado.

Antes	Agora
14:43:26	02:43:26 PM

Clonagem simplificada

Anteriormente, era possível *Clonar* e *Clonar completamente* hosts, templates e mapas.

Agora, a opção *Clonar* foi removida e a opção *Clonar completamente* foi renomeada para *Clonar*, mantendo toda a funcionalidade anterior de *Clonar completamente*.

Ícones substituídos por fontes

Todos os ícones no frontend foram trocados de imagens para fontes.

Formulários modais

Vários formulários do frontend agora são abertos em janelas modais (pop-up):

- Configuração de regra de descoberta de rede;
- Configuração de script global;
- Configuração de correlação de eventos;

- Configuração de módulo;
- Configuração de tipo de mídia;
- Configuração de template;
- Configuração de trigger e protótipo de trigger;
- Configuração de item e protótipo de item.

Configuração avançada recolhível

As caixas de seleção *Configuração avançada*, responsáveis por exibir opções de configuração avançadas, foram substituídas por blocos recolhíveis (consulte, por exemplo, Configuração do conector, Configuração do serviço, Configuração do widget *Clock*, etc.). Isso melhora a experiência do usuário, pois recolher esses blocos e salvar a configuração não redefinirá mais as opções avançadas configuradas para seus valores padrão.

Seção de menu aprimorada para triggers principais

A seção de menu para visualizar os principais triggers agora se chama *Top 100 triggers*. A possibilidade de filtrar triggers por nome do problema e tags foi adicionada. Além disso, o número de problemas detectados em vez do número de alterações de status agora é exibido para cada trigger.

Limite de caracteres aumentado para campos de configuração

Campos de URL

O limite de caracteres para todos os campos de URL agora é de 2048 caracteres. Isso agora inclui: *URL do tile* para configurações relacionadas a mapas geográficos, *URL do frontend* para configuração de parâmetros diversos do frontend, *URLs* para mapas de rede e elementos de mapas de rede, *URL A-C* para campos de inventário de host, e *URL* para o widget de dashboard de URL.

Campos de autenticação

O limite de caracteres para os campos de autenticação *Usuário/Nome de usuário* e *Senha* agora é de 255 caracteres. Isso se aplica à configuração de autenticação HTTP para itens do tipo HTTP agent, cenários web e conectores, bem como à configuração de autenticação para verificações simples, monitoramento ODBC, verificações SSH, verificações Telnet, e monitoramento JMX.

Truncamento de resultados de teste de item e pré-processamento

Ao testar items ou testar etapas de pré-processamento, os valores recuperados de um host e os resultados dos testes agora são truncados para um tamanho máximo de 512KB quando enviados para o frontend. Observe que dados maiores que 512KB ainda são processados integralmente pelo Zabbix server.

Abas dos painéis de host

Todos os painéis de host configurados para o host selecionado agora são exibidos como abas sob o cabeçalho da página de painéis de host, substituindo o menu suspenso anterior no canto superior direito. Isso permite uma transição fácil entre vários painéis de host e melhora a navegação pelos dados de monitoramento.

Log de auditoria

Em *Administração* → *Log de auditoria*, agora você pode ativar/desativar o registro de auditoria das atividades de descoberta de baixo nível, descoberta de rede e autorregistro realizadas pelo server (usuário do sistema).

O período padrão de armazenamento dos registros de log de auditoria antes de serem excluídos pelo housekeeper foi alterado de 365 dias para 31 dias.

Filtro de dados mais recentes

Em *Monitoramento* → *Dados mais recentes*, o subfiltro e os dados não são mais exibidos por padrão se o filtro não estiver definido.

Se estiver atualizando de versões anteriores do Zabbix, consulte também: Notas de atualização para 7.0.0.

Versão mínima do PHP necessária

A versão mínima do PHP necessária foi aumentada de 7.4.0 para 8.0.0.

Elementos renomeados

- Alguns parâmetros de widget de dashboard com o rótulo *Tags* foram renomeados para maior clareza: *Tags do item* (para o widget *Visão geral dos dados*), *Tags do cenário* (para o widget *Monitoramento web*); *Tags do problema* (para os widgets *Gráfico*, *Hosts com problemas*, *Problemas*, *Problemas por severidade* e *Visão geral dos triggers*);
- O link de ação para edição do conteúdo do mapa, disponível na lista de mapas na seção *Monitoramento* → *Mapas*, foi renomeado de *Construtor* para *Editar*;
- Os campos para definir os períodos de armazenamento de histórico e tendências nos formulários de configuração de item e protótipo de item foram renomeados;
- No formulário de configuração do widget *Hosts principais*, os campos *Coluna de ordenação* e *Contagem de hosts* foram renomeados para *Ordenar por* e *Limite de hosts* para descrever melhor suas funções.

- Na configuração do widget *Gráfico*, o campo da legenda *Exibir min/máx/média* foi renomeado para *Exibir min/média/máx*, e os campos do conjunto de dados *padrão do host* e *padrão do item* foram renomeados para *padrões do host* e *padrões do item*.
- Nas configurações do Perfil do usuário, a aba *Mensagens* foi renomeada para *Notificações do frontend*, na qual a opção *Mensagens do frontend* também foi renomeada para *Notificações do frontend*.

Diversos

- Os ícones do menu principal foram atualizados;
- As mensagens que indicam a ausência de dados ou filtros não definidos (em widgets ou filtros pop-up sem dados para exibir) foram atualizadas. Além disso, o rodapé "Exibindo 0 de 0 encontrados" foi removido nos casos em que não há dados para exibir ou quando o filtro (ou o uso da pesquisa global) não retorna resultados.
- Os **números de versão** do frontend e do server do Zabbix agora podem ser visualizados na página de informações do sistema;
- Todas as ações em que o tipo de mídia é usado agora são exibidas na lista de tipos de mídia (coluna *Usado em ações*). Anteriormente, as ações em que a opção *Enviar apenas para* opção na configuração da operação de ação era "Todos" não eram incluídas na coluna *Usado em ações* do tipo de mídia;
- Uma nova opção de filtro foi adicionada à seção Últimos dados: agora é possível filtrar os items pelo seu estado (suportado/não suportado);
- Uma nova opção de filtro *Status de reconhecimento* foi adicionada à seção Problemas: agora é possível filtrar os problemas pelo seu estado (não reconhecido/reconhecido/reconhecido por mim);
- O botão padrão de fechar janela foi adicionado às janelas pop-up destinadas à configuração e atualização em massa dos elementos e formas do mapa;
- A configuração das permissões do grupo de usuários e das tags para filtrar os problemas visualizáveis foi aprimorada. Agora é possível selecionar vários grupos de host/template de uma só vez para atribuir as mesmas permissões a eles.
- Adiar notificações globais em um navegador agora irá adiá-las em todos os navegadores/dispositivos onde o usuário estiver logado.
- O parâmetro *Substituir host* no widget *Valor do item* foi movido antes da seção *Configuração avançada* para melhor usabilidade.

Plugins Ember+

Um novo plugin para monitoramento direto de Ember+ pelo Zabbix agent 2 foi adicionado.

Para mais informações, consulte:

- Leia-me do plugin Ember+
- Itens do agent 2
- Parâmetros do plugin Ember+
- Instalação do agent 2

Instalação Pacotes de instalação separados para derivados do RHEL

Pacotes de instalação dedicados estão disponíveis para as versões 8 e 9 do AlmaLinux, CentOS Stream, Oracle Linux e Rocky Linux. Anteriormente, pacotes de instalação únicos eram fornecidos para o RHEL e distribuições baseadas no RHEL. Agora, pacotes separados são usados para o RHEL e cada um de seus derivados mencionados acima para evitar possíveis problemas de incompatibilidade binária.

Suporte para ARM64/AArch64

Os pacotes de instalação ARM64/AArch64 agora estão disponíveis para Debian, RHEL 8, 9 e seus derivados, bem como SLES/OpenSUSE Leap 15.

6 O que há de novo no Zabbix 7.0.1

Veja as alterações incompatíveis para esta versão.

Items Monitoramento do log de eventos do VMware

O item vmware.eventlog agora também retorna informações do usuário.

Desempenho do serviço DNS

O item net.dns.perf agora retorna um tempo de resposta em vez de 0 quando o servidor DNS responde com um código de erro (por exemplo, `NXDOMAIN` ou `SERVFAIL`).

Pré-processamento JavaScript mais seguro

A função JavaScript atob agora retorna um array de inteiros sem sinal de 8 bits em vez de uma string decodificada.

Frontend Opções visuais do gráfico de pizza

Com o widget *Pie chart* configurado para exibir um gráfico de rosca, agora é possível configurar o parâmetro *Stroke width* - a largura da borda do setor do gráfico.

Além disso, ao passar o mouse sobre um setor, ele agora se expande para fora para um efeito visual mais suave, substituindo o comportamento anterior de destacar o setor.

Opções de fonte de dados para Gráfico de pizza e Gráfico

Os conjuntos de dados do tipo ”Lista de items” para os widgets *Gráfico* e *Gráfico de pizza* agora suportam a seleção de widgets compatíveis como fonte de dados para items.

Templates Para novos templates e alterações em templates existentes, consulte Alterações de templates.

Bancos de dados Suporte ao MySQL 8.4

A versão máxima suportada para o MySQL agora é 8.4.X.

Suporte ao MariaDB 11.4

A versão suportada máxima para o MariaDB agora é 11.4.X.

Suporte ao TimescaleDB 2.15

A versão máxima suportada para o TimescaleDB agora é 2.15.X.

7 Novidades do Zabbix 7.0.2

Veja as alterações incompatíveis para esta versão.

Bancos de dados Suporte ao MySQL 9.0

A versão máxima suportada para o MySQL agora é 9.0.X.

Histórico de dados binários convertido para hypertable no TimescaleDB

A tabela `history_bin` foi convertida para hypertable no TimescaleDB para se beneficiar do particionamento automático por tempo (1 dia por padrão) e melhor desempenho.

Para atualizar com sucesso instalações existentes, consulte Atualizando o schema do TimescaleDB.

Veja também: Versões suportadas do TimescaleDB

Macros Suporte a macro

A macro {EVENT.UPDATE.STATUS} agora é suportada em notificações e comandos de atualização de serviço.

Templates Para novos templates e alterações em templates existentes, consulte Alterações em templates.

Plugins Um repositório de exemplo foi criado e pode ser usado como ponto de partida para criar um plugin customizado carregável com o Zabbix agent 2.

Para mais informações, veja:

- Plugin de exemplo para o Zabbix agent 2
- Central do desenvolvedor

8 Novidades do Zabbix 7.0.3

Veja as alterações incompatíveis para esta versão.

Items Descoberta de grupo de proxy

Um **zabbix[proxy group,discovery]** item interno foi adicionado, retornando uma lista de grupos de proxy com dados de configuração e dados em tempo real.

Caracteres especiais suportados na senha do usuário da verificação ODBC

Agora, caracteres especiais são suportados ao especificar a senha do usuário da verificação ODBC no frontend.

9 Novidades do Zabbix 7.0.4

Veja as alterações incompatíveis para esta versão.

Requisitos libevent 2.0.10 necessário

A versão mínima necessária do libevent foi aumentada para 2.0.10.

Macros Funções de macro

Novas funções de macro foram adicionadas:

Funções	Descrição
regrepl	Substitui um caractere ou substring em um valor de macro. *Nota* que esta função só é suportada com a biblioteca **libpcre2**. Se o Zabbix server/proxy foi compilado com `libpcre`, esta função retornará UNKNOWN.
tr	Translitera caracteres em um valor de macro.
btoa	Codifica o valor da macro para o formato Base64.
htmlencode,htmldecode	Codifica/decodifica o valor da macro para/de formato codificado em HTML.
urlencode,urldecode	Codifica/decodifica o valor da macro para/de formato codificado em URL.
uppercase,lowercase	Transforma o texto para maiúsculas ou minúsculas.

Para mais informações, consulte funções de macro.

Items Nova tentativa para itens SNMP bulk

Uma nova tentativa foi adicionada para itens SNMP bulk para evitar falhas caso um dos pacotes seja perdido.

O timeout para itens SNMP com `get` e `walk` é definido para toda a sessão. Se o timeout for atingido, uma nova tentativa ocorrerá uma vez, o timeout será redefinido e a última solicitação será reenviada, permitindo continuar a sessão a partir da última solicitação caso um único pacote seja perdido ou chegue tarde demais.

Seleção de frame em items de navegador

Items de navegador agora suportam um novo método browser.switchFrame() para alternar para um frame específico especificado nos parâmetros do método.

Cenários web Variáveis de cenário web aprimoradas

As variáveis em cenários web foram aprimoradas com a adição de:

- suporte a funções de macro, por exemplo, `{newVar}={{myVar}.btoa()}`
- suporte a JSONPath, por exemplo, `{url}=jsonpath:$.host_url`
- suporte a XML Xpath, por exemplo, `{status}=xmlxpath://host/response/status`

Bancos de dados Suporte ao TimescaleDB 2.16

A versão máxima suportada para o TimescaleDB agora é 2.16.X.

Suporte ao MariaDB 11.5

A versão suportada máxima para o MariaDB agora é 11.5.X.

Notificações Integrações de webhook

Uma nova integração está disponível que permite integrar o Zabbix com o recurso de fluxo de trabalho do MS Teams usando o tipo de mídia webhook do Zabbix.

Templates Para novos templates e alterações em templates existentes, consulte Alterações em templates.

Frontend Tecla TAB em scripts

Pressionar a tecla TAB no editor modal de JavaScript agora insere um caractere de tabulação (equivalente a 4 espaços) em vez de mover o foco para outro elemento. Isso se aplica ao parâmetro *Script* em itens Browser, itens Script, Webhooks e à etapa de pré-processamento JavaScript.

Acesso rápido à configuração de host ou template

Nas subseções *Coleta de dados* > *Hosts* e *Coleta de dados* > *Templates* (como *Itens* ou *Regras de descoberta*), agora você pode clicar no nome do host ou template para abrir seu formulário de configuração.

Além disso, o menu de item agora inclui a opção *Host* ou *Template*, permitindo que você acesse e configure rapidamente o host ou template ao qual o item pertence.

Filtrar o log de auditoria por IP

A opção IP foi adicionada ao filtro do log de auditoria. Agora você pode filtrar o log de auditoria pelo IP de onde a atividade foi iniciada.

Além disso, os IPs na lista agora são clicáveis. Clicar no hiperlink filtrará os registros pelo respectivo IP.

Ações numeradas onde o tipo de mídia/script é usado

As ações onde um tipo de mídia/script é usado agora são precedidas por um número indicando o total de tais ações. Veja Tipos de mídia/Scripts.

Criptografia Opções de cifra TLS no Zabbix agent 2

Opções de cifra TLS foram adicionadas à configuração do Zabbix agent 2 e do agent 2 para Windows.

Opção	Zabbix agent 2	Agent 2 para Windows
TLSCipherCert	sim	-
TLSCipherCert13	sim	sim
TLSCipherPSK	sim	-
TLSCipherPSK13	sim	sim
TLSCipherAll	sim	-
TLSCipherAll13	sim	sim

10 Novidades do Zabbix 7.0.5

Veja as alterações incompatíveis para esta versão.

Item interno para monitoramento do gerenciador de alta disponibilidade

O processo `ha manager` (gerenciador de alta disponibilidade) agora pode ser monitorado pelo item interno zabbix[process*].

Renovação automática de token HashiCorp

Para segredos do vault do HashiCorp, a renovação automática de tokens de serviço e tokens de serviço periódicos agora é suportada. Se o Zabbix detectar que um token é renovável, ele continuará renovando-o automaticamente até que o TTL máximo do token seja atingido ou, no caso de tokens de serviço periódicos, indefinidamente.

Seleção de item binário para exportação

A seleção de item binário agora é suportada ao configurar um conector.

Templates

Para novos templates e alterações em templates existentes, consulte Alterações em templates.

11 Novidades do Zabbix 7.0.6

Suporte ao TimescaleDB 2.17

A versão máxima suportada para o TimescaleDB agora é 2.17.X.

Suporte ao PostgreSQL 17

O PostgreSQL 17 agora é suportado.

Cálculo do estado do poller assíncrono

Os pollers assíncronos agora são considerados ocupados se atingirem o limite definido pelo parâmetro de configuração MaxConcurrentChecksPerPoller do server/proxy. A utilização dos processos de poller pode ser monitorada com o item interno zabbix[process,,].

StartPreprocessors padrão aumentado

O valor padrão do parâmetro de configuração StartPreprocessors do server/proxy foi aumentado de 3 para 16. Para aproveitar ao máximo os recursos disponíveis, o número de instâncias pré-iniciadas de threads de trabalho de pré-processamento deve ser definido para pelo menos corresponder ao número de núcleos de CPU disponíveis. Se as tarefas de pré-processamento não forem limitadas pela CPU e envolverem solicitações de rede frequentes, é recomendada a configuração de trabalhadores adicionais.

Nomes de instância MSSQL suportados no plugin agent 2

Agora é possível especificar nomes de instância MSSQL como parte da string de conexão em items do plugin MSSQL agent 2, por exemplo:

```
mssql.custom.query[sqlserver://localhost/InstanceName,...]
```

Observe que, neste caso, **nenhuma porta** deve ser especificada. Se a porta for especificada, o nome da instância será ignorado e a porta especificada será usada para conexão.

Consulta de contador de desempenho atualizada no plugin MSSQL agent 2

A consulta de contador de desempenho do plugin MSSQL agent 2 foi atualizada, resolvendo problemas na recuperação de dados de instâncias MSSQL sem nome (padrão).

Número máximo de linhas em widgets

O número máximo de linhas que podem ser exibidas foi aumentado de 100 para **1000** em vários widgets do dashboard:

- Log de ações
- Histórico de item
- Problemas
- Principais hosts
- Principais triggers

1000 agora é o valor do parâmetro ZBX_MAX_WIDGET_LINES nas definições do frontend.

Templates

Para novos templates e alterações em templates existentes, consulte Alterações em templates.

12 Novidades do Zabbix 7.0.7

Filtrar o uso do tipo de mídia em ações

A opção *Exibir ações* foi adicionada ao filtro de tipos de mídia. Agora você pode usar o filtro para exibir ações na coluna *Usado em ações* com base no escopo de uso do tipo de mídia (definido pelo parâmetro *Enviar para o tipo de mídia* nos detalhes da operação da ação).

Além disso, o parâmetro *Enviar apenas para* nos detalhes da operação da ação foi renomeado para *Enviar para o tipo de mídia*, e sua opção *Todos* foi renomeada para *Todos disponíveis*.

Transmissão automática da primeira entidade nos widgets

Os widgets que suportam a seleção de entidades (por exemplo, *Navegador de item*) agora transmitem automaticamente os dados da primeira entidade disponível, sem a necessidade de selecionar a entidade primeiro. Para mais detalhes, consulte Comportamento do widget.

Identificando items com a maior fila de tarefas de pré-processamento

Uma propriedade top.peak foi adicionada à saída do diaginfo, permitindo identificar os items com a maior fila de tarefas de pré-processamento.

Templates

Todos os templates foram atualizados para reintroduzir os prefixos de nome de trigger que foram removidos no Zabbix 7.0.0.

Para outras alterações em templates existentes e informações sobre novos templates, consulte Alterações em templates.

13 O que há de novo no Zabbix 7.0.8

Templates

Para alterações em templates existentes e informações sobre novos templates, consulte Alterações em templates.

14 Novidades do Zabbix 7.0.9

Veja as alterações incompatíveis para esta versão.

Configurações de compressão de histórico, tendências e log de auditoria

O bloco *Compressão de histórico e tendências* foi renomeado para *Compressão de histórico, tendências e log de auditoria* na seção Administração > Housekeeping do frontend do Zabbix, acessível após instalar ou atualizar o TimescaleDB.

15 Novidades no Zabbix 7.0.10

Veja as mudanças incompatíveis para esta versão.

Suporte ao PHP

A versão máxima suportada para o PHP agora é a 8.4.

Suporte ao TimescaleDB 2.18

A versão máxima suportada para o TimescaleDB agora é 2.18.X.

Cálculo do status de disponibilidade do host

O status de disponibilidade de todas as interfaces de host de um único tipo (Agent, SNMP, IPMI, JMX) agora é determinado apenas por aquelas interfaces que são usadas por pelo menos um item habilitado. Não é mais determinado por todas as interfaces do mesmo tipo, independentemente de qualquer item usá-las.

Coluna de informações na lista de ações

Uma coluna *Info* foi adicionada à lista de ações. Se tudo estiver funcionando corretamente, nada é exibido nesta coluna. Em caso de erros, como operações de ação ausentes ou condições após upgrade/descoberta de baixo nível, um ícone de aviso é exibido. Você pode passar o mouse sobre o ícone para ver uma dica de ferramenta com a descrição do erro.

Além disso, as ações não estão mais sendo desabilitadas:

- se um usuário excluir objetos usados em condições ou operações de ação;
- se objetos usados em condições ou operações de ação não forem mais descobertos pela descoberta de baixo nível e tiverem sido removidos (após o período *Excluir recursos perdidos*).

Agora, ao tentar excluir objetos que são usados em condições ou operações de ação, o usuário receberá uma mensagem de erro correspondente. Anteriormente, os objetos eram simplesmente excluídos sem qualquer aviso e a ação era desabilitada, o que não era totalmente transparente.

Parâmetro de usuário adicionado ao plugin Redis do Zabbix agent 2

No arquivo de configuração do plugin Redis do Zabbix agent 2, o parâmetro `User` agora permite especificar um nome de usuário para autenticação com servidores Redis que usam Listas de Controle de Acesso (ACLs), o que fornece uma alternativa ao uso de uma senha global de administrador.

Templates

Para alterações em templates existentes e informações sobre novos templates, consulte Alterações em templates.

16 Novidades do Zabbix 7.0.11

Resolução de DNS assíncrona

O agent poller e o SNMP poller agora podem ser configurados para usar a biblioteca *c-ares* (versão 1.16.0 ou mais recente) em vez do resolvedor de DNS *libevent*. Isso permite uma resolução de DNS potencialmente mais rápida e flexível.

Para habilitar o c-ares, o Zabbix deve ser configurado com a opção `--with-ares`.

Observe que, a partir do c-ares 1.31.0, o cache de DNS é ativado por padrão.

Integrações de Webhook atualizadas

Os seguintes webhooks prontos para uso agora suportam todos os tipos de eventos e foram aprimorados para eficiência de código, manutenção e funcionalidade, com alterações refletidas na documentação:

- Discord - script atualizado para usar a API Discord v10; formato de mensagem personalizada codificado para eventos baseados em trigger removido.
- GitHub - parâmetro de versão da API do GitHub introduzido para usar uma versão estática em vez da mais recente; em casos de escalonamento, problemas existentes agora são atualizados em vez de novos serem criados; rótulos de problemas e prefixos ajustáveis nos parâmetros do webhook adicionados.
- GLPI - modelos de mensagens atualizados.
- Jira - ações de trigger e serviço agora suportam o mapeamento das severidades de eventos do Zabbix para as prioridades de tickets do Jira (via parâmetros `severity_information`, `severity_high`, etc.); suporte para prioridades personalizadas do Jira adicionado a todas as ações de autorregistro, descoberta e internas (via parâmetros `jira_priority_discovery`, `jira_priority_autoregistration`, `jira_priority_internal`); campos personalizados do Jira para criação de tickets adicionados a todas as ações (anteriormente apenas trigger); mensagens de eventos de problema agora são enviadas para a descrição do ticket do Jira, e mensagens padrão de eventos OK agora são adicionadas como comentários.
- Jira Service Management (anteriormente Jira Service Desk) - ações de trigger e serviço agora suportam o mapeamento das severidades de eventos do Zabbix para as prioridades de tickets do Jira; suporte para prioridades personalizadas do Jira adicionado a todas as ações de autorregistro, descoberta e internas; campos personalizados do Jira para criação de tickets adicionados a todas as ações (anteriormente apenas trigger); suporte adicionado para campos personalizados de data e hora, botão de opção, caixa de seleção, escolha única e múltipla; comportamento do campo personalizado de URL ajustado; suporte para rótulos de tickets adicionado e formato do nome do problema atualizado para corresponder ao tipo de mídia Jira; opções de comentário privado (padrão) e público agora disponíveis para escalonamentos; em casos em que a severidade do evento é atualizada entre escalonamentos, a prioridade da tarefa não retorna mais ao original.
- MS Teams - modelos de mensagens atualizados.
- OTRS CE - suporte adicionado para fechamento de incidentes de problemas resolvidos.
- PagerDuty - suporte adicionado para fechamento de incidentes de problemas resolvidos; token da API movido para as configurações de mídia do usuário via `{ALERT.SENDTO}`.
- Slack - parâmetro obsoleto `slack_as_user` removido; modelos implementados para mensagens enviadas ao Slack.
- Telegram - suporte para mensagens de resposta adicionado.
- Zammad - grupo de usuários anteriormente codificado “Users” movido para um parâmetro do webhook; tags de eventos não são mais criadas por padrão.

Plugin SMART para o Zabbix agent 2 atualizado

A chave de item `smart.disk.get` no plugin SMART do Zabbix agent 2 agora retorna valores brutos e normalizados para os atributos SMART.

Melhorias de desempenho

O desempenho da atualização de tendências no TimescaleDB foi aumentado usando upserts em vez de updates.

O desempenho do Zabbix server foi aprimorado implementando a sincronização paralela de tendências e evitando consultas de intervalo, resultando em tempos de desligamento mais rápidos no MySQL, PostgreSQL e TimescaleDB.

Templates

Para alterações em templates existentes e informações sobre novos templates, consulte Alterações em templates.

Layout aprimorado para a opção “Personalizado em caso de falha” no pré-processamento de item

As opções ”Personalizado em caso de falha” da aba pré-processamento de valor do item agora são exibidas corretamente em todos os idiomas suportados. Anteriormente, o desalinhamento em algumas traduções levava a ambiguidades na configuração ao definir ações personalizadas de tratamento de erros para etapas de pré-processamento. Com esta correção, o controle é renderizado de forma consistente e, após selecionar a caixa de seleção ”Personalizado em caso de falha”, as opções disponíveis

agora são apresentadas como uma lista suspensa em vez de botões de opção — proporcionando uma interface mais clara e compacta.

17 Novidades do Zabbix 7.0.12

Item interno de taxa de transferência de pré-processamento

Um novo `zabbix[preprocessing]` item interno foi adicionado que retorna um JSON contendo estatísticas dos valores recebidos pelo gerenciador de pré-processamento:

- *queued* - o número e o tamanho dos valores em fila que requerem pré-processamento (contador);
- *direct* - o número e o tamanho dos valores em fila que não requerem pré-processamento (contador);
- *queue* - a contagem de valores enfileirados na fila de pré-processamento (igual a `zabbix[preprocessing_queue]`)

Precisão aprimorada no tempo de consulta do net.dns.get

O item `net.dns.get` agora relata o atributo `query_time` com precisão de microssegundos, aumentando de duas para seis casas decimais. Para mais detalhes sobre os valores de retorno, consulte a página Valores de retorno para net.dns.get.

Plugin MSSQL para o agent 2 do Zabbix atualizado

No plugin MSSQL para o agent 2 do Zabbix, o item `mssql.last.backup.get` foi atualizado para excluir bancos de dados inexistentes da consulta.

Comportamento de `ListenPort` no endpoint `/status` do Zabbix agent 2

Quando as verificações passivas estão desabilitadas na configuração do Zabbix agent 2, o parâmetro `ListenPort` não possui mais um valor.

Templates

Para alterações em templates existentes e informações sobre novos templates, consulte Alterações em templates.

18 Novidades no Zabbix 7.0.13

Suporte ao TimescaleDB 2.19

A versão suportada máxima para o TimescaleDB agora é 2.19.X.

Plugin PostgreSQL para o Zabbix agent 2 atualizado

O plugin PostgreSQL para o Zabbix agent 2 foi atualizado para compatibilidade com o PostgreSQL 17 e agora utiliza a nova visão de sistema `pg_stat_checkpointer` para estatísticas de checkpoint (anteriormente `pg_stat_bgwriter`).

Descoberta de datastore VMware

O item vmware.datastore.discovery[] agora suporta filtragem por ID exclusivo de hipervisor ou máquina virtual.

Resolução de macro secreta

Os valores das macros de usuário secretas agora permanecerão mascarados nos locais que suportam a resolução de macro secreta se forem referenciados indiretamente. Por exemplo, as macros internas {ITEM.KEY}, {ITEM.KEY<1-9>}, {LLDRULE.KEY} usadas em tipos de mídia (parâmetros de Script ou Webhook) serão resolvidas para chaves de item contendo macros secretas mascaradas, como `net.tcp.port[******,******]` em vez de `net.tcp.port[192.0.2.0,80]`.

Templates

Para alterações em templates existentes e informações sobre novos templates, consulte Alterações em templates.

19 Novidades no Zabbix 7.0.14

SNMP walk e get com tentativas aumentadas

SNMP agent `walk` e `get` agora fazem até 5 tentativas (em vez de 1) para recuperar todos os valores se as tentativas anteriores expirarem ou falharem. Cada tentativa redefine o tempo limite e retoma a partir da última solicitação bem-sucedida, melhorando a confiabilidade dos dados em caso de perda de pacotes ou respostas atrasadas durante a coleta de dados SNMP.

Considere definir um tempo limite baixo no formulário de configuração do item para evitar longos atrasos se o dispositivo estiver inacessível (por exemplo, um tempo limite de 3 segundos pode resultar em um tempo de espera de 15 segundos).

Informações de diagnóstico de pré-processamento do server/proxy

Seções de principais items classificadas pelo número e tamanho do valor recebido foram adicionadas à seção de pré-processamento das informações de diagnóstico em tempo de execução do server/proxy (acessadas executando o comando de tempo de execução `diaginfo`).

Integrações de Webhook atualizadas

Os seguintes webhooks prontos para uso foram atualizados, com alterações refletidas na documentação:

- OTRS CE - parâmetro de tipo de ticket e arquivo de serviço web ausente adicionados.
- Telegram - suporte para tópicos de supergrupo adicionado.

Plugin do Oracle Database para o Zabbix agent 2 atualizado

O plugin do Oracle Database para o Zabbix agent 2 agora suporta tanto o nome do container quanto o ID do container no parâmetro `conname` da chave `oracle.ts.stats`.

Templates

Para alterações em templates existentes e informações sobre novos templates, consulte Alterações em templates.

Widgets Tamanho dinâmico da unidade no widget Clock

O tamanho da unidade de informação do relógio digital (data, hora, fuso horário) do widget Clock agora se ajusta dinamicamente para preencher todo o espaço disponível com base nas unidades exibidas, no tamanho do widget e na escala de exibição. Os parâmetros de configuração para definir o tamanho da unidade foram removidos.

20 Novidades do Zabbix 7.0.15

Esta versão menor não possui alterações funcionais.

21 O que há de novo no Zabbix 7.0.16

Esta versão menor não possui alterações funcionais.

22 Novidades do Zabbix 7.0.17

Suporte ao TimescaleDB 2.20

A versão máxima suportada para o TimescaleDB agora é 2.20.X.

Informações de diagnóstico de pré-processamento do server/proxy

O diagnóstico de pré-processamento foi aprimorado destacando os principais IDs de item com os maiores tempos de processamento. Essas informações podem ser acessadas executando o server ou proxy do Zabbix com a opção de tempo de execução diaginfo, por exemplo, `zabbix_server -R diaginfo=preprocessing`.

Ignorar o gerenciador de pré-processamento para items sem pré-processamento

Para reduzir a carga de processamento, os valores de item de items que não possuem pré-processamento nem items dependentes são adicionados diretamente ao cache de histórico ou enviados ao gerenciador LLD (consulte Detalhes do pré-processamento). Anteriormente, todos os valores de item passavam pelo gerenciador de pré-processamento, independentemente da configuração de pré-processamento.

Integração do webhook GLPI atualizada

O tipo de mídia GLPI agora suporta autenticação usando tokens de aplicativo.

Templates

Para alterações em templates existentes e informações sobre novos templates, consulte Alterações em templates.

23 Novidades do Zabbix 7.0.18

Suporte ao TimescaleDB 2.21

A versão máxima suportada para o TimescaleDB agora é 2.21.X.

Recarregamento automático das credenciais SNMPv3

As verificações de item SNMP agent walk[] e get[] agora recarregam automaticamente as configurações de autenticação e privacidade SNMPv3 atualizadas quando a interface SNMPv3 correspondente é modificada. O comando manual de recarregamento de cache -R snmp_cache_reload permanece disponível para solução de problemas e para dispositivos que não estão totalmente em conformidade com as especificações SNMPv3.

Suporte adicional de função para o plugin Oracle

Os items do plugin Oracle para o Zabbix agent 2 agora suportam funções adicionais, como SysBACKUP, SysDG, SysKM ou SysRAC nas opções de login.

smart.disk.discovery — novo parâmetro **type**

O item smart.disk.discovery (plugin S.M.A.R.T. do Zabbix agent 2) agora aceita um parâmetro **type** opcional para especificar um valor a ser pesquisado nos discos.

24 Novidades do Zabbix 7.0.19

Suporte ao nome TNS para o plugin Oracle

O suporte nativo ao nome TNS do Oracle foi adicionado ao plugin Oracle no Zabbix agent 2. A chave TNS ou o valor TNS podem ser especificados no nível dos parâmetros da chave do item ou na configuração do plugin (para sessão nomeada ou parâmetros padrão).

O suporte ao nome TNS permite a descrição da conexão em um ambiente de banco de dados em cluster com um único host Zabbix. Para mais informações, consulte o readme do plugin Oracle.

Aumento do tempo limite máximo para o zabbix_get e zabbix_js

O valor máximo para o parâmetro `timeout` das utilitários de linha de comando zabbix_get e zabbix_js foi aumentado para 600 segundos.

Suporte ao MariaDB 12.0

A versão suportada máxima para o MariaDB agora é 12.0.X.

Templates

Para alterações em templates existentes e informações sobre novos templates, consulte Alterações em templates.

Destacar toda a linha na visualização de Problemas e no widget de Problemas

A opção *Destacar toda a linha* agora está disponível na visualização não compacta de Monitoramento > Problemas e no widget de painel Problemas.

25 Novidades do Zabbix 7.0.20

Atenção! Não é recomendado atualizar para esta versão devido a erros relacionados a gráficos e ao plugin MySQL do Zabbix agent 2.

Suporte ao PostgreSQL 18

A versão máxima suportada para o PostgreSQL agora é 18.X.

Suporte ao TimescaleDB 2.22

A versão máxima suportada para o TimescaleDB agora é 2.22.X.

Plugin Redis — Suporte a TLS e validação em tempo de inicialização

O suporte a TLS foi adicionado ao plugin Redis para o Zabbix agent 2.

A validação da configuração TLS do plugin em tempo de inicialização foi implementada e as mensagens de validação/erro foram aprimoradas. Uma lógica de configuração inválida (por exemplo: usar o tipo de conexão `verify_full` sem especificar `TLSCAFile`) pode impedir o Zabbix agent 2 de iniciar.

Templates

Para alterações em templates existentes e informações sobre novos templates, consulte Alterações em templates.

26 Novidades do Zabbix 7.0.21

Esta versão menor não possui alterações funcionais.

27 Novidades do Zabbix 7.0.22

Veja as alterações incompatíveis para esta versão.

Suporte ao TimescaleDB 2.23

A versão suportada máxima para o TimescaleDB agora é 2.23.X.

Suporte ao MySQL 9.5

A versão máxima suportada para o MySQL agora é 9.5.X.

Parâmetros de linha de comando expandidos para pacotes de instalação MSI do agent

Os pacotes de instalação MSI do agent Zabbix (tanto para o agent Zabbix quanto para o agent Zabbix 2) agora suportam os seguintes parâmetros ao realizar uma instalação a partir da linha de comando:

- `HOSTNAMEITEM` - especifica o item do agent Zabbix usado para obter o nome do host
- `UNSAFEUSERPARAMETERS` - permite que todos os caracteres (por exemplo, ' " ` * ?) sejam passados como argumentos para parâmetros definidos pelo usuário
- `STARTAGENTS` - especifica o número de instâncias `zabbix_agentd` pré-forked que processam verificações passivas
- `DONOTSTART` - impede que o instalador MSI inicie o serviço do agent Zabbix
- `SOURCEIP` - especifica o endereço IP de origem para conexões de saída para o server/proxy Zabbix, ou para conexões feitas ao executar determinados items (por exemplo, `web.page.get`, `net.tcp.port`)

Tempo limite máximo para itens de ping ICMP

Um tempo limite máximo possível de 600 segundos agora é aplicado aos itens icmpping*.

Templates

Para alterações em templates existentes e informações sobre novos templates, consulte Alterações em templates.

28 Novidades do Zabbix 7.0.23

Suporte ao TimescaleDB 2.24

A versão máxima suportada para o TimescaleDB agora é 2.24.X.

29 Novidades do Zabbix 7.0.24

Suporte ao TimescaleDB 2.25

A versão máxima suportada para o TimescaleDB agora é 2.25.X.

30 Novidades no Zabbix 7.0.25

Integrações de webhook

- Uma nova integração permite vincular o Zabbix ao IBM Maximo Service Request usando o tipo de mídia webhook do Zabbix.
- O tipo de mídia GLPI webhook foi atualizado para suportar a API RESTful v2 com autorização OAuth2 (habilitada por padrão), suporte opcional à API REST legada v1 via flag de configuração, criação de tickets para eventos de problema e mapeamento de urgência configurável pelo usuário.

Suporte ao MySQL 9.6

A versão suportada máxima para o MySQL agora é 9.6.X.

Suporte ao TimescaleDB 2.26

A versão máxima suportada para o TimescaleDB agora é 2.26.X.

Suporte ao PHP 8.5

A versão máxima suportada para o PHP agora é 8.5.

Suporte ao MariaDB 12.2

A versão suportada máxima para o MariaDB agora é 12.2.X.

2. Definições

Visual geral Nesta seção você pode aprender o significado de alguns termos comumente usados no Zabbix.

Definições ***host***

- qualquer dispositivo físico ou virtual, aplicação, serviço ou qualquer outra coleção logicamente relacionada de parâmetros monitorados.

grupo de hosts

- um agrupamento lógico de hosts. Os grupos de hosts são usados ao atribuir direitos de acesso a hosts para diferentes grupos de usuários.

item

- um dado específico que você deseja receber de um host, uma métrica de dados.

pré-processamento de valor

- uma transformação do valor da métrica recebida antes de salvá-la no banco de dados.

trigger

- uma expressão lógica que define um limite de problema e é usada para "avaliar" os dados recebidos nos items.

Quando os dados recebidos estão acima do limite, os triggers passam de 'Ok' para um estado de 'Problema'. Quando os dados recebidos estão abaixo do limite, os triggers permanecem/retornam ao estado 'Ok'.

template

- um conjunto de entidades (items, triggers, gráficos, regras de descoberta de baixo nível, cenários web) prontos para serem aplicados a um ou vários hosts.

O objetivo dos templates é acelerar a implantação de tarefas de monitoramento em um host; também para facilitar a aplicação de alterações em massa nas tarefas de monitoramento. Os templates são vinculados diretamente a hosts individuais.

grupo de templates

- um agrupamento lógico de templates. Os grupos de templates são usados ao atribuir direitos de acesso a templates para diferentes grupos de usuários.

evento

- uma ocorrência única de algo que merece atenção, como um trigger mudando de estado ou um registro automático de descoberta/agent ocorrendo.

tag de evento

- *um marcador pré-definido para o evento.* Pode ser usado em correlação de eventos, granularidade de permissões, etc.

correlação de eventos

- *um método de correlacionar problemas à sua resolução de forma flexível e precisa.*

Por exemplo, você pode definir que um problema relatado por um trigger pode ser resolvido por outro trigger, que pode até usar um método de coleta de dados diferente.

problema

- *um trigger que está no estado "Problema".*

atualização de problema

- *opções de gerenciamento de problemas fornecidas pelo Zabbix, como adicionar comentário, reconhecer, alterar a gravidade ou fechar manualmente.*

ação

- *um meio predefinido de reagir a um evento.*

Uma ação consiste em operações (por exemplo, enviar uma notificação) e condições (*quando* a operação é executada)

escalação

- *um cenário personalizado para executar operações dentro de uma ação; uma sequência de envio de notificações/execução de comandos remotos.*

macro

- *um espaço reservado para variável que é resolvido para um valor específico dependendo do contexto.*

mídia

- *um meio de entrega de notificações; canal de entrega.*

notificação

- *uma mensagem sobre algum evento enviada a um usuário pelo canal de mídia escolhido.*

comando remoto

- *um comando predefinido que é executado automaticamente em um host monitorado sob alguma condição.*

cenário web

- *uma ou várias requisições HTTP para verificar a disponibilidade de um site.*

frontend

- *a interface web fornecida com o Zabbix.*

dashboard

- *seção personalizável da interface web que exibe resumos e visualizações* de informações importantes em unidades visuais chamadas widgets.

widget

- *unidade visual que exibe informações de determinado tipo e origem* (um resumo, um mapa, um gráfico, o relógio, etc.), usada no dashboard.

Zabbix API

- *A Zabbix API permite usar o protocolo JSON RPC para criar, atualizar e buscar objetos do Zabbix (como hosts, items, gráficos e outros) ou executar quaisquer outras tarefas personalizadas.*

Zabbix server

- *um processo central do software Zabbix que realiza o monitoramento, interage com proxies e agents do Zabbix, calcula triggers, envia notificações; um repositório central de dados.*

Zabbix proxy

- *um processo que pode coletar dados em nome do Zabbix server, tirando parte da carga de processamento do server.*

Zabbix agent

- um processo implantado em alvos de monitoramento para monitorar ativamente recursos e aplicações locais.

Zabbix agent 2

- uma nova geração de Zabbix agent para monitorar ativamente recursos e aplicações locais, permitindo o uso de plugins personalizados para monitoramento.

> **Attention:**
> Como o Zabbix agent 2 compartilha muitas funcionalidades com o Zabbix agent, o termo "Zabbix agent" na documentação se refere a ambos - Zabbix agent e Zabbix agent 2, se o comportamento funcional for o mesmo. O Zabbix agent 2 só é nomeado especificamente quando sua funcionalidade é diferente.

criptografia

- suporte a comunicações criptografadas entre os componentes do Zabbix (server, proxy, agent, utilitários zabbix_sender e zabbix_get) usando o protocolo Transport Layer Security (TLS).

autoregistro de agent

- processo automatizado pelo qual um Zabbix agent é registrado como um host e começa a ser monitorado.

descoberta de rede

- descoberta automatizada de dispositivos de rede.

descoberta de baixo nível

- descoberta automatizada de entidades de baixo nível em um dispositivo específico (por exemplo, sistemas de arquivos, interfaces de rede, etc).

regra de descoberta de baixo nível

- conjunto de definições para descoberta automatizada de entidades de baixo nível em um dispositivo.

protótipo de item

- uma métrica com determinados parâmetros como variáveis, pronta para descoberta de baixo nível. Após a descoberta de baixo nível, as variáveis são automaticamente substituídas pelos parâmetros reais descobertos e a métrica automaticamente começa a coletar dados.

protótipo de trigger

- um trigger com determinados parâmetros como variáveis, pronto para descoberta de baixo nível. Após a descoberta de baixo nível, as variáveis são automaticamente substituídas pelos parâmetros reais descobertos e o trigger automaticamente começa a avaliar os dados.

Protótipos de algumas outras entidades do Zabbix também são usados na descoberta de baixo nível - protótipos de gráficos, protótipos de hosts, protótipos de grupos de hosts.

3 Processos do Zabbix

Por favor, use a barra lateral para acessar o conteúdo na seção de processos do Zabbix.

1 Server

Visão geral

O Zabbix server é o processo central do software Zabbix.

O server realiza a coleta e o recebimento de dados, calcula triggers e envia notificações aos usuários. É o componente central para o qual os agents e proxies do Zabbix reportam dados sobre a disponibilidade e integridade dos sistemas. O server pode, ele mesmo, verificar remotamente serviços em rede (como servidores web e servidores de e-mail) usando verificações simples de serviço.

O server é o repositório central no qual todas as informações de configuração, dados estatísticos e operacionais são armazenados, e é a entidade no Zabbix que irá alertar ativamente os administradores quando surgirem problemas em qualquer um dos sistemas monitorados.

O funcionamento de um Zabbix server básico é dividido em três componentes distintos: são eles: Zabbix server, web frontend e armazenamento de banco de dados.

Todas as informações de configuração do Zabbix são armazenadas no banco de dados, com o qual tanto o server quanto o frontend web interagem. Por exemplo, quando você cria um novo item usando o frontend web (ou API), ele é adicionado à tabela de items no banco de dados. Então, cerca de uma vez por minuto, o Zabbix server consulta a tabela de items para obter uma lista dos items que estão ativos, que é então armazenada em um cache dentro do Zabbix server. É por isso que pode levar até dois minutos para que quaisquer alterações feitas no frontend do Zabbix apareçam na seção de dados mais recentes.

Executando o server

Se instalado como pacote

O Zabbix server é executado como um processo daemon. O server pode ser iniciado executando:

```
systemctl start zabbix-server
```

Isso funcionará na maioria dos sistemas GNU/Linux. Em outros sistemas, pode ser necessário executar:

```
/etc/init.d/zabbix-server start
```

Da mesma forma, para parar/reiniciar/ver o status, use os seguintes comandos:

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

Iniciar manualmente

Se o acima não funcionar, você terá que iniciá-lo manualmente. Encontre o caminho para o binário zabbix_server e execute:

```
zabbix_server
```

Você pode usar os seguintes parâmetros de linha de comando com o Zabbix server:

```
-c --config <file>              caminho para o arquivo de configuração (o padrão é /usr/local/etc/zabbix_s
-f --foreground                 executa o Zabbix server em primeiro plano
-R --runtime-control <option>   executa funções administrativas
-T --test-config                valida o arquivo de configuração e sai
-h --help                       exibe esta ajuda
-V --version                    exibe o número da versão
```

Exemplos de execução do Zabbix server com parâmetros de linha de comando:

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

Controle de tempo de execução

Opções de controle de tempo de execução:

Opção	Descrição	Alvo
config_cache_reload	Recarrega o cache de configuração. Ignorado se o cache estiver sendo carregado atualmente.	
diaginfo[=<**section**>]	Coleta informações de diagnóstico no arquivo de log do server.	**historycache** - estatísticas do cache de histórico **valuecache** - estatísticas do cache de valores **preprocessing** - estatísticas do gerenciador de pré-processamento **alerting** - estatísticas do gerenciador de alertas **lld** - estatísticas do gerenciador de LLD **locks** - lista de mutexes (fica vazia em sistemas *BSD*) **connector** - estatísticas para conectores com a maior fila
ha_status	Registra o status do cluster de alta disponibilidade (HA).	
ha_remove_node=target	Remove o nó de alta disponibilidade (HA) especificado pelo nome ou ID. Observe que nós ativos/standby não podem ser removidos.	**target** - nome ou ID do nó (pode ser obtido executando ha_status)

Opção	Descrição	Alvo
ha_set_failover_delay=delay	Defina o atraso de failover de alta disponibilidade (HA). Sufixos de tempo são suportados, por exemplo, 10s, 1m.	
proxy_config_cache_reload[=<target>]	Recarrega o cache de configuração do proxy.	**target** - lista de nomes de proxies separados por vírgula Se nenhum alvo for especificado, recarrega a configuração de todos os proxies
secrets_reload	Recarrega segredos do Vault.	
service_cache_reload	Recarrega o cache do gerenciador de serviços.	
snmp_cache_reload	Recarrega o cache SNMP — limpa as propriedades do mecanismo SNMP (engine time, engine boots, engine id, credenciais) para todos os hosts. Use para forçar uma limpeza global do cache ao solucionar problemas de SNMP.	
housekeeper_execute	Inicia o procedimento de housekeeping. Ignorado se o procedimento de housekeeping estiver em andamento.	
trigger_housekeeper_execute	Inicia o procedimento de housekeeping de trigger. Ignorado se o procedimento de housekeeping de trigger estiver em andamento. Até que o procedimento de housekeeping de trigger seja iniciado, problemas causados por triggers que foram excluídos ainda podem gerar problemas de serviço e atribuí-los a serviços. Se sua configuração envolver muitas regras de cálculo de status de serviço baseadas em triggers frequentemente descobertos/não descobertos, considere aumentar a frequência do procedimento de housekeeping ajustando o parâmetro de configuração do server `ProblemHousekeepingFrequency`.	
log_level_increase[=<target>]	Aumenta o nível de log, afeta todos os processos se o alvo não for especificado. Não suportado em sistemas *BSD*.	**tipo de processo** - Todos os processos do tipo especificado (por exemplo, poller) Veja todos os tipos de processos do server. **tipo de processo,N** - Tipo de processo e número (por exemplo, poller,3) **pid** - Identificador do processo (1 a 65535). Para valores maiores, especifique o alvo como 'tipo de processo,N'.
log_level_decrease[=<target>]	Diminui o nível de log, afeta todos os processos se o alvo não for especificado. Não suportado em sistemas *BSD*.	

Opção	Descrição	Alvo
prof_enable[=<**target**>]	Habilita o profiling. Afeta todos os processos se o alvo não for especificado. O profiling habilitado fornece detalhes de todos os rwlocks/mutexes por nome de função.	**tipo de processo** - Todos os processos do tipo especificado (por exemplo, history syncer) Tipos de processo suportados como alvos de profiling: alerter, alert manager, availability manager, configuration syncer, discovery manager, escalator, history poller, history syncer, housekeeper, http poller, icmp pinger, ipmi manager, ipmi poller, java poller, lld manager, lld worker, odbc poller, poller, preprocessing manager, preprocessing worker, proxy poller, self-monitoring, service manager, snmp trapper, task manager, timer, trapper, unreachable poller, vmware collector **tipo de processo,N** - Tipo de processo e número (por exemplo, history syncer,1) **pid** - Identificador do processo (1 a 65535). Para valores maiores, especifique o alvo como 'tipo de processo,N'. **scope** - `rwlock`, `mutex`, `processing` podem ser usados com o tipo de processo e número (por exemplo, history syncer,1,processing) ou todos os processos do tipo (por exemplo, history syncer,rwlock)
prof_disable[=<**target**>]	Desabilita o profiling. Afeta todos os processos se o alvo não for especificado.	**tipo de processo** - Todos os processos do tipo especificado (por exemplo, history syncer) Tipos de processo suportados como alvos de profiling: veja `prof_enable` **tipo de processo,N** - Tipo de processo e número (por exemplo, history syncer,1) **pid** - Identificador do processo (1 a 65535). Para valores maiores, especifique o alvo como 'tipo de processo,N'.

Exemplo de uso do controle em tempo de execução para recarregar o cache de configuração do server:

```
zabbix_server -c /usr/local/etc/zabbix_server.conf -R config_cache_reload
```

Exemplos de uso do controle em tempo de execução para recarregar a configuração do proxy:

```
### Recarregar a configuração de todos os proxies:
zabbix_server -R proxy_config_cache_reload

### Recarregar a configuração do Proxy1 e Proxy2:
zabbix_server -R proxy_config_cache_reload=Proxy1,Proxy2
```

Exemplos de uso do controle em tempo de execução para coletar informações de diagnóstico:

```
### Coletar todas as informações de diagnóstico disponíveis no arquivo de log do server:
zabbix_server -R diaginfo

### Coletar estatísticas do cache de histórico no arquivo de log do server:
zabbix_server -R diaginfo=historycache
```

Exemplo de uso do controle em tempo de execução para recarregar o cache SNMP:

```
zabbix_server -R snmp_cache_reload
```

Attention:
Quando uma interface SNMPv3 é atualizada pela interface web do Zabbix, o Zabbix recarregará automaticamente as novas credenciais SNMPv3 para essa interface na maioria dos casos; use `-R snmp_cache_reload` apenas se a coleta ainda falhar após alterações de credenciais (por exemplo, devido a inconsistências de engineBoots/engineID ou dispositivos que não seguem o RFC), ou quando você precisar forçar uma limpeza global do cache SNMP para fins de troubleshooting.

Exemplo de uso do controle em tempo de execução para acionar a execução do housekeeper:

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

Exemplos de uso do controle em tempo de execução para alterar o nível de log:

```
### Aumentar o nível de log de todos os processos:
zabbix_server -c /usr/local/etc/zabbix_server.conf -R log_level_increase

### Aumentar o nível de log do segundo processo poller:
zabbix_server -c /usr/local/etc/zabbix_server.conf -R log_level_increase=poller,2

### Aumentar o nível de log do processo com PID 1234:
zabbix_server -c /usr/local/etc/zabbix_server.conf -R log_level_increase=1234

### Diminuir o nível de log de todos os processos http poller:
zabbix_server -c /usr/local/etc/zabbix_server.conf -R log_level_decrease="http poller"
```

Exemplo de definição do atraso de failover do HA para o mínimo de 10 segundos:

```
zabbix_server -R ha_set_failover_delay=10s
```

Processo do usuário

O Zabbix server foi projetado para ser executado como um usuário não-root. Ele será executado como qualquer usuário não-root que for iniciado. Portanto, você pode executar o server como qualquer usuário não-root sem problemas.

Se você tentar executá-lo como 'root', ele mudará para o usuário 'zabbix' codificado, que deve estar presente em seu sistema. Você só pode executar o server como 'root' se modificar o parâmetro 'AllowRoot' no arquivo de configuração do server de acordo.

Se o Zabbix server e o agent forem executados na mesma máquina, é recomendável usar um usuário diferente para executar o server do que para executar o agent. Caso contrário, se ambos forem executados como o mesmo usuário, o agent poderá acessar o arquivo de configuração do server e qualquer usuário com nível de Admin no Zabbix poderá recuperar facilmente, por exemplo, a senha do banco de dados.

Arquivo de configuração

Veja as opções do arquivo de configuração para detalhes sobre a configuração do zabbix_server.

Scripts de inicialização

Os scripts são usados para iniciar/parar automaticamente os processos do Zabbix durante a inicialização/desligamento do sistema. Os scripts estão localizados no diretório misc/init.d.

Tipos de processos e threads do servidor

- `agent poller` - processo de poller assíncrono para verificações passivas com uma thread de trabalho
- `alert manager` - gerenciador da fila de alertas
- `alert syncer` - gravador de alertas no banco de dados
- `alerter` - processo para envio de notificações
- `availability manager` - processo para atualizações de disponibilidade de host
- `browser poller` - poller para verificações de item de navegador
- `configuration syncer` - processo para gerenciar o cache em memória dos dados de configuração
- `configuration syncer worker` - processo para resolver e sincronizar valores de macros de usuário em nomes de item
- `connector manager` - processo gerenciador de conectores
- `connector worker` - processo para lidar com solicitações do gerenciador de conectores
- `discovery manager` - processo gerenciador de descoberta de dispositivos
- `discovery worker` - processo para lidar com tarefas de descoberta do discovery manager
- `escalator` - processo para escalonamento de ações
- `ha manager` - processo para gerenciamento de alta disponibilidade
- `history poller` - processo para lidar com verificações calculadas que exigem conexão com o banco de dados
- `history syncer` - gravador de histórico no banco de dados
- `housekeeper` - processo para remoção de dados desatualizados (histórico e tendências de item, sessões de usuário, eventos, etc.), bem como dados deixados por objetos excluídos
- `http agent poller` - processo de poller assíncrono para verificações HTTP com uma thread de trabalho
- `http poller` - poller de monitoramento web
- `icmp pinger` - poller para verificações icmpping
- `internal poller` - poller para verificações internas
- `ipmi manager` - gerenciador de poller IPMI
- `ipmi poller` - poller para verificações IPMI

- `java poller` - poller para verificações Java
- `lld manager` - processo gerenciador de tarefas de descoberta de baixo nível
- `lld worker` - processo de trabalho de tarefas de descoberta de baixo nível
- `odbc poller` - poller para verificações ODBC
- `poller` - poller normal para verificações passivas
- `preprocessing manager` - gerenciador de tarefas de pré-processamento com threads de trabalho de pré-processamento
- `preprocessing worker` - thread para pré-processamento de dados
- `proxy poller` - poller para proxies passivos
- `proxy group manager` - gerenciador de balanceamento de carga e alta disponibilidade de proxy
- `report manager`- gerenciador de tarefas de geração de relatórios agendados
- `report writer` - processo para geração de relatórios agendados
- `self-monitoring` - processo para coleta de estatísticas internas do servidor
- `service manager` - processo para gerenciar serviços recebendo informações sobre problemas, tags de problemas e recuperação de problemas do history syncer, task manager e alert manager
- `snmp poller` - processo de poller assíncrono para verificações SNMP com uma thread de trabalho (apenas itens `walk[OID]` e `get[OID]`)
- `snmp trapper` - trapper para traps SNMP
- `task manager` - processo para execução remota de tarefas solicitadas por outros componentes (por exemplo, fechar problema, reconhecer problema, verificar valor de item agora, funcionalidade de comando remoto)
- `timer` - temporizador para processamento de manutenções
- `trapper` - trapper para verificações ativas, traps, comunicação com proxy
- `trigger housekeeper` - processo para remoção de problemas e eventos gerados por triggers que foram excluídos posteriormente
- `unreachable poller` - poller para dispositivos inacessíveis
- `vmware collector` - coletor de dados VMware responsável pela coleta de dados de serviços VMware

O arquivo de log do servidor pode ser usado para observar esses tipos de processos.

Desde o Zabbix 7.0.22, o arquivo de log do servidor é criado com permissões de leitura e gravação apenas para o proprietário do arquivo. Além disso, o arquivo é legível pelo grupo do proprietário. Todas as outras permissões são negadas.

Vários tipos de processos do servidor Zabbix podem ser monitorados usando o item interno **zabbix[process,<type>,<mode>,<state>]**.

Plataformas suportadas

Devido aos requisitos de segurança e à natureza crítica da operação do server, o UNIX é o único sistema operacional que pode fornecer consistentemente o desempenho, a tolerância a falhas e a resiliência necessários. O Zabbix opera nas versões líderes de mercado.

O server Zabbix é testado nas seguintes plataformas:

- Linux
- Solaris
- AIX
- HP-UX
- Mac OS X
- FreeBSD
- OpenBSD
- NetBSD
- SCO Open Server

Note:
O Zabbix também pode funcionar em outros sistemas operacionais semelhantes ao Unix.

Localidade

Observe que o server requer uma localidade UTF-8 para que alguns itens textuais possam ser interpretados corretamente. A maioria dos sistemas modernos semelhantes ao Unix tem uma localidade UTF-8 como padrão, no entanto, há alguns sistemas em que isso pode precisar ser definido especificamente.

1 Alta disponibilidade

Visão geral

Alta disponibilidade (HA) é normalmente exigida em infraestruturas críticas que não podem se dar ao luxo de ter praticamente nenhum tempo de inatividade. Portanto, para qualquer serviço que possa falhar, deve haver uma opção de failover disponível para assumir o controle caso o serviço atual falhe.

O Zabbix oferece uma solução de alta disponibilidade **nativa** que é fácil de configurar e não requer nenhum conhecimento prévio de HA. A HA nativa do Zabbix pode ser útil como uma camada extra de proteção contra falhas de software/hardware do Zabbix server ou para ter menos tempo de inatividade devido à manutenção.

No modo de alta disponibilidade do Zabbix, vários Zabbix servers são executados como nós em um cluster. Enquanto um Zabbix server no cluster está ativo, os outros estão em espera, prontos para assumir se necessário.

A mudança para o Zabbix HA não é comprometedora. Você pode voltar para a operação autônoma a qualquer momento.

Veja também: Detalhes da implementação

Habilitando alta disponibilidade

Iniciando o Zabbix server como nó de cluster

Dois parâmetros são necessários na configuração do server para iniciar um Zabbix server como nó de cluster:

- O parâmetro **HANodeName** deve ser especificado para cada Zabbix server que será um nó de cluster HA.

Este é um identificador de nó exclusivo (por exemplo, `zabbix-node-01`) pelo qual o server será referenciado nas configurações do agent e do proxy. Se você não especificar o HANodeName, o server será iniciado no modo standalone.

- O parâmetro **NodeAddress** deve ser especificado para cada nó.

O parâmetro NodeAddress (endereço:porta) será usado pelo frontend do Zabbix para se conectar ao nó ativo do server. O NodeAddress deve corresponder ao IP ou nome FQDN do respectivo Zabbix server.

Reinicie todos os Zabbix servers após fazer alterações nos arquivos de configuração. Eles serão agora iniciados como nós de cluster. O novo status dos servers pode ser visto em *Relatórios* → *Informações do sistema* e também executando:

```
zabbix_server -R ha_status
```

Este comando em tempo de execução registrará o status atual do cluster HA no log do Zabbix server (e no stdout):

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

Preparando o frontend

Certifique-se de que o endereço:porta do Zabbix server **não esteja definido** na configuração do frontend (encontrada em `conf/zabbix.conf.php` no diretório de arquivos do frontend).

```
// Uncomment and set to desired values to override Zabbix hostname/IP and port.
// $ZBX_SERVER                  = '';
// $ZBX_SERVER_PORT             = '';
```

O frontend do Zabbix irá detectar automaticamente o nó ativo lendo as configurações da tabela de nós no banco de dados do Zabbix. O endereço do nó ativo será usado como o endereço do Zabbix server.

Configuração do proxy

Os nós do cluster HA (servidores) devem ser listados na configuração do proxy Zabbix passivo ou ativo.

Para um proxy passivo, os nomes dos nós devem ser listados no parâmetro Server parameter do proxy, separados por uma **vírgula**.

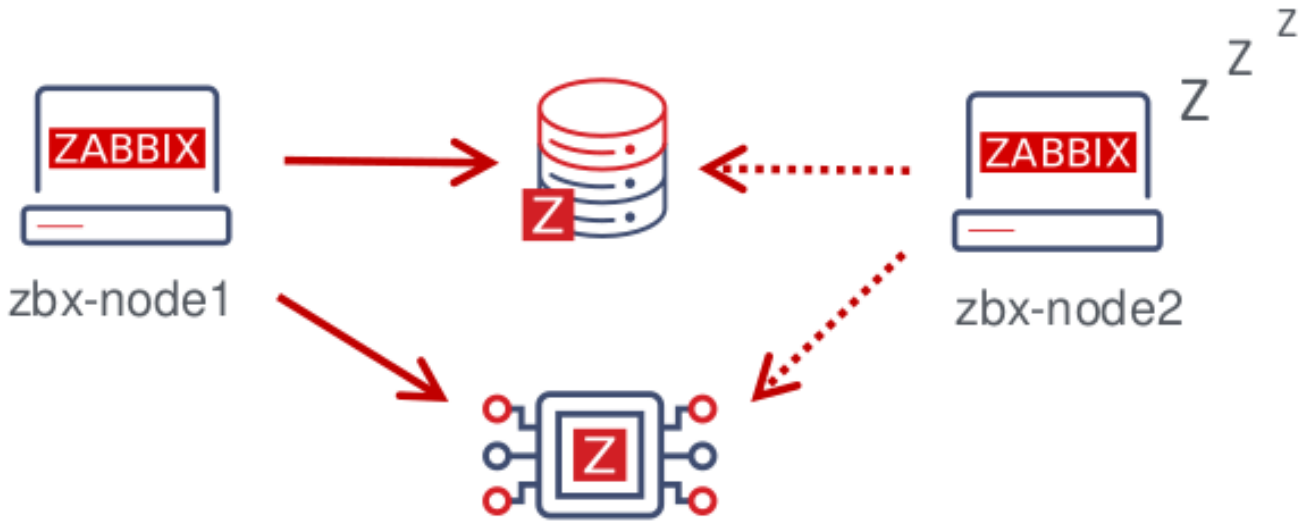
`Server=zabbix-node-01,zabbix-node-02`

Para um proxy ativo, os nomes dos nós devem ser listados no parâmetro Server parameter do proxy, separados por um **ponto e vírgula**.

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

Configuração do agent

Os nós do cluster HA (servers) devem ser listados na configuração do Zabbix agent ou Zabbix agent 2.



Para habilitar checagens passivas, os nomes dos nós devem ser listados no parâmetro Server parameter, separados por **vírgula**.

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

Para habilitar checagens ativas, os nomes dos nós devem ser listados no parâmetro ServerActive parameter. Observe que para checagens ativas os nós devem ser separados por ponto e vírgula de quaisquer outros servers, enquanto os próprios nós devem ser separados por **ponto e vírgula**, por exemplo:

`ServerActive=zabbix-node-01;zabbix-node-02`

Failover para o nó de standby

O Zabbix fará o failover para outro nó automaticamente se o nó ativo parar. Deve haver pelo menos um nó em status de standby para que o failover aconteça.

Quão rápido será o failover? Todos os nós atualizam seu último tempo de acesso (e status, se for alterado) a cada 5 segundos. Então:

- Se o nó ativo for desligado e conseguir reportar seu status como ”parado”, outro nó assumirá em **5 segundos**.
- Se o nó ativo for desligado/tornar-se indisponível sem conseguir atualizar seu status, os nós de standby esperarão pelo **failover delay** + 5 segundos para assumir

O failover delay é configurável, com o intervalo suportado entre 10 segundos e 15 minutos (um minuto por padrão). Para alterar o failover delay, você pode executar:

`zabbix_server -R ha_set_failover_delay=5m`

Gerenciando o cluster HA

O status atual do cluster HA pode ser gerenciado usando as opções dedicadas de controle em tempo de execução:

- `ha_status` - registra o status do cluster HA no log do Zabbix server (e no stdout)
- `ha_remove_node=target` - remove um nó HA identificado por seu <target> - nome ou ID do nó (o nome/ID pode ser obtido a partir da saída da execução do ha_status), por exemplo:

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

Observe que nós ativos/standby não podem ser removidos.

- `ha_set_failover_delay=delay` - define o atraso de failover do HA (entre 10 segundos e 15 minutos; sufixos de tempo são suportados, por exemplo, 10s, 1m)

O status do nó pode ser monitorado:

- em *Relatórios* → *Informações do sistema*
- no widget *Informações do sistema* do dashboard
- usando a opção de controle em tempo de execução `ha_status` do server (veja acima).

O item interno `zabbix[cluster,discovery,nodes]` pode ser usado para descoberta de nós, pois retorna um JSON com as informações do nó de alta disponibilidade.

Desabilitando o cluster HA

Para desabilitar um cluster de alta disponibilidade:

- faça cópias de backup dos arquivos de configuração
- pare os nodes em standby
- remova o parâmetro HANodeName do servidor primário ativo
- reinicie o servidor primário (ele será iniciado no modo standalone)

Atualizando o cluster HA

Para realizar uma atualização de versão principal para os nós HA:

- pare todos os nós;
- crie um backup completo do banco de dados;
- se o banco de dados usar replicação, certifique-se de que todos os nós estejam sincronizados e sem problemas. Não atualize se a replicação estiver quebrada.
- selecione um único nó que executará a atualização do banco de dados, altere sua configuração para o modo standalone comentando HANodeName e atualize ele;
- certifique-se de que a atualização do banco de dados foi totalmente concluída (*Informações do sistema* deve exibir que o Zabbix server está em execução);
- reinicie o nó em modo HA;
- atualize e inicie o restante dos nós (não é necessário alterá-los para o modo standalone, pois o banco de dados já está atualizado neste ponto).

Em uma atualização de versão secundária, basta atualizar o primeiro nó, certificar-se de que ele foi atualizado e está em execução e, em seguida, iniciar a atualização no próximo nó.

Detalhes da implementação

O cluster de alta disponibilidade (HA) é uma solução opcional e é suportado para o Zabbix server. A solução nativa de HA foi projetada para ser simples de usar, funcionará entre sites e não possui requisitos específicos para os bancos de dados que o Zabbix reconhece. Os usuários são livres para usar a solução nativa de HA do Zabbix ou uma solução de HA de terceiros, dependendo do que melhor se adapta aos requisitos de alta disponibilidade em seu ambiente.

A solução consiste em várias instâncias ou nós do zabbix_server. Cada nó:

- é configurado separadamente
- usa o mesmo banco de dados
- pode ter vários modos: ativo, standby, indisponível, parado

Apenas um nó pode estar ativo (em funcionamento) por vez. Um nó standby executa apenas um processo - o gerenciador de HA. Um nó standby não faz coleta de dados, processamento ou outras atividades regulares do server; eles não escutam em portas; possuem conexões mínimas com o banco de dados.

Tanto os nós ativos quanto os standby atualizam seu último tempo de acesso a cada 5 segundos. Cada nó standby monitora o último tempo de acesso do nó ativo. Se o último tempo de acesso do nó ativo ultrapassar o número de segundos definido em 'failover delay', o nó standby se torna o nó ativo e atribui o status 'indisponível' ao nó ativo anterior.

O nó ativo monitora sua própria conectividade com o banco de dados - se ela for perdida por mais de `failover delay`−5 segundos, ele deve parar todo o processamento e mudar para o modo standby. O nó ativo também monitora o status dos nós standby - se o último tempo de acesso de um nó standby ultrapassar o número de segundos definido em 'failover delay', o nó standby recebe o status 'indisponível'.

Os nós foram projetados para serem compatíveis entre versões menores do Zabbix.

2 Agent

Visão geral

O agent Zabbix é implantado em um alvo de monitoramento para monitorar ativamente recursos e aplicativos locais (discos rígidos, memória, estatísticas do processador, etc.).

O agent coleta informações operacionais localmente e relata os dados ao server Zabbix para processamento posterior. Em caso de falhas (como um disco rígido cheio ou um processo de serviço travado), o server Zabbix pode alertar ativamente os administradores da máquina específica que relatou a falha.

Os agents Zabbix são altamente eficientes devido ao uso de chamadas nativas do sistema para coletar informações estatísticas.

Checagens passivas e ativas

Os agents do Zabbix podem realizar checagens passivas e ativas:

- Checagens passivas - O agent do Zabbix responde a uma solicitação do server (ou proxy) do Zabbix. Por exemplo, o server solicita dados (por exemplo, carga da CPU) e o agent retorna o resultado.
- Checagens ativas - O agent do Zabbix coleta e envia dados sem esperar por uma solicitação do server (ou proxy) do Zabbix. Primeiro, ele recupera uma lista de items de monitoramento do server (carga da CPU, memória disponível, etc.), depois coleta os dados necessários e periodicamente envia novos valores de volta para ele.

O tipo de checagem do agent é configurado selecionando o respectivo tipo de item de monitoramento. O agent do Zabbix processa items do tipo ”Zabbix agent” ou ”Zabbix agent (active)”.

Plataformas suportadas

Para plataformas suportadas, consulte a página de Requisitos.

Agent em sistemas UNIX-like

O agent do Zabbix em sistemas UNIX-like é executado no host que está sendo monitorado.

Instalação

O agent do Zabbix pode ser instalado em sistemas baseados em Linux usando um dos seguintes métodos:

- Pacotes Zabbix - selecione o componente Agent (após selecionar sua versão do Zabbix, distribuição do SO e versão do SO) e siga as instruções fornecidas.
- Fontes do Zabbix - baixe os arquivos fonte e compile o agent do Zabbix configurando as fontes com a opção `--enable-agent`.

Attention:
Em geral, agents Zabbix de 32 bits funcionarão em sistemas de 64 bits, mas podem falhar em alguns casos.

Note:
Binários pré-compilados do agent Zabbix estão disponíveis para download para macOS, IBM AIX, FreeBSD, OpenBSD e Solaris. Binários legados, compatíveis com a versão atual do server/proxy Zabbix, estão disponíveis para NetBSD e HP-UX.

Se instalado como pacote

O agent do Zabbix é executado como um processo daemon. O agent pode ser iniciado executando:

```
systemctl start zabbix-agent
```

Isso funcionará na maioria dos sistemas GNU/Linux. Em outros sistemas, pode ser necessário executar:

```
/etc/init.d/zabbix-agent start
```

Para parar, reiniciar ou verificar o status do agent do Zabbix, use os seguintes comandos:

```
systemctl stop zabbix-agent
systemctl restart zabbix-agent
systemctl status zabbix-agent
```

Iniciar manualmente

Você pode iniciar o agent do Zabbix localizando o binário `zabbix_agentd` e executando-o diretamente; por exemplo:

```
zabbix_agentd
```

Agent em sistemas Windows

O Zabbix Agent no Windows é executado como um Serviço do Windows.

Instalação

O agent Zabbix pode ser instalado no Windows usando um dos seguintes métodos:

- Binários pré-compilados do agent Zabbix - baixe o pacote do instalador MSI do agent Zabbix e siga as instruções na página Instalação do agent Windows a partir do MSI.
- Fontes do Zabbix - baixe os arquivos fonte e siga as instruções na página Compilando o agent Zabbix no Windows.

Consulte a página Agent Zabbix no Microsoft Windows para obter detalhes adicionais sobre a instalação do agent Zabbix (a partir de um arquivo ZIP) como um serviço do Windows.

Opções

É possível executar várias instâncias do agent em um host. Uma única instância pode usar o arquivo de configuração padrão ou um arquivo de configuração especificado na linha de comando. No caso de múltiplas instâncias, cada instância do agent deve ter seu próprio arquivo de configuração (uma das instâncias pode usar o arquivo de configuração padrão).

Os seguintes parâmetros de linha de comando podem ser usados com o agent Zabbix:

Parâmetro	Descrição
agent UNIX e Windows	
-c --config <config-file>	Caminho para o arquivo de configuração. Você pode usar esta opção para especificar um arquivo de configuração que não seja o padrão. No UNIX, o padrão é /usr/local/etc/zabbix_agentd.conf ou conforme definido pelas variáveis de tempo de compilação ***--sysconfdir*** ou ***--prefix*** No Windows, o padrão é C:\Program Files\Zabbix Agent\zabbix_agentd.conf
-f --foreground	Executa o agent Zabbix em primeiro plano (padrão: true).
-p --print	Imprime os items conhecidos e sai. *Nota*: Para retornar também os resultados de user parameter, você deve especificar o arquivo de configuração (se não estiver no local padrão).
-t --test <item key>	Testa o item especificado e sai. *Nota*: Para retornar também os resultados de user parameter, você deve especificar o arquivo de configuração (se não estiver no local padrão).
-T --test-config	Valida o arquivo de configuração e sai.
-h --help	Exibe informações de ajuda.
-V --version	Exibe o número da versão.
agent UNIX apenas	
-R --runtime-control <option>	Executa funções administrativas. Veja runtime control.
agent Windows apenas	
-m --multiple-agents	Usa múltiplas instâncias do agent (com as opções -i, -d, -s, -x). Para distinguir os nomes dos serviços das instâncias, cada nome de serviço incluirá o valor de Hostname do arquivo de configuração especificado.
-S --startup-type <value>	Define o tipo de inicialização do serviço agent Zabbix para Windows. Valores permitidos: `automatic` - *(padrão)* inicia o serviço automaticamente na inicialização do Windows; `delayed` - atrasa a inicialização do serviço após a conclusão da inicialização dos serviços iniciados automaticamente (disponível no Windows Server 2008/Vista e versões posteriores); `manual` - inicia o serviço manualmente (por um usuário ou aplicativo); `disabled` - desativa o serviço, para que não possa ser iniciado por um usuário ou aplicativo. Você pode usar esta opção junto com a opção -i, ou separadamente para modificar o tipo de inicialização de um serviço já instalado.
-i --install	Instala o agent Zabbix para Windows como serviço.
-d --uninstall	Desinstala o serviço agent Zabbix para Windows.
-s --start	Inicia o serviço agent Zabbix para Windows.
-x --stop	Para o serviço agent Zabbix para Windows.

Exemplos específicos de uso de parâmetros de linha de comando:

- imprimir todos os items internos do agent com valores
- testar um user parameter com a chave "mysql.ping" definida no arquivo de configuração especificado
- instalar um serviço "Zabbix Agent" para Windows usando o caminho padrão para o arquivo de configuração C:\Program Files\Zabbix Agent\zabbix_agentd.conf
- instalar um serviço "Zabbix Agent [Hostname]" para Windows usando o arquivo de configuração zabbix_agentd.conf localizado na mesma pasta que o executável do agent e tornando o nome do serviço único, estendendo-o pelo valor de Hostname do arquivo de configuração

- modificar o tipo de inicialização de um serviço "Zabbix Agent" já instalado para Windows usando o arquivo de configuração zabbix_agentd.conf localizado na mesma pasta que o executável do agent

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

Controle em tempo de execução

Com as opções de controle em tempo de execução, você pode alterar o nível de log dos processos do agent.

Opção	Descrição	Alvo
log_level_increase[=<alvo>]	Aumenta o nível de log. Se o alvo não for especificado, todos os processos serão afetados.	O alvo pode ser especificado como: **tipo de processo** - todos os processos do tipo especificado (por exemplo, listener) Veja todos os tipos de processos do agent. **tipo de processo,N** - tipo de processo e número (por exemplo, listener,3) **pid** - identificador do processo (1 a 65535). Para valores maiores, especifique o alvo como 'tipo-de-processo,N'.
log_level_decrease[=<alvo>]	Diminui o nível de log. Se o alvo não for especificado, todos os processos serão afetados.	
userparameter_reload	Recarrega os valores das opções *UserParameter* e *Include* do arquivo de configuração atual.	

Exemplos:

- aumentando o nível de log de todos os processos
- aumentando o nível de log do terceiro processo listener
- aumentando o nível de log do processo com PID 1234
- diminuindo o nível de log de todos os processos de active check

```
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:
O controle em tempo de execução não é suportado no OpenBSD, NetBSD e Windows.

Tipos de processos do agent

- `active checks` - processo para realizar verificações ativas
- `collector` - processo para coleta de dados
- `listener` - processo para escutar verificações passivas

O arquivo de log do agent pode ser usado para observar esses tipos de processos.

Desde o Zabbix 7.0.22, o arquivo de log do agent é criado com permissões de leitura e gravação apenas para o proprietário do arquivo. Além disso, o arquivo é legível pelo grupo do proprietário. Todas as outras permissões são negadas.

Usuário do processo

O agent do Zabbix em UNIX foi projetado para ser executado como um usuário não-root. Ele será executado como qualquer usuário não-root que o iniciar. Portanto, você pode executar o agent como qualquer usuário não-root sem problemas.

Se você tentar executá-lo como 'root', ele mudará para o usuário 'zabbix' (pré-definido), que deve estar presente no seu sistema. Você só pode executar o agent como 'root' se modificar o parâmetro 'AllowRoot' no arquivo de configuração do agent de acordo.

Arquivo de configuração

Para obter detalhes sobre a configuração do agent Zabbix, consulte as opções do arquivo de configuração para zabbix_agentd ou agent para Windows.

Localidade

Observe que o agent requer uma localidade UTF-8 para que alguns itens textuais do agent possam retornar o conteúdo esperado. A maioria dos sistemas modernos semelhantes ao Unix possui uma localidade UTF-8 como padrão, no entanto, existem alguns sistemas onde isso pode precisar ser definido especificamente.

Código de saída

O agent retorna 0 em caso de saída bem-sucedida e 1 em caso de falha.

3 Agent 2

Visão geral

O agent 2 do Zabbix é uma nova geração do agent do Zabbix, escrito em Go (com algum código C reutilizado do agent do Zabbix). Ele foi projetado para:

- Reduzir o número de conexões TCP.
- Fornecer melhor concorrência de verificações.
- Ser facilmente extensível com plugins, que fornecem verificações simples com código mínimo e suportam verificações complexas consistindo em scripts de longa duração e coleta de dados independente com relatórios periódicos.
- Funcionar como um substituto para o agent do Zabbix, suportando todos os recursos anteriores.

Todas as métricas suportadas pelo agent 2 do Zabbix são coletadas por plugins.

Checagens passivas e ativas

O Zabbix agent 2 suporta checagens passivas e ativas, assim como o Zabbix agent. Além disso, as checagens ativas do Zabbix agent 2 suportam intervalos flexíveis/agendados e concorrência de checagens dentro de um mesmo server ativo.

Note:
Por padrão, após uma reinicialização, o Zabbix agent 2 agendará a primeira coleta de dados para checagens ativas em um momento condicionalmente aleatório dentro do intervalo de atualização do item para evitar picos no uso de recursos. Para realizar checagens ativas que não possuem *Agendamento* intervalo de atualização imediatamente após a reinicialização do agent, defina o parâmetro `ForceActiveChecksOnStart` (nível global) ou `Plugins.<Plugin name>.System.ForceActiveChecksOnStart` (afeta apenas checagens de plugins específicos) no arquivo de configuração. O parâmetro no nível do plugin, se definido, substituirá o parâmetro global.

Verificação de concorrência

As verificações de diferentes plugins podem ser executadas simultaneamente. O número de verificações simultâneas dentro de um plugin é limitado pela configuração de capacidade do plugin. Cada plugin pode ter uma configuração de capacidade codificada (1000 por padrão) que pode ser reduzida usando a configuração `Plugins.<PluginName>.System.Capacity=N` no parâmetro de configuração *Plugins*.

Plataformas suportadas

Para plataformas suportadas, consulte a página de Requisitos.

Agent 2 em sistemas UNIX-like

O agent 2 do Zabbix em sistemas UNIX-like é executado no host que está sendo monitorado.

Instalação

O agent 2 do Zabbix pode ser instalado em sistemas baseados em Linux usando um dos seguintes métodos:

- Pacotes Zabbix - selecione o componente Agent 2 (após selecionar sua versão do Zabbix, distribuição do SO e versão do SO) e siga as instruções.
- Fontes do Zabbix - baixe os arquivos fonte e compile o agent configurando com a opção `--enable-agent2`.

Note:
As capacidades de monitoramento do agent 2 do Zabbix podem ser estendidas com plugins carregáveis, que estão disponíveis separadamente. Para mais detalhes, veja Plugins carregáveis.

Attention:
Desde o Zabbix 7.0.5, os pacotes do agent 2 do Zabbix não estão mais disponíveis para RHEL 6 devido ao Go 1.22 (e mais recente) ser incompatível com o GCC 4.4.7 fornecido pelo RHEL 6, o que resulta em erros de compilação. Note, entretanto, que os pacotes do agent do Zabbix ainda são suportados.

Se instalado como pacote

O agent Zabbix 2 é executado como um processo em primeiro plano e depende de um gerenciador de serviços externo (por exemplo, systemd) para execução em segundo plano; O agent Zabbix 2 não possui suporte interno à execução como daemon no Linux.

O agent pode ser iniciado executando:

```
systemctl start zabbix-agent2
```

Para parar, reiniciar ou verificar o status do agent Zabbix 2, use os seguintes comandos:

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

Iniciar manualmente

Você pode iniciar o agent do Zabbix localizando o binário zabbix_agent2 e executando-o diretamente; por exemplo:

```
zabbix_agent2
```

Agent 2 em sistemas Windows

O agent 2 do Zabbix é executado como um processo independente; no entanto, também pode ser executado como um serviço do Windows.

Instalação

O agent Zabbix 2 pode ser instalado no Windows usando um dos seguintes métodos:

- Binários pré-compilados do agent Zabbix - baixe o pacote do instalador MSI do agent e siga as instruções na página Instalação do agent no Windows a partir do MSI.
- Fontes do Zabbix - baixe os arquivos fonte e siga as instruções na página Compilando o agent Zabbix 2 no Windows.

Note:
As capacidades de monitoramento do agent Zabbix 2 podem ser estendidas com plugins carregáveis, que estão disponíveis separadamente. Para mais detalhes, veja Plugins carregáveis.

Veja a página Agent Zabbix no Microsoft Windows para detalhes adicionais sobre a instalação do agent Zabbix 2 (a partir de um arquivo ZIP) como um serviço do Windows.

Opções

Os seguintes parâmetros de linha de comando podem ser usados com o Zabbix agent 2:

Parâmetro	Descrição
agent UNIX e Windows	
-c --config <config-file>	Caminho para o arquivo de configuração. Você pode usar esta opção para especificar um arquivo de configuração que não seja o padrão. No UNIX, o padrão é /usr/local/etc/zabbix_agent2.conf ou conforme definido pelas variáveis de tempo de compilação *--sysconfdir* ou *--prefix* No Windows, o padrão é C:\Program Files\Zabbix Agent 2\zabbix_agent2.conf
-f --foreground	Executa o Zabbix agent em primeiro plano (padrão: true).
-p --print	Imprime os items conhecidos e sai. *Nota*: Para retornar também os resultados de parâmetros de usuário, você deve especificar o arquivo de configuração (se não estiver no local padrão).
-t --test <item key>	Testa o item especificado e sai. *Nota*: Para retornar também os resultados de parâmetros de usuário, você deve especificar o arquivo de configuração (se não estiver no local padrão).
-T --test-config	Valida o arquivo de configuração e sai.
-h --help	Imprime informações de ajuda e sai.
-v --verbose	Imprime informações de depuração. Use esta opção com as opções -p e -t.

Parâmetro	Descrição
-V --version	Imprime a versão do agent e informações de licença.
-R --runtime-control <option>	Executa funções administrativas. Veja controle em tempo de execução.
Somente agent Windows	
-m --multiple-agents	Usa múltiplas instâncias do agent (com as opções -i, -d, -s, -x). Para distinguir os nomes dos serviços das instâncias, cada nome de serviço incluirá o valor de Hostname do arquivo de configuração especificado.
-S --startup-type <value>	Define o tipo de inicialização do serviço Zabbix agent para Windows. Valores permitidos: `automatic` - *(padrão)* inicia o serviço automaticamente na inicialização do Windows; `delayed` - atrasa o início do serviço após a conclusão da inicialização dos serviços iniciados automaticamente; `manual` - inicia o serviço manualmente (por um usuário ou aplicativo); `disabled` - desativa o serviço, para que não possa ser iniciado por um usuário ou aplicativo. Você pode usar esta opção junto com a opção -i, ou separadamente para modificar o tipo de inicialização de um serviço já instalado.
-i --install	Instala o Zabbix agent para Windows como serviço.
-d --uninstall	Desinstala o serviço Zabbix agent para Windows.
-s --start	Inicia o serviço Zabbix agent para Windows.
-x --stop	Para o serviço Zabbix agent para Windows.

Exemplos específicos de uso de parâmetros de linha de comando:

- imprime todos os items internos do agent com valores
- testa um parâmetro de usuário com a chave ”mysql.ping” definida no arquivo de configuração especificado
- instala um serviço ”Zabbix Agent” para Windows usando o caminho padrão para o arquivo de configuração C:\Program Files\Zabbix Agent 2\zabbix_agent2.conf
- modifica o tipo de inicialização de um serviço ”Zabbix Agent” já instalado para Windows usando o arquivo de configuração zabbix_agent2.conf localizado na mesma pasta que o executável do agent

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

Controle em tempo de execução

O controle em tempo de execução fornece algumas opções para controle remoto.

Opção	Descrição
log_level_increase	Aumenta o nível de log.
log_level_decrease	Diminui o nível de log.
metrics	Lista as métricas disponíveis.
version	Exibe a versão do agent.
userparameter_reload	Recarrega os valores das opções *UserParameter* e *Include* do arquivo de configuração atual.
help	Exibe informações de ajuda sobre o controle em tempo de execução.

Exemplos:

- aumentando o nível de log para o agent 2
- imprimindo as opções de controle em tempo de execução

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

Desde o Zabbix 7.0.17, os comandos de controle em tempo de execução do agent 2 escrevem a saída em `stdout` (saída padrão) em vez de `stderr` (erro padrão).

Arquivo de configuração

Os parâmetros de configuração do Zabbix agent 2 correspondem em sua maioria aos do Zabbix agent, com algumas exceções:

Parâmetros		Descrição
Parâmetros de configuração específicos do Zabbix agent 2		
	ControlSocket	O caminho do socket de controle em tempo de execução. O Zabbix agent 2 usa um socket de controle para comandos em tempo de execução.
	EnablePersistentBuffer, *PersistentBufferFile*, *PersistentBufferPeriod*	Esses parâmetros são usados para configurar o armazenamento persistente no Zabbix agent 2 para itens ativos.
	ForceActiveChecksOnStart	Determina se o agent deve realizar verificações ativas imediatamente após a reinicialização ou distribuí-las uniformemente ao longo do tempo.
	Plugins	Os plugins podem ter seus próprios parâmetros, no formato `Plugins.<Nome do Plugin>.<Parâmetro>=<valor>`. Um parâmetro comum de plugin é `System.Capacity`, que define o limite de verificações que podem ser executadas ao mesmo tempo.
	StatusPort	A porta em que o Zabbix agent 2 ficará ouvindo para solicitações de status HTTP e exibição de uma lista de plugins configurados e alguns parâmetros internos.
Parâmetros de configuração específicos do Zabbix agent não suportados no Zabbix agent 2		
	AllowRoot, *User*	Não suportado no Zabbix agent 2 porque ele não suporta daemonização.
	EnableRemoteCommands	Obsoleto; os parâmetros `AllowKey` e `DenyKey` são usados em ambos os agents.
	ListenBacklog	Usado no Zabbix agent para definir o número máximo de conexões pendentes na fila TCP. Não suportado no Zabbix agent 2.
	LoadModule, *LoadModulePath*	Módulos carregáveis não são suportados no Zabbix agent 2.
	MaxLinesPerSecond, *LogRemoteCommands*	No Zabbix agent 2, `Plugins.Log.MaxLinesPerSecond` e `Plugins.SystemRun.LogRemoteCommands` são usados em vez disso.

Parâmetros	Descrição
StartAgents	Este parâmetro é usado no Zabbix agent para aumentar a concorrência de verificações passivas ou desativá-las. No Zabbix agent 2, a concorrência é configurada em nível de plugin e pode ser limitada por uma configuração de capacidade. As verificações passivas são desativadas no Zabbix Agent 2 se o parâmetro `Server` não for especificado.

Para mais detalhes, consulte as opções do arquivo de configuração para o Zabbix agent2.

Códigos de saída

O agent 2 do Zabbix também pode ser compilado com versões mais antigas do OpenSSL (1.0.1, 1.0.2).

Neste caso, o Zabbix fornece mutexes para bloqueio no OpenSSL. Se um bloqueio ou desbloqueio de mutex falhar, uma mensagem de erro é impressa no fluxo de erro padrão (STDERR) e o Agent 2 é encerrado com o código de retorno 2 ou 3, respectivamente.

4 Proxy

Visão geral

O proxy Zabbix é um processo que pode coletar dados de monitoramento de um ou mais dispositivos monitorados e enviar as informações para o servidor Zabbix, funcionando essencialmente em nome do servidor. Todos os dados coletados são armazenados localmente em buffer e, em seguida, transferidos para o servidor Zabbix ao qual o proxy pertence.

Implantar um proxy é opcional, mas pode ser muito benéfico para distribuir a carga de um único servidor Zabbix. Se apenas proxies coletarem dados, o processamento no servidor se torna menos exigente em CPU e I/O de disco.

Um proxy Zabbix é a solução ideal para monitoramento centralizado de locais remotos, filiais e redes sem administradores locais.

O proxy Zabbix requer um banco de dados separado.

> **Attention:**
> Observe que os bancos de dados suportados com o proxy Zabbix são SQLite, MySQL e PostgreSQL. Usar Oracle é por sua conta e risco e pode conter algumas limitações, como, por exemplo, nos valores de retorno das regras de descoberta de baixo nível.

Veja também: Usando proxies em um ambiente distribuído

Executando o proxy

Se instalado como pacote

O proxy Zabbix é executado como um processo daemon. O proxy pode ser iniciado executando:

```
systemctl start zabbix-proxy
```

Isso funcionará na maioria dos sistemas GNU/Linux. Em outros sistemas, pode ser necessário executar:

```
/etc/init.d/zabbix-proxy start
```

Da mesma forma, para parar/reiniciar/verificar o status do proxy Zabbix, use os seguintes comandos:

```
systemctl stop zabbix-proxy
systemctl restart zabbix-proxy
systemctl status zabbix-proxy
```

Iniciar manualmente

Se o acima não funcionar, você terá que iniciá-lo manualmente. Encontre o caminho para o binário zabbix_proxy e execute:

```
zabbix_proxy
```

Você pode usar os seguintes parâmetros de linha de comando com o Zabbix proxy:

```
-c --config <file>              caminho para o arquivo de configuração
-f --foreground                 executa o Zabbix proxy em primeiro plano
-R --runtime-control <option>   executa funções administrativas
-T --test-config                valida o arquivo de configuração e sai
-h --help                       exibe esta ajuda
-V --version                    exibe o número da versão
```

Exemplos de execução do Zabbix proxy com parâmetros de linha de comando:

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

Controle em tempo de execução

Opções de controle em tempo de execução:

Opção	Descrição	Destino
config_cache_reload	Recarrega o cache de configuração. Ignorado se o cache estiver sendo carregado no momento. O proxy Zabbix ativo se conectará ao servidor Zabbix e solicitará os dados de configuração. O proxy Zabbix passivo solicitará os dados de configuração ao servidor Zabbix na próxima vez que o servidor se conectar ao proxy.	
diaginfo[=**section**]	Coleta informações de diagnóstico no arquivo de log do proxy.	**historycache** - estatísticas do cache de histórico **preprocessing** - estatísticas do gerenciador de pré-processamento **locks** - lista de mutexes (fica vazia em sistemas *BSD*)
snmp_cache_reload	Recarrega o cache SNMP — limpa as propriedades do mecanismo SNMP (engine time, engine boots, engine id, credenciais) para todos os hosts. Use para forçar uma limpeza global do cache ao solucionar problemas SNMP.	
housekeeper_execute	Inicia o procedimento de limpeza. Ignorado se o procedimento de limpeza estiver em andamento.	
log_level_increase[=**target**]	Aumenta o nível de log, afeta todos os processos se o destino não for especificado. Não suportado em sistemas *BSD*.	**tipo de processo** - Todos os processos do tipo especificado (por exemplo, poller) Veja todos os tipos de processos do proxy. **tipo de processo,N** - Tipo de processo e número (por exemplo, poller,3) **pid** - Identificador do processo (1 a 65535). Para valores maiores, especifique o destino como 'tipo de processo,N'.
log_level_decrease[=**target**]	Diminui o nível de log, afeta todos os processos se o destino não for especificado. Não suportado em sistemas *BSD*.	
prof_enable[=**target**]	Habilita o profiling. Afeta todos os processos se o destino não for especificado. O profiling habilitado fornece detalhes de todos os rwlocks/mutexes por nome de função.	**tipo de processo** - Todos os processos do tipo especificado (por exemplo, history syncer) Veja todos os tipos de processos do proxy. **tipo de processo,N** - Tipo de processo e número (por exemplo, history syncer,1) **pid** - Identificador do processo (1 a 65535). Para valores maiores, especifique o destino como 'tipo de processo,N'. **escopo** - `rwlock`, `mutex`, `processing` podem ser usados com o tipo de processo e número (por exemplo, history syncer,1,processing) ou todos os processos do tipo (por exemplo, history syncer,rwlock)

Opção	Descrição	Destino
prof_disable[=<**target**>]	Desabilita o profiling. Afeta todos os processos se o destino não for especificado.	**tipo de processo** - Todos os processos do tipo especificado (por exemplo, history syncer) Veja todos os tipos de processos do proxy. **tipo de processo,N** - Tipo de processo e número (por exemplo, history syncer,1) **pid** - Identificador do processo (1 a 65535). Para valores maiores, especifique o destino como 'tipo de processo,N'.

Exemplo de uso do controle em tempo de execução para recarregar o cache de configuração do proxy:

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

Exemplos de uso do controle em tempo de execução para coletar informações de diagnóstico:

```
### Coletar todas as informações de diagnóstico disponíveis no arquivo de log do proxy:
zabbix_proxy -R diaginfo

### Coletar estatísticas do cache de histórico no arquivo de log do proxy:
zabbix_proxy -R diaginfo=historycache
```

Exemplo de uso do controle em tempo de execução para recarregar o cache SNMP:

```
zabbix_proxy -R snmp_cache_reload
```

Attention:
Quando uma interface SNMPv3 é atualizada pela interface web do Zabbix, o Zabbix recarregará automaticamente as novas credenciais SNMPv3 para essa interface na maioria dos casos; use `-R snmp_cache_reload` apenas se a coleta ainda falhar após a alteração das credenciais (por exemplo, devido a inconsistências de engineBoots/engineID ou dispositivos que não seguem RFC), ou quando for necessário forçar uma limpeza global do cache SNMP para solução de problemas.

Exemplo de uso do controle em tempo de execução para acionar a execução do housekeeper:

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

Exemplos de uso do controle em tempo de execução para alterar o nível de log:

```
### Aumentar o nível de log de todos os processos:
zabbix_proxy -c /usr/local/etc/zabbix_proxy.conf -R log_level_increase

### Aumentar o nível de log do segundo processo poller:
zabbix_proxy -c /usr/local/etc/zabbix_proxy.conf -R log_level_increase=poller,2

### Aumentar o nível de log do processo com PID 1234:
zabbix_proxy -c /usr/local/etc/zabbix_proxy.conf -R log_level_increase=1234

### Diminuir o nível de log de todos os processos http poller:
zabbix_proxy -c /usr/local/etc/zabbix_proxy.conf -R log_level_decrease="http poller"
```

Processo do usuário

O proxy Zabbix foi projetado para ser executado como um usuário não-root. Ele será executado como qualquer usuário não-root que for iniciado. Portanto, você pode executar o proxy como qualquer usuário não-root sem problemas.

Se você tentar executá-lo como 'root', ele mudará para o usuário 'zabbix', que deve estar presente no seu sistema. Você só pode executar o proxy como 'root' se modificar o parâmetro 'AllowRoot' no arquivo de configuração do proxy adequadamente.

Arquivo de configuração

Veja as opções do arquivo de configuração para detalhes sobre como configurar o zabbix_proxy.

Tipos de processos e threads do proxy

- `agent poller` - processo de poller assíncrono para verificações passivas com uma thread de trabalho
- `availability manager` - processo para atualizações de disponibilidade do host
- `browser poller` - poller para verificações de item de navegador
- `configuration syncer` - processo para gerenciar o cache em memória dos dados de configuração

- `data sender` - remetente de dados do proxy
- `discovery manager` - processo gerenciador para descoberta de dispositivos
- `discovery worker` - processo para lidar com tarefas de descoberta do discovery manager
- `history syncer` - gravador de histórico no banco de dados
- `housekeeper` - processo para remover histórico de item desatualizado
- `http agent poller` - processo de poller assíncrono para verificações HTTP com uma thread de trabalho
- `http poller` - poller de monitoramento web
- `icmp pinger` - poller para verificações icmpping
- `internal poller` - poller para verificações internas
- `ipmi manager` - gerenciador de poller IPMI
- `ipmi poller` - poller para verificações IPMI
- `java poller` - poller para verificações Java
- `odbc poller` - poller para verificações ODBC
- `poller` - poller normal para verificações passivas
- `preprocessing manager` - gerenciador de tarefas de pré-processamento com threads de trabalho de pré-processamento
- `preprocessing worker` - thread para pré-processamento de dados
- `self-monitoring` - processo para coletar estatísticas internas do servidor
- `snmp poller` - processo de poller assíncrono para verificações SNMP com uma thread de trabalho (apenas itens `walk[OID]` e `get[OID]`)
- `snmp trapper` - trapper para traps SNMP
- `task manager` - processo para execução remota de tarefas solicitadas por outros componentes (por exemplo, fechar problema, reconhecer problema, verificar valor de item agora, funcionalidade de comando remoto)
- `trapper` - trapper para verificações ativas, traps, comunicação de proxy
- `unreachable poller` - poller para dispositivos inalcançáveis
- `vmware collector` - coletor de dados VMware responsável pela coleta de dados dos serviços VMware

O arquivo de log do proxy pode ser usado para observar esses tipos de processos.

Desde o Zabbix 7.0.22, o arquivo de log do proxy é criado com permissões de leitura e gravação apenas para o proprietário do arquivo. Além disso, o arquivo é legível pelo grupo do proprietário. Todas as outras permissões são negadas.

Vários tipos de processos do proxy Zabbix podem ser monitorados usando o item interno **zabbix[process,<type>,<mode>,<state>]**.

Plataformas suportadas

O proxy Zabbix é executado na mesma lista de plataformas suportadas que o server Zabbix.

Buffer de memória

O buffer de memória permite armazenar novos dados (valores de item, descoberta de rede, autorregistro de host) no buffer e enviá-los para o Zabbix server sem acessar o banco de dados. O buffer de memória foi introduzido para o proxy a partir do Zabbix 7.0.

Em instalações anteriores ao Zabbix 7.0, os dados coletados eram armazenados no banco de dados antes de serem enviados para o Zabbix server. Para essas instalações, esse continua sendo o comportamento padrão após a atualização para o Zabbix 7.0.

Para desempenho otimizado, recomenda-se configurar o uso do buffer de memória no proxy. Isso é possível modificando o valor do ProxyBufferMode de ”disk” (padrão fixo para instalações existentes) para ”hybrid” (recomendado) ou ”memory”. Também é necessário definir o tamanho do buffer de memória (parâmetro ProxyMemoryBufferSize).

No modo híbrido, o buffer é protegido contra perda de dados, gravando os dados não enviados no banco de dados se o proxy for parado, o buffer estiver cheio ou os dados estiverem muito antigos. Quando todos os valores forem gravados no banco de dados, o proxy volta a usar o buffer de memória.

No modo de memória, o buffer de memória será usado, porém, não há proteção contra perda de dados. Se o proxy for parado ou a memória ficar cheia, os dados não enviados serão descartados.

O modo híbrido (ProxyBufferMode=hybrid) é aplicado a todas as novas instalações a partir do Zabbix 7.0.

Parâmetros adicionais, como ProxyMemoryBufferSize e ProxyMemoryBufferAge, definem, respectivamente, o tamanho do buffer de memória e a idade máxima dos dados no buffer.

Nota que, em caso de configuração conflitante, o proxy exibirá um erro e não iniciará, por exemplo, se:

- ProxyBufferMode estiver definido como ”hybrid” ou ”memory” e ProxyMemoryBufferSize for ”0”;
- ProxyBufferMode estiver definido como ”hybrid” ou ”memory” e ProxyLocalBuffer não for ”0”.

Localidade

Observe que o proxy requer uma localidade UTF-8 para que alguns itens textuais possam ser interpretados corretamente. A maioria dos sistemas modernos semelhantes ao Unix possuem uma localidade UTF-8 como padrão, no entanto, existem alguns sistemas

onde isso pode precisar ser definido especificamente.

Cálculo de filas durante a manutenção

> **Attention:**
> O proxy Zabbix não tem conhecimento dos períodos de manutenção; consulte Cálculo de filas durante a manutenção para obter detalhes.

5 Java gateway

Visão geral

O Zabbix Java gateway pode ser instalado a partir do código-fonte ou de pacotes.

O suporte nativo para monitoramento de aplicações JMX existe na forma de um daemon Zabbix chamado "Zabbix Java gateway". O Zabbix Java gateway é um daemon escrito em Java. Para descobrir o valor de um contador JMX específico em um host, o Zabbix server consulta o Zabbix Java gateway, que usa a API de gerenciamento JMX para consultar remotamente a aplicação de interesse. A aplicação não precisa de nenhum software adicional instalado, basta ser iniciada com a opção `-Dcom.sun.management.jmxremote` na linha de comando.

O Java gateway aceita conexões recebidas do Zabbix server ou proxy e só pode ser usado como um "proxy passivo". Ao contrário do Zabbix proxy, ele também pode ser usado a partir do Zabbix proxy (Zabbix proxies não podem ser encadeados). O acesso a cada Java gateway é configurado diretamente no arquivo de configuração do Zabbix server ou proxy, portanto, apenas um Java gateway pode ser configurado por Zabbix server ou Zabbix proxy. Se um host tiver items do tipo **JMX agent** e items de outro tipo, apenas os items **JMX agent** serão passados para o Java gateway para coleta.

Quando um item precisa ser atualizado via Java gateway, o Zabbix server ou proxy irá se conectar ao Java gateway e solicitar o valor, que o Java gateway, por sua vez, recupera e repassa ao server ou proxy. Assim, o Java gateway não armazena em cache nenhum valor.

O Zabbix server ou proxy possui um tipo específico de processo que se conecta ao Java gateway, controlado pela opção **StartJavaPollers**. Internamente, o Java gateway inicia múltiplas threads, controladas pela opção **START_POLLERS** opção. No lado do server, se uma conexão demorar mais do que **Timeout** segundos, ela será encerrada, mas o Java gateway ainda pode estar ocupado recuperando o valor do contador JMX. Para resolver isso, existe a opção **TIMEOUT** no Java gateway que permite definir o tempo limite para operações de rede JMX.

O server ou proxy do Zabbix tentará agrupar as solicitações para um único destino JMX tanto quanto possível (afetado pelos intervalos dos items) e enviá-las para o Java gateway em uma única conexão para melhor desempenho.

Sugere-se que o **StartJavaPollers** seja menor ou igual a **START_POLLERS**, caso contrário, pode haver situações em que nenhuma thread esteja disponível no Java gateway para atender às solicitações recebidas; nesse caso, o Java gateway usa ThreadPoolExecutor.CallerRunsPolicy, o que significa que a thread principal atenderá à solicitação recebida e não aceitará novas solicitações temporariamente.

Se você estiver tentando monitorar aplicações Java baseadas em Wildfly com o Zabbix Java gateway, instale o jboss-client.jar mais recente disponível na página de download do Wildfly.

2 Instalação a partir de pacotes RHEL

Visão geral

Se instalado a partir de pacotes, as informações a seguir ajudarão você a configurar o Zabbix Java gateway.

Configurando e executando o Java gateway

Os parâmetros de configuração do Zabbix Java gateway podem ser ajustados no arquivo:

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

Para mais detalhes, consulte os parâmetros de configuração do Zabbix Java gateway.

Para iniciar o Zabbix Java gateway:

```
systemctl restart zabbix-java-gateway
```

Para iniciar automaticamente o Zabbix Java gateway na inicialização:

RHEL 7 e posterior:

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

RHEL anterior ao 7:

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

Configurando o server para uso com o Java gateway

Com o Java gateway em execução, você deve informar ao Zabbix server onde encontrar o Zabbix Java gateway. Isso é feito especificando os parâmetros JavaGateway e JavaGatewayPort no arquivo de configuração do server. Se o host no qual a aplicação JMX está sendo executada for monitorado por um Zabbix proxy, então você deve especificar os parâmetros de conexão no arquivo de configuração do proxy.

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

Por padrão, o server não inicia nenhum processo relacionado ao monitoramento JMX. Se você deseja usá-lo, no entanto, deve especificar o número de instâncias pré-iniciadas de Java pollers. Você faz isso da mesma forma que especifica os pollers e trappers regulares.

```
StartJavaPollers=5
```

Não se esqueça de reiniciar o server ou proxy após terminar de configurá-los.

Depurando o Java gateway

O arquivo de log do Zabbix Java gateway é:

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

Se você quiser aumentar o nível de log, edite o arquivo:

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

e altere `level="info"` para ”debug” ou até mesmo ”trace” (para uma depuração mais profunda):

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

</configuration>
```

Monitoramento JMX

Consulte a página Monitoramento JMX para obter mais detalhes.

3 Instalação a partir de pacotes Debian/Ubuntu

Visão geral

Se instalado a partir de pacotes, as informações a seguir irão ajudá-lo a configurar o Zabbix Java gateway.

Configurando e executando o Java gateway

A configuração do Java gateway pode ser ajustada no arquivo:

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

Para mais detalhes, consulte os parâmetros de configuração do Zabbix Java gateway parâmetros.

Para iniciar o Zabbix Java gateway:

```
systemctl restart zabbix-java-gateway
```

Para iniciar automaticamente o Zabbix Java gateway na inicialização:

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

Configurando o server para uso com o Java gateway

Com o Java gateway em execução, você deve informar ao Zabbix server onde encontrar o Zabbix Java gateway. Isso é feito especificando os parâmetros JavaGateway e JavaGatewayPort no arquivo de configuração do server. Se o host no qual a aplicação JMX está sendo executada for monitorado por um Zabbix proxy, então você deve especificar os parâmetros de conexão no arquivo de configuração do proxy.

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

Por padrão, o server não inicia nenhum processo relacionado ao monitoramento JMX. Se você deseja usá-lo, deve especificar o número de instâncias pré-iniciadas de Java pollers. Você faz isso da mesma forma que especifica pollers e trappers regulares.

```
StartJavaPollers=5
```

Não se esqueça de reiniciar o server ou o proxy após terminar de configurá-los.

Depurando o Java gateway

O arquivo de log do Zabbix Java gateway é:

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

Se você quiser aumentar o nível de log, edite o arquivo:

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

e altere `level="info"` para ”debug” ou até mesmo ”trace” (para uma depuração mais profunda):

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

</configuration>
```

Monitoramento JMX

Consulte a página Monitoramento JMX para mais detalhes.

Instalação a partir do código-fonte

Visão geral

Se instalado a partir do código-fonte, as informações a seguir ajudarão você a configurar o Java gateway do Zabbix.

Visão geral dos arquivos

Se você obteve o Java gateway a partir do código-fonte, deve ter terminado com uma coleção de scripts shell, arquivos JAR e arquivos de configuração em $PREFIX/sbin/zabbix_java. O papel desses arquivos é resumido abaixo.

```
bin/zabbix-java-gateway-$VERSION.jar
```

Arquivo JAR do Java gateway propriamente dito.

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

Dependências do Java gateway: Logback, SLF4J e Android JSON library.

```
lib/logback.xml
lib/logback-console.xml
```

Arquivos de configuração para o Logback.

```
shutdown.sh
startup.sh
```

Scripts de conveniência para iniciar e parar o Java gateway.

```
settings.sh
```

Arquivo de configuração que é utilizado pelos scripts de inicialização e parada acima.

Configurando e executando o Java gateway

Por padrão, o Java gateway escuta na porta 10052. Se você planeja executar o Java gateway em uma porta diferente, pode especificar isso no script settings.sh. Veja a descrição do arquivo de configuração do Java gateway para saber como especificar esta e outras opções.

Warning:
A porta 10052 não é registrada na IANA.

Depois de ajustar as configurações, você pode iniciar o Java gateway executando o script de inicialização:

```
./startup.sh
```

Da mesma forma, quando não precisar mais do Java gateway, execute o script de desligamento para pará-lo:

```
./shutdown.sh
```

Observe que, ao contrário do server ou proxy, o Java gateway é leve e não precisa de um banco de dados.

Configurando o server para uso com o Java gateway

Com o Java gateway em execução, você deve informar ao Zabbix server onde encontrar o Zabbix Java gateway. Isso é feito especificando os parâmetros JavaGateway e JavaGatewayPort no arquivo de configuração do server. Se o host no qual a aplicação JMX está sendo executada for monitorado por um Zabbix proxy, então você deve especificar os parâmetros de conexão no arquivo de configuração do proxy.

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

Por padrão, o server não inicia nenhum processo relacionado ao monitoramento JMX. Se você deseja usá-lo, no entanto, deve especificar o número de instâncias pré-iniciadas de Java pollers. Você faz isso da mesma forma que especifica pollers e trappers regulares.

```
StartJavaPollers=5
```

Não se esqueça de reiniciar o server ou o proxy após terminar de configurá-los.

Depuração do Java gateway

Caso haja algum problema com o Java gateway ou uma mensagem de erro que você veja sobre um item no frontend não seja descritiva o suficiente, você pode querer dar uma olhada no arquivo de log do Java gateway.

Por padrão, o Java gateway registra suas atividades no arquivo /tmp/zabbix_java.log com o nível de log "info". Às vezes, essa informação não é suficiente e há necessidade de informações no nível de log "debug". Para aumentar o nível de log, modifique o arquivo lib/logback.xml e altere o atributo level da tag <root> para "debug":

```
<root level="debug">
  <appender-ref ref="FILE" />
</root>
```

Observe que, ao contrário do Zabbix server ou do Zabbix proxy, não há necessidade de reiniciar o Zabbix Java gateway após alterar o arquivo logback.xml - as alterações em logback.xml serão aplicadas automaticamente. Quando terminar a depuração, você pode retornar o nível de log para "info".

Se desejar registrar em um arquivo diferente ou em um meio completamente diferente, como um banco de dados, ajuste o arquivo logback.xml conforme necessário. Veja o Manual do Logback para mais detalhes.

Às vezes, para fins de depuração, é útil iniciar o Java gateway como um aplicativo de console em vez de um daemon. Para fazer isso, comente a variável PID_FILE em settings.sh. Se PID_FILE for omitida, o script startup.sh inicia o Java gateway como um aplicativo de console e faz com que o Logback use o arquivo lib/logback-console.xml, que não apenas registra no console, mas também tem o nível de log "debug" ativado.

Por fim, observe que, como o Java gateway usa SLF4J para registro em log, você pode substituir o Logback pelo framework de sua escolha, colocando o arquivo JAR apropriado no diretório lib. Veja o Manual do SLF4J para mais detalhes.

Monitoramento JMX

Consulte a página Monitoramento JMX para mais detalhes.

6 Sender

Visão geral

O Zabbix sender é uma ferramenta de linha de comando que pode ser usada para enviar dados de desempenho para o Zabbix server para processamento.

A ferramenta é normalmente usada em scripts de usuário de longa execução para o envio periódico de dados de disponibilidade e desempenho.

Para enviar resultados diretamente para o Zabbix server ou proxy, um item trapper deve ser configurado.

Veja também a biblioteca Python para Zabbix, que possui funcionalidade embutida para agir como o Zabbix sender.

Executando o Zabbix sender

Um exemplo de execução do Zabbix sender em UNIX:

```
cd bin
./zabbix_sender -z zabbix -s "Linux DB3" -k db.connections -o 43
```

onde:

- z - host do Zabbix server (o endereço IP também pode ser usado)
- s - nome técnico do host monitorado (conforme registrado no Zabbix frontend)
- k - chave do item
- o - valor a ser enviado

Attention:
Opções que contêm espaços em branco devem ser colocadas entre aspas duplas.

O Zabbix sender pode ser usado para enviar múltiplos valores a partir de um arquivo de entrada. Consulte a manpage do Zabbix sender para mais informações.

Se um arquivo de configuração for especificado, o Zabbix sender usa todos os endereços definidos no parâmetro de configuração ServerActive do agent para enviar os dados. Se o envio para um endereço falhar, o sender tenta enviar para os outros endereços. Se o envio de dados em lote falhar para um endereço, os lotes seguintes não serão enviados para esse endereço.

O Zabbix sender aceita strings em codificação UTF-8 (tanto para sistemas UNIX-like quanto para Windows) sem byte order mark (BOM) no início do arquivo.

O Zabbix sender no Windows pode ser executado de forma semelhante:

```
zabbix_sender.exe [opções]
```

Cenários de envio em tempo real do zabbix_sender irão agrupar múltiplos valores passados a ele em rápida sucessão e enviá-los ao server em uma única conexão. Um valor que não esteja mais distante do valor anterior do que 0,2 segundos pode ser colocado na mesma pilha, mas o tempo máximo de polling ainda é de 1 segundo.

Note:
O Zabbix sender será encerrado se uma entrada de parâmetro inválida (que não siga a notação *parâmetro=valor*) estiver presente no arquivo de configuração especificado.

Executando o Zabbix sender com descoberta de baixo nível

Um exemplo de execução do Zabbix sender para enviar um valor no formato JSON para descoberta de baixo nível:

```
./zabbix_sender -z 192.168.1.113 -s "Zabbix server" -k trapper.discovery.item -o '[{"{#FSNAME}":"/","{#FST
```

Para que isso funcione, a regra de descoberta de baixo nível deve ter um tipo de item Zabbix trapper (neste exemplo, com a chave `trapper.discovery.item`).

7 Get

Visão geral

Zabbix get é uma ferramenta de linha de comando, que pode ser usada para se comunicar com o agent Zabbix e recuperar as informações necessárias do agent.

A ferramenta é geralmente usada para solução de problemas de agents Zabbix.

Veja também a biblioteca Python para Zabbix que possui funcionalidade embutida para agir como o Zabbix get.

Executando o Zabbix get

Um exemplo de execução do Zabbix get em UNIX para obter o valor de carga do processador do agent:

```
cd bin
./zabbix_get -s 127.0.0.1 -p 10050 -k system.cpu.load[all,avg1]
```

Outro exemplo de execução do Zabbix get para capturar uma string de um site:

```
cd bin
./zabbix_get -s 192.168.1.1 -p 10050 -k "web.page.regexp[www.example.com,,,\"USA: ([a-zA-Z0-9.-]+)\",,\1]"
```

Observe que a chave do item aqui contém um espaço, então aspas são usadas para marcar a chave do item para o shell. As aspas não fazem parte da chave do item; elas serão removidas pelo shell e não serão passadas para o Zabbix agent.

Se uma chave de item não for suportada, o Zabbix get retornará o código de saída 1.

O Zabbix get aceita os seguintes parâmetros de linha de comando:

```
-s --host <host name or IP>             Especifica o nome do host ou endereço IP de um host
-p --port <port number>                 Especifica o número da porta do agent em execução no host (padrão:
-I --source-address <IP address>        Especifica o endereço IP de origem
-t --timeout <seconds>                  Especifica o tempo limite. Faixa válida: 1-600 segundos (padrão: 3
-k --key <item key>                     Especifica a chave do item para recuperar o valor
-P --protocol <value>                   Protocolo usado para se comunicar com o agent. Valores:
                                            auto - conecta usando o protocolo JSON, faz fallback e tenta n
                                            json - conecta usando o protocolo JSON
                                            plaintext - conecta usando o protocolo em texto simples, onde
-h --help                               Exibe esta mensagem de ajuda
-V --version                            Exibe o número da versão

--tls-connect <value>                   Como conectar ao agent. Valores:
                                            unencrypted - conecta sem criptografia (padrão)
                                            psk - conecta usando TLS e uma chave pré-compartilhada
                                            cert - conecta usando TLS e um certificado
--tls-ca-file <CA file>                 Caminho completo de um arquivo contendo os certificados da(s) CA(s
--tls-crl-file <CRL file>               Caminho completo de um arquivo contendo certificados revogados
--tls-agent-cert-issuer <cert issuer>   Emissor do certificado do agent permitido
--tls-agent-cert-subject <cert subject> Assunto do certificado do agent permitido
--tls-cert-file <cert file>             Caminho completo de um arquivo contendo o certificado ou cadeia de
--tls-key-file <key file>               Caminho completo de um arquivo contendo a chave privada
--tls-psk-identity <PSK-identity>       String exclusiva e sensível a maiúsculas/minúsculas usada para ide
--tls-psk-file <PSK-file>               Caminho completo de um arquivo contendo a chave pré-compartilhada
--tls-cipher13 <cipher-string>          String de cifra para OpenSSL 1.1.1 ou mais recente para TLS 1.3. S
--tls-cipher <cipher-string>            String de prioridade do GnuTLS (para TLS 1.2 e superior) ou string
```

Veja também a manpage do Zabbix get para mais informações.

O Zabbix get no Windows pode ser executado de forma semelhante:

```
zabbix_get.exe [opções]
```

8 JS

Visão geral

zabbix_js é uma ferramenta de linha de comando que pode ser usada para testar scripts incorporados.

Esta ferramenta executará um script do usuário com um parâmetro de string e imprimirá o resultado. Os scripts são executados usando o mecanismo de script incorporado do Zabbix.

Em caso de erros de compilação ou execução, o zabbix_js imprimirá o erro no stderr e sairá com o código 1.

Uso

```
zabbix_js -s script-file -p input-param [-l log-level] [-t timeout]
zabbix_js -s script-file -i input-file [-l log-level] [-t timeout]
zabbix_js -h
zabbix_js -V
```

zabbix_js aceita os seguintes parâmetros de linha de comando:

```
-s, --script script-file          Especifica o nome do arquivo do script a ser executado. Se "-" for espec
-i, --input input-file            Especifica o nome do arquivo das informações de entrada. O conteúdo será
-p, --param input-param           Especifica o parâmetro de entrada: a variável a ser passada para o scrip
-l, --loglevel log-level          Especifica o nível de log. Intervalo: 0-5.
-t, --timeout timeout             Especifica o tempo limite em segundos. Intervalo válido: 1-600 segundos
-h, --help                        Exibe informações de ajuda.
-V, --version                     Exibe o número da versão.
-w <webdriver url>                Habilita o monitoramento de navegador.
```

Exemplos:

Exemplo 1: Executando um script com uma variável contendo um valor

O script (`script-file.js`):

```
return value;
```

Uso:

```
zabbix_js -s script-file.js -p example_value
```

Saída: `example_value`

Exemplo 2: Executando um script com um arquivo contendo informações de entrada

O arquivo com o script (`script-file.js`):

```
return value;
```

O arquivo com as informações de entrada (`example.txt`):

```
Exemplo de informação de entrada do arquivo
```

Uso:

```
zabbix_js -s script-file.js -i example.txt
```

Saída: `Exemplo de informação de entrada do arquivo`

Exemplo 3: Executando um script com entrada lida do stdin

O arquivo com o script (`script-file.js`):

```
return value;
```

Uso:

```
zabbix_js -s script-file.js -i -
```

Entrada (stdin):

```
Exemplo de entrada do stdin
```

Saída: `Exemplo de entrada do stdin`

9 Serviço da Web

Visão geral

O serviço web do Zabbix é um processo utilizado para comunicação com serviços web externos. Atualmente, o serviço web do Zabbix é utilizado para gerar e enviar relatórios agendados, com planos para adicionar funcionalidades adicionais no futuro.

O servidor Zabbix conecta-se ao serviço web via HTTP(S). O serviço web do Zabbix requer que o Google Chrome esteja instalado no mesmo host; em algumas distribuições, o serviço também pode funcionar com o Chromium (consulte problemas conhecidos).

Instalação

O pacote oficial zabbix-web-service está disponível no repositório Zabbix.

Para compilar o Zabbix web service a partir do código-fonte, especifique a opção de configuração `--enable-webservice`.

Para configurar o Zabbix web service, atualize os parâmetros do arquivo de configuração `zabbix_web_service.conf`.

Attention:
É altamente recomendável configurar a criptografia entre o Zabbix server e o Zabbix web service usando certificados. Por padrão, os dados transmitidos entre o Zabbix server e o Zabbix web service não são criptografados, o que pode levar ao acesso não autorizado.

4 Instalação

Use a barra lateral para acessar o conteúdo da seção Instalação.

1 Obtendo o Zabbix

Visão geral

Há quatro formas de obter o Zabbix:

- Instale-o através dos pacotes de distribuição
- Baixe o arquivo fonte mais recente e o compile você mesmo
- Instale a partir dos contêineres
- Baixe a aplicação virtual

Para baixar os pacotes de distribuição mais recentes, fontes pré-compilados ou a aplicação virtual, acesse a página de download do Zabbix, onde links diretos para as versões mais recentes são disponibilizados.

Obtendo o código-fonte do Zabbix

Existem várias maneiras de obter o código-fonte do Zabbix:

- Você pode baixar as versões estáveis lançadas no site oficial do Zabbix
- Você pode baixar builds noturnos na página de desenvolvedores do site oficial do Zabbix
- Você pode obter a versão de desenvolvimento mais recente do sistema de repositório de código-fonte Git:
  - O local principal do repositório completo é em https://git.zabbix.com/scm/zbx/zabbix.git
  - O master e os lançamentos suportados também são espelhados no Github em https://github.com/zabbix/zabbix

Um cliente Git deve ser instalado para clonar o repositório. O pacote oficial do cliente Git de linha de comando é comumente chamado de **git** nas distribuições. Para instalar, por exemplo, no Debian/Ubuntu, execute:

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

Para obter todo o código-fonte do Zabbix, altere para o diretório onde deseja colocar o código e execute:

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

2 Requisitos

Hardware

Memória

O Zabbix requer memória física e em disco. A quantidade de memória em disco necessária obviamente depende do número de hosts e parâmetros que estão sendo monitorados. Se você planeja manter um histórico longo dos parâmetros monitorados, deve considerar pelo menos alguns gigabytes para ter espaço suficiente para armazenar o histórico no banco de dados. Cada processo do daemon do Zabbix requer várias conexões com um servidor de banco de dados. A quantidade de memória alocada para a conexão depende da configuração do mecanismo de banco de dados.

Note:
Quanto mais memória física você tiver, mais rápido o banco de dados (e, portanto, o Zabbix) funcionará.

CPU

O Zabbix e especialmente o banco de dados do Zabbix podem exigir recursos significativos de CPU, dependendo do número de parâmetros monitorados e do mecanismo de banco de dados escolhido.

Outro hardware

Uma porta de comunicação serial e um modem GSM serial são necessários para usar o suporte a notificações SMS no Zabbix. Conversores USB-para-serial também funcionarão.

Exemplos de configuração de hardware

A tabela fornece exemplos de configuração de hardware, assumindo uma plataforma **Linux/BSD/Unix**.

Estes são exemplos de tamanho e configuração de hardware para começar. Cada instalação do Zabbix é única. Certifique-se de avaliar o desempenho do seu sistema Zabbix em um ambiente de homologação ou desenvolvimento, para que você possa entender totalmente seus requisitos antes de implantar a instalação do Zabbix em seu ambiente de produção.

Tamanho da instalação	Métricas monitoradas[1]	Núcleos de CPU/vCPU	Memória (GiB)	Banco de dados	Amazon EC2[2]
Pequena	1 000	2	8	MySQL Server, Percona Server, MariaDB Server, PostgreSQL	m6i.large/m6g.large
Média	10 000	4	16	MySQL Server, Percona Server, MariaDB Server, PostgreSQL	m6i.xlarge/m6g.xlarge
Grande	100 000	16	64	MySQL Server, Percona Server, MariaDB Server, PostgreSQL, Oracle	m6i.4xlarge/m6g.4xlarge
Muito grande	1 000 000	32	96	MySQL Server, Percona Server, MariaDB Server, PostgreSQL, Oracle	m6i.8xlarge/m6g.8xlarge

[1] 1 métrica = 1 item + 1 trigger + 1 gráfico
 [2] Exemplo com instâncias EC2 de uso geral da Amazon, usando arquitetura ARM64 ou x86_64, um tipo de instância apropriado como Compute/Memory/Storage otimizado deve ser selecionado durante a avaliação e teste da instalação do Zabbix antes de instalar em seu ambiente de produção.

Note:
A configuração real depende muito do número de items ativos e das taxas de atualização (veja a seção tamanho do banco de dados desta página para mais detalhes). É altamente recomendável executar o banco de dados em um servidor separado para instalações grandes.

Plataformas suportadas

Devido aos requisitos de segurança e à natureza crítica do servidor de monitoramento, o UNIX é o único sistema operacional que pode fornecer consistentemente o desempenho, a tolerância a falhas e a resiliência necessários. O Zabbix opera nas versões líderes de mercado.

Os componentes do Zabbix estão disponíveis e foram testados nas seguintes plataformas:

Plataforma	Server	Agent	Agent 2	Comentários
Linux	x	x	x	

Plataforma	Server	Agent	Agent 2	Comentários
Windows	-	x	x	O agent do Zabbix é suportado em todas as versões desktop e server desde o Windows XP (64 bits)/Server 2003. O agent 2 do Zabbix é suportado em todas as versões desktop e server desde o Windows 10 (32 bits)/Server 2016, pois é compilado apenas com uma versão Go suportada para evitar vulnerabilidades críticas de segurança. Desde o Go 1.21, as versões mínimas exigidas do Windows foram elevadas, tornando o Windows 10/Server 2016 a versão mínima para o agent 2 do Zabbix.
macOS	x	x	-	
IBM AIX	x	x	-	O agent do Zabbix não funciona em plataformas AIX abaixo das versões 6.1 TL07 / 7.1 TL01.
FreeBSD	x	x	-	
OpenBSD	x	x	-	
Solaris	x	x	-	
NetBSD	x	x	-	
HP-UX	x	x	-	

Note:
O server/agent do Zabbix pode funcionar em outros sistemas operacionais semelhantes ao Unix.

Attention:
O Zabbix desabilita core dumps se compilado com criptografia e não inicia se o sistema não permitir a desativação de core dumps.

Software necessário

O Zabbix é construído em torno de servidores web modernos, principais mecanismos de banco de dados e linguagem de script PHP.

Software externo de terceiros

Se declarado como obrigatório, o software/biblioteca é estritamente necessário. Os opcionais são necessários para suportar alguma função específica.

Software	Status obrigatório	Versões suportadas	Comentários
MySQL/Percona	Um dos	8.0.30-9.6.X	Necessário se o MySQL (ou Percona) for usado como banco de dados backend do Zabbix. O mecanismo InnoDB é obrigatório. Adicionado suporte para versões do MySQL: - 8.4.X desde o Zabbix 7.0.1; - 9.0.X desde o Zabbix 7.0.2; - 9.5.X desde o Zabbix 7.0.22; - 9.6.X desde o Zabbix 7.0.25. Recomendamos o uso da biblioteca C API (libmysqlclient) para compilar o server/proxy.

Software	Status obrigatório	Versões suportadas	Comentários
MariaDB		10.5.00-12.2.X	O mecanismo InnoDB é obrigatório. A versão recomendada é a 11.4. Recomendamos o uso da biblioteca MariaDB Connector/C para compilar o server/proxy. Adicionado suporte para versões do MariaDB: - 11.4.X desde o Zabbix 7.0.1; - 11.5.X desde o Zabbix 7.0.4; - 12.0.X desde o Zabbix 7.0.19; - 12.2.X desde o Zabbix 7.0.25. Veja também: Possíveis deadlocks com MariaDB e Acesso a elementos da interface com MariaDB 10.5.1-10.5.9.
Oracle		19c - 21c	Necessário se o Oracle for usado como banco de dados backend do Zabbix. O suporte ao Oracle DB está **obsoleto** desde o Zabbix 7.0.
PostgreSQL		13.0-18.X	Necessário se o PostgreSQL for usado como banco de dados backend do Zabbix. Dependendo do tamanho da instalação, pode ser necessário aumentar a propriedade de configuração *work_mem* do PostgreSQL (4MB é o valor padrão), para que a quantidade de memória usada pelo banco de dados para uma determinada operação seja suficiente e a execução da consulta não demore muito. Adicionado suporte para versões do PostgreSQL: - 17.X desde o Zabbix 7.0.6; - 18.X desde o Zabbix 7.0.20.
TimescaleDB para PostgreSQL		2.13.0-2.26.X	Necessário se o TimescaleDB for usado como extensão do banco de dados PostgreSQL. Certifique-se de instalar o TimescaleDB Community Edition, que suporta compressão. Observe que o PostgreSQL 15 é suportado desde o TimescaleDB 2.10. Você também pode consultar a documentação do TimescaleDB para detalhes sobre a compatibilidade de versões entre PostgreSQL e TimescaleDB. Adicionado suporte para versões do TimescaleDB: - 2.15.X desde o Zabbix 7.0.1; - 2.16.X desde o Zabbix 7.0.4; - 2.17.X desde o Zabbix 7.0.6; - 2.18.X desde o Zabbix 7.0.10; - 2.19.X desde o Zabbix 7.0.13; - 2.20.X desde o Zabbix 7.0.17; - 2.21.X desde o Zabbix 7.0.18; - 2.22.X desde o Zabbix 7.0.20; - 2.23.X desde o Zabbix 7.0.22; - 2.24.X desde o Zabbix 7.0.23; - 2.25.X desde o Zabbix 7.0.24; - 2.26.X desde o Zabbix 7.0.25.
SQLite	Opcional	3.3.5-3.34.X	O SQLite é suportado apenas com proxies Zabbix. Necessário se o SQLite for usado como banco de dados do proxy Zabbix.
Elasticsearch		7.X	O Elasticsearch é suportado apenas com servers Zabbix, exclusivamente para armazenar dados de histórico. O suporte ao Elasticsearch é atualmente experimental. Veja também o software necessário para server/proxy.
smartmontools		7.1 ou superior	Necessário para o Zabbix agent 2.
who			Necessário para o plugin user count.

Software	Status obrigatório	Versões suportadas	Comentários
dpkg			Necessário para o plugin system.sw.packages.
pkgtool			Necessário para o plugin system.sw.packages.
rpm			Necessário para o plugin system.sw.packages.
pacman			Necessário para o plugin system.sw.packages.
q applets			`qlist` e `qsize`, como parte do q applets, são necessários para o plugin system.sw.packages no Gentoo Linux.

Note:
Embora o Zabbix possa funcionar com bancos de dados disponíveis nos sistemas operacionais, para a melhor experiência, recomendamos o uso de bancos de dados instalados a partir dos repositórios oficiais dos desenvolvedores dos bancos de dados.

Frontend

A largura mínima de tela suportada para o frontend do Zabbix é de 1200px.

Se declarado como obrigatório, o software/biblioteca é estritamente necessário. Os opcionais são necessários para suportar alguma função específica.

Software	Status obrigatório	Versões suportadas	Comentários
PHP	Sim	8.0.0 - 8.5.X	Adicionado suporte para versões do PHP: - 8.4.X desde o Zabbix 7.0.10; - 8.5.X desde o Zabbix 7.0.25.
Apache	Um dos	2.4 ou superior	
Nginx		1.20 ou superior	
MySQL	Um dos	Veja Software externo de terceiros	
PostgreSQL			
Oracle			
Extensões PHP			
mysqli	Sim		Necessário se o MySQL for usado como banco de dados backend do Zabbix.
pgsql			Necessário se o PostgreSQL for usado como banco de dados backend do Zabbix.
oci8			Necessário se o Oracle for usado como banco de dados backend do Zabbix.
bcmath			php-bcmath (*--enable-bcmath*)
mbstring			php-mbstring (*--enable-mbstring*)
sockets			php-net-socket (*--enable-sockets*); necessário para suporte a scripts de usuário.
gd		2.0.28 ou superior	php-gd (se fornecido como pacote separado pelo distribuidor); a extensão PHP GD deve suportar imagens PNG (*--with-png-dir*), imagens JPEG (*--with-jpeg-dir*) e FreeType 2 (*--with-freetype-dir*). A versão 2.3.0 ou superior pode ser necessária para evitar possíveis sobreposições de texto em gráficos para alguns idiomas do frontend.
libxml		2.6.15 ou superior	php-xml (se fornecido como pacote separado pelo distribuidor)
xmlwriter			php-xmlwriter (se fornecido como pacote separado pelo distribuidor)
xmlreader			php-xmlreader (se fornecido como pacote separado pelo distribuidor)
ctype			php-ctype (*--enable-ctype*)
session			php-session (se fornecido como pacote separado pelo distribuidor)
ldap	Não		php-ldap; necessário para autenticação LDAP.
openssl			php-openssl; necessário para autenticação SAML.
gettext			php-gettext (*--with-gettext*); necessário para traduções.

Software	Status obrigatório	Versões suportadas	Comentários
cURL			php-curl; necessário para Duo Universal Prompt MFA.

Bibliotecas de frontend de terceiros fornecidas com o Zabbix:

Biblioteca	Status obrigatório	Versão fornecida	Comentários
jQuery JavaScript Library	Sim	3.6.0	Biblioteca JavaScript que simplifica o processo de desenvolvimento entre navegadores.
jQuery UI		1.12.1	Um conjunto de interações de interface do usuário, efeitos, widgets e temas construídos sobre o jQuery.
SAML PHP Toolkit		4.3.1	Um toolkit PHP que adiciona suporte à autenticação SAML 2.0 para poder fazer login no Zabbix. Histórico de versões fornecidas: - 4.0.0 desde o Zabbix 7.0.0; - 4.3.1 desde o Zabbix 7.0.24 (recomendado).
Symfony Yaml Component		5.1.0	Adiciona suporte para exportar e importar elementos de configuração do Zabbix no formato YAML.

Note:
O Zabbix também pode funcionar em versões anteriores do Apache, MySQL, Oracle e PostgreSQL.

Attention:
Para outras fontes além da padrão DejaVu, a função PHP imagerotate pode ser necessária. Se estiver ausente, essas fontes podem ser renderizadas incorretamente quando um gráfico for exibido. Esta função só está disponível se o PHP for compilado com GD incorporado, o que não é o caso no Debian e em outras distribuições.

Bibliotecas de terceiros usadas para escrever e depurar o código do frontend do Zabbix:

Biblioteca	Status obrigatório	Versão mínima	Descrição
Composer	Não	2.4.1	Um gerenciador de pacotes em nível de aplicativo para PHP que fornece um formato padrão para gerenciar dependências de software PHP e bibliotecas necessárias.
PHPUnit		8.5.29	Um framework de teste unitário PHP para testar o frontend do Zabbix.
SASS		3.4.22	Uma linguagem de script de pré-processador que é interpretada e compilada em Cascading Style Sheets (CSS).

Navegador web no lado do cliente

Cookies e JavaScript devem estar habilitados.

As versões estáveis mais recentes do Google Chrome, Mozilla Firefox, Microsoft Edge, Apple Safari e Opera são suportadas.

Warning:
A política de mesma origem para IFrames está implementada, o que significa que o Zabbix não pode ser colocado em frames em um domínio diferente.

 Ainda assim, páginas colocadas em um frame do Zabbix terão acesso ao frontend do Zabbix (através de JavaScript) se a página que é colocada no frame e o frontend do Zabbix estiverem no mesmo domínio. Uma página como `http://secure-zabbix.com/cms/page.html`, se colocada em painéis em `http://secure-zabbix.com/zabbix/`, terá acesso total ao Zabbix via JS.

Server/proxy

Se declarado como obrigatório, o software/biblioteca exigido é estritamente necessário. Os opcionais são necessários para suportar alguma função específica.

Requisito	Status obrigatório	Descrição
libpcre/libpcre2	Um dos	A biblioteca PCRE/PCRE2 é necessária para suporte a Expressão Regular Compatível com Perl (PCRE). O nome pode variar dependendo da distribuição GNU/Linux, por exemplo, 'libpcre3' ou 'libpcre1'. PCRE v8.x e PCRE2 v10.x são suportados.
libevent	Sim	Necessário para comunicação entre processos. Versão 2.0.10 ou superior (desde o Zabbix 7.0.4; 1.4 antes disso).
libevent-pthreads		Necessário para comunicação entre processos.
libpthread		Necessário para suporte a mutex e bloqueio de leitura-escrita (pode fazer parte da libc).
libresolv		Necessário para resolução de DNS (pode fazer parte da libc).
libiconv		Necessário para conversão de codificação/formato de texto (pode fazer parte da libc). Obrigatório para o Zabbix server no Linux.
libz		Necessário para suporte a compressão.
libm		Biblioteca matemática. Necessária apenas para o Zabbix server.
libmysqlclient	Um dos	Necessário se o MySQL for usado.
libmariadb		Necessário se o MariaDB for usado.
libclntsh		Necessário se o Oracle for usado; a versão do *libclntsh* deve corresponder ou ser superior à versão do banco de dados Oracle utilizado.
libpq5		Necessário se o PostgreSQL for usado; a versão do *libpq5* deve corresponder ou ser superior à versão do banco de dados PostgreSQL utilizado.
libsqlite3		Necessário se o SQLite for usado. Necessário apenas para o Zabbix proxy.
libOpenIPMI	Não	Necessário para suporte a IPMI. Necessário apenas para o Zabbix server.
libssh2 ou *libssh*		Necessário para verificações SSH. Versão 1.0 ou superior (libssh2); 0.9.0 ou superior (libssh).
libcurl		Necessário para os seguintes recursos: - Monitoramento web, Monitoramento VMware e itens HTTP agent (para todos: versão 7.19.1 ou superior); - Itens Zabbix agent web.page.* (versão 7.19.1 ou superior; veja também os requisitos do agent); - Autenticação SMTP (versão 7.20.0 ou superior); - Elasticsearch (versão 7.28.0 ou superior). É recomendada a versão 7.28.0 ou superior para todos os recursos. Para usar recursos atualizados do cURL para itens `web.page.*`, reinicie o Zabbix server/proxy. Para autenticação SMTP, use o pacote `libcurl-full` em tempo de execução.
libxml2		Necessário para monitoramento VMware e pré-processamento XML XPath.
net-snmp		Necessário para suporte a SNMP. Versão 5.3.0 ou superior. O suporte a protocolos de criptografia forte (AES192/AES192C, AES256/AES256C) está disponível a partir da biblioteca net-snmp 5.8; em sistemas baseados em RHEL 8+ recomenda-se usar net-snmp 5.8.15 ou superior.
libunixodbc		Necessário para monitoramento de banco de dados.
libgnutls ou *libopenssl*		Necessário ao usar criptografia. Versões mínimas: *libgnutls* - 3.1.18, *libopenssl* - 1.0.1
libldap		Necessário para suporte a LDAP.
fping		Necessário para itens ICMP ping.
c-ares		Necessário para resolução assíncrona de DNS se o Zabbix for configurado com a opção `--with-ares`. Caso contrário, *libevent* será usado. Suportado desde o Zabbix 7.0.11; versão mínima: 1.16.0

Agent

Requisito	Status obrigatório	Descrição
libpcre/libpcre2	Um dos	A biblioteca PCRE/PCRE2 é necessária para suporte a Expressão Regular Compatível com Perl (PCRE). A nomenclatura pode variar dependendo da distribuição GNU/Linux, por exemplo, 'libpcre3' ou 'libpcre1'. PCRE v8.x e PCRE2 v10.x são suportados. Necessário para monitoramento de log. Também é necessário no Windows.

Requisito	Status obrigatório	Descrição
libpthread	Sim	Necessário para suporte a mutex e bloqueio de leitura-escrita (pode fazer parte do libc). Não é necessário no Windows.
libresolv		Necessário para resolução de DNS (pode fazer parte do libc). Não é necessário no Windows.
libiconv		Necessário para conversão de codificação/formato de texto para UTF-8 em itens de log, conteúdo de arquivo, regex de arquivo e itens regmatch (pode fazer parte do libc). Não é necessário no Windows.
libgnutls ou *libopenssl*	Não	Necessário se estiver usando criptografia. Versões mínimas: *libgnutls* - 3.1.18, *libopenssl* - 1.0.1 No Microsoft Windows, é necessário o OpenSSL 1.1.1 ou superior.
libldap		Necessário se o LDAP for usado. Não é suportado no Windows.
libcurl		Necessário para suporte estendido aos itens web.page.* do agent Zabbix. Sem o libcurl, a funcionalidade básica está disponível (por exemplo, web.page.get[http://example.com]). Com o libcurl, o agent suporta recursos adicionais, como URLs HTTP com credenciais (por exemplo, http://user:password@example.com) e URLs HTTPS. É necessária a versão 7.19.1 ou superior (7.28.0 ou superior é recomendada). Para usar recursos atualizados do cURL, reinicie o agent Zabbix.
libmodbus		Necessário apenas se o monitoramento Modbus for usado. Versão 3.0 ou superior.

Agent 2

Requisito	Status obrigatório	Descrição
Go	Sim	Necessário para compilar o Zabbix agent 2 e seus plugins a partir do código-fonte. Go 1.24.10 ou posterior é suportado. Veja go.dev para instruções de instalação. As bibliotecas Go usadas pelo Zabbix agent 2 e seus plugins estão listadas no repositório Git do Zabbix (as bibliotecas marcadas como `indirect` no repositório são dependências de outras bibliotecas necessárias): - Zabbix agent 2 - Suporte a plugin - PostgreSQL - MongoDB - MSSQL - Ember+ - Plugin de exemplo
libpcre/libpcre2	Um dos	A biblioteca PCRE/PCRE2 é necessária para suporte a Expressão Regular Compatível com Perl (PCRE). O nome pode variar dependendo da distribuição GNU/Linux, por exemplo, 'libpcre3' ou 'libpcre1'. PCRE v8.x e PCRE2 v10.x são suportados. Necessário para monitoramento de logs. Também é necessário no Windows.
libopenssl	Não	Necessário ao usar criptografia. OpenSSL 1.0.1 ou posterior é necessário em plataformas UNIX. A biblioteca OpenSSL deve ter suporte a PSK habilitado. LibreSSL não é suportado. Em sistemas Microsoft Windows, OpenSSL 1.1.1 ou posterior é necessário.

Serviço web

A versão estável mais recente do Google Chrome é suportada para gerar relatórios agendados usando o serviço web do Zabbix.

A versão do Go necessária para compilar o serviço web corresponde à utilizada para o Zabbix agent 2.

Java gateway

Se você obteve o Zabbix a partir do repositório fonte ou de um arquivo compactado, então as dependências necessárias já estão incluídas na árvore de fontes.

Se você obteve o Zabbix a partir do pacote da sua distribuição, então as dependências necessárias já são fornecidas pelo sistema de empacotamento.

Em ambos os casos acima, o software está pronto para ser usado e nenhum download adicional é necessário.

Se, no entanto, você desejar fornecer suas próprias versões dessas dependências (por exemplo, se estiver preparando um pacote para alguma distribuição Linux), abaixo está a lista de versões de bibliotecas com as quais o Java gateway é conhecido por funcionar. O Zabbix pode funcionar com outras versões dessas bibliotecas também.

A tabela a seguir lista os arquivos JAR que atualmente acompanham o Java gateway no código original:

Biblioteca	Status obrigatório	Versão incluída	Comentários
android-json	Sim	4.3r1	JSON (JavaScript Object Notation) é um formato leve de intercâmbio de dados. Esta é a implementação Android compatível com org.json extraída do Android SDK.
logback-classic		1.5.16	
logback-core		1.5.16	
slf4j-api		2.0.16	

O Java gateway pode ser compilado usando tanto o Oracle Java quanto o OpenJDK de código aberto (versão 1.6 ou superior). Os pacotes fornecidos pelo Zabbix são compilados usando o OpenJDK. A tabela a seguir lista os pacotes OpenJDK usados para compilar os pacotes Zabbix por distribuição:

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

Números de portas padrão

A seguinte lista de portas abertas por componente é aplicável para a configuração padrão:

Componente Zabbix	Número da porta	Protocolo	Tipo de conexão
Zabbix agent	10050	TCP	sob demanda
Zabbix agent 2	10050	TCP	sob demanda
Zabbix server	10051	TCP	sob demanda
Zabbix proxy	10051	TCP	sob demanda
Zabbix Java gateway	10052	TCP	sob demanda
Zabbix web service	10053	TCP	sob demanda
Zabbix frontend	80	HTTP	sob demanda
	443	HTTPS	sob demanda

Componente Zabbix	Número da porta	Protocolo	Tipo de conexão
Zabbix trapper	10051	TCP	sob demanda

Note:
Os números das portas devem ser abertos no firewall para permitir as comunicações do Zabbix. Conexões TCP de saída geralmente não requerem configurações explícitas de firewall.

Tamanho do banco de dados

Os dados de configuração do Zabbix requerem uma quantidade fixa de espaço em disco e não crescem muito.

O tamanho do banco de dados do Zabbix depende principalmente dessas variáveis, que definem a quantidade de dados históricos armazenados:

- Número de valores processados por segundo

Este é o número médio de novos valores que o Zabbix server recebe a cada segundo. Por exemplo, se tivermos 3000 items para monitoramento com uma taxa de atualização de 60 segundos, o número de valores por segundo é calculado como 3000/60 = **50**.

Isso significa que 50 novos valores são adicionados ao banco de dados do Zabbix a cada segundo.

- Configurações do housekeeper para histórico

O Zabbix mantém os valores por um período fixo de tempo, normalmente várias semanas ou meses. Cada novo valor requer uma certa quantidade de espaço em disco para dados e índice.

Portanto, se quisermos manter 30 dias de histórico e recebermos 50 valores por segundo, o número total de valores será em torno de (**30***24*3600)* **50** = 129.600.000, ou cerca de 130M de valores.

Dependendo do mecanismo de banco de dados usado, do tipo de valores recebidos (floats, inteiros, strings, arquivos de log, etc), o espaço em disco para manter um único valor pode variar de 40 bytes a centenas de bytes. Normalmente é em torno de 90 bytes por valor para items numéricos[2]. No nosso caso, isso significa que 130M de valores exigirão 130M * 90 bytes = **10,9GB** de espaço em disco.

Note:
O tamanho dos valores de items de texto/log é impossível de prever exatamente, mas você pode esperar cerca de 500 bytes por valor.

- Configuração do housekeeper para tendências

O Zabbix mantém um conjunto de valores máximo/mínimo/média/contagem de 1 hora para cada item na tabela **trends**. Os dados são usados para tendências e gráficos de longo período. O período de uma hora não pode ser personalizado.

O banco de dados do Zabbix, dependendo do tipo de banco de dados, requer cerca de 90 bytes para cada total. Suponha que queremos manter dados de tendências por 5 anos. Valores para 3000 items exigirão 3000*24*365* **90** = **2,2GB** por ano, ou **11GB** para 5 anos.

- Configurações do housekeeper para eventos

Cada evento do Zabbix requer aproximadamente 250 bytes de espaço em disco[1]. É difícil estimar o número de eventos gerados pelo Zabbix diariamente. No pior cenário, podemos supor que o Zabbix gera um evento por segundo.

Para cada evento recuperado, um registro event_recovery é criado. Normalmente, a maioria dos eventos será recuperada, então podemos assumir um registro event_recovery por evento. Isso significa mais 80 bytes por evento.

Opcionalmente, os eventos podem ter tags, cada registro de tag exigindo aproximadamente 100 bytes de espaço em disco[1]. O número de tags por evento (#tags) depende da configuração. Portanto, cada um precisará de mais #tags * 100 bytes de espaço em disco.

Isso significa que, se quisermos manter 3 anos de eventos, isso exigiria 3*365*24*3600* (250+80+#tags*100) = **~30GB**+#tags*100B de espaço em disco[2].

Note:
[1] Mais quando houver nomes de eventos, tags e valores não ASCII.
 [2] As aproximações de tamanho são baseadas em MySQL e podem ser diferentes para outros bancos de dados.

A tabela contém fórmulas que podem ser usadas para calcular o espaço em disco necessário para o sistema Zabbix:

Parâmetro	Fórmula para espaço em disco necessário (em bytes)
Configuração do Zabbix	Tamanho fixo. Normalmente 10MB ou menos.
Histórico	dias*(items/taxa de atualização)*24*3600*bytes items : número de items dias : número de dias para manter o histórico taxa de atualização : taxa de atualização média dos items bytes : número de bytes necessários para manter um único valor, depende do mecanismo do banco de dados, normalmente ~90 bytes.
Tendências	dias*(items/3600)*24*3600*bytes items : número de items dias : número de dias para manter o histórico bytes : número de bytes necessários para manter uma única tendência, depende do mecanismo do banco de dados, normalmente ~90 bytes.
Eventos	dias*eventos*24*3600*bytes eventos : número de eventos por segundo. Um (1) evento por segundo no pior cenário. dias : número de dias para manter o histórico bytes : número de bytes necessários para manter um único evento, depende do mecanismo do banco de dados, normalmente ~330 + número médio de tags por evento * 100 bytes.

Portanto, o espaço total necessário em disco pode ser calculado como:

Configuração + Histórico + Tendências + Eventos

O espaço em disco NÃO será usado imediatamente após a instalação do Zabbix. O tamanho do banco de dados crescerá e depois parará de crescer em algum momento, o que depende das configurações do housekeeper.

Sincronização de tempo

É muito importante ter o tempo do sistema preciso no servidor com o Zabbix em execução. ntpd é o daemon mais popular que sincroniza o tempo do host com o tempo de outras máquinas. É altamente recomendável manter o tempo do sistema sincronizado em todos os sistemas em que os componentes do Zabbix estão em execução.

Requisitos de rede

A seguinte lista de portas abertas por componente é aplicável para a configuração padrão.

Componentes	Porta
Frontend	http na 80, https na 443
Server	10051 (para uso com proxy/agents ativos)
Proxy Ativo	10051
Proxy Passivo	10051
Agent2	10050
Trapper	
JavaGateway	10052
WebService	10053

Note:
Os números das portas devem ser abertos no firewall para permitir comunicações externas com o Zabbix. Conexões TCP de saída geralmente não requerem configurações explícitas de firewall.

3 Instalação a partir do código-fonte

Você pode obter a versão mais recente do Zabbix compilando a partir do código-fonte.

Um tutorial passo a passo para instalar o Zabbix a partir do código-fonte é fornecido aqui.

Instalando os daemons do Zabbix

1 Baixe o arquivo de origem

Acesse a página de download do Zabbix e baixe o arquivo de origem. Após o download, extraia os arquivos de origem, executando:

```
tar -zxvf zabbix-7.0.0.tar.gz
```

> **Note:**
> Digite a versão correta do Zabbix no comando. Ela deve corresponder ao nome do arquivo baixado.

2 Criar conta de usuário

Todos os processos do daemon do Zabbix são executados sob usuários de sistema sem privilégios. Se um daemon do Zabbix for iniciado a partir de uma conta de usuário sem privilégios, ele continuará sendo executado como esse usuário.

Na configuração padrão, se um daemon for iniciado como `root`, ele mudará para a conta de usuário `zabbix`, que deve estar presente. Para criar um usuário e grupo de sistema `zabbix`, execute os comandos listados abaixo.

Sistema baseado em RedHat:

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

Sistema baseado em Debian:

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

Não é necessário criar uma conta de usuário separada para o frontend do Zabbix.

Recomendação de segurança

Se o server e o agent do Zabbix forem executados na mesma máquina, recomenda-se executá-los sob **contas de usuário separadas**. Executar ambos como o mesmo usuário permite que o agent acesse o arquivo de configuração do server, o que pode expor informações confidenciais—como a senha do banco de dados—para qualquer usuário com nível de Admin no Zabbix.

> **Attention:**
> Executar o Zabbix como `root`, `bin` ou qualquer outra conta com direitos especiais é um risco de segurança.

Diretório home (opcional)

Os processos do Zabbix não exigem um diretório home, portanto, geralmente não é recomendada a criação de um. No entanto, se você precisar de uma funcionalidade que exija um diretório home (por exemplo, armazenar credenciais do MySQL em $HOME/.my.cnf), você pode criá-lo usando os comandos listados abaixo.

Em sistemas baseados em RedHat, execute:

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

Em sistemas baseados em Debian, execute:

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

3 Criar o banco de dados do Zabbix

Para os daemons do Zabbix server e proxy, assim como para o frontend do Zabbix, é necessário um banco de dados. Ele não é necessário para executar o agent do Zabbix.

Scripts SQL são fornecidos para criar o esquema do banco de dados e inserir o conjunto de dados. O banco de dados do proxy Zabbix precisa apenas do esquema, enquanto o banco de dados do server Zabbix requer também o conjunto de dados além do esquema.

Tendo criado um banco de dados Zabbix, prossiga para as etapas seguintes de compilação do Zabbix.

4 Configurar os fontes

C99 com extensões GNU é necessário para compilar o Zabbix server, Zabbix proxy ou Zabbix agent. Esta versão pode ser explicitamente especificada definindo CFLAGS="-std=gnu99":

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

> **Note:**
> Se estiver instalando a partir do repositório Git do Zabbix, é necessário executar primeiro:
> `./bootstrap.sh`

Ao configurar os fontes para um Zabbix server ou proxy, você deve especificar o tipo de banco de dados a ser usado. Apenas um tipo de banco de dados pode ser compilado com um processo de server ou proxy por vez.

Para ver todas as opções de configuração suportadas, dentro do diretório de fontes extraído do Zabbix, execute:

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

Para configurar os fontes para um Zabbix server e agent, você pode executar algo como:

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

Para configurar os fontes para um Zabbix server (com PostgreSQL etc.), você pode executar:

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

Para configurar os fontes para um Zabbix proxy (com SQLite etc.), você pode executar:

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

Para configurar os fontes para um Zabbix agent, você pode executar:

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

ou, para o Zabbix agent 2:

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

Note:
Uma versão Go suportada deve ser instalada para compilar o Zabbix agent 2.

Notas sobre as opções de compilação:

- `--enable-agent` - compila o Zabbix agent, bem como as utilitários de linha de comando Zabbix get e Zabbix sender.
- `--with-libcurl` - necessário para monitoramento de máquinas virtuais, autenticação SMTP e `web.page.*` itens do Zabbix agent. Veja também: Requisitos (libcurl).
- `--with-libxml2` - necessário para monitoramento de máquinas virtuais.
- `--with-libpcre[=DIR]` - o Zabbix sempre compila com a biblioteca PCRE; esta opção apenas permite especificar um caminho de instalação personalizado para o PCRE.
- `--with-mysql=/path/to/mysql_config` - especifica o caminho para uma configuração específica da biblioteca cliente MySQL. Útil quando múltiplas versões do MySQL ou MariaDB estão instaladas.
- `--with-oracle` - especifica a localização da API Oracle Call Interface (OCI).
- `--enable-static` - vincula bibliotecas estaticamente (não suportado no Solaris). Use isto se você planeja distribuir binários compilados para sistemas sem as bibliotecas necessárias. Não recomendado ao compilar o Zabbix server. Para compilar o server estaticamente, uma versão estática de cada biblioteca externa é necessária. O script configure não verifica isso automaticamente.
- `--with-stacksize=<valor>` - define o tamanho da pilha por thread em kilobytes (por exemplo, `--with-stacksize=512`). Você pode aumentar este valor se o Zabbix travar ou congelar devido a estouros de pilha (por exemplo, durante o pré-processamento em sistemas com limites de pilha de thread padrão baixos).

Attention:
Se o `./configure` falhar devido à falta de bibliotecas ou outros problemas, verifique o arquivo `config.log` para obter informações detalhadas sobre o erro.
Por exemplo, se o `libssl` estiver ausente, a mensagem de erro imediata pode ser enganosa:

```
checking for main in -lmysqlclient... no
configure: error: Not found mysqlclient library
```

Neste caso, o `config.log` revela a causa real:

```
/usr/bin/ld: cannot find -lssl
/usr/bin/ld: cannot find -lcrypto
```

Veja também:

- Compilando o Zabbix com suporte a criptografia
- Problemas conhecidos de compilação

5 Compile e instale tudo

Note:
Se estiver instalando a partir do repositório Git do Zabbix, é necessário executar primeiro:

```
$ make dbschema
```

```
make install
```

Esta etapa deve ser executada como um usuário com permissões suficientes (comumente 'root', ou usando sudo).

Executar `make install` irá, por padrão, instalar os binários dos daemons (zabbix_server, zabbix_agentd, zabbix_proxy) em /usr/local/sbin e os binários dos clientes (zabbix_get, zabbix_sender) em /usr/local/bin.

> **Note:**
> Para especificar um local diferente de /usr/local, use a chave --prefix na etapa anterior de configuração das fontes, por exemplo --prefix=/home/zabbix. Neste caso, os binários dos daemons serão instalados em <prefix>/sbin, enquanto as utilidades em <prefix>/bin. As páginas de manual serão instaladas em <prefix>/share.

6 Revise e edite os arquivos de configuração

- edite o arquivo de configuração do agent Zabbix **/usr/local/etc/zabbix_agentd.conf**

Você precisa configurar este arquivo para cada host com o zabbix_agentd instalado.

Você deve especificar o **endereço IP** do server Zabbix no arquivo. Conexões de outros hosts serão negadas.

- edite o arquivo de configuração do server Zabbix **/usr/local/etc/zabbix_server.conf**

Você deve especificar o nome do banco de dados, usuário e senha (se estiver usando algum).

O restante dos parâmetros será adequado para você com seus valores padrão se você tiver uma instalação pequena (até dez hosts monitorados). No entanto, você deve alterar os parâmetros padrão se quiser maximizar o desempenho do server (ou proxy) Zabbix.

- se você instalou um proxy Zabbix, edite o arquivo de configuração do proxy **/usr/local/etc/zabbix_proxy.conf**

Você deve especificar o endereço IP do server e o hostname do proxy (deve ser conhecido pelo server), bem como o nome do banco de dados, usuário e senha (se estiver usando algum).

> **Note:**
> Com o SQLite, o caminho completo para o arquivo do banco de dados deve ser especificado; usuário e senha do banco de dados não são necessários.

7 Inicie os daemons

Execute o zabbix_server no lado do servidor.

```
zabbix_server
```

> **Note:**
> Certifique-se de que seu sistema permite a alocação de 36MB (ou um pouco mais) de memória compartilhada, caso contrário, o servidor pode não iniciar e você verá "Cannot allocate shared memory for <type of cache>." no arquivo de log do servidor. Isso pode acontecer no FreeBSD, Solaris 8.

Execute o zabbix_agentd em todas as máquinas monitoradas.

```
zabbix_agentd
```

> **Note:**
> Certifique-se de que seu sistema permite a alocação de 2MB de memória compartilhada, caso contrário, o agent pode não iniciar e você verá "Cannot allocate shared memory for collector." no arquivo de log do agent. Isso pode acontecer no Solaris 8.

Se você instalou o Zabbix proxy, execute o zabbix_proxy.

```
zabbix_proxy
```

Instalando a interface web do Zabbix

Copiando arquivos PHP

O frontend do Zabbix é escrito em PHP, portanto, para executá-lo, é necessário um servidor web com suporte a PHP. A instalação é feita simplesmente copiando os arquivos PHP do diretório ui para o diretório de documentos HTML do servidor web.

Os locais comuns dos diretórios de documentos HTML para servidores web Apache incluem:

- /usr/local/apache2/htdocs (diretório padrão ao instalar o Apache a partir do código-fonte)

- /srv/www/htdocs (OpenSUSE, SLES)
- /var/www/html (Debian, Ubuntu, Fedora, RHEL)

É sugerido usar um subdiretório em vez da raiz HTML. Para criar um subdiretório e copiar os arquivos do frontend do Zabbix para ele, execute os seguintes comandos, substituindo <htdocs> pelo diretório real:

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

Se planeja usar qualquer outro idioma além do inglês, consulte Instalação de idiomas adicionais do frontend para obter instruções.

Instalando o frontend

Consulte a página Instalação da interface web para obter informações sobre o assistente de instalação do frontend do Zabbix.

Instalando plugins carregáveis do agent 2 do Zabbix

A instalação de plugins carregáveis do agent 2 do Zabbix só é necessária se você quiser monitorar alvos não cobertos pelos plugins nativos (por exemplo, servidores ou clusters MongoDB, PostgreSQL e seus forks, etc.). Veja a lista completa de plugins carregáveis e plugins nativos.

Attention:
Antes de instalar um plugin, verifique o arquivo README. Ele pode conter requisitos específicos e instruções de instalação.

Para instalar a partir do código-fonte, primeiro baixe e extraia o arquivo de origem do plugin carregável.

Para compilar o plugin, navegue até o diretório extraído do plugin e execute `make`:

```
make
```

Note:
Uma versão suportada do Go deve estar instalada para compilar plugins carregáveis do agent 2 do Zabbix.

O executável do plugin pode ser colocado em qualquer lugar, desde que seja carregável pelo agent 2 do Zabbix. Especifique o caminho para o binário do plugin no arquivo de configuração do plugin, por exemplo, em postgresql.conf para o plugin PostgreSQL:

```
Plugins.PostgreSQL.System.Path=/caminho/para/executavel/zabbix-agent2-plugin-postgresql
```

O caminho para o arquivo de configuração do plugin deve ser especificado no parâmetro Include do arquivo de configuração do agent 2 do Zabbix:

```
Include=/caminho/para/arquivo/de/configuracao/do/plugin/postgresql.conf
```

Prossiga para a configuração para mais detalhes sobre como configurar plugins.

Plugins carregáveis fornecidos pelo Zabbix usam makefiles simples com os seguintes alvos de compilação:

- `make` - compila o plugin
- `make clean` - exclui todos os arquivos criados pela compilação do plugin
- `make check` - executa autotestes (requer um alvo real de monitoramento, por exemplo, um banco de dados PostgreSQL)
- `make style` - verifica o estilo do código Go com `golangci-lint`
- `make format` - formata o código Go com `go fmt`
- `make dist` - cria um arquivo de origem incluindo todas as dependências

Instalando o Java gateway

É necessário instalar o Java gateway apenas se você quiser monitorar aplicações JMX. O Java gateway é leve e não requer um banco de dados.

Para instalar a partir do código-fonte, primeiro baixe e extraia o arquivo de origem.

Para compilar o Java gateway, execute o script `./configure` com a opção `--enable-java`. É recomendável que você especifique a opção `--prefix` para solicitar um caminho de instalação diferente do padrão /usr/local, pois a instalação do Java gateway criará toda uma árvore de diretórios, não apenas um único executável.

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

Para compilar e empacotar o Java gateway em um arquivo JAR, execute `make`. Observe que, para esta etapa, você precisará dos executáveis `javac` e `jar` em seu caminho.

```
make
```

Agora você tem um arquivo zabbix-java-gateway-$VERSION.jar em src/zabbix_java/bin. Se você se sentir confortável em executar o Java gateway a partir de src/zabbix_java no diretório de distribuição, então pode prosseguir para as instruções de configuração e execução do Java gateway. Caso contrário, certifique-se de ter privilégios suficientes e execute `make install`.

```
make install
```

Prossiga para a configuração para mais detalhes sobre como configurar e executar o Java gateway.

Instalando o serviço web do Zabbix

A instalação do serviço web do Zabbix só é necessária se você quiser usar relatórios agendados.

Para instalar a partir do código-fonte, primeiro baixe e extraia o arquivo de origem.

Para compilar o serviço web do Zabbix, execute o script `./configure` com a opção `--enable-webservice`.

> **Note:**
> Uma versão do Go suportada deve ser instalada para compilar o serviço web do Zabbix.

Execute o zabbix_web_service na máquina onde o serviço web está instalado:

```
zabbix_web_service
```

Prossiga para a configuração para mais detalhes sobre a configuração da geração de relatórios agendados.

1 Compilando o agent do Zabbix no Windows

Visão geral

Esta página demonstra como compilar o agent do Zabbix a partir do código-fonte no Windows 10 (64 bits).

Essas instruções se aplicam às versões do Windows que suportam o Visual Studio 2022.

A compilação do agent do Zabbix requer:

- Compilador C (incluído nas Build Tools do Visual Studio 2022)
- OpenSSL (para recursos de criptografia no Zabbix)
- PCRE2 (Expressões Regulares Compatíveis com Perl; para recursos de correspondência de padrões de expressões regulares no Zabbix)

Você pode compilar o agent do Zabbix usando um dos seguintes métodos:

- Usando vcpkg—uma abordagem automatizada que simplifica o gerenciamento de dependências usando um gerenciador de pacotes C++.
- Compilação manual—uma abordagem manual que requer a instalação de todas as dependências antes de compilar o agent.

Dependendo das suas necessidades de monitoramento, bibliotecas adicionais podem ser necessárias. Para mais informações, consulte Requisitos.

> **Attention:**
> Antes de iniciar o processo de compilação, por favor, tenha em mente:

> - Para executar comandos, use o Prompt de Comando x64 Native Tools (incluído nas Build Tools do Visual Studio 2022), iniciado por um usuário com permissões suficientes para gravar em pastas protegidas.
> - Recomenda-se criar um diretório de trabalho em `C:\Zabbix` para todos os arquivos-fonte e pastas de compilação. No entanto, os componentes compilados devem ser instalados em `C:\Program Files\Zabbix\x64`.

Compilando o agent do Zabbix com vcpkg

Esta seção contém instruções para compilar o agent do Zabbix com o vcpkg, um gerenciador de pacotes que simplifica o gerenciamento de dependências e a integração com projetos C++.

1. Baixe e instale o Build Tools for Visual Studio 2022. Durante a instalação, certifique-se de selecionar a carga de trabalho *Desenvolvimento para desktop com C++*, que inclui as ferramentas necessárias para compilar o agent com o vcpkg:

- Compilador C (Microsoft Visual C++)
- Ferramenta de linha de comando NMake
- Gerenciador de pacotes vcpkg
- Prompt de comando de ferramentas nativas x64

2. Inicialize o vcpkg e instale as dependências necessárias para compilar o agent do Zabbix (observe que isso pode levar algum tempo):

```
cd C:\Zabbix
vcpkg new --application
vcpkg add port pcre2
vcpkg add port openssl
vcpkg install --triplet x64-windows-static --x-install-root="C:\Program Files\Zabbix\x64"
```

3. Baixe o arquivo fonte do Zabbix e extraia-o para `C:\Zabbix\zabbix-7.0.0`.

4. Navegue até o diretório de build do Zabbix (`C:\Zabbix\zabbix-7.0.0\build\win32\project`) e crie o seguinte script `build.bat`; certifique-se de especificar corretamente os diretórios onde o OpenSSL e o PCRE2 estão instalados:

```
:: Defina o caminho de instalação do vcpkg:
set vcpkg=C:\Program Files\Zabbix\x64\x64-windows-static

:: Execute o processo de build:
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. Compile o agent do Zabbix executando o script:

```
build.bat
```

Após a compilação, os binários dos componentes do Zabbix estarão localizados em `C:\Zabbix\zabbix-7.0.0\bin\win64`. O arquivo de configuração do agent do Zabbix está localizado em `C:\Zabbix\zabbix-7.0.0\conf`.

Para executar o agent, copie o `zabbix_agent.exe` e seu arquivo de configuração para uma pasta dedicada (por exemplo, `C:\Zabbix\agent`) e, em seguida, execute o agent:

```
mkdir C:\Zabbix\agent
copy C:\Zabbix\zabbix-7.0.0\bin\win64\zabbix_agent.exe C:\Zabbix\agent\
copy C:\Zabbix\zabbix-7.0.0\conf\zabbix_agent.win.conf C:\Zabbix\agent\

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

Compilando o agent Zabbix manualmente

Attention:
Este método de compilação do agent Zabbix é adequado para usuários que precisam de controle total sobre o ambiente de compilação ou estão em um ambiente restrito onde usar o vcpkg não é possível.

Esta seção contém instruções para compilar o agent Zabbix manualmente, o que inclui instalar as ferramentas de compilação e dependências necessárias (Perl, OpenSSL, PCRE2) e, em seguida, compilar o agent.

Instalando ferramentas de compilação

1. Baixe e instale o Build Tools for Visual Studio 2022. Durante a instalação, certifique-se de selecionar a carga de trabalho *Desenvolvimento de área de trabalho com C++*, que inclui as ferramentas necessárias para compilar o agent manualmente:

- Compilador C (Microsoft Visual C++)
- Ferramenta de linha de comando NMake
- Prompt de comando de ferramentas nativas x64

Instalando o OpenSSL

Note:
Para compilar o agent do Zabbix sem suporte a TLS, vá para a seção Instalando o PCRE2.

1. Baixe e instale o Strawberry Perl (disponível como instalador MSI). Durante a instalação, certifique-se de especificar `C:\Zabbix\Strawberry` como a pasta de instalação.

2. Instale o módulo Perl `Text::Template`:

```
cpanm Text::Template
```

3. Verifique se o Netwide Assembler (NASM; necessário para compilar o OpenSSL) foi compilado durante a instalação do Strawberry Perl:

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

Se o NASM não estiver compilado, instale-o manualmente. Para mais informações, consulte a documentação do NASM.

4. Baixe o arquivo de origem do OpenSSL e extraia-o para `C:\Zabbix\openssl-3.5.0`.

5. Navegue até o diretório extraído e configure o OpenSSL, por exemplo:

```
cd C:\Zabbix\openssl-3.5.0
perl Configure VC-WIN64A no-shared no-capieng no-winstore no-srp no-gost no-dgram no-dtls1-method no-dtls1
```

Attention:
Se você escolher um diretório personalizado para o OpenSSL ao compilar o agent do Zabbix no Windows (por exemplo, `C:\zabbix` ou `C:\openssl-64bit`), certifique-se de revogar o acesso de gravação de usuários não administradores a esse diretório. Caso contrário, o agent carregará as configurações SSL de um caminho que pode ser modificado por usuários sem privilégios, resultando em uma potencial vulnerabilidade de segurança.

- A opção `no-shared` faz com que as bibliotecas estáticas libcrypto.lib e libssl.lib do OpenSSL sejam autossuficientes, de modo que os binários do Zabbix incluam o OpenSSL sem precisar de DLLs externas. Isso significa que os binários do Zabbix podem ser copiados para outras máquinas Windows sem as bibliotecas OpenSSL; no entanto, quando uma nova versão de correção de bugs do OpenSSL for lançada, o agent do Zabbix precisará ser recompilado.
- Sem a opção `no-shared`, o Zabbix depende das DLLs do OpenSSL em tempo de execução. Isso significa que as atualizações do OpenSSL podem não exigir a recompilação do agent do Zabbix; no entanto, ao copiá-lo para outras máquinas, as DLLs do OpenSSL também devem ser copiadas.

Para mais informações sobre outras opções de configuração do OpenSSL, consulte a documentação do OpenSSL.

6. Compile o OpenSSL e execute os testes (observe que isso pode levar algum tempo):

Attention:
Execute os testes sem privilégios administrativos; caso contrário, isso pode levar a resultados inesperados ou riscos de segurança. Se alguns testes falharem, consulte a documentação do OpenSSL para solução de problemas.

```
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. Instale o OpenSSL:

```
nmake install
```

Para instalar apenas os componentes de software (bibliotecas, arquivos de cabeçalho, mas sem documentação), você pode usar `nmake install_sw`.

Instalando o PCRE2

1. Baixe e instale o CMake (disponível como instalador MSI). Durante a instalação, certifique-se de especificar `C:\Zabbix\CMake` como a pasta de instalação e selecione a opção *Add CMake to the PATH environment variable*.

2. Baixe o arquivo de origem do PCRE2 e extraia-o para `C:\Zabbix\pcre2-10.45`.

3. Crie um diretório `build` no diretório extraído do PCRE2 e navegue até ele:

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

4. Configure o PCRE2:

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

Note:
Se ocorrerem erros, recomenda-se excluir o cache do CMake antes de tentar repetir o processo de build do CMake. O cache (`CMakeCachecache.txt`) pode ser localizado no diretório build do diretório extraído do PCRE2.

5. Compile o PCRE2 usando o NMake:

```
nmake
```

6. Instale o PCRE2:

```
cmake --install .
```

Compilando o agent do Zabbix

1. Baixe o arquivo fonte do Zabbix e extraia-o para `C:\Zabbix\zabbix-7.0.0`.

Se você precisar gerar um arquivo fonte a partir do repositório de código-fonte bruto (por exemplo, para aplicar patches personalizados ou compilar a partir do código-fonte mais recente), execute os seguintes comandos em uma máquina **Linux**:

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

Isso criará um arquivo fonte, que pode então ser copiado para uma máquina Windows.

2. Navegue até o diretório de compilação do Zabbix e compile o agent do Zabbix (ou outros componentes); certifique-se de especificar corretamente os diretórios onde o OpenSSL e o PCRE2 estão instalados:

```
cd C:\Zabbix\zabbix-7.0.0\build\win32\project

#### Com suporte a TLS:
nmake /K -f Makefile_agent PCRE2INCDIR="C:\Program Files\Zabbix\x64\PCRE2\include" PCRE2LIBDIR="C:\Program
nmake /K -f Makefile_get PCRE2INCDIR="C:\Program Files\Zabbix\x64\PCRE2\include" PCRE2LIBDIR="C:\Program F
nmake /K -f Makefile_sender PCRE2INCDIR="C:\Program Files\Zabbix\x64\PCRE2\include" PCRE2LIBDIR="C:\Progra

#### Sem suporte a TLS:
nmake /K -f Makefile_agent PCRE2INCDIR="C:\Program Files\Zabbix\x64\PCRE2\include" PCRE2LIBDIR="C:\Program
nmake /K -f Makefile_get PCRE2INCDIR="C:\Program Files\Zabbix\x64\PCRE2\include" PCRE2LIBDIR="C:\Program F
nmake /K -f Makefile_sender PCRE2INCDIR="C:\Program Files\Zabbix\x64\PCRE2\include" PCRE2LIBDIR="C:\Progra
```

Após a compilação, os binários dos componentes do Zabbix estarão localizados em `C:\Zabbix\zabbix-7.0.0\bin\win64`. O arquivo de configuração do agent do Zabbix está localizado em `C:\Zabbix\zabbix-7.0.0\conf`.

Para executar o agent, copie `zabbix_agent.exe` e seu arquivo de configuração para uma pasta dedicada (por exemplo, `C:\Zabbix\agent`) e, em seguida, execute o agent:

```
mkdir C:\Zabbix\agent
copy C:\Zabbix\zabbix-7.0.0\bin\win64\zabbix_agentd.exe C:\Zabbix\agent\
copy C:\Zabbix\zabbix-7.0.0\conf\zabbix_agentd.win.conf C:\Zabbix\agent\

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

2 Compilando o Zabbix agent 2 no Windows

Visão geral

Esta página demonstra como compilar o agent 2 do Zabbix a partir do código-fonte no Windows 10 (64 bits ou 32 bits).

Ambas as versões de 32 bits e 64 bits podem ser compiladas em uma plataforma de 64 bits, mas apenas a versão de 32 bits pode ser compilada em uma plataforma de 32 bits.

A compilação do agent 2 do Zabbix requer:

- Ferramentas de compilação MinGW
- Linguagem de programação Go
- OpenSSL (para recursos de criptografia no Zabbix)

- PCRE2 (Expressões Regulares Compatíveis com Perl; para recursos de correspondência de padrões de expressões regulares no Zabbix)

Você pode compilar o agent 2 do Zabbix usando um dos seguintes métodos:

- Usando vcpkg—uma abordagem automatizada que simplifica o gerenciamento de dependências usando um gerenciador de pacotes C++.
- Compilação manual—uma abordagem manual que requer a instalação de todas as dependências antes de compilar o agent.

Attention:
Antes de iniciar o processo de compilação, por favor, tenha em mente:

- Para executar comandos, use o Prompt de Comando, iniciado por um usuário com permissões suficientes para gravar em pastas protegidas. No entanto, ao instalar o OpenSSL e o PCRE2, use o terminal MSYS2.
- Recomenda-se criar um diretório de trabalho em `C:\Zabbix` para todos os arquivos-fonte e pastas de compilação. No entanto, os componentes compilados devem ser instalados em `C:\Zabbix\x64` (ou `C:\Zabbix\x86` para compilações de 32 bits).

Compilando o Zabbix agent 2 com vcpkg

Esta seção contém instruções para compilar o Zabbix agent com o vcpkg, um gerenciador de pacotes que simplifica o gerenciamento de dependências e a integração com projetos C++.

1. Baixe e instale o Build Tools for Visual Studio 2022. Durante a instalação, certifique-se de selecionar a carga de trabalho *Desenvolvimento para desktop com C++*, que inclui o gerenciador de pacotes vcpkg.

2. Baixe e instale o Go (disponível como instalador MSI). Durante a instalação, certifique-se de especificar `C:\Zabbix\Go` como a pasta de instalação.

3. Baixe a distribuição MinGW que utiliza a biblioteca de tempo de execução Microsoft Visual C; por exemplo:

- Para compilações 64-bit: `x86_64-15.1.0-release-win32-seh-msvcrt-rt_v12-rev0.7z`
- Para compilações 32-bit: `i686-15.1.0-release-win32-dwarf-msvcrt-rt_v12-rev0.7z`

Em seguida, extraia para `C:\Zabbix\mingw64` (ou `C:\Zabbix\mingw32` para compilações 32-bit).

4. Inicialize o vcpkg e instale as dependências necessárias para compilar o Zabbix agent 2 (observe que isso pode levar algum tempo):

```
cd C:\Zabbix

set PATH=%PATH%;"C:\Program Files (x86)\Microsoft Visual Studio\2022\BuildTools\VC\vcpkg"
vcpkg new --application
vcpkg add port pcre2
vcpkg add port libiconv
vcpkg add port openssl

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

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

5. Baixe o arquivo-fonte do Zabbix e extraia para `C:\Zabbix\zabbix-7.0.0`.

6. Navegue até o diretório de compilação do Zabbix (`C:\Zabbix\zabbix-7.0.0\build\mingw`) e crie o seguinte script `build.bat`:

- Para compilações 64-bit:

```
:: Adicione o MinGW e o Go à variável de ambiente `PATH` para a sessão atual:
set PATH=C:\Zabbix\mingw64\bin;%PATH%
set PATH=C:\Zabbix\Go\bin;%PATH%

:: Defina o caminho de instalação do vcpkg:
set vcpkg="C:\Zabbix\x64\x64-mingw-static"

:: Defina as flags do linker para a biblioteca Crypt32:
```

```
SET CGO_LDFLAGS="-lCrypt32"

:: Execute o processo de compilação:
mingw32-make GOFLAGS="-buildvcs=false" ARCH=AMD64 ^
    PCRE2="%vcpkg%" ^
    OPENSSL="%vcpkg%" ^
    all
```

- Para compilações 32-bit:

```
:: Adicione o MinGW e o Go à variável de ambiente `PATH` para a sessão atual:
set PATH=C:\Zabbix\mingw32\bin;%PATH%
set PATH=C:\Zabbix\Go\bin;%PATH%

:: Defina o caminho de instalação do vcpkg:
set vcpkg="C:\Zabbix\x86\x86-mingw-static"

:: Defina as flags do linker para a biblioteca Crypt32:
SET CGO_LDFLAGS="-lCrypt32"

:: Execute o processo de compilação:
mingw32-make GOFLAGS="-buildvcs=false" ARCH=x86 ^
    PCRE2="%vcpkg%" ^
    OPENSSL="%vcpkg%" ^
    all
```

7. Compile o Zabbix agent 2 executando o script:

```
build.bat
```

Após a compilação, o binário do Zabbix agent 2 estará localizado em `C:\Zabbix\zabbix-7.0.0\bin\win64` (para compilações 64-bit) ou `C:\Zabbix\zabbix-7.0.0\bin\win32` (para compilações 32-bit). Os arquivos de configuração do Zabbix agent 2 estão localizados em `C:\Zabbix\zabbix-7.0.0\src\go\conf`.

Para executar o agent, copie o `zabbix_agent2.exe` e seus arquivos de configuração para uma pasta dedicada (por exemplo, `C:\Zabbix\agent2`) e então execute o agent:

```
mkdir C:\Zabbix\agent2

#### Para compilações 64-bit:
copy C:\Zabbix\zabbix-7.0.0\bin\win64\zabbix_agent2.exe C:\Zabbix\agent2\

#### Para compilações 32-bit:
copy C:\Zabbix\zabbix-7.0.0\bin\win32\zabbix_agent2.exe C:\Zabbix\agent2\

copy C:\Zabbix\zabbix-7.0.0\src\go\conf\zabbix_agent2.win.conf C:\Zabbix\agent2\
xcopy /E /I C:\Zabbix\zabbix-7.0.0\src\go\conf\zabbix_agent2.d C:\Zabbix\agent2\zabbix_agent2.d\

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

Se necessário, continue compilando os plugins carregáveis do Zabbix agent 2.

Compilando plugins carregáveis do agent 2 do Zabbix

1. Baixe o código-fonte do plugin do Zabbix que corresponda à sua versão do agent 2 do Zabbix (por exemplo, `zabbix-agent2-plugin-post` e extraia-o para `C:\Zabbix`.

> **Attention:**
> Antes de instalar um plugin, verifique o arquivo README. Ele pode conter requisitos específicos e instruções de instalação.

2. Navegue até o diretório do plugin extraído e compile o plugin:

```
cd C:\Zabbix\zabbix-agent2-plugin-ember-plus-7.0.0

#### Para builds 64-bit:
mingw32-make ARCH=AMD64
```

```
#### Para builds 32-bit:
mingw32-make ARCH=x86
```

Após a compilação, o binário do plugin do agent 2 do Zabbix e seu arquivo de configuração estarão localizados no mesmo diretório do plugin.

O executável do plugin pode ser colocado em qualquer lugar, desde que seja carregável pelo agent 2 do Zabbix. Especifique o caminho para o binário do plugin no arquivo de configuração do plugin, por exemplo, em postgresql.conf para o plugin PostgreSQL:

```
Plugins.PostgreSQL.System.Path=/caminho/para/o/executavel/zabbix-agent2-plugin-postgresql
```

O caminho para o arquivo de configuração do plugin deve ser especificado no parâmetro Include do arquivo de configuração do agent 2 do Zabbix:

```
Include=/caminho/para/o/arquivo/de/configuracao/do/plugin/postgresql.conf
```

Prossiga para a configuração para mais detalhes sobre como configurar plugins.

Compilando o Zabbix agent 2 manualmente

> **Attention:**
> Este método de compilação do Zabbix agent 2 é adequado para usuários que precisam de controle total sobre o ambiente de compilação ou estão em um ambiente restrito onde usar o vcpkg não é possível.

Esta seção contém instruções para compilar o Zabbix agent 2 manualmente, o que inclui instalar as ferramentas de compilação e dependências necessárias e, em seguida, compilar o agent.

Configurando as ferramentas de compilação

1. Baixe e instale o MSYS2 (disponível como um instalador MSI). Durante a instalação, certifique-se de especificar `C:\Zabbix\msys64` como a pasta de instalação.

2. Baixe e instale o Go (disponível como um instalador MSI; veja as versões Go atualmente suportadas). Durante a instalação, certifique-se de especificar `C:\Zabbix\Go` como a pasta de instalação.

3. Baixe a distribuição MinGW que utiliza a biblioteca de tempo de execução Microsoft Visual C; por exemplo:

- Para builds de 64 bits: `x86_64-15.1.0-release-win32-seh-msvcrt-rt_v12-rev0.7z`
- Para builds de 32 bits: `i686-15.1.0-release-win32-dwarf-msvcrt-rt_v12-rev0.7z`

Em seguida, extraia para `C:\Zabbix\mingw64` (ou `C:\Zabbix\mingw32` para builds de 32 bits).

Instalando o OpenSSL

> **Note:**
> Para compilar o agent Zabbix sem suporte a TLS, vá para a seção Instalando o PCRE2.

1. Abra o terminal MSYS2 MSYS com privilégios de administrador e execute os seguintes comandos:

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

2. Baixe o arquivo de origem do OpenSSL e extraia-o para `C:\Zabbix\openssl-3.5.0`.

3. Navegue até o diretório extraído do OpenSSL e crie o seguinte script `build.sh`:

- Para builds 64-bit:

```
####!/usr/bin/env bash

export PATH="/c/Zabbix/mingw64/bin:$PATH"
export d="/c/Zabbix/x64/OpenSSL-Win64-350-static"

perl Configure mingw64 no-shared no-capieng no-winstore no-srp no-gost no-dgram no-dtls1-method no-dtls1_2

make
make install
```

- Para builds 32-bit:

```
####!/usr/bin/env bash

export PATH="/c/Zabbix/mingw32/bin:$PATH"
export d="/c/Zabbix/x86/OpenSSL-Win64-350-static"

perl Configure mingw no-shared no-capieng no-winstore no-srp no-gost no-dgram no-dtls1-method no-dtls1_2-m

make
make install
```

Attention:
Certifique-se de revogar o acesso de gravação de usuários não administradores ao diretório `C:\Zabbix\x64\OpenSSL-Win64-350-static`. Caso contrário, o agent carregará as configurações SSL de um caminho que pode ser modificado por usuários sem privilégios, resultando em uma potencial vulnerabilidade de segurança.

- A opção `no-shared` faz com que as bibliotecas estáticas libcrypto.lib e libssl.lib do OpenSSL sejam autossuficientes, de modo que os binários do Zabbix incluam o OpenSSL sem a necessidade de DLLs externas. Isso significa que os binários do Zabbix podem ser copiados para outras máquinas Windows sem as bibliotecas OpenSSL; no entanto, quando uma nova versão de correção de bugs do OpenSSL for lançada, o agent Zabbix precisará ser recompilado.
- Sem a opção `no-shared`, o Zabbix depende das DLLs do OpenSSL em tempo de execução. Isso significa que as atualizações do OpenSSL podem não exigir a recompilação do agent Zabbix; no entanto, ao copiá-lo para outras máquinas, as DLLs do OpenSSL também devem ser copiadas.

Para mais informações sobre outras opções de configuração do OpenSSL, consulte a documentação do OpenSSL.

4. Configure e instale o OpenSSL executando o script (observe que isso pode levar algum tempo):

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

Instalando o PCRE2

1. Baixe o arquivo fonte do PCRE2 e extraia para `C:\Zabbix\pcre2-10.45`.

2. Abra o terminal MSYS2 MSYS com privilégios de administrador. Em seguida, crie um diretório `build` no diretório extraído do PCRE2 e navegue até ele:

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

3. Configure o PCRE2:

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

#### Para builds 32-bit:
export PATH="/c/Zabbix/mingw32/bin:$PATH"
cmake -DCMAKE_C_COMPILER=gcc -DCMAKE_C_FLAGS="-m32 -O2 -g" -DCMAKE_EXE_LINKER_FLAGS="-Wl,-mi386pe" -DCMAKE
```

Note:
Se ocorrerem erros, recomenda-se excluir o cache do CMake antes de tentar repetir o processo de build do CMake. O cache (`CMakeCachecache.txt`) pode ser localizado no diretório build do diretório extraído do PCRE2.

4. Instale o PCRE2:

```
make install
```

Compilando o Zabbix agent 2

1. Baixe o arquivo fonte do Zabbix e extraia para `C:\Zabbix\zabbix-7.0.0`.

Se você precisar gerar um arquivo fonte a partir do repositório de código-fonte bruto (por exemplo, para aplicar patches personalizados ou compilar a partir do código-fonte mais recente), execute os seguintes comandos em uma máquina **Linux** com Go instalado (necessário para configurar o Zabbix agent 2):

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

Isso criará um arquivo fonte, que pode então ser copiado para uma máquina Windows.

2. Abra o Prompt de Comando com privilégios de administrador. Em seguida, navegue até o diretório de build do Zabbix e compile o Zabbix agent; certifique-se de especificar corretamente os diretórios onde o OpenSSL e o PCRE2 estão instalados:

- Para builds 64-bit:

```
cd C:\Zabbix\zabbix-7.0.0\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

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

#### Sem suporte a TLS:
mingw32-make ARCH=AMD64 PCRE2="C:\Zabbix\x64\PCRE2"
```

- Para builds 32-bit:

```
cd C:\Zabbix\zabbix-7.0.0\build\mingw
set PATH=C:\Zabbix\mingw32\bin;%PATH%

#### Com suporte a TLS:
mingw32-make ARCH=x86 PCRE2="C:\Zabbix\x86\PCRE2" OPENSSL="C:\Zabbix\x86\OpenSSL-Win64-350-static"

#### Sem suporte a TLS:
mingw32-make ARCH=x86 PCRE2="C:\Zabbix\x86\PCRE2"
```

Após a compilação, o binário do Zabbix agent 2 estará localizado em `C:\Zabbix\zabbix-7.0.0\bin\win64` (ou `C:\Zabbix\zabbix-7.0` para builds 32-bit). Os arquivos de configuração do Zabbix agent 2 estão localizados em `C:\Zabbix\zabbix-7.0.0\src\go\conf`.

Para executar o agent, copie o binário `zabbix_agent2.exe` e seus arquivos de configuração para uma pasta dedicada (por exemplo, `C:\Zabbix\agent2`) e então execute o agent:

```
mkdir C:\Zabbix\agent2
copy C:\Zabbix\zabbix-7.0.0\bin\win64\zabbix_agent2.exe C:\Zabbix\agent2\
copy C:\Zabbix\zabbix-7.0.0\src\go\conf\zabbix_agent2.win.conf C:\Zabbix\agent2\
xcopy /E /I C:\Zabbix\zabbix-7.0.0\src\go\conf\zabbix_agent2.d C:\Zabbix\agent2\zabbix_agent2.d\

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

Se necessário, continue com a compilação dos plugins carregáveis do Zabbix agent 2.

3 Compilando o agent do Zabbix no macOS

Visão geral

Esta seção demonstra como construir binários do agent Zabbix para macOS a partir do código-fonte com ou sem TLS.

Pré-requisitos

Você precisará das ferramentas de desenvolvedor de linha de comando (Xcode não é necessário), Automake, pkg-config e PCRE (v8.x) ou PCRE2 (v10.x). Se você quiser compilar os binários do agent com TLS, também precisará do OpenSSL ou GnuTLS.

Para instalar o Automake e o pkg-config, você precisará do gerenciador de pacotes Homebrew de https://brew.sh/. Para instalá-lo, abra o terminal e execute o seguinte comando:

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

Em seguida, instale o Automake e o pkg-config:

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

A preparação das bibliotecas PCRE, OpenSSL e GnuTLS depende da forma como elas serão vinculadas ao agent.

Se você pretende executar os binários do agent em uma máquina macOS que já possui essas bibliotecas, pode usar as bibliotecas pré-compiladas fornecidas pelo Homebrew. Normalmente, essas são máquinas macOS que usam o Homebrew para compilar os binários do agent do Zabbix ou para outros fins.

Se os binários do agent forem usados em máquinas macOS que não possuem a versão compartilhada das bibliotecas, você deve compilar as bibliotecas estáticas a partir dos fontes e vincular o agent do Zabbix a elas.

Compilando binários do agent com bibliotecas compartilhadas

Instale o PCRE2 (substitua *pcre2* por *pcre* nos comandos abaixo, se necessário):

```
brew install pcre2
```

Ao compilar com TLS, instale OpenSSL e/ou GnuTLS:

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

Baixe o código-fonte do Zabbix:

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

Compile o agent sem TLS:

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

Compile o agent com OpenSSL:

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

Compile o agent com 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
```

Compilando binários do agent com bibliotecas estáticas sem TLS

Vamos supor que as bibliotecas estáticas do PCRE serão instaladas em `$HOME/static-libs`. Usaremos o PCRE2 10.39.

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

Baixe e compile o PCRE com suporte a propriedades Unicode:

```
mkdir static-libs-source
cd static-libs-source
curl --remote-name https://github.com/PhilipHazel/pcre2/releases/download/pcre2-10.39/pcre2-10.39.tar.gz
tar xf pcre2-10.39.tar.gz
cd pcre2-10.39
./configure --prefix="$PCRE_PREFIX" --disable-shared --enable-static --enable-unicode-properties
make
make check
make install
```

Baixe o código-fonte do Zabbix e compile o agent:

```
git clone https://git.zabbix.com/scm/zbx/zabbix.git
cd zabbix
./bootstrap.sh
./configure --sysconfdir=/usr/local/etc/zabbix --enable-agent --enable-ipv6 --with-libpcre2="$PCRE_PREFIX"
make
make install
```

Compilando binários do agent com bibliotecas estáticas com OpenSSL

Ao compilar o OpenSSL, é recomendável executar `make test` após a compilação bem-sucedida. Mesmo que a compilação tenha sido bem-sucedida, os testes às vezes falham. Se este for o caso, os problemas devem ser pesquisados e resolvidos antes de continuar.

Vamos supor que as bibliotecas estáticas do PCRE e do OpenSSL serão instaladas em `$HOME/static-libs`. Usaremos PCRE2 10.39 e OpenSSL 1.1.1a.

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

Vamos compilar as bibliotecas estáticas em `static-libs-source`:

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

Baixe e compile o PCRE com suporte a propriedades Unicode:

```
curl --remote-name https://github.com/PhilipHazel/pcre2/releases/download/pcre2-10.39/pcre2-10.39.tar.gz
tar xf pcre2-10.39.tar.gz
cd pcre2-10.39
./configure --prefix="$PCRE_PREFIX" --disable-shared --enable-static --enable-unicode-properties
make
make check
make install
cd ..
```

Baixe e compile o 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 ..
```

Baixe o código-fonte do Zabbix e compile o agent:

```
git clone https://git.zabbix.com/scm/zbx/zabbix.git
cd zabbix
./bootstrap.sh
./configure --sysconfdir=/usr/local/etc/zabbix --enable-agent --enable-ipv6 --with-libpcre2="$PCRE_PREFIX"
make
make install
```

Compilando binários do agent com bibliotecas estáticas com GnuTLS

O GnuTLS depende do backend criptográfico Nettle e da biblioteca aritmética GMP. Em vez de usar a biblioteca GMP completa, este guia usará mini-gmp, que está incluída no Nettle.

Ao compilar o GnuTLS e o Nettle, é recomendável executar `make check` após a compilação bem-sucedida. Mesmo que a compilação tenha sido bem-sucedida, os testes às vezes falham. Se este for o caso, os problemas devem ser pesquisados e resolvidos antes de continuar.

Vamos supor que as bibliotecas estáticas do PCRE, Nettle e GnuTLS serão instaladas em `$HOME/static-libs`. Usaremos PCRE2 10.39, Nettle 3.4.1 e GnuTLS 3.6.5.

```
PCRE_PREFIX="$HOME/static-libs/pcre2-10.39"
NETTLE_PREFIX="$HOME/static-libs/nettle-3.4.1"
GNUTLS_PREFIX="$HOME/static-libs/gnutls-3.6.5"
```

Vamos compilar as bibliotecas estáticas em `static-libs-source`:

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

Baixe e compile o 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 ..
```

Baixe e compile o 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 ..
```

Baixe o código-fonte do Zabbix e compile o agent:

```
git clone https://git.zabbix.com/scm/zbx/zabbix.git
cd zabbix
./bootstrap.sh
CFLAGS="-Wno-unused-command-line-argument -framework Foundation -framework Security" \
> LIBS="-lgnutls -lhogweed -lnettle" \
> LDFLAGS="-L$GNUTLS_PREFIX/lib -L$NETTLE_PREFIX/lib" \
> ./configure --sysconfdir=/usr/local/etc/zabbix --enable-agent --enable-ipv6 --with-libpcre2="$PCRE_PREFI
make
make install
```

4 Instalação a partir de pacotes

Visão geral

Os pacotes oficiais do Zabbix 7.0 estão disponíveis no site do Zabbix, onde você pode selecionar seu sistema operacional e o componente do Zabbix para gerar instruções de instalação adequadas ao seu ambiente. Veja também as notas de instalação de pacotes nesta página para informações adicionais importantes, bem como a seção Instalação e configuração.

Os pacotes estão disponíveis para as seguintes distribuições Linux:

- Red Hat Enterprise Linux e seus derivados: AlmaLinux, Amazon Linux 2023, CentOS Stream, CentOS, Oracle Linux, Rocky Linux
- Debian, Ubuntu, Raspberry Pi OS, Raspbian
- SUSE Linux Enterprise Server, openSUSE Leap

> **Attention:**
> Algumas distribuições de SO (em particular, distribuições baseadas em Debian) fornecem seus próprios pacotes do Zabbix. Esses pacotes **não** são suportados pelo Zabbix e podem estar desatualizados ou sem os recursos e correções de bugs mais recentes. Recomenda-se usar apenas pacotes oficiais do Repositório Oficial do Zabbix. Se você instalou anteriormente o Zabbix a partir do repositório do seu sistema operacional, consulte as etapas para atualizar pacotes do Zabbix a partir de repositórios do SO.

Os pacotes suportam banco de dados MySQL/PostgreSQL e servidor web Apache/Nginx. Observe que o Zabbix server e o proxy não podem compartilhar o mesmo banco de dados; use nomes de banco de dados diferentes se ambos forem instalados no mesmo host.

Se necessário, pacotes separados para Zabbix agent/agent 2, Zabbix get e Zabbix sender estão disponíveis no Repositório Oficial do Zabbix.

> **Attention:**
> Desde o Zabbix 7.0.5, os pacotes do Zabbix agent 2 não estão mais disponíveis para o RHEL 6 devido ao Go 1.22 (e mais recente) ser incompatível com o GCC 4.4.7 fornecido pelo RHEL 6, o que resulta em erros de compilação. Observe, no entanto, que os pacotes do Zabbix agent ainda são suportados.

O Zabbix também fornece binários pré-compilados do Zabbix agent para sistemas operacionais não-Linux; veja:

- Instalação do agent no Windows a partir do MSI
- Instalação do agent no macOS a partir do PKG
- Binários legados (para sistemas mais antigos/menos comuns, como HP-UX, NetBSD, Tru64 e versões mais antigas do SLES)

Notas sobre a instalação de pacotes

As seguintes notas se aplicam a todos os sistemas:

- Se estiver usando PostgreSQL, `DBHost=localhost` (ou um endereço IP) na configuração do Zabbix server/proxy faz com que o PostgreSQL use um socket de rede em vez de um socket UNIX local; veja Configuração do SELinux para instruções relacionadas.
- Se estiver usando TimescaleDB, veja a configuração adicional do TimescaleDB.
- Se estiver instalando o Java gateway do Zabbix (para monitorar aplicações JMX), veja instruções adicionais de configuração para sistemas baseados em RHEL e sistemas baseados em Debian.
- Para executar o agent do Zabbix como root, veja Executando o agent como root.

As seguintes notas se aplicam ao RHEL e seus derivados:

- Se você ativou o repositório EPEL para EL9, que também fornece pacotes do Zabbix, ele deve ser excluído da resolução de pacotes antes de instalar os pacotes oficiais do Zabbix; veja Instalação acidental de pacotes Zabbix do EPEL.
- Para instalar pacotes do Zabbix em ambientes Red Hat UBI, veja Pacotes Zabbix para RHEL em ambientes Red Hat UBI.
- Para usar itens de ping ICMP, pacotes para `fping` também estão disponíveis no Repositório Oficial do Zabbix.

Configuração do SELinux

O Zabbix utiliza comunicação entre processos baseada em socket. Em sistemas onde o Security-Enhanced Linux (SELinux) está habilitado, pode ser necessário adicionar regras SELinux para permitir que o Zabbix crie/utilize sockets de domínio UNIX no diretório SocketDir. Os arquivos de socket são usados pelo Zabbix server (alerter, preprocessing, IPMI) e pelo Zabbix proxy (IPMI), e estão presentes enquanto o processo está em execução.

Com o SELinux habilitado no modo enforcing, execute os seguintes comandos para habilitar a comunicação entre o Zabbix frontend e o server:

Para RHEL 7 (e posteriores), AlmaLinux, CentOS Stream, Oracle Linux, Rocky Linux 8 (e posteriores):

```
setsebool -P httpd_can_connect_zabbix on
```

Se o banco de dados for acessado pela rede (incluindo `localhost` para PostgreSQL), permita também que o Zabbix frontend se conecte ao banco de dados:

```
setsebool -P httpd_can_network_connect_db on
```

Para RHEL anterior ao 7:

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

Após aplicar as configurações do SELinux, reinicie o Apache:

```
systemctl restart httpd
```

Opcionalmente, você pode instalar um pacote pré-definido `zabbix-selinux-policy` do Repositório Oficial do Zabbix. Este pacote é fornecido para todas as versões de SO suportadas para simplificar a implantação do Zabbix e evitar que os usuários desativem o SELinux devido à complexidade da configuração.

> **Attention:**
> Para máxima segurança, recomenda-se definir configurações personalizadas do SELinux.

O pacote `zabbix-selinux-policy` contém uma política SELinux básica, permitindo que o Zabbix crie e utilize sockets e habilitando a conexão HTTPd ao PostgreSQL (usado pelo frontend).

O arquivo fonte `zabbix_policy.te` contém as seguintes regras:

```
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;
```

Pacotes Debuginfo

Os pacotes Debuginfo contêm símbolos de depuração para os binários do Zabbix. Eles não são necessários para a instalação ou operação normal, mas são úteis para solução de problemas avançada.

Para habilitar o repositório `zabbix-debuginfo`:

- No RHEL 6 ou 7, edite `/etc/yum.repos.d/zabbix.repo` e defina `enabled=1` para a seção `zabbix-debuginfo`:

```
[zabbix-debuginfo]
name=Zabbix Official Repository debuginfo - $basearch
baseurl=https://repo.zabbix.com/zabbix/7.0/rhel/7/$basearch/debuginfo/
enabled=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-ZABBIX-A14FE591
gpgcheck=1
```

- No SUSE, edite `/etc/zypp/repos.d/zabbix.repo` e defina `enabled=1` para a seção `zabbix-debuginfo`:

```
[zabbix-debuginfo]
name=Zabbix Official Repository debuginfo
type=rpm-md
baseurl=https://repo.zabbix.com/zabbix/7.0/sles/15/$basearch/debuginfo/
gpgcheck=1
gpgkey=https://repo.zabbix.com/zabbix/7.0/sles/15/$basearch/debuginfo/repodata/repomd.xml.key
enabled=0
update=1
```

Uma vez habilitado, instale os pacotes:

- No RHEL, instale um único pacote com informações de depuração para todos os componentes do Zabbix:

```
dnf install zabbix-debuginfo
```

- No SUSE, instale pacotes debuginfo específicos para cada componente:

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

1 Instalação do agent do Windows a partir do MSI

Visão geral

O agent do Zabbix pode ser instalado no Windows usando pacotes de instalação MSI de 32 ou 64 bits, disponíveis para download.

Os requisitos mínimos do sistema operacional para instalação via MSI são:

- **Para o agent do Zabbix:** Windows XP (64 bits) ou Windows Server 2003
- **Para o agent 2 do Zabbix:** Windows 10 (32 bits) ou Windows Server 2016

Pacotes de 32 bits não podem ser instalados em sistemas de 64 bits.

Os pacotes incluem:

- Suporte a TLS (a configuração de TLS é opcional)
- Utilitários Zabbix get e Zabbix sender (podem ser instalados junto com o agent/agent 2 do Zabbix ou separadamente)

Attention:
Os pacotes do agent 2 do Zabbix não incluem plugins carregáveis (MongoDB, PostgreSQL, MSSQL), que precisam ser baixados e instalados separadamente.

A instalação pode ser feita usando o Assistente de Instalação ou a linha de comando.

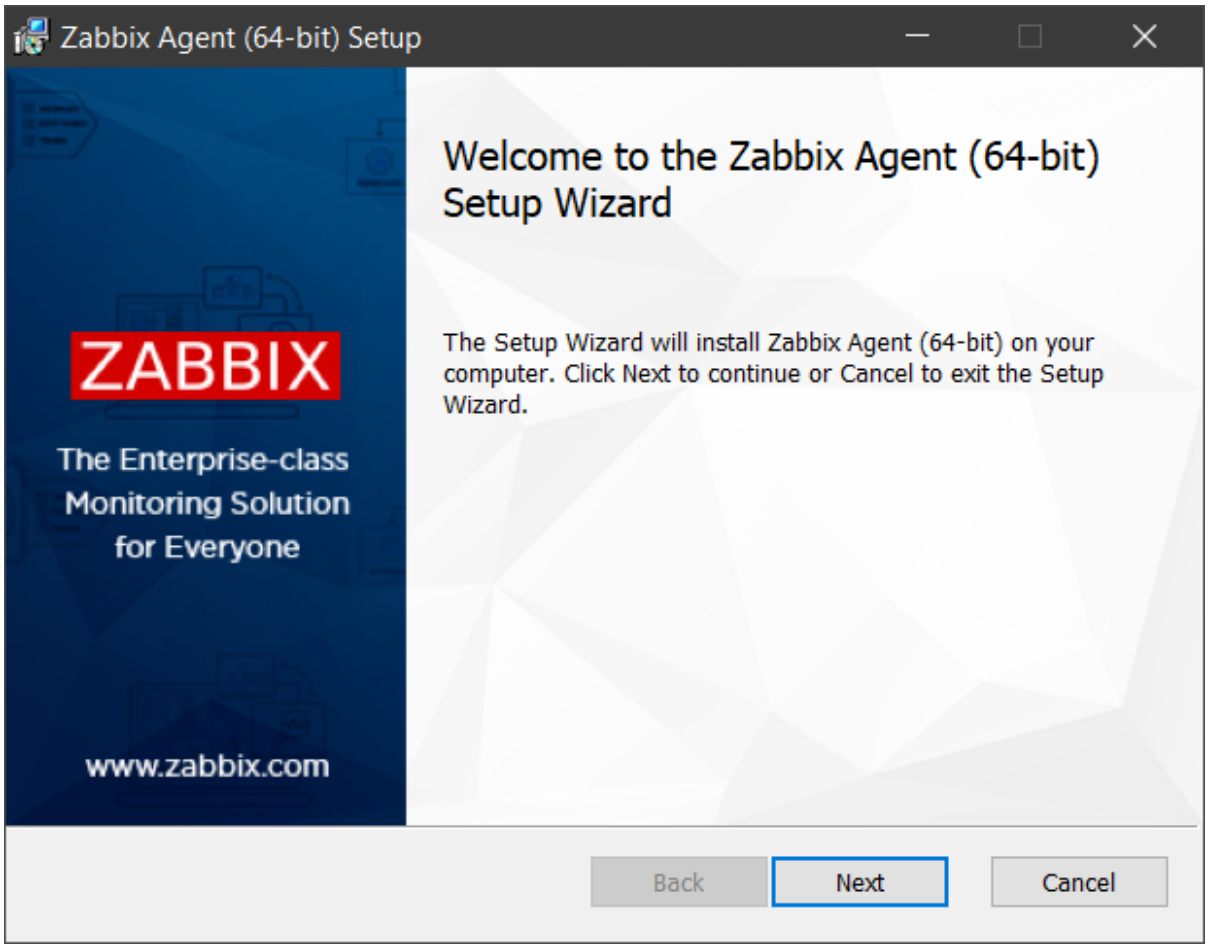
Embora a instalação usando pacotes MSI seja totalmente suportada, recomenda-se instalar pelo menos o Microsoft .NET Framework 2 para um tratamento adequado de erros.

Attention:
Recomenda-se usar o local de instalação padrão fornecido pelo instalador. Usar um local personalizado sem as permissões necessárias pode comprometer a segurança da instalação.

Instalação pelo Assistente de Configuração

As etapas de instalação a seguir se aplicam tanto ao Zabbix agent quanto ao Zabbix agent 2.

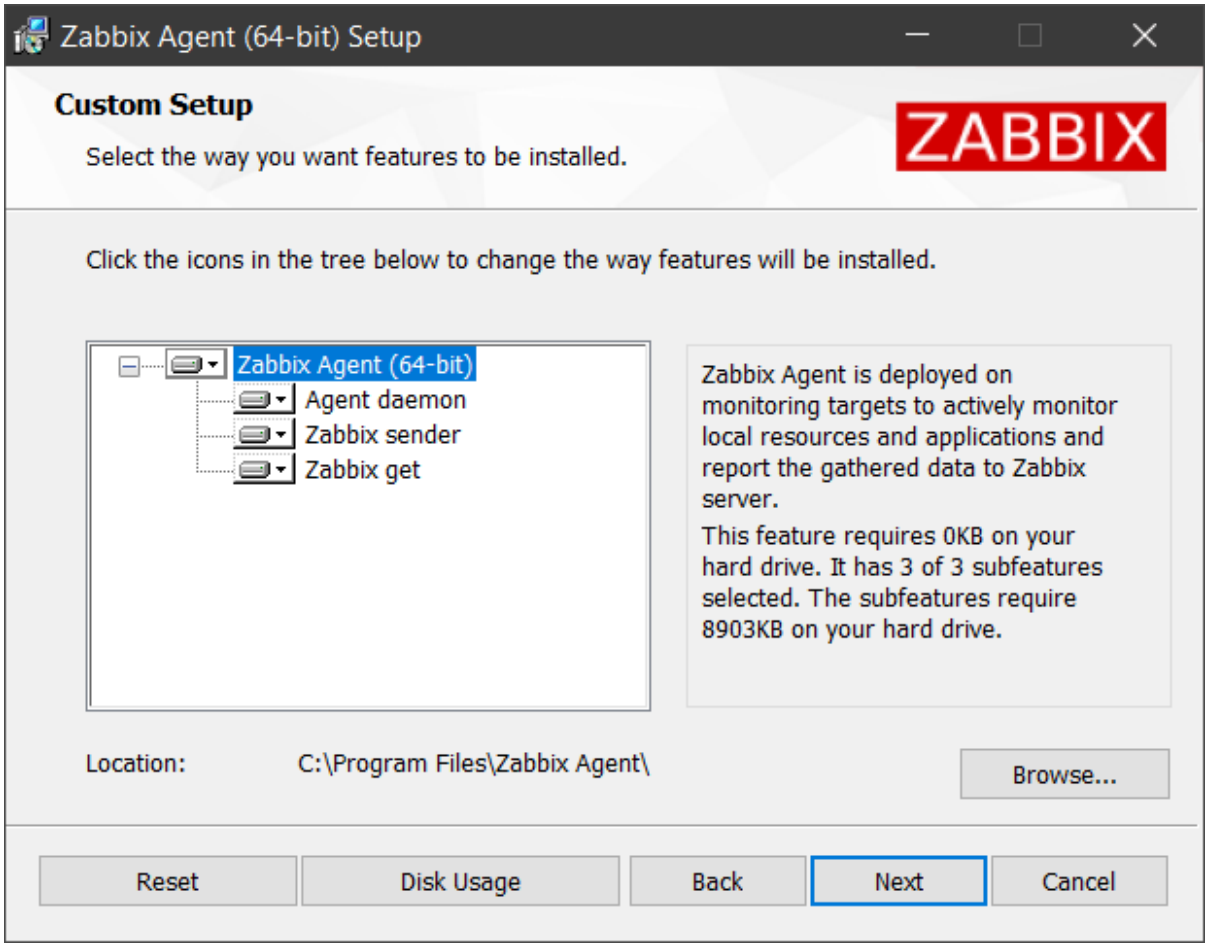
1. Clique duas vezes no arquivo MSI baixado para iniciar a instalação:



2. Aceite o Contrato de Licença do Usuário Final:

3. Selecione os componentes do Zabbix (Agent daemon, Zabbix sender, Zabbix get) a serem instalados:

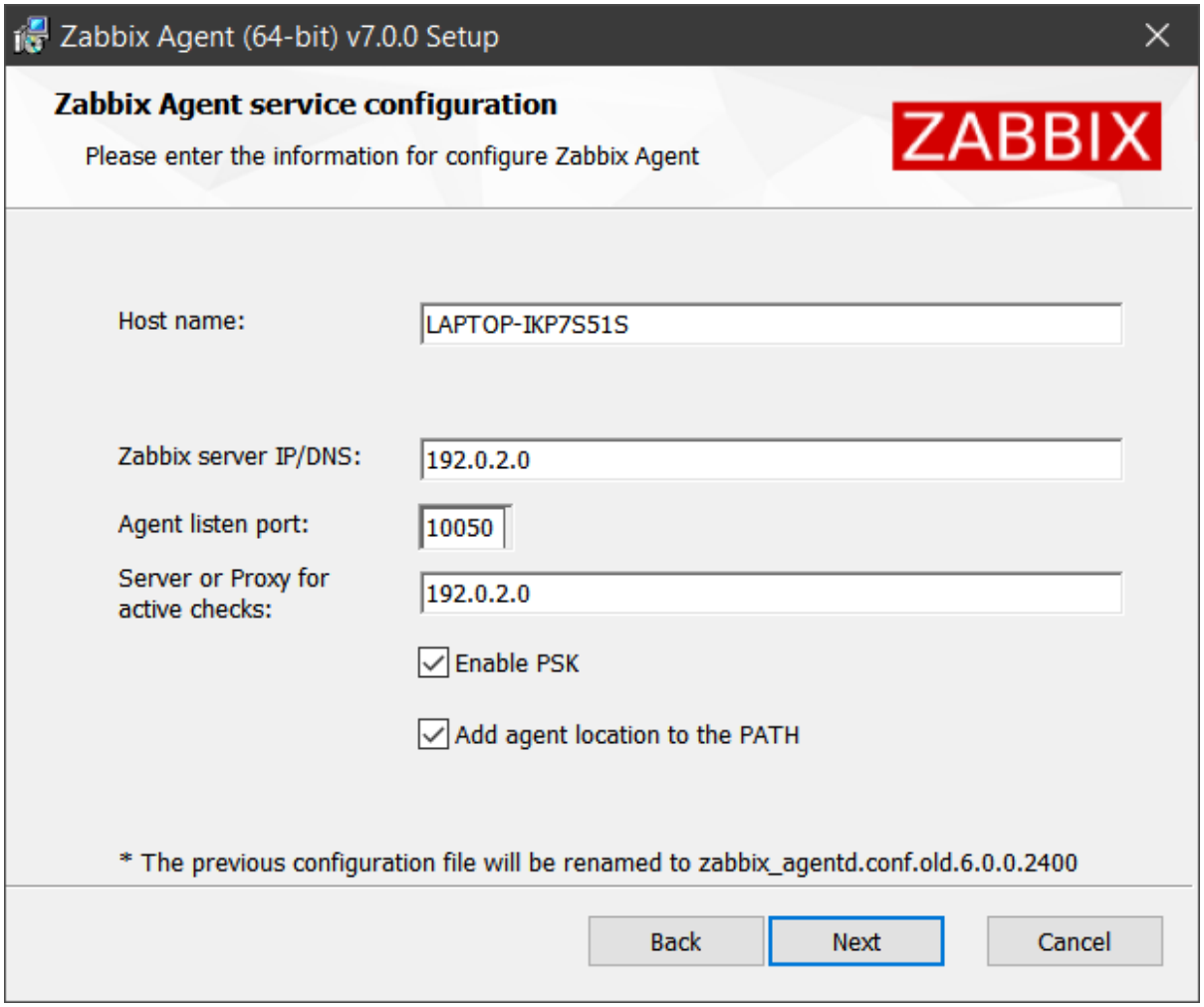
Attention:
Recomenda-se usar o local de instalação padrão fornecido pelo instalador. Usar um local personalizado sem as permissões necessárias pode comprometer a segurança da instalação.



4. Configure os seguintes parâmetros. Seus valores serão definidos no arquivo de configuração do Zabbix agent:

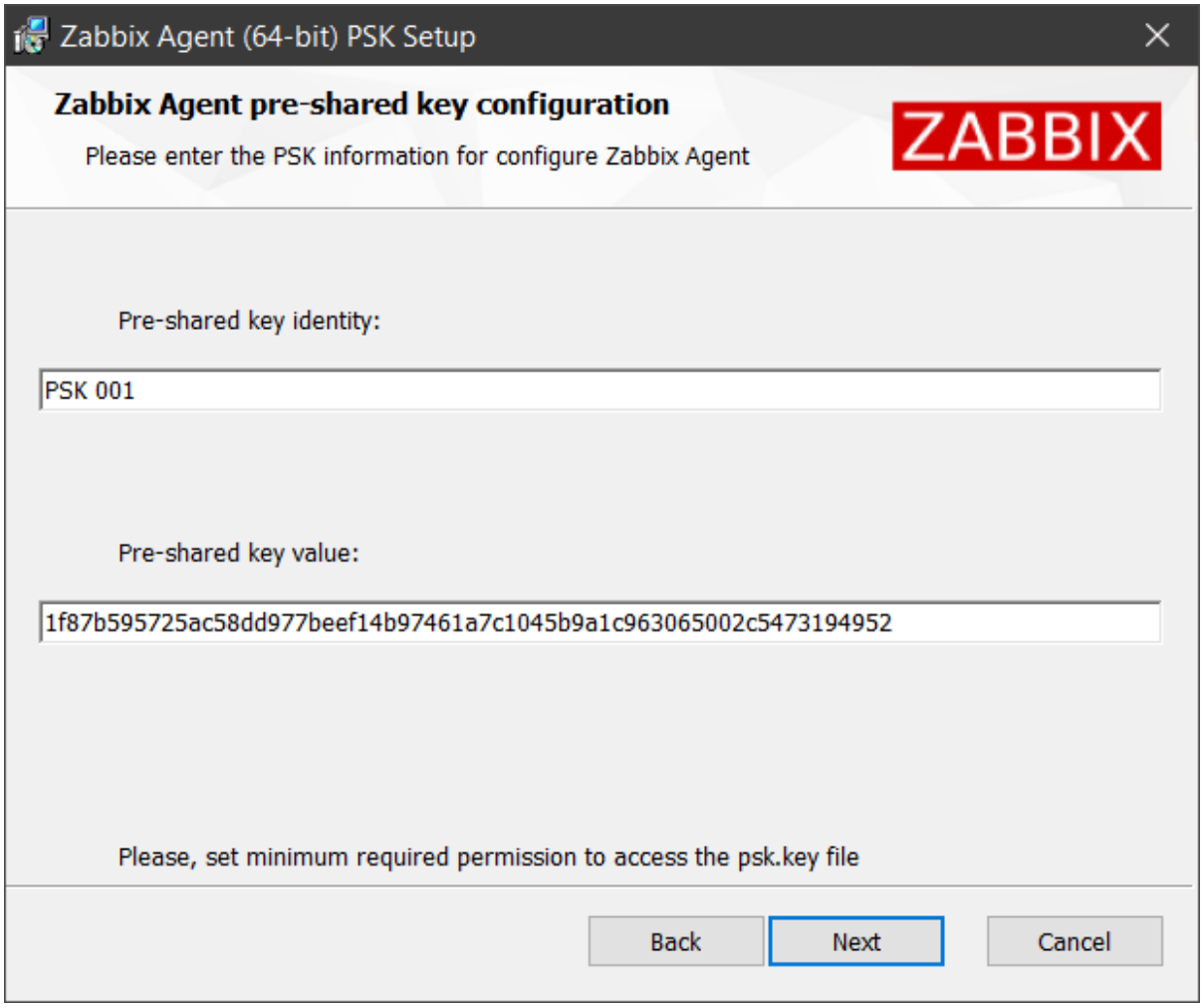
Parâmetro	Descrição
Host name	O nome do host da máquina onde o Zabbix agent está sendo instalado. Define o parâmetro Hostname.
Zabbix server IP/DNS	Uma lista de endereços IP separados por vírgula, opcionalmente na notação CIDR, ou nomes DNS dos Zabbix servers ou Zabbix proxies. Este parâmetro é **obrigatório**. Define o parâmetro Server.
Agent listen port	O agent irá escutar neste porto por conexões do server. Define o parâmetro ListenPort.
Server or Proxy for active checks	O endereço do Zabbix server/proxy ou configuração de cluster para obter verificações ativas. O endereço do server/proxy é um endereço IP ou nome DNS e porta opcional separada por dois pontos. Define o parâmetro ServerActive.
Enable PSK	Marque a caixa de seleção para habilitar o suporte a TLS usando chaves pré-compartilhadas. Define os parâmetros TLSConnect e TLSAccept como `psk`.
Add agent location to the PATH	Marque a caixa de seleção para adicionar o local do Zabbix agent à variável PATH do sistema.

Note:
Se um Zabbix agent existente for detectado, os valores dos parâmetros de seu arquivo de configuração serão exibidos. Além disso, o arquivo de configuração existente será renomeado durante a instalação e um novo arquivo de configuração será criado.



5. Configure os parâmetros PSK se você marcou a caixa *Enable PSK* na etapa anterior. Esses parâmetros também serão definidos no arquivo de configuração do Zabbix agent:

Parâmetro	Descrição
Pre-shared key identity	A string de identidade da chave pré-compartilhada. Define o parâmetro TLSPSKIdentity.
Pre-shared key value	O valor da string da chave pré-compartilhada. Cria o arquivo psk.key contendo a chave e define o parâmetro TLSPSKFile para o local da chave (padrão: `C:\Program Files\Zabbix Agent\psk.key`). É **recomendado** restringir o acesso ao arquivo da chave pré-compartilhada ajustando as configurações de segurança do arquivo para que apenas o Zabbix agent (ou o usuário que executa o agent) possa lê-lo.

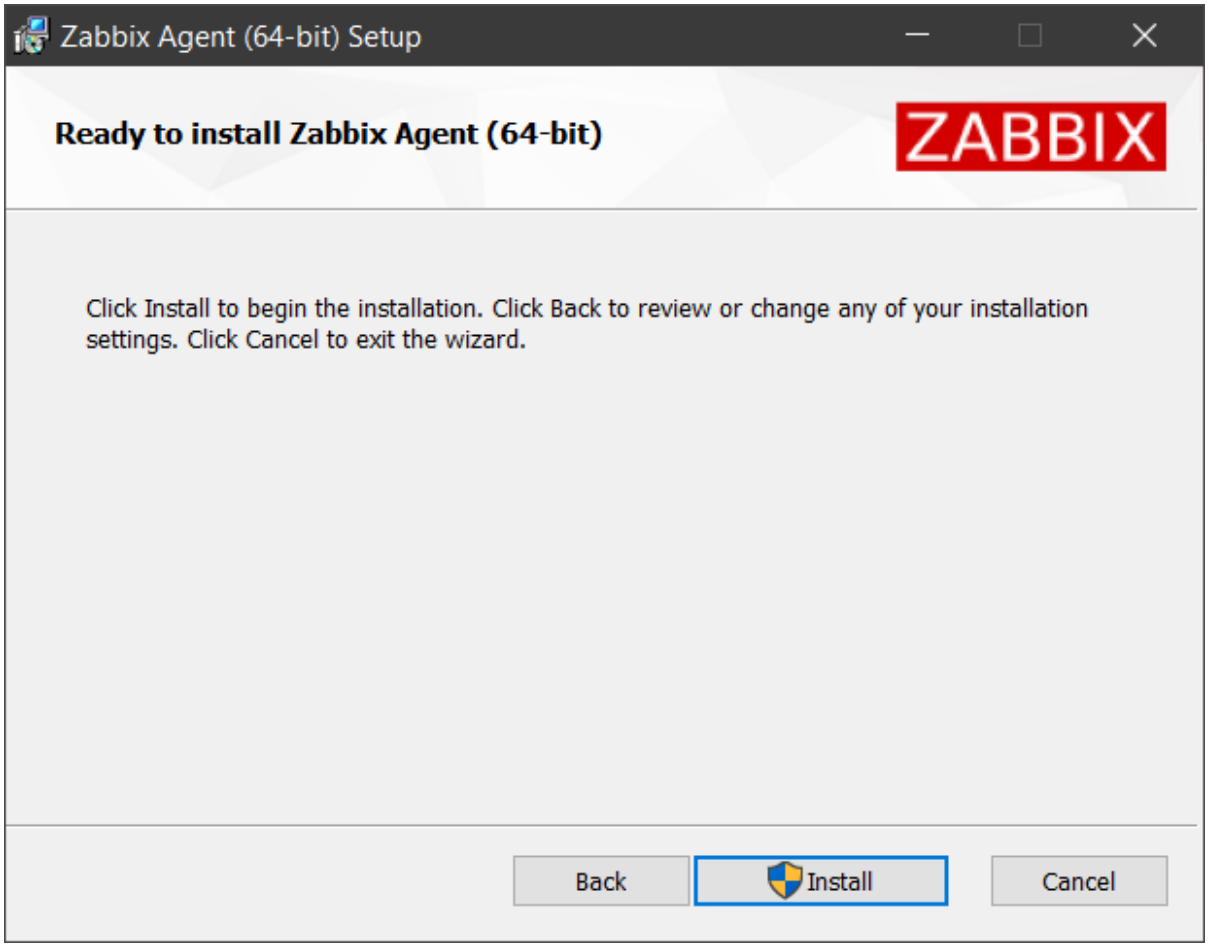


6. Clique em *Install* para iniciar a instalação.

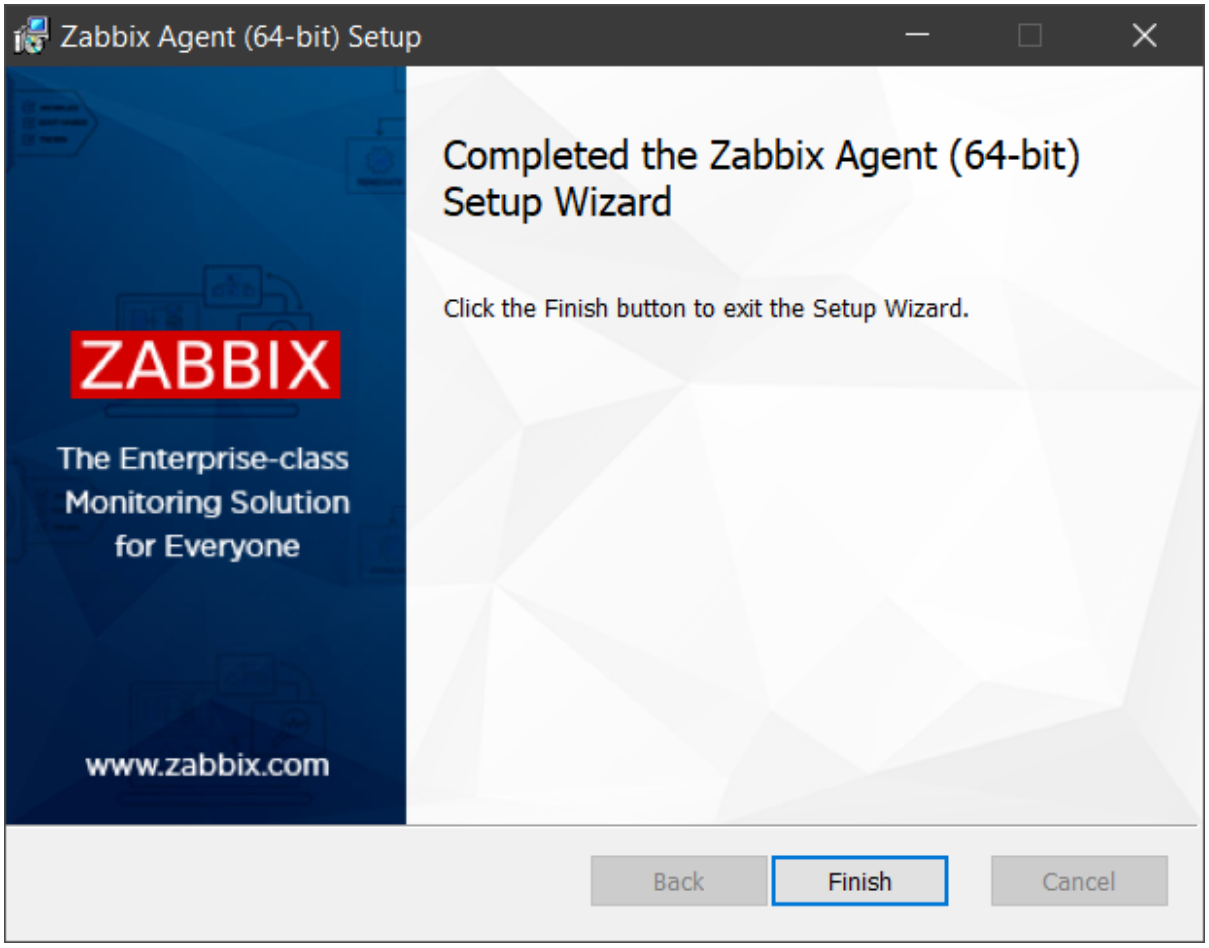
Todos os componentes Zabbix selecionados e o arquivo de configuração do Zabbix agent serão instalados no local especificado (padrão: `C:\Program Files\Zabbix Agent`). O mesmo se aplica ao Zabbix agent 2, exceto que arquivos de configuração adicionais para seus plugins integrados serão instalados na subpasta `zabbix_agent2.d\plugins.d`.

Além disso, o zabbix_agentd.exe (ou zabbix_agent2.exe) será configurado como um serviço do Windows com inicialização automática atrasada (ou inicialização automática em versões do Windows anteriores ao Windows Vista/Server 2008).

Se uma versão diferente do Zabbix agent estiver em execução durante a instalação, será solicitado que você escolha entre fechar o aplicativo e tentar reiniciá-lo ou deixá-lo aberto, caso em que será necessário reiniciar o computador.



7. Clique no botão *Finish* para sair do Assistente de Configuração.



Instalação pela linha de comando

O agent do Zabbix pode ser instalado pela linha de comando executando o instalador MSI com o msiexec. Por exemplo:

```
msiexec.exe /l*v "C:\package.log" /i "C:\zabbix_agent-7.0.0-windows-amd64-openssl.msi" /qn+ SERVER=192.0.2
```

Este método permite instalações não assistidas e configurações personalizadas usando parâmetros.

Parâmetros suportados

Os pacotes de instalação MSI do Zabbix agent suportam os seguintes parâmetros tanto para o Zabbix agent quanto para o Zabbix agent 2.

Note:
Os parâmetros do Zabbix agent/agent2 são definidos no arquivo de configuração durante a instalação. Clique no nome do parâmetro para ver sua descrição detalhada e exemplos de configuração na página Zabbix agent (Windows). Para o Zabbix agent 2, consulte a página Zabbix agent 2 (Windows).

Parâmetro	Descrição
ADDDEFAULT	Uma lista de componentes separados por vírgula para instalar em sua configuração padrão. Para mais informações, consulte propriedade ADDDEFAULT. Valores possíveis: `AgentProgram`, `GetProgram`, `SenderProgram`, `ALL` Exemplo: `ADDDEFAULT=AgentProgram,GetProgram`
ADDLOCAL	Uma lista de componentes separados por vírgula para instalar localmente. Para mais informações, consulte propriedade ADDLOCAL. Valores possíveis: `AgentProgram`, `GetProgram`, `SenderProgram`, `ALL` Exemplo: `ADDLOCAL=AgentProgram,SenderProgram`
ALLOWDENYKEY	Uma lista de parâmetros AllowKey ou DenyKey separados por ponto e vírgula para restringir verificações do Zabbix agent. Se necessário, use uma barra invertida para escapar o delimitador (`\;`). Define os parâmetros AllowKey e DenyKey no arquivo de configuração do agent. Exemplo: `ALLOWDENYKEY="AllowKey=system.run[more C:\Windows\System32\drivers\etc\hosts\\; echo 'File read complete'];DenyKey=system.run[*]"`
CONF	O caminho completo para um arquivo de configuração de template para o Zabbix agent. Durante a instalação, este arquivo se tornará o arquivo de configuração do agent. O arquivo deve conter pelo menos os parâmetros Server e LogFile. Exemplo: `CONF="C:\full\path\to\example.conf"`
DONOTSTART	Use `DONOTSTART=1` para impedir que o instalador MSI inicie o serviço do Zabbix agent. Suportado em pacotes de instalação MSI desde o Zabbix 7.0.22.
ENABLEPATH	Use `ENABLEPATH=1` para adicionar a localização do Zabbix agent à variável PATH do sistema.
ENABLEPERSISTENTBUFFER	Somente Zabbix agent 2. Habilita o uso de armazenamento persistente local para itens ativos.
HOSTINTERFACE	Um parâmetro opcional que define a interface do host.
HOSTMETADATA	Um parâmetro opcional que define os metadados do host.
HOSTMETADATAITEM	Um parâmetro opcional que define um item usado para obter os metadados do host.
HOSTNAME	Um parâmetro opcional que define o nome do host.
HOSTNAMEITEM	Um parâmetro opcional que define um item usado para obter o nome do host. Suportado em pacotes de instalação MSI desde o Zabbix 7.0.22.
INCLUDE	Uma lista de arquivos individuais separados por ponto e vírgula ou todos os arquivos em um diretório para incluir no arquivo de configuração do Zabbix agent.
INSTALLFOLDER	O caminho completo para uma pasta onde os componentes do Zabbix e o arquivo de configuração do Zabbix agent serão instalados. Para o Zabbix agent 2, arquivos de configuração adicionais para plugins embutidos serão instalados na subpasta `zabbix_agent2.d\plugins.d`. Exemplo: `INSTALLFOLDER="C:\Program Files\Zabbix Agent"`
LISTENIP	Uma lista de endereços IP separados por vírgula nos quais o agent deve escutar.
LISTENPORT	O agent irá escutar nesta porta por conexões do server.
LOGFILE	O nome do arquivo de log do Zabbix agent.
LOGTYPE	O tipo de saída do log.
NONMSICONFNAME	O caminho completo para um arquivo de configuração personalizado para o Zabbix agent. Durante a instalação, quaisquer parâmetros de configuração válidos do agent presentes neste arquivo (limitados aos listados nesta tabela) serão gravados no novo arquivo de configuração do agent. O arquivo deve conter pelo menos o parâmetro Server. Exemplo: `NONMSICONFNAME="C:\full\path\to\example.conf"`
PERSISTENTBUFFERFILE	Somente Zabbix agent 2. O arquivo onde o Zabbix agent 2 deve manter o banco de dados SQLite.

Parâmetro	Descrição
PERSISTENTBUFFERPERIOD	Somente Zabbix agent 2. O período de tempo pelo qual os dados devem ser armazenados quando não houver conexão com o server ou proxy.
SERVER	Uma lista de endereços IP separados por vírgula, opcionalmente em notação CIDR, ou nomes DNS de Zabbix servers ou Zabbix proxies. Este parâmetro é **obrigatório**, exceto quando STARTAGENTS está definido como 0.
SERVERACTIVE	O endereço do Zabbix server/proxy ou configuração de cluster para obter verificações ativas.
SKIP	Use `SKIP=fw` para impedir que o instalador MSI adicione uma regra de exceção do Firewall do Windows para o Zabbix agent.
SOURCEIP	O endereço IP de origem para conexões de saída para o Zabbix server ou Zabbix proxy, ou para fazer conexões ao executar alguns itens (web.page.get, net.tcp.port, etc.). Suportado em pacotes de instalação MSI desde o Zabbix 7.0.22.
STARTAGENTS	O número de instâncias pré-forked do zabbix_agentd que processam verificações passivas. Se definido como 0, as verificações passivas são desabilitadas e o agent não escutará em nenhuma porta TCP. Suportado em pacotes de instalação MSI desde o Zabbix 7.0.22.
STARTUPTYPE	Tipo de inicialização do serviço do Zabbix agent. Valores possíveis: **automatic** - inicia o serviço automaticamente na inicialização do Windows; **delayed** - *(padrão)* atrasa o início do serviço após a conclusão da inicialização dos serviços iniciados automaticamente (disponível no Windows Vista/Server 2008 e versões posteriores); **manual** - inicia o serviço manualmente (por um usuário ou aplicativo); **disabled** - desabilita o serviço para que ele não possa ser iniciado por um usuário ou aplicativo. Exemplo: `STARTUPTYPE=disabled`
STATUSPORT	Somente Zabbix agent 2. Se definido, o agent irá escutar nesta porta por solicitações de status HTTP (http://localhost:<port>/status).
TIMEOUT	Especifica quanto tempo esperar (em segundos) para estabelecer conexão e trocar dados com o Zabbix proxy ou server.
TLSACCEPT	As conexões de entrada a serem aceitas (usado para verificações passivas). Se definido como `psk`, então TLSCONNECT também será definido como `psk` (a menos que especificado de outra forma).
TLSCAFILE	O caminho completo de um arquivo contendo os certificados CA(s) de nível superior para verificação do certificado do peer.
TLSCERTFILE	O caminho completo de um arquivo contendo o certificado do agent ou cadeia de certificados.
TLSCONNECT	Como o agent deve se conectar ao Zabbix server ou proxy (usado para verificações ativas). Se definido como `psk`, então TLSACCEPT também será definido como `psk` (a menos que especificado de outra forma).
TLSCRLFILE	O caminho completo de um arquivo contendo certificados revogados.
TLSKEYFILE	O caminho completo de um arquivo contendo a chave privada do Zabbix agent.
TLSPSKFILE	O caminho completo de um arquivo contendo a chave pré-compartilhada do Zabbix agent. Se ambos TLSPSKFILE e TLSPSKVALUE estiverem definidos, o valor de TLSPSKVALUE será gravado no arquivo especificado em TLSPSKFILE. É **recomendado** restringir o acesso ao arquivo da chave pré-compartilhada ajustando as configurações de segurança do arquivo para que apenas o Zabbix agent (ou o usuário que executa o agent) possa lê-lo.
TLSPSKIDENTITY	A string de identidade da chave pré-compartilhada.
TLSPSKVALUE	O valor da string da chave pré-compartilhada. Se ambos TLSPSKFILE e TLSPSKVALUE estiverem definidos, o valor de TLSPSKVALUE será gravado no arquivo especificado em TLSPSKFILE. Exemplo: `TLSPSKVALUE=1f87b595725ac58dd977beef14b97461a7c1045b9a1c963065002c5473194952`
TLSSERVERCERTISSUER	O emissor do certificado do server (proxy) permitido.
TLSSERVERCERTSUBJECT	O assunto do certificado do server (proxy) permitido.
UNSAFEUSERPARAMETERS	Permite que todos os caracteres sejam passados como argumentos para parâmetros definidos pelo usuário. Suportado em pacotes de instalação MSI desde o Zabbix 7.0.22.

Exemplos

O exemplo a seguir instala o agent Zabbix com configuração personalizada. Também habilita o suporte a TLS usando chaves pré-compartilhadas.

```
mkdir "C:\Program Files\Zabbix Agent" 2>nul
msiexec.exe /l*v "C:\package.log" /i "C:\zabbix_agent-7.0.0-windows-amd64-openssl.msi" /qn+^
 SERVER=192.0.2.0^
 INSTALLFOLDER="C:\Program Files\Zabbix Agent"^
```

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

O próximo exemplo instala uma versão mais recente do agent Zabbix e usa um **arquivo de configuração de template** (`CONF="C:\agent-template.conf"`). Durante a instalação, este arquivo se tornará o arquivo de configuração do agent. Para herdar parâmetros do arquivo de configuração antigo, use marcadores de parâmetro (por exemplo, `[AllowDenyKey]`).

```
msiexec.exe /l*v "C:\package.log" /i "C:\zabbix_agent-7.0.1-windows-amd64-openssl.msi" /qn+ CONF="C:\agent

#### Exemplo de agent-template.conf:
LogFile=[LogFile]
[AllowDenyKey]
Server=192.0.2.8
Hostname=DESKTOP-X9F4A2B
[Include]
[TLSConnect]
[TLSAccept]
[TLSPSKIdentity]
[TLSPSKFile]
```

Alternativamente, você pode usar um **arquivo de configuração personalizado** (`NONMSICONFNAME="C:\agent-custom.conf"`). Durante a instalação, quaisquer parâmetros de configuração válidos do agent presentes neste arquivo (limitados aos listados na tabela acima) serão gravados no novo arquivo de configuração do agent criado. Para manter a configuração existente do agent, defina os parâmetros a serem preservados.

```
msiexec.exe /l*v "C:\package.log" /i "C:\zabbix_agent-7.0.1-windows-amd64-openssl.msi" /qn+ NONMSICONFNAME

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

Plugins carregáveis do Zabbix agent 2

Plugins carregáveis do Zabbix agent 2 podem ser instalados no Windows usando pacotes de instalação MSI de 64 bits, disponíveis para download. O plugin Ember+ está incluído nesses pacotes desde o Zabbix 7.0.10.

Os requisitos mínimos de SO para instalação via MSI são Windows 10 (64 bits) ou Windows Server 2016.

Assim como o Zabbix agent/agent2, os plugins carregáveis podem ser instalados usando o Assistente de Instalação ou a linha de comando.

> **Attention:**
> Antes de instalar um plugin, verifique o arquivo README do mesmo. Ele pode conter requisitos específicos e instruções de instalação.

Instalação pelo Assistente de Configuração

1. Clique duas vezes no arquivo MSI baixado para iniciar a instalação.
2. Aceite o Contrato de Licença do Usuário Final.
3. Selecione os plugins carregáveis do Zabbix agent 2 a serem instalados.

> **Attention:**
> Recomenda-se usar o local de instalação padrão fornecido pelo instalador. Usar um local personalizado sem as permissões necessárias pode comprometer a segurança da instalação.

4. Clique em *Instalar* para iniciar a instalação.

Todos os plugins carregáveis do Zabbix agent 2 selecionados serão instalados no local especificado (padrão: `C:\Program Files\Zabbix Agent 2`), com seus arquivos de configuração instalados na subpasta `zabbix_agent2.d`.

5. Clique no botão *Concluir* para sair do Assistente de Configuração.

Instalação pela linha de comando

Os plugins carregáveis do agent 2 do Zabbix podem ser instalados pela linha de comando executando o instalador MSI com o msiexec. Por exemplo:

```
msiexec.exe /l*v "C:\package.log" /i "C:\zabbix_agent2_plugins-7.0.0-windows-amd64.msi" /qn+
```

Os pacotes do instalador MSI dos plugins carregáveis do agent 2 do Zabbix suportam os seguintes parâmetros.

Parâmetro	Descrição
ADDDEFAULT	Uma lista de componentes separados por vírgula para instalar em sua configuração padrão. Para mais informações, veja propriedade ADDDEFAULT. Valores possíveis: `ALL`, `EmberplusPlugin`, `MongodbPlugin`, `MssqlPlugin`, `PostgresqlPlugin` Exemplo: `ADDDEFAULT=MongodbPlugin,PostgresqlPlugin`
ADDLOCAL	Uma lista de componentes separados por vírgula para instalar localmente. Para mais informações, veja propriedade ADDLOCAL. Valores possíveis: `ALL`, `EmberplusPlugin`, `MongodbPlugin`, `MssqlPlugin`, `PostgresqlPlugin` Exemplo: `ADDLOCAL=MongodbPlugin,MssqlPlugin`
INSTALLFOLDER	O caminho completo para uma pasta onde os componentes do Zabbix serão instalados, com seus arquivos de configuração instalados na subpasta `zabbix_agent2.d`. Exemplo: `INSTALLFOLDER="C:\Program Files\Zabbix Agent 2"`

2 Instalação do agent do macOS a partir do PKG

Visão geral

O agent do Zabbix pode ser instalado no macOS usando pacotes de instalação PKG, disponíveis para download.

Os pacotes do agent do Zabbix estão disponíveis com ou sem criptografia.

Instalando o agent

O agent pode ser instalado usando a interface gráfica do usuário ou a partir da linha de comando, por exemplo:

```
sudo installer -pkg zabbix_agent-7.0.0-macos-arm64-openssl.pkg -target /
```

Certifique-se de usar a versão correta do pacote Zabbix no comando. Ela deve corresponder ao nome do pacote baixado.

Executando o agent

O agent será iniciado automaticamente após a instalação ou reinicialização.

Você pode editar o arquivo de configuração em `/usr/local/etc/zabbix/zabbix_agentd.conf` se necessário.

Para iniciar o agent manualmente, execute:

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

Para parar o agent manualmente:

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

Durante a atualização, o arquivo de configuração existente não é sobrescrito. Em vez disso, um novo arquivo `zabbix_agentd.conf.NEW` é criado para ser usado na revisão e atualização do arquivo de configuração existente, se necessário. Lembre-se de reiniciar o agent após alterações manuais no arquivo de configuração.

Solucionando problemas e removendo o agent

Esta seção lista alguns comandos úteis que podem ser usados para solucionar problemas e remover a instalação do agent do Zabbix.

Veja se o agent do Zabbix está em execução:

```
ps aux | grep zabbix_agentd
```

Veja se o agent do Zabbix foi instalado a partir de pacotes:

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

Veja os arquivos que foram instalados a partir do pacote do instalador (observe que o / inicial não é exibido nesta visualização):

```
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
```

Pare o agent do Zabbix se ele foi iniciado com `launchctl`:

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

Remova os arquivos (incluindo configuração e logs) que foram instalados com o pacote do instalador:

```
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
```

Esqueça que o agent do Zabbix foi instalado:

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

3 Lançamentos instáveis

Visão geral

As instruções abaixo são para habilitar os repositórios de versões instáveis do Zabbix (desabilitados por padrão) usados para candidatos a lançamento de versões menores do Zabbix.

Primeiro, instale ou atualize para o pacote zabbix-release mais recente. Para habilitar os pacotes rc em seu sistema, faça o seguinte:

Red Hat Enterprise Linux

Abra o arquivo `/etc/yum.repos.d/zabbix.repo` e defina enabled=1 para o repositório `zabbix-unstable`.

```
[zabbix-unstable]
name=Zabbix Official Repository (unstable) - $basearch
baseurl=https://repo.zabbix.com/zabbix/6.5/rhel/8/$basearch/
enabled=1
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-ZABBIX-A14FE591
```

Debian/Ubuntu

Abra o arquivo `/etc/apt/sources.list.d/zabbix.list` e descomente o repositório "Zabbix unstable repository".

```
#### Zabbix unstable repository
deb https://repo.zabbix.com/zabbix/6.5/debian bullseye main
deb-src https://repo.zabbix.com/zabbix/6.5/debian bullseye main
```

SUSE

Abra o arquivo `/etc/zypp/repos.d/zabbix.repo` e defina enable=1 para o repositório `zabbix-unstable`.

```
[zabbix-unstable]
name=Zabbix Official Repository
type=rpm-md
baseurl=https://repo.zabbix.com/zabbix/6.5/sles/15/x86_64/
gpgcheck=1
gpgkey=https://repo.zabbix.com/zabbix/6.5/sles/15/x86_64/repodata/repomd.xml.key
```

```
enabled=1
update=1
```

5 Instalação a partir de containers

Visão geral Esta seção descreve como implantar o Zabbix com Docker ou Docker Compose.

O Zabbix fornece oficialmente:

- Imagens Docker separadas para cada componente do Zabbix para serem executadas como containers portáteis e autossuficientes.
- Arquivos Compose para definir e executar componentes Zabbix de múltiplos containers no Docker.

Attention:
Desde o Zabbix 6.0, triggers determinísticos precisam ser criados durante a instalação. Se o log binário estiver habilitado para MySQL/MariaDB, isso requer privilégios de superusuário ou a configuração da variável/parâmetro de configuração *log_bin_trust_function_creators = 1*. Veja Scripts de criação de banco de dados para instruções de como definir a variável.
 Observe que, se executar a partir de um console, a variável será definida apenas temporariamente e será descartada quando o Docker for reiniciado. Neste caso, mantenha seu serviço SQL em execução, pare apenas o serviço zabbix-server executando 'docker compose down zabbix-server' e depois 'docker compose up -d zabbix-server'.
 Alternativamente, você pode definir esta variável no arquivo de configuração.

Arquivos de origem

Os arquivos de origem do Docker estão armazenados no repositório oficial do Zabbix no GitHub, onde você pode acompanhar as últimas alterações nos arquivos ou fazer um fork do projeto para criar suas próprias imagens.

Docker O Zabbix fornece imagens baseadas em uma variedade de imagens de sistemas operacionais. Para obter a lista de imagens de sistemas operacionais base suportadas para um componente específico do Zabbix, consulte a descrição do componente no Docker Hub. Todas as imagens do Zabbix são configuradas para reconstruir as imagens mais recentes se as imagens base forem atualizadas.

Instalação

Para obter a imagem do componente Zabbix, execute:

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

Substitua `zabbix/zabbix-server-mysql` pelo nome do repositório docker necessário.

Este comando irá baixar a versão estável mais recente do componente Zabbix baseada no sistema operacional Alpine Linux. Você pode adicionar tags ao nome do repositório para obter uma imagem baseada em outro sistema operacional ou de uma versão principal ou secundária específica do Zabbix.

Os seguintes repositórios estão disponíveis no Docker Hub:

Componente		Repositório Docker
Zabbix agent		zabbix/zabbix-agent
Zabbix server		
	com suporte a MySQL	zabbix/zabbix-server-mysql
	com suporte a PostgreSQL	zabbix/zabbix-server-pgsql
Zabbix web interface		
	baseado no servidor web Apache2 com suporte a MySQL	zabbix/zabbix-web-apache-mysql
	baseado no servidor web Apache2 com suporte a PostgreSQL	zabbix/zabbix-web-apache-pgsql
	baseado no servidor web Nginx com suporte a MySQL	zabbix/zabbix-web-nginx-mysql

Componente		Repositório Docker
	baseado no servidor web Nginx com suporte a PostgreSQL	zabbix/zabbix-web-nginx-pgsql
Zabbix proxy		
	com suporte a SQLite3	zabbix/zabbix-proxy-sqlite3
	com suporte a MySQL	zabbix/zabbix-proxy-mysql
Zabbix Java gate-way		zabbix/zabbix-java-gateway

Note:
O suporte a SNMP trap é fornecido em um repositório separado zabbix/zabbix-snmptraps. Ele pode ser vinculado ao Zabbix server e ao Zabbix proxy.

Tags

As imagens oficiais dos componentes Zabbix podem conter as seguintes tags:

Tag	Descrição	Exemplo
latest	A versão estável mais recente de um componente Zabbix baseada na imagem Alpine Linux.	zabbix-agent:latest
<OS>-trunk	O build noturno mais recente da versão Zabbix que está sendo desenvolvida atualmente em um sistema operacional específico. **<OS>** - o sistema operacional base. Valores suportados: *alpine* - Alpine Linux; *ltsc2019* - Windows 10 LTSC 2019 (apenas agent); *ol* - Oracle Linux; *ltsc2022* - Windows 11 LTSC 2022 (apenas agent); *ubuntu* - Ubuntu	zabbix-agent:ubuntu-trunk
<OS>-latest	A versão estável mais recente de um componente Zabbix em um sistema operacional específico. **<OS>** - o sistema operacional base. Valores suportados: *alpine* - Alpine Linux; *ltsc2019* - Windows 10 LTSC 2019 (apenas agent); *ol* - Oracle Linux; *ltsc2022* - Windows 11 LTSC 2022 (apenas agent); *ubuntu* - Ubuntu	zabbix-agent:ol-latest
<OS>-X.X-latest	A versão menor mais recente de um componente Zabbix de uma versão principal e sistema operacional específicos. **<OS>** - o sistema operacional base. Valores suportados: *alpine* - Alpine Linux; *ltsc2019* - Windows 10 LTSC 2019 (apenas agent); *ol* - Oracle Linux; *ltsc2022* - Windows 11 LTSC 2022 (apenas agent); *ubuntu* - Ubuntu **X.X** - a versão principal do Zabbix (ex.: *6.0*, *7.0*, *7.2*).	zabbix-agent:alpine-7.0-latest

Tag	Descrição	Exemplo
<OS>-X.X.*	A versão menor específica de um componente Zabbix de uma versão principal e sistema operacional específicos. **<OS>** - o sistema operacional base. Valores suportados: *alpine* - Alpine Linux; *ltsc2019* - Windows 10 LTSC 2019 (apenas agent); *ol* - Oracle Linux; *ltsc2022* - Windows 11 LTSC 2022 (apenas agent); *ubuntu* - Ubuntu **X.X** - a versão principal do Zabbix (ex.: *6.0*, *7.0*, *7.2*). * - a versão menor do Zabbix	zabbix-agent:alpine-7.0.1

Configuração inicial

Após baixar as imagens, inicie os containers executando o comando `docker run` seguido de argumentos adicionais para especificar as variáveis de ambiente e/ou pontos de montagem necessários. Alguns exemplos de configuração são fornecidos abaixo.

Attention:
O Zabbix não deve ser executado como PID1/como um processo init em containers nas versões 7.0.0-7.0.2 do Zabbix.

Note:
Para habilitar a comunicação entre os componentes do Zabbix, algumas portas, como 10051/TCP para o Zabbix server (trapper), 10050/TCP para o Zabbix agent, 162/UDP para traps SNMP e 80/TCP para a interface web do Zabbix, serão expostas para a máquina host. A lista completa das portas padrão usadas pelos componentes do Zabbix está disponível na página de Requisitos. Para o Zabbix server e agent, a porta padrão pode ser alterada definindo a variável de ambiente ZBX_LISTENPORT environment variable.

Variáveis de ambiente

Todas as imagens dos componentes do Zabbix fornecem variáveis de ambiente para controlar a configuração. As variáveis de ambiente suportadas estão listadas no repositório do componente.

Essas variáveis de ambiente são opções dos arquivos de configuração do Zabbix, mas com um método de nomenclatura diferente. Por exemplo, `ZBX_LOGSLOWQUERIES` é igual a `LogSlowQueries` dos arquivos de configuração do Zabbix server ou do Zabbix proxy.

Attention:
Algumas opções de configuração (por exemplo, `PIDFile` e `LogType`) não podem ser alteradas.

As seguintes variáveis de ambiente são específicas para componentes Docker e não existem nos arquivos de configuração do Zabbix:

Variável	Componentes	Valor padrão	Descrição
`DB_SERVER_HOST`	Server Proxy Web interface	`mysql-server` para MYSQL `postgres-server` para PostgreSQL	IP ou nome DNS do servidor MySQL ou PostgreSQL.
`DB_SERVER_PORT`	Server Proxy Web interface	3306 para MYSQL 5432 para PostgreSQL	Porta do servidor MySQL ou PostgreSQL.
`MYSQL_USER`	Server Proxy Web-interface	`zabbix`	Usuário do banco de dados MySQL.
`MYSQL_PASSWORD`	Server Proxy Web interface	`zabbix`	Senha do banco de dados MySQL.

Variável	Componentes	Valor padrão	Descrição
MYSQL_DATABASE	Server Proxy Web interface	zabbix para Zabbix server zabbix_proxy para Zabbix proxy	Nome do banco de dados Zabbix.
POSTGRES_USER	Server Web interface	zabbix	Usuário do banco de dados PostgreSQL.
POSTGRES_PASSWORD	Server Web interface	zabbix	Senha do banco de dados PostgreSQL.
POSTGRES_DB	Server Web interface	zabbix para Zabbix server zabbix_proxy para Zabbix proxy	Nome do banco de dados Zabbix.
PHP_TZ	Web-interface	Europe/Riga	Fuso horário no formato PHP. A lista completa de fusos horários suportados está disponível em php.net.
ZBX_SERVER_NAME	Web interface	Zabbix Docker	Nome da instalação do Zabbix visível abaixo do logotipo do Zabbix no menu vertical da interface web.
ZBX_JAVAGATEWAY_ENABLE	Server Proxy	false	Habilita a comunicação com o Zabbix Java gateway para coletar verificações relacionadas ao Java.
ZBX_ENABLE_SNMP_TRAPS	Server Proxy	false	Habilita o recurso de SNMP trap. Requer uma instância **zabbix-snmptraps** e o volume compartilhado */var/lib/zabbix/snmptraps* para o Zabbix server ou Zabbix proxy.

Volumes

As imagens permitem montar volumes usando os seguintes pontos de montagem:

Volume	Descrição
Zabbix agent	
/etc/zabbix/zabbix_agentd	O volume permite incluir arquivos **.conf* e estender o Zabbix agent usando o recurso UserParameter
/var/lib/zabbix/modules	O volume permite carregar módulos adicionais e estender o Zabbix agent usando o recurso LoadModule
/var/lib/zabbix/enc	O volume é usado para armazenar arquivos relacionados a TLS. Os nomes desses arquivos são especificados usando as variáveis de ambiente ZBX_TLSCAFILE, ZBX_TLSCRLFILE, ZBX_TLSKEY_FILE e ZBX_TLSPSKFILE
Zabbix server	
/usr/lib/zabbix/alertscripts	O volume é usado para scripts de alerta personalizados. É o parâmetro AlertScriptsPath em zabbix_server.conf
/usr/lib/zabbix/externalscripts	O volume é usado por external checks. É o parâmetro ExternalScripts em zabbix_server.conf
/var/lib/zabbix/modules	O volume permite carregar módulos adicionais e estender o Zabbix server usando o recurso LoadModule
/var/lib/zabbix/enc	O volume é usado para armazenar arquivos relacionados a TLS. Os nomes desses arquivos são especificados usando as variáveis de ambiente ZBX_TLSCAFILE, ZBX_TLSCRLFILE, ZBX_TLSKEY_FILE e ZBX_TLSPSKFILE
/var/lib/zabbix/ssl/certs	O volume é usado como local dos arquivos de certificado de cliente SSL para autenticação de cliente. É o parâmetro SSLCertLocation em zabbix_server.conf
/var/lib/zabbix/ssl/keys	O volume é usado como local dos arquivos de chave privada SSL para autenticação de cliente. É o parâmetro SSLKeyLocation em zabbix_server.conf
/var/lib/zabbix/ssl/ssl_ca	O volume é usado como local dos arquivos da autoridade certificadora (CA) para verificação do certificado do servidor SSL. É o parâmetro SSLCALocation em zabbix_server.conf
/var/lib/zabbix/snmptraps	O volume é usado como local do arquivo snmptraps.log. Ele pode ser compartilhado pelo container zabbix-snmptraps e herdado usando a opção volumes_from do Docker ao criar uma nova instância do Zabbix server. O recurso de processamento de SNMP trap pode ser habilitado usando o volume compartilhado e alterando a variável de ambiente ZBX_ENABLE_SNMP_TRAPS para 'true'
/var/lib/zabbix/mibs	O volume permite adicionar novos arquivos MIB. Não suporta subdiretórios, todas as MIBs devem ser colocadas em /var/lib/zabbix/mibs

Volume	Descrição
Zabbix proxy	
/usr/lib/zabbix/externalscripts	O volume é usado por external checks. É o parâmetro `ExternalScripts` em zabbix_proxy.conf
/var/lib/zabbix/db_data	O volume permite armazenar arquivos de banco de dados em dispositivos externos. Suportado apenas para Zabbix proxy com SQLite3
/var/lib/zabbix/modules	O volume permite carregar módulos adicionais e estender o Zabbix server usando o recurso LoadModule
/var/lib/zabbix/enc	O volume é usado para armazenar arquivos relacionados a TLS. Os nomes desses arquivos são especificados usando as variáveis de ambiente `ZBX_TLSCAFILE`, `ZBX_TLSCRLFILE`, `ZBX_TLSKEY_FILE` e `ZBX_TLSPSKFILE`
/var/lib/zabbix/ssl/certs	O volume é usado como local dos arquivos de certificado de cliente SSL para autenticação de cliente. É o parâmetro `SSLCertLocation` em zabbix_proxy.conf
/var/lib/zabbix/ssl/keys	O volume é usado como local dos arquivos de chave privada SSL para autenticação de cliente. É o parâmetro `SSLKeyLocation` em zabbix_proxy.conf
/var/lib/zabbix/ssl/ssl_ca	O volume é usado como local dos arquivos da autoridade certificadora (CA) para verificação do certificado do servidor SSL. É o parâmetro `SSLCALocation` em zabbix_proxy.conf
/var/lib/zabbix/snmptraps	O volume é usado como local do arquivo snmptraps.log. Ele pode ser compartilhado pelo container zabbix-snmptraps e herdado usando a opção volumes_from do Docker ao criar uma nova instância do Zabbix server. O recurso de processamento de SNMP trap pode ser habilitado usando o volume compartilhado e alterando a variável de ambiente `ZBX_ENABLE_SNMP_TRAPS` para 'true'
/var/lib/zabbix/mibs	O volume permite adicionar novos arquivos MIB. Não suporta subdiretórios, todas as MIBs devem ser colocadas em `/var/lib/zabbix/mibs`
Zabbix web interface based on Apache2 web server	
/etc/ssl/apache2	O volume permite habilitar HTTPS para a interface web do Zabbix. O volume deve conter os dois arquivos `ssl.crt` e `ssl.key` preparados para conexões SSL do Apache2
Zabbix web interface based on Nginx web server	
/etc/ssl/nginx	O volume permite habilitar HTTPS para a interface web do Zabbix. O volume deve conter os dois arquivos `ssl.crt`, `ssl.key` e o arquivo `dhparam.pem` preparados para conexões SSL do Nginx
Zabbix snmptraps	
/var/lib/zabbix/snmptraps	O volume contém o arquivo de log `snmptraps.log` nomeado com os SNMP traps recebidos
/var/lib/zabbix/mibs	O volume permite adicionar novos arquivos MIB. Não suporta subdiretórios, todas as MIBs devem ser colocadas em `/var/lib/zabbix/mibs`

Para informações adicionais, consulte os repositórios oficiais do Zabbix no Docker Hub.

Exemplos

Exemplo 1

O exemplo demonstra como executar o Zabbix server com suporte ao banco de dados MySQL, interface web do Zabbix baseada

no servidor web Nginx e Zabbix Java gateway.

1. Crie uma rede dedicada para os containers dos componentes do Zabbix:

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

2. Inicie uma instância vazia do MySQL server:

> **Attention:**
> Para versões do MySQL 8.4+, `caching_sha2_password` deve ser usado em vez de `mysql_native_password`.

```
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:8.0-oracle \
      --character-set-server=utf8 --collation-server=utf8_bin \
      --default-authentication-plugin=mysql_native_password
```

3. Inicie a instância do Zabbix Java gateway:

```
docker run --name zabbix-java-gateway -t \
      --network=zabbix-net \
      --restart unless-stopped \
      -d zabbix/zabbix-java-gateway:alpine-7.0-latest
```

4. Inicie a instância do Zabbix server e vincule a instância à instância do MySQL server criada:

```
docker run --name zabbix-server-mysql -t \
      -e DB_SERVER_HOST="mysql-server" \
      -e MYSQL_DATABASE="zabbix" \
      -e MYSQL_USER="zabbix" \
      -e MYSQL_PASSWORD="zabbix_pwd" \
      -e MYSQL_ROOT_PASSWORD="root_pwd" \
      -e ZBX_JAVAGATEWAY="zabbix-java-gateway" \
      --network=zabbix-net \
      -p 10051:10051 \
      --restart unless-stopped \
      -d zabbix/zabbix-server-mysql:alpine-7.0-latest
```

5. Inicie a interface web do Zabbix e vincule a instância às instâncias do MySQL server e do Zabbix server criadas:

```
docker run --name zabbix-web-nginx-mysql -t \
      -e ZBX_SERVER_HOST="zabbix-server-mysql" \
      -e DB_SERVER_HOST="mysql-server" \
      -e MYSQL_DATABASE="zabbix" \
      -e MYSQL_USER="zabbix" \
      -e MYSQL_PASSWORD="zabbix_pwd" \
      -e MYSQL_ROOT_PASSWORD="root_pwd" \
      --network=zabbix-net \
      -p 80:8080 \
      --restart unless-stopped \
      -d zabbix/zabbix-web-nginx-mysql:alpine-7.0-latest
```

Exemplo 2

O exemplo demonstra como executar o Zabbix server com suporte ao banco de dados PostgreSQL, interface web do Zabbix baseada no servidor web Nginx e recurso de SNMP trap.

1. Crie uma rede dedicada para os containers dos componentes do Zabbix:

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

2. Inicie uma instância vazia do PostgreSQL server:

```
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. Inicie a instância do Zabbix snmptraps:

```
docker run --name zabbix-snmptraps -t \
      -v /zbx_instance/snmptraps:/var/lib/zabbix/snmptraps:rw \
      --network=zabbix-net \
      -p 162:1162/udp \
      --restart unless-stopped \
      -d zabbix/zabbix-snmptraps:alpine-7.0-latest
```

4. Inicie a instância do Zabbix server e vincule a instância à instância do PostgreSQL server criada:

```
docker run --name zabbix-server-pgsql -t \
      -e DB_SERVER_HOST="postgres-server" \
      -e POSTGRES_USER="zabbix" \
      -e POSTGRES_PASSWORD="zabbix_pwd" \
      -e POSTGRES_DB="zabbix" \
      -e ZBX_ENABLE_SNMP_TRAPS="true" \
      --network=zabbix-net \
      -p 10051:10051 \
      --volumes-from zabbix-snmptraps \
      --restart unless-stopped \
      -d zabbix/zabbix-server-pgsql:alpine-7.0-latest
```

5. Inicie a interface web do Zabbix e vincule a instância às instâncias do PostgreSQL server e do Zabbix server criadas:

```
docker run --name zabbix-web-nginx-pgsql -t \
      -e ZBX_SERVER_HOST="zabbix-server-pgsql" \
      -e DB_SERVER_HOST="postgres-server" \
      -e POSTGRES_USER="zabbix" \
      -e POSTGRES_PASSWORD="zabbix_pwd" \
      -e POSTGRES_DB="zabbix" \
      --network=zabbix-net \
      -p 443:8443 \
      -p 80:8080 \
      -v /etc/ssl/nginx:/etc/ssl/nginx:ro \
      --restart unless-stopped \
      -d zabbix/zabbix-web-nginx-pgsql:alpine-7.0-latest
```

Exemplo 3

O exemplo demonstra como executar o Zabbix server com suporte ao banco de dados MySQL, interface web do Zabbix baseada no servidor web Nginx e Zabbix Java gateway usando `podman` no Red Hat 8.

1. Crie um novo pod com o nome `zabbix` e portas expostas (interface web, Zabbix server trapper):

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

2. (opcional) Inicie o container do Zabbix agent na localização do pod `zabbix`:

```
podman run --name zabbix-agent \
    -e ZBX_SERVER_HOST="127.0.0.1,localhost" \
    --restart=always \
    --pod=zabbix \
    -d registry.connect.redhat.com/zabbix/zabbix-agent-70:latest
```

3. Crie o diretório `./mysql/` no host e inicie o Oracle MySQL server 8.0:

Attention:
Para versões do MySQL 8.4+, `caching_sha2_password` deve ser usado em vez de `mysql_native_password`.

```
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:8.0 \
      --character-set-server=utf8 --collation-server=utf8_bin \
      --default-authentication-plugin=mysql_native_password
```

4. Inicie o container do Zabbix server:

```
podman run --name zabbix-server-mysql -t \
                  -e DB_SERVER_HOST="127.0.0.1" \
                  -e MYSQL_DATABASE="zabbix" \
                  -e MYSQL_USER="zabbix" \
                  -e MYSQL_PASSWORD="zabbix_pwd" \
                  -e MYSQL_ROOT_PASSWORD="root_pwd" \
                  -e ZBX_JAVAGATEWAY="127.0.0.1" \
                  --restart=always \
                  --pod=zabbix \
                  -d registry.connect.redhat.com/zabbix/zabbix-server-mysql-70
```

5. Inicie o container do Zabbix Java Gateway:

```
podman run --name zabbix-java-gateway -t \
      --restart=always \
      --pod=zabbix \
      -d registry.connect.redhat.com/zabbix/zabbix-java-gateway-70
```

6. Inicie o container da interface web do Zabbix:

```
podman run --name zabbix-web-mysql -t \
                  -e ZBX_SERVER_HOST="127.0.0.1" \
                  -e DB_SERVER_HOST="127.0.0.1" \
                  -e MYSQL_DATABASE="zabbix" \
                  -e MYSQL_USER="zabbix" \
                  -e MYSQL_PASSWORD="zabbix_pwd" \
                  -e MYSQL_ROOT_PASSWORD="root_pwd" \
                  --restart=always \
                  --pod=zabbix \
                  -d registry.connect.redhat.com/zabbix/zabbix-web-mysql-70
```

Note:
O pod `zabbix` expõe a porta 80/TCP (HTTP) para a máquina host a partir da porta 8080/TCP do container `zabbix-web-mysql`.

Docker Compose Como alternativa, o Zabbix pode ser instalado usando o plugin Docker Compose. Os arquivos Compose para definir e executar componentes Zabbix multi-contêiner estão disponíveis no repositório oficial do Zabbix Docker no GitHub.

Attention:
Os arquivos compose oficiais do Zabbix suportam a versão 3 do Docker Compose.

Esses arquivos compose são adicionados como exemplos; eles são sobrecarregados. Por exemplo, eles contêm proxies com suporte tanto para MySQL quanto para SQLite3.

Para obter os arquivos Docker compose fornecidos pelo Zabbix, clone o repositório:

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

Altere para a versão necessária:

```
git checkout 7.0
```

Configure os arquivos de configuração e crie e inicie os contêineres:

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

Substitua `docker-compose_v3_alpine_mysql_latest.yaml` no comando acima pelo arquivo de configuração necessário.

As seguintes opções estão disponíveis:

Nome do arquivo	Descrição
`docker-compose_v3_alpine_mysql_latest.yaml`	O arquivo compose executa a versão mais recente dos componentes do Zabbix 7.0 no Alpine Linux com suporte ao banco de dados MySQL.
`docker-compose_v3_alpine_mysql_local.yaml`	O arquivo compose constrói localmente a versão mais recente do Zabbix 7.0 e executa os componentes do Zabbix no Alpine Linux com suporte ao banco de dados MySQL.
`docker-compose_v3_alpine_pgsql_latest.yaml`	O arquivo compose executa a versão mais recente dos componentes do Zabbix 7.0 no Alpine Linux com suporte ao banco de dados PostgreSQL.
`docker-compose_v3_alpine_pgsql_local.yaml`	O arquivo compose constrói localmente a versão mais recente do Zabbix 7.0 e executa os componentes do Zabbix no Alpine Linux com suporte ao banco de dados PostgreSQL.
`docker-compose_v3_ol_mysql_latest.yaml`	O arquivo compose executa a versão mais recente dos componentes do Zabbix 7.0 no Oracle Linux com suporte ao banco de dados MySQL.
`docker-compose_v3_ol_mysql_local.yaml`	O arquivo compose constrói localmente a versão mais recente do Zabbix 7.0 e executa os componentes do Zabbix no Oracle Linux com suporte ao banco de dados MySQL.
`docker-compose_v3_ol_pgsql_latest.yaml`	O arquivo compose executa a versão mais recente dos componentes do Zabbix 7.0 no Oracle Linux com suporte ao banco de dados PostgreSQL.
`docker-compose_v3_ol_pgsql_local.yaml`	O arquivo compose constrói localmente a versão mais recente do Zabbix 7.0 e executa os componentes do Zabbix no Oracle Linux com suporte ao banco de dados PostgreSQL.
`docker-compose_v3_ubuntu_mysql_latest.yaml`	O arquivo compose executa a versão mais recente dos componentes do Zabbix 7.0 no Ubuntu 20.04 com suporte ao banco de dados MySQL.
`docker-compose_v3_ubuntu_mysql_local.yaml`	O arquivo compose constrói localmente a versão mais recente do Zabbix 7.0 e executa os componentes do Zabbix no Ubuntu 20.04 com suporte ao banco de dados MySQL.
`docker-compose_v3_ubuntu_pgsql_latest.yaml`	O arquivo compose executa a versão mais recente dos componentes do Zabbix 7.0 no Ubuntu 20.04 com suporte ao banco de dados PostgreSQL.
`docker-compose_v3_ubuntu_pgsql_local.yaml`	O arquivo compose constrói localmente a versão mais recente do Zabbix 7.0 e executa os componentes do Zabbix no Ubuntu 20.04 com suporte ao banco de dados PostgreSQL.

Armazenamento

Os arquivos Compose são configurados para suportar armazenamento local em uma máquina host. O Docker Compose criará um diretório `zbx_env` na pasta com o arquivo compose quando você executar os componentes do Zabbix usando o arquivo compose. O diretório conterá a mesma estrutura descrita na seção Volumes e o diretório para armazenamento do banco de dados.

Também existem volumes no modo somente leitura para os arquivos `/etc/localtime` e `/etc/timezone`.

Variáveis de ambiente

Os arquivos de variáveis têm a seguinte estrutura de nomenclatura: `.env_<tipo de componente>` e estão localizados no diretório *env_vars* diretório. Veja variáveis de ambiente para detalhes sobre a nomenclatura das variáveis e seleção disponível.

Exemplos

Exemplo 1

```
git checkout 7.0
docker compose -f ./docker-compose_v3_alpine_mysql_latest.yaml up -d
```

O comando irá baixar as imagens mais recentes do Zabbix 7.0 para cada componente do Zabbix e executá-las em modo detach.

> **Attention:**
> Não se esqueça de baixar os arquivos `.env_<tipo de componente>` do repositório oficial do Zabbix no github.com com os arquivos compose.

Exemplo 2

```
git checkout 7.0
docker compose -f ./docker-compose_v3_ubuntu_mysql_local.yaml up -d
```

O comando irá baixar a imagem base Ubuntu 24.04 (noble), então irá construir os componentes do Zabbix 7.0 localmente e executá-los em modo detach.

6 Instalação da interface web

Esta seção fornece instruções passo a passo para instalar a interface web do Zabbix. O frontend do Zabbix é escrito em PHP, portanto, para executá-lo, é necessário um servidor web com suporte a PHP.

> **Note:**
> Você pode saber mais sobre como configurar SSL para o frontend do Zabbix consultando estas melhores práticas.

Tela de boas-vindas

Abra a URL do frontend do Zabbix no navegador. Se você instalou o Zabbix a partir de pacotes, a URL é:

- para Apache: *http://<server_ip_or_name>/zabbix*
- para Nginx: *http://<server_ip_or_name>*

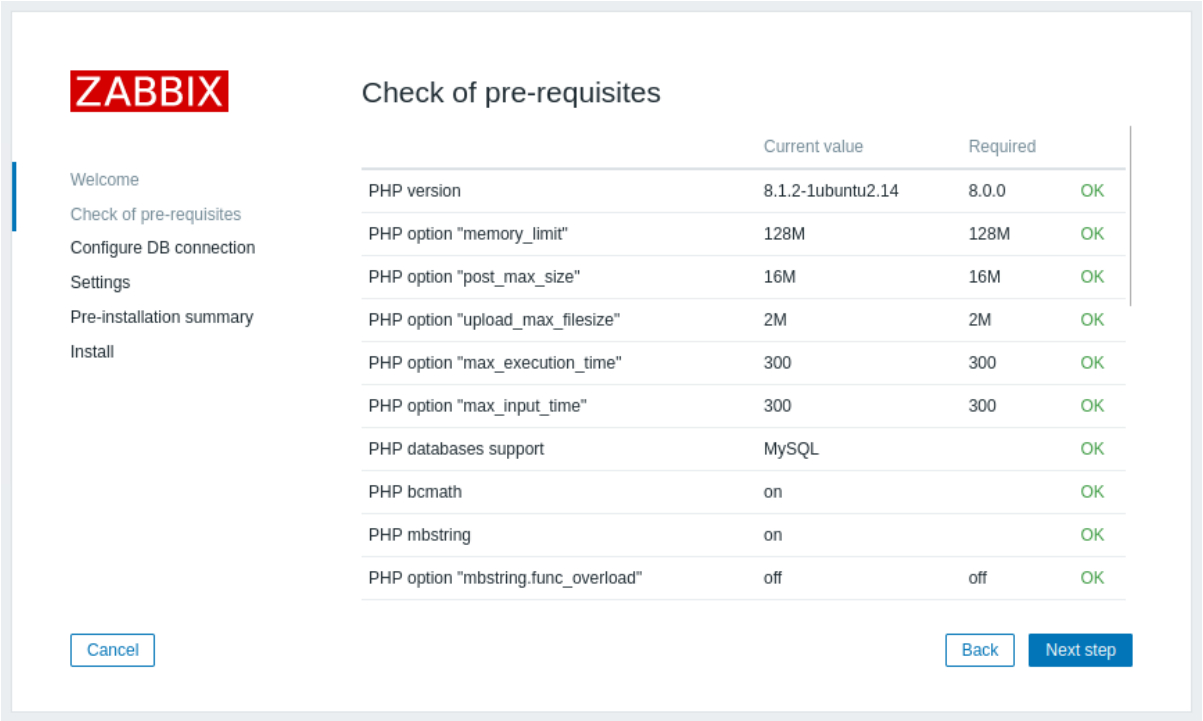
Você deve ver a primeira tela do assistente de instalação do frontend.

Use o menu suspenso *Idioma padrão* para alterar o idioma padrão do sistema e continuar o processo de instalação no idioma selecionado (opcional). Para mais informações, consulte Instalação de idiomas adicionais do frontend.

Observe que definir o idioma como *English (en_US)* também ativará o formato de data/hora dos EUA no frontend.

Verificação dos pré-requisitos

Certifique-se de que todos os pré-requisitos obrigatórios do frontend do Zabbix foram atendidos.



Pré-requisito	Valor mínimo	Descrição
Versão do PHP	8.0.0	
Opção do PHP "memory_limit"	128MB	No php.ini: memory_limit = 128M
Opção do PHP "post_max_size"	16MB	No php.ini: post_max_size = 16M
Opção do PHP "upload_max_filesize"	2MB	No php.ini: upload_max_filesize = 2M
Opção do PHP "max_execution_time"	300 segundos	No php.ini: max_execution_time = 300 (os valores 0 e -1 também são permitidos)
Opção do PHP "max_input_time"	300 segundos	No php.ini: max_input_time = 300 (os valores 0 e -1 também são permitidos)
Suporte a bancos de dados PHP	Um dos: MySQL, PostgreSQL, Oracle	Veja Requisitos para a lista de todas as extensões PHP obrigatórias e opcionais. Observe que pré-requisitos opcionais não atendidos são exibidos com o status *Aviso* em vermelho, e o processo de configuração pode continuar mesmo que eles não sejam atendidos.
PHP bcmath	deve estar habilitado	
PHP mbstring	deve estar habilitado	
Opção do PHP "mbstring.func_overload"	deve estar desabilitado	No php.ini: mbstring.func_overload = 0
Opção do PHP "session.auto_start"	deve estar desabilitado	No php.ini: session.auto_start = 0
Opção do PHP "arg_separator.output"	&	No php.ini: arg_separator.output = "&" (o valor "&" também é permitido)

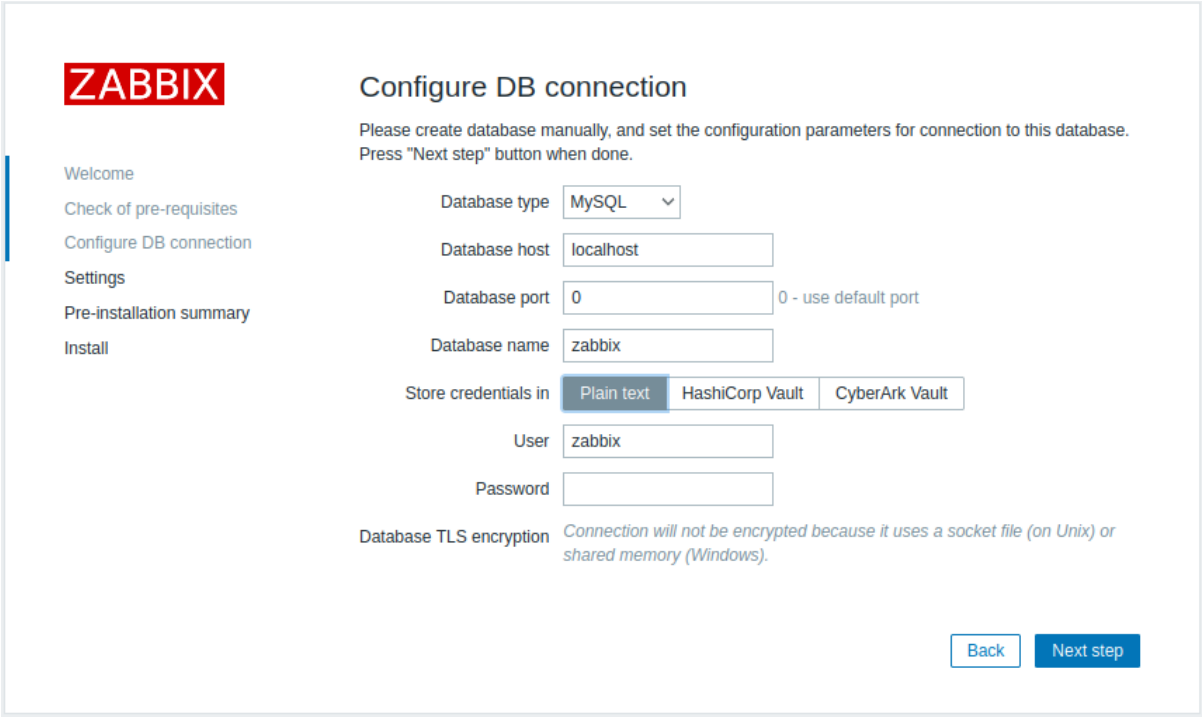
Attention:
Se o usuário ou grupo de usuários do Apache precisar ser alterado, verifique as permissões para a pasta de sessão; caso contrário, a configuração do Zabbix pode não conseguir continuar.

Configurar conexão com o banco de dados

Insira os detalhes para conectar ao banco de dados. O banco de dados do Zabbix já deve estar criado.

Para MySQL, inserir `localhost` ou deixar o campo *Host do banco de dados* vazio resulta em conexão via o socket Unix padrão. O formulário de configuração não fornece um campo separado para *Socket do banco de dados*, portanto, para usar um socket personalizado, configure-o nas configurações do Zabbix server (por exemplo, com `DBSocket=` no zabbix_server.conf). Isso mantém o frontend alinhado com as configurações de conexão do server com o banco de dados.

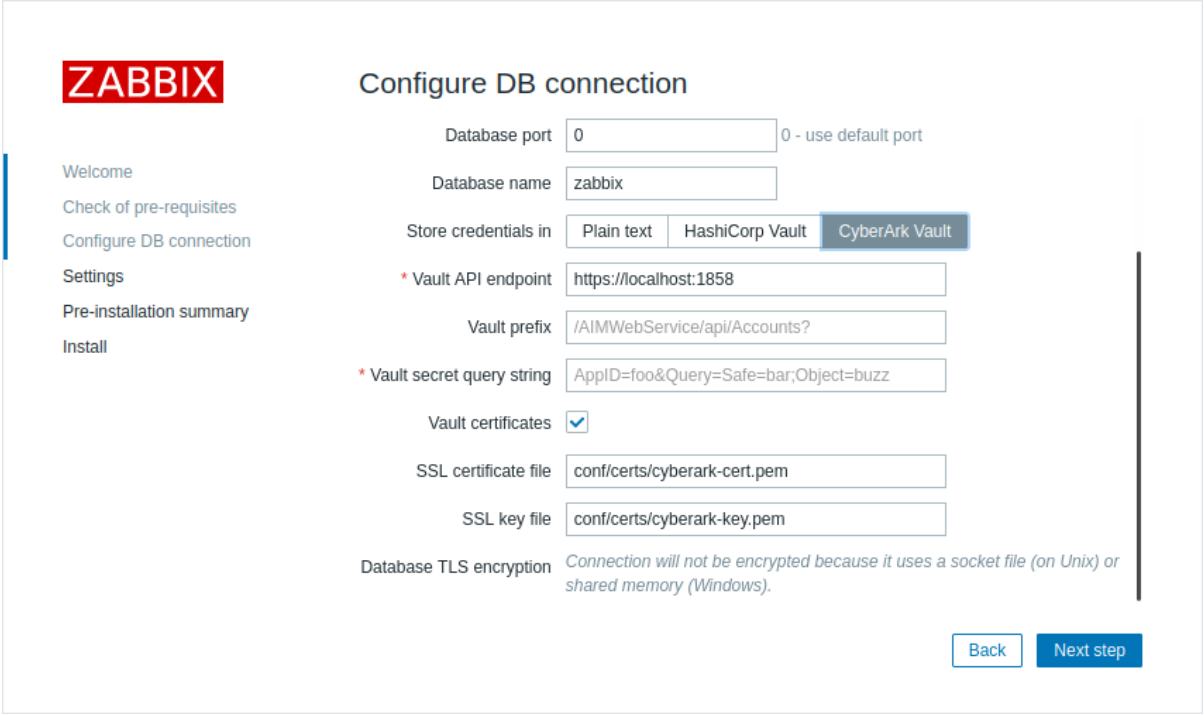
Para PostgreSQL, se o campo *Host do banco de dados* for deixado vazio, o socket Unix padrão será usado. Se um caminho de socket for inserido (por exemplo, `/var/run/pgbouncer`), esse socket Unix será usado em vez disso.



Se a opção *Criptografia TLS do banco de dados* estiver marcada, campos adicionais para configurar a conexão TLS com o banco de dados aparecerão no formulário (apenas MySQL ou PostgreSQL).

Se *Armazenar credenciais em* estiver definido como HashiCorp Vault ou CyberArk Vault, parâmetros adicionais ficarão disponíveis:

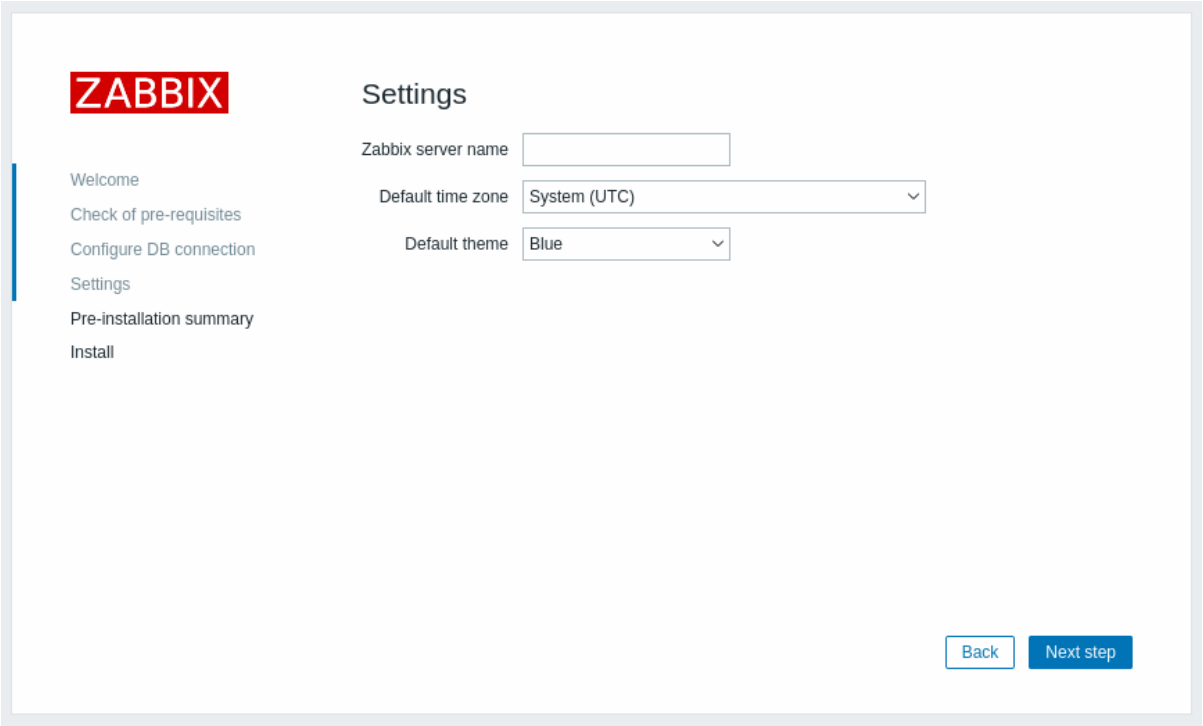
- para HashiCorp Vault: endpoint da API do Vault, prefixo do vault, caminho do segredo e token de autenticação;
- para CyberArk Vault: endpoint da API do Vault, prefixo do vault, string de consulta do segredo e certificados. Ao marcar a caixa *Certificados do Vault*, dois novos campos para especificar os caminhos para o arquivo de certificado SSL e o arquivo de chave SSL aparecerão.



Configurações

A entrada de um nome para o Zabbix server é opcional, no entanto, se informado, ele será exibido na barra de menu e nos títulos das páginas.

Defina o fuso horário padrão e o tema para o frontend.

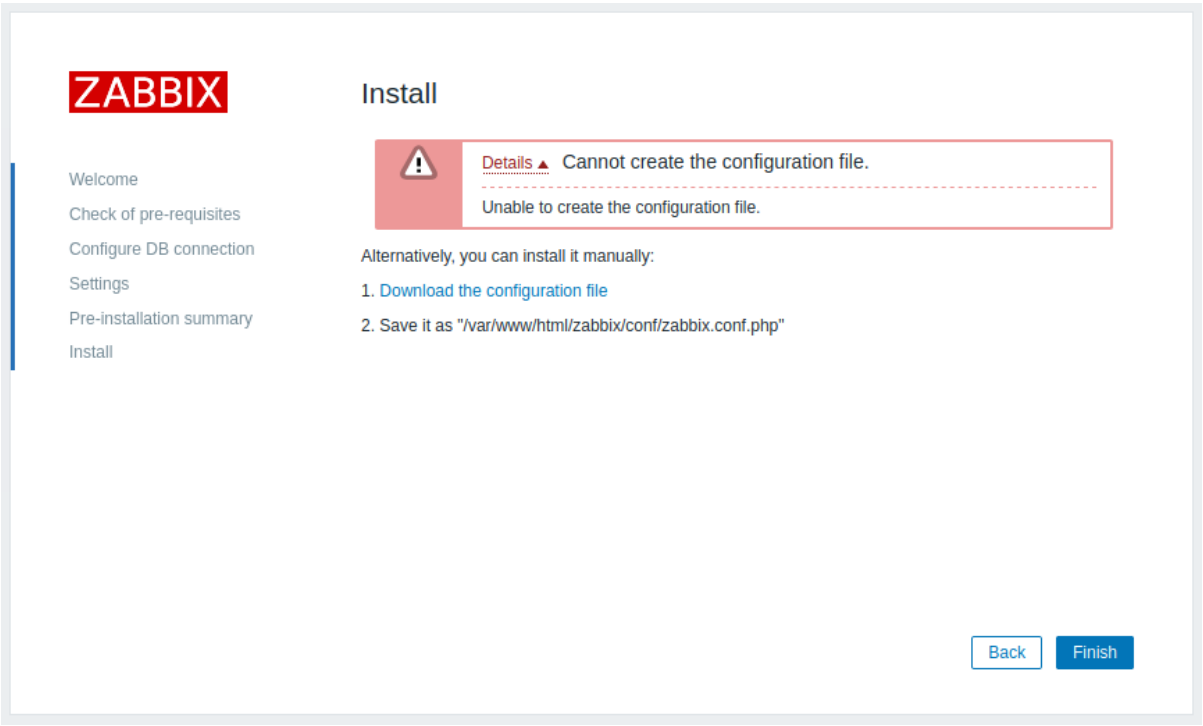


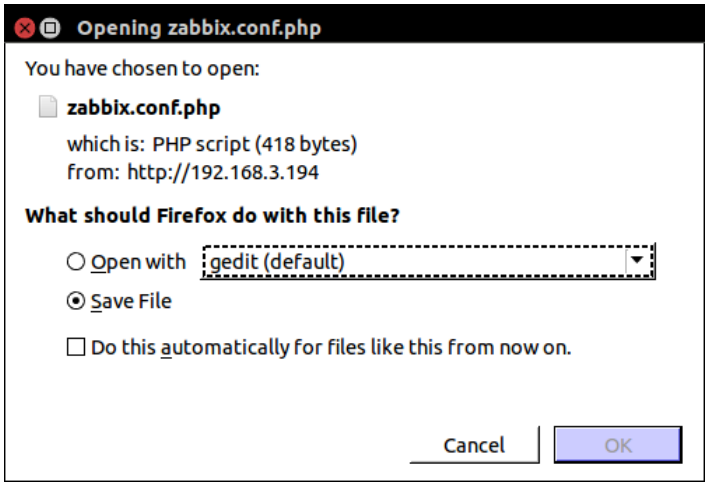
Resumo da pré-instalação

Revise um resumo das configurações.

Instalar

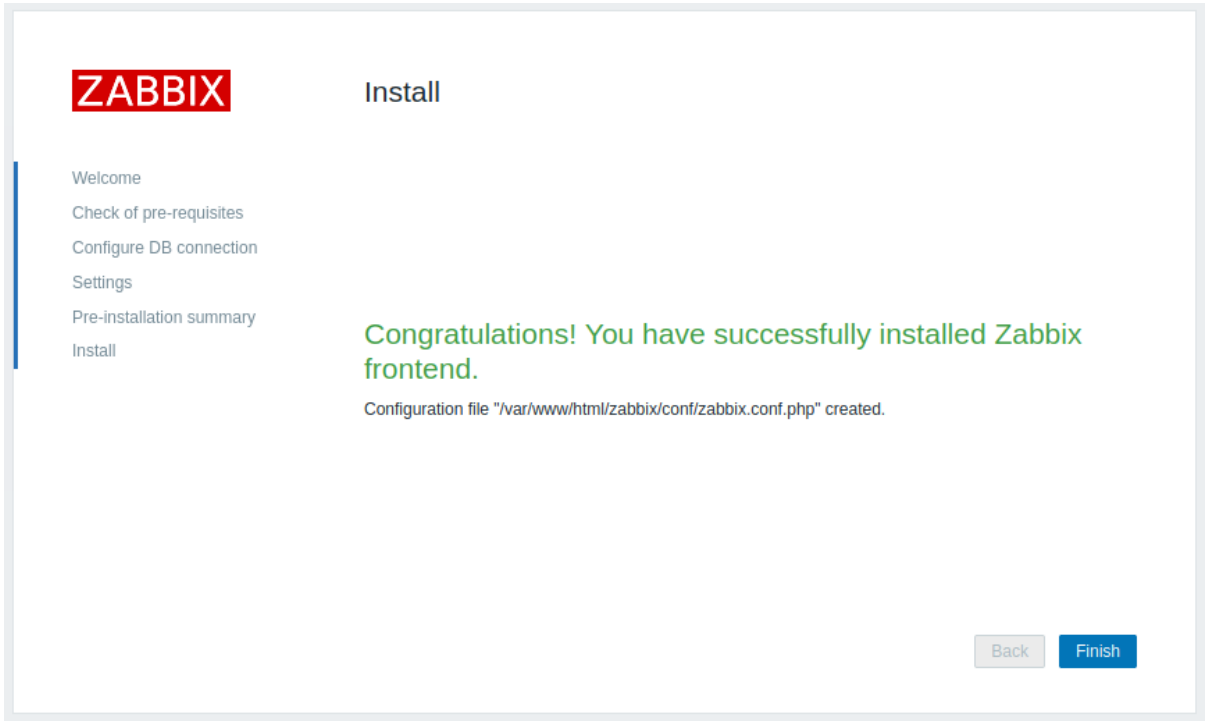
Se estiver instalando o Zabbix a partir do código-fonte, baixe o arquivo de configuração e coloque-o em conf/ no subdiretório de documentos HTML do servidor web onde você copiou os arquivos PHP do Zabbix.





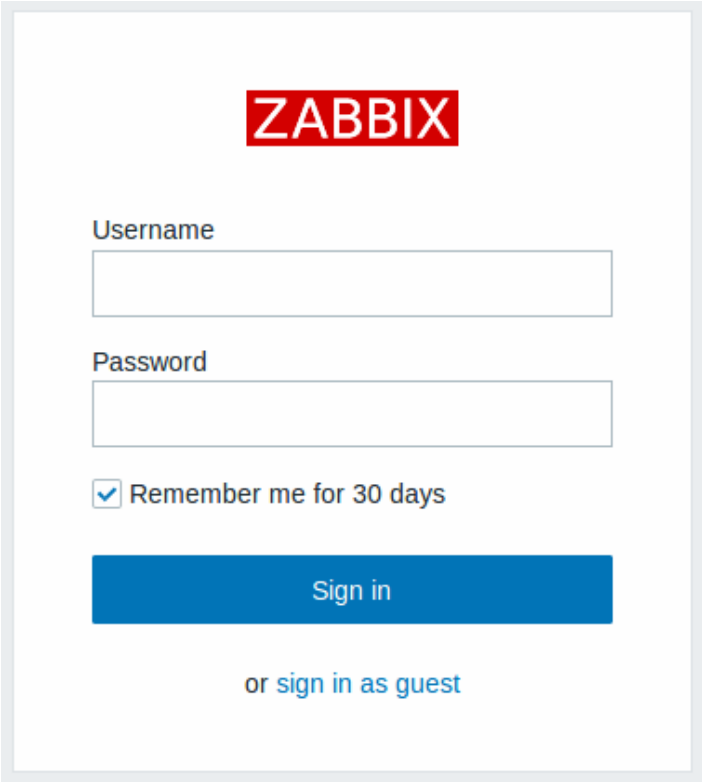
Note:
Desde que o usuário do servidor web tenha acesso de gravação ao diretório conf/, o arquivo de configuração será salvo automaticamente e será possível prosseguir para a próxima etapa imediatamente.

Conclua a instalação.



Login

O frontend do Zabbix está pronto! O nome de usuário padrão é **Admin**, a senha é **zabbix**.



Prossiga para primeiros passos com o Zabbix.

7 Procedimento de atualização

Visão geral

Esta seção fornece informações de atualização para o Zabbix **7.0**:

- Usando pacotes para Red Hat Enterprise Linux ou Debian/Ubuntu
- Usando fontes

Instruções relacionadas:

- Para servidores em um cluster de alta disponibilidade (HA), consulte Atualizando o cluster HA
- Para o banco de dados TimescaleDB, consulte Atualizando o schema do TimescaleDB

A atualização dos agents do Zabbix é recomendada, mas não obrigatória.

A atualização dos proxies do Zabbix é altamente recomendada. O server do Zabbix oferece suporte total a proxies que sejam da mesma versão principal do server. O server do Zabbix também oferece suporte a proxies que sejam **no máximo** da versão LTS anterior do server do Zabbix, mas com funcionalidade limitada (coleta de dados, execução de comandos remotos, verificações imediatas de valores de item). A atualização da configuração também é desabilitada, e proxies desatualizados funcionarão apenas com a configuração antiga.

> **Attention:**
> Proxies que são mais antigos do que a versão LTS anterior do Zabbix server ou mais recentes do que a versão principal do Zabbix server não são suportados. O Zabbix server irá ignorar dados de proxies não suportados e toda a comunicação com o Zabbix server falhará com um aviso. Para mais informações, consulte Compatibilidade de versão.

Para minimizar o tempo de inatividade e a perda de dados durante a atualização, recomenda-se parar, atualizar e iniciar o Zabbix server e, em seguida, parar, atualizar e iniciar os proxies Zabbix um após o outro. Durante o tempo de inatividade do server, os proxies em execução continuarão a coletar dados. Assim que o server estiver em funcionamento, os proxies desatualizados enviarão os dados para o novo server (a configuração do proxy não será atualizada, porém) e permanecerão parcialmente funcionais. Quaisquer notificações de problemas durante o tempo de inatividade do Zabbix server serão geradas somente após o server atualizado ser iniciado.

Se o proxy Zabbix for iniciado pela primeira vez e o arquivo de banco de dados SQLite estiver ausente, o proxy o criará automaticamente.

Observe que se o proxy Zabbix usar SQLite3 e, na inicialização, detectar que a versão do arquivo de banco de dados existente é mais antiga do que a necessária, ele **excluirá o arquivo de banco de dados automaticamente** e criará um novo. Portanto, os dados de histórico armazenados no arquivo de banco de dados SQLite serão perdidos. Se a versão do proxy Zabbix for mais antiga do que a versão do arquivo de banco de dados, o Zabbix registrará um erro e encerrará.

Dependendo do tamanho do banco de dados, a atualização do banco de dados para a versão 7.0 pode levar muito tempo.

A atualização direta para o Zabbix 7.0.x é possível a partir do Zabbix **6.4**.x, **6.2**.x, **6.0**.x, **5.4**.x, **5.2**.x, **5.0**.x, **4.4**.x, **4.2**.x, **4.0**.x, **3.4**.x, **3.2**.x, **3.0**.x, **2.4**.x, **2.2**.x e **2.0**.x. Para atualizar a partir de versões anteriores, consulte a documentação do Zabbix para 2.0 e anteriores.

Note:
Esteja ciente de que, após a atualização, algumas integrações de software de terceiros no Zabbix podem ser afetadas se o software externo não for compatível com a versão do Zabbix atualizada.

As seguintes notas de atualização estão disponíveis:

Atualizar de	Leia as notas de atualização completas	Principais mudanças entre as versões
6.4.x	Para: Zabbix 7.0	Versão mínima do PHP aumentada de 7.4.0 para 8.0.0. Pollers assíncronos para verificações de agent, HTTP agent e SNMP walk[oid]. Tabela de banco de dados separada para proxies. Local padrão do arquivo de configuração do agent do Windows alterado. Banco de dados Oracle obsoleto. Tipo de valor numérico antigo (float) removido.
6.2.x	Para: Zabbix 6.4 Zabbix 7.0	Versão mínima do MySQL aumentada de 8.0.0 para 8.0.30. A biblioteca 'libevent_pthreads' é necessária para o Zabbix server/proxy. Na primeira inicialização após uma atualização, o Zabbix proxy com SQLite3 descarta automaticamente a versão antiga do banco de dados (com todo o histórico) e cria um novo.
6.0.x LTS	Para: Zabbix 6.2 Zabbix 6.4 Zabbix 7.0	Versão mínima do PHP aumentada de 7.2.5 para 7.4.0. Monitoramento de serviços reformulado significativamente. Triggers determinísticos precisam ser criados durante a atualização. Se o log binário estiver habilitado para MySQL/MariaDB, isso requer privilégios de superusuário ou a configuração da variável/parâmetro de configuração *log_bin_trust_function_creators = 1*. Veja scripts de criação de banco de dados para instruções de como definir a variável.
5.4.x	Para: Zabbix 6.0 Zabbix 6.2 Zabbix 6.4 Zabbix 7.0	Versões mínimas dos bancos de dados aumentadas. Server/proxy não iniciará se o banco de dados estiver desatualizado. Registros do log de auditoria perdidos devido à alteração da estrutura do banco de dados.
5.2.x	Para: Zabbix 5.4 Zabbix 6.0 Zabbix 6.2 Zabbix 6.4 Zabbix 7.0	Versões mínimas dos bancos de dados aumentadas. Itens agregados removidos como um tipo separado.
5.0.x LTS	Para: Zabbix 5.2 Zabbix 5.4 Zabbix 6.0 Zabbix 6.2 Zabbix 6.4 Zabbix 7.0	Versão mínima do PHP aumentada de 7.2.0 para 7.2.5. Algoritmo de hash de senha alterado de MD5 para bcrypt.

Atualizar de	Leia as notas de atualização completas	Principais mudanças entre as versões
4.4.x	Para: Zabbix 5.0 Zabbix 5.2 Zabbix 5.4 Zabbix 6.0 Zabbix 6.2 Zabbix 6.4 Zabbix 7.0	Suporte ao IBM DB2 removido. Versão mínima do PHP aumentada de 5.4.0 para 7.2.0. Versões mínimas dos bancos de dados aumentadas. Diretório de arquivos PHP do Zabbix alterado.
4.2.x	Para: Zabbix 4.4 Zabbix 5.0 Zabbix 5.2 Zabbix 5.4 Zabbix 6.0 Zabbix 6.2 Zabbix 6.4 Zabbix 7.0	Tipos de mídia Jabber, Ez Texting removidos.
4.0.x LTS	Para: 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	Proxies antigos não podem mais reportar dados para um server atualizado. Agents mais recentes não poderão mais trabalhar com um Zabbix server mais antigo.
3.4.x	Para: 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	Bibliotecas ’libpthread’ e ’zlib’ agora obrigatórias. Suporte ao protocolo texto puro removido e cabeçalho é obrigatório. Agents Zabbix versão pré-1.4 não são mais suportados. O parâmetro Server na configuração do proxy passivo agora é obrigatório.
3.2.x	Para: 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	Suporte ao SQLite como banco de dados backend removido para Zabbix server/frontend. Expressões Regulares Compatíveis com Perl (PCRE) suportadas em vez de POSIX extended. Bibliotecas ’libpcre’ e ’libevent’ obrigatórias para o Zabbix server. Verificações de código de saída adicionadas para parâmetros de usuário, comandos remotos e itens system.run[] sem o flag ’nowait’, bem como scripts executados pelo Zabbix server. Zabbix Java gateway deve ser atualizado para suportar a nova funcionalidade.

Atualizar de	Leia as notas de atualização completas	Principais mudanças entre as versões
3.0.x LTS	Para: 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	A atualização do banco de dados pode ser lenta, dependendo do tamanho da tabela de histórico.
2.4.x	Para: 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	Versão mínima do PHP aumentada de 5.3.0 para 5.4.0. O parâmetro LogFile do agent deve ser especificado.
2.2.x LTS	Para: 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	Monitoramento distribuído baseado em nó removido.
2.0.x	Para: 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	Versão mínima do PHP aumentada de 5.1.6 para 5.3.0. Banco de dados MySQL case-sensitive necessário para o funcionamento adequado do server; conjunto de caracteres utf8 e collation utf8_bin são necessários para o Zabbix server funcionar corretamente com o banco de dados MySQL. Veja scripts de criação de banco de dados. Extensão PHP ’mysqli’ necessária em vez de ’mysql’.

1 Atualização a partir do código-fonte

Visão geral

Esta seção fornece as etapas necessárias para uma atualização bem-sucedida do Zabbix **6.4**.x para o Zabbix **7.0**.x usando fontes oficiais do Zabbix.

> **Warning:**
> Antes da atualização, certifique-se de ler as **notas de atualização** relevantes!

Você também pode querer verificar os requisitos para a versão 7.0.

> **Note:**
> Pode ser útil executar duas sessões SSH paralelas durante a atualização, executando as etapas de atualização em uma e monitorando os logs do server/proxy em outra. Por exemplo, execute `tail -f zabbix_server.log` ou `tail -f zabbix_proxy.log` na segunda sessão SSH para mostrar as últimas entradas do arquivo de log e possíveis erros em tempo real. Isso pode ser crítico para instâncias de produção.

Processo de atualização do server

1 Parar o server

Pare o Zabbix server para garantir que nenhum novo dado seja inserido no banco de dados.

2 Faça backup do banco de dados Zabbix existente

Esta é uma etapa muito importante. Certifique-se de que você tem um backup do seu banco de dados. Isso ajudará caso o procedimento de atualização falhe (falta de espaço em disco, desligamento, qualquer problema inesperado).

3 Faça backup dos arquivos de configuração, arquivos PHP e binários do Zabbix

Faça uma cópia de backup dos binários do Zabbix, arquivos de configuração e do diretório de arquivos PHP.

4 Instale os novos binários do server

Use estas instruções para compilar o Zabbix server a partir do código-fonte.

5 Revisar os parâmetros de configuração do server

Certifique-se de revisar as Notas de atualização para verificar se são necessárias alterações nos parâmetros de configuração.

Para novos parâmetros opcionais, consulte a página O que há de novo.

6 Inicie os novos binários do Zabbix

Inicie os novos binários. Verifique os arquivos de log para ver se os binários foram iniciados com sucesso.

O servidor Zabbix atualizará automaticamente o banco de dados. Ao iniciar, o servidor Zabbix informa as versões atuais (obrigatórias e opcionais) e as versões exigidas do banco de dados. Se a versão obrigatória atual for mais antiga que a versão exigida, o servidor Zabbix executa automaticamente os patches de atualização necessários no banco de dados. O início e o nível de progresso (porcentagem) da atualização do banco de dados são registrados no arquivo de log do servidor Zabbix. Quando a atualização for concluída, uma mensagem "database upgrade fully completed" será registrada no arquivo de log. Se algum dos patches de atualização falhar, o servidor Zabbix não será iniciado. O servidor Zabbix também não será iniciado se a versão obrigatória atual do banco de dados for mais recente que a exigida. O servidor Zabbix só será iniciado se a versão obrigatória atual do banco de dados corresponder à versão obrigatória exigida.

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

Antes de iniciar o servidor:

- Certifique-se de que o usuário do banco de dados tenha permissões suficientes (create table, drop table, create index, drop index)
- Certifique-se de que há espaço livre suficiente em disco.

7 Instale a nova interface web do Zabbix

A versão mínima do PHP necessária é 8.0.0. Atualize se necessário e siga as instruções de instalação.

8 Limpe os cookies e o cache do navegador

Após a atualização, pode ser necessário limpar os cookies e o cache do navegador para que a interface web do Zabbix funcione corretamente.

Processo de atualização do proxy

1 Parar o proxy

Pare o proxy Zabbix.

2 Faça backup dos arquivos de configuração e binários do Zabbix proxy

Faça uma cópia de backup do binário do Zabbix proxy e do arquivo de configuração.

3 Instale os novos binários do proxy

Use estas instruções para compilar o proxy Zabbix a partir do código-fonte.

4 Revisar os parâmetros de configuração do proxy

Certifique-se de revisar as Notas de atualização para verificar se são necessárias alterações nos parâmetros de configuração.

Para novos parâmetros opcionais, consulte a página O que há de novo.

5 Inicie o novo proxy Zabbix

Inicie o novo proxy Zabbix. Verifique os arquivos de log para ver se o proxy foi iniciado com sucesso.

O proxy Zabbix atualizará automaticamente o banco de dados. A atualização do banco de dados ocorre de forma semelhante à inicialização do server Zabbix.

Processo de atualização do agent

> **Attention:**
> A atualização dos agents não é obrigatória. Você só precisa atualizar os agents se for necessário acessar a nova funcionalidade.

O procedimento de atualização descrito nesta seção pode ser usado para atualizar tanto o agent Zabbix quanto o agent Zabbix 2.

1 Parar o agent

Pare o agent do Zabbix.

2 Faça backup dos arquivos de configuração e dos binários do agent Zabbix

Faça uma cópia de backup do binário do agent Zabbix e do arquivo de configuração.

3 Instale os binários do novo agent

Use estas instruções para compilar o agent Zabbix a partir do código-fonte.

Como alternativa, você pode baixar agents Zabbix pré-compilados da página de download do Zabbix.

4 Revisar os parâmetros de configuração do agent

Certifique-se de revisar as Notas de atualização para verificar se são necessárias alterações nos parâmetros de configuração.

Para novos parâmetros opcionais, consulte a página O que há de novo.

5 Inicie o novo agent do Zabbix

Inicie o novo agent do Zabbix. Verifique os arquivos de log para ver se o agent foi iniciado com sucesso.

Atualização entre versões secundárias

Ao atualizar entre versões secundárias do 7.0.x (por exemplo, da 7.0.1 para a 7.0.3), é necessário executar as mesmas ações para server/proxy/agent que durante a atualização entre versões principais. A única diferença é que, ao atualizar entre versões secundárias, nenhuma alteração no banco de dados é feita.

2 Atualização a partir de pacotes

Visão geral

Esta seção fornece as etapas necessárias para uma atualização bem-sucedida usando os pacotes oficiais RPM e DEB fornecidos pelo Zabbix para:

- Red Hat Enterprise Linux
- Debian/Ubuntu

Pacotes Zabbix dos repositórios do sistema operacional

Algumas distribuições de sistema operacional (em particular, distribuições baseadas em Debian) fornecem seus próprios pacotes Zabbix. Esses pacotes **não são suportados pelo Zabbix** e podem estar desatualizados ou sem os recursos e correções de bugs mais recentes. Recomenda-se usar apenas pacotes oficiais do Repositório Oficial do Zabbix.

Se você estiver atualizando a partir de pacotes fornecidos por distribuições de sistema operacional (ou os teve instalados em algum momento), siga este procedimento para mudar para os pacotes oficiais do Zabbix:

1. Desinstale os pacotes antigos.
2. Verifique e remova quaisquer arquivos restantes da instalação antiga.
3. Instale os pacotes oficiais do Zabbix seguindo as instruções de instalação fornecidas pelo Zabbix.

Não faça uma atualização direta, pois isso pode resultar em uma instalação corrompida.

1 Red Hat Enterprise Linux

Visão geral

Esta seção fornece instruções para atualizar do Zabbix **6.4.x** para a versão mais recente do Zabbix **7.0.x** usando os pacotes oficiais do Zabbix para Red Hat Enterprise Linux ou seus derivados - AlmaLinux, CentOS Stream, Oracle Linux e Rocky Linux.

> **Note:**
> Antes do Zabbix 7.0, pacotes de instalação únicos eram fornecidos para distribuições baseadas em RHEL e RHEL. A partir da versão 7.0, pacotes separados são usados para o RHEL e cada um de seus derivados mencionados acima para evitar possíveis problemas de incompatibilidade binária.

> **Warning:**
> Antes de atualizar, revise as notas de atualização relevantes e certifique-se de que seu sistema atenda aos requisitos para o Zabbix 7.0.

> **Note:**
> Considere executar duas sessões SSH paralelas durante a atualização: uma para executar as etapas de atualização e outra para monitorar os logs do server/proxy. Por exemplo, execute `tail -f zabbix_server.log` ou `tail -f zabbix_proxy.log` na segunda sessão para visualizar as entradas de log mais recentes e possíveis erros em tempo real. Isso pode ser crítico para instâncias de produção.

Para instruções sobre como atualizar entre versões menores do Zabbix 7.0.x (por exemplo, da 7.0.1 para a 7.0.3), consulte Atualização entre versões menores.

Procedimento de atualização

1 Parar os processos do Zabbix

Pare o servidor Zabbix para garantir que nenhum novo dado seja inserido no banco de dados:

```
systemctl stop zabbix-server
```

Se estiver atualizando o proxy, agent ou agent 2 do Zabbix, pare esses componentes também:

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

2 Faça backup do banco de dados do Zabbix

Faça backup do seu banco de dados Zabbix existente para se proteger contra falhas de atualização (por exemplo, problemas de espaço em disco, falta de energia ou problemas inesperados).

3 Faça backup dos arquivos de configuração do Zabbix, arquivos PHP e binários do Zabbix

Faça backup dos arquivos de configuração existentes do Zabbix, arquivos PHP e binários do Zabbix.

Para os arquivos de configuração, execute:

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

Para arquivos PHP e binários do Zabbix, execute:

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

4 Atualizar o pacote de configuração do repositório

Antes de prosseguir com a atualização, atualize o pacote de repositório atual para a versão mais recente para garantir a compatibilidade com os pacotes mais novos e incluir quaisquer correções de segurança ou bugs recentes.

No **RHEL 9**, execute:

```
rpm -Uvh https://repo.zabbix.com/zabbix/7.0/rhel/9/x86_64/zabbix-release-latest.el9.noarch.rpm
```

No **RHEL 8**, execute:

```
rpm -Uvh https://repo.zabbix.com/zabbix/7.0/rhel/8/x86_64/zabbix-release-latest.el8.noarch.rpm
```

Note:
Para versões mais antigas do RHEL ou seus derivados, substitua o link acima pelo correto do repositório Zabbix. Observe, no entanto, que os pacotes para essas versões podem não incluir todos os componentes do Zabbix e, para atualizar esses componentes a partir dos pacotes, considere atualizar seu sistema operacional. Para uma lista dos componentes incluídos, consulte Pacotes Zabbix.

Em seguida, limpe o cache do gerenciador de pacotes `dnf` (incluindo cabeçalhos, metadados e arquivos de pacotes baixados durante instalações ou atualizações anteriores):

```
dnf clean all
```

Na próxima operação do `dnf`, o `dnf` fará o download de metadados atualizados dos repositórios, pois os metadados antigos foram limpos.

Veja também: Problemas conhecidos para atualizar o pacote de configuração do repositório no RHEL.

5 Atualizar componentes do Zabbix

Para atualizar os componentes do Zabbix, execute:

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

- Se estiver usando PostgreSQL, substitua `mysql` por `pgsql` no comando.
- Se estiver atualizando o proxy, substitua `server` por `proxy` no comando.
- Se estiver atualizando o agent 2, substitua `zabbix-agent` por `zabbix-agent2 zabbix-agent2-plugin-*` no comando.

Attention:
Atualizar o Zabbix agent 2 com o comando `dnf install zabbix-agent2` pode causar um erro. Para mais informações, consulte *Problemas conhecidos*.

Em seguida, para atualizar o frontend do Zabbix com Apache e reiniciar o Apache, execute:

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

6 Revisar os parâmetros de configuração dos componentes

Revise as notas de atualização relevantes para verificar se são necessárias alterações nos parâmetros de configuração.

Para novos parâmetros opcionais, consulte a página O que há de novo.

7 Inicie os processos do Zabbix

Inicie os componentes atualizados do Zabbix:

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

8 Limpe os cookies e o cache do navegador

Após a atualização, pode ser necessário limpar os cookies e o cache do navegador para que a interface web do Zabbix funcione corretamente.

Atualização entre versões secundárias

É possível atualizar entre versões secundárias do Zabbix 7.0.x (por exemplo, de 7.0.1 para 7.0.3).

Para atualizar todos os componentes do Zabbix, execute:

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

- Para atualizar apenas o Zabbix server, substitua `'zabbix-*'` por `'zabbix-server-*'` no comando.
- Para atualizar apenas o Zabbix proxy, substitua `'zabbix-*'` por `'zabbix-proxy-*'` no comando.
- Para atualizar apenas o Zabbix agent, substitua `'zabbix-*'` por `'zabbix-agent-*'` no comando.
- Para atualizar apenas o Zabbix agent 2, substitua `'zabbix-*'` por `'zabbix-agent2-*'` no comando.

2 Debian/Ubuntu

Visão geral

Esta seção fornece instruções para atualizar do Zabbix **6.4.x** para a versão mais recente do Zabbix **7.0.x** usando os pacotes oficiais do Zabbix para Debian/Ubuntu.

Warning:
Antes de atualizar, revise as notas de atualização relevantes e certifique-se de que seu sistema atenda aos requisitos para o Zabbix 7.0.

Note:
Considere executar duas sessões SSH paralelas durante a atualização: uma para executar as etapas de atualização e outra para monitorar os logs do server/proxy. Por exemplo, execute `tail -f zabbix_server.log` ou `tail -f zabbix_proxy.log` na segunda sessão para visualizar as entradas de log mais recentes e possíveis erros em tempo real. Isso pode ser crítico para instâncias de produção.

Para instruções sobre como atualizar entre versões menores do Zabbix 7.0.x (por exemplo, de 7.0.1 para 7.0.3), consulte Atualizar entre versões menores.

Procedimento de atualização

1 Parar os processos do Zabbix

Pare o servidor Zabbix para garantir que nenhum novo dado seja inserido no banco de dados:

```
systemctl stop zabbix-server
```

Se estiver atualizando o proxy, agent ou agent 2 do Zabbix, pare esses componentes também:

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

2 Faça backup do banco de dados do Zabbix

Faça backup do seu banco de dados Zabbix existente para se proteger contra falhas de atualização (por exemplo, problemas de espaço em disco, falta de energia ou problemas inesperados).

3 Faça backup dos arquivos de configuração do Zabbix, arquivos PHP e binários do Zabbix

Faça backup dos arquivos de configuração existentes do Zabbix, arquivos PHP e binários do Zabbix.

Para os arquivos de configuração, execute:

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

Para os arquivos PHP e binários do Zabbix, execute:

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

4 Atualizar o pacote de configuração do repositório

Antes de prosseguir com a atualização, desinstale o pacote de repositório Zabbix atual:

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

Você também pode precisar remover manualmente quaisquer pacotes antigos do Zabbix do seu diretório de trabalho (por exemplo, `rm zabbix-release_latest+debian12_all.deb`) antes de baixar o novo, para evitar que o gerenciador de pacotes reutilize uma versão desatualizada durante o processo de atualização.

Em seguida, instale o pacote de configuração do repositório mais recente para garantir a compatibilidade com os pacotes mais recentes e incluir quaisquer correções de segurança ou bugs recentes.

No **Debian 12**, execute:

```
wget https://repo.zabbix.com/zabbix/7.0/debian/pool/main/z/zabbix-release/zabbix-release_latest+debian12_a
dpkg -i zabbix-release_latest+debian12_all.deb
```

> **Note:**
> Para versões mais antigas do Debian, substitua o link acima pelo correto do repositório Zabbix. Observe, no entanto, que os pacotes para essas versões podem não incluir todos os componentes do Zabbix e, para atualizar esses componentes a partir dos pacotes, considere atualizar seu sistema operacional. Para uma lista de componentes incluídos, consulte o repositório Zabbix.

No **Ubuntu 24.04**, execute:

```
wget https://repo.zabbix.com/zabbix/7.0/ubuntu/pool/main/z/zabbix-release/zabbix-release_latest+ubuntu24.0
dpkg -i zabbix-release_latest+ubuntu24.04_all.deb
```

No **Ubuntu 22.04**, execute:

```
wget https://repo.zabbix.com/zabbix/7.0/ubuntu/pool/main/z/zabbix-release/zabbix-release_latest+ubuntu22.0
dpkg -i zabbix-release_latest+ubuntu22.04_all.deb
```

> **Note:**
> Para versões mais antigas do Ubuntu, substitua o link acima pelo correto do repositório Zabbix. Observe, no entanto, que os pacotes para essas versões podem não incluir todos os componentes do Zabbix e, para atualizar esses componentes a partir dos pacotes, considere atualizar seu sistema operacional. Para uma lista de componentes incluídos, consulte Zabbix packages.

Você pode ver um prompt sobre a configuração do repositório Zabbix:

```
Configuration file '/etc/apt/sources.list.d/zabbix.list'
==> Deleted (by you or by a script) since installation.
==> Package distributor has shipped an updated version.
What would you like to do about it ?  Your options are:
Y or I  : install the package maintainer's version
N or O  : keep your currently-installed version
D       : show the differences between the versions
Z       : start a shell to examine the situation
The default action is to keep your current version.
*** zabbix.list (Y/I/N/O/D/Z) [default=N] ?
```

Digite `Y` (ou `I`) para instalar a versão do mantenedor do pacote da configuração do repositório Zabbix.

Em seguida, atualize as informações do repositório:

```
apt update
```

5 Atualizar componentes do Zabbix

Para atualizar os componentes do Zabbix, execute:

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

- Se estiver usando PostgreSQL, substitua `mysql` por `pgsql` no comando.
- Se estiver atualizando o proxy, substitua `server` por `proxy` no comando.
- Se estiver atualizando o Zabbix agent 2, substitua `zabbix-agent` por `zabbix-agent2 zabbix-agent2-plugin-*` no comando.

Attention:
Atualizar o Zabbix agent 2 com o comando `apt install zabbix-agent2` pode causar um erro. Para mais informações, consulte *Problemas conhecidos*.

Você pode ver um prompt sobre a configuração do Zabbix server (ou proxy):

```
Configuration file '/etc/zabbix/zabbix_server.conf'
==> Modified (by you or by a script) since installation.
==> Package distributor has shipped an updated version.
What would you like to do about it ?  Your options are:
Y or I  : install the package maintainer's version
N or O  : keep your currently-installed version
D       : show the differences between the versions
Z       : start a shell to examine the situation
The default action is to keep your current version.
*** zabbix_server.conf (Y/I/N/O/D/Z) [default=N] ?
```

Digite a opção que melhor se encaixa na sua situação. Por exemplo, digite `D` para comparar a configuração atual e a nova, depois decida se deseja instalar a versão do mantenedor do pacote (`Y` ou `I`).

Em seguida, para atualizar o frontend do Zabbix com Apache e reiniciar o Apache, execute:

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

6 Revisar os parâmetros de configuração dos componentes

Revise as notas de atualização relevantes para verificar se são necessárias alterações nos parâmetros de configuração.

Para novos parâmetros opcionais, consulte a página O que há de novo.

7 Inicie os processos do Zabbix

Inicie os componentes atualizados do Zabbix:

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

8 Limpar cookies e cache do navegador

Após a atualização, pode ser necessário limpar os cookies e o cache do navegador para que a interface web do Zabbix funcione corretamente.

Atualização entre versões secundárias

É possível atualizar entre versões secundárias do Zabbix 7.0.x (por exemplo, de 7.0.1 para 7.0.3).

Primeiro, atualize as informações do repositório:

```
apt update
```

Em seguida, para atualizar todos os componentes do Zabbix, execute:

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

- Para atualizar apenas o Zabbix server, substitua `'zabbix*'` por `'zabbix-server*'` no comando.
- Para atualizar apenas o Zabbix proxy, substitua `'zabbix*'` por `'zabbix-proxy*'` no comando.
- Para atualizar apenas o Zabbix agent, substitua `'zabbix*'` por `'zabbix-agent*'` no comando.
- Para atualizar apenas o Zabbix agent 2, substitua `'zabbix*'` por `'zabbix-agent2*'` no comando.

3 Atualização a partir de containers

Visão geral

Esta seção descreve as etapas necessárias para uma atualização bem-sucedida para os containers do Zabbix **7.0**.x.

Conjuntos separados de instruções estão disponíveis para atualizar individualmente as imagens dos componentes do Zabbix e os arquivos compose do Docker.

Warning:
Antes da atualização, certifique-se de ler as **notas de atualização** relevantes!

Attention:
Antes de iniciar a atualização, verifique se os usuários possuem as permissões necessárias no banco de dados para fins de atualização.
Para atualizações a partir do Zabbix 6.0 ou anterior, triggers determinísticos precisarão ser criados durante a atualização. Se o log binário estiver habilitado para MySQL/MariaDB, isso requer privilégios de superusuário ou a configuração da variável/parâmetro de configuração *log_bin_trust_function_creators = 1*. Consulte Scripts de criação de banco de dados para instruções de como definir a variável.
 Observe que, se executar a partir de um console, a variável será definida apenas temporariamente e será descartada quando o Docker for reiniciado. Neste caso, mantenha seu serviço SQL em execução, pare apenas o serviço zabbix-server executando 'docker compose down zabbix-server' e depois 'docker compose up -d zabbix-server'.
 Como alternativa, você pode definir essa variável no arquivo de configuração.

Dependendo do tamanho do banco de dados, a atualização para a versão 7.0 pode levar bastante tempo.

Atualização da imagem do Zabbix

As etapas listadas abaixo podem ser usadas para atualizar qualquer componente do Zabbix. Substitua `zabbix-server-mysql` pelo nome da imagem do componente desejado.

1. Verifique a versão atual da imagem:

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

2. Baixe a versão desejada da imagem, por exemplo:

```
docker pull zabbix/zabbix-server-mysql:alpine-7.0-latest
```

`zabbix/zabbix-server-mysql:alpine-7.0-latest` irá baixar a versão mais recente lançada do Zabbix server 7.0 com suporte a MySQL baseada no Alpine Linux. Substitua pelo nome do repositório Docker e combinação de tags que você precisa. Veja Instalação a partir de containers para uma lista de opções disponíveis.

3. Pare o container:

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

4. Remova o container:

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

5. Inicie o container atualizado executando o comando `docker run` seguido de argumentos adicionais para especificar as variáveis de ambiente e/ou pontos de montagem necessários.

Exemplos de configuração

Zabbix server com MySQL:

```
docker run --name zabbix-server-mysql -t \
    -e DB_SERVER_HOST="mysql-server" \
    -e MYSQL_DATABASE="zabbix" \
    -e MYSQL_USER="zabbix" \
    -e MYSQL_PASSWORD="zabbix_pwd" \
    -e MYSQL_ROOT_PASSWORD="root_pwd" \
    -e ZBX_JAVAGATEWAY="zabbix-java-gateway" \
    --network=zabbix-net \
    -p 10051:10051 \
    --restart unless-stopped \
    -d zabbix/zabbix-server-mysql:alpine-7.0-latest
```

Zabbix server com PostgreSQL:

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

```
    -p 10051:10051 \
    --volumes-from zabbix-snmptraps \
    --restart unless-stopped \
    -d zabbix/zabbix-server-pgsql:alpine-7.0-latest
```

MySQL server:

```
docker run --name mysql-server -t \
    -e MYSQL_DATABASE="zabbix" \
    -e MYSQL_USER="zabbix" \
    -e MYSQL_PASSWORD="zabbix_pwd" \
    -e MYSQL_ROOT_PASSWORD="root_pwd" \
    --network=zabbix-net \
    --restart unless-stopped \
    -d mysql:8.0-oracle \
    --character-set-server=utf8mb4--collation-server=utf8mb4_bin \
```

Mais exemplos de configuração, incluindo exemplos para outros componentes do Zabbix, estão disponíveis na página Instalação a partir de containers.

6. Verifique a atualização:

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

Arquivos Compose

Siga as instruções de atualização nesta seção, se você instalou o Zabbix usando o arquivo compose.

1. Verifique a versão atual da imagem:

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

2. Baixe as atualizações mais recentes do repositório do GitHub e alterne para o branch necessário:

```
git pull
git checkout 7.0
```

3. Inicie os componentes do Zabbix usando o novo arquivo compose:

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

4. Verifique a atualização:

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

Veja Instalação a partir de containers para mais detalhes, incluindo listas de variáveis de ambiente suportadas e pontos de montagem de volumes.

8 Problemas conhecidos

Veja também: Problemas de compilação.

Problemas conhecidos na 7.0.20

Não é recomendado atualizar para esta versão devido a:

- aumento repentino de uso de CPU se você estiver usando o plugin MySQL do Zabbix agent 2 (veja ZBX-27156)
- gráfico para itens ativos do Zabbix agent exibindo um aviso "Undefined array key" devido a um erro de índice indefinido (veja ZBX-27153)

Atualização

Configuração do modo SQL para atualização bem-sucedida

A configuração `sql_mode` no MySQL/MariaDB deve ter o modo "STRICT_TRANS_TABLES" definido. Se estiver ausente, a atualização do banco de dados Zabbix falhará (veja também ZBX-19435).

Atualização com MariaDB 10.2.1 e anteriores

A atualização do Zabbix pode falhar se as tabelas do banco de dados foram criadas com o MariaDB 10.2.1 ou anterior, porque nessas versões o formato de linha padrão é o compact. Isso pode ser corrigido alterando o formato de linha para dynamic (veja também ZBX-17690).

Templates

Compatibilidade de template em ambientes dual-stack (IPv4/IPv6)

Em ambientes dual-stack (sistemas configurados para suportar tanto IPv4 quanto IPv6), o nome de host `localhost` normalmente é resolvido para endereços IPv4 e IPv6. Devido à priorização comum do IPv6 sobre o IPv4 por muitos sistemas operacionais e resolvedores de DNS, os templates do Zabbix podem não funcionar corretamente se o serviço monitorado estiver configurado para escutar apenas em IPv4.

Serviços que não estão configurados para escutar em endereços IPv6 podem se tornar inacessíveis, levando a falhas no monitoramento. Os usuários podem configurar o acesso corretamente para IPv4, mas ainda assim enfrentar problemas de conectividade devido ao comportamento padrão de priorizar o IPv6.

Uma solução alternativa para isso é garantir que os serviços (Nginx, Apache, PostgreSQL, etc.) estejam configurados para escutar em endereços IPv4 e IPv6, e que o Zabbix server/agent tenha permissão de acesso via IPv6. Além disso, nos templates e configurações do Zabbix, utilize `localhost` explicitamente em vez de `127.0.0.1` para garantir a compatibilidade com IPv4 e IPv6.

Por exemplo, ao monitorar o PostgreSQL com o template PostgreSQL by Zabbix agent 2, pode ser necessário editar o arquivo `pg_hba.conf` para permitir conexões para o usuário `zbx_monitor`. Se o ambiente dual-stack priorizar o IPv6 (o sistema resolve localhost para `::1`) e você configurar `localhost`, mas adicionar apenas uma entrada IPv4 (`127.0.0.1/32`), a conexão falhará porque não há uma entrada correspondente para IPv6.

O exemplo a seguir do arquivo `pg_hba.conf` garante que o usuário `zbx_monitor` possa se conectar a qualquer banco de dados a partir da máquina local usando endereços IPv4 e IPv6 com diferentes métodos de autenticação:

```
### TYPE      DATABASE      USER             ADDRESS          METHOD
  host      all           zbx_monitor      localhost        trust
  host      all           zbx_monitor      127.0.0.1/32     md5
  host      all           zbx_monitor      ::1/128          scram-sha-256
```

Se necessário, você também pode usar o endereço IPv4 (`127.0.0.1`) diretamente ao configurar a macro de string de conexão do template PostgreSQL by Zabbix agent 2.

Instalação acidental de pacotes Zabbix do EPEL

Se o repositório EPEL estiver instalado e habilitado, a instalação de pacotes Zabbix pode puxar versões do EPEL em vez dos pacotes oficiais do Zabbix. Para resolver isso:

1. Remova quaisquer pacotes Zabbix instalados a partir do EPEL:

```
dnf remove zabbix-server-mysql
```

2. Exclua os pacotes Zabbix do EPEL adicionando a seguinte linha ao arquivo `/etc/yum.repos.d/epel.repo`:

```
[epel]
...
excludepkgs=zabbix*
```

3. Reinstale o pacote oficial do Zabbix server:

```
dnf install zabbix-server-mysql
```

Durante a instalação, os pacotes oficiais do Zabbix incluem a palavra `release` em sua string de versão (por exemplo, `7.0.0-release1.el8`), distinguindo-os dos pacotes do EPEL.

Pacotes Zabbix para RHEL em ambientes Red Hat UBI

Ao instalar o Zabbix a partir de pacotes do Red Hat Enterprise Linux em ambientes Red Hat Universal Base Image, garanta o acesso aos repositórios e dependências necessários. Os pacotes do Zabbix dependem das bibliotecas `libOpenIPMI.so` e `libOpenIPMIposix.so`, que não são fornecidas por nenhum pacote nos repositórios padrão do gerenciador de pacotes habilitados em sistemas UBI e resultarão em falhas na instalação.

As bibliotecas `libOpenIPMI.so` e `libOpenIPMIposix.so` estão disponíveis no pacote `OpenIPMI-libs`, que é fornecido pelo repositório `redhat-#-for-<arch>-appstream-rpms`. O acesso a este repositório é controlado por assinaturas, que, no caso de ambientes UBI, são propagadas montando os diretórios de configuração do repositório e de segredos do host RHEL no namespace do sistema de arquivos do container.

Para mais informações, consulte ZBX-24291.

Chave de assinatura expirada para pacotes RHEL

Ao atualizar o Zabbix no Red Hat Enterprise Linux ou em seus derivados, você pode encontrar um problema de chave de assinatura expirada para pacotes no repositório Zabbix. Quando uma chave de assinatura expira, as tentativas de verificar as assinaturas dos pacotes resultarão em um erro indicando que o certificado ou a chave não é mais válida. Por exemplo:

```
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
```

Para resolver esses problemas, reinstale manualmente o pacote `zabbix-release` mais recente para sua variante específica do RHEL (substitua o link abaixo pelo correto do repositório Zabbix).

Por exemplo, no **RHEL 9**, execute:

```
rpm -Uvh https://repo.zabbix.com/zabbix/7.0/rhel/9/x86_64/zabbix-release-latest.el9.noarch.rpm
```

Em seguida, atualize as informações do repositório:

```
dnf update
```

Para mais informações, consulte ZBX-24761.

Conexão TLS com banco de dados MariaDB

A conexão TLS com banco de dados não é suportada com a opção 'verify_ca' para o parâmetro DBTLSConnect parâmetro se MariaDB for usado.

Possíveis deadlocks com MySQL/MariaDB

Ao executar sob alta carga e com mais de um worker LLD envolvido, é possível encontrar um deadlock causado por um erro do InnoDB relacionado à estratégia de bloqueio de linhas (veja o bug upstream). O erro foi corrigido no MySQL a partir da versão 8.0.29, mas não no MariaDB. Para mais detalhes, veja ZBX-21506.

Correlação de eventos globais

Os eventos podem não ser correlacionados corretamente se o intervalo de tempo entre o primeiro e o segundo evento for muito pequeno, ou seja, meio segundo ou menos.

Intervalo do tipo de dado numérico (float) com PostgreSQL 11 e versões anteriores

O PostgreSQL 11 e versões anteriores suportam apenas o intervalo de valores de ponto flutuante de aproximadamente -1.34E-154 a 1.34E+154.

NetBSD 8.0 e mais recentes

Vários processos do Zabbix podem travar aleatoriamente na inicialização nas versões 8.X e 9.X do NetBSD. Isso ocorre devido ao tamanho de pilha padrão ser muito pequeno (4MB), que deve ser aumentado executando:

```
ulimit -s 10240
```

Para mais informações, consulte o relatório de problema relacionado: ZBX-18275.

Limitações de expressões regulares no Zabbix agent 2

O Zabbix agent 2 não suporta lookaheads e lookbehinds em expressões regulares devido às limitações da biblioteca padrão Go regexp.

Verificações IPMI

As verificações IPMI não funcionarão com o pacote padrão da biblioteca OpenIPMI no Debian anterior ao 9 (stretch) e no Ubuntu anterior ao 16.04 (xenial). Para corrigir isso, recompile a biblioteca OpenIPMI com o OpenSSL habilitado, conforme discutido em ZBX-6139.

IPMI — hosts não confiáveis podem travar o OpenIPMI

Existe um bug na biblioteca OpenIPMI usada pelo Zabbix para coletar dados IPMI que pode ser acionado por respostas especialmente criadas de um dispositivo não confiável.
Um dispositivo IPMI não confiável pode enviar dados manipulados que fazem com que a biblioteca OpenIPMI trave, o que, por sua vez, pode fazer com que o processo do Zabbix server que executa a coleta IPMI seja encerrado.

Verificações SSH

- Algumas distribuições Linux como Debian, Ubuntu não suportam chaves privadas criptografadas (com senha) se a biblioteca libssh2 estiver instalada a partir de pacotes. Consulte ZBX-4850 para mais detalhes.
- Ao usar libssh 0.9.x em algumas distribuições Linux com OpenSSH 8, as verificações SSH podem ocasionalmente relatar "Cannot read data from SSH server". Isso é causado por um problema na libssh (relatório mais detalhado). Espera-se que o erro tenha sido corrigido na versão estável libssh 0.9.5. Veja também ZBX-17756 para detalhes.
- Usar o pipe "|" no script SSH pode levar ao erro "Cannot read data from SSH server". Neste caso, recomenda-se atualizar a versão da biblioteca libssh. Veja também ZBX-21337 para detalhes.

Verificações ODBC

- O driver MySQL unixODBC não deve ser usado com o Zabbix server ou Zabbix proxy compilado contra a biblioteca do conector MariaDB e vice-versa. Se possível, também é melhor evitar usar o mesmo conector como driver devido a um bug upstream. Configuração sugerida:

Conector PostgreSQL, SQLite ou Oracle → driver MariaDB ou MySQL unixODBC
 Conector MariaDB → driver MariaDB unixODBC
 Conector MySQL → driver MySQL unixODBC

Veja ZBX-7665 para mais informações e soluções alternativas disponíveis.

- Dados XML consultados do Microsoft SQL Server podem ser truncados de várias maneiras em sistemas Linux e UNIX.
- Foi observado que o uso de verificações ODBC para monitorar bancos de dados Oracle usando várias versões do Oracle Instant Client para Linux faz com que o Zabbix server trave.
 Veja também: ZBX-18402, ZBX-20803.
- Se estiver usando o driver FreeTDS UnixODBC, você precisa adicionar uma instrução 'SET NOCOUNT ON' antes da consulta SQL (por exemplo, `SET NOCOUNT ON DECLARE @strsql NVARCHAR(max) SET @strsql = ....`). Caso contrário, o item de monitoramento do banco de dados no Zabbix falhará ao recuperar as informações com o erro "A consulta SQL retornou resultado vazio".
 Veja ZBX-19917 para mais informações.

Parâmetro de método de requisição incorreto em items

O parâmetro do método de requisição, usado apenas em verificações HTTP, pode estar incorretamente definido como '1', um valor não padrão para todos os items como resultado da atualização de uma versão do Zabbix anterior à 4.0. Para obter detalhes sobre como corrigir essa situação, consulte ZBX-19308.

Monitoramento web e agente HTTP

O Zabbix server apresenta vazamento de memória em algumas distribuições Linux devido a um bug upstream quando a opção "SSL verify peer" está habilitada em cenários web ou no agente HTTP. Consulte ZBX-10486 para mais informações e soluções alternativas disponíveis.

Verificações simples

Existe um bug nas versões do **fping** anteriores à v3.10 que lida incorretamente com pacotes de resposta de eco duplicados. Isso pode causar resultados inesperados para os items `icmpping`, `icmppingloss`, `icmppingsec`. Recomenda-se usar a versão mais recente do **fping**. Consulte ZBX-11726 para mais detalhes.

Erros na execução do fping em containers rootless

Quando os containers estão sendo executados em modo rootless ou em um ambiente com restrições específicas, você pode enfrentar erros relacionados à execução do fping ao realizar verificações ICMP, como `fping: Operation not permitted` ou todos os pacotes para todos os recursos perdidos.

Para corrigir esse problema, adicione `--cap-add=net_raw` aos comandos "docker run" ou "podman run".

Além disso, a execução do fping em ambientes sem root pode exigir a modificação do sysctl, ou seja:

```
sudo sysctl -w "net.ipv4.ping_group_range=0 1995"
```

onde "1995" é o GID do zabbix. Para mais detalhes, consulte ZBX-22833.

Verificações SNMP

Se o sistema operacional OpenBSD for usado, um bug de use-after-free na biblioteca Net-SNMP até a versão 5.7.3 pode causar uma falha no Zabbix server se o parâmetro SourceIP estiver definido no arquivo de configuração do Zabbix server. Como solução alternativa, por favor, não defina o parâmetro SourceIP. O mesmo problema também se aplica ao Linux, mas não faz com que o Zabbix server pare de funcionar. Um patch local para o pacote net-snmp no OpenBSD foi aplicado e será lançado com o OpenBSD 6.3.

Picos de dados SNMP

Foram observados picos em dados SNMP que podem estar relacionados a certos fatores físicos, como picos de tensão na rede elétrica. Veja ZBX-14318 para mais detalhes.

Traps SNMP

O pacote "net-snmp-perl", necessário para traps SNMP, foi removido no RHEL 8.0-8.2; readicionado no RHEL 8.3.

Portanto, se você estiver usando o RHEL 8.0-8.2, a melhor solução é atualizar para o RHEL 8.3.

Veja também ZBX-17192 para mais informações.

Falha do processo de alerta no RHEL 7

Foram encontrados casos de falha do processo de alerta do Zabbix server no RHEL 7. Consulte ZBX-10461 para obter mais detalhes.

Atualizando o Zabbix agent 2 (6.0.5 ou anterior)

Ao atualizar o Zabbix agent 2 (versão 6.0.5 ou anterior) a partir de pacotes, pode ocorrer um erro de conflito de arquivos relacionado a plugins. Para corrigir o erro, faça backup da configuração do agent 2 (se necessário), desinstale o agent 2 e instale-o novamente.

Em sistemas baseados em RHEL, execute:

```
dnf remove zabbix-agent2
dnf install zabbix-agent2
```

Em sistemas baseados em Debian, execute:

```
apt remove zabbix-agent2
apt install zabbix-agent2
```

Para mais informações, consulte ZBX-23250.

Alternância de idiomas no frontend

Foi observado que os idiomas do frontend podem alternar sem lógica aparente, ou seja, algumas páginas (ou partes de páginas) são exibidas em um idioma enquanto outras páginas (ou partes de páginas) em outro idioma. Normalmente, o problema pode aparecer quando há vários usuários, alguns dos quais usam um idioma, enquanto outros usam outro.

Uma solução conhecida para isso é desabilitar o multithreading no PHP e no Apache.

O problema está relacionado a como a configuração de idioma funciona no PHP: as informações de idioma são mantidas por processo, não por thread. Portanto, em um ambiente multithread, quando há vários projetos executados pelo mesmo processo do Apache, é possível que o idioma seja alterado em outra thread e isso mude a forma como os dados podem ser processados na thread do Zabbix.

Para mais informações, consulte os relatórios de problemas relacionados:

- ZBX-10911 (Problema com alternância de idiomas no frontend)
- ZBX-16297 (Problema com processamento de números em gráficos usando a função `bcdiv` das funções BC Math)

Gráficos

Problemas com gráficos (clássicos)

Se você encontrar problemas com gráficos clássicos, recomenda-se atualizar a biblioteca GD (libgd) para a versão 2.3.3-13 ou superior, e o PHP para a versão 8.0.19, 8.1.33, 8.2.29, 8.3.25, 8.4.12 ou superior.

Horário de verão

Alterações no horário de verão (DST) resultam em irregularidades ao exibir os rótulos do eixo X (duplicação de datas, datas ausentes, etc).

Agregação de soma

Ao usar agregação de soma em um gráfico para um período inferior a uma hora, os gráficos exibem valores incorretos (multiplicados) quando os dados vêm de tendências.

Sobreposição de texto

Para alguns idiomas do frontend (por exemplo, japonês), fontes locais podem causar sobreposição de texto na legenda do gráfico. Para evitar isso, use a versão 2.3.0 (ou posterior) da extensão PHP GD.

Monitoramento de arquivos de log

Os items `log[]` e `logrt[]` relêem repetidamente o arquivo de log desde o início se o sistema de arquivos estiver 100% cheio e o arquivo de log estiver sendo anexado (veja ZBX-10884 para mais informações).

Consultas lentas do MySQL

O Zabbix server gera consultas `SELECT` lentas no caso de valores inexistentes para os items. Esse problema é conhecido por ocorrer nas versões 5.6/5.7 do MySQL (para uma discussão mais detalhada, veja ZBX-10652), e, em casos específicos, também pode ocorrer em versões mais recentes do MySQL. Uma solução alternativa para isso é desabilitar o otimizador

`index_condition_pushdown` ou `prefer_ordering_index` no MySQL. Observe, no entanto, que essa solução alternativa pode não corrigir todos os problemas relacionados a consultas lentas.

Sincronização de configuração lenta com Oracle

A sincronização de configuração pode ser lenta em instalações do Zabbix com Oracle DB que possuem um grande número de items e etapas de pré-processamento de items. Isso é causado pela velocidade de processamento do mecanismo de banco de dados Oracle para campos do tipo *nclob*.

Para melhorar o desempenho, você pode converter os tipos de campo de *nclob* para *nvarchar2* aplicando manualmente o patch de banco de dados items_nvarchar_prepare.sql. Observe que essa conversão reduzirá o limite máximo de tamanho do campo de 65535 bytes para 4000 bytes para parâmetros de pré-processamento de items e parâmetros de items como *Description*, campo *Script* do item Script, campos *Request body* e *Headers* do item HTTP agent, campo *SQL query* do item Database monitor. Consultas para determinar os nomes de templates que precisam ser excluídos antes de aplicar o patch são fornecidas no patch como comentário. Como alternativa, se MAX_STRING_SIZE estiver definido, você pode alterar *nvarchar2(4000)* para *nvarchar2(32767)* nas consultas do patch para definir o limite de tamanho do campo para 32767 bytes.

Para uma discussão mais detalhada, consulte ZBX-22363.

Configurações de filtro persistentes a partir de links

Ao abrir um link para uma página do frontend do Zabbix que contenha configurações de filtro, incluindo o seletor de tempo, o filtro é salvo automaticamente no banco de dados para o usuário, substituindo as configurações de filtro e/ou seletor de tempo salvas anteriormente para essa página. Essas configurações permanecem ativas até que o usuário as atualize ou redefina manualmente.

Problema de endereço IPv6 em traps SNMPv3

Devido a um bug no net-snmp, o endereço IPv6 pode não ser exibido corretamente ao usar SNMPv3 em traps SNMP. Para mais detalhes e uma possível solução alternativa, consulte ZBX-14541.

Endereço IP IPv6 longo aparado nas informações de login com falha

Uma mensagem de tentativa de login com falha exibirá apenas os primeiros 39 caracteres de um endereço IP armazenado, pois esse é o limite de caracteres no campo do banco de dados. Isso significa que endereços IP IPv6 com mais de 39 caracteres serão exibidos de forma incompleta.

Verificações do agent Zabbix no Windows

Entradas DNS inexistentes em um parâmetro `Server` no arquivo de configuração do agent Zabbix (zabbix_agentd.conf) podem aumentar o tempo de resposta do agent Zabbix no Windows. Isso acontece porque o daemon de cache DNS do Windows não armazena respostas negativas para endereços IPv4. No entanto, para endereços IPv6, as respostas negativas são armazenadas em cache, então uma possível solução alternativa para isso é desabilitar o IPv4 no host.

Exportação/importação YAML

Existem alguns problemas conhecidos com a exportação/importação YAML:

- As mensagens de erro não são traduzíveis;
- JSON válido com uma extensão de arquivo .yaml às vezes não pode ser importado;
- Datas legíveis por humanos não citadas são automaticamente convertidas em timestamps Unix.

Assistente de configuração no SUSE com NGINX e php-fpm

O assistente de configuração do frontend não pode salvar o arquivo de configuração no SUSE com NGINX + php-fpm. Isso é causado por uma configuração na unidade /usr/lib/systemd/system/php-fpm.service, que impede o Zabbix de gravar em /etc. (introduzido no PHP 7.4).

Existem duas opções de solução alternativa disponíveis:

- Defina a opção ProtectSystem como 'true' em vez de 'full' na unidade systemd do php-fpm.
- Salve manualmente o arquivo /etc/zabbix/web/zabbix.conf.php.

Encaminhamento do cabeçalho de autorização

Em alguns casos, o Apache ou o NGINX podem impedir que o cabeçalho Authorization em solicitações de API chegue ao Zabbix. Isso pode causar problemas de autenticação ao usar a API do Zabbix ou serviços de logon único (SSO), como SAML com Okta.

Para resolver isso, atualize a configuração do seu servidor web.

Para o **Apache**, se você estiver usando-o como um proxy reverso (configuração não-CGI), adicione a seguinte diretiva em `/etc/httpd/conf/httpd.conf` (em sistemas baseados em RHEL) ou `/etc/apache2/apache2.conf` (em Debian/Ubuntu):

```
SetEnvIfNoCase ^Authorization$ "(.+)" HTTP_AUTHORIZATION=$1
```

Se o Apache executar scripts diretamente para lidar com solicitações (por exemplo, usando mod_cgi), adicione a seguinte diretiva:

```
CGIPassAuth On
```

Por outro lado, o **NGINX** lida automaticamente com o cabeçalho Authorization. No entanto, se o NGINX estiver atuando como um proxy reverso, você pode encaminhar explicitamente o cabeçalho Authorization adicionando as seguintes diretivas em `/etc/nginx/nginx.conf` (para o local do seu frontend Zabbix):

```
...
location / {
...
    proxy_set_header Authorization $http_authorization;
    proxy_pass http://backend_server;
...
}
```

Após atualizar a configuração, reinicie o seu servidor web.

Para mais detalhes, consulte:

- ZBX-22952
- Apache 2.4 + PHP-FPM e cabeçalhos de autorização
- Diretrizes SetEnvIfNoCase e CGIPassAuth
- NGINX Reverse Proxy

Chromium para o serviço web do Zabbix no Ubuntu 20

Embora na maioria dos casos o serviço web do Zabbix possa ser executado com o Chromium, no Ubuntu 20.04 o uso do Chromium causa o seguinte erro:

```
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
```

Esse erro ocorre porque `/var/lib/zabbix` é usado como diretório home do usuário 'zabbix'.

Códigos de erro personalizados do MySQL

Quando o Zabbix detecta que o banco de dados backend está inacessível, ele envia uma notificação e continua tentando se conectar. Para certos mecanismos de banco de dados, códigos de erro específicos são reconhecidos. No MySQL, esses códigos de erro reconhecidos incluem:

- 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

Além disso, ao usar o Zabbix com uma instalação MySQL no Azure, a mensagem de erro genérica *[9002] Some errors occurred* pode aparecer nos logs do Zabbix. Esta mensagem é enviada pelo banco de dados para o servidor ou proxy Zabbix. Para determinar a causa do erro, consulte os logs do Azure.

Expressões regulares inválidas após a mudança para PCRE2

No Zabbix 6.0, o suporte ao PCRE2 foi adicionado. Embora o PCRE ainda seja suportado, os pacotes de instalação do Zabbix para RHEL 7 e mais recentes, SLES (todas as versões), Debian 9 e mais recentes, Ubuntu 16.04 e mais recentes foram atualizados para usar o PCRE2. Embora forneça muitos benefícios, a mudança para o PCRE2 pode fazer com que certos padrões de regexp do PCRE existentes se tornem inválidos ou se comportem de maneira diferente. Em particular, isso afeta o padrão `^[\w-\.]`. Para tornar essa regexp válida novamente sem afetar a semântica, altere a expressão para `^[-\w\.]` . Isso acontece devido ao fato de que o PCRE2 trata o sinal de traço como um delimitador, criando um intervalo dentro de uma classe de caracteres.

Erro no widget Geomap

Os mapas no widget Geomap podem não carregar corretamente se você atualizou de uma versão mais antiga do Zabbix com NGINX e não mudou para o novo arquivo de configuração do NGINX durante a atualização.

Para corrigir o problema, você pode descartar o arquivo de configuração antigo, usar o arquivo de configuração do pacote da versão atual e reconfigurá-lo conforme descrito nas instruções de download na seção *e. Configure o PHP para o frontend do Zabbix*.

Como alternativa, você pode editar manualmente um arquivo de configuração NGINX existente (normalmente, */etc/zabbix/nginx.conf*). Para isso, abra o arquivo e localize o seguinte bloco:

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

Em seguida, substitua este bloco por:

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

location /vendor {
        deny            all;
        return          404;
}
```

Pré-processamento — variáveis globais não são seguras

O JavaScript de pré-processamento é executado por solicitação, mas atribuições a identificadores não declarados (por exemplo, `secret = value`) criam variáveis globais implícitas que podem persistir além da execução atual. Armazenar dados sensíveis (tokens, senhas, etc.) em variáveis globais implícitas aumenta o risco de exposição acidental ou reutilização por execuções subsequentes de pré-processamento ou outras integrações executando no mesmo ambiente.

Não confie em variáveis globais implícitas. Sempre declare variáveis com `var` ou `const` e evite anexar segredos a objetos globais (por exemplo, `globalThis` ou `window`). Não há uma maneira suportada de substituir objetos globais internos a partir do pré-processamento.

Exemplo seguro:

```
var apiToken = payload.token;
var count = 1;
return JSON.stringify({ token: apiToken, calls: count });
```

Falha do server com TimescaleDB após atualização da versão 7.0

Atualizar para o Zabbix 7.0.1 (ou posterior) a partir do Zabbix 7.0.0 com TimescaleDB resulta em uma falha do server. Esse problema é causado por uma solução alternativa para um problema de job de compressão na tabela auditlog no Zabbix 7.0 que alterou irreversivelmente a política de compressão da tabela auditlog.

Para corrigir o problema, execute uma reconstrução manual da tabela auditlog. A tabela auditlog com problema pode ser detectada usando esta consulta:

```
SELECT config FROM timescaledb_information.jobs WHERE application_name LIKE 'Compression%' AND hypertable_
```

Se retornar um objeto JSON contendo a propriedade `compress_after` (como {"hypertable_id": 14, "compress_after": 612000}) então você deve reconstruir a tabela.

> **Attention:**
> Certifique-se de que o Zabbix server esteja pelo menos na versão 7.0.1rc2 (ou posterior); caso contrário, ele definirá novamente a política de compressão incorreta. Além disso, pare o Zabbix server antes de executar o script e confirme que `auditlog` pertence ao usuário *zabbix*.

A maneira mais simples de reconstruir a tabela auditlog é:

```
CREATE TABLE auditlog_tmp (
LIKE auditlog INCLUDING DEFAULTS INCLUDING CONSTRAINTS INCLUDING INDEXES
);

SELECT create_hypertable('auditlog_tmp', 'auditid', chunk_time_interval => 604800,
time_partitioning_func => 'cuid_timestamp', migrate_data => true, if_not_exists => true);

WITH moved_rows AS (
DELETE FROM auditlog
RETURNING *
)
```

```
INSERT INTO auditlog_tmp
SELECT * FROM moved_rows;

DROP TABLE auditlog;
ALTER TABLE auditlog_tmp RENAME TO auditlog;
```

Veja também a documentação do TimescaleDB para formas mais otimizadas de migrar dados.

> **Note:**
> Como o timestamp necessário para o particionamento é extraído do campo `auditid` com uma função personalizada, os procedimentos auxiliares usados para migração de dados do timescaledb-extras não funcionarão.

Erro de restauração do banco de dados com PostgreSQL/TimescaleDB após atualização da versão 7.0.0-7.0.4

Usar `pg_restore` para restaurar um backup do PostgreSQL ou TimescaleDB criado no Zabbix 7.0.0-7.0.4 resultará em um erro de função `base36_decode` ausente, fazendo com que a restauração falhe:

```
ERROR:  function base36_decode(text) does not exist
LINE 1: CAST(base36_decode(substring(cuid FROM 2 FOR 8))/1000 AS int...
             ^
HINT:  No function matches the given name and argument types. You might need to add explicit type casts.
```

Esse erro ocorre ao restaurar um backup criado com `pg_dump`.

Para corrigir esse problema, substitua a função `cuid_timestamp` no seu banco de dados Zabbix **antes** de criar o backup (é recomendável parar o PostgreSQL/TimescaleDB antes de executar o script):

```
CREATE OR REPLACE FUNCTION cuid_timestamp(cuid varchar(25)) RETURNS integer AS $$
DECLARE
    base36 varchar;
    a char[];
    ret bigint;
    i int;
    val int;
    chars varchar;
BEGIN
    base36 := substring(cuid FROM 2 FOR 8);

    chars := '0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ';

    FOR i IN REVERSE char_length(base36)..1 LOOP
        a := a || substring(upper(base36) FROM i FOR 1)::char;
    END LOOP;
    i := 0;
    ret := 0;
    WHILE i < (array_length(a, 1)) LOOP
        val := position(a[i + 1] IN chars) - 1;
        ret := ret + (val * (36 ^ i));
        i := i + 1;
    END LOOP;

    RETURN CAST(ret/1000 AS integer);
END;
$$ LANGUAGE 'plpgsql' IMMUTABLE;
DROP FUNCTION IF EXISTS base36_decode(character varying);
```

Veja também ZBX-24955 (para detalhes adicionais sobre o erro) e documentação do TimescaleDB (para opções adicionais de backup e restauração).

Grupos de processadores no Windows

A documentação da Microsoft afirma que sistemas com menos de 64 processadores lógicos sempre têm um único grupo de processadores, o Grupo 0. No entanto, usuários do Zabbix relataram um bug raro ZBX-20260, quando há dois grupos de processadores em sistemas com 64 ou menos processadores lógicos. Isso resultou em ter os contadores de desempenho "\Processor(n)" para apenas um grupo de processadores entre dois. A causa raiz real desse bug não é conhecida. No entanto, um caso semelhante foi descrito em stackoverflow.com, e a causa raiz lá estava na interoperabilidade entre BIOS e Windows.

Limites de filtragem com collations utf8mb4

Os filtros (por exemplo, em *Coleta de dados* > *Manutenção*) podem não funcionar corretamente quando aplicados a entidades que contêm certos caracteres Unicode (por exemplo, 𝓔, 𝔈). Esse problema ocorre devido à forma como a collation padrão utf8mb4_bin para bancos de dados MySQL ou MariaDB lida com a ordenação e comparação de caracteres Unicode.

Para contornar essa limitação, os usuários podem alterar a collation das colunas do banco de dados para alternativas como utf8mb4_0900_bin, utf8mb4_0900_ai_ci ou utf8mb4_unicode_520_ci. Observe, no entanto, que a alteração da collation pode causar comportamentos inesperados no tratamento de espaços em branco, bem como na ordenação e filtragem de outros caracteres.

Para mais informações sobre como alterar collations, consulte a documentação do MySQL ou a documentação do MariaDB. Para detalhes sobre diferenças entre collations, consulte Unicode Character Sets na documentação do MySQL.

Informações incorretas de grupos de hosts aninhados em mapas

As informações de grupos de hosts aninhados são exibidas incorretamente em mapas, por exemplo:

- O rótulo do grupo de hosts exibe o resumo do problema sem incluir todos os hosts nos grupos de hosts aninhados;
- A visualização "Elementos do grupo de hosts" não exibe um elemento de mapa separado para cada host nos grupos de hosts aninhados;
- O rótulo do mapa exibe o resumo de todos os problemas sem incluir aqueles nos grupos de hosts aninhados.

Substituições de regras LLD quebradas no 7.0.7

Na versão 7.0.7, o Zabbix server trava ao processar substituições de regras de descoberta de baixo nível. Como solução alternativa, desative as regras LLD que contenham substituições. O problema foi corrigido no Zabbix 7.0.8rc2.

Problema de desempenho do gerenciador de pré-processamento no Zabbix 7.0.14

Na versão 7.0.14, o gerenciador de pré-processamento do Zabbix pode causar alta utilização da CPU quando vários valores são recebidos para um único item de uma só vez (como durante o monitoramento de logs) e mais de um worker de pré-processamento está configurado. Como solução temporária, defina o parâmetro de configuração `StartPreprocessors` do server/proxy como 1. O problema foi corrigido no Zabbix 7.0.15.

Funções de macro

As funções de macro não funcionam nos parâmetros de item de script e nos parâmetros de script de item de navegador. Corrigido no Zabbix 7.0.7.

Acesso aos elementos da interface com MariaDB 10.5.1–10.5.9

Acessar o frontend web do Zabbix com uma função diferente de Super Admin pode resultar na mensagem: "Ocorreu um erro no sistema. Por favor, entre em contato com o administrador do Zabbix.". Esse problema afeta instalações que utilizam as versões do MariaDB de 10.5.1 até 10.5.9.

Para evitar esse problema, atualize o MariaDB para uma versão superior à 10.5.9. Para mais detalhes, consulte ZBX-25746.

Perfilando uso excessivo de memória com tcmalloc

Se você suspeita que sua instalação do Zabbix está usando muita memória, pode usar o recurso de profiling de memória do tcmalloc para investigar o consumo de memória do Zabbix server/proxy.

1. Ao instalar o Zabbix a partir do código-fonte, configure flags adicionais:

```
export CFLAGS="-std=gnu99 -g -O0"
```

A flag `-std=gnu99` é necessária para compilar o Zabbix server, Zabbix proxy ou Zabbix agent. A flag `-g` adiciona informações extras de depuração, enquanto `-O0` desativa otimizações, que podem interferir no profiling do tcmalloc.

2. Defina as seguintes variáveis de ambiente antes de iniciar o Zabbix server. Essas variáveis informam ao tcmalloc como rastrear e relatar o uso de memória:

```
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. Dispare um dump de perfil enviando o sinal 5 para o processo de destino. Substitua 1234 pelo ID real do processo (PID):

```
kill -5 1234
```

4. Imprima o perfil gerado:

```
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
```

Neste exemplo, zbx_malloc2 é responsável por quase todas as alocações de memória.

Veja também:

- ZBX-25050 e ZBX-25584 para relatórios de problemas relacionados.
- Resumo das opções do GCC sobre opções de compilação (`-std=gnu99`, `-g`, `-O0`, etc.).
- Documentação do Gperftools Heap Profiler sobre variáveis de ambiente para profiling do tcmalloc.

MySQL 8.0 Group Replication em modo multi-primary

Ao usar o MySQL 8.0 Group Replication em modo multi-primary, você pode encontrar um erro durante a confirmação de transações semelhante ao seguinte:

```
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
```

Esse erro parece ser causado por problemas com operações de rollback envolvendo restrições de chave estrangeira.

Veja também:

- ZBX-26060 para o relatório de problema relacionado.
- MySQL Bug #96758 "Rollbacks with Foreign Keys on single node" para o problema upstream.

1 Problemas de compilação

Estes são os problemas conhecidos relacionados à compilação do Zabbix a partir do código-fonte. Para todos os outros casos, consulte a página de Problemas conhecidos.

Compilando o agent do Zabbix no HP-UX

Se você instalar a biblioteca PCRE a partir do popular site de pacotes HP-UX http://hpux.connect.org.uk (por exemplo, a partir do arquivo `pcre-8.42-ia64_64-11.31.depot`), apenas a versão 64 bits da biblioteca será instalada no diretório `/usr/local/lib/hpux64`.

Neste caso, para uma compilação bem-sucedida do agent, uma opção personalizada é necessária para o script `configure`, por exemplo:

```
CFLAGS="+DD64" ./configure --enable-agent --with-libpcre-include=/usr/local/include --with-libpcre-lib=/us
```

Biblioteca em um local não padrão

O Zabbix permite que você especifique uma biblioteca localizada em um local não padrão. No exemplo abaixo, o Zabbix executará o `curl-config` a partir do local não padrão especificado e usará sua saída para determinar a libcurl correta a ser usada.

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

Isso funcionará se for a única libcurl instalada no sistema, mas pode não funcionar se houver outra libcurl instalada em um local padrão (pelo gerenciador de pacotes, por exemplo). Isso ocorre quando você precisa de uma versão mais recente da biblioteca para o Zabbix e de uma mais antiga para outros aplicativos.

Portanto, especificar um componente em um local não padrão nem sempre funcionará quando o mesmo componente também existir em um local padrão.

Por exemplo, se você usar uma libcurl mais recente instalada em `/usr/local` com o pacote libcurl ainda instalado, o Zabbix pode escolher a errada e a compilação falhará:

```
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
```

Aqui, a função `curl_easy_header()` não está disponível na antiga `/usr/lib/x86_64-linux-gnu/libcurl.so`, mas está disponível na nova `/usr/local/lib/libcurl.so`.

O problema está na ordem das flags do linker, e uma solução é especificar o caminho completo para a biblioteca em uma variável LDFLAGS:

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

Observe a opção `-Wl,--no-as-needed`, que pode ser necessária em alguns sistemas (veja também: opções de link padrão em sistemas baseados em Debian).

Tamanho da pilha muito pequeno em alguns sistemas

Se o Zabbix travar ou congelar devido a estouros de pilha, você pode aumentar o tamanho da pilha por thread usando a opção `--with-stacksize` ao configurar os fontes. Esse problema pode ocorrer em sistemas com limites de pilha de thread padrão baixos, especialmente durante o pré-processamento, onde várias threads são criadas.

O exemplo a seguir define o tamanho da pilha para 512 KB por thread:

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

Você pode verificar os limites de pilha de thread do sistema em tempo de execução usando o comando `ulimit -s` em sistemas baseados em Linux.

2 Problemas de atualização relacionados ao escape

Limite de caracteres excedido para parâmetros de função após a atualização

A escape adequada de barras invertidas foi adicionada nos parâmetros de string da função de histórico no Zabbix 7.0.

Como barras invertidas adicionais são adicionadas durante a atualização de versões do Zabbix anteriores à 7.0, isso leva a parâmetros mais longos, o que pode resultar em funções de trigger quebradas se o comprimento do parâmetro exceder o tamanho máximo de dados de 255 caracteres.

Para ilustrar quais problemas podem surgir durante a atualização do banco de dados, qual é o seu impacto, como reconhecê-los e corrigi-los, vamos tomar um exemplo de configuração com 3 items e 2 triggers.

Configuração

1. Versão 6.0 do Zabbix.
2. Arquivo `/tmp/ONE` que contém 114 barras invertidas com 'a' no final:

```
\\\
```

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

```
\\\
```

Templates atualizados

Os templates Zabbix server health, Remote Zabbix server health, Zabbix proxy health e Remote Zabbix proxy health foram atualizados para melhorar a representação visual, transferindo as visualizações dos dados dos items dos gráficos para os widgets do dashboard e reagrupando as métricas exibidas.

Alterações na 7.0.2 Novos templates

- AWS Lambda por HTTP, um template para monitorar métricas do AWS Lambda.

Alterações na 7.0.4 Novos templates

- Repositório GitHub por HTTP, um template para monitorar repositórios GitHub.
- Relatórios Microsoft 365 por HTTP, um template para monitorar serviços Microsoft 365.

Templates atualizados

Os templates dentro dos conjuntos de templates VMware e VMware FQDN foram atualizados e complementados com várias melhorias, incluindo as seguintes:

- novas regras de LLD ”VMware overall health discovery” e ”VMware alarm discovery”, bem como novos dashboards para todos os templates VMware;
- novos items sobre estatísticas de rede - como parte do template VMware Hypervisor;
- novos items para obter informações sobre snapshots e triggers para espaço livre em disco, bem como uma atualização para o trigger “VM has been restarted” - como parte do template VMware Guest.

Alterações na 7.0.5 Novos templates

- Huawei OceanStor V6 por SNMP, um template para monitorar SAN Huawei OceanStor V6.

Templates atualizados

- Em todos os templates APC, a regra de descoberta para pacotes de baterias externas defeituosas foi complementada com um filtro para evitar a criação de items não suportados.
- Nos templates MSSQL por Zabbix agent 2 e MSSQL por ODBC, um timeout foi adicionado e certas chaves de item foram ajustadas para aumentar a estabilidade do monitoramento.

Alterações na 7.0.6 Novos templates

- Nutanix Prism Element por HTTP, um conjunto de templates que também inclui os templates *Nutanix Cluster Prism Element por HTTP* e *Nutanix Host Prism Element por HTTP*.

Além disso, vários templates de agent ativo foram adicionados para o monitoramento da pilha LAMP:

- Apache por Zabbix agent ativo
- MySQL por Zabbix agent ativo
- MySQL por Zabbix agent 2 ativo
- Nginx por Zabbix agent ativo
- PHP-FPM por Zabbix agent ativo
- PostgreSQL por Zabbix agent ativo
- PostgreSQL por Zabbix agent 2 ativo

Templates atualizados

- O template *VMware Hypervisor* dentro dos conjuntos de templates VMware e VMware FQDN foi atualizado com a opção de ser usado como um template independente.
- Proxmox VE por HTTP foi atualizado com novos items e protótipos de trigger para uso de espaço em disco em containers LXC.
- No template MSSQL por ODBC, a consulta do contador de desempenho foi atualizada para melhorar a recuperação de dados. Tanto a documentação do template *MSSQL por ODBC* quanto do MSSQL por Zabbix agent 2 foi revisada e atualizada.

Alterações na 7.0.7 Templates atualizados

- Todos os templates foram atualizados para reintroduzir os prefixos de nome de trigger que foram removidos no Zabbix 7.0.0.

Alterações na 7.0.8 Novos templates

- Juniper MX por SNMP, um template para a série Juniper MX.

Alterações na 7.0.10 Novos templates

- O conjunto de templates Azure por HTTP foi complementado com o template *Azure SQL Managed Instance por HTTP*.
- Palo Alto PA-440 por HTTP, um template que utiliza items de script para coletar métricas usando a API PAN-OS XML.

Templates atualizados

- Todos os templates Dell HTTP e SNMP foram atualizados com melhorias, incluindo correções para items, macros e scripts.
- Os templates MSSQL por ODBC e MSSQL por Zabbix agent 2 foram atualizados com filtros para incluir ou excluir membros de quorum descobertos por nome, bem como um filtro de serviço que permite filtrar por nome de cluster para excluir clusters vazios. Consulte Alterações incompatíveis para informações adicionais sobre *MSSQL por Zabbix agent 2*.

Alterações na 7.0.11 Novos templates

- O conjunto de templates Azure por HTTP foi complementado com o template *Azure Microsoft SQL DTU Database por HTTP*.
- O conjunto de templates Zabbix server health foi complementado com os templates *Zabbix server health por Zabbix agent*, *Zabbix server health por Zabbix agent active*, e Zabbix proxy health - com *Zabbix proxy health por Zabbix agent* e *Zabbix proxy health por Zabbix agent active*, permitindo o monitoramento de métricas internas do Zabbix via Zabbix agent.

Modelos atualizados

- Os modelos nos conjuntos de modelos VMware e VMware FQDN foram atualizados com correções para protótipos de item de datastore. No modelo *VMware Guest*, um multiplicador de pré-processamento incorreto foi corrigido no item para utilização de rede de máquina virtual, e um problema foi corrigido na regra de descoberta para sistemas de arquivos montados.

Alterações na 7.0.12 Novos templates

- Pure Storage FlashArray v1 e v2 por HTTP, um conjunto de templates para monitorar sistemas Pure Storage FlashArray.

Templates atualizados

- MySQL por agent Zabbix, MySQL por agent 2 Zabbix, MySQL por agent ativo Zabbix, MySQL por agent 2 ativo Zabbix e MySQL por ODBC foram atualizados com uma correção para a fórmula de eficiência do buffer pool.
- No template MSSQL por ODBC, o item ”Obter último backup” foi atualizado para excluir bancos de dados inexistentes da consulta.

Alterações na 7.0.13 Novos templates

- O conjunto de templates Azure por HTTP foi complementado com o template *Azure Backup Jobs por HTTP*.

Templates atualizados

- PostgreSQL por Zabbix agent 2 foi atualizado para compatibilidade com o PostgreSQL 17 e agora utiliza a nova visualização do sistema `pg_stat_checkpointer` para estatísticas de checkpoint (anteriormente `pg_stat_bgwriter`).

Alterações na 7.0.14 Templates atualizados

- Oracle por ODBC e Oracle por Zabbix agent 2 foram atualizados para consultar estatísticas de PDB e tablespace usando o ID do container em vez do nome do container.

Alterações na 7.0.17 Novos templates

- Cisco Secure Firewall Threat Defense por HTTP, um template que fornece recursos de monitoramento para dispositivos Cisco Secure Firewall Threat Defense usando a REST API.

Templates atualizados

- O template *VMware Guest* dentro dos conjuntos de templates VMware e VMware FQDN foi atualizado com um novo item *Boot time*. Este item agora substitui *Uptime of guest OS* no trigger *VM has been restarted* para evitar falsos positivos após migrações de máquinas virtuais, como vMotion.

Alterações na 7.0.18 Novos templates

- O conjunto de templates AWS por HTTP foi complementado com o template *AWS Backup Vault por HTTP*.

Alterações na 7.0.19 Novos templates

- Aruba CX 8300s por SNMP, um template que fornece monitoramento baseado em SNMP para a série de switches Aruba CX 8300.
- Ciena 3906 por SNMP, um template para monitorar dispositivos Ciena 3906.
- Vyatta Virtual Router por SNMP, um template para monitorar o roteador virtual Vyatta 1908e.

Templates atualizados

- Huawei OceanStor Dorado por SNMP foi atualizado com detalhes sobre a cobertura do dispositivo.
- Os conjuntos de templates Zabbix server health e Zabbix proxy health foram atualizados com capacidades estendidas de monitoramento de throughput do pré-processador.

Alterações na 7.0.20 Novos templates

- Stormshield SNS por SNMP, um template para monitorar dispositivos Stormshield Network Security (SNS) via SNMP.

Alterações na 7.0.22 Novos templates

- Ribbon SBC Edge por HTTP, um template para monitorar dispositivos Ribbon SBC Edge (anteriormente SWe Lite) via HTTP.
- Ribbon SBC SWe Core por HTTP, um template para monitorar dispositivos SBC SWe Core via HTTP, e Ribbon SBC SWe CE por HTTP, um template para monitorar instâncias Ribbon SBC SWe Call Engine (CE) via HTTP.
- VeloCloud SD-WAN Edge por HTTP, um template para monitorar dispositivos VeloCloud SD-WAN Edge via HTTP.

Templates atualizados

- MySQL por Zabbix agent, MySQL por Zabbix agent 2, MySQL por Zabbix agent ativo, MySQL por Zabbix agent 2 ativo e MySQL por ODBC foram atualizados para suportar tanto `SHOW SLAVE STATUS` (sintaxe antiga) quanto `SHOW REPLICA STATUS` (nova sintaxe).
- VeloCloud SD-WAN por HTTP, anteriormente *VMWare SD-WAN VeloCloud por HTTP*, foi renomeado e atualizado para alinhar-se à última plataforma VeloCloud SD-WAN.

Alterações na 7.0.23 Modelos atualizados

- Ciena 3906 por SNMP foi atualizado para incluir itens de sistema de arquivos e carga da CPU.
- Vyatta Virtual Router por SNMP foi atualizado com novos itens de OID e melhorias no dashboard.

11 Notas de atualização para 7.0.0

Estas notas são para atualização do Zabbix 6.4.x para o Zabbix 7.0.0.

Todas as notas estão agrupadas em:

- **Alterações incompatíveis** - alterações que podem quebrar instalações existentes e outras informações críticas relacionadas ao processo de atualização
- **Outras** - todas as demais informações descrevendo as mudanças na funcionalidade do Zabbix

Veja também:

- Procedimento de atualização para todas as informações relevantes sobre atualização a partir de versões anteriores ao Zabbix 6.4.0;
- Atualizando cluster HA para instruções sobre atualização de servers em um cluster de **alta disponibilidade** (HA).

Processo de atualização

Para concluir uma atualização bem-sucedida do Zabbix server no MySQL/MariaDB, pode ser necessário definir `GLOBAL log_bin_trust_function_creators = 1` no MySQL se o log binário estiver ativado, não houver privilégios de superusuário e `log_bin_trust_function_creators = 1` não estiver definido no arquivo de configuração do MySQL.

Para definir a variável usando o console do MySQL, execute:

```
mysql> SET GLOBAL log_bin_trust_function_creators = 1;
```

Após a conclusão bem-sucedida da atualização, esta opção pode ser desativada:

```
mysql> SET GLOBAL log_bin_trust_function_creators = 0;
```

Mudanças incompatíveis Falha do servidor com TimescaleDB após atualização para 7.0.1 a partir de 7.0.0

Atualizar para o Zabbix 7.0.1 (ou posterior) a partir do Zabbix 7.0.0 com TimescaleDB resulta em uma falha do servidor. Esse problema é causado por uma solução alternativa para um problema de job de compressão na tabela auditlog no Zabbix 7.0 que altera irreversivelmente a política de compressão da tabela auditlog.

Veja problemas conhecidos para detalhes de como corrigir manualmente a tabela auditlog.

Se você ainda não atualizou para o Zabbix 7.0.0, observe que não deve haver tais problemas para atualizações de versões do Zabbix anteriores à 7.0 para a 7.0.1. Para instruções de atualização, consulte Atualizando o schema do TimescaleDB.

Erro de restauração do banco de dados com PostgreSQL/TimescaleDB após atualização da versão 7.0.0-7.0.4

Usar o `pg_restore` para restaurar um backup do PostgreSQL ou TimescaleDB criado no Zabbix 7.0.0-7.0.4 resultará em um erro de função `base36_decode` ausente, fazendo com que a restauração falhe. Esse erro ocorre ao restaurar um backup criado com `pg_dump`.

Veja problemas conhecidos para detalhes sobre como corrigir esse problema manualmente **antes** de criar o backup.

Se você ainda não atualizou para o Zabbix 7.0.0, observe que não deve haver tais problemas para atualizações de versões anteriores ao Zabbix 7.0 para a 7.0.5. Para instruções de atualização, consulte Atualizando o schema do TimescaleDB.

Versão mínima do PHP necessária

A versão mínima do PHP necessária foi alterada de 7.4.0 para 8.0.0.

Comandos remotos em agents ativos

Comandos remotos agora também podem ser executados em um agent 7.0 que está operando apenas no modo ativo.

Observe que, para scripts que são executados apenas em um agent ativo, é provável que o timeout padrão do server ou do proxy precise ser aumentado. O timeout do server/proxy deve ser maior do que a frequência de atualização da verificação ativa, caso contrário, o timeout do server/proxy pode ser excedido antes mesmo que o agent ativo receba o script.

Escape de barras invertidas nos parâmetros da função de histórico

O escape adequado de barras invertidas foi adicionado nos parâmetros de string da função de histórico.

Como barras invertidas adicionais são adicionadas durante a atualização de versões do Zabbix anteriores à 7.0, isso leva a parâmetros mais longos, o que pode resultar em funções de trigger quebradas se o comprimento do parâmetro exceder o tamanho máximo de dados de 255 caracteres.

Para evitar esse problema, sugere-se mover manualmente parâmetros longos para macros de usuário antes da atualização.

Observe que, desde o Zabbix 7.0.11, verificações adicionais para o comprimento do parâmetro resultante foram adicionadas para atualizações de versões do Zabbix anteriores à 7.0. Se o comprimento resultante exceder o tamanho máximo, tais parâmetros **não são atualizados**, enquanto uma mensagem de aviso é impressa nos logs pedindo aos usuários que corrijam manualmente os parâmetros listados.

Veja também problemas de atualização relacionados ao escape.

Local padrão para o arquivo de configuração do agent no Windows

O local padrão onde o agent do Zabbix no Windows procura o arquivo de configuração foi alterado. Agora o agent o procura no diretório onde o binário do agent zabbix_agentd.exe está localizado (em vez de `C:\zabbix_agentd.conf`, como anteriormente).

O agent 2 do Zabbix no Windows já procurava o arquivo de configuração padrão no diretório onde o binário zabbix_agent2.exe está localizado. No entanto, na nova versão o agent 2 espera que o arquivo de configuração seja nomeado como `zabbix_agent2.conf` (em vez de `zabbix_agent2.win.conf`)

Veja também: Instalando o agent do Zabbix no Windows.

Valores vazios permitidos na configuração do plugin do agent 2

Agora, valores vazios são permitidos nos parâmetros de configuração relacionados ao plugin no agent 2 do Zabbix.

TimescaleDB 1.x não é mais suportado

O suporte ao TimescaleDB 1.x foi removido.

Tipos de dados de precisão dupla no TimescaleDB

Antes de atualizar para o Zabbix 7.0.0, é necessário atualizar manualmente o TimescaleDB para usar tipos de dados de precisão dupla se o TimescaleDB for usado com compressão. Você pode saber se o TimescaleDB não está usando tipos de dados de precisão dupla pelo aviso na seção Informações do sistema do frontend ou no log do server do Zabbix: "O banco de dados não foi atualizado para usar valores de precisão dupla. O suporte para o tipo numérico antigo será removido em versões futuras."

Para mais informações, consulte Atualizando para valores numéricos de faixa estendida.

Auditlog convertido em hypertable no TimescaleDB

A tabela `auditlog` foi convertida em hypertable no TimescaleDB em novas instalações para se beneficiar do particionamento automático por tempo (7 dias por padrão) e melhor desempenho.

Para atualizar com sucesso instalações existentes, consulte Atualizando o schema do TimescaleDB.

Veja também: Versões do TimescaleDB suportadas

Tabela de banco de dados separada para proxies

Os registros de proxy foram movidos da tabela `hosts` e agora são armazenados na nova tabela `proxy`.

Além disso, os dados operacionais dos proxies (como último acesso, versão, compatibilidade) foram movidos da tabela `host_rtdata` e agora são armazenados na nova tabela `proxy_rtdata`.

Há também um novo objeto **proxy** na API. Todas as operações com proxies devem ser atualizadas para serem feitas por meio desse novo objeto proxy.

Tempo limite de execução da consulta para itens de monitoramento de banco de dados

Com base nas alterações na configuração de tempo limite do item, tanto o tempo limite de login ODBC quanto o tempo limite de execução da consulta para itens de monitoramento de banco de dados agora são limitados ao valor do parâmetro *Timeout* definido no formulário de configuração do item.

Itens de agent/agent2 atualizados

- `wmi.get` e `wmi.getall`, quando usados com o Zabbix agent 2, agora retornam um JSON com valores booleanos representados como strings (por exemplo, `"RealTimeProtectionEnabled": "True"` em vez de `"RealTimeProtectionEnabled": true` retornado anteriormente) para corresponder ao formato de saída desses itens no Zabbix agent;
- `oracle.ts.stats` possui um novo parâmetro **conname** para especificar o nome do container de destino. O formato JSON dos dados retornados foi atualizado. Quando nenhum **tablespace**, **type** ou **conname** for especificado nos parâmetros da chave, os dados retornados incluirão um nível adicional de JSON com o nome do container, permitindo a diferenciação entre containers.
- Itens `net.dns.*` não podem mais ser configurados sem o parâmetro **name**. Embora sempre listado como obrigatório, o parâmetro **name**, se omitido, anteriormente era resolvido para um valor padrão (zabbix.com), o que não é mais o caso.

Para a lista de alterações de itens que não quebram a compatibilidade, consulte Novidades no Zabbix 7.0.0.

Traps SNMP com alta disponibilidade do Zabbix

O Zabbix agora pode ler arquivos de trap SNMP do local correto caso o nó ativo seja alternado em uma configuração de alta disponibilidade.

No entanto, para que essa funcionalidade funcione, é necessário atualizar o formato de data/hora em todos os scripts bash, perl e SNMPTT para "%Y-%m-%dT%H:%M:%S%z" (ou seja, 2024−01−10T11:56:14+0300).

Tamanho máximo e número de widgets do dashboard aumentados

A largura padrão foi aumentada 3 vezes para todos os widgets. Observe que, se você estiver usando widgets personalizados, talvez seja necessário atualizar os respectivos parâmetros do arquivo *manifest.json* (por exemplo, ao configurar um widget *Clock* personalizado, o parâmetro `width` deve ser alterado de 4 para 12).

Agora, um widget pode ter até 72 colunas de largura (anteriormente 24) e de 1 a 64 linhas de altura (anteriormente de 2 a 32). Portanto, o dashboard agora pode conter até 72 widgets horizontalmente.

Histórico de item e widgets de Texto simples

O novo widget de dashboard *Histórico de item* substituiu o widget *Texto simples*, oferecendo várias melhorias.

Ao contrário do widget *Texto simples*, que exibia apenas o dado mais recente do item em texto simples, o widget *Histórico de item* suporta várias opções de exibição para múltiplos tipos de item (numérico, caractere, log, texto e binário). Por exemplo, ele pode mostrar barras de progresso ou indicadores, imagens para tipos de dados binários (útil para itens de navegador) e destacar valores de texto (útil para monitoramento de arquivos de log).

Após a atualização, todos os widgets *Texto simples* previamente configurados serão automaticamente substituídos por widgets *Histórico de item*, mantendo as mesmas configurações. No entanto, quaisquer scripts de API que referenciem o widget *Texto simples* devem ser atualizados manualmente.

Alterações na resolução de certas macros internas após a atualização para 7.0.4

No Zabbix 7.0.4, o comportamento de resolução das macros `{HOST.CONN}`, `{HOST.DNS}`, `{HOST.IP}`, `{EVENT.CAUSE.*}`, `{EVENT.TAGS.*}` e `{EVENT.SYMPTOMS}` foi corrigido para garantir consistência com outras macros internas. Agora, elas serão resolvidas como `*UNKNOWN*` se ocorrer um erro ao avaliar seu valor.

Como resultado dessa alteração, para garantir o funcionamento adequado, é necessário atualizar os tipos de mídia para as seguintes integrações: iTop, Jira Service Desk, ManageEngine ServiceDesk, SolarWinds Service Desk.

Essa alteração também pode afetar quaisquer scripts personalizados ou integrações que utilizem as macros mencionadas acima.

Mudanças na API

Veja a lista de mudanças na API no Zabbix 7.0.0.

Outros Oracle DB obsoleto

O suporte ao Oracle como banco de dados backend está obsoleto desde o Zabbix 7.0 e espera-se que seja completamente removido em versões futuras.

Verificação de atualização de software

Uma verificação de atualização de software agora é adicionada por padrão a novas instalações e instalações existentes - o frontend do Zabbix se comunicará com o endpoint público do Zabbix para verificar atualizações.

Você pode desabilitar essa verificação definindo AllowSoftwareUpdateCheck=0 na configuração do server.

Valores de ponto flutuante truncados para itens inteiros

Agora, se um valor de ponto flutuante for recebido para um item inteiro sem sinal, o valor será truncado da parte decimal e salvo como um inteiro. Anteriormente, um valor de ponto flutuante tornava um item inteiro não suportado.

Formato de hora dos EUA

As exibições de hora e data no frontend agora estão em conformidade com o padrão de exibição de hora/data dos EUA quando o idioma padrão (en_US) do frontend é usado.

Antes	Agora
14:43:26	02:43:26 PM

Pollers assíncronos

Após a atualização, todas as verificações de agent, HTTP agent e SNMP `walk[OID]` serão movidas para pollers assíncronos.

Detectando recursos da biblioteca cURL em tempo de execução

Anteriormente, os recursos da biblioteca cURL eram detectados no momento da compilação do Zabbix server, proxy ou agent. Se os recursos do cURL fossem atualizados, para utilizá-los, o respectivo componente do Zabbix precisava ser recompilado.

Agora, apenas um reinício é necessário para que os recursos atualizados da biblioteca cURL fiquem disponíveis no Zabbix. A recompilação não é mais necessária. Isso é válido para o Zabbix server, proxy ou agent.

Além disso:

- A versão mínima exigida do cURL em tempo de compilação foi elevada para 7.19.1;
- Ao ser compilado, o Zabbix apenas verifica se a biblioteca cURL está disponível (se solicitado) e atende ao requisito de versão;
- Ao emitir um erro da biblioteca cURL em tempo de execução, a versão em uso será adicionada (por exemplo, "A biblioteca cURL não suporta SSL/TLS (usando a versão 7.88.1)");
- Quando o Zabbix server é iniciado e a autenticação SMTP é registrada, ele escreverá a disponibilidade da autenticação SMTP da biblioteca cURL em uso.

Alterações na configuração do timeout do item

Após a atualização, os timeouts globais para todos os tipos de item suportados serão definidos com base no valor do parâmetro `Timeout` do arquivo de configuração do server. Se um proxy estiver configurado, por padrão, ele usará as configurações globais de timeout do server.

Ao usar um server atualizado (versão 7.0.0 ou mais recente) com um proxy ou agent mais antigo, o proxy ou agent funcionará como antes:

- o proxy usará o parâmetro `Timeout` do arquivo de configuração do proxy;
- o agent usará o parâmetro `Timeout` do arquivo de configuração do agent.

Os parâmetros de timeout foram removidos dos arquivos de configuração dos plugins Modbus e MQTT. Os timeouts de execução das requisições agora podem ser definidos usando o formulário de configuração do item.

Itens de navegador

Um novo tipo de item - Item de navegador - foi adicionado ao Zabbix, permitindo o monitoramento de sites e aplicações web complexas usando um navegador. Itens de navegador permitem a execução de código JavaScript definido pelo usuário para simular ações relacionadas ao navegador, como clicar, inserir texto, navegar por páginas da web, etc.

Além disso, este recurso adiciona as seguintes alterações:

- O template Website by Browser foi adicionado aos templates prontos para uso;

- O tipo de item ITEM_TYPE_BROWSER (22) foi adicionado à exportação/importação de configuração de item, regra de descoberta de baixo nível e protótipo de item de template ou host;
- Os parâmetros de arquivo de configuração do Zabbix server/proxy `StartBrowserPollers` e `WebDriverURL` foram adicionados;
- O timeout do item de navegador foi adicionado aos timeouts do proxy ou aos timeouts globais (se um proxy não for usado);
- O parâmetro de linha de comando `-w <webdriver url>` para habilitar o monitoramento de navegador foi adicionado à utilidade de linha de comando `zabbix_js`.

Concorrência na descoberta de rede

Na nova versão, o processo de descoberta de rede foi reformulado para permitir a concorrência entre as verificações de serviço. Um novo processo de gerenciador de descoberta foi adicionado juntamente com um número configurável de workers de descoberta (ou threads). O gerenciador de descoberta processa as regras de descoberta e cria um job de descoberta para cada regra com tarefas (verificações de serviço). As verificações de serviço são coletadas e executadas pelos workers de descoberta.

O parâmetro StartDiscoverers agora determina o número total de workers de descoberta disponíveis para a descoberta. O número padrão de StartDiscoverers foi aumentado de 1 para 5, e o intervalo de 0-250 para 0-1000. Os processos `discoverer` das versões anteriores do Zabbix foram removidos.

Além disso, o número de workers disponíveis para cada regra agora é configurável no frontend. Este parâmetro é opcional. Durante a atualização, ele será definido como "Um", como nas versões anteriores do Zabbix.

Ícones substituídos por fontes

Todos os ícones no frontend foram trocados de imagens para fontes.

Filtro de dados mais recentes

Em *Monitoramento* → *Dados mais recentes*, o subfiltro e os dados não são mais exibidos por padrão se o filtro não estiver definido. Observe, no entanto, que os filtros salvos anteriormente que foram definidos usando apenas o subfiltro permanecem inalterados. Nesses casos, o subfiltro permanecerá visível e os dados serão exibidos mesmo sem que o filtro principal seja definido.

Parâmetros de configuração

O valor padrão de vários parâmetros de configuração foi alterado:

- O parâmetro de configuração BufferSize para o Zabbix agent 2 foi aumentado de 100 para 1000;
- O parâmetro de configuração Plugins.<PluginName>.System.Capacity para o Zabbix agent 2 foi aumentado de 100 para 1000 (máximo). Observe que o parâmetro `Plugins.<PluginName>.Capacity`, obsoleto no Zabbix 6.0, foi removido completamente;
- O parâmetro de configuração StartAgents para o Zabbix agent foi aumentado de 3 para 10. Observe que, para sistemas menores (ou seja, Raspberry Pi), o valor padrão pode permanecer 3.

Essas alterações não afetam instalações existentes onde esses parâmetros estão definidos explicitamente.

Cálculos agregados

Várias funções agregadas foram atualizadas. Agora:

- As funções agregadas agora também suportam tipos não numéricos para cálculo. Isso pode ser útil, por exemplo, com as funções count e count_foreach.
- As funções agregadas count e count_foreach suportam os parâmetros opcionais *operator* e *pattern*, que podem ser usados para ajustar a filtragem de items e contar apenas valores que correspondam a determinados critérios.
- Todas as funções foreach não incluem mais items não suportados na contagem.
- A função **last_foreach**, anteriormente configurada para ignorar o argumento de período de tempo, agora o aceita como parâmetro opcional.

Tipo de valor numérico antigo (float) removido

Desde o Zabbix 5.0, o tipo de dado numérico (float) suporta precisão de aproximadamente 15 dígitos e intervalo de aproximadamente -1,79E+308 a 1,79E+308. O tipo numérico antigo foi desabilitado, mas ainda era suportado. A partir do Zabbix 6.4, ele foi depreciado e, a partir do Zabbix 7.0, foi totalmente removido.

Se sua instalação não foi atualizada para usar valores numéricos de intervalo estendido, o seguinte aviso é exibido em *Relatórios* → *Informações do sistema*: "Tabelas de histórico do banco de dados atualizadas: Não. O suporte ao tipo numérico antigo está depreciado. Atualize para valores numéricos de intervalo estendido".

Ao atualizar para o Zabbix 7.0, os seguintes bancos de dados são atualizados **automaticamente**:

- MySQL
- PostgreSQL/TimescaleDB (sem compressão)
- Oracle

No entanto, para Oracle, versões antigas do MySQL e instalações grandes, recomenda-se atualizar o tipo de dado manualmente antes de atualizar para o Zabbix 7.0.

Para TimescaleDB com compressão, o banco de dados deve ser atualizado **manualmente**.

Para detalhes sobre a atualização, consulte Atualizando para valores numéricos de intervalo estendido.

Definindo o tipo de inicialização do serviço do agent no Windows

A opção para definir o tipo de inicialização do serviço do agent/agent 2 do Zabbix no Windows (`-S --startup-type`) foi adicionada. Esta opção permite configurar o serviço do agent/agent 2 para iniciar automaticamente na inicialização do Windows (`automatic`), após a inicialização dos serviços iniciados automaticamente ter sido concluída (`delayed`), quando iniciado manualmente por um usuário ou aplicativo (`manual`) ou para desabilitar o serviço completamente (`disabled`).

Ao realizar a instalação do agent do Windows a partir do MSI, o tipo de inicialização padrão no Windows Server 2008/Vista e versões posteriores agora é `delayed` se não for especificado de outra forma no parâmetro de linha de comando `STARTUPTYPE` parâmetro de linha de comando. Isso melhora a confiabilidade e o desempenho do serviço do agent/agent 2 do Zabbix no Windows, especialmente durante reinicializações do sistema.

Templates

Para novos templates e alterações em templates existentes, consulte Alterações em templates.

Nova estrutura de diretórios para arquivos relacionados ao banco de dados

Ao instalar o Zabbix a partir de pacotes e preparar o schema do banco de dados, a localização dos arquivos relacionados ao banco de dados foi alterada para corresponder melhor à estrutura de arquivos nas fontes:

- **Arquivos de inicialização base** (`schema.sql`, `data.sql`, `images.sql`) estão localizados na raiz do diretório do banco de dados.
- **Arquivos/patches opcionais** para atualização de tabelas do banco de dados estão localizados no diretório `option-patches`.
- **Extensões e complementos do banco de dados** agora são subdiretórios, nomeados de acordo com a extensão e localizados no respectivo diretório do banco de dados.
- **Alterações específicas do TimescaleDB:**
  - A abreviação `tsdb` foi substituída por `timescaledb`.
  - O diretório `option-patches` inclui os subdiretórios `with-compression` e `without-compression`; estes contêm arquivos/patches opcionais para atualização de tabelas do banco de dados dependendo das configurações de compressão do TimescaleDB.
  - O arquivo de criação do schema hypertable para o TimescaleDB foi movido para `database/postgresql/timescaledb/schema`

Além disso, observe as seguintes alterações:

- Todos os arquivos `history_pk_*.sql` foram renomeados para `history_upgrade_*.sql`, e um arquivo `trends_upgrade.sql` foi adicionado para melhorar os scripts de atualização do banco de dados.
- O arquivo `double.sql` foi removido devido à remoção do antigo tipo de valor numérico (float).

Abaixo está uma comparação da estrutura de diretórios anterior e atual para os bancos de dados MySQL e PostgreSQL.

```
### Anterior:

database
  mysql
    data.sql
    double.sql
    history_pk_prepare.sql
    images.sql
    schema.sql


  postgresql
    tsdb_history_pk_upgrade_no_compression
      history_pk.sql
      history_pk_log.sql
      history_pk_str.sql
      history_pk_text.sql
      history_pk_uint.sql
    tsdb_history_pk_upgrade_with_compression
      history_pk.sql
      history_pk_log.sql
      history_pk_str.sql
```

```
 # Atual:

database
 mysql
   option-patches
     history_upgrade_prepare.sql
   data.sql
   images.sql
   schema.sql


 postgresql
   option-patches
     history_upgrade_prepare.sql
   timescaledb
     option-patches
       with-compression
         history_upgrade.sql
          history_upgrade_log.sql
         history_upgrade_str.sql
         history_upgrade_text.sql
         history_upgrade_uint.sql
```

```
        history_pk_text.sql                    trends_upgrade.sql
        history_pk_uint.sql                  without-compression
      data.sql                                   history_upgrade.sql
      double.sql                                 history_upgrade_log.sql
      history_pk_prepare.sql                     history_upgrade_str.sql
      images.sql                                 history_upgrade_text.sql
      schema.sql                                 history_upgrade_uint.sql
      timescaledb.sql                            trends_upgrade.sql
                                               schema.sql
                                             data.sql
                                             images.sql
                                             schema.sql
    ...                                    ...
```

Atualize seus scripts, se houver, que contenham referências à estrutura de diretórios ou arquivos anteriores.

Protocolo JSON para verificações passivas do agent

Um protocolo baseado em JSON para verificações passivas do agent foi implementado.

Para compatibilidade com agents mais antigos, foi adicionado um failover para o antigo protocolo de texto simples. Se o agent retornar "ZBX_NOTSUPPORTED", o Zabbix irá armazenar em cache a interface como protocolo antigo e tentar novamente a verificação enviando apenas a chave do item em texto simples.

O Zabbix get agora pode ser executado com uma nova opção `-P --protocol <valor>`, onde "valor" pode ser:

- auto - conecta usando o protocolo JSON, faz fallback e tenta novamente com o protocolo de texto simples (padrão);
- json - conecta usando a chave do protocolo JSON;
- plaintext - conecta usando o protocolo de texto simples, onde apenas a chave do item é enviada.

Se uma chave de item não for suportada, o Zabbix get agora retornará o código de saída 1 em vez de 0.

12 Notas de atualização para 7.0.1

Mudanças incompatíveis Falha do server com TimescaleDB após atualização da versão 7.0

Atualizar para o Zabbix 7.0.1 (ou posterior) a partir do Zabbix 7.0.0 com TimescaleDB resulta em uma falha do server. Esse problema é causado por uma solução alternativa para um problema de compressão na tabela auditlog no Zabbix 7.0 que alterou irreversivelmente a política de compressão da tabela auditlog.

Veja problemas conhecidos para detalhes de como corrigir a tabela auditlog manualmente.

Observe que não deve haver tais problemas para atualizações de versões do Zabbix anteriores à 7.0 para a 7.0.1 (ou posterior). Para instruções de atualização, consulte Atualizando o schema do TimescaleDB.

Erro de restauração do banco de dados com PostgreSQL/TimescaleDB após atualização da versão 7.0.0-7.0.4

Usar o `pg_restore` para restaurar um backup do PostgreSQL ou TimescaleDB criado no Zabbix 7.0.0-7.0.4 resultará em um erro de função `base36_decode` ausente, fazendo com que a restauração falhe. Esse erro ocorre ao restaurar um backup criado com `pg_dump`.

Consulte problemas conhecidos para obter detalhes sobre como corrigir esse problema manualmente **antes** de criar o backup.

Se você ainda não atualizou para o Zabbix 7.0.0, observe que não deve haver tais problemas para atualizações de versões anteriores à 7.0 do Zabbix para a 7.0.5. Para instruções de atualização, consulte Atualizando o schema do TimescaleDB.

Outros Novo índice na tabela auditlog

Um novo índice foi adicionado à tabela `auditlog` para melhorar os tempos de resposta do banco de dados e do frontend ao filtrar registros por *ID do conjunto de registros* no Log de auditoria.

Observe que usuários com grandes logs de auditoria podem experimentar tempos de atualização prolongados devido ao tamanho do banco de dados.

Versão mínima exigida do TimescaleDB

A versão mínima exigida do TimescaleDB agora é 2.13.0.

O server será interrompido com banco de dados somente leitura

Um server Zabbix independente agora será interrompido se o banco de dados se tornar somente leitura.

Items Monitoramento de log de eventos do VMware

O item vmware.eventlog agora também retorna informações do usuário.

Desempenho do serviço DNS

O item net.dns.perf agora retorna um tempo de resposta em vez de 0 quando o servidor DNS responde com um código de erro (por exemplo, `NXDOMAIN` ou `SERVFAIL`).

13 Notas de atualização para 7.0.2

Mudanças incompatíveis Histórico de dados binários convertido para hypertable no TimescaleDB

A tabela `history_bin` foi convertida para hypertable no TimescaleDB para se beneficiar do particionamento automático por tempo (1 dia por padrão) e melhor desempenho.

Para atualizar com sucesso instalações existentes, consulte Atualizando o esquema do TimescaleDB.

Veja também: Versões do TimescaleDB suportadas

Erro de restauração do banco de dados com PostgreSQL/TimescaleDB após atualização da versão 7.0.0-7.0.4

Usar `pg_restore` para restaurar um backup do PostgreSQL ou TimescaleDB criado no Zabbix 7.0.0-7.0.4 resultará em um erro de função `base36_decode` ausente, fazendo com que a restauração falhe. Esse erro ocorre ao restaurar um backup criado com `pg_dump`.

Consulte problemas conhecidos para obter detalhes sobre como corrigir esse problema manualmente **antes** de criar o backup.

Se você ainda não atualizou para o Zabbix 7.0.0, observe que não deve haver tais problemas para atualizações de versões do Zabbix anteriores à 7.0 para a 7.0.5. Para instruções de atualização, consulte Atualizando o schema do TimescaleDB.

14 Notas de atualização para 7.0.3

Mudanças incompatíveis Erro de restauração do banco de dados com PostgreSQL/TimescaleDB após atualização da versão 7.0.0-7.0.4

Usar o `pg_restore` para restaurar um backup do PostgreSQL ou TimescaleDB criado no Zabbix 7.0.0-7.0.4 resultará em um erro de função `base36_decode` ausente, fazendo com que a restauração falhe. Esse erro ocorre ao restaurar um backup criado com `pg_dump`.

Consulte problemas conhecidos para obter detalhes sobre como corrigir esse problema manualmente **antes** de criar o backup.

Se você ainda não atualizou para o Zabbix 7.0.0, observe que não deve haver tais problemas para atualizações de versões do Zabbix anteriores à 7.0 para a 7.0.5. Para instruções de atualização, consulte Atualizando o schema do TimescaleDB.

Outros Patch do banco de dados removendo índice redundante

Um patch opcional do banco de dados foi adicionado para remover um índice redundante `userdirectory_usrgrp_3` durante a atualização.

15 Notas de atualização para 7.0.4

Mudanças incompatíveis Alterações na resolução de certas macros internas

O comportamento de resolução das macros `{HOST.CONN}`, `{HOST.DNS}`, `{HOST.IP}`, `{EVENT.CAUSE.*}`, `{EVENT.TAGS.*}` e `{EVENT.SYMPTOMS}` foi corrigido para garantir consistência com outras macros internas. Agora, elas serão resolvidas como `*UNKNOWN*` se ocorrer um erro ao avaliar seu valor.

Como resultado dessa alteração, para garantir o funcionamento adequado, é necessário atualizar os tipos de mídia para as seguintes integrações: iTop, Jira Service Desk, ManageEngine ServiceDesk, SolarWinds Service Desk.

Essa alteração também pode afetar quaisquer scripts personalizados ou integrações que utilizem as macros mencionadas acima.

Erro de restauração do banco de dados com PostgreSQL/TimescaleDB após atualização da versão 7.0.0-7.0.4

Usar o `pg_restore` para restaurar um backup do PostgreSQL ou TimescaleDB criado no Zabbix 7.0.0-7.0.4 resultará em um erro de função `base36_decode` ausente, fazendo com que a restauração falhe. Esse erro ocorre ao restaurar um backup criado com `pg_dump`.

Veja problemas conhecidos para detalhes sobre como corrigir esse problema manualmente **antes** de criar o backup.

Se você ainda não atualizou para o Zabbix 7.0.0, observe que não deve haver tais problemas para atualizações de versões anteriores ao Zabbix 7.0 para a 7.0.5. Para instruções de atualização, consulte Atualizando o schema do TimescaleDB.

Requisitos libevent 2.0.10 necessário

A versão mínima necessária do libevent foi aumentada para 2.0.10.

Bancos de dados Alterações no banco de dados

Um problema de lentidão na clonagem de template foi resolvido com a adição de índices para o campo uuid. Essa alteração pode causar longos tempos de atualização em grandes conjuntos de dados.

Novo índice na tabela auditlog

Um novo índice foi adicionado à tabela `auditlog` para melhorar os tempos de resposta do banco de dados e do frontend ao filtrar registros por *IP* no Log de auditoria.

Observe que usuários com grandes logs de auditoria podem experimentar tempos de atualização prolongados devido ao tamanho do banco de dados.

Processos Problemas não são mais armazenados em cache sem serviços habilitados

O gerenciador de serviços não armazena mais problemas em cache se não houver serviços habilitados.

O gerenciador de serviços verificará se há serviços habilitados no cache de configuração durante a inicialização e armazenará problemas em cache apenas se houver serviços habilitados; ele também detectará quando o primeiro serviço for adicionado/o último serviço removido e inicializará/liberará o cache de problemas.

16 Notas de atualização para 7.0.5

Mudanças incompatíveis Erro de restauração do banco de dados com PostgreSQL/TimescaleDB após atualização da versão 7.0.0-7.0.4

Usar o `pg_restore` para restaurar um backup do PostgreSQL ou TimescaleDB criado no Zabbix 7.0.0-7.0.4 resultará em um erro de função `base36_decode` ausente, fazendo com que a restauração falhe. Esse erro ocorre ao restaurar um backup criado com `pg_dump`.

Veja problemas conhecidos para detalhes sobre como corrigir esse problema manualmente **antes** de criar o backup.

Se você ainda não atualizou para o Zabbix 7.0.0, observe que não deve haver tais problemas para atualizações de versões do Zabbix anteriores à 7.0 para a 7.0.5. Para instruções de atualização, consulte Atualizando o schema do TimescaleDB.

Outros Suporte ao pacote Agent 2 para RHEL 6 removido

Desde o Zabbix 7.0.5, os pacotes do Zabbix agent 2 não estão mais disponíveis para RHEL 6 devido ao Go 1.22 (e mais recente) ser incompatível com o GCC 4.4.7 fornecido pelo RHEL 6, o que resulta em erros de compilação. Observe, no entanto, que os pacotes do Zabbix agent ainda são suportados.

Widget de histórico de item

As quebras de linha agora são observadas ao exibir um valor textual com retornos de linha no widget *Histórico de item* com a opção de exibição definida como ”Como está”. Além disso, não foi implementada quebra de linha no caso de exibição ”Como está” com layout horizontal ou layout vertical com uma coluna. Com layout vertical e mais de uma coluna, a quebra de linha é observada.

17 Notas de atualização para 7.0.6

Cálculo do estado do poller assíncrono

Os pollers assíncronos agora são considerados ocupados se atingirem o limite definido pelo parâmetro de configuração MaxConcurrentChecksPerPoller do server/proxy. A utilização dos processos de poller pode ser monitorada com o item interno zabbix[process,,].

StartPreprocessors padrão aumentado

O valor padrão do parâmetro de configuração StartPreprocessors do server/proxy foi aumentado de 3 para 16. Para aproveitar ao máximo os recursos disponíveis, o número de instâncias pré-iniciadas de threads de trabalho de pré-processamento deve ser definido para corresponder pelo menos ao número de núcleos de CPU disponíveis. Se as tarefas de pré-processamento não forem limitadas pela CPU e envolverem solicitações de rede frequentes, é recomendada a configuração de trabalhadores adicionais.

18 Notas de atualização para 7.0.7

Atenção! Nesta versão, o Zabbix server trava ao processar regras de descoberta de baixo nível com substituições. Como solução alternativa, desative as regras de LLD que contenham substituições. O problema foi corrigido no Zabbix 7.0.8rc2.

Recomendamos atualizar para o Zabbix 7.0.8 o mais rápido possível.

19 Notas de atualização para 7.0.8

Esta versão menor não possui notas de atualização.

20 Notas de atualização para 7.0.9

Mudanças incompatíveis Java 11 necessário para o Java gateway

O Zabbix Java gateway agora requer Java 11 para execução (a compilação a partir do código-fonte ainda é possível com Java 8), devido à atualização das versões mínimas das bibliotecas logback:

Biblioteca	Nova versão mínima	Versão mínima antiga
logback-classic	**1.5.16**	1.2.9
logback-core	**1.5.16**	1.2.9
slf4j-api	**2.0.16**	1.7.32

21 Notas de atualização para 7.0.10

Mudanças incompatíveis Atualização do plugin MSSQL Zabbix agent 2

O template MSSQL por Zabbix agent 2 foi atualizado com filtros para incluir ou excluir membros de quórum descobertos por nome, bem como um filtro de serviço que permite filtrar por nome do cluster para excluir clusters vazios. Para que o template funcione sem erros, o plugin MSSQL Zabbix agent 2 deve ser atualizado para uma versão igual ou superior à 7.0.10.

Outros Manipulação aprimorada da resolução de DNS para agentes de polling

Nesta versão, o agente de polling tentará se conectar a todos os endereços retornados pela pesquisa de DNS. Essa melhoria garante que, se um endereço IP estiver inacessível, o agente de polling possa tentar o próximo, aumentando a probabilidade de uma conexão bem-sucedida.

22 Notas de atualização para 7.0.11

Resolução de DNS assíncrona

O agent poller e o SNMP poller agora podem ser configurados para usar a biblioteca *c-ares* (versão 1.16.0 ou mais recente) em vez do resolvedor de DNS *libevent*, proporcionando uma resolução de DNS potencialmente mais rápida e flexível.

Se você deseja usar a biblioteca c-ares, configure o Zabbix com a opção `--with-ares`:

```
  --with-ares[=ARG]       use c-ares library [default=no],
  --with-ares-include=DIR use c-ares include headers from given path.
  --with-ares-lib=DIR     use c-ares libraries from given path.
```

Observe que, a partir do c-ares 1.31.0, o cache de DNS está ativado por padrão.

Escape de barras invertidas nos parâmetros da função de histórico

O escape adequado de barras invertidas foi adicionado nos parâmetros de string da função de histórico desde o Zabbix 7.0.0.

Como barras invertidas adicionais são adicionadas durante a atualização de versões do Zabbix anteriores à 7.0, isso leva a parâmetros mais longos, o que pode resultar em funções de trigger quebradas se o comprimento do parâmetro exceder o tamanho máximo de dados de 255 caracteres.

Para evitar esse problema, sugere-se mover manualmente os parâmetros longos para macros de usuário antes da atualização.

Observe que, desde o Zabbix 7.0.11, verificações adicionais para o comprimento resultante do parâmetro foram adicionadas para atualizações de versões do Zabbix anteriores à 7.0. Se o comprimento resultante exceder o tamanho máximo, esses parâmetros **não são atualizados**, enquanto uma mensagem de aviso é impressa nos logs solicitando aos usuários que corrijam manualmente os parâmetros listados.

Veja também problemas de atualização relacionados ao escape.

23 Notas de atualização para 7.0.12

Esta versão menor não possui notas de atualização.

24 Notas de atualização para 7.0.13

Resolução de macro secreta

Os valores das macros de usuário secretas agora permanecerão mascarados nos locais que suportam a resolução de macro secreta se forem referenciados indiretamente. Por exemplo, as macros internas {ITEM.KEY}, {ITEM.KEY<1-9>}, {LLDRULE.KEY} usadas em tipos de mídia (parâmetros de Script ou Webhook) serão resolvidas para chaves de item contendo macros secretas mascaradas, como `net.tcp.port[******,******]` em vez de `net.tcp.port[192.0.2.0,80]`.

Cálculo de disponibilidade da interface

O cálculo do status geral de disponibilidade da interface do host agora é afetado apenas por itens do tipo agent Zabbix, agent IPMI, agent JMX ou agent SNMP. Outros tipos de item que usam a interface, como verificações simples, não podem alterar o status da interface, portanto, agora estão excluídos do cálculo do status geral.

25 Notas de atualização para 7.0.14

Warning:
Nesta versão, o gerenciador de pré-processamento do Zabbix pode causar alta utilização da CPU quando vários valores são recebidos para um único item ao mesmo tempo (como durante o monitoramento de logs) e mais de um worker de pré-processamento está configurado. Como solução temporária, defina o parâmetro de configuração `StartPreprocessors` do server/proxy como 1. O problema foi corrigido no Zabbix 7.0.15.

 Recomendamos atualizar para o Zabbix 7.0.15 o mais rápido possível.

SNMP walk e get com tentativas aumentadas

SNMP agent `walk` e `get` agora fazem até 5 tentativas (em vez de 1) para recuperar todos os valores se as tentativas anteriores expirarem ou falharem. Cada tentativa redefine o tempo limite e retoma a partir da última solicitação bem-sucedida, melhorando a confiabilidade dos dados em caso de perda de pacotes ou respostas atrasadas durante a coleta de dados SNMP.

Considere definir um tempo limite baixo no formulário de configuração do item para evitar longos atrasos se o dispositivo estiver inacessível (por exemplo, um tempo limite de 3 segundos pode resultar em um tempo de espera de 15 segundos).

Versão mínima do Go suportada

A versão mínima do Go suportada foi aumentada de 1.21 para 1.23.

Se você compilou anteriormente o agent 2 do Zabbix, plugins carregáveis do agent 2 ou o web service a partir do código-fonte usando uma versão do Go mais antiga que 1.23, é recomendável recompilar esses componentes usando uma versão do Go suportada para receber as últimas atualizações de segurança e correções de bugs. Componentes compilados com uma versão mais antiga do Go continuarão funcionando, mas atualizá-los exigirá um ambiente Go mais recente.

Parâmetros de conexão com o banco de dados

DBPort e DBSocket agora são mutuamente exclusivos. Ao especificar os parâmetros de conexão com o banco de dados, você pode definir DBPort ou DBSocket, ou deixar ambos indefinidos para usar os valores padrão.

26 Notas de atualização para 7.0.15

Esta versão menor não possui notas de atualização.

27 Notas de atualização para 7.0.16

Esta versão secundária não possui notas de atualização.

28 Notas de atualização para 7.0.17

Saída de controle de tempo de execução do Agent 2

Os comandos de controle de tempo de execução do Zabbix agent 2 (`zabbix_agent2 -R <option>`) agora escrevem a saída em `stdout` (saída padrão) em vez de `stderr` (erro padrão). Atualize quaisquer scripts que dependam do comportamento anterior.

29 Notas de atualização para 7.0.18

Esta versão menor não possui notas de atualização.

30 Notas de atualização para 7.0.19

Esta versão menor não possui notas de atualização.

31 Notas de atualização para 7.0.20

Atenção! Não é recomendada a atualização para esta versão devido a erros relacionados a gráficos e ao plugin MySQL do Zabbix agent 2.

32 Notas de atualização para 7.0.21

Esta versão menor não possui notas de atualização.

33 Notas de atualização para 7.0.22

Mudanças incompatíveis Permissões restritas para arquivos de log

Os arquivos de log dos componentes do Zabbix agora são criados com permissões de leitura e gravação apenas para o proprietário do arquivo. Além disso, os arquivos são legíveis pelo grupo do proprietário. Todas as outras permissões são negadas.

No Linux, essa alteração é de permissões 0666 para 0640. A alteração afeta os arquivos de log do server, proxy e agent do Zabbix. As permissões para os arquivos de log do agent 2 do Zabbix foram atualizadas de 0644 para 0640 (removendo a permissão de leitura para outros).

Além disso, vários **outros arquivos** agora são criados com permissões 0640:

- Arquivos pid UNIX para processos do Zabbix
- Arquivos de exportação de dados
- Arquivos temporários para verificações `icmpping*`
- Arquivos log persistent

Outros Consultas de status de replicação do MySQL tratadas pelo Zabbix agent 2

O plugin MySQL do Zabbix agent 2 agora suporta tanto `SHOW SLAVE STATUS` quanto `SHOW REPLICA STATUS`. O plugin seleciona e executa automaticamente a instrução apropriada dependendo da versão do servidor e retorna o resultado no formato JSON usual.

Itens do agent 2 como `mysql.replication.discovery` e `mysql.replication.get_slave_status` funcionam com servidores que usam `SHOW SLAVE STATUS` ou `SHOW REPLICA STATUS`. Não são necessárias alterações nos itens existentes do agent 2 ou nos templates do agent 2.

Versão mínima do Go suportada

A versão mínima do Go suportada foi elevada de 1.23 para 1.24.10.

Se você compilou anteriormente o agent 2 do Zabbix, plugins carregáveis do agent 2 ou o serviço web a partir do código-fonte usando uma versão do Go anterior à 1.24.10, é recomendável recompilar esses componentes usando uma versão do Go suportada para receber as últimas atualizações de segurança e correções de bugs. Os componentes compilados com uma versão mais antiga do Go continuarão funcionando, mas atualizá-los exigirá um ambiente Go mais recente.

Incluir arquivos ou pastas na configuração do serviço web

O parâmetro Include foi adicionado ao arquivo de configuração do serviço web do Zabbix. Este parâmetro permite incluir arquivos individuais ou todos os arquivos em um diretório na configuração.

34 Notas de atualização para 7.0.23

Esta versão menor não possui notas de atualização.

35 Notas de atualização para 7.0.24

Valores de item exibidos como HTML usando iframes

Os dados do item no widget Histórico do item, quando exibidos como texto formatado em HTML, agora são isolados em iframes.

Biblioteca SAML PHP Toolkit atualizada

A biblioteca SAML PHP Toolkit (fornecida com o Zabbix) foi atualizada da versão 4.0.0 para a 4.3.1.

36 Notas de atualização para 7.0.25

Esta versão menor não possui notas de atualização.

5. Início rápido

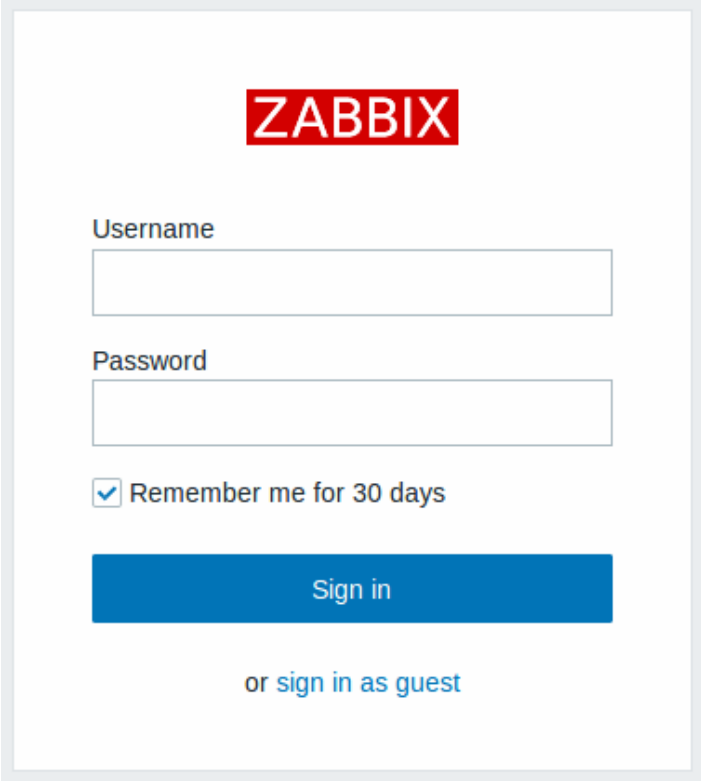
Por favor utilize a barra lateral para acessar o conteúdo na seção de Início rápido.

1 Autenticando e configurando usuário

Visão geral

Nesta seção você aprenderá como autenticar e configurar um usuário de sistema no Zabbix.

Login



Esta é a tela de boas-vindas do Zabbix. Digite o nome de usuário **Admin** com a senha **zabbix** para fazer login como superusuário do Zabbix. O acesso a todas as seções do menu será concedido.

Por motivos de segurança, é altamente recomendável alterar a senha padrão da conta Admin imediatamente após o primeiro login.

Login persistente

Para permanecer logado por até 30 dias, selecione *Lembrar por 30 dias* antes de clicar em *Entrar*.

Lembrar por 30 dias ativado:

- Sua sessão permanece ativa por 30 dias.
- O *Logout automático* é substituído, mantendo você logado até o final do período.
- Você será logado automaticamente em visitas futuras dentro de 30 dias sem precisar digitar as credenciais novamente.

Lembrar por 30 dias desativado:

- Qualquer login automático previamente ativado é removido.
- A sessão expirará de acordo com o intervalo de *Logout automático* configurado.

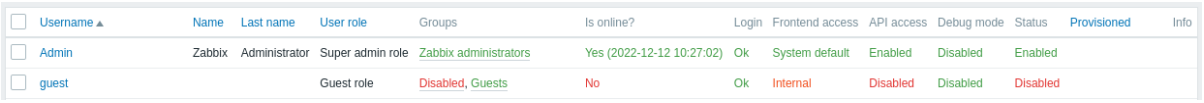
Proteção contra ataques de força bruta

No caso de cinco tentativas falhas de autenticação consecutivas, o Zabbix pausará por 30 segundos para prevenir ataques de força bruta e de dicionário.

O endereço IP de uma tentativa de autenticação falha será exibido após uma autenticação bem-sucedida.

Adicionando usuário

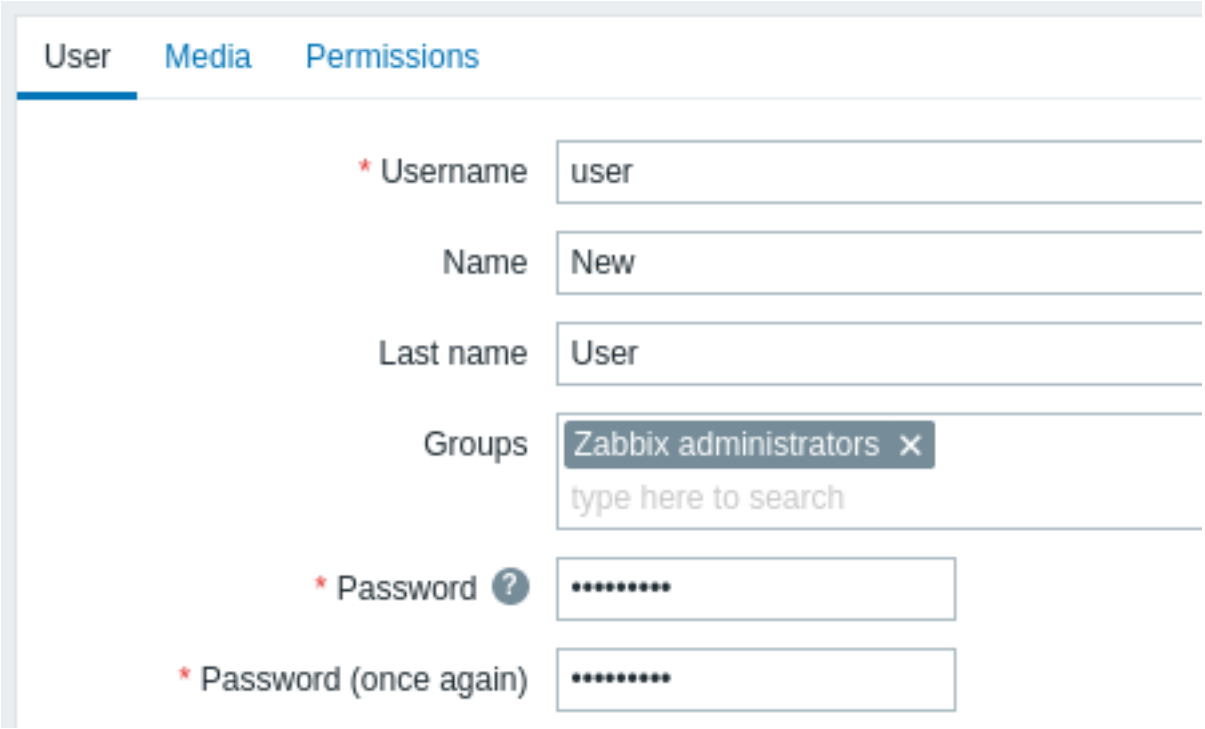
Para visualizar informações sobre usuários, vá para *Usuários* → *Usuários*.



Username ▲	Name	Last name	User role	Groups	Is online?	Login	Frontend access	API access	Debug mode	Status	Provisioned	Info
Admin	Zabbix	Administrator	Super admin role	Zabbix administrators	Yes (2022-12-12 10:27:02)	Ok	System default	Enabled	Disabled	Enabled		
guest			Guest role	Disabled, Guests	No	Ok	Internal	Disabled	Disabled	Disabled		

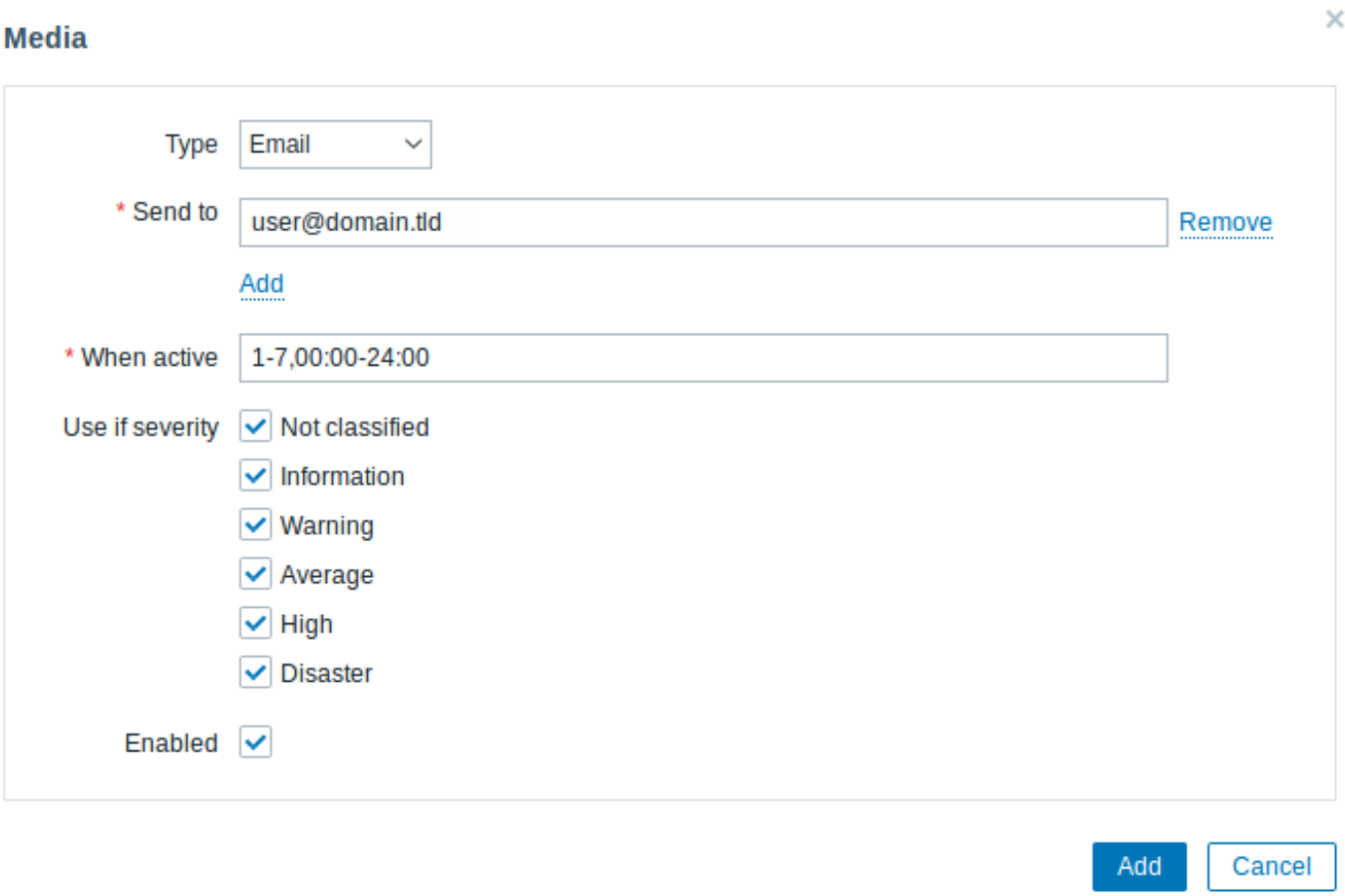
Para adicionar um novo usuário, clique em *Criar usuário*.

No formulário de novo usuário, certifique-se de adicionar seu usuário a um dos grupos de usuários existentes, por exemplo, 'Zabbix administrators'.



Todos os campos obrigatórios estão marcados com um asterisco vermelho.

Por padrão, novos usuários não têm mídias (métodos de entrega de notificação) definidas para eles. Para criar uma, vá até a aba 'Mídia' e clique em *Adicionar*.

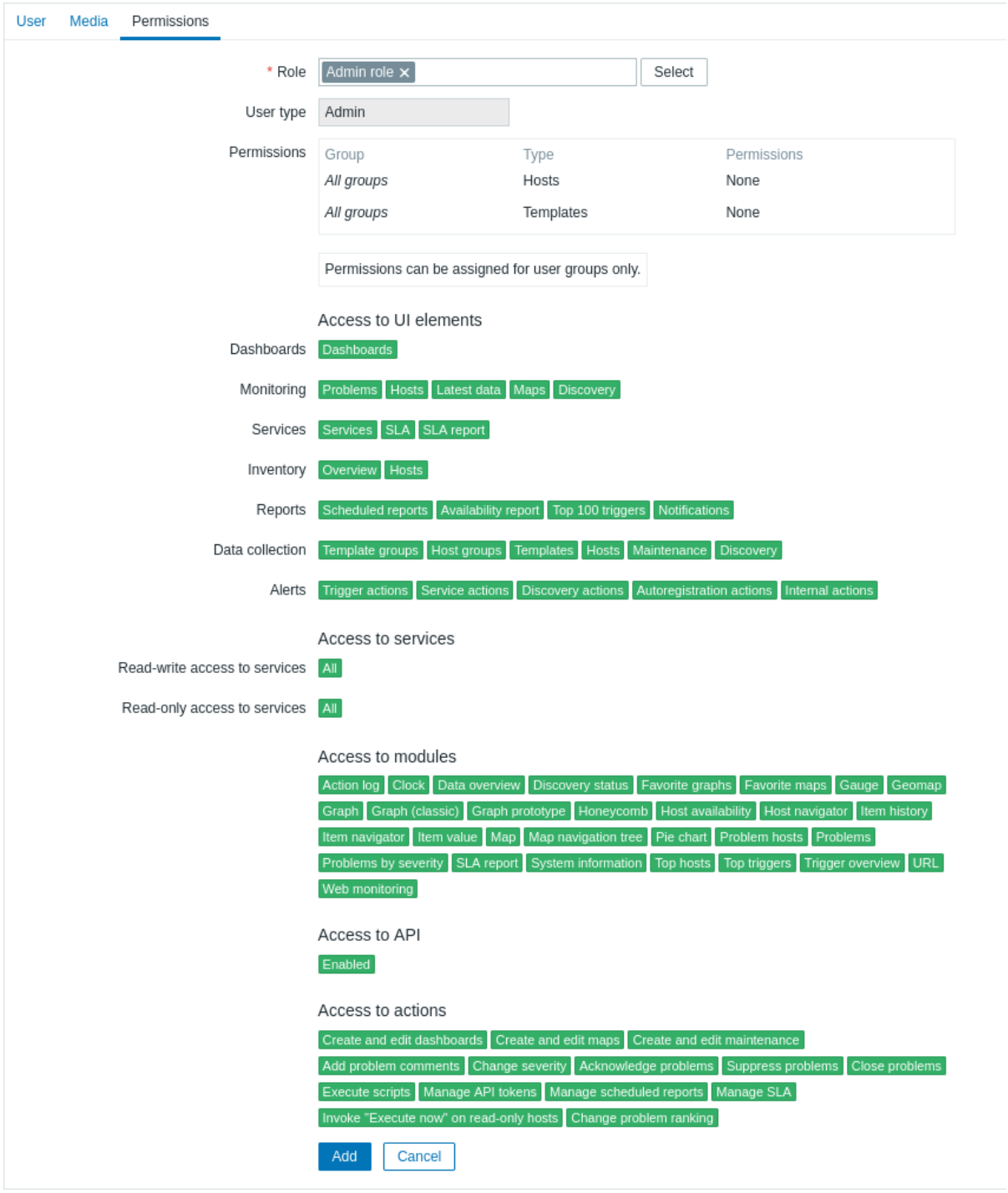


Neste pop-up, insira um endereço de e-mail para o usuário.

Você pode especificar um período de tempo em que a mídia estará ativa (veja a página Especificação de período de tempo para uma descrição do formato), por padrão uma mídia está sempre ativa. Você também pode personalizar os níveis de severidade do trigger para os quais a mídia estará ativa, mas deixe todos eles habilitados por enquanto.

Clique em *Adicionar* para salvar a mídia e, em seguida, vá para a aba Permissões.

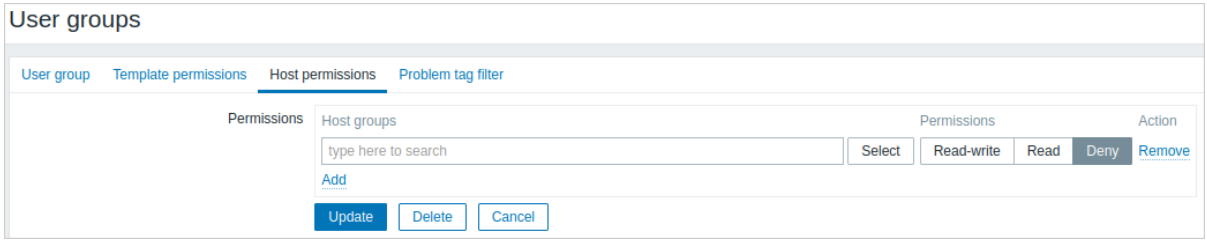
A aba Permissões possui um campo obrigatório *Função*. A função determina quais elementos do frontend o usuário pode visualizar e quais ações ele tem permissão para executar. Pressione Selecionar e selecione uma das funções da lista. Por exemplo, selecione *Função de administrador* para permitir o acesso a todas as seções do frontend do Zabbix, exceto Administração. Posteriormente, você pode modificar as permissões ou criar mais funções de usuário. Ao selecionar uma função, as permissões aparecerão na mesma aba:



Clique em *Adicionar* no formulário de propriedades do usuário para salvar o usuário. O novo usuário aparece na lista de usuários.

Adicionando permissões

Por padrão, um novo usuário não tem permissões para acessar hosts e templates. Para conceder direitos ao usuário, clique no grupo do usuário na coluna *Grupos* (neste caso - 'Zabbix administrators'). No formulário de propriedades do grupo, vá para a aba *Permissões de host* para atribuir permissões aos grupos de hosts. Clique em Add para que o campo de seleção de grupo de hosts seja exibido:



Em seguida, clique em *Selecionar* ao lado do campo para ver a lista de grupos de hosts. Este usuário terá acesso somente leitura ao grupo *Linux servers*, então marque a caixa de seleção apropriada na lista e clique em *Selecionar* para confirmar sua escolha.

Clique no botão *Leitura* para definir o nível de permissão e depois em *Atualizar* para salvar as alterações feitas na configuração do grupo de usuários.

Para conceder permissões a templates, você precisará alternar para a aba *Permissões de template* e especificar os grupos de templates.

Attention:
No Zabbix, os direitos de acesso a hosts e templates são atribuídos a grupos de usuários, não a usuários individuais.

Pronto! Você pode tentar fazer login usando as credenciais do novo usuário.

2 Novo host

Visão geral

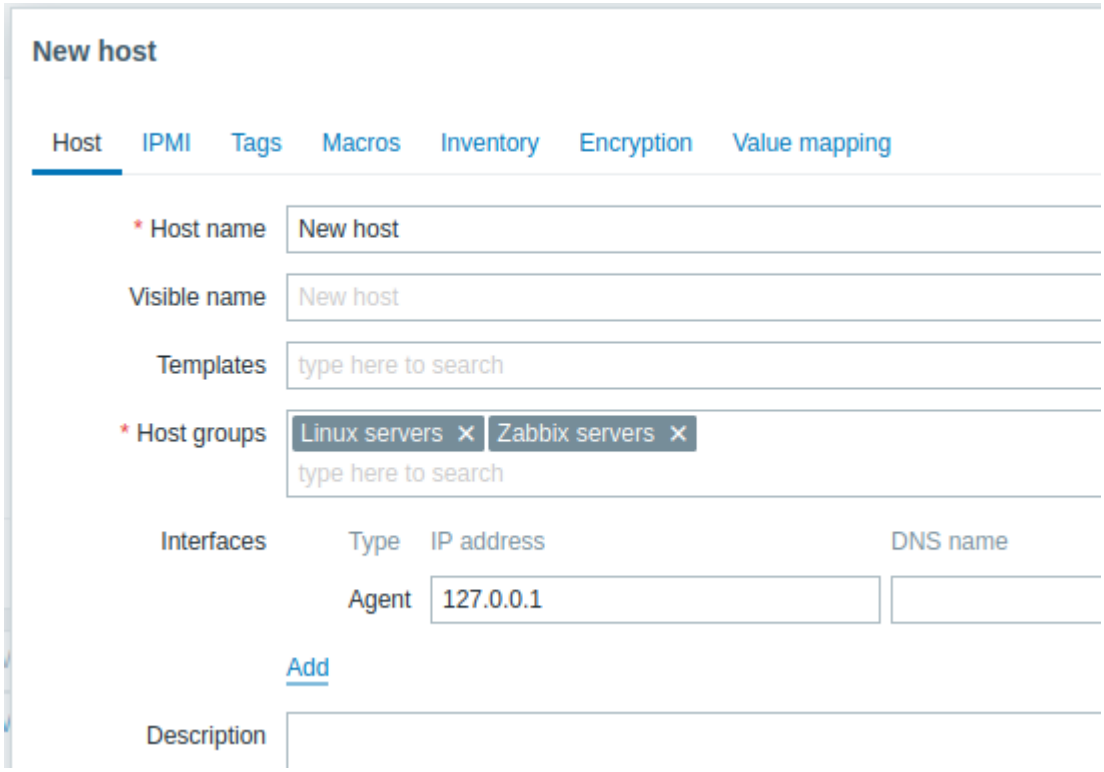
Nesta seção você aprenderá a como configurar um novo host.

Um host no Zabbix é uma entidade de rede (física, virtual) que você deseja monitorar. A definição do que pode ser um ”host” no Zabbix é bastante flexível. Ele pode ser um servidor físico, um switch de rede, uma máquina virtual ou alguma aplicação.

Adicionando host

As informações sobre hosts configurados no Zabbix estão disponíveis em *Coleta de dados* → *Hosts* assim como em *Monitoramento* → *Hosts*. Já existe um host pré-definido, chamado ”Zabbix server”, mas queremos aprender a adicionar outro.

Para adicionar um novo host, clique em *Criar host*. Isso apresentará um formulário de configuração de host.



Todos os campos obrigatórios estão marcados com um asterisco vermelho.

O mínimo necessário para preencher aqui é:

Nome do host

- Insira um nome para o host. São permitidos caracteres alfanuméricos, espaços, pontos, traços e sublinhados.

Grupos de hosts

- Selecione um ou vários grupos existentes clicando no botão *Selecionar* ou insira o nome de um grupo inexistente para criar um novo grupo.

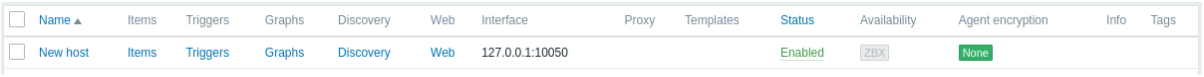
Note:
Todas as permissões de acesso são atribuídas a grupos de hosts, não a hosts individuais. Por isso, um host deve pertencer a pelo menos um grupo.

Interfaces: Endereço IP

- Embora tecnicamente não seja um campo obrigatório, uma interface de host é necessária para coletar certas métricas. Para usar verificações passivas do agent Zabbix, especifique o IP ou DNS do agent neste campo. Observe que você também deve especificar o IP ou DNS do Zabbix server na diretiva 'Server' do arquivo de configuração do agent Zabbix. Se o agent Zabbix e o Zabbix server estiverem instalados na mesma máquina, você deve especificar o mesmo IP/DNS em ambos os lugares.

Outras opções servirão para nós com seus valores padrão por enquanto.

Quando terminar, clique em *Adicionar*. Seu novo host deve estar visível na lista de hosts.



A coluna Disponibilidade contém indicadores de disponibilidade do host para cada interface. Definimos uma interface de agent Zabbix, então podemos usar o ícone de disponibilidade do agent (com 'ZBX' nele) para entender a disponibilidade do host:

- - o status do host não foi estabelecido; nenhuma verificação de métrica foi realizada ainda
- - o host está disponível, uma verificação de métrica foi bem-sucedida
- - o host está indisponível, uma verificação de métrica falhou (mova o cursor do mouse sobre o ícone para ver a mensagem de erro). Pode haver algum erro de comunicação, possivelmente causado por credenciais de interface incorretas. Verifique se o Zabbix server está em execução e tente atualizar a página mais tarde também.

3 Novo item

Visão geral

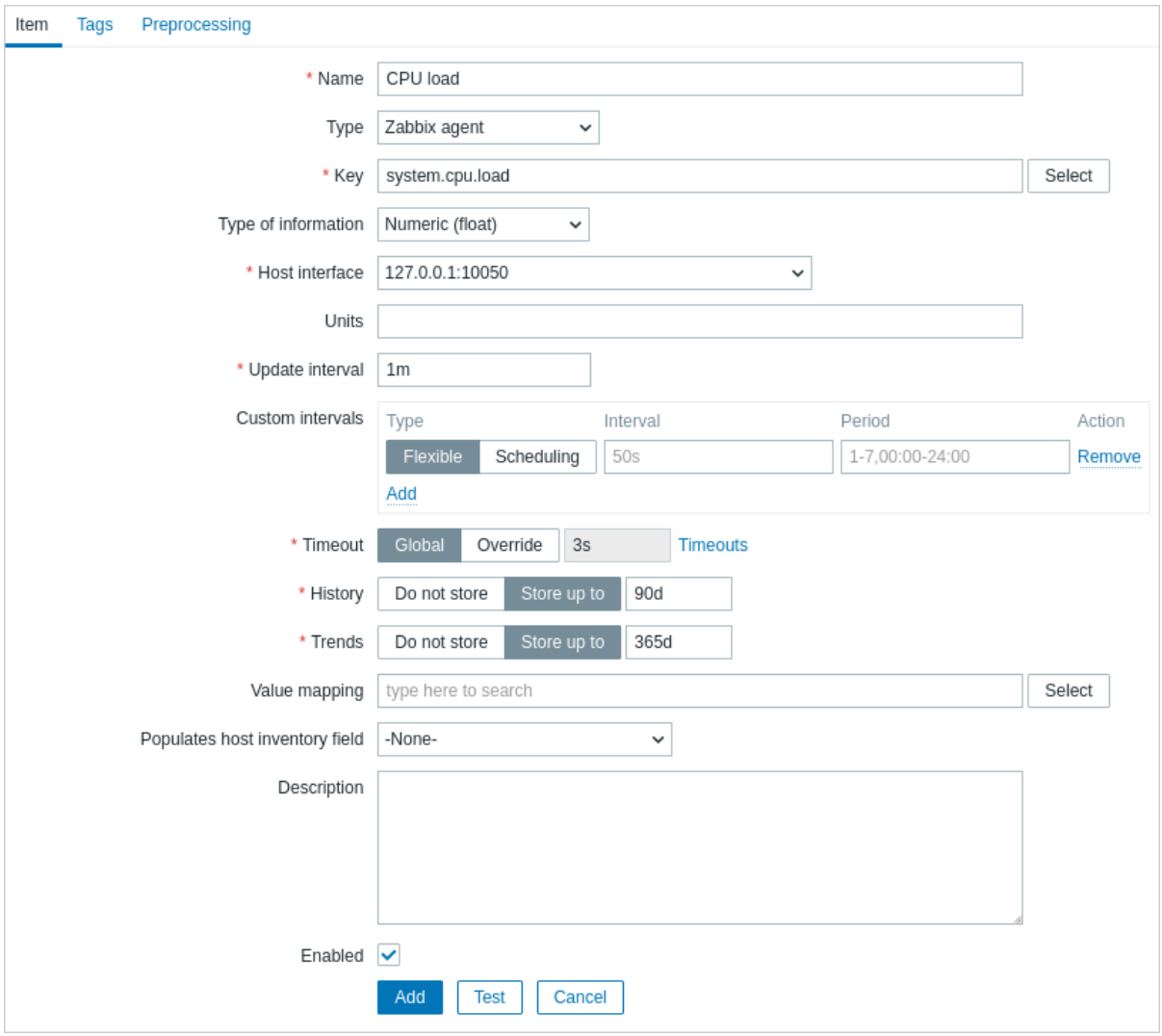
Nesta seção, você aprenderá como configurar um item.

Itens são a base da coleta de dados no Zabbix. Sem os itens, não há dados - porque apenas um item define uma métrica ou que tipo de dado coletar de um host.

Adicionando um item

Todos os itens são agrupados em torno de hosts. Por isso, para configurar um item de exemplo, vá para *Coleta de dados* → *Hosts* e encontre o "Novo host" que você criou.

Clique no link *Itens* na linha do "Novo host" e, em seguida, clique em *Criar item*. Isso apresentará um formulário de configuração de item.



Todos os campos obrigatórios estão marcados com um asterisco vermelho.

Para nosso item de exemplo, as informações essenciais a serem inseridas são:

Nome

- Digite *Carga da CPU* como valor. Este será o nome do item exibido em listas e outros locais.

Chave

- Digite manualmente *system.cpu.load* como valor. Este é o nome técnico de um item que identifica o tipo de informação que será coletada. Esta chave em particular é apenas uma das chaves predefinidas que vêm com o agent Zabbix.

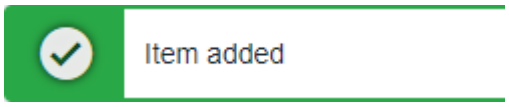
Tipo de informação

- Este atributo define o formato dos dados esperados. Para a chave *system.cpu.load*, este campo será definido automaticamente como *Numérico (float)*.

Note:
Você também pode querer reduzir o número de dias que o histórico do item será mantido, para 7 ou 14. Esta é uma boa prática para aliviar o banco de dados de manter muitos valores históricos.

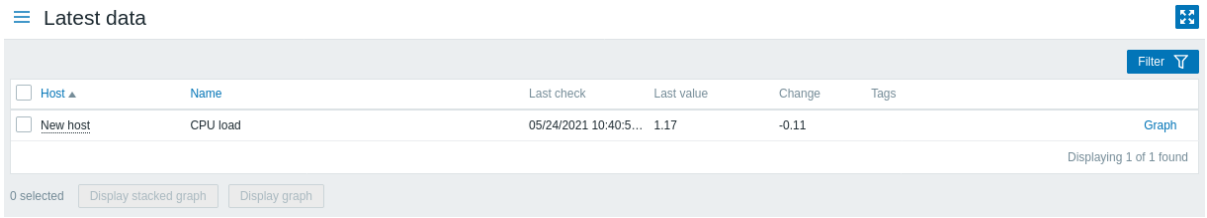
Outras opções serão adequadas com seus valores padrão por enquanto.

Quando terminar, clique em *Adicionar*. O novo item deve aparecer na lista de itens e você verá uma mensagem de sucesso.



Visualizando dados

Com um item definido, você pode estar curioso para saber se ele está realmente coletando dados. Para isso, vá em *Monitoramento → Últimos dados*, selecione 'Novo host' no filtro e clique em *Aplicar*.



Dito isso, pode levar até 60 segundos para que os primeiros dados cheguem. Esse é, por padrão, o intervalo em que o server lê as alterações de configuração e coleta novos itens para executar.

Se você não vir nenhum valor na coluna 'Alteração', talvez apenas um valor tenha sido recebido até agora. Aguarde 30 segundos para que outro valor chegue.

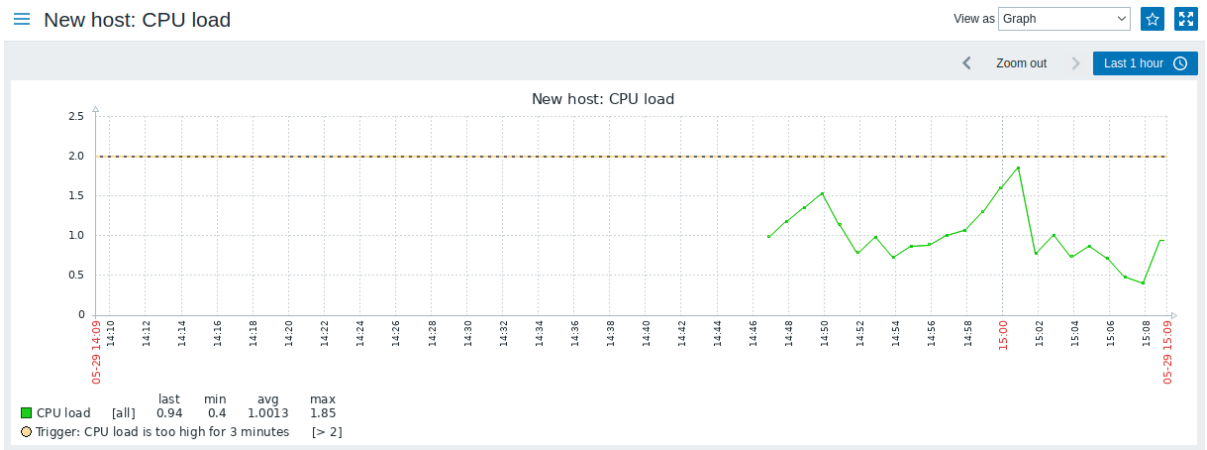
Se você não vir informações sobre o item como na captura de tela, certifique-se de que:

- você preencheu os campos 'Chave' e 'Tipo de informação' do item exatamente como na captura de tela;
- tanto o agent quanto o server estão em execução;
- o status do host está como 'Monitorado' e seu ícone de disponibilidade está verde;
- o host selecionado no filtro de host está correto;
- o item está habilitado.

Gráficos

Com o item funcionando por um tempo, deve ser tempo de visualizar alguma coisa visual. Gráficos simples estão disponíveis para qualquer item numérico monitorado sem qualquer configuração adicional. Estes gráficos são gerados em tempo de execução.

Para visualizar o gráfico, vá até *Monitoramento → Últimos dados* e clique no link 'Gráfico' próximo ao item.



4 Novo trigger

Visão geral

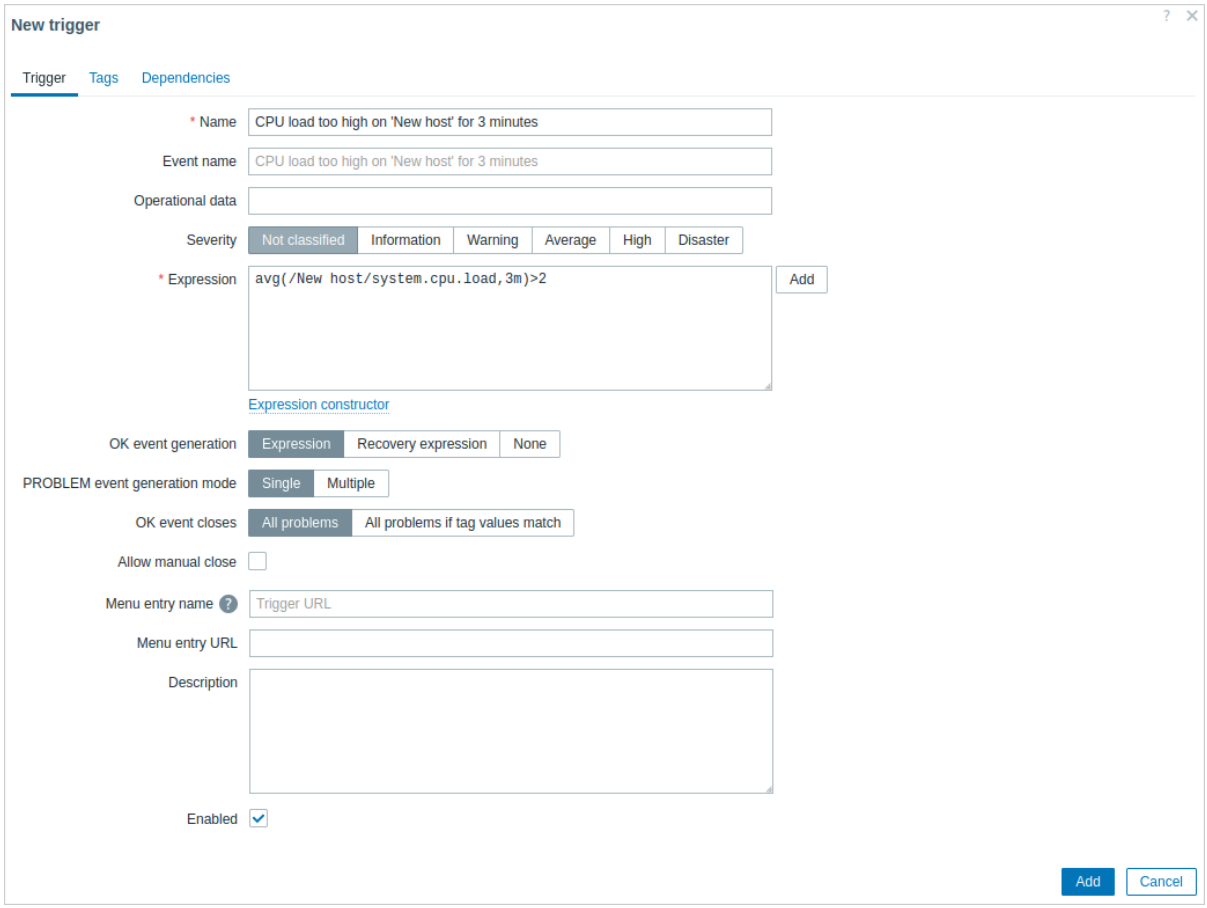
Nesta seção você aprenderá como configurar um gatilho.

Itens apenas coletam dados. Para avaliar automaticamente os dados entrantes nós precisamos definir gatilhos. Um gatilho contém uma expressão que define um limite do que é um nível aceitável para o dado.

Se este nível é ultrapassado pelo dado de entrada, um gatilho irá ”disparar” ou entrará em um estado de ’Problema’ - permitindo que tomemos consciência de que algo aconteceu e que pode necessitar de atenção. Se o nível voltar a ser aceitável, o gatilho retorna para um estado ’Ok’.

Adicionando um trigger

Para configurar um trigger para nosso item, vá para *Coleta de dados* → *Hosts*. Clique no link *Triggers* na linha do ’Novo host’ e, em seguida, clique em *Criar trigger*. Isso apresentará um formulário de configuração de trigger.



Para nosso trigger, as informações essenciais a serem inseridas aqui são:

Nome

- Insira *Carga de CPU muito alta no ’Novo host’ por 3 minutos* como valor. Este será o nome do trigger exibido em listas e outros lugares.

Expressão

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

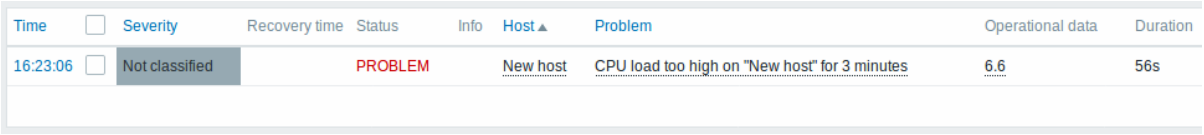
Esta é a expressão do trigger. Certifique-se de que a expressão seja inserida corretamente, até o último símbolo. A chave do item aqui (system.cpu.load) é usada para se referir ao item. Esta expressão específica basicamente diz que o limite de problema é excedido quando o valor médio da carga da CPU por 3 minutos está acima de 2. Você pode aprender mais sobre a sintaxe das expressões de trigger.

Quando terminar, clique em *Adicionar*. O novo trigger deve aparecer na lista de triggers.

Exibindo estado do gatilho

Com um gatilho definido, você deve estar interessado em ver seu estado.

Se a carga de CPU excedeu o nível limite que você definiu no gatilho, o problema será exibido em *Monitoramento* → *Problemas*.



A intermitência na coluna de estado indica uma mudança recente do estado do gatilho, uma que tenha surgido nos últimos 30 minutos.

5 Recebendo notificação de problema

Visão geral

Nesta seção, você aprenderá como configurar alertas na forma de notificações no Zabbix.

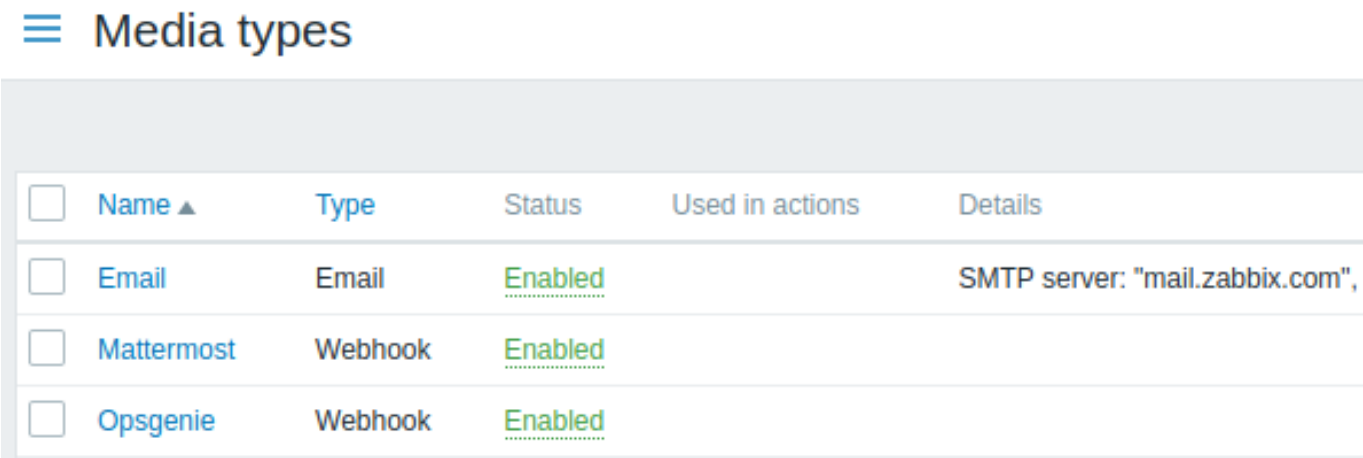
Com os items coletando dados e triggers projetados para ”disparar” em situações problemáticas, também seria útil ter algum mecanismo de alerta que nos notificasse sobre eventos importantes, mesmo quando não estivermos olhando diretamente para o frontend do Zabbix.

É isso que as notificações fazem. Como o e-mail é o método de entrega mais popular para notificações de problemas, aprenderemos como configurar uma notificação por e-mail.

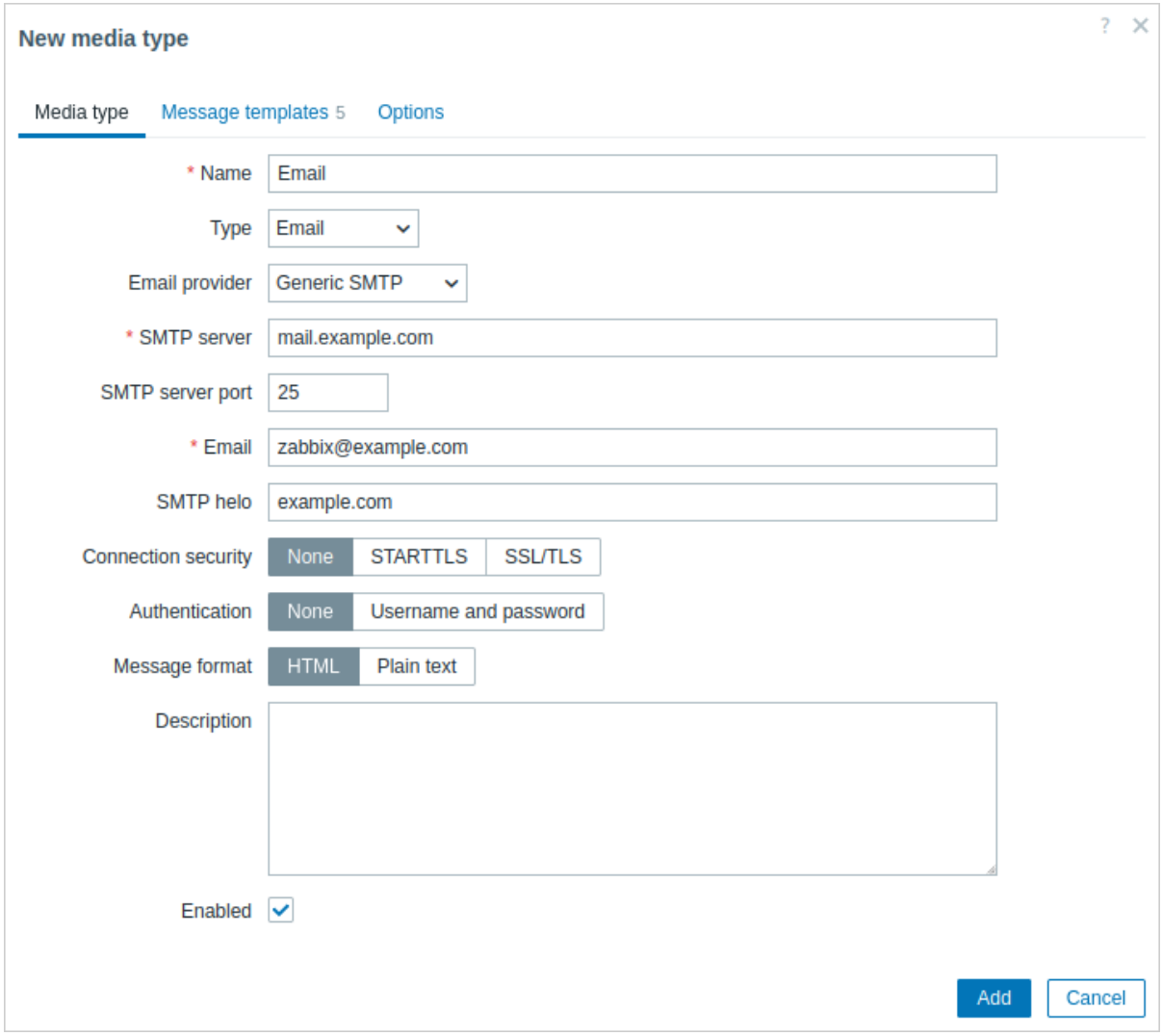
Configurações de e-mail

Inicialmente, existem vários métodos de entrega de notificações predefinidos no Zabbix. Email é um deles.

Para configurar as definições de e-mail, vá para *Alertas* > *Tipos de mídia* e clique em *Email* na lista de tipos de mídia predefinidos.



Isso nos apresentará o formulário de configuração das definições de e-mail.



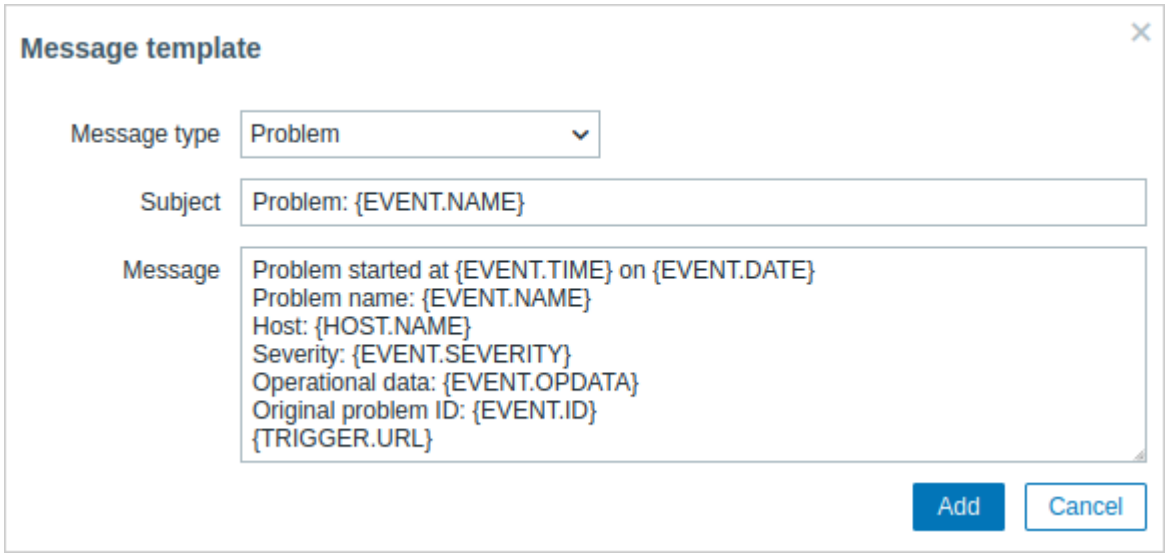
Todos os campos obrigatórios estão marcados com um asterisco vermelho.

Na aba *Tipo de mídia*, defina os valores de *Servidor SMTP*, *SMTP helo* e *E-mail* de acordo com o seu ambiente.

Note:
O valor no campo *E-mail* será usado como o endereço 'De' para as notificações enviadas pelo Zabbix.

Em seguida, é necessário definir o conteúdo da mensagem de problema. O conteúdo é definido por meio de um modelo de mensagem, configurado na aba *Modelos de mensagem*.

Clique em *Adicionar* para criar um modelo de mensagem e selecione *Problema* como o tipo de mensagem.

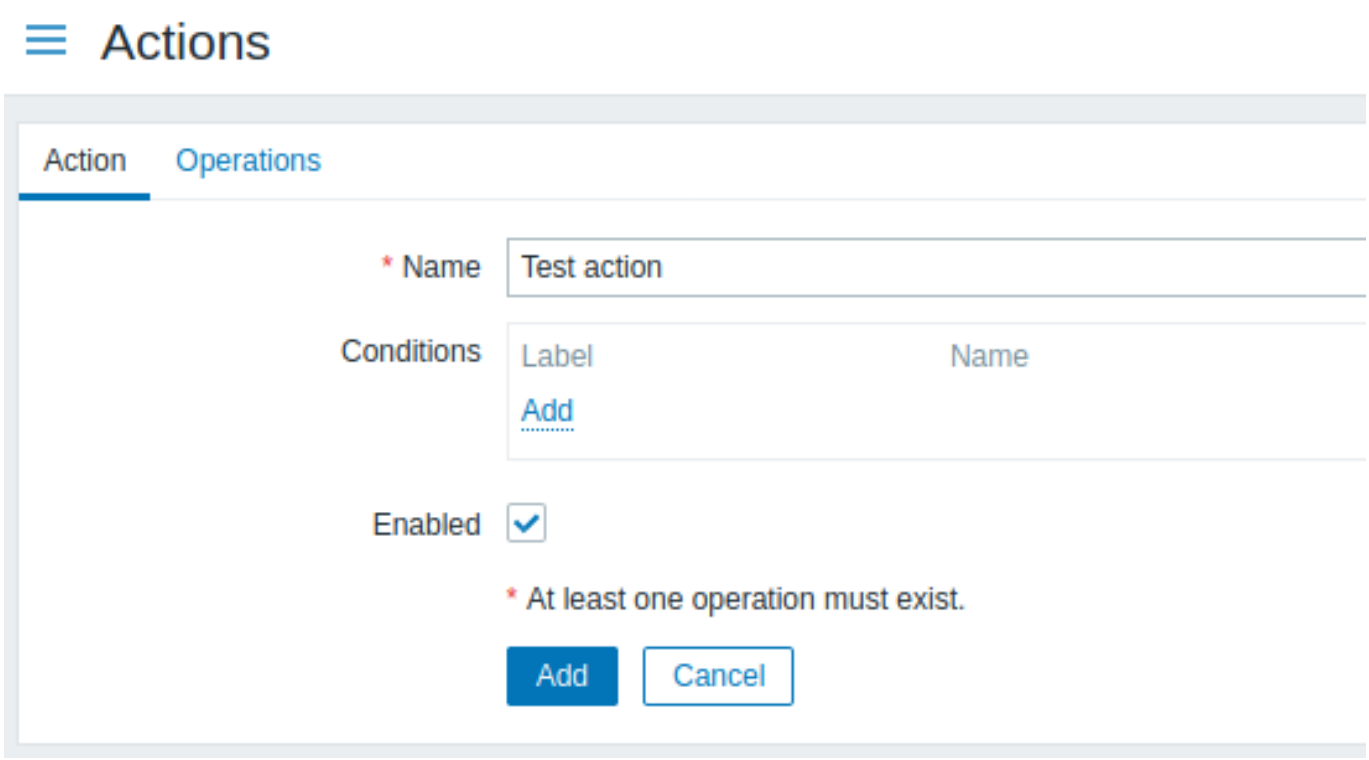


Clique em *Adicionar* quando estiver pronto e salve o formulário.

Agora você configurou *Email* como um tipo de mídia funcional. O tipo de mídia também deve ser vinculado aos usuários, definindo endereços de entrega específicos (como fizemos ao configurar um novo usuário), caso contrário, ele não será utilizado.

Nova ação

Entregar notificações é uma das coisas que as ações fazem no Zabbix. Portanto, para configurar uma notificação, vá para *Alertas* > *Ações* > *Ações de trigger* e clique em *Criar ação*.

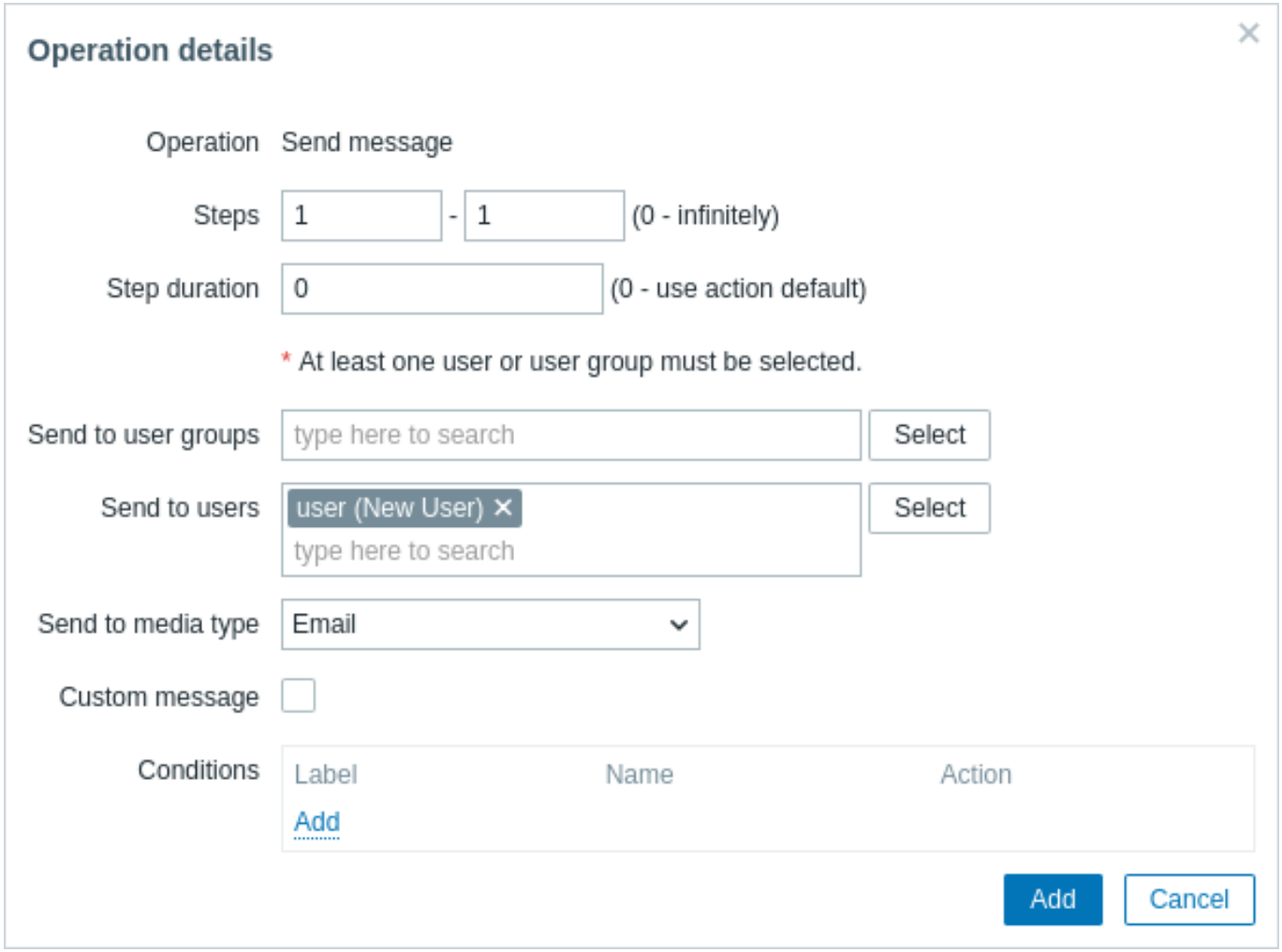


Todos os campos obrigatórios estão marcados com um asterisco vermelho.

Neste formulário, insira um nome para a ação.

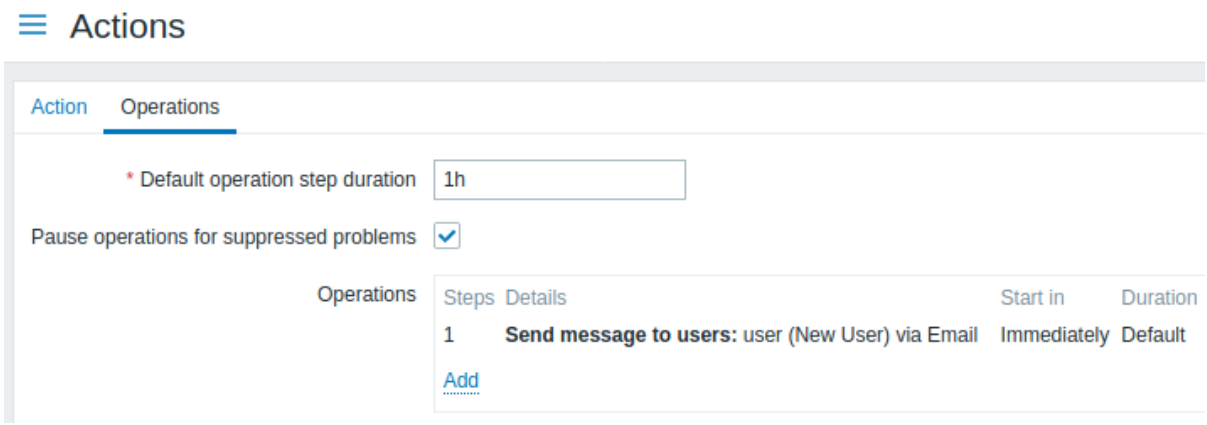
No caso mais simples, se não adicionarmos nenhuma condição mais específica, a ação será executada em qualquer alteração de trigger de 'Ok' para 'Problema'.

Ainda devemos definir o que a ação deve fazer - e isso é feito na guia *Operações*. Clique em *Adicionar* no bloco *Operações*, o que abrirá um novo formulário de operação.



Todos os campos obrigatórios estão marcados com um asterisco vermelho.

Aqui, clique em *Selecionar* no bloco *Enviar para usuários* e selecione o usuário ('user') que definimos. Selecione "Email" como valor de *Enviar para tipo de mídia* (antes do Zabbix 7.0.7, *Enviar apenas para*). Quando terminar, clique em *Adicionar* e a operação será adicionada:



Isso é tudo para uma configuração simples de ação, então clique em *Adicionar* no formulário de ação.

Recebendo notificação

Agora, com a entrega de notificações configurada, seria interessante realmente receber uma. Para ajudar com isso, podemos aumentar propositalmente a carga em nosso host - para que nosso trigger "dispare" e recebamos uma notificação de problema.

Abra o console em seu host e execute:

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

Você pode executar um ou vários desses processos.

Agora vá para *Monitoramento* > *Últimos dados* e veja como os valores de 'Carga da CPU' aumentaram. Lembre-se, para que nosso trigger "dispare", o valor de 'Carga da CPU' deve ultrapassar '2' por 3 minutos consecutivos. Assim que isso acontecer:

- em *Monitoramento* > *Problemas* você deve ver o trigger com o status 'Problema' piscando;
- você deve receber uma notificação de problema em seu e-mail.

Attention:
Se as notificações não funcionarem:
- verifique novamente se tanto as configurações de e-mail quanto a ação foram configuradas corretamente
- certifique-se de que o usuário criado tenha pelo menos permissões de leitura no host que gerou o evento, conforme observado na etapa *Adicionando usuário*. O usuário, sendo parte do grupo de usuários 'Zabbix administrators', deve ter pelo menos acesso de leitura ao grupo de hosts 'Linux servers' ao qual nosso host pertence.
- Além disso, você pode verificar o log de ações acessando *Relatórios* > *Log de ações*.

6 Novo modelo (template)

Visão geral

Nesta seção você aprenderá como configurar um modelo.

Previamente nós aprendemos como configurar um item, um gatilho e como receber uma notificação de problema para o host.

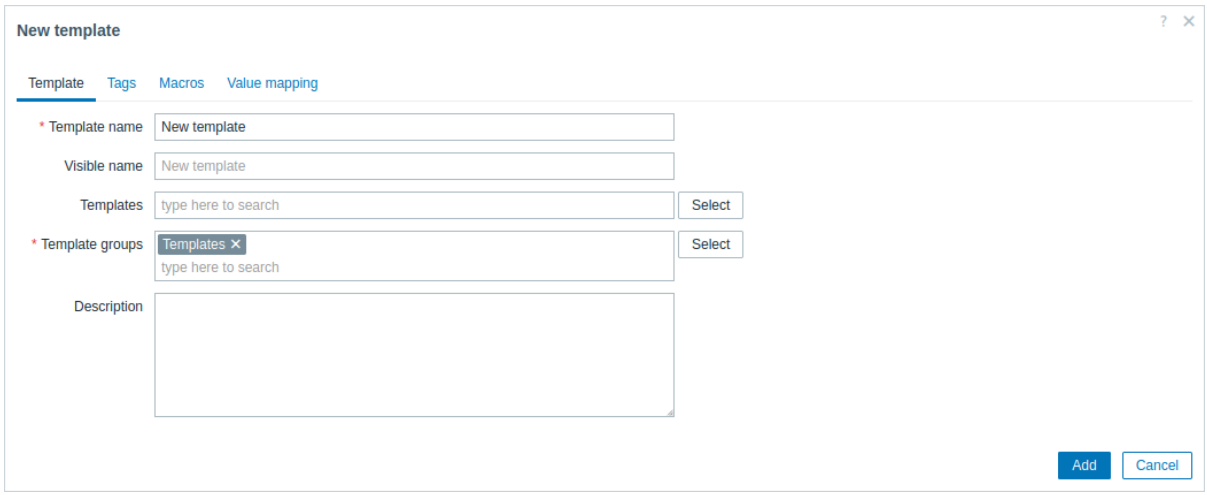
Enquanto todas estas etapas oferecem uma boa flexibilidade por si mesmas, pode parecer como um monte de etapas a serem tomadas se necessário para, digamos, mil hosts. Alguma automação seria uma mão na roda.

Aqui é onde os modelos vêm nos socorrer. Modelos permitem agrupar itens, gatilhos e outras entidades úteis de forma que possam ser reutilizados continuamente pela aplicação aos hosts em uma única etapa.

Quando um modelo é associado a um host, o host herda todas as entidades do modelo. Então, basicamente um monte de verificações podem ser aplicadas muito rapidamente.

Adicionando template

Para começar a trabalhar com templates, primeiro devemos criar um. Para isso, em *Coleta de dados* → *Templates*, clique em *Criar template*. Isso apresentará um formulário de configuração de template.



Todos os campos obrigatórios estão marcados com um asterisco vermelho.

Os parâmetros obrigatórios a serem preenchidos são:

Nome do template

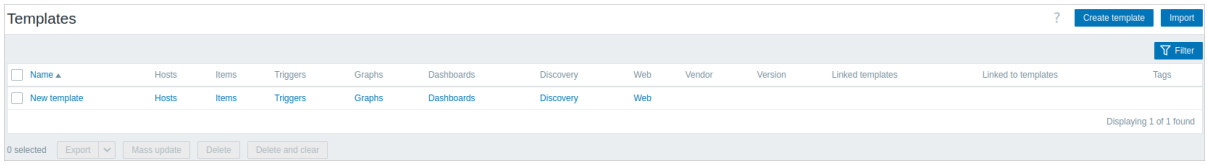
- Insira um nome para o template. São permitidos caracteres alfanuméricos, espaços e sublinhados.

Grupos de template

- Selecione um ou mais grupos clicando no botão *Selecionar*. O template deve pertencer a um grupo.

Note:
As permissões de acesso aos grupos de template são atribuídas na configuração do grupo de usuários na aba **Permissões de template**, da mesma forma que as permissões de host. Todas as permissões de acesso são atribuídas a grupos, não a templates individuais, por isso é obrigatório incluir o template em pelo menos um grupo.

Quando terminar, clique em *Adicionar*. Seu novo template deve estar visível na lista de templates. Você também pode usar o filtro para encontrar seu template.



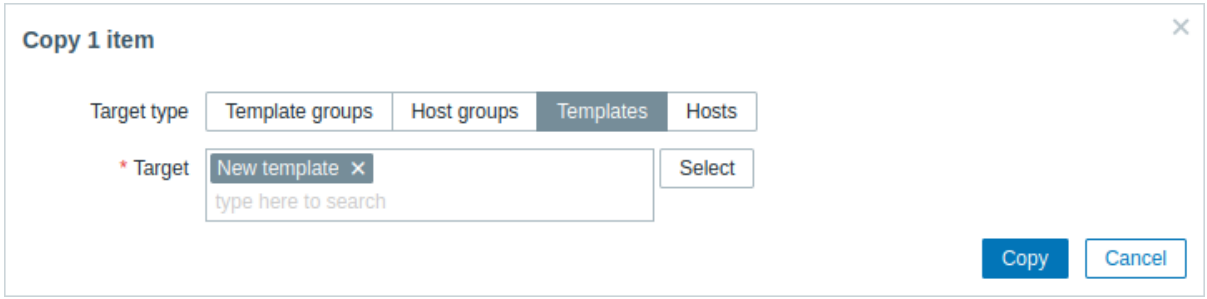
Como você pode ver, o template está lá, mas não contém nada - nenhum item, trigger ou outra entidade.

Adicionando item ao template

Para adicionar um item ao template, vá para a lista de itens do 'Novo host'. Em *Coleta de dados* → *Hosts* clique em *Itens* ao lado de 'Novo host'.

Então:

- Marque a caixa de seleção do item 'CPU Load' na lista.
- Clique em *Copiar* abaixo da lista.
- Selecione o template para o qual deseja copiar o item.



Todos os campos obrigatórios estão marcados com um asterisco vermelho.

- Clique em *Copiar*.

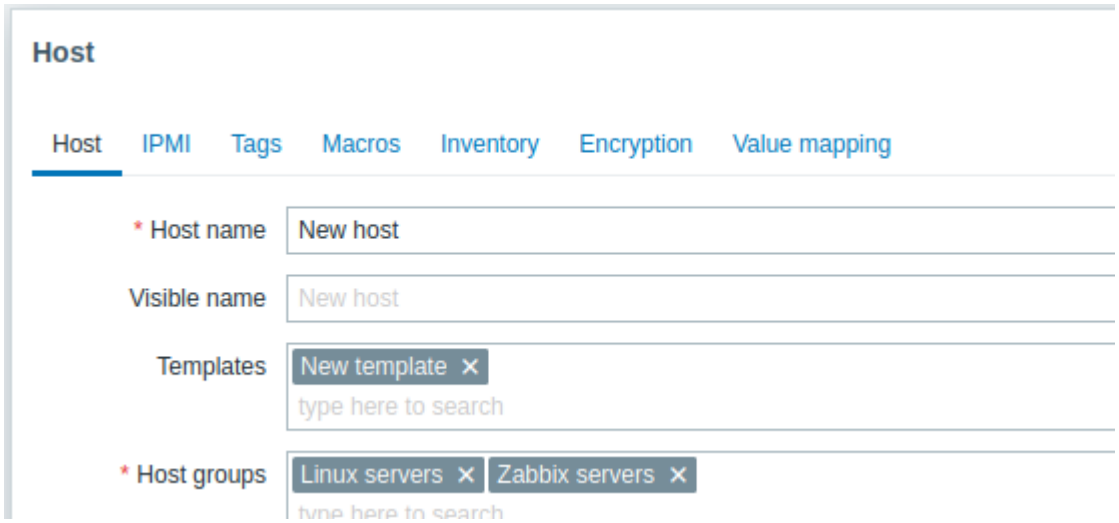
Se você for agora em *Coleta de dados* → *Templates*, o 'Novo template' deve ter um novo item nele.

Vamos parar em apenas um item por enquanto, mas da mesma forma você pode adicionar quaisquer outros itens, triggers ou outras entidades ao template até que seja um conjunto razoavelmente completo de entidades para o propósito desejado (monitoramento de SO, monitoramento de uma única aplicação).

Vinculando template ao host

Com um template pronto, basta adicioná-lo a um host. Para isso, vá em *Coleta de dados* → *Hosts*, clique em 'Novo host' para abrir o formulário de propriedades e localize o campo **Templates**.

Comece a digitar *Novo template* no campo *Templates*. O nome do template que criamos deve aparecer na lista suspensa. Role para baixo para selecionar. Veja que ele aparece no campo *Templates*.



Clique em *Atualizar* no formulário para salvar as alterações. O template agora está adicionado ao host, com todas as entidades que ele possui.

Dessa forma, ele pode ser aplicado a qualquer outro host também. Quaisquer alterações nos items, triggers e outras entidades no nível do template serão propagadas para os hosts aos quais o template está vinculado.

Associando modelos pré-definidos aos hosts

Como você deve ter notado, o Zabbix vem com um conjunto de modelo pré-definidos para vários SOs, dispositivos e aplicações. Para começar com o monitoramento de forma muito rápida, você pode associar o modelo apropriado a um host, mas cuide que estes modelos precisam estar bem ajustados para o seu ambiente. Algumas verificações podem não ser necessárias, e intervalos de sondagem (polling) podem ser frequentes demais.

Mais informações sobre modelos estão disponíveis.

6 Zabbix appliance

Visão geral Como alternativa à configuração manual ou reutilização de um servidor existente para o Zabbix, os usuários podem baixar um appliance Zabbix ou uma imagem de CD de instalação do appliance Zabbix.

As versões do appliance Zabbix e do CD de instalação são baseadas no AlmaLinux 8 (x86_64).

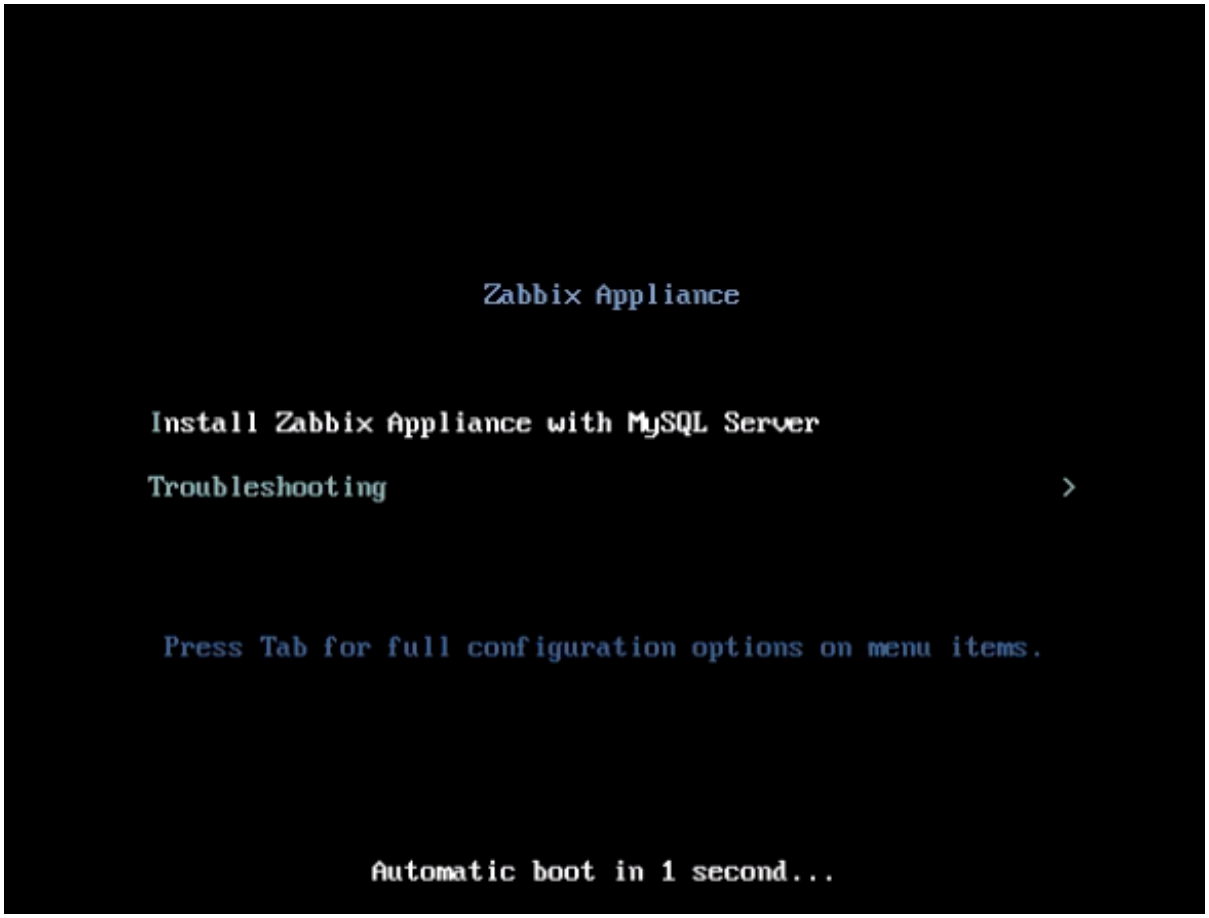
O CD de instalação do appliance Zabbix pode ser usado para implantação instantânea do Zabbix server (MySQL).

> **Attention:**
> Você pode usar este Appliance para avaliar o Zabbix. O Appliance não se destina ao uso sério em produção.

Requisitos do sistema:

- *RAM*: 4 GB
- *Espaço em disco*: pelo menos 8 GB devem ser alocados para a máquina virtual
- *CPU*: mínimo de 2 núcleos

Menu de inicialização do CD/DVD de instalação do Zabbix:



O appliance Zabbix contém um Zabbix server (configurado e em execução no MySQL) e um frontend.

O appliance virtual Zabbix está disponível nos seguintes formatos:

- VMware (.vmx)
- Formato de virtualização aberto (.ovf)
- Microsoft Hyper-V (.vhd / .vhdx)
- KVM, Parallels, QEMU, USB stick, VirtualBox, Xen (.raw)
- KVM, QEMU (.qcow2)

Para começar, inicialize o appliance e aponte um navegador para o IP que o appliance recebeu via DHCP.

> **Attention:**
> O DHCP deve estar habilitado no host.

Para obter o endereço IP de dentro da máquina virtual, execute:

```
ip addr show
```

Para acessar o frontend do Zabbix, vá para **http://<host_ip>** (para acesso a partir do navegador do host, o modo bridge deve estar habilitado nas configurações de rede da VM).

> **Note:**
> Se o appliance não iniciar no Hyper-V, você pode pressionar `Ctrl+Alt+F2` para alternar as sessões tty.

Alterações na configuração do AlmaLinux 8 O appliance é baseado no AlmaLinux 8. Algumas alterações são aplicadas à configuração base do AlmaLinux.

Repositórios

O repositório oficial do Zabbix foi adicionado em */etc/yum.repos.d*:

```
[zabbix]
name=Zabbix Official Repository - $basearch
baseurl=http://repo.zabbix.com/zabbix/7.0/rhel/8/$basearch/
enabled=1
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-ZABBIX-A14FE591
```

Configuração do firewall

O appliance usa o firewall iptables com regras predefinidas:

- Porta SSH aberta (22 TCP);
- Portas do agent Zabbix (10050 TCP) e do trapper Zabbix (10051 TCP) abertas;
- Portas HTTP (80 TCP) e HTTPS (443 TCP) abertas;
- Porta SNMP trap aberta (162 UDP);
- Conexões de saída abertas para a porta NTP (123 UDP);
- Pacotes ICMP limitados a 5 pacotes por segundo;
- Todas as outras conexões de entrada são descartadas.

Usando um endereço IP estático

Por padrão, o appliance usa DHCP para obter o endereço IP. Para especificar um endereço IP estático:

- Faça login como usuário root;
- Abra o arquivo */etc/sysconfig/network-scripts/ifcfg-eth0*;
- Substitua *BOOTPROTO=dhcp* por *BOOTPROTO=none*
- Adicione as seguintes linhas:
  - *IPADDR=<endereço IP do appliance>*
  - *PREFIX=<prefixo CIDR>*
  - *GATEWAY=<endereço IP do gateway>*
  - *DNS1=<endereço IP do servidor DNS>*
- Execute o comando **systemctl restart network**.

Consulte a documentação oficial da Red Hat se necessário.

Alterando o fuso horário

Por padrão, o appliance usa UTC para o relógio do sistema. Para alterar o fuso horário, copie o arquivo apropriado de */usr/share/zoneinfo* para */etc/localtime*, por exemplo:

```
cp /usr/share/zoneinfo/Europe/Riga /etc/localtime
```

Configuração do Zabbix A configuração do appliance Zabbix possui as seguintes senhas e alterações de configuração:

Credenciais (login:senha)

Sistema:

- root:zabbix

Zabbix frontend:

- Admin:zabbix

Banco de dados:

- root:<random>
- zabbix:<random>

> **Note:**
> As senhas do banco de dados são geradas aleatoriamente durante o processo de instalação.
 A senha do root é armazenada no arquivo /root/.my.cnf. Não é necessário inserir uma senha na conta "root".

Para alterar a senha do usuário do banco de dados, as alterações devem ser feitas nos seguintes locais:

- MySQL;
- /etc/zabbix/zabbix_server.conf;
- /etc/zabbix/web/zabbix.conf.php.

> **Note:**
> Usuários separados `zabbix_srv` e `zabbix_web` são definidos para o server e para o frontend, respectivamente.

Localização dos arquivos

- Os arquivos de configuração estão localizados em **/etc/zabbix**.
- Os arquivos de log do server, proxy e agent do Zabbix estão localizados em **/var/log/zabbix**.
- O frontend do Zabbix está localizado em **/usr/share/zabbix**.
- O diretório home do usuário **zabbix** é **/var/lib/zabbix**.

Alterações na configuração do Zabbix

- O fuso horário do frontend é definido como Europe/Riga (isso pode ser modificado em **/etc/php-fpm.d/zabbix.conf**);

Acesso ao frontend Por padrão, o acesso ao frontend é permitido de qualquer lugar.

O frontend pode ser acessado em *http://<host>*.

Isso pode ser personalizado em **/etc/nginx/conf.d/zabbix.conf**. O Nginx precisa ser reiniciado após modificar este arquivo. Para isso, faça login via SSH como usuário **root** e execute:

```
systemctl restart nginx
```

Firewall Por padrão, apenas as portas listadas nas alterações de configuração acima estão abertas. Para abrir portas adicionais, modifique o arquivo *"/etc/sysconfig/iptables"* e recarregue as regras do firewall:

```
systemctl reload iptables
```

Atualizando Os pacotes do appliance Zabbix podem ser atualizados. Para isso, execute:

```
dnf update zabbix*
```

Serviços do sistema Serviços systemd estão disponíveis:

```
systemctl list-units zabbix*
```

Notas específicas do formato VMware

As imagens no formato *vmdk* podem ser usadas diretamente nos produtos VMware Player, Server e Workstation. Para uso no ESX, ESXi e vSphere, elas devem ser convertidas usando o VMware vCenter Converter (autenticação necessária para download). Se você usar o VMWare vCenter Converter, pode encontrar problemas com o adaptador de rede híbrido. Nesse caso, você pode tentar especificar o adaptador E1000 durante o processo de conversão. Alternativamente, após a conversão ser concluída, você pode excluir o adaptador existente e adicionar um adaptador E1000.

Imagem HDD/flash (raw)

```
dd if=./zabbix_appliance_7.0.0.raw of=/dev/sdc bs=4k conv=fdatasync
```

Substitua */dev/sdc* pelo dispositivo do seu disco Flash/HDD.

Solução de problemas Se você encontrar a mensagem de erro *Access denied for user 'replace_user'@'localhost' (using password: YES)* ao tentar fazer login no frontend, isso pode indicar que a instalação ainda está em andamento.

Se o erro persistir após aguardar alguns minutos, ou se você observar qualquer outro comportamento inesperado, provavelmente significa que o processo de instalação não foi concluído com sucesso. Nesse caso, recomendamos excluir o appliance atual e implantá-lo novamente seguindo as mesmas instruções de instalação. Essa etapa geralmente resolve o problema.

Observe que tentar corrigir manualmente uma instalação corrompida não é recomendado, pois pode levar a complicações adicionais.

7 Configuração

Use a barra lateral para acessar o conteúdo da seção Configuração.

1 Configurando um modelo

Visão geral

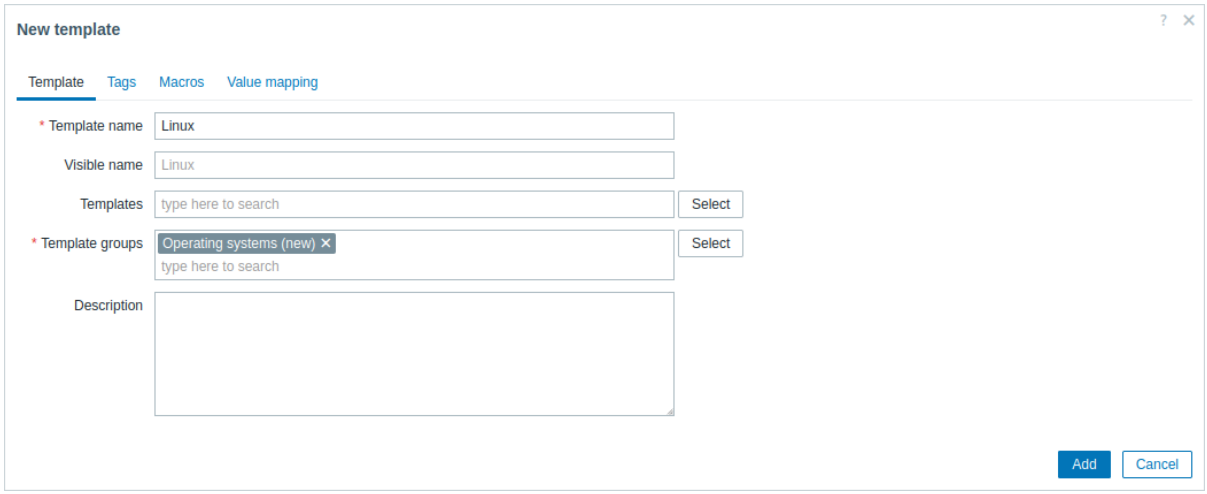
A configuração de um modelo requer que você primeiro crie um modelo definindo seus parâmetros gerais e então você adiciona entidades (itens, gatilhos, gráficos, etc.)

Criando um template

Para criar um template, faça o seguinte:

1. Vá para *Coleta de dados* → *Templates*.
2. Clique em *Criar template*.
3. Edite os atributos do template.

A aba **Template** contém os atributos gerais do template.

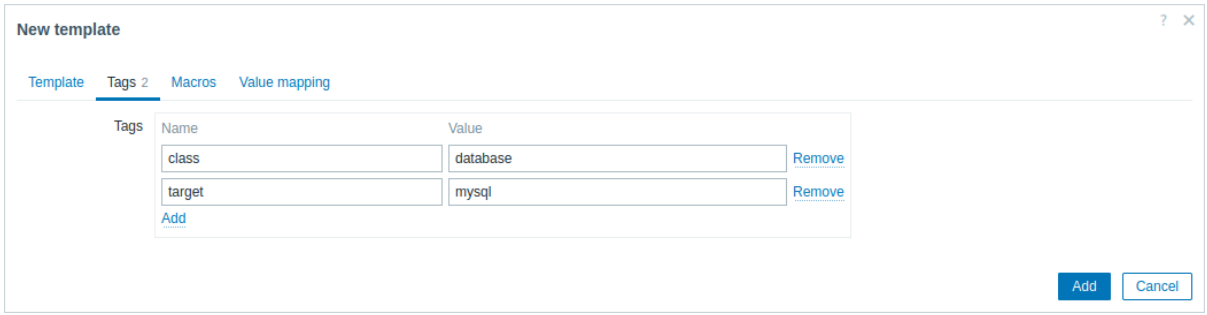


Todos os campos obrigatórios estão marcados com um asterisco vermelho.

Atributos do template:

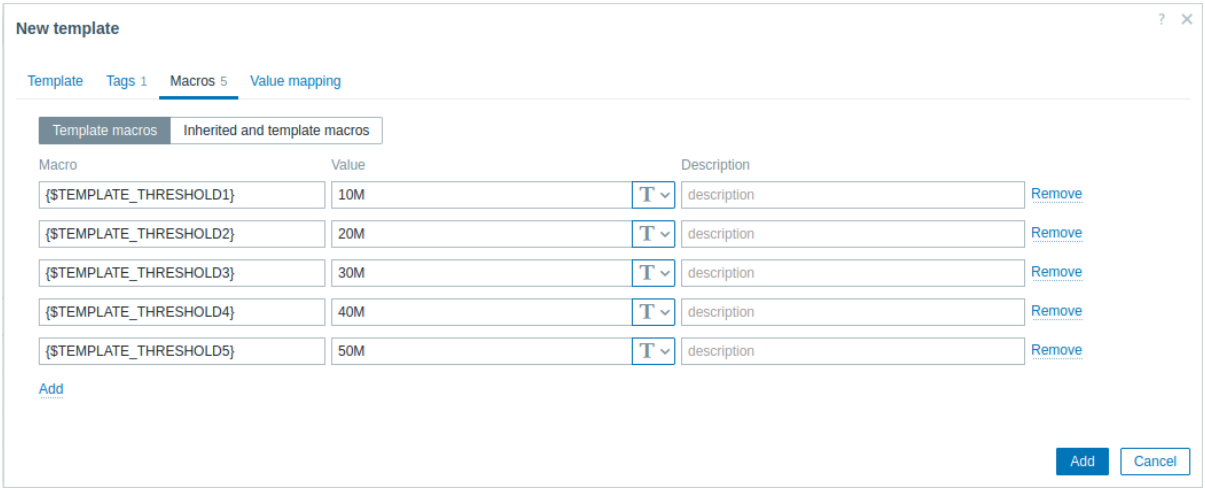
Parâmetro	Descrição
Nome do template	Nome único do template. Alfanuméricos, espaços, pontos, traços e sublinhados são permitidos. Espaços à esquerda e à direita não são permitidos.
Nome visível	Se você definir este nome, ele será o visível em listas, mapas, etc.
Templates	Vincule um ou mais templates a este template. Todas as entidades (items, triggers, etc.) serão herdadas dos templates vinculados. Para vincular um novo template, digite seu nome no campo *Templates* — isso exibirá uma lista suspensa de templates correspondentes. Alternativamente, clique em *Selecionar* para abrir o pop-up *Templates*, que inicialmente não mostra nenhum template. Para preencher a lista, digite o nome de um grupo de templates no campo *Grupo de templates* (e selecione um da lista sugerida) ou pressione o botão *Selecionar* ao lado do campo *Grupo de templates* para abrir um pop-up de *Grupos de templates*. Depois de escolher um grupo de templates, o pop-up *Templates* é atualizado para mostrar os templates pertencentes a esse grupo. Os templates selecionados serão vinculados quando a configuração for salva ou atualizada. Para desvincular um template, use uma das duas opções no bloco *Templates*: *Desvincular* - desvincula o template, mas preserva suas entidades (items, triggers, etc.); *Desvincular e limpar* - desvincula o template e remove todas as suas entidades (items, triggers, etc.).
Grupos de templates	Grupos de templates aos quais o template pertence.
Descrição	Descrição do template.
Fornecedor e versão	Fornecedor e versão do template; exibido apenas ao atualizar templates existentes (templates prontos para uso fornecidos pelo Zabbix, templates importados ou templates modificados através da API de Template) se a configuração do template contiver essas informações. Não pode ser modificado no frontend do Zabbix. Para templates prontos para uso, a versão é exibida da seguinte forma: versão principal do Zabbix, delimitador (”-”), número de revisão (aumentado a cada nova versão do template e reiniciado a cada versão principal do Zabbix). Por exemplo, 6.4-0, 6.4-5, 7.0-0, 7.0-3.

A aba **Tags** permite definir tags em nível de template. Todos os problemas dos hosts vinculados a este template serão marcados com os valores inseridos aqui.

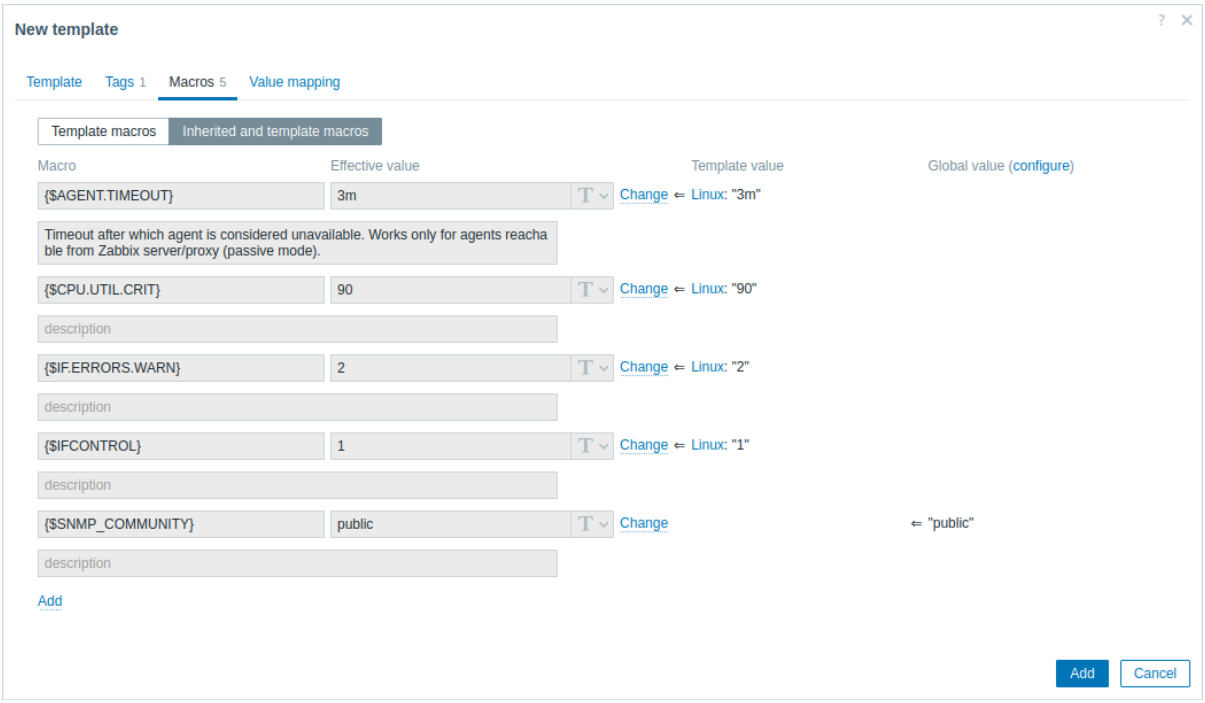


Macros de usuário, macros {INVENTORY.*}, {HOST.HOST}, {HOST.NAME}, {HOST.CONN}, {HOST.DNS}, {HOST.IP}, {HOST.PORT} e {HOST.ID} são suportadas em tags.

A aba **Macros** permite definir macros de usuário em nível de template como pares nome-valor. Observe que os valores das macros podem ser mantidos como texto simples, texto secreto ou segredo do Vault. Também é possível adicionar uma descrição.



Se você selecionar a opção *Macros herdadas e do template*, também poderá visualizar macros de templates vinculados e macros globais que o template herdará, bem como os valores que as macros serão resolvidas.



Para maior comodidade, são fornecidos links para os respectivos templates, bem como um link para a configuração de macro global. Também é possível editar uma macro de template vinculada ou macro global no nível do template, criando efetivamente uma cópia da macro no template.

A aba **Mapeamento de valores** permite configurar uma representação amigável dos dados do item em mapeamentos de valores.

Botões:

Add	Adiciona o template. O template adicionado deve aparecer na lista.
Update	Atualiza as propriedades de um template existente.
Clone	Cria outro template com base nas propriedades do template atual. Isso **inclui** as entidades (items, triggers, etc.) herdadas de templates vinculados e diretamente anexadas ao template atual, mas **exclui** o fornecedor e a versão do template atual para que o template clonado seja distinguível do original.
Delete	Exclui o template; as entidades do template (items, triggers, etc.) permanecem com os hosts vinculados.
Delete and clear	Exclui o template e todas as suas entidades dos hosts vinculados.

Cancel	Cancela a edição das propriedades do template.

Adicionando items, triggers, gráficos

> **Attention:**
> Os items devem ser adicionados primeiro a um template. Triggers e gráficos não podem ser adicionados sem o respectivo item.

Existem duas maneiras de adicionar items ao template:

1. Para criar novos items, siga as orientações em Criando um item.
2. Para adicionar items existentes ao template:

- Vá para *Coleta de dados* > *Hosts* (ou *Templates*).
- Clique em *Items* na linha do host/template desejado.
- Marque as caixas de seleção dos items que deseja adicionar ao template.
- Clique em *Copiar* abaixo da lista de items.
- Selecione o template (ou grupo de templates) para o qual os items devem ser copiados e clique em *Copiar*.
 Todos os items selecionados devem ser copiados para o template.

A adição de triggers e gráficos é feita de forma semelhante (a partir da lista de triggers e gráficos, respectivamente), lembrando novamente que eles só podem ser adicionados se os items necessários forem adicionados primeiro.

Adicionando dashboards

Para adicionar dashboards a um template em *Coleta de dados* → *Templates*, faça o seguinte:

1. Clique em *Dashboards* na linha do template.
2. Configure um dashboard seguindo as diretrizes de configuração de dashboards.

> **Attention:**
> Ao configurar widgets em um dashboard de template (em vez de um dashboard global), os parâmetros relacionados ao host não estão disponíveis e alguns parâmetros possuem um rótulo diferente. Isso ocorre porque os dashboards de template exibem dados apenas do host ao qual o template está vinculado. Por exemplo, os parâmetros *Grupos de hosts*, *Excluir grupos de hosts* e *Hosts* no widget *Problemas* não estão disponíveis, o parâmetro *Grupos de hosts* no widget *Disponibilidade do host* não está disponível, e o parâmetro *Exibir hosts em manutenção* é renomeado para *Exibir dados em manutenção*, etc. Para mais informações sobre a disponibilidade de parâmetros em widgets de dashboard de template, consulte os parâmetros específicos de cada widget de dashboard.

> **Note:**
> Para detalhes sobre como acessar dashboards de host que são criados a partir de dashboards de template, consulte a seção dashboards de host.

Configurando regras de descoberta de baixo nível

Consulte a descoberta de baixo nível seção do manual.

Adicionando cenários web

Para adicionar cenários web a um template em *Coleta de dados* → *Templates*, faça o seguinte:

1. Clique em *Web* na linha do template.
2. Configure um cenário web seguindo o método usual de configuração de cenários web.

2 Configurando um grupo de templates

Visão geral

Os grupos de templates são usados para o agrupamento lógico de templates e para atribuir permissões de usuário a eles.

Cada template deve ter pelo menos um grupo de template atribuído. Um template pode pertencer a vários grupos de templates, e cada grupo de templates pode conter vários templates.

Observe que no Zabbix todas as permissões são baseadas em grupos: grupos de usuários, grupos de hosts e grupos de templates. Portanto, mesmo que um único usuário precise de acesso a um único template, isso é concedido adicionando o usuário a um grupo de usuários que tenha permissão para acessar o grupo de templates que contém esse template.

Configuração

> **Attention:**
> Apenas usuários Super admin podem criar grupos de template.

Existem duas opções para criar um grupo de template no frontend do Zabbix.

Opção um:

- Vá para: *Coleta de dados* → *Grupos de template*
- Clique em *Criar grupo de template* no canto superior direito da tela
- Insira o nome do grupo no formulário

Opção dois: ao configurar um template, insira um nome de grupo inexistente no campo de entrada *Grupos de template*.

Depois que o grupo de template for criado, você pode clicar no nome do template na lista em *Coleta de dados* → *Grupos de template* para editar o nome do grupo, clonar o grupo ou excluir o grupo.

Excluir um grupo de template exclui apenas o grupo lógico, não os templates do grupo. Não é possível excluir um grupo de template que seja o único grupo de qualquer template existente.

Criando subgrupos de template

Um subgrupo de template (ou grupo de template aninhado) é um filho do grupo de template pai que o contém.

Um subgrupo é criado usando a barra '/' no campo de entrada do nome do grupo para denotar sua relação com o(s) grupo(s) pai(s). Por exemplo:

- inserir `Servidores Linux/Bancos de dados` cria o subgrupo `Servidores Linux/Bancos de dados` do grupo pai `Servidores Linux`.
- inserir `Servidores Linux/Bancos de dados/MySQL/Tokyo` cria o respectivo subgrupo dentro dos grupos pais aninhados `Servidores Linux`, `Servidores Linux/Bancos de dados`, `Servidores Linux/Bancos de dados/MySQL`.

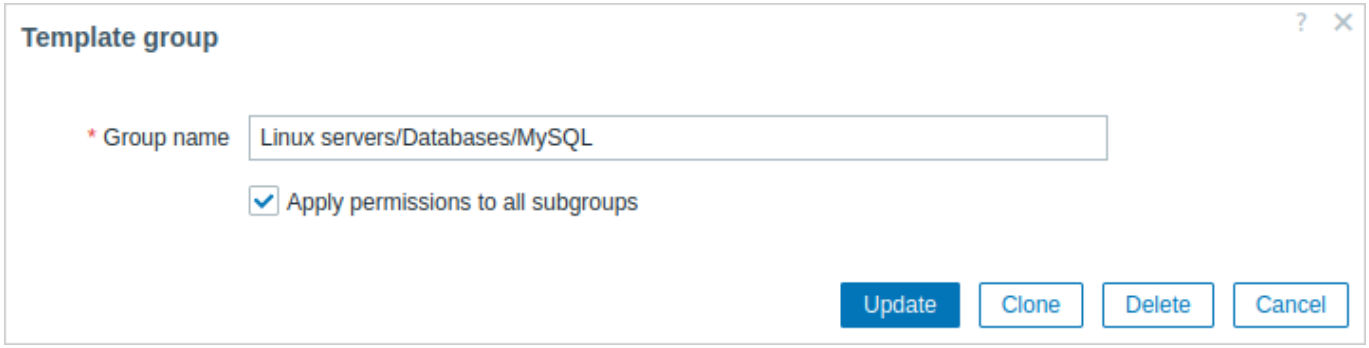
Ao criar um subgrupo, não é permitido usar barras no início ou no final, nem várias barras seguidas. Não há suporte para escape de '/'.

Não é necessário criar nenhum grupo de template pai antes de criar um subgrupo. Você pode escolher se deseja começar criando um subgrupo (por exemplo, `Servidores Linux/Bancos de dados`) ou qualquer grupo de template pai (em nosso exemplo, `Servidores Linux`). Se você começar criando um subgrupo, o(s) grupo(s) de template pai **não** será(ão) criado(s) automaticamente.

Permissões para grupos de templates

- Ao criar um subgrupo para um grupo de template pai existente (por exemplo, criar `Linux servers/Databases` quando `Linux servers` já existe), as permissões do grupo de usuários para o subgrupo são herdadas do pai.
- Ao criar um grupo de template pai para um subgrupo existente (por exemplo, criar `Linux servers` quando `Linux servers/Databases` já existe), nenhuma permissão para o pai é definida.

Ao editar qualquer grupo de template, você também pode definir uma opção adicional, *Aplicar permissões a todos os subgrupos*.



Marcar esta caixa de seleção e clicar em *Atualizar* aplicará o mesmo nível de permissões a todos os subgrupos atuais e futuros do grupo de template que está sendo editado.

Portanto, se algum grupo de usuários recebeu diferentes permissões para os subgrupos do grupo de template que está sendo editado, marcar a caixa de seleção concederá a todos os subgrupos atuais e futuros as mesmas permissões de usuário que o grupo que está sendo editado.

Observe que esta opção não é salva no banco de dados e substituirá as permissões existentes. Quaisquer alterações feitas por meio desta opção só podem ser revertidas manualmente.

3 Vinculando/desvinculando

Visão geral

O link é um processo pelo qual templates são aplicados a hosts, enquanto o desvinculamento remove a associação com o template de um host.

Vinculando um template

Para vincular um template ao host, faça o seguinte:

1. Vá para *Coleta de dados* → *Hosts*.
2. Clique no host desejado.
3. Comece a digitar o nome do template no campo *Templates*. Uma lista de templates correspondentes aparecerá; role para baixo para selecionar. Alternativamente, você pode clicar em *Selecionar* ao lado do campo e selecionar um ou vários templates da lista em uma janela pop-up.
4. Clique em *Adicionar/Atualizar* no formulário de atributos do host.

O host agora terá todas as entidades do template. Isso inclui items, triggers, gráficos, regras de descoberta de baixo nível, cenários web, bem como dashboards.

Attention:
Vincular vários templates ao mesmo host falhará se esses templates contiverem items com a mesma chave de item. E, como triggers e gráficos usam items, eles também não podem ser vinculados a um único host a partir de vários templates, se usarem chaves de item idênticas.

Quando entidades (items, triggers, etc.) são adicionadas a partir do template:

- entidades idênticas previamente existentes no host são atualizadas como entidades do template, e **quaisquer personalizações existentes no nível do host para a entidade são perdidas**;
- entidades do template são adicionadas;
- quaisquer entidades vinculadas diretamente que, antes da vinculação do template, existiam apenas no host permanecem inalteradas.

Nas listas, todas as entidades do template agora são prefixadas pelo nome do template, indicando que pertencem ao template em questão. O próprio nome do template (em texto cinza) é um link que permite acessar a lista dessas entidades no nível do template.

Note:
Para alguns items, como verificações externas, verificações HTTP agent, verificações simples, verificações SSH e verificações Telnet, uma interface de host é opcional. Se, no momento da vinculação de um template, o host não tiver uma interface definida, esses items serão adicionados sem uma interface. Se você adicionar uma interface de host posteriormente, ela não será atribuída automaticamente aos items já existentes. Para atribuir a nova interface de host a todos os items do template de uma só vez, desvincule o template do host e, em seguida, vincule-o novamente. Para preservar o histórico do item, use a opção *Desvincular*, não use *Desvincular e limpar*.

Se alguma entidade não estiver prefixada pelo nome do template, isso significa que ela já existia no host antes e não foi adicionada pelo template.

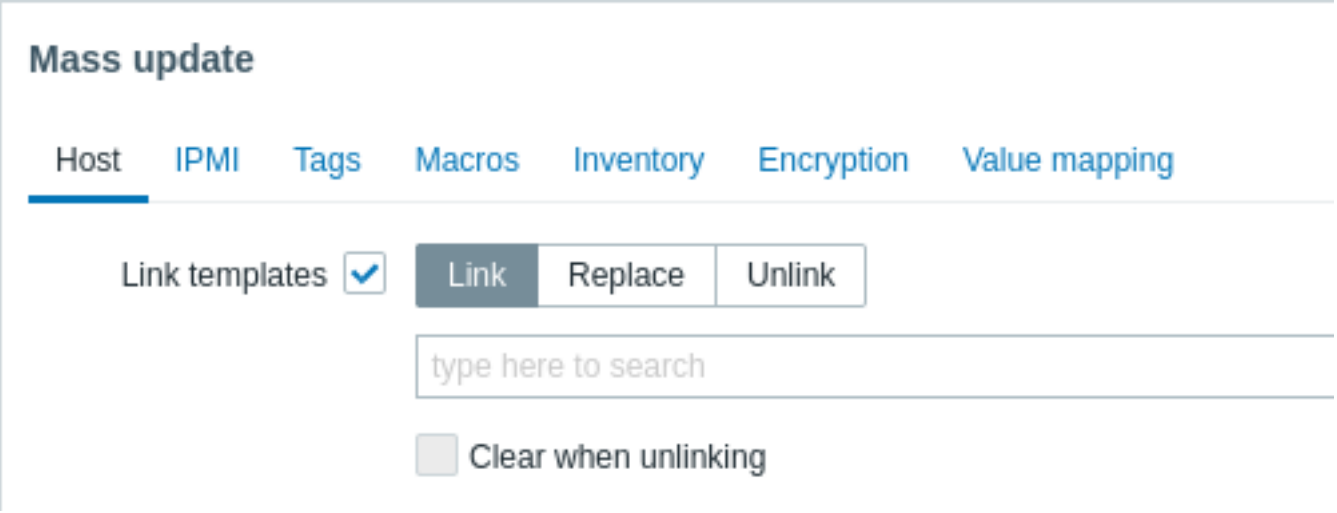
Critérios de unicidade de entidades

Ao adicionar entidades (items, triggers, etc.) de um template, é importante saber quais dessas entidades já existem no host e precisam ser atualizadas e quais entidades diferem. Os critérios de unicidade para decidir sobre a igualdade/diferença são:

- para items - a chave do item;
- para triggers - nome e expressão do trigger;
- para gráficos personalizados - nome do gráfico e seus items.

Associando templates a vários hosts

Para atualizar a associação de templates de vários hosts, em *Coleta de dados* → *Hosts* selecione alguns hosts marcando suas caixas de seleção, depois clique em **Atualização em massa** abaixo da lista e então selecione *Associar templates*:



Para associar templates adicionais, comece a digitar o nome do template no campo de auto-completar até que um menu suspenso apareça oferecendo os templates correspondentes. Basta rolar para baixo para selecionar o template a ser associado.

A opção *Substituir* permitirá associar um novo template enquanto desassocia qualquer template que estava associado aos hosts anteriormente. A opção *Desassociar* permitirá especificar quais templates desassociar. A opção *Limpar ao desassociar* permitirá não apenas desassociar qualquer template previamente associado, mas também remover todas as entidades herdadas deles (items, triggers, etc.).

Note:
O Zabbix oferece um conjunto considerável de templates predefinidos. Você pode usá-los como referência, mas tenha cuidado ao usá-los sem alterações em produção, pois podem conter muitos items e coletar dados com muita frequência. Se você quiser usá-los, ajuste-os para atender às suas necessidades reais.

Editando entidades vinculadas

Se você tentar editar um item ou trigger que foi vinculado a partir do template, poderá perceber que muitas opções principais estão desabilitadas para edição. Isso faz sentido, pois a ideia dos templates é que as coisas sejam editadas de forma centralizada no nível do template. No entanto, ainda é possível, por exemplo, habilitar/desabilitar um item em hosts individuais e definir o intervalo de atualização, o tempo de histórico e alguns outros parâmetros.

Se você quiser editar totalmente a entidade, deverá editá-la no nível do template (um atalho para o nível do template é exibido no nome do formulário), lembrando que essas alterações afetarão todos os hosts que têm esse template vinculado a eles.

Attention:
Quaisquer personalizações das entidades implementadas em nível de template substituirão as personalizações anteriores das entidades em nível de host.

Desvinculando um template

Para desvincular um template de um host, faça o seguinte:

1. Vá para *Coleta de dados* → *Hosts*.
2. Clique no host desejado e encontre o campo *Templates*.
3. Clique em *Desvincular* ou *Desvincular e limpar* ao lado do template para desvincular o template.

4. Clique em *Atualizar* no formulário de atributos do host.

Escolher a opção *Desvincular* simplesmente removerá a associação com o template, deixando todas as suas entidades com o host. Isso inclui items, triggers, gráficos, regras de descoberta de baixo nível e cenários web, mas exclui dashboards. Observe que mapas de valores e tags herdados de templates vinculados também serão removidos.

Escolher a opção *Desvincular e limpar* removerá tanto a associação com o template quanto todas as suas entidades (items, triggers, etc.).

4 Aninhamento

Visão geral

O aninhamento é uma forma de um template englobar um ou mais outros templates.

Como faz sentido separar entidades em templates individuais para vários serviços, aplicações, etc., você pode acabar com vários templates, todos os quais podem precisar ser vinculados a vários hosts. Para simplificar o cenário, é possível vincular alguns templates juntos em um único template.

O benefício do aninhamento é que você só precisa vincular um template ao host, e o host herdará automaticamente todas as entidades dos templates que estão vinculados ao template. Por exemplo, se vinculamos *T1* e *T2* a *T3*, suplementamos *T3* com todas as entidades de *T1* e *T2*, mas não vice-versa. Se vinculamos *T1* a *T2* e *T3*, suplementamos *T2* e *T3* com entidades de *T1*.

Configurando templates aninhados

Para vincular templates, você precisa pegar um template existente (ou criar um novo) e, em seguida:

1. Abra o formulário de configuração do template.
2. Encontre o campo *Templates*.
3. Clique em *Selecionar* para abrir a janela pop-up *Templates*.
4. Na janela pop-up, escolha os templates necessários e clique em *Selecionar* para adicionar os templates à lista.
5. Clique em *Adicionar* ou *Atualizar* no formulário de configuração do template.

Assim, todas as entidades do template configurado, bem como todas as entidades dos templates vinculados, agora aparecerão na configuração do template. Isso inclui items, triggers, gráficos, regras de descoberta de baixo nível e cenários web, mas exclui dashboards. No entanto, os dashboards dos templates vinculados, ainda assim, serão herdados pelos hosts.

Para desvincular qualquer um dos templates vinculados, clique em *Desvincular* ou *Desvincular e limpar* no formulário de configuração do template e, em seguida, clique em *Atualizar*.

A opção *Desvincular* simplesmente removerá a associação com o template vinculado, mas não removerá nenhuma de suas entidades (items, triggers, etc.).

A opção *Desvincular e limpar* removerá tanto a associação com o template vinculado quanto todas as suas entidades (items, triggers, etc.).

5 Atualização em massa

Visão geral

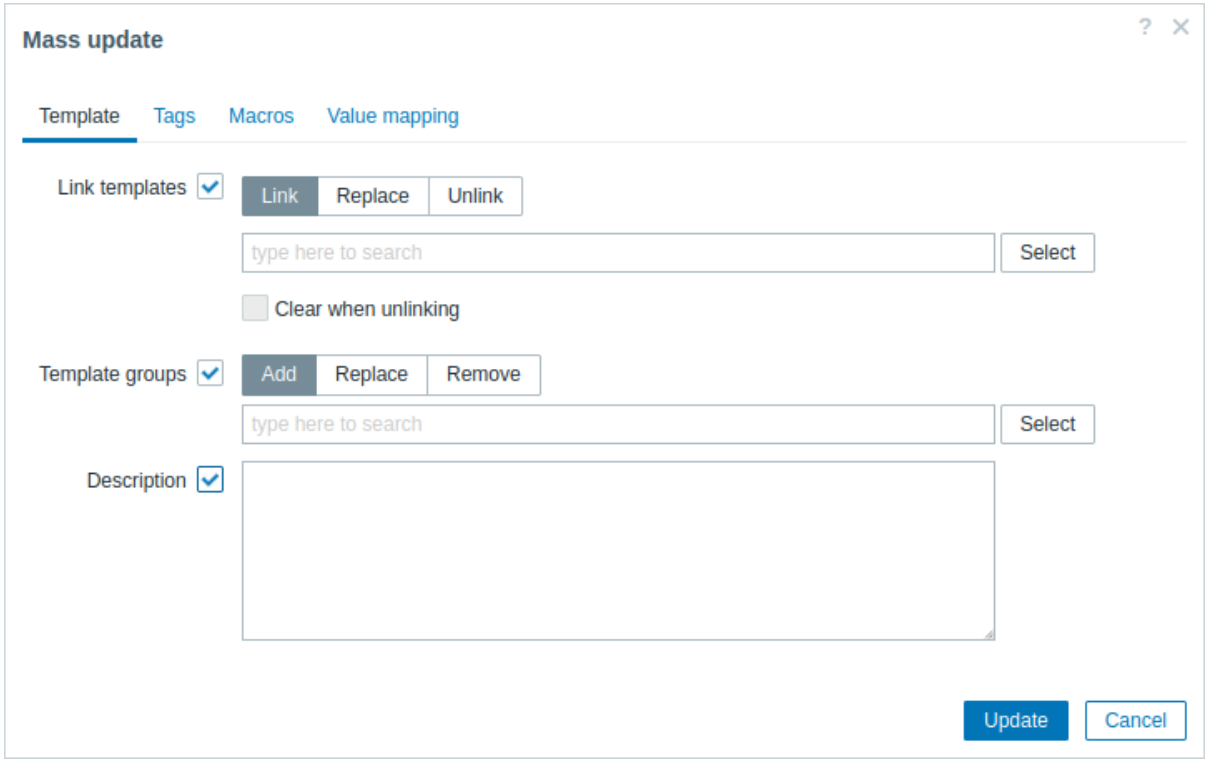
Às vezes você pode querer mudar algum atributo para um número de modelos de uma só vez. Em vez de abrir cada modelo individual para edição, você pode usar a função de atualização em massa para isso.

Usando atualização em massa

Para atualizar várias templates de uma vez, faça o seguinte:

1. Marque as caixas de seleção antes das templates que você deseja atualizar na lista de templates.
2. Clique em *Atualização em massa* abaixo da lista.
3. Navegue até a aba com os atributos necessários (*Template*, *Tags*, *Macros* ou *Mapeamento de valores*).
4. Marque as caixas de seleção de qualquer atributo a ser atualizado e insira um novo valor para eles.

A guia **Template** contém opções gerais de atualização em massa de templates.



As seguintes opções estão disponíveis ao selecionar o respectivo botão para a atualização de *Linkar templates*:

- *Linkar* - especifique quais templates adicionais serão linkados;
- *Substituir* - especifique quais templates serão linkados, desvinculando ao mesmo tempo quaisquer templates previamente vinculados;
- *Desvincular* - especifique quais templates serão desvinculados.

Para especificar os templates a serem vinculados/desvinculados, comece a digitar o nome do template no campo de preenchimento automático até que um menu suspenso apareça oferecendo os templates correspondentes. Basta rolar para baixo para selecionar os templates necessários.

A opção *Limpar ao desvincular* permitirá desvincular quaisquer templates previamente vinculados, bem como remover todos os elementos herdados deles (items, triggers, gráficos, etc.).

As seguintes opções estão disponíveis ao selecionar o respectivo botão para a atualização de *Grupos de templates*:

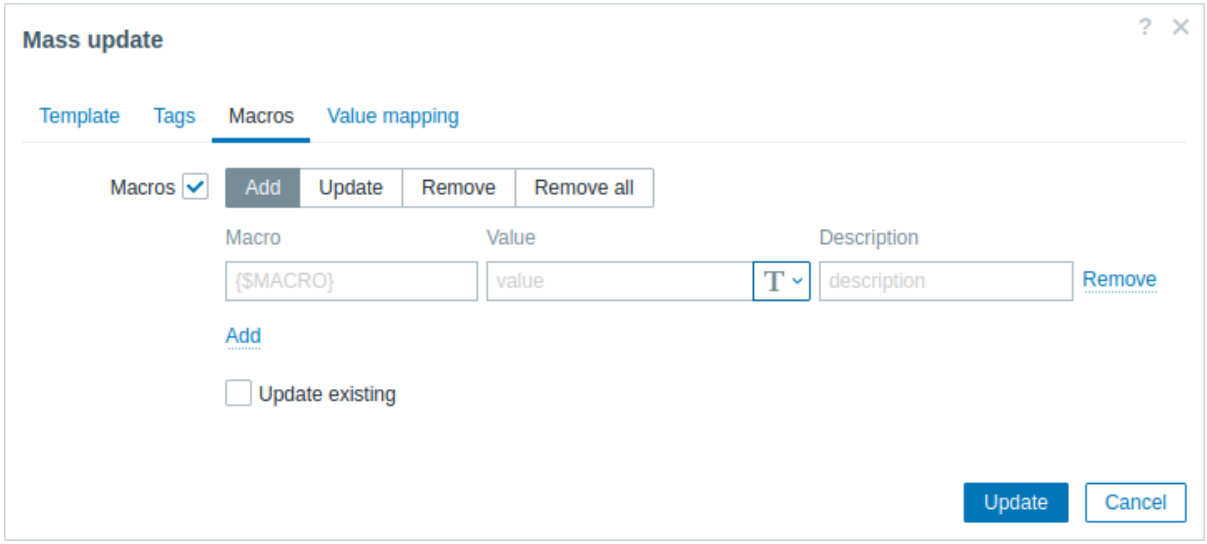
- *Adicionar* - permite especificar grupos de templates adicionais a partir dos existentes ou inserir grupos de templates completamente novos para os templates;
- *Substituir* - removerá o template de quaisquer grupos de templates existentes e os substituirá pelos especificados neste campo (grupos de templates existentes ou novos);
- *Remover* - removerá grupos de templates específicos dos templates.

Esses campos são de preenchimento automático - ao começar a digitar neles, é exibido um menu suspenso com os grupos de templates correspondentes. Se o grupo de templates for novo, ele também aparecerá no menu suspenso e será indicado por *(novo)* após a string. Basta rolar para baixo para selecionar.

A guia **Tags** permite atualizar em massa as tags no nível do template.

Macros de usuário, macros {INVENTORY.*}, {HOST.HOST}, {HOST.NAME}, {HOST.CONN}, {HOST.DNS}, {HOST.IP}, {HOST.PORT} e {HOST.ID} são suportadas em tags. Observe que tags com o mesmo nome, mas valores diferentes não são consideradas 'duplicadas' e podem ser adicionadas ao mesmo template.

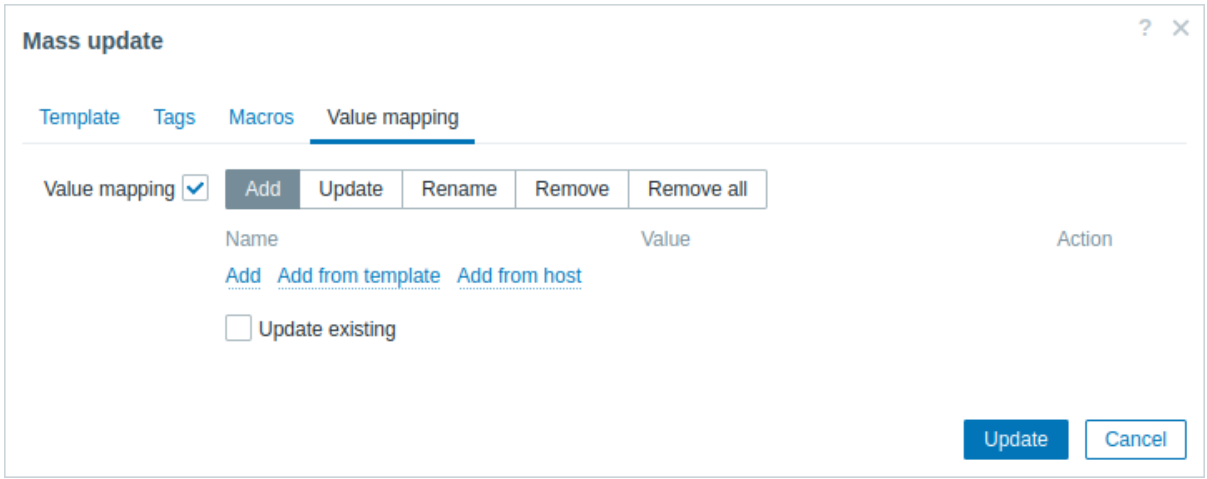
A aba **Macros** permite atualizar em massa macros no nível de template.



As seguintes opções estão disponíveis ao selecionar o respectivo botão para atualização de macros:

- *Adicionar* - permite especificar macros de usuário adicionais para os templates. Se a caixa *Atualizar existentes* estiver marcada, o valor, tipo e descrição para o nome da macro especificado serão atualizados. Se desmarcada, se uma macro com esse nome já existir no(s) template(s), ela não será atualizada.
- *Atualizar* - substituirá valores, tipos e descrições das macros especificadas nesta lista. Se a caixa *Adicionar ausentes* estiver marcada, a macro que não existia anteriormente em um template será adicionada como nova macro. Se desmarcada, apenas macros que já existem em um template serão atualizadas.
- *Remover* - removerá as macros especificadas dos templates. Se a caixa *Exceto selecionadas* estiver marcada, todas as macros, exceto as especificadas na lista, serão removidas. Se desmarcada, apenas as macros especificadas na lista serão removidas.
- *Remover todas* - removerá todas as macros de usuário dos templates. Se a caixa *Confirmo a remoção de todas as macros* não estiver marcada, uma nova janela pop-up será aberta solicitando a confirmação da remoção de todas as macros.

A guia **Mapeamento de valores** permite atualizar em massa os mapeamentos de valores.



Botões com as seguintes opções estão disponíveis para atualização de mapas de valores:

- *Adicionar* - adiciona mapas de valores aos templates. Se você marcar *Atualizar existentes*, todas as propriedades do mapa de valores com esse nome serão atualizadas. Caso contrário, se já existir um mapa de valores com esse nome, ele não será atualizado.
- *Atualizar* - atualiza mapas de valores existentes. Se você marcar *Adicionar ausentes*, um mapa de valores que não existia anteriormente em um template será adicionado como um novo mapa de valores. Caso contrário, apenas os mapas de valores que já existem em um template serão atualizados.
- *Renomear* - atribui um novo nome a um mapa de valores existente.
- *Remover* - remove os mapas de valores especificados dos templates. Se você marcar *Exceto selecionados*, todos os mapas de valores serão removidos **exceto** os que forem especificados.
- *Remover todos* - remove todos os mapas de valores dos templates. Se a caixa de seleção *Confirmo a remoção de todos os mapas de valores* não estiver marcada, uma nova janela pop-up será aberta solicitando a confirmação da remoção.

As opções *Adicionar do template* e *Adicionar do host* estão disponíveis para operações de adição/atualização de mapeamento de valores. Elas permitem selecionar mapeamentos de valores de um template ou de um host, respectivamente.

Quando terminar todas as alterações necessárias, clique em *Atualizar*. Os atributos serão atualizados de acordo para todos os templates selecionados.

1 Hosts e grupos de hosts

O que é um ”host”?

No Zabbix, um ”host” refere-se a qualquer dispositivo físico ou virtual, aplicativo, serviço ou qualquer outra coleção logicamente relacionada de parâmetros monitorados.

Criar hosts é uma das primeiras tarefas de monitoramento no Zabbix. Por exemplo, se você deseja monitorar alguns parâmetros em um servidor ”x”, você deve primeiro criar um host chamado, por exemplo, ”Servidor X” e então você pode adicionar itens de monitoramento a ele.

Hosts são organizados em grupos de hosts.

Prossiga para criar e configurar um host.

1 Configurando um host

Visão geral

Para configurar um host no frontend do Zabbix, faça o seguinte:

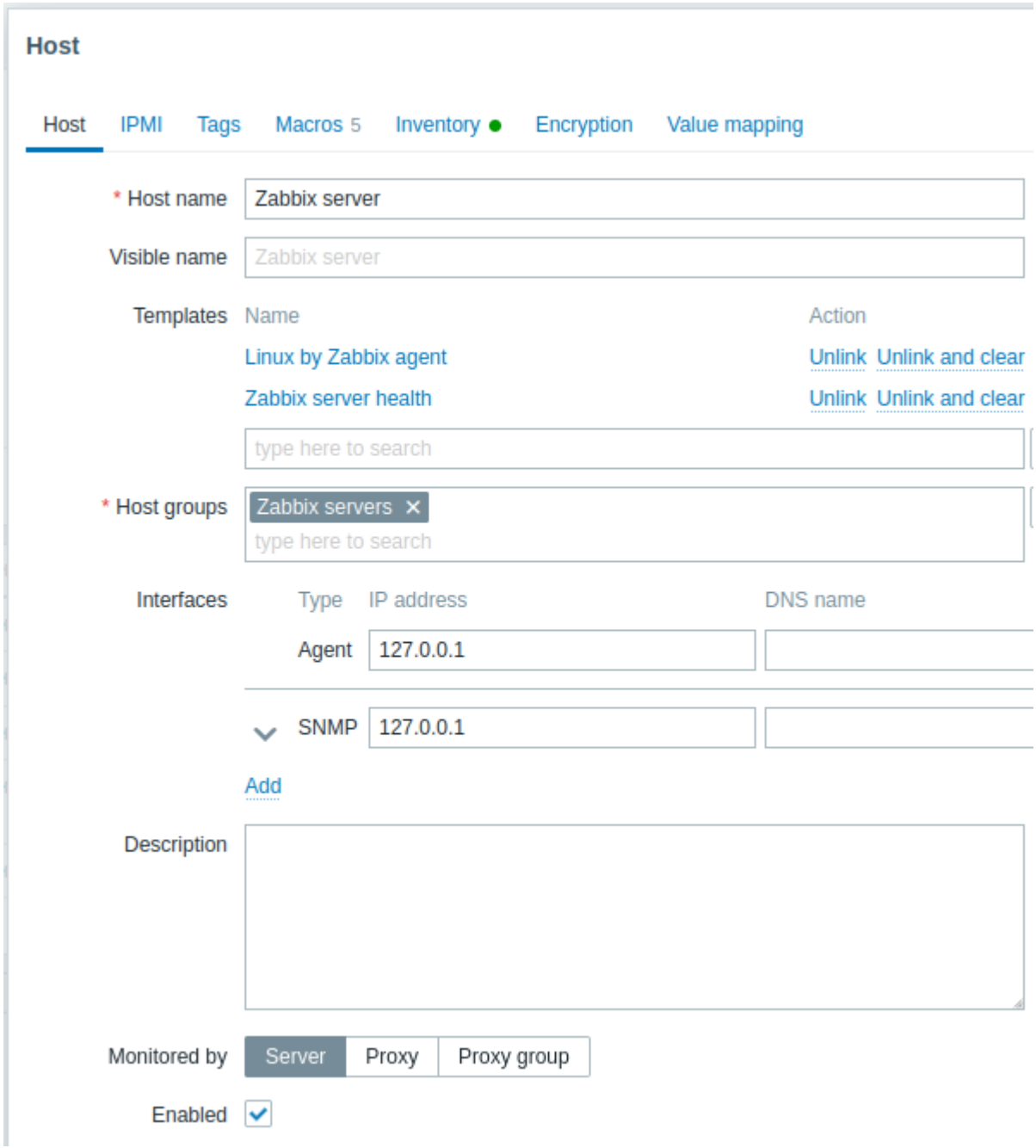
- Vá para: *Coleta de dados* > *Hosts* ou *Monitoramento* > *Hosts*
- Clique em *Criar host* no canto superior direito da tela (ou no nome do host para editar um host existente)
- Insira os parâmetros do host no formulário

Você também pode usar o botão *Clonar* no formulário de configuração de um host existente para criar um novo host. Esse host terá todas as propriedades do host existente, incluindo templates vinculados, entidades (items, triggers, etc.) desses templates, bem como as entidades diretamente vinculadas ao host existente.

Observe que, quando um host é clonado, ele manterá todas as entidades do template conforme estão originalmente no template. Quaisquer alterações nessas entidades feitas no nível do host existente (como alteração do intervalo do item, modificação da expressão regular ou adição de protótipos à regra de descoberta de baixo nível) não serão clonadas para o novo host; em vez disso, elas serão como estão no template.

Configuração

A aba **Host** contém os atributos gerais do host:



Todos os campos obrigatórios são marcados com um asterisco vermelho.

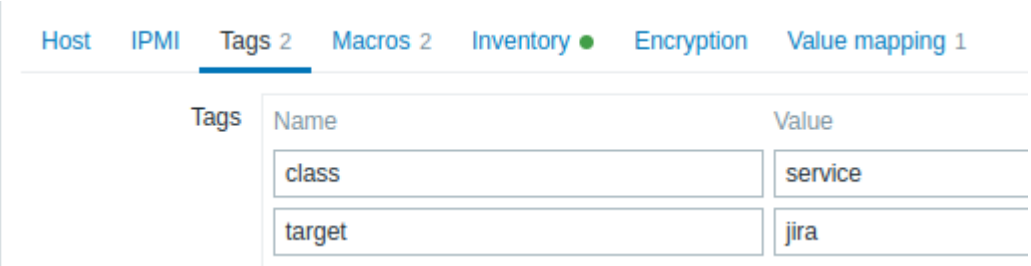
Parâmetro	Descrição
Nome do host	Digite um nome de host exclusivo. Alfanuméricos, espaços, pontos, traços e sublinhados são permitidos. No entanto, espaços à esquerda e à direita não são permitidos. *Nota:* Com o agent Zabbix em execução no host que você está configurando, o parâmetro *Hostname* no arquivo de configuração do agent deve ter o mesmo valor que o nome do host inserido aqui. O nome no parâmetro é necessário no processamento de checagens ativas.
Nome visível	Digite um nome visível exclusivo para o host. Se você definir esse nome, ele será o visível em listas, mapas, etc., em vez do nome técnico do host. Esse atributo tem suporte a UTF-8.

Parâmetro	Descrição
Templates	Vincule templates ao host. Todas as entidades (items, triggers, etc.) serão herdadas do template. Para vincular um novo template, comece a digitar o nome do template no campo de texto. Uma lista de templates correspondentes aparecerá; role para baixo para selecionar. Alternativamente, você pode clicar em *Selecionar* ao lado do campo e selecionar templates da lista em uma janela pop-up. Todos os templates selecionados serão vinculados ao host quando o formulário de configuração do host for salvo ou atualizado. Para desvincular um template, use uma das duas opções no bloco *Templates vinculados*: *Desvincular* - desvincula o template, mas preserva suas entidades (items, triggers, etc.); *Desvincular e limpar* - desvincula o template e remove todas as suas entidades (items, triggers, etc.). Os nomes dos templates listados são links clicáveis que levam ao formulário de configuração do template.
Grupos de hosts	Selecione os grupos de hosts aos quais o host pertence. Um host deve pertencer a pelo menos um grupo de hosts. Um novo grupo pode ser criado e vinculado ao host adicionando um nome de grupo inexistente.
Interfaces	Vários tipos de interface de host são suportados para um host: *Agent*, *SNMP*, *JMX* e *IPMI*. Nenhuma interface é definida por padrão. Para adicionar uma nova interface, clique em *Adicionar* no bloco *Interfaces*, selecione o tipo de interface e insira as informações de *IP/DNS*, *Conectar a* e *Porta*. *Nota:* Interfaces que são usadas em qualquer item não podem ser removidas e o link *Remover* fica desabilitado para elas. O ”IP” ou ”DNS” de uma interface SNMP também é usado para SNMP traps. Durante a correspondência, apenas o ”IP” ou ”DNS” selecionado na interface do host é usado. Veja Configurando monitoramento SNMP para detalhes adicionais sobre a configuração de uma interface SNMP (v1, v2 e v3).
Endereço IP	Endereço IP do host (opcional).
Nome DNS	Nome DNS do host (opcional).
Conectar a	Clicar no respectivo botão informará ao Zabbix server o que usar para recuperar dados dos agents: **IP** - Conectar ao endereço IP do host (recomendado) **DNS** - Conectar ao nome DNS do host
Porta	Número da porta TCP/UDP. Os valores padrão são: 10050 para agent Zabbix, 161 para agent SNMP, 12345 para JMX e 623 para IPMI.
Padrão	Marque o botão de opção para definir a interface padrão.
Descrição	Digite a descrição do host.
Monitorado por	Selecione se o host é monitorado por: **Server** - o host é monitorado pelo Zabbix server; **Proxy** - o host é monitorado por um proxy independente; **Grupo de proxy** - o host é monitorado por um grupo de proxy.
Proxy	O nome do proxy atribuído é exibido (somente se o Zabbix server tiver atribuído um do grupo de proxy selecionado). Este campo é exibido apenas se o host for monitorado por um grupo de proxy.
Habilitado	Quando a caixa de seleção está marcada, o host está habilitado - pronto para monitoramento. Quando a caixa de seleção está desmarcada, o host está desabilitado - não monitorado: Para solicitações de dados passivos iniciadas pelo Zabbix server/proxy (por exemplo, Zabbix agent, SNMP agent, checagens simples), o monitoramento é desabilitado após a sincronização da configuração. Triggers e ações vinculadas ao host também são desabilitadas somente após o recarregamento da configuração. Para checagens ativas do agent Zabbix, o monitoramento para dentro do intervalo de tempo (aprox. 5 segundos) em que o agent Zabbix recebe informações sobre o host ter sido desabilitado. Durante esse breve intervalo, o host continuará a coletar dados localmente para as checagens ativas e tentará enviá-los ao server/proxy; no entanto, como o host está marcado como *Desabilitado*, o server/proxy rejeitará os dados.

A aba **IPMI** contém atributos de gerenciamento IPMI.

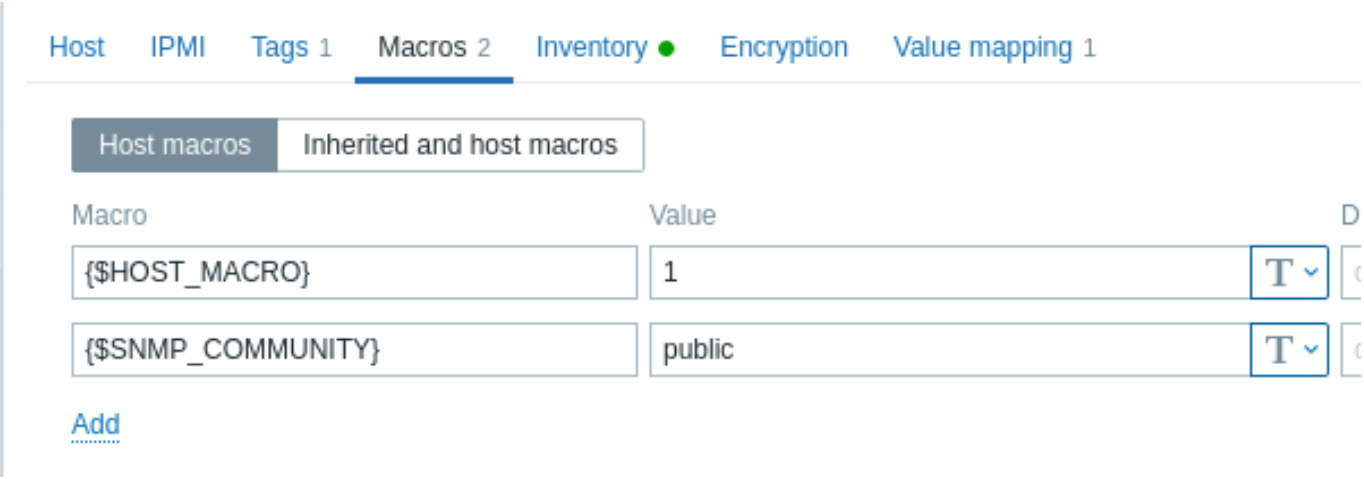
Parâmetro	Descrição
Algoritmo de autenticação	Selecione o algoritmo de autenticação.
Nível de privilégio	Selecione o nível de privilégio.
Nome de usuário	Nome de usuário para autenticação. Macros de usuário podem ser usadas.
Senha	Senha para autenticação. Macros de usuário podem ser usadas.

A aba **Tags** permite definir tags em nível de host. Todos os problemas deste host serão marcados com os valores inseridos aqui.

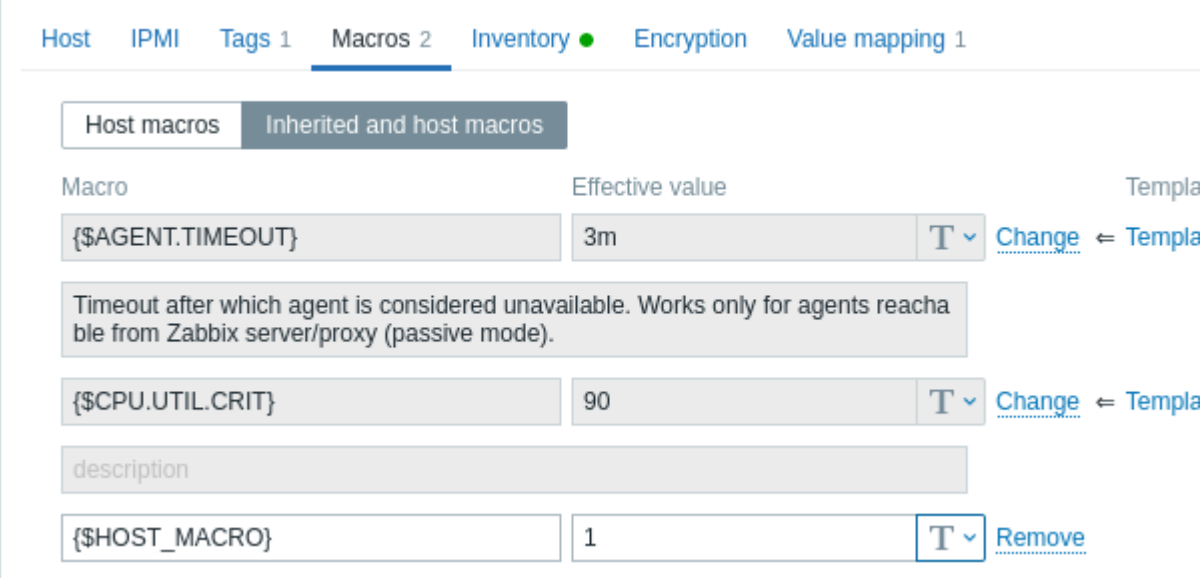


Macros de usuário, macros {INVENTORY.*}, {HOST.HOST}, {HOST.NAME}, {HOST.CONN}, {HOST.DNS}, {HOST.IP}, {HOST.PORT} e {HOST.ID} são suportadas em tags.

A aba **Macros** permite definir macros de usuário em nível de host como pares nome-valor. Observe que os valores das macros podem ser mantidos como texto simples, texto secreto ou segredo do Vault. Adicionar uma descrição também é suportado.

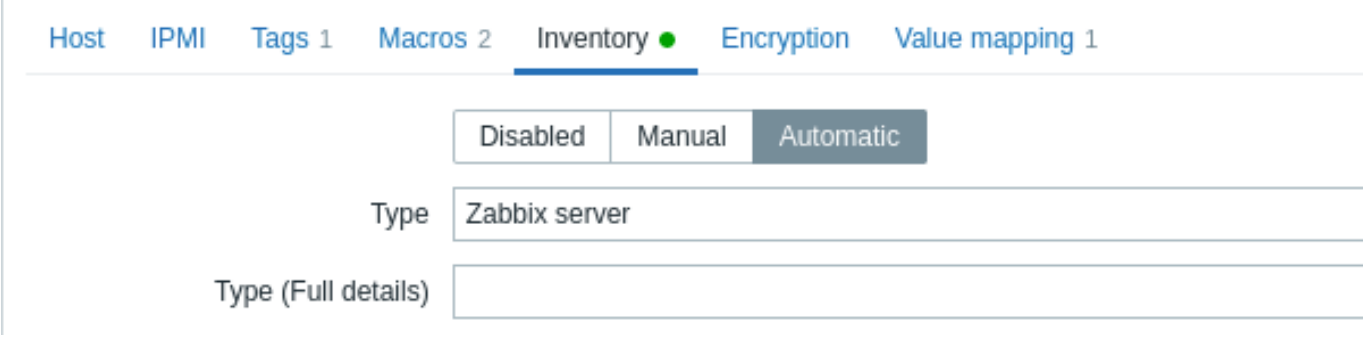


Você também pode visualizar aqui macros de usuário em nível de template e globais se selecionar a opção *Macros herdadas e do host*. É aqui que todas as macros de usuário definidas para o host são exibidas com o valor ao qual elas se resolvem, bem como sua origem.



Para maior comodidade, são fornecidos links para os respectivos templates e configuração de macro global. Também é possível editar uma macro de template/global no nível do host, criando efetivamente uma cópia da macro no host.

A aba **Inventário** permite inserir manualmente informações de inventário para o host. Você também pode selecionar para habilitar o preenchimento *Automático* do inventário ou desabilitar o preenchimento do inventário para este host.



Se o inventário estiver habilitado (manual ou automático), um ponto verde é exibido com o nome da aba.

Criptografia

A aba **Criptografia** permite exigir conexões criptografadas com o host.

Parâmetro	Descrição
Conexões para o host	Como o Zabbix server ou proxy se conecta ao Zabbix agent em um host: sem criptografia (padrão), usando PSK (chave pré-compartilhada) ou certificado.
Conexões do host	Selecione que tipo de conexões são permitidas a partir do host (ou seja, do Zabbix agent e Zabbix sender). Vários tipos de conexão podem ser selecionados ao mesmo tempo (útil para testes e troca para outro tipo de conexão). O padrão é "Sem criptografia".
Emissor	Emissor permitido do certificado. O certificado é primeiro validado com a CA (autoridade certificadora). Se for válido, assinado pela CA, então o campo *Emissor* pode ser usado para restringir ainda mais a CA permitida. Este campo deve ser usado se sua instalação do Zabbix usar certificados de várias CAs. Se este campo estiver vazio, qualquer CA será aceita.
Sujeito	Sujeito permitido do certificado. O certificado é primeiro validado com a CA. Se for válido, assinado pela CA, então o campo *Sujeito* pode ser usado para permitir apenas um valor da string *Sujeito*. Se este campo estiver vazio, qualquer certificado válido assinado pela CA configurada será aceito.
Identidade PSK	String de identidade da chave pré-compartilhada. Não coloque informações sensíveis na identidade PSK, ela é transmitida sem criptografia pela rede para informar ao receptor qual PSK usar.
PSK	Chave pré-compartilhada (hex-string). Comprimento máximo: 512 dígitos hexadecimais (PSK de 256 bytes) se o Zabbix usar a biblioteca GnuTLS ou OpenSSL, 64 dígitos hexadecimais (PSK de 32 bytes) se o Zabbix usar a biblioteca mbed TLS (PolarSSL). Exemplo: 1f87b595725ac58dd977beef14b97461a7c1045b9a1c963065002c5473194952

Mapeamento de valores

A aba **Mapeamento de valores** permite configurar uma representação amigável dos dados do item em mapeamentos de valores.

2 Configurando um grupo de hosts

Visão geral

Os grupos de hosts são usados para o agrupamento lógico de hosts e para atribuir permissões de usuário a eles.

Cada host deve ter pelo menos um grupo de hosts atribuído. Um host pode pertencer a vários grupos de hosts, e cada grupo de hosts pode conter vários hosts.

Observe que no Zabbix, todas as permissões são baseadas em grupos: grupos de usuários, grupos de hosts e grupos de templates. Portanto, mesmo que um único usuário precise de acesso a um único host, isso é concedido adicionando o usuário a um grupo de usuários que tenha permissão para acessar o grupo de hosts que contém esse host.

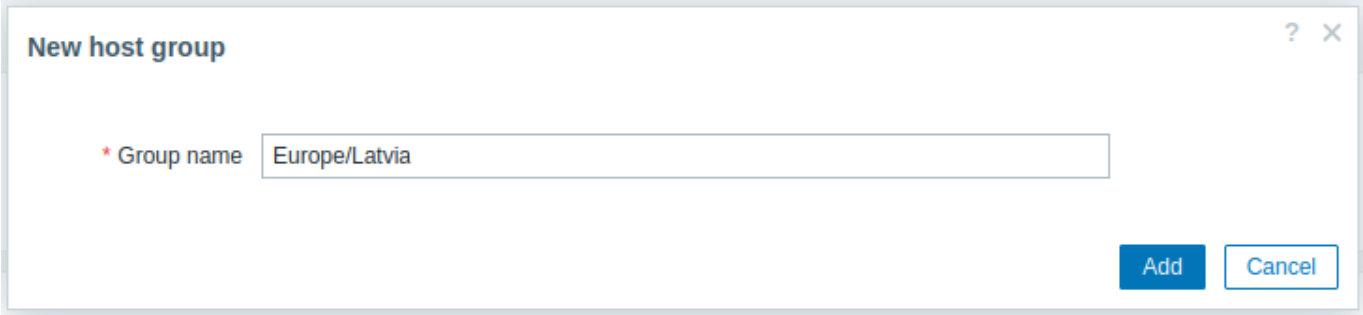
Configuração

Attention:
Somente usuários Super admin podem criar grupos de hosts.

Existem duas opções para criar um grupo de hosts no frontend do Zabbix.

Opção um:

- Vá para: *Coleta de dados* → *Grupos de hosts*
- Clique em *Criar grupo de hosts* no canto superior direito da tela
- Insira o nome do grupo no formulário



Opção dois: ao configurar um host, insira um nome de grupo inexistente no campo de entrada *Grupos de hosts*.

Depois que o grupo de hosts for criado, você pode clicar no nome do grupo na lista em *Coleta de dados* → *Grupos de hosts* para editar o nome do grupo, clonar o grupo ou excluir o grupo.

Excluir um grupo de hosts exclui apenas o grupo lógico, não os hosts do grupo. Não é possível excluir um grupo de hosts que seja o único grupo de qualquer host existente.

Criando subgrupos de hosts

Um subgrupo de host (ou grupo de host aninhado) é um filho do grupo de host pai que o contém.

Um subgrupo é criado usando a barra '/' no campo de entrada do nome do grupo para denotar sua relação com o(s) grupo(s) pai(s). Por exemplo:

- inserir `Europe/Latvia` cria o subgrupo `Europe/Latvia` do grupo pai `Europe`.
- inserir `Europe/Latvia/Riga/Zabbix servers` cria o respectivo subgrupo dentro dos grupos pai aninhados `Europe`, `Europe/Latvia`, `Europe/Latvia/Riga`.

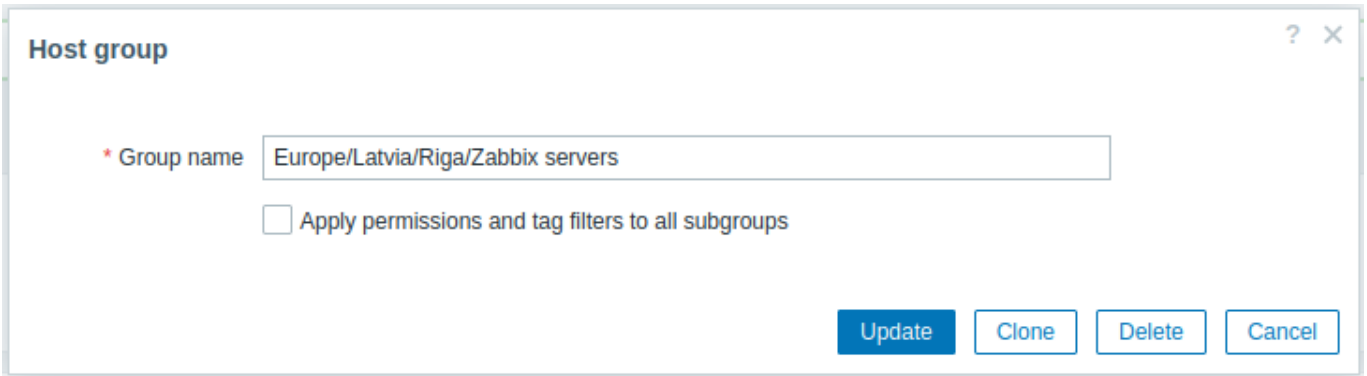
Ao criar um subgrupo, não é permitido usar barras no início ou no final, nem várias barras seguidas. Não há suporte para escape de '/'.

Não é necessário criar nenhum grupo de host pai antes de criar um subgrupo. Você pode escolher se deseja começar criando um subgrupo (por exemplo, `Europe/Latvia`) ou qualquer grupo de host pai (em nosso exemplo, `Europe`). Se você começar criando um subgrupo, o(s) grupo(s) de host pai **não** será(ão) criado(s) automaticamente.

Permissões para grupos de hosts

- Ao criar um subgrupo para um grupo de hosts pai existente (por exemplo, criar `Europe/Latvia` quando `Europe` já existe), as permissões do grupo de usuários para o subgrupo são herdadas do pai.
- Ao criar um grupo de hosts pai para um subgrupo existente (por exemplo, criar `Europe` quando `Europe/Latvia` já existe), nenhuma permissão para o pai é definida.

Ao editar qualquer grupo de hosts, você também pode definir uma opção adicional, *Aplicar permissões e filtros de tags a todos os subgrupos*.



Marcar esta caixa de seleção e clicar em *Atualizar* aplicará o mesmo nível de permissões e filtros de tags a todos os subgrupos atuais e futuros do grupo de hosts que está sendo editado.

Portanto, se algum grupo de usuários recebeu diferentes permissões para os subgrupos do grupo de hosts que está sendo editado, marcar a caixa de seleção concederá a todos os subgrupos atuais e futuros as mesmas permissões de usuário e permissões baseadas em tags que o grupo que está sendo editado.

Observe que esta opção não é salva no banco de dados e substituirá as permissões existentes. Quaisquer alterações feitas por meio desta opção só podem ser revertidas manualmente.

3 Inventário

Visão geral

Você pode manter o inventário de dispositivos de rede no Zabbix.

Existe um menu especial *Inventário* no frontend do Zabbix. No entanto, você não verá nenhum dado lá inicialmente e não é lá que você insere os dados. A construção dos dados de inventário é feita manualmente ao configurar um host ou automaticamente usando algumas opções de preenchimento automático.

Construindo inventário

Modo manual

Ao configurar um host, na aba *Inventário* você pode inserir detalhes como tipo de dispositivo, número de série, localização, responsável, URLs, etc. - dados que irão preencher as informações de inventário.

Se uma URL for incluída nas informações de inventário do host e começar com 'http' ou 'https', ela resultará em um link clicável na seção *Inventário*.

Modo automático

O inventário do host também pode ser preenchido automaticamente. Para que isso funcione, ao configurar um host, o modo de inventário na aba *Inventário* deve ser definido como *Automático*.

Em seguida, você pode configurar itens do host para preencher qualquer campo do inventário do host com seu valor, indicando o campo de destino com o respectivo atributo (chamado *O item preencherá o campo do inventário do host*) na configuração do item.

Itens que são especialmente úteis para a coleta automatizada de dados de inventário:

- system.hw.chassis[full|type|vendor|model|serial] - o padrão é [full], permissões de root necessárias
- system.hw.cpu[all|cpunum,full|maxfreq|vendor|model|curfreq] - o padrão é [all,full]
- system.hw.devices[pci|usb] - o padrão é [pci]
- system.hw.macaddr[interface,short|full] - o padrão é [all,full], interface é regexp
- system.sw.arch
- system.sw.os[name|short|full] - o padrão é [name]
- system.sw.packages[regexp,manager,short|full] - o padrão é [all,all,full]

Seleção do modo de inventário

O modo de inventário pode ser selecionado no formulário de configuração do host.

O modo de inventário padrão para novos hosts é selecionado com base na configuração *Modo de inventário de host padrão* em *Administração → Geral → Outros*.

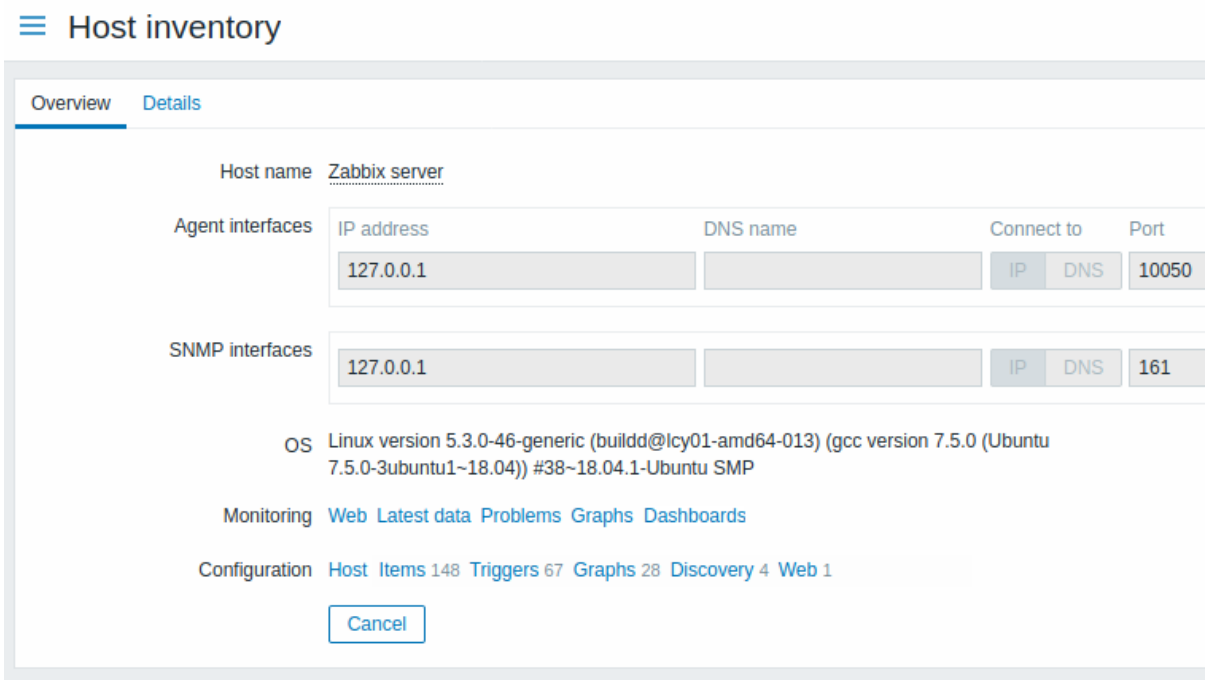
Para hosts adicionados por descoberta de rede ou ações de autorregistro, é possível definir uma operação *Definir modo de inventário do host* selecionando o modo manual ou automático. Esta operação substitui a configuração *Modo de inventário de host padrão*.

Visão geral do inventário

Os detalhes de todos os dados de inventário existentes estão disponíveis no menu *Inventário*.

Em *Inventário → Visão geral* você pode obter uma contagem de hosts por vários campos do inventário.

Em *Inventário → Hosts* você pode ver todos os hosts que possuem informações de inventário. Clicar no nome do host revelará os detalhes do inventário em um formulário.



A aba **Visão geral** mostra:

Parâmetro	Descrição
Nome do host	Nome do host. Clicar no nome abre um menu com os scripts definidos para o host. O nome do host é exibido com um ícone laranja, se o host estiver em manutenção.
Nome visível	Nome visível do host (se definido).
Host (Agent, SNMP, JMX, IPMI) interfaces	Este bloco fornece detalhes das interfaces configuradas para o host.
SO	Campo de inventário do sistema operacional do host (se definido).
Hardware	Campo de inventário de hardware do host (se definido).
Software	Campo de inventário de software do host (se definido).
Descrição	Descrição do host.
Monitoramento	Links para seções de monitoramento com dados para este host: *Web*, *Últimos dados*, *Problemas*, *Gráficos*, *Dashboards*.
Configuração	Links para seções de configuração para este host: *Host*, *Itens*, *Triggers*, *Gráficos*, *Descoberta*, *Web*. A quantidade de entidades configuradas é listada após cada link.

A aba **Detalhes** mostra todos os campos de inventário que estão preenchidos (não estão vazios).

Macros de inventário

Existem macros de inventário de host {INVENTORY.*} disponíveis para uso em notificações, por exemplo:

"O servidor em {INVENTORY.LOCATION1} tem um problema, a pessoa responsável é {INVENTORY.CONTACT1}, número de telefone {INVENTORY.POC.PRIMARY.PHONE.A1}."

Para mais detalhes, consulte a página Macros suportadas.

4 Atualização em massa

Visão geral

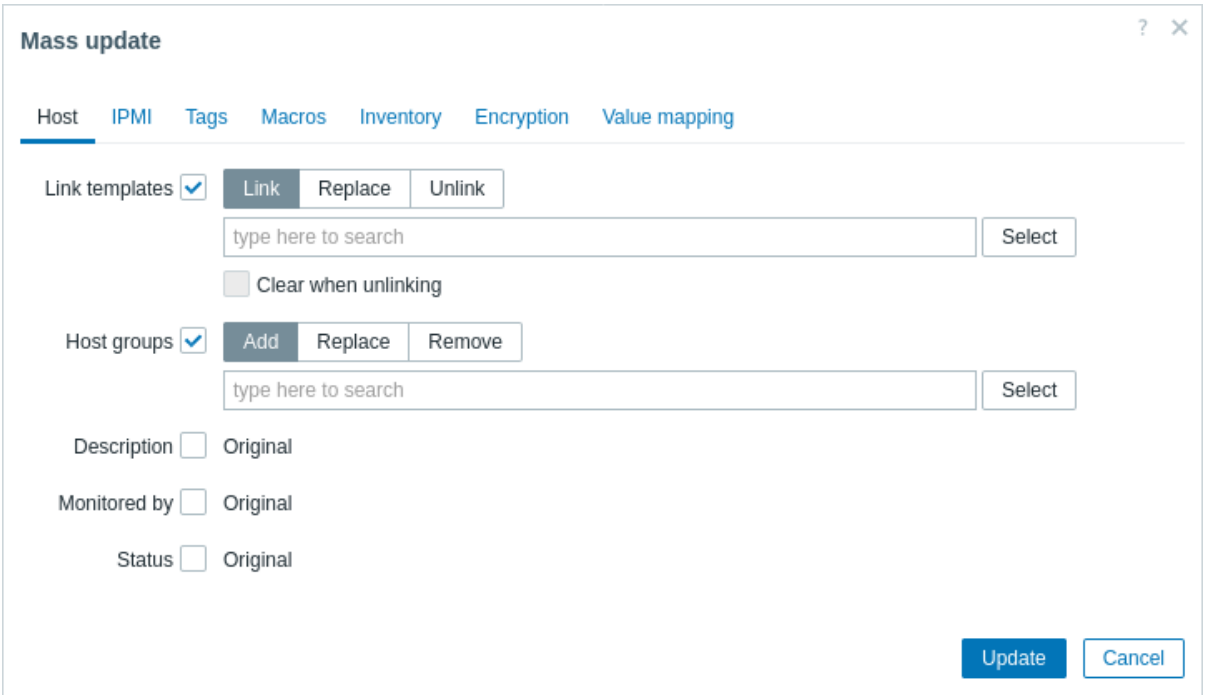
Às vezes, você pode querer alterar algum atributo para vários hosts de uma só vez. Em vez de abrir cada host individualmente para edição, você pode usar a função de atualização em massa para isso.

Usando atualização em massa

Para atualizar em massa alguns hosts, faça o seguinte:

- Marque as caixas de seleção antes dos hosts que deseja atualizar na lista de hosts
- Clique em *Atualizar em massa* abaixo da lista

- Navegue até a aba com os atributos necessários (*Host*, *IPMI*, *Tags*, *Macros*, *Inventário*, *Criptografia* ou *Mapeamento de valores*)
- Marque as caixas de seleção de qualquer atributo a ser atualizado e insira um novo valor para eles



As seguintes opções estão disponíveis ao selecionar o respectivo botão para atualização de vínculo de **template**:

- *Vincular* - especifique quais templates adicionais vincular
- *Substituir* - especifique quais templates vincular, desvinculando qualquer template que estava vinculado aos hosts anteriormente
- *Desvincular* - especifique quais templates desvincular

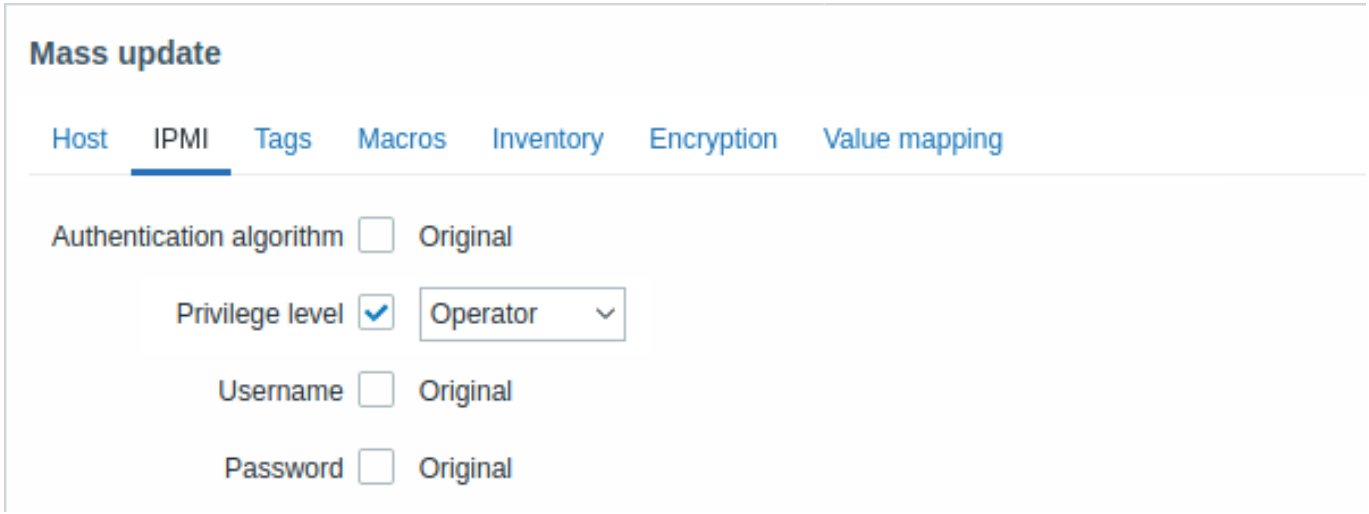
Para especificar os templates a serem vinculados/desvinculados, comece a digitar o nome do template no campo de auto-completar até que um menu suspenso apareça oferecendo os templates correspondentes. Basta rolar para baixo para selecionar o template desejado.

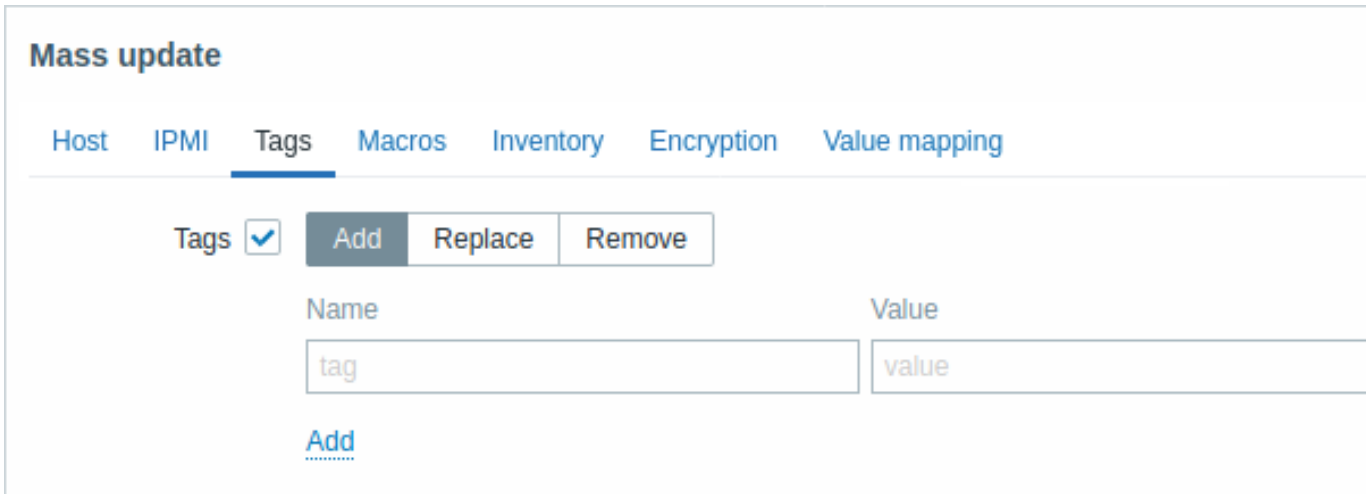
A opção *Limpar ao desvincular* permitirá não apenas desvincular quaisquer templates vinculados anteriormente, mas também remover todos os elementos herdados deles (items, triggers, etc.).

As seguintes opções estão disponíveis ao selecionar o respectivo botão para atualização de **grupo de hosts**:

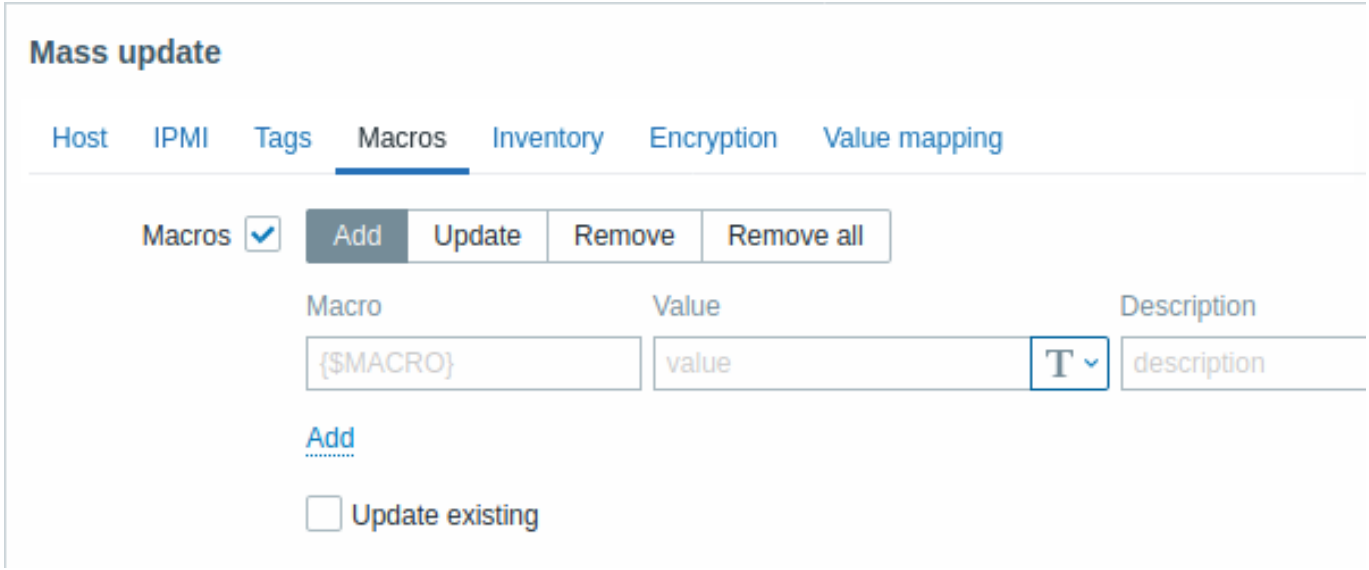
- *Adicionar* - permite especificar grupos de hosts adicionais dos existentes ou inserir grupos de hosts completamente novos para os hosts
- *Substituir* - removerá o host de quaisquer grupos de hosts existentes e os substituirá pelos especificados neste campo (grupos de hosts existentes ou novos)
- *Remover* - removerá grupos de hosts específicos dos hosts

Esses campos são de auto-completar - ao começar a digitar neles, é exibido um menu suspenso com grupos de hosts correspondentes. Se o grupo de hosts for novo, ele também aparecerá no menu suspenso e será indicado por *(novo)* após a string. Basta rolar para baixo para selecionar.





Macros de usuário, macros {INVENTORY.*}, {HOST.HOST}, {HOST.NAME}, {HOST.CONN}, {HOST.DNS}, {HOST.IP}, {HOST.PORT} e {HOST.ID} são suportadas em tags. Observe que tags com o mesmo nome, mas valores diferentes, não são consideradas 'duplicadas' e podem ser adicionadas ao mesmo host.

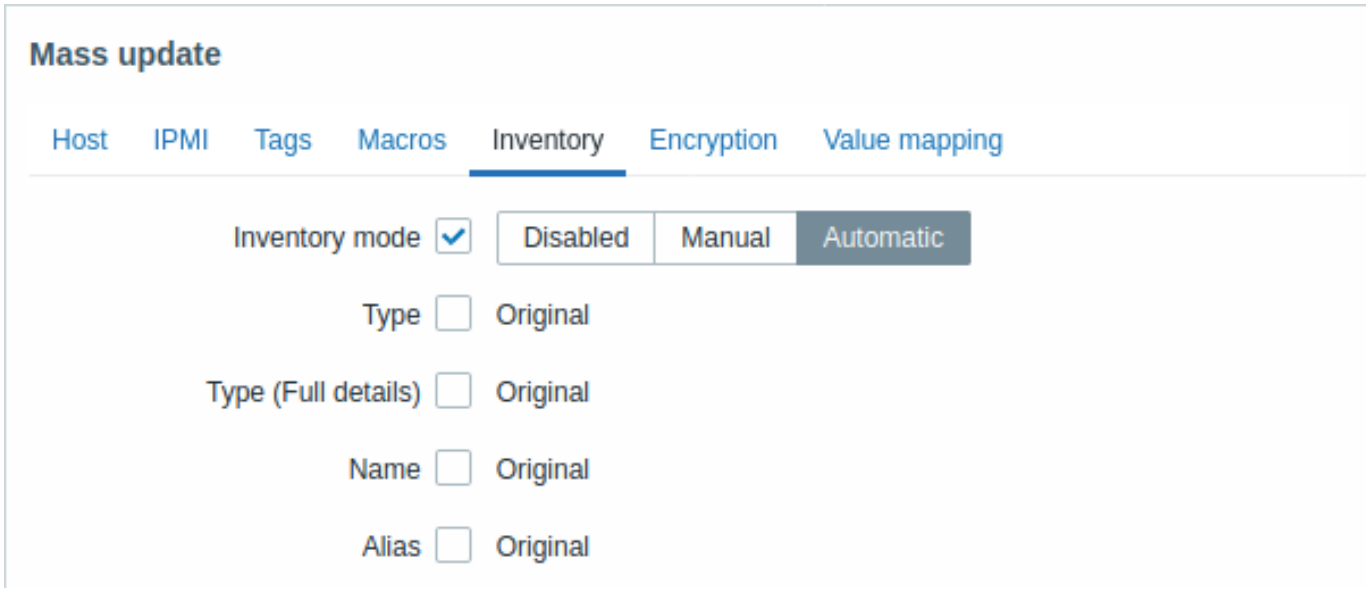


As seguintes opções estão disponíveis ao selecionar o respectivo botão para atualização de macros:

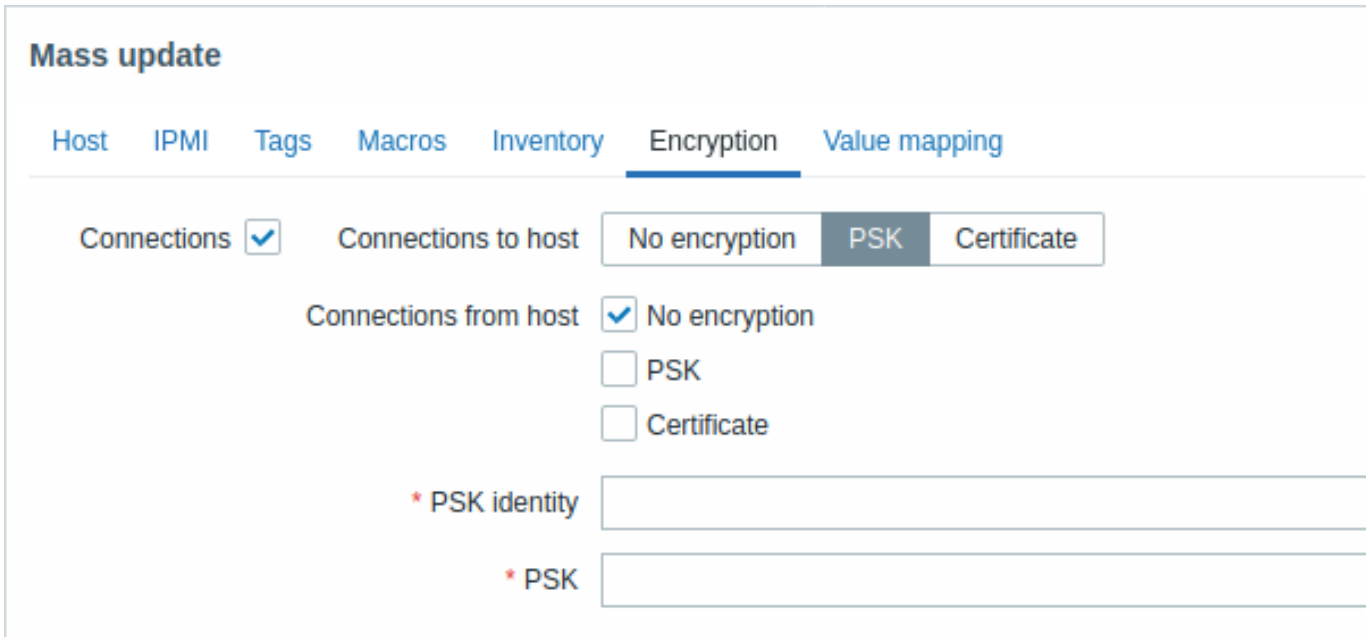
- *Adicionar* - permite especificar macros de usuário adicionais para os hosts. Se a caixa *Atualizar existentes* estiver marcada, o valor, tipo e descrição para o nome da macro especificada serão atualizados. Se desmarcada, se uma macro com esse nome já existir no(s) host(s), ela não será atualizada.
- *Atualizar* - substituirá valores, tipos e descrições das macros especificadas nesta lista. Se a caixa *Adicionar ausentes* estiver marcada, a macro que não existia anteriormente em um host será adicionada como nova macro. Se desmarcada, apenas macros que já existem em um host serão atualizadas.
- *Remover* - removerá as macros especificadas dos hosts. Se a caixa *Exceto selecionadas* estiver marcada, todas as macros,

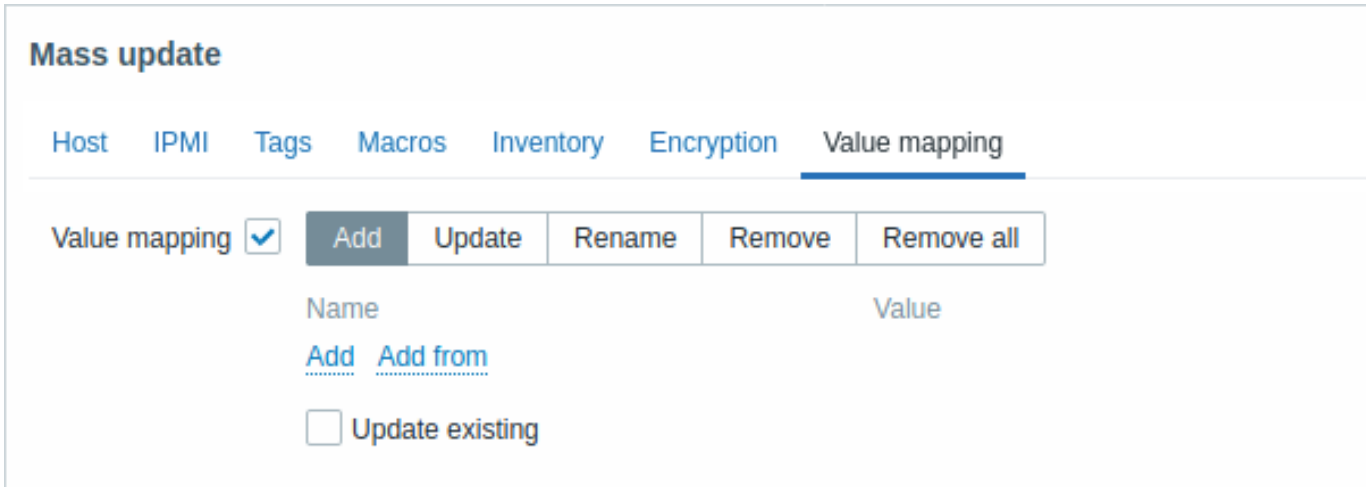
exceto as especificadas na lista, serão removidas. Se desmarcada, apenas as macros especificadas na lista serão removidas.

- *Remover todas* - removerá todas as macros de usuário dos hosts. Se a caixa *Confirmo a remoção de todas as macros* não estiver marcada, uma nova janela pop-up será aberta solicitando a confirmação da remoção de todas as macros.



Para poder atualizar em massa os campos de inventário, o *Modo de inventário* deve ser definido como 'Manual' ou 'Automático'.





Botões com as seguintes opções estão disponíveis para atualização de mapeamento de valores:

- *Adicionar* - adiciona mapeamentos de valores aos hosts. Se você marcar *Atualizar existentes*, todas as propriedades do

mapeamento de valores com esse nome serão atualizadas. Caso contrário, se um mapeamento de valores com esse nome já existir, ele não será atualizado.

- *Atualizar* - atualiza mapeamentos de valores existentes. Se você marcar *Adicionar ausentes*, um mapeamento de valores que não existia anteriormente em um host será adicionado como novo mapeamento de valores. Caso contrário, apenas os mapeamentos de valores que já existem em um host serão atualizados.
- *Renomear* - atribui um novo nome a um mapeamento de valores existente
- *Remover* - remove os mapeamentos de valores especificados dos hosts. Se você marcar *Exceto selecionados*, todos os mapeamentos de valores serão removidos **exceto** os que estão especificados.
- *Remover todos* - remove todos os mapeamentos de valores dos hosts. Se a caixa *Confirmo a remoção de todos os mapeamentos de valores* não estiver marcada, uma nova janela pop-up será aberta solicitando a confirmação da remoção.

Quando terminar todas as alterações necessárias, clique em *Atualizar*. Os atributos serão atualizados de acordo para todos os hosts selecionados.

2 Items

Visão geral

Um item é uma métrica individual.

Os itens são usados para coletar dados. Depois de configurar um host, você deve adicionar itens para obter dados reais. Uma maneira de adicionar rapidamente muitos itens é anexar um dos templates predefinidos a um host. No entanto, para otimizar o desempenho do sistema, pode ser necessário ajustar os templates para ter tantos itens e monitoramento tão frequente quanto necessário.

Para especificar que tipo de dados coletar de um host, use a chave do item. Por exemplo, um item com o nome de chave **system.cpu.load** coletará dados de carga do processador, enquanto um item com o nome de chave **net.if.in** coletará informações de tráfego de entrada.

Parâmetros adicionais podem ser especificados entre colchetes após o nome da chave. Por exemplo, system.cpu.load**[avg5]** retornará a média de carga do processador nos últimos 5 minutos, enquanto net.if.in**[eth0]** mostrará o tráfego de entrada na interface ”eth0”.

Note:
Consulte as seções individuais de tipos de itens para todos os tipos de itens e chaves de itens suportados.

Prossiga para criar e configurar um item.

Tipos de Item

Visão geral

Os tipos de item abrangem vários métodos de aquisição de dados do seu sistema. Cada tipo de item possui seu próprio conjunto de chaves de item suportadas e parâmetros obrigatórios.

Atualmente, os seguintes tipos de item são oferecidos pelo Zabbix:

- Verificações do agent Zabbix
- Verificações do agent SNMP
- Traps SNMP
- Verificações IPMI
- Verificações simples
  - Monitoramento VMware
- Monitoramento de arquivos de log
- Itens calculados
  - Cálculos agregados
- Verificações internas do Zabbix
- Verificações SSH
- Verificações Telnet
- Verificações externas
- Itens trapper
- Monitoramento JMX
- Verificações ODBC

- Itens dependentes
- Verificações HTTP
- Verificações Prometheus
- Itens de script
- Itens de navegador

Detalhes para todos os tipos de item estão incluídos nas subpáginas desta seção. Embora os tipos de item ofereçam muitas opções para coleta de dados, existem opções adicionais por meio de parâmetros de usuário ou módulos carregáveis.

Algumas verificações são realizadas apenas pelo server Zabbix (como monitoramento sem agent), enquanto outras exigem o agent Zabbix ou até mesmo o Java gateway Zabbix (com monitoramento JMX).

> **Attention:**
> Se um determinado tipo de item exigir uma interface específica (como uma verificação IPMI que precisa de uma interface IPMI no host), essa interface deve existir na definição do host.

Várias interfaces podem ser definidas na definição do host: agent Zabbix, agent SNMP, JMX e IPMI. Se um item puder usar mais de uma interface, ele procurará as interfaces de host disponíveis (na ordem: Agent→SNMP→JMX→IPMI) pela primeira apropriada para ser vinculada.

Todos os itens que retornam texto (caractere, log, tipos de informação de texto) também podem retornar apenas espaços em branco (quando aplicável), definindo o valor de retorno como uma string vazia (suportado desde a versão 2.0).

1 agente Zabbix

Visão geral

Esta seção fornece detalhes sobre as chaves de item que utilizam comunicação com o agent Zabbix para coleta de dados.

Existem verificações de agent passivas e ativas. Ao configurar um item, você pode selecionar o tipo necessário:

- *Zabbix agent* - para verificações passivas
- *Zabbix agent (active)* - para verificações ativas

Observe que todas as chaves de item suportadas pelo agent Zabbix também são suportadas pelo agent Zabbix 2 de nova geração. Veja as chaves de item adicionais que você pode usar apenas com o agent 2.

Chaves de item suportadas

As chaves de item que você pode usar com o agent Zabbix estão listadas abaixo.

As chaves de item são listadas sem parâmetros e informações adicionais. Clique na chave do item para ver todos os detalhes.

Chave do item	Descrição	Grupo de item
kernel.maxfiles	O número máximo de arquivos abertos suportados pelo SO.	Kernel
kernel.maxproc	O número máximo de processos suportados pelo SO.	
kernel.openfiles	O número de descritores de arquivos abertos atualmente.	
log	O monitoramento de um arquivo de log.	Monitoramento de log
log.count	A contagem de linhas correspondentes em um arquivo de log monitorado.	
logrt	O monitoramento de um arquivo de log que é rotacionado.	
logrt.count	A contagem de linhas correspondentes em um arquivo de log monitorado que é rotacionado.	
modbus.get	Lê dados Modbus.	Modbus
net.dns	Verifica o status de um serviço DNS.	Rede
net.dns.perf	Verifica o desempenho de um serviço DNS.	
net.dns.record	Executa uma consulta DNS.	
net.if.collisions	O número de colisões fora da janela.	
net.if.discovery	A lista de interfaces de rede.	
net.if.in	As estatísticas de tráfego de entrada em uma interface de rede.	
net.if.out	As estatísticas de tráfego de saída em uma interface de rede.	
net.if.total	A soma das estatísticas de tráfego de entrada e saída em uma interface de rede.	
net.tcp.listen	Verifica se esta porta TCP está no estado LISTEN.	
net.tcp.port	Verifica se é possível fazer uma conexão TCP com a porta especificada.	
net.tcp.service	Verifica se um serviço está em execução e aceitando conexões TCP.	
net.tcp.service.perf	Verifica o desempenho de um serviço TCP.	

Chave do item	Descrição	Grupo de item
net.tcp.socket.count	Retorna o número de sockets TCP que correspondem aos parâmetros.	
net.udp.listen	Verifica se esta porta UDP está no estado LISTEN.	
net.udp.service	Verifica se um serviço está em execução e respondendo a solicitações UDP.	
net.udp.service.perf	Verifica o desempenho de um serviço UDP.	
net.udp.socket.count	Retorna o número de sockets UDP que correspondem aos parâmetros.	
proc.cpu.util	A porcentagem de utilização da CPU do processo.	Processos
proc.get	A lista de processos do SO e seus parâmetros.	
proc.mem	A memória usada pelo processo em bytes.	
proc.num	O número de processos.	
sensor	Leitura do sensor de hardware.	Sensores
system.boottime	O tempo de inicialização do sistema.	Sistema
system.cpu.discovery	A lista de CPUs/núcleos de CPU detectados.	
system.cpu.intr	As interrupções do dispositivo.	
system.cpu.load	A carga da CPU.	
system.cpu.num	O número de CPUs.	
system.cpu.switches	A contagem de trocas de contexto.	
system.cpu.util	A porcentagem de utilização da CPU.	
system.hostname	O nome do host do sistema.	
system.hw.chassis	As informações do chassi.	
system.hw.cpu	As informações da CPU.	
system.hw.devices	A listagem de dispositivos PCI ou USB.	
system.hw.macaddr	A listagem de endereços MAC.	
system.localtime	A hora do sistema.	
system.run	Executa o comando especificado no host.	
system.stat	As estatísticas do sistema.	
system.sw.arch	As informações da arquitetura do software.	
system.sw.os	As informações do sistema operacional.	
system.sw.os.get	Informações detalhadas sobre o sistema operacional (versão, tipo, nome da distribuição, versão menor e maior, etc).	
system.sw.packages	A listagem de pacotes instalados.	
system.sw.packages.get	Uma listagem detalhada de pacotes instalados.	
system.swap.in	As estatísticas de swap-in (do dispositivo para a memória).	
system.swap.out	As estatísticas de swap-out (da memória para o dispositivo).	
system.swap.size	O tamanho do espaço de swap em bytes ou em porcentagem do total.	
system.uname	Identificação do sistema.	
system.uptime	O tempo de atividade do sistema em segundos.	
system.users.num	O número de usuários conectados.	
vfs.dev.discovery	A lista de dispositivos de bloco e seu tipo.	Sistemas de arquivos virtuais
vfs.dev.read	As estatísticas de leitura do disco.	
vfs.dev.write	As estatísticas de gravação do disco.	
vfs.dir.count	A contagem de entradas do diretório.	
vfs.dir.get	A lista de entradas do diretório.	
vfs.dir.size	O tamanho do diretório.	
vfs.file.cksum	O checksum do arquivo, calculado pelo algoritmo UNIX cksum.	
vfs.file.contents	Recuperando o conteúdo de um arquivo.	
vfs.file.exists	Verifica se o arquivo existe.	
vfs.file.get	Retorna informações sobre um arquivo.	
vfs.file.md5sum	O checksum MD5 do arquivo.	
vfs.file.owner	Recupera o proprietário de um arquivo.	
vfs.file.permissions	Retorna uma string de 4 dígitos contendo o número octal com permissões UNIX.	
vfs.file.regexp	Recupera uma string no arquivo.	
vfs.file.regmatch	Encontra uma string no arquivo.	
vfs.file.size	O tamanho do arquivo.	
vfs.file.time	As informações de tempo do arquivo.	
vfs.fs.discovery	A lista de sistemas de arquivos montados com seu tipo e opções de montagem.	
vfs.fs.get	A lista de sistemas de arquivos montados com seu tipo, espaço em disco disponível, estatísticas de inode e opções de montagem.	
vfs.fs.inode	O número ou porcentagem de inodes.	
vfs.fs.size	O espaço em disco em bytes ou em porcentagem do total.	

Chave do item	Descrição	Grupo de item
vm.memory.size	O tamanho da memória em bytes ou em porcentagem do total.	Memória virtual
web.page.get	Obtém o conteúdo de uma página web.	Monitoramento web
web.page.perf	O tempo de carregamento de uma página web completa.	
web.page.regexp	Encontra uma string na página web.	
agent.hostmetadata	Os metadados do host do agent.	Zabbix
agent.hostname	O nome do host do agent.	
agent.ping	A verificação de disponibilidade do agent.	
agent.variant	A variante do agent Zabbix (agent Zabbix ou agent Zabbix 2).	
agent.version	A versão do agent Zabbix.	
zabbix.stats	Retorna um conjunto de métricas internas do server ou proxy Zabbix. Se `ip` e `port` forem especificados, as métricas são recuperadas da instância remota; caso contrário, da instância local.	
zabbix.stats	Retorna o número de items monitorados na fila que estão atrasados no server ou proxy Zabbix. Se `ip` e `port` forem especificados, as métricas são recuperadas da instância remota; caso contrário, da instância local.	

Plataformas suportadas

Exceto quando especificado de forma diferente nos detalhes do item, os itens do agent (e todos os parâmetros) são suportados em:

- **Linux**
- **FreeBSD**
- **Solaris**
- **HP-UX**
- **AIX**
- **MacOS X**
- **OpenBSD**
- **NetBSD**

Muitos itens do agent também são suportados no **Windows**. Consulte a página Itens do agent do Windows para obter detalhes.

Detalhes da chave do item

Parâmetros sem colchetes angulares são obrigatórios. Parâmetros marcados com colchetes angulares **< >** são opcionais.

kernel.maxfiles

 O número máximo de arquivos abertos suportados pelo SO.
 Valor de retorno: *Integer*.
 Plataformas suportadas: Linux, FreeBSD, MacOS X, OpenBSD, NetBSD.

kernel.maxproc

 O número máximo de processos suportados pelo SO.
 Valor de retorno: *Inteiro*.
 Plataformas suportadas: Linux 2.6 e posterior, FreeBSD, Solaris, MacOS X, OpenBSD, NetBSD.

kernel.openfiles

 O número de descritores de arquivos abertos atualmente.
 Valor de retorno: *Integer*.
 Plataformas suportadas: Linux (o item pode funcionar em outras plataformas semelhantes ao UNIX).

log[file,<regexp>,<encoding>,<maxlines>,<mode>,<output>,<maxdelay>,<options>,<persistent dir>]

 O monitoramento de um arquivo de log.
 Valor de retorno: *Log*.
 Veja as plataformas suportadas.

Parâmetros:

- **file** - o caminho completo e o nome de um arquivo de log;
- **regexp** - uma expressão regular descrevendo o padrão necessário;
- **encoding** - o identificador da página de código;
- **maxlines** - o número máximo de novas linhas por segundo que o agent enviará para o Zabbix server ou proxy. Este parâmetro sobrescreve o valor de 'MaxLinesPerSecond' em zabbix_agentd.conf;
- **mode** - valores possíveis: *all* (padrão) ou *skip* - ignora o processamento de dados antigos (afeta apenas itens recém-criados);

- **output** - um modelo de formatação de saída opcional. A sequência de escape **\0** é substituída pela parte do texto correspondente (do primeiro caractere onde a correspondência começa até o caractere onde termina), enquanto uma sequência de escape **\N** (onde N=1...9) é substituída pelo N-ésimo grupo correspondente (ou uma string vazia se N exceder o número de grupos capturados);
- **maxdelay** - o atraso máximo em segundos. Tipo: float. Valores: 0 - (padrão) nunca ignora linhas do arquivo de log; > 0.0 - ignora linhas mais antigas para que as linhas mais recentes sejam analisadas dentro de ”maxdelay” segundos. Leia as notas sobre maxdelay antes de usá-lo!
- **options** - opções adicionais:
mtime-noreread - registros não exclusivos, relê apenas se o tamanho do arquivo mudar (ignora a alteração do tempo de modificação). (Este parâmetro está obsoleto desde 5.0.2, pois agora o mtime é ignorado.)
- **persistent dir** (apenas em zabbix_agentd em sistemas Unix; não suportado no Zabbix agent 2) - o caminho absoluto do diretório onde armazenar arquivos persistentes. Veja também notas adicionais sobre arquivos persistentes.

Comentários:

- O item deve ser configurado como uma verificação ativa;
- Se o arquivo estiver ausente ou as permissões não permitirem o acesso, o item se tornará não suportado;
- Se `output` for deixado em branco - toda a linha contendo o texto correspondente será retornada. Observe que todos os tipos de expressão regular global, exceto 'Result is TRUE', sempre retornam a linha inteira correspondente e o parâmetro `output` é ignorado.
- A extração de conteúdo usando o parâmetro `output` ocorre no agent.

Exemplos:

```
log[/var/log/syslog]
log[/var/log/syslog,error]
log[/home/zabbix/logs/logfile,,,100]
```

Exemplo de uso do parâmetro `output` para extrair um número do registro de log:

```
log[/app1/app.log,"task run [0-9.]+ sec, processed ([0-9]+) records, [0-9]+ errors",,,,\1] #este item corr
```

Exemplo de uso do parâmetro `output` para reescrever um registro de log antes de enviar para o server:

```
log[/app1/app.log,"([0-9 :-]+) task run ([0-9.]+) sec, processed ([0-9]+) records, ([0-9]+) errors",,,,"\1
```

log.count[file,<regexp>,<encoding>,<maxproclines>,<mode>,<maxdelay>,<options>,<persistent dir>]

 A contagem de linhas correspondentes em um arquivo de log monitorado.
 Valor de retorno: *Integer*.
 Veja as plataformas suportadas.

Parâmetros:

- **file** - o caminho completo e o nome do arquivo de log;
- **regexp** - uma expressão regular descrevendo o padrão necessário;
- **encoding** - o identificador da página de código;
- **maxproclines** - o número máximo de novas linhas por segundo que o agent irá analisar (não pode exceder 10000). O valor padrão é 10*'MaxLinesPerSecond' em zabbix_agentd.conf.
- **mode** - valores possíveis: *all* (padrão) ou *skip* - ignora o processamento de dados antigos (afeta apenas itens recém-criados).
- **maxdelay** - o atraso máximo em segundos. Tipo: float. Valores: 0 - (padrão) nunca ignora linhas do arquivo de log; > 0.0 - ignora linhas mais antigas para que as linhas mais recentes sejam analisadas dentro de ”maxdelay” segundos. Leia as notas sobre maxdelay antes de usá-lo!
- **options** - opções adicionais:
mtime-noreread - registros não exclusivos, relê apenas se o tamanho do arquivo mudar (ignora a alteração do tempo de modificação). (Este parâmetro está obsoleto desde a versão 5.0.2, pois agora o mtime é ignorado.)
- **persistent dir** (apenas no zabbix_agentd em sistemas Unix; não suportado no Zabbix agent 2) - o caminho absoluto do diretório onde armazenar arquivos persistentes. Veja também notas adicionais sobre arquivos persistentes.

Comentários:

- O item deve ser configurado como uma verificação ativa;
- As linhas correspondentes são contadas nas novas linhas desde a última verificação de log pelo agent e, portanto, dependem do intervalo de atualização do item;
- Se o arquivo estiver ausente ou as permissões não permitirem o acesso, o item se tornará não suportado.

logrt[file regexp,<regexp>,<encoding>,<maxlines>,<mode>,<output>,<maxdelay>,<options>,<persistent dir>]

 Monitoramento de um arquivo de log que é rotacionado.
 Valor de retorno: *Log*.
 Veja as plataformas suportadas.

Parâmetros:

- **file regexp** - o caminho absoluto para o arquivo, com o nome do arquivo especificado usando uma expressão regular. Observe que a expressão regular se aplica apenas ao nome do arquivo e não precisa corresponder ao nome inteiro (por exemplo, /path/to/agent corresponderá a zabbix_agentd.log).
- **regexp** - uma expressão regular descrevendo o padrão de conteúdo necessário.
- **encoding** - o identificador da página de código.
- **maxlines** - o número máximo de novas linhas por segundo que o agent enviará para o Zabbix server ou proxy. Este parâmetro substitui o valor de 'MaxLinesPerSecond' em zabbix_agentd.conf.
- **mode** - valores possíveis: *all* (padrão) ou *skip* - ignora o processamento de dados antigos (afeta apenas items recém-criados).
- **output** - um modelo de formatação de saída opcional. A sequência de escape **\0** é substituída pela parte do texto correspondente (do primeiro caractere onde a correspondência começa até o caractere onde a correspondência termina), enquanto uma sequência de escape **\N** (onde N=1...9) é substituída pelo N-ésimo grupo correspondente (ou uma string vazia se N exceder o número de grupos capturados).
- **maxdelay** - o atraso máximo em segundos. Tipo: float. Valores: 0 - (padrão) nunca ignora linhas do arquivo de log; > 0.0 - ignora linhas mais antigas para que as linhas mais recentes sejam analisadas dentro de "maxdelay" segundos. Leia as notas sobre maxdelay antes de usá-lo!
- **options** - o tipo de rotação do arquivo de log e outras opções. Valores possíveis:
rotate (padrão),
copytruncate - observe que *copytruncate* não pode ser usado junto com *maxdelay*. Neste caso, *maxdelay* deve ser 0 ou não especificado; veja as notas sobre copytruncate,
mtime-reread - registros não exclusivos, reler se a hora de modificação ou o tamanho mudar (usado por padrão se nenhuma opção `mtime-*` for explicitamente definida),
mtime-noreread - registros não exclusivos, reler apenas se o tamanho mudar (ignorar alteração na hora de modificação).
- **persistent dir** (apenas no zabbix_agentd em sistemas Unix; não suportado no Zabbix agent 2) - o caminho absoluto do diretório onde armazenar arquivos persistentes. Veja também notas adicionais sobre arquivos persistentes.

Comentários:

- O item deve ser configurado como um active check;
- A rotação do log é baseada na última hora de modificação dos arquivos;
- Observe que logrt foi projetado para funcionar com um arquivo de log atualmente ativo, com vários outros arquivos inativos correspondentes rotacionados. Se, por exemplo, um diretório tiver muitos arquivos de log ativos, um item logrt separado deve ser criado para cada um. Caso contrário, se um item logrt capturar muitos arquivos, isso pode levar ao esgotamento da memória e a uma falha no monitoramento.
- Se `output` for deixado em branco - toda a linha contendo o texto correspondente será retornada. Observe que todos os tipos de expressão regular global, exceto 'Result is TRUE', sempre retornam a linha inteira correspondente e o parâmetro `output` é ignorado.
- A extração de conteúdo usando o parâmetro `output` ocorre no agent.
- No parâmetro `file regexp`, o caminho do diretório de log e a expressão regular do arquivo de log devem ser separados pelo separador de diretório correto:
  - no Windows, o separador deve ser uma barra invertida (\); barras normais podem ser toleradas em outras posições, exceto na que separa o caminho do diretório de log e a expressão regular do arquivo de log (veja exemplos abaixo).
  - em outros sistemas, deve ser uma barra normal (/).

Exemplos para Windows:

```
logrt["c:/dir1/dir2/dir3\filename.*\.log","pattern_to_match"] #este item coletará dados de arquivos de log
logrt["//example.com/share/dir1/dir2/dir3\filename.*\.log","pattern_to_match"] #este item coletará dados d
```

Exemplos para outros sistemas:

```
logrt["/home/zabbix/logs/^logfile[0-9]{1,3}$",,,100] #este item corresponderá a um arquivo como "logfile1"
logrt["/home/user/^logfile_.*_[0-9]{1,3}$","pattern_to_match","UTF-8",100] #este item coletará dados de ar
```

Exemplo de uso do parâmetro `output` para extrair um número do registro de log:

```
logrt[/app1/^test.*log$,"task run [0-9.]+ sec, processed ([0-9]+) records, [0-9]+ errors",,,,\1] #este ite
```

Exemplo de uso do parâmetro `output` para reescrever um registro de log antes de enviar para o server:

```
logrt[/app1/^test.*log$,"([0-9 :-]+) task run ([0-9.]+) sec, processed ([0-9]+) records, ([0-9]+) errors",
```

logrt.count[file regexp,<regexp>,<encoding>,<maxproclines>,<mode>,<maxdelay>,<options>,<persistent dir>]

 A contagem de linhas correspondentes em um arquivo de log monitorado que é rotacionado.
 Valor de retorno: *Integer*.
 Veja as plataformas suportadas.

Parâmetros:

- **file regexp** - o caminho absoluto para o arquivo, com o nome do arquivo especificado usando uma expressão regular. Observe que a expressão regular se aplica apenas ao nome do arquivo e não precisa corresponder ao nome inteiro (por

exemplo, /path/to/agent corresponderá a zabbix_agentd.log).
- **regexp** - uma expressão regular descrevendo o padrão necessário.
- **encoding** - o identificador da página de código.
- **maxproclines** - o número máximo de novas linhas por segundo que o agent irá analisar (não pode exceder 10000). O valor padrão é 10*'MaxLinesPerSecond' em zabbix_agentd.conf.
- **mode** - valores possíveis: *all* (padrão) ou *skip* - ignora o processamento de dados antigos (afeta apenas itens recém-criados).
- **maxdelay** - o atraso máximo em segundos. Tipo: float. Valores: 0 - (padrão) nunca ignora linhas do arquivo de log; > 0.0 - ignora linhas mais antigas para que as linhas mais recentes sejam analisadas dentro dos segundos definidos em "maxdelay". Leia as notas sobre maxdelay antes de usá-lo!
- **options** - o tipo de rotação do arquivo de log e outras opções. Valores possíveis:
rotate (padrão),
copytruncate - observe que *copytruncate* não pode ser usado junto com *maxdelay*. Neste caso, *maxdelay* deve ser 0 ou não especificado; veja as notas sobre copytruncate,
mtime-reread - registros não exclusivos, reler se o tempo de modificação ou tamanho mudar (padrão),
mtime-noreread - registros não exclusivos, reler apenas se o tamanho mudar (ignorar alteração no tempo de modificação).
- **persistent dir** (apenas em zabbix_agentd em sistemas Unix; não suportado no Zabbix agent 2) - o caminho absoluto do diretório onde armazenar arquivos persistentes. Veja também notas adicionais sobre arquivos persistentes.

Comentários:

- O item deve ser configurado como um active check;
- As linhas correspondentes são contadas nas novas linhas desde a última verificação de log pelo agent e, portanto, dependem do intervalo de atualização do item;
- A rotação do log é baseada no último tempo de modificação dos arquivos;
- No parâmetro `file regexp`, o caminho do diretório do log e a expressão regular do arquivo de log devem ser separados pelo separador de diretório correto:
  - no Windows, o separador deve ser uma barra invertida (\); barras normais podem ser toleradas em outras posições, exceto na que separa o caminho do diretório do log e a expressão regular do arquivo de log (veja exemplos abaixo).
  - em outros sistemas, deve ser uma barra normal (/).

Exemplos para Windows:

```
logrt.count["c:/dir1/dir2/dir3\filename.*\.log","pattern_to_match"] #este item contará o número de corresp
logrt.count["//example.com/share/dir1/dir2/dir3\filename.*\.log","pattern_to_match"] #este item contará o
```

Exemplos para outros sistemas:

```
logrt.count["/home/zabbix/logs/.*","^logfile[0-9]{1,3}$",,100] #este item contará o número de correspondên
logrt.count["/home/user/^logfile_.*_[0-9]{1,3}$","pattern_to_match","UTF-8",100] #este item contará o núme
```

modbus.get[endpoint,<slave id>,<function>,<address>,<count>,<type>,<endianness>,<offset>]

 Lê dados Modbus.
 Valor de retorno: *objeto JSON*.
 Plataformas suportadas: Linux, Windows.

Parâmetros:

- **endpoint** - o endpoint definido como `protocol://connection_string`;
- **slave id** - o ID do slave;
- **function** - a função Modbus;
- **address** - o endereço do primeiro registro, bobina ou entrada;
- **count** - o número de registros a serem lidos;
- **type** - o tipo de dado;
- **endianness** - a configuração de endianness;
- **offset** - o número de registros, a partir do 'address', cujos resultados serão descartados.

No Windows, este item segue caminhos UNC no parâmetro **endpoint**, o que pode ser um risco de segurança;

Veja uma descrição detalhada dos parâmetros.

net.dns[<ip>,name,<type>,<timeout>,<count>,<protocol>]

 Verifica o status de um serviço DNS.
 Valores de retorno: 0 - resolução de DNS falhou (o servidor DNS não respondeu ou retornou um erro); 1 - resolução de DNS bem-sucedida.
 Veja as plataformas suportadas.

Parâmetros:

- **ip** (ignorado no Windows, a menos que esteja usando o Zabbix agent 2) - o endereço IP do servidor DNS (deixe em branco para o servidor DNS padrão);
- **name** - o nome DNS a ser consultado;
- **type** - o tipo de registro a ser consultado (o padrão é *SOA*);

- **timeout** (ignorado no Windows, a menos que esteja usando o Zabbix agent 2) - o tempo limite para a solicitação em segundos (o padrão é 1 segundo);
- **count** (ignorado no Windows, a menos que esteja usando o Zabbix agent 2) - o número de tentativas para a solicitação (o padrão é 2);
- **protocol** - o protocolo usado para realizar consultas DNS: *udp* (padrão) ou *tcp*.

Comentários:

- Os valores possíveis para `type` são: *ANY*, *A*, *NS*, *CNAME*, *MB*, *MG*, *MR*, *PTR*, *MD*, *MF*, *MX*, *SOA*, *NULL*, *WKS* (não suportado para o Zabbix agent no Windows, Zabbix agent 2 em todos os SO), *HINFO*, *MINFO*, *TXT*, *SRV*
- Para pesquisas de DNS reverso (quando `type` é definido como *PTR*), você pode fornecer o nome DNS tanto no formato invertido quanto no não invertido (veja os exemplos abaixo). Observe que quando o registro PTR é solicitado, o nome DNS é na verdade um endereço IP.
- Nomes de domínio internacionalizados não são suportados, use nomes codificados em IDNA.

Exemplos:

```
net.dns[198.51.100.1,example.com,MX,2,1]

net.dns[,198.51.100.1,PTR]
net.dns[,1.100.51.198.in-addr.arpa,PTR]

net.dns[,2a00:1450:400f:800::200e,PTR]
net.dns[,e.0.0.2.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.0.f.0.0.4.0.5.4.1.0.0.a.2.ip6.arpa,PTR]
```

net.dns.perf[<ip>,name,<type>,<timeout>,<count>,<protocol>]

 Verifica o desempenho de um serviço DNS.
 Valor de retorno: *Float* (0 - serviço está inativo; segundos - o número de segundos gastos aguardando uma resposta do serviço).
 Veja as plataformas suportadas.

Parâmetros:

- **ip** (ignorado no Windows, a menos que use o Zabbix agent 2) - o endereço IP do servidor DNS (deixe em branco para o servidor DNS padrão);
- **name** - o nome DNS a ser consultado;
- **type** - o tipo de registro a ser consultado (o padrão é *SOA*);
- **timeout** (ignorado no Windows, a menos que use o Zabbix agent 2) - o tempo limite para a solicitação em segundos (o padrão é 1 segundo);
- **count** (ignorado no Windows, a menos que use o Zabbix agent 2) - o número de tentativas para a solicitação (o padrão é 2);
- **protocol** - o protocolo usado para executar consultas DNS: *udp* (padrão) ou *tcp*.

Comentários:

- Os valores possíveis para `type` são:
ANY, *A*, *NS*, *CNAME*, *MB*, *MG*, *MR*, *PTR*, *MD*, *MF*, *MX*, *SOA*, *NULL*, *WKS* (não suportado para Zabbix agent no Windows, Zabbix agent 2 em todos os SO), *HINFO*, *MINFO*, *TXT*, *SRV*
- Para pesquisas de DNS reverso (quando `type` é definido como *PTR*), você pode fornecer o nome DNS tanto no formato invertido quanto no não invertido (veja exemplos abaixo). Observe que quando o registro PTR é solicitado, o nome DNS é na verdade um endereço IP.
- Nomes de domínio internacionalizados não são suportados, use nomes codificados em IDNA.
- Desde o Zabbix 7.0.1, o item retorna um tempo de resposta em vez de 0 quando o servidor DNS responde com um código de erro (por exemplo, `NXDOMAIN` ou `SERVFAIL`).

Exemplos:

```
net.dns.perf[198.51.100.1,example.com,MX,2,1]

net.dns.perf[,198.51.100.1,PTR]
net.dns.perf[,1.100.51.198.in-addr.arpa,PTR]

net.dns.perf[,2a00:1450:400f:800::200e,PTR]
net.dns.perf[,e.0.0.2.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.0.f.0.0.4.0.5.4.1.0.0.a.2.ip6.arpa,PTR]
```

net.dns.record[<ip>,name,<type>,<timeout>,<count>,<protocol>]

 Executa uma consulta DNS.

O Zabbix agent 2 também fornece net.dns.get, que oferece recursos adicionais, como mais tipos de registros e maior controle sobre o monitoramento de DNS. Valor de retorno: uma string de caracteres com o tipo de informação solicitado.
 Veja plataformas suportadas.

Parâmetros:

- **ip** (ignorado no Windows, a menos que use o Zabbix agent 2) - o endereço IP do servidor DNS (deixe em branco para o servidor DNS padrão);
- **name** - o nome DNS a ser consultado;
- **type** - o tipo de registro a ser consultado (o padrão é *SOA*);
- **timeout** (ignorado no Windows, a menos que use o Zabbix agent 2) - o tempo limite para a solicitação em segundos (o padrão é 1 segundo);
- **count** (ignorado no Windows, a menos que use o Zabbix agent 2) - o número de tentativas para a solicitação (o padrão é 2);
- **protocol** - o protocolo usado para executar consultas DNS: *udp* (padrão) ou *tcp*.

Comentários:

- Os valores possíveis para `type` são:
ANY, *A*, *NS*, *CNAME*, *MB*, *MG*, *MR*, *PTR*, *MD*, *MF*, *MX*, *SOA*, *NULL*, *WKS* (não suportado para Zabbix agent no Windows, Zabbix agent 2 em todos os SO), *HINFO*, *MINFO*, *TXT*, *SRV*
- Para pesquisas de DNS reverso (quando `type` é definido como *PTR*), você pode fornecer o nome DNS no formato invertido ou não invertido (veja exemplos abaixo). Observe que quando o registro PTR é solicitado, o nome DNS é na verdade um endereço IP.
- Nomes de domínio internacionalizados não são suportados, use nomes codificados em IDNA.

Exemplos:

```
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]

 O número de colisões fora da janela.
 Valor de retorno: *Integer*.
 Plataformas suportadas: Linux, FreeBSD, Solaris, AIX, MacOS X, OpenBSD, NetBSD. Privilégios de root são necessários no NetBSD.

Parâmetro:

- **if** - nome da interface de rede

net.if.discovery

 A lista de interfaces de rede. Usado para descoberta de baixo nível.
 Valor de retorno: *objeto JSON*.
 Plataformas suportadas: Linux, FreeBSD, Solaris, HP-UX, AIX, OpenBSD, NetBSD, Windows.

net.if.in[if,<mode>]

 As estatísticas de tráfego de entrada em uma interface de rede.
 Valor de retorno: *Integer*.
 Plataformas suportadas: Linux, FreeBSD, Solaris[5], HP-UX, AIX, MacOS X, OpenBSD, NetBSD, Windows. Privilégios de root são necessários no NetBSD.

Parâmetros:

- **if** - nome da interface de rede (Unix); descrição completa da interface de rede ou endereço IPv4; ou, se entre chaves, GUID da interface de rede (Windows);
- **mode** - valores possíveis:
bytes - número de bytes (padrão)
packets - número de pacotes
errors - número de erros
dropped - número de pacotes descartados
overruns (fifo) - número de erros de buffer FIFO
frame - número de erros de enquadramento de pacotes
compressed - número de pacotes compactados recebidos pelo driver do dispositivo
multicast - número de quadros multicast recebidos pelo driver do dispositivo

Comentários:

- Você pode usar esta chave com a etapa de pré-processamento *Change per second* para obter as estatísticas de bytes por segundo;
- O modo *dropped* é suportado apenas em Linux, FreeBSD, HP-UX, MacOS X, OpenBSD, NetBSD;
- Os modos *overruns*, *frame*, *compressed*, *multicast* são suportados apenas em Linux;
- No HP-UX, este item não fornece detalhes sobre interfaces de loopback (por exemplo, lo0).

Exemplos:

```
net.if.in[eth0]
net.if.in[eth0,errors]
```

net.if.out[if,<mode>]

 As estatísticas de tráfego de saída em uma interface de rede.
 Valor de retorno: *Integer*.
 Plataformas suportadas: Linux, FreeBSD, Solaris[5], HP-UX, AIX, MacOS X, OpenBSD, NetBSD, Windows. Privilégios de root são necessários no NetBSD.

Parâmetros:

- **if** - nome da interface de rede (Unix); descrição completa da interface de rede ou endereço IPv4; ou, se entre chaves, GUID da interface de rede (Windows);
- **mode** - valores possíveis:
bytes - número de bytes (padrão)
packets - número de pacotes
errors - número de erros
dropped - número de pacotes descartados
overruns (fifo) - número de erros de buffer FIFO
collisions (colls) - número de colisões detectadas na interface
carrier - número de perdas de portadora detectadas pelo driver do dispositivo
compressed - número de pacotes comprimidos transmitidos pelo driver do dispositivo

Comentários:

- Você pode usar esta chave com a etapa de pré-processamento *Change per second* para obter as estatísticas de bytes por segundo;
- O modo *dropped* é suportado apenas no Linux, HP-UX;
- Os modos *overruns*, *collision*, *carrier*, *compressed* são suportados apenas no Linux;
- No HP-UX este item não fornece detalhes sobre interfaces de loopback (por exemplo, lo0).

Exemplos:

```
net.if.out[eth0]
net.if.out[eth0,errors]
```

net.if.total[if,<mode>]

 A soma das estatísticas de tráfego de entrada e saída em uma interface de rede.
 Valor de retorno: *Integer*.
 Plataformas suportadas: Linux, FreeBSD, Solaris[5], HP-UX, AIX, MacOS X, OpenBSD, NetBSD, Windows. Privilégios de root são necessários no NetBSD.

Parâmetros:

- **if** - nome da interface de rede (Unix); descrição completa da interface de rede ou endereço IPv4; ou, se entre chaves, GUID da interface de rede (Windows);
- **mode** - valores possíveis:
bytes - número de bytes (padrão)
packets - número de pacotes
errors - número de erros
dropped - número de pacotes descartados
overruns (fifo) - número de erros de buffer FIFO
collisions (colls) - número de colisões detectadas na interface
compressed - número de pacotes compactados transmitidos ou recebidos pelo driver do dispositivo

Comentários:

- Você pode usar esta chave com a etapa de pré-processamento *Change per second* para obter as estatísticas de bytes por segundo;
- O modo *dropped* é suportado apenas no Linux, HP-UX. Pacotes descartados são suportados apenas se ambos `net.if.in` e `net.if.out` funcionarem para pacotes descartados em sua plataforma.
- Os modos *overruns*, *collision*, *compressed* são suportados apenas no Linux;
- No HP-UX este item não fornece detalhes sobre interfaces de loopback (por exemplo, lo0).

Exemplos:

```
net.if.total[eth0]
net.if.total[eth0,errors]
```

net.tcp.listen[port]

 Verifica se esta porta TCP está no estado LISTEN.
 Valores de retorno: 0 - não está no estado LISTEN; 1 - está no estado LISTEN.
 Plataformas suportadas: Linux, FreeBSD, Solaris, MacOS X, Windows.

Parâmetro:

- **port** - número da porta TCP

Em kernels Linux 2.6.14 e superiores, as informações sobre sockets TCP em escuta são obtidas da interface NETLINK do kernel, se possível. Caso contrário, as informações são recuperadas dos arquivos /proc/net/tcp e /roc/net/tcp6.

Exemplo:

```
net.tcp.listen[80]
```

net.tcp.port[<ip>,port]

 Verifica se é possível fazer uma conexão TCP para a porta especificada.
 Valores de retorno: 0 - não é possível conectar; 1 - é possível conectar.
 Veja as plataformas suportadas.

Parâmetros:

- **ip** - o endereço IP ou nome DNS (o padrão é 127.0.0.1);
- **port** - o número da porta.

Comentários:

- Para testes simples de desempenho TCP, use `net.tcp.service.perf[tcp,<ip>,<port>]`;
- Essas verificações podem resultar em mensagens adicionais nos arquivos de log do sistema (sessões SMTP e SSH geralmente são registradas).

Exemplo:

```
net.tcp.port[,80] #este item pode ser usado para testar a disponibilidade do servidor web em execução na p
```

net.tcp.service[service,<ip>,<port>]

 Verifica se um serviço está em execução e aceitando conexões TCP.
 Valores de retorno: 0 - serviço está inativo; 1 - serviço está em execução.
 Veja as plataformas suportadas.

Parâmetros:

- **service** - *ssh*, *ldap*, *smtp*, *ftp*, *http*, *pop*, *nntp*, *imap*, *tcp*, *https* ou *telnet* (veja detalhes);
- **ip** - o endereço IP ou nome DNS (o padrão é 127.0.0.1);
- **port** - o número da porta (por padrão, o número da porta padrão do serviço é usado).

Comentários:

- Essas verificações podem resultar em mensagens adicionais nos arquivos de log do daemon do sistema (sessões SMTP e SSH geralmente são registradas);
- A verificação de protocolos criptografados (como IMAP na porta 993 ou POP na porta 995) atualmente não é suportada. Como alternativa, use `net.tcp.port[]` para verificações como essas.
- A verificação de LDAP e HTTPS no Windows é suportada apenas pelo Zabbix agent 2;
- A verificação de telnet procura por um prompt de login (':' no final).

Exemplo:

```
net.tcp.service[ftp,,45] #este item pode ser usado para testar a disponibilidade do servidor FTP na porta
```

net.tcp.service.perf[service,<ip>,<port>]

 Verifica o desempenho de um serviço TCP.
 Valores de retorno: *Float* (0 - serviço está inativo; segundos - o número de segundos gastos esperando por uma resposta do serviço).
 Veja as plataformas suportadas.

Parâmetros:

- **service** - *ssh*, *ldap*, *smtp*, *ftp*, *http*, *pop*, *nntp*, *imap*, *tcp*, *https* ou *telnet* (veja detalhes);
- **ip** - o endereço IP ou nome DNS (padrão é 127.0.0.1);
- **port** - o número da porta (por padrão, o número da porta padrão do serviço é usado).

Comentários:

- A verificação de protocolos criptografados (como IMAP na porta 993 ou POP na porta 995) atualmente não é suportada. Como alternativa, use `net.tcp.service.perf[tcp,<ip>,<port>]` para verificações como essas.
- A verificação do telnet procura por um prompt de login (':' no final).

Exemplo:

```
net.tcp.service.perf[ssh] #este item pode ser usado para testar a velocidade da resposta inicial do servid
```

net.tcp.socket.count[<laddr>,<lport>,<raddr>,<rport>,<state>]

 Retorna o número de sockets TCP que correspondem aos parâmetros.
 Valor de retorno: *Integer*.
 Plataformas suportadas: Linux, Windows.

Parâmetros:

- **laddr** - o endereço IPv4/6 local ou sub-rede CIDR;
- **lport** - o número da porta local ou nome do serviço;
- **raddr** - o endereço IPv4/6 remoto ou sub-rede CIDR;
- **rport** - o número da porta remota ou nome do serviço;

- **state** - o estado da conexão (*established*, *syn_sent*, *syn_recv*, *fin_wait1*, *fin_wait2*, *time_wait*, *close*, *close_wait*, *last_ack*, *listen*, *closing*).

Exemplo:

```
net.tcp.socket.count[,80,,,established] #o número de conexões para a porta TCP local 80 no estado establis
```

net.udp.listen[port]

 Verifica se esta porta UDP está no estado LISTEN.
 Valores de retorno: 0 - não está no estado LISTEN; 1 - está no estado LISTEN.
 Plataformas suportadas: Linux, FreeBSD, Solaris, MacOS X.

Parâmetro:

- **port** - número da porta UDP

Exemplo:

```
net.udp.listen[68]
```

net.udp.service[service,<ip>,<port>]

 Verifica se um serviço está em execução e respondendo a solicitações UDP.
 Valores de retorno: 0 - serviço está inativo; 1 - serviço está em execução.
 Veja as plataformas suportadas.

Parâmetros:

- **service** - *ntp* (veja detalhes);
- **ip** - o endereço IP ou nome DNS (padrão é 127.0.0.1);
- **port** - o número da porta (por padrão, o número da porta padrão do serviço é usado).

Exemplo:

```
net.udp.service[ntp,,45] #este item pode ser usado para testar a disponibilidade do serviço NTP na porta U
```

net.udp.service.perf[service,<ip>,<port>]

 Verifica o desempenho de um serviço UDP.
 Valores de retorno: *Float* (0 - serviço está inativo; segundos - o número de segundos gastos aguardando uma resposta do serviço).
 Veja as plataformas suportadas.

Parâmetros:

- **service** - *ntp* (veja detalhes);
- **ip** - o endereço IP ou nome DNS (padrão é 127.0.0.1);
- **port** - o número da porta (por padrão, o número da porta padrão do serviço é usado).

Exemplo:

```
net.udp.service.perf[ntp] #este item pode ser usado para testar o tempo de resposta do serviço NTP
```

net.udp.socket.count[<laddr>,<lport>,<raddr>,<rport>,<state>]

 Retorna o número de sockets UDP que correspondem aos parâmetros.
 Valor de retorno: *Integer*.
 Plataformas suportadas: Linux, Windows.

Parâmetros:

- **laddr** - o endereço IPv4/6 local ou sub-rede CIDR;
- **lport** - o número da porta local ou nome do serviço;
- **raddr** - o endereço IPv4/6 remoto ou sub-rede CIDR;
- **rport** - o número da porta remota ou nome do serviço;
- **state** - o estado da conexão (*established*, *unconn*).

Exemplo:

```
net.udp.socket.count[,,,,established] #retorna o número de sockets UDP no estado conectado
```

proc.cpu.util[<name>,<user>,<type>,<cmdline>,<mode>,<zone>]

 A porcentagem de utilização da CPU pelo processo.
 Valor de retorno: *Float*.
 Plataformas suportadas: Linux, Solaris[6].

Parâmetros:

- **name** - o nome do processo (o padrão é *todos os processos*);
- **user** - o nome do usuário (o padrão é *todos os usuários*);
- **type** - o tipo de utilização da CPU: *total* (padrão), *user* ou *system*;
- **cmdline** - filtro pela linha de comando (é uma expressão regular);

- **mode** - o modo de coleta de dados: *avg1* (padrão), *avg5* ou *avg15*;
- **zone** - a zona de destino: *current* (padrão) ou *all*. Este parâmetro é suportado apenas no Solaris.

Comentários:

- O valor retornado é baseado na porcentagem de utilização de um único núcleo de CPU. Por exemplo, a utilização da CPU de um processo usando totalmente dois núcleos é 200%.
- Os dados de utilização da CPU do processo são coletados por um coletor que suporta no máximo 1024 consultas únicas (por nome, usuário e linha de comando). Consultas não acessadas nas últimas 24 horas são removidas do coletor.
- Ao definir o parâmetro `zone` como *current* (ou padrão) no caso de o agent ter sido compilado em um Solaris sem suporte a zonas, mas executado em um Solaris mais recente onde as zonas são suportadas, o agent retornará NOTSUPPORTED (o agent não pode limitar os resultados apenas à zona atual). No entanto, *all* é suportado neste caso.

Exemplos:

```
proc.cpu.util[,root] #Utilização da CPU de todos os processos executados pelo usuário "root"
proc.cpu.util[zabbix_server,zabbix] #Utilização da CPU de todos os processos zabbix_server executados pelo
```

proc.get[<name>,<user>,<cmdline>,<mode>]

 A lista de processos do SO e seus parâmetros. Pode ser usado para descoberta de baixo nível.
 Valor de retorno: *objeto JSON*.
 Plataformas suportadas: Linux, FreeBSD, Windows, OpenBSD, NetBSD.

Parâmetros:

- **name** - o nome do processo (padrão *todos os processos*);
- **user** - o nome do usuário (padrão *todos os usuários*);
- **cmdline** - filtro pela linha de comando (é uma expressão regular). Este parâmetro não é suportado para Windows; em outras plataformas não é suportado se o modo estiver definido como 'summary'.
- **mode** - valores possíveis:
process (padrão), *thread* (não suportado para NetBSD), *summary*. Veja uma lista de parâmetros de processo retornados para cada modo e SO.

Comentários:

- Se um valor não puder ser recuperado, por exemplo, devido a um erro (processo já morreu, falta de permissões, falha na chamada do sistema), −1 será retornado;
- Veja notas sobre a seleção de processos com os parâmetros `name` e `cmdline` (específico do Linux).

Exemplos:

```
proc.get[zabbix_server,zabbix,,process] #lista de todos os processos zabbix_server executados pelo usuário
proc.get[java,,,thread] #lista de todos os processos Java, retorna uma entrada por thread
proc.get[,zabbix,,summary] #dados combinados para processos de cada tipo executados pelo usuário zabbix, r
```

proc.mem[<name>,<user>,<mode>,<cmdline>,<memtype>]

 A memória usada pelo processo em bytes.
 Valor de retorno: *Integer* - com `mode` como *max*, *min*, *sum*; *Float* - com `mode` como *avg*
 Plataformas suportadas: Linux, FreeBSD, Solaris, AIX, OpenBSD, NetBSD.

Parâmetros:

- **name** - o nome do processo (o padrão é *todos os processos*);
- **user** - o nome do usuário (o padrão é *todos os usuários*);
- **mode** - valores possíveis: *avg*, *max*, *min* ou *sum* (padrão);
- **cmdline** - filtra pela linha de comando (é uma expressão regular);
- **memtype** - o tipo de memória usada pelo processo

Comentários:

- O parâmetro `memtype` é suportado apenas em Linux, FreeBSD, Solaris[6], AIX;
- Quando vários processos usam memória compartilhada, a soma da memória usada pelos processos pode resultar em valores grandes e irreais.

Veja notas sobre seleção de processos com os parâmetros `name` e `cmdline` (específico do Linux).

Quando este item é invocado a partir da linha de comando e contém um parâmetro de linha de comando (por exemplo, usando o modo de teste do agent: `zabbix_agentd -t proc.mem[,,,apache2]`), um processo extra será contado, pois o agent contará a si mesmo.

Exemplos:

```
proc.mem[,root] #a memória usada por todos os processos executados pelo usuário "root"
proc.mem[zabbix_server,zabbix] #a memória usada por todos os processos zabbix_server executados pelo usuár
proc.mem[,oracle,max,oracleZABBIX] #a memória usada pelo processo que mais consome memória executado pelo
```

proc.num[<name>,<user>,<state>,<cmdline>,<zone>]

 O número de processos.
 Valor de retorno: *Integer*.
 Plataformas suportadas: Linux, FreeBSD, Solaris[6], HP-UX, AIX, OpenBSD, NetBSD, Windows.

Parâmetros:

- **name** - o nome do processo (padrão é *todos os processos*);
- **user** - o nome do usuário (padrão é *todos os usuários*);
- **state** - valores possíveis:
all (padrão),
disk - sono ininterrupto,
run - em execução,
sleep - sono interrompível,
trace - parado,
zomb - zumbi;
- **cmdline** - filtro pela linha de comando (é uma expressão regular);
- **zone** - a zona de destino: *current* (padrão) ou *all*. Este parâmetro é suportado apenas no Solaris.

Comentários:

- Os parâmetros de estado *disk* e *trace* são suportados apenas em Linux, FreeBSD, OpenBSD, NetBSD;
- No Windows, apenas os parâmetros `name` e `user` são suportados;
- Quando este item é invocado a partir da linha de comando e contém um parâmetro de linha de comando (por exemplo, usando o modo de teste do agent: `zabbix_agentd -t proc.num[,,,apache2]`), um processo extra será contado, pois o agent contará a si mesmo;
- Ao definir o parâmetro `zone` como *current* (ou padrão) no caso de o agent ter sido compilado em um Solaris sem suporte a zonas, mas executando em um Solaris mais recente onde as zonas são suportadas, então o agent retornará NOTSUPPORTED (o agent não pode limitar os resultados apenas à zona atual). No entanto, *all* é suportado neste caso.
- Veja notas sobre seleção de processos com os parâmetros `name` e `cmdline` (específico do Linux).

Exemplos:

```
proc.num[,mysql] #o número de processos em execução sob o usuário mysql
proc.num[apache2,www-data] #o número de processos apache2 em execução sob o usuário www-data
proc.num[,oracle,sleep,oracleZABBIX] #o número de processos em estado de sono em execução sob o Oracle com
```

sensor[device,sensor,<mode>]

 Leitura de sensor de hardware.
 Valor de retorno: *Float*.
 Plataformas suportadas: Linux, OpenBSD.

Parâmetros:

- **device** - o nome do dispositivo;
- **sensor** - o nome do sensor;
- **mode** - valores possíveis: *avg*, *max* ou *min*.

Comentários:

- Lê /proc/sys/dev/sensors no Linux 2.4;
- Lê /sys/class/hwmon no Linux 2.6+. Veja uma descrição mais detalhada do item sensor no Linux;
- Lê o MIB *hw.sensors* no OpenBSD;
- No Linux 2.4, se você especificar o modo, então device e sensor são tratados como expressões regulares;
- No Linux 2.6+ e OpenBSD, ou se o modo for omitido, device e sensor são tratados literalmente.

Exemplos:

```
sensor[w83781d-i2c-0-2d,temp1]
sensor[cpu0,temp0] #a temperatura de uma CPU
sensor["cpu[0-2]$",temp,avg] #a temperatura média das três primeiras CPUs
```

system.boottime

 O tempo de inicialização do sistema.
 Valor de retorno: *Inteiro (timestamp Unix)*.
 Plataformas suportadas: Linux, FreeBSD, Solaris, MacOS X, OpenBSD, NetBSD.

system.cpu.discovery

 A lista de CPUs/núcleos de CPU detectados. Usado para descoberta de baixo nível.
 Valor de retorno: *objeto JSON*.
 Veja plataformas suportadas.

system.cpu.intr

 As interrupções do dispositivo.
 Valor de retorno: *Inteiro*.
 Plataformas suportadas: Linux, FreeBSD, Solaris, AIX, OpenBSD, NetBSD.

system.cpu.load[<cpu>,<mode>]

 A carga da CPU.
 Valor de retorno: *Float*.
 Veja as plataformas suportadas.

Parâmetros:

- **cpu** - valores possíveis: *all* (padrão) ou *percpu* (a carga total dividida pelo número de CPUs online);
- **mode** - valores possíveis: *avg1* (média de um minuto, padrão), *avg5* ou *avg15*.

Exemplo:

```
system.cpu.load[,avg5]
```

system.cpu.num[<type>]

 O número de CPUs.
 Valor de retorno: *Integer*.
 Plataformas suportadas: Linux, FreeBSD, Solaris, HP-UX, AIX, MacOS X, OpenBSD, NetBSD, Windows.

Parâmetro:

- **type** - valores possíveis: *online* (padrão) ou *max*

O parâmetro de tipo *max* é suportado apenas em Linux, FreeBSD, Solaris, MacOS X.

Exemplo:

```
system.cpu.num
```

system.cpu.switches

 A contagem de trocas de contexto.
 Valor de retorno: *Inteiro*.
 Plataformas suportadas: Linux, FreeBSD, Solaris, AIX, OpenBSD, NetBSD.

system.cpu.util[<cpu>,<type>,<mode>,<logical or physical>]

 A porcentagem de utilização da CPU.
 Valor de retorno: *Float*.
 Plataformas suportadas: Linux, FreeBSD, Solaris, HP-UX, AIX, OpenBSD, NetBSD, Windows.

Parâmetros:

- **cpu** - *<número da CPU>* ou *all* (padrão);
- **type** - valores possíveis: *user* (padrão), *idle*, *nice*, *system*, *iowait*, *interrupt*, *softirq*, *steal*, *guest* (em kernels Linux 2.6.24 e superiores), ou *guest_nice* (em kernels Linux 2.6.33 e superiores);
- **mode** - valores possíveis: *avg1* (média de um minuto, padrão), *avg5* ou *avg15*;
- **logical or physical** - valores possíveis: *logical* (padrão) ou *physical*. Este parâmetro é suportado apenas no AIX.

Comentários:

- O parâmetro de tipo *nice* é suportado apenas em Linux, FreeBSD, HP-UX, OpenBSD, NetBSD.
- O parâmetro de tipo *iowait* é suportado apenas em Linux 2.6 e superior, Solaris, AIX.
- O parâmetro de tipo *interrupt* é suportado apenas em Linux 2.6 e superior, FreeBSD, OpenBSD.
- Os parâmetros de tipo *softirq*, *steal*, *guest*, *guest_nice* são suportados apenas em Linux 2.6 e superior.
- Os parâmetros de modo *avg5* e *avg15* são suportados em Linux, FreeBSD, Solaris, HP-UX, AIX, OpenBSD, NetBSD.

Exemplo:

```
system.cpu.util[0,user,avg5]
```

system.hostname[<type>,<transform>]

 O nome do host do sistema.
 Valor de retorno: *String*.
 Veja plataformas suportadas.

Parâmetros:

- **type** - valores possíveis: *netbios* (padrão no Windows), *host* (padrão no Linux), *shorthost* (retorna parte do nome do host antes do primeiro ponto, uma string completa para nomes sem pontos), *fqdn* (retorna o Fully Qualified Domain Name);
- **transform** - valores possíveis: *none* (padrão) ou *lower* (converte para minúsculas).

O valor é obtido pegando o `nodename` da saída da API do sistema uname().

Exemplos de valores retornados:

```
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>]

 Informação do chassi.
 Valor de retorno: *String*.
 Plataformas suportadas: Linux.

Parâmetro:

- **info** - valores possíveis: *full* (padrão), *model*, *serial*, *type* ou *vendor*

Comentários:

- Esta chave de item depende da disponibilidade da tabela SMBIOS;
- Tentará ler a tabela DMI do sysfs, se o acesso ao sysfs falhar, tentará ler diretamente da memória;
- **Permissões de root** são necessárias porque o valor é obtido lendo do sysfs ou da memória.

Exemplo:

```
system.hw.chassis[full] → Hewlett-Packard HP Pro 3010 Small Form Factor PC CZXXXXXXXX Desktop
```

system.hw.cpu[<cpu>,<info>]

 Informações da CPU.
 Valor de retorno: *String* ou *Integer*.
 Plataformas suportadas: Linux.

Parâmetros:

- **cpu** - *<número da CPU>* ou *all* (padrão);
- **info** - valores possíveis: *full* (padrão), *curfreq*, *maxfreq*, *model* ou *vendor*.

Comentários:

- Coleta informações de `/proc/cpuinfo` e `/sys/devices/system/cpu/[cpunum]/cpufreq/cpuinfo_max_freq`;
- Se um número de CPU e *curfreq* ou *maxfreq* forem especificados, um valor numérico é retornado (Hz).

Exemplo:

```
system.hw.cpu[0,vendor] → AuthenticAMD
```

system.hw.devices[<type>]

 A listagem de dispositivos PCI ou USB.
 Valor de retorno: *Texto*.
 Plataformas suportadas: Linux.

Parâmetro:

- **type** - *pci* (padrão) ou *usb*

Retorna a saída do utilitário lspci ou lsusb (executado sem quaisquer parâmetros).

Exemplo:

```
system.hw.devices → 00:00.0 Host bridge: Advanced Micro Devices [AMD] RS780 Host Bridge
```

system.hw.macaddr[<interface>,<format>]

 A listagem dos endereços MAC.
 Valor de retorno: *String*.
 Plataformas suportadas: Linux.

Parâmetros:

- **interface** - *all* (padrão) ou uma expressão regular;
- **format** - *full* (padrão) ou *short*

Comentários:

- Lista os endereços MAC das interfaces cujo nome corresponde à expressão regular `interface` fornecida (*all* lista para todas as interfaces);
- Se `format` for especificado como *short*, nomes de interface e endereços MAC idênticos não são listados.

Exemplo:

```
system.hw.macaddr["eth0$",full] → [eth0] 00:11:22:33:44:55
```

system.localtime[<type>]

 A hora do sistema.
 Valor de retorno: *Integer* - com `type` como *utc*; *String* - com `type` como *local*.
 Veja as plataformas suportadas.

Parâmetros:

- **type** - valores possíveis: *utc* - (padrão) o tempo desde a Época (00:00:00 UTC, 1º de janeiro de 1970), medido em segundos ou *local* - o tempo no formato 'yyyy-mm-dd,hh:mm:ss.nnn,+hh:mm'

Deve ser usado como uma verificação passiva para o agent Zabbix; no agent Zabbix 2 pode ser configurado como uma verificação ativa.

Exemplo:

```
system.localtime[local] #crie um item usando esta chave e depois use-o para exibir a hora do host no widge
```

system.run[command,<mode>]

 Executa o comando especificado no host.
 Valor de retorno: resultado *Texto* do comando ou 1 - com `mode` como *nowait* (independentemente do resultado do comando).
 Veja as plataformas suportadas.

Parâmetros:

- **command** - comando para execução;
- **mode** - valores possíveis: *wait* - aguarda o fim da execução (padrão) ou *nowait* - não aguarda.

Comentários:

- Este item está desabilitado por padrão. Saiba como habilitá-los;
- O valor de retorno do item é a saída padrão junto com o erro padrão produzido pelo comando. A verificação do código de saída não é realizada;
- Para ser processado corretamente, o valor de retorno do comando deve ser do tipo de dado `text`. Um resultado vazio também é permitido;
- O valor de retorno é limitado a 16MB (incluindo espaços em branco à direita que são truncados); limites do banco de dados também se aplicam;
- Veja também: Execução de comandos.

Exemplo:

```
system.run[ls -l /] #retorna uma lista detalhada de arquivos do diretório raiz
```

system.stat[resource,<type>]

 As estatísticas do sistema.
 Valor de retorno: *Integer* ou *float*.
 Plataformas suportadas: AIX.

Parâmetros:

- **ent** - o número de unidades de processador que esta partição tem direito de receber (float);
- **kthr,<type>** - informações sobre os estados das threads do kernel:
r - número médio de threads do kernel executáveis (float)
b - número médio de threads do kernel colocadas na fila de espera do Gerenciador de Memória Virtual (float)
- **memory,<type>** - informações sobre o uso de memória virtual e real:
avm - páginas virtuais ativas (integer)
fre - tamanho da lista livre (integer)
- **page,<type>** - informações sobre falhas de página e atividade de paginação:
fi - page-ins de arquivo por segundo (float)
fo - page-outs de arquivo por segundo (float)
pi - páginas paginadas do espaço de paginação (float)
po - páginas paginadas para o espaço de paginação (float)
fr - páginas liberadas (substituição de página) (float)
sr - páginas examinadas pelo algoritmo de substituição de página (float)
- **faults,<type>** - taxa de trap e interrupção:
in - interrupções de dispositivo (float)
sy - chamadas de sistema (float)
cs - trocas de contexto de threads do kernel (float)
- **cpu,<type>** - detalhamento da porcentagem de uso do tempo do processador:
us - tempo de usuário (float)
sy - tempo de sistema (float)
id - tempo ocioso (float)
wa - tempo ocioso durante o qual o sistema tinha solicitações pendentes de E/S de disco/NFS (float)
pc - número de processadores físicos consumidos (float)
ec - porcentagem da capacidade de direito consumida (float)
lbusy - indica a porcentagem de utilização do(s) processador(es) lógico(s) que ocorreu durante a execução no nível de usuário e sistema (float)
app - indica os processadores físicos disponíveis no pool compartilhado (float)
- **disk,<type>** - estatísticas de disco:
bps - indica a quantidade de dados transferidos (lidos ou gravados) para a unidade em bytes por segundo (integer)
tps - indica o número de transferências por segundo que foram emitidas para o disco/tape físico (float)

Comentários:

- Observe as seguintes limitações nesses itens:
 `system.stat[cpu,app]` - suportado apenas em AIX LPAR do tipo ”Shared”
 `system.stat[cpu,ec]` - suportado em AIX LPAR do tipo ”Shared” e ”Dedicated” (”Dedicated” sempre retorna 100 (por cento))
 `system.stat[cpu,lbusy]` - suportado apenas em AIX LPAR do tipo ”Shared”
 `system.stat[cpu,pc]` - suportado em AIX LPAR do tipo ”Shared” e ”Dedicated”
 `system.stat[ent]` - suportado em AIX LPAR do tipo ”Shared” e ”Dedicated”

system.sw.arch

 A informação da arquitetura do software.
 Valor de retorno: *String*.
 Veja plataformas suportadas.

A informação é obtida da função `uname()`.

Exemplo:

```
system.sw.arch → i686
```

system.sw.os[<info>]

 Informação do sistema operacional.
 Valor de retorno: *String*.
 Plataformas suportadas: Linux, Windows.

Parâmetro:

- **info** - valores possíveis: *full* (padrão), *short* ou *name*

A informação é adquirida de (note que nem todos os arquivos e opções estão presentes em todas as distribuições):

- `/proc/version` (*full*) no Linux;
- `/proc/version_signature` (*short*) no Linux;
- o parâmetro PRETTY_NAME de `/etc/os-release` em sistemas Linux que o suportam ou `/etc/issue.net` (*name*);
- a chave de registro `HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion` no Windows.

Exemplos:

```
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

 Informações detalhadas sobre o sistema operacional (versão, tipo, nome da distribuição, versão secundária e principal, etc).
 Valor de retorno: *objeto JSON*.
 Plataformas suportadas: Linux, Windows.

system.sw.packages[<regexp>,<manager>,<format>]

 A listagem dos pacotes instalados.
 Valor de retorno: *Texto*.
 Plataformas suportadas: Linux.

Parâmetros:

- **regexp** - *all* (padrão) ou uma expressão regular;
- **manager** - *all* (padrão) ou um gerenciador de pacotes;
- **format** - *full* (padrão) ou *short*.

Comentários:

- Lista (em ordem alfabética) os pacotes instalados cujo nome corresponde à expressão regular fornecida (*all* lista todos);
- Gerenciadores de pacotes suportados (comando executado):
dpkg (dpkg --get-selections)
pkgtool (ls /var/log/packages)
r (rpm -qa)
pacman (pacman -Q)
portage
- Se `format` for especificado como *full*, os pacotes são agrupados por gerenciadores de pacotes (cada gerenciador em uma linha separada começando com seu nome entre colchetes);
- Se `format` for especificado como *short*, os pacotes não são agrupados e são listados em uma única linha.

Exemplo:

```
system.sw.packages[mini,dpkg,short] → python-minimal, python2.6-minimal, ubuntu-minimal
```

system.sw.packages.get[<regexp>,<manager>]

 Uma listagem detalhada dos pacotes instalados.
 Valor de retorno: *objeto JSON*.
 Plataformas suportadas: Linux.

Parâmetros:

- **regexp** - *all* (padrão) ou uma expressão regular;
- **manager** - *all* (padrão) ou um gerenciador de pacotes (valores possíveis: *rpm*, *dpkg*, *pkgtool*, *pacman* ou *portage*).

Comentários:

- Retorna um JSON não formatado com os pacotes instalados cujo nome corresponde à expressão regular fornecida;
- A saída é um array de objetos, cada um contendo as seguintes chaves: name, manager, version, size, architecture, buildtime e installtime (veja mais detalhes).

system.swap.in[<device>,<type>]

 As estatísticas de swap-in (do dispositivo para a memória).
 Valor de retorno: *Integer*.
 Plataformas suportadas: Linux, FreeBSD, OpenBSD.

Parâmetros:

- **device** - especifica o dispositivo usado para swap (apenas Linux) ou *all* (padrão);
- **type** - valores possíveis: *count* (número de swapins, padrão em plataformas que não sejam Linux), *sectors* (setores trocados), ou *pages* (páginas trocadas, padrão no Linux).

Comentários:

- A fonte desta informação é:
/proc/swaps, /proc/partitions, /proc/stat (Linux 2.4)
/proc/swaps, /proc/diskstats, /proc/vmstat (Linux 2.6)
- Observe que *pages* só funcionará se o dispositivo não for especificado;
- O parâmetro de tipo *sectors* é suportado apenas no Linux.

Exemplo:

```
system.swap.in[,pages]
```

system.swap.out[<device>,<type>]

 As estatísticas de swap-out (da memória para o dispositivo).
 Valor de retorno: *Integer*.
 Plataformas suportadas: Linux, FreeBSD, OpenBSD.

Parâmetros:

- **device** - especifica o dispositivo usado para swap (apenas Linux) ou *all* (padrão);
- **type** - valores possíveis: *count* (número de swapouts, padrão em plataformas que não sejam Linux), *sectors* (setores trocados), ou *pages* (páginas trocadas, padrão no Linux).

Comentários:

- A fonte desta informação é:
`/proc/swaps`, `/proc/partitions`, `/proc/stat` (Linux 2.4)
`/proc/swaps`, `/proc/diskstats`, `/proc/vmstat` (Linux 2.6)
- Observe que *pages* só funcionará se o dispositivo não for especificado;
- O parâmetro de tipo *sectors* é suportado apenas no Linux.

Exemplo:

```
system.swap.out[,pages]
```

system.swap.size[<device>,<type>]

 O tamanho do espaço de swap em bytes ou em porcentagem do total.
 Valor de retorno: *Integer* - para bytes; *Float* - para porcentagem.
 Plataformas suportadas: Linux, FreeBSD, Solaris, AIX, OpenBSD, Windows.

Parâmetros:

- **device** - especifica o dispositivo usado para swap (apenas FreeBSD) ou *all* (padrão);
- **type** - valores possíveis: *free* (espaço de swap livre, padrão), *pfree* (espaço de swap livre, em porcentagem), *pused* (espaço de swap usado, em porcentagem), *total* (espaço de swap total) ou *used* (espaço de swap usado).

Comentários:

- Note que *pfree*, *pused* não são suportados no Windows se o tamanho do swap for 0;
- Se o dispositivo não for especificado, o agent Zabbix levará em consideração apenas dispositivos de swap (arquivos), a memória física será ignorada. Por exemplo, em sistemas Solaris, o comando `swap -s` inclui uma parte da memória física e dispositivos de swap (diferente do `swap -l`).

Exemplo:

```
system.swap.size[,pfree] → porcentagem de espaço de swap livre
```

system.uname

 Identificação do sistema.
 Valor de retorno: *String*.
 Veja plataformas suportadas.

Comentários:

- No UNIX, o valor para este item é obtido com a chamada de sistema uname();
- No Windows, o item retorna a arquitetura do SO, enquanto no UNIX retorna a arquitetura da CPU.

Exemplos:

```
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

 O tempo de atividade do sistema em segundos.
 Valor de retorno: *Integer*.
 Plataformas suportadas: Linux, FreeBSD, Solaris, AIX, MacOS X, OpenBSD, NetBSD, Windows.

Na configuração do item, use as unidades **s** ou **uptime** para obter valores legíveis.

system.users.num

 O número de usuários conectados.
 Valor de retorno: *Integer*.
 Veja as plataformas suportadas.

O comando **who** é usado no lado do agent para obter o valor.

vfs.dev.discovery

 A lista de dispositivos de bloco e seus tipos. Usado para descoberta de baixo nível.
 Valor de retorno: *objeto JSON*.
 Plataformas suportadas: Linux.

vfs.dev.read[<device>,<type>,<mode>]

 As estatísticas de leitura do disco.
 Valor de retorno: *Integer* - com `type` em *sectors*, *operations*, *bytes*; *Float* - com `type` em *sps*, *ops*, *bps*.
 Plataformas suportadas: Linux, FreeBSD, Solaris, AIX, OpenBSD.

Parâmetros:

- **device** - dispositivo de disco (o padrão é *all* [3]);
- **type** - valores possíveis: *sectors*, *operations*, *bytes*, *sps*, *ops* ou *bps* (*sps*, *ops*, *bps* significam: setores, operações, bytes por segundo, respectivamente);
- **mode** - valores possíveis: *avg1* (média de um minuto, padrão), *avg5* ou *avg15*. Este parâmetro é suportado apenas com `type` em: sps, ops, bps.

Comentários:

- Se usar um intervalo de atualização de três horas ou mais[2], este item sempre retornará '0';
- Os parâmetros de tipo *sectors* e *sps* são suportados apenas no Linux;
- O parâmetro de tipo *ops* é suportado apenas no Linux e FreeBSD;
- O parâmetro de tipo *bps* é suportado apenas no FreeBSD;
- O parâmetro de tipo *bytes* é suportado apenas no FreeBSD, Solaris, AIX, OpenBSD;
- O parâmetro `mode` é suportado apenas no Linux, FreeBSD;
- Você pode usar nomes de dispositivos relativos (por exemplo, `sda`) bem como um prefixo /dev/ opcional (por exemplo, `/dev/sda`);
- Volumes lógicos LVM são suportados;
- Os valores padrão do parâmetro 'type' para diferentes sistemas operacionais:
AIX - operations
FreeBSD - bps
Linux - sps
OpenBSD - operations
Solaris - bytes
- *sps*, *ops* e *bps* nas plataformas suportadas são limitados a 1024 dispositivos (1023 individuais e um para *all*).

Exemplo:

```
vfs.dev.read[,operations]
```

vfs.dev.write[<device>,<type>,<mode>]

 As estatísticas de escrita em disco.
 Valor de retorno: *Integer* - com `type` em *sectors*, *operations*, *bytes*; *Float* - com `type` em *sps*, *ops*, *bps*.
 Plataformas suportadas: Linux, FreeBSD, Solaris, AIX, OpenBSD.

Parâmetros:

- **device** - dispositivo de disco (o padrão é *all* [3]);
- **type** - valores possíveis: *sectors*, *operations*, *bytes*, *sps*, *ops* ou *bps* (*sps*, *ops*, *bps* significam: setores, operações, bytes por segundo, respectivamente);
- **mode** - valores possíveis: *avg1* (média de um minuto, padrão), *avg5* ou *avg15*. Este parâmetro é suportado apenas com `type` em: sps, ops, bps.

Comentários:

- Se usar um intervalo de atualização de três horas ou mais[2], este item sempre retornará '0';
- Os parâmetros de tipo *sectors* e *sps* são suportados apenas no Linux;
- O parâmetro de tipo *ops* é suportado apenas no Linux e FreeBSD;
- O parâmetro de tipo *bps* é suportado apenas no FreeBSD;
- O parâmetro de tipo *bytes* é suportado apenas no FreeBSD, Solaris, AIX, OpenBSD;
- O parâmetro `mode` é suportado apenas no Linux, FreeBSD;
- Você pode usar nomes de dispositivos relativos (por exemplo, `sda`) bem como um prefixo /dev/ opcional (por exemplo, `/dev/sda`);
- Volumes lógicos LVM são suportados;
- Os valores padrão do parâmetro 'type' para diferentes sistemas operacionais:
AIX - operations
FreeBSD - bps
Linux - sps
OpenBSD - operations
Solaris - bytes
- *sps*, *ops* e *bps* nas plataformas suportadas são limitados a 1024 dispositivos (1023 individuais e um para *all*).

Exemplo:

```
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>]

 A contagem de entradas do diretório.
 Valor de retorno: *Integer*.
 Veja as plataformas suportadas.

Parâmetros:

- **dir** - o caminho absoluto para o diretório;
- **regex incl** - uma expressão regular descrevendo o padrão de nome da entidade (arquivo, diretório, link simbólico) a ser incluída; inclui todas se vazio (valor padrão);
- **regex excl** - uma expressão regular descrevendo o padrão de nome da entidade (arquivo, diretório, link simbólico) a ser excluída; não exclui nenhuma se vazio (valor padrão);
- **types incl** - tipos de entradas de diretório a serem contadas, valores possíveis: *file* - arquivo regular, *dir* - subdiretório, *sym* - link simbólico, *sock* - socket, *bdev* - dispositivo de bloco, *cdev* - dispositivo de caractere, *fifo* - FIFO, *dev* - sinônimo de "bdev,cdev", *all* - todos os tipos (padrão), ou seja, "file,dir,sym,sock,bdev,cdev,fifo". Vários tipos devem ser separados por vírgula e entre aspas.
- **types excl** - tipos de entradas de diretório (veja `types incl`) a NÃO serem contadas. Se algum tipo de entrada estiver em ambos `types incl` e `types excl`, as entradas desse tipo NÃO serão contadas.
- **max depth** - a profundidade máxima de subdiretórios a serem percorridos:
-1 (padrão) - ilimitado,
0 - não desce em subdiretórios.
- **min size** - o tamanho mínimo (em bytes) para o arquivo ser contado. Arquivos menores não serão contados. Sufixos de memória podem ser usados.
- **max size** - o tamanho máximo (em bytes) para o arquivo ser contado. Arquivos maiores não serão contados. Sufixos de memória podem ser usados.
- **min age** - a idade mínima (em segundos) da entrada do diretório a ser contada. Entradas mais recentes não serão contadas. Sufixos de tempo podem ser usados.
- **max age** - a idade máxima (em segundos) da entrada do diretório a ser contada. Entradas tão antigas e mais antigas não serão contadas (tempo de modificação). Sufixos de tempo podem ser usados.
- **regex excl dir** - uma expressão regular descrevendo o padrão de nome do diretório a ser excluído. Todo o conteúdo do diretório será excluído (em contraste com regex_excl)

Comentários:

- No Windows, este item segue caminhos UNC, o que pode ser um risco de segurança;
- Variáveis de ambiente, por exemplo, %APP_HOME%, $HOME e %TEMP% não são suportadas;
- Pseudo-diretórios "." e ".." nunca são contados;
- Links simbólicos nunca são seguidos para percorrer diretórios;
- Tanto `regex incl` quanto `regex excl` são aplicados a arquivos e diretórios ao calcular a contagem de entradas, mas são ignorados ao escolher subdiretórios para percorrer (se `regex incl` for "(?i)^.+\.zip$" e `max depth` não estiver definido, então todos os subdiretórios serão percorridos, mas apenas os arquivos do tipo zip serão contados).
- O tempo de execução é limitado pelo valor de timeout na configuração do agent (3 seg). Como a varredura de diretórios grandes pode levar mais tempo do que isso, nenhum dado será retornado e o item ficará como não suportado. A contagem parcial não será retornada.
- Ao filtrar por tamanho, apenas arquivos regulares têm tamanhos significativos. No Linux e BSD, diretórios também têm tamanhos diferentes de zero (normalmente alguns Kb). Dispositivos têm tamanho zero, por exemplo, o tamanho de **/dev/sda1** não reflete o tamanho da partição respectiva. Portanto, ao usar <`min_size`> e <`max_size`>, é aconselhável especificar <`types_incl`> como "*file*", para evitar surpresas.

Exemplos:

```
vfs.dir.count[/dev] #monitora o número de dispositivos em /dev (Linux)
vfs.dir.count["C:\Users\ADMINI~1\AppData\Local\Temp"] #monitora o número de arquivos em um diretório tempo
```

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>]

 A lista de entradas do diretório.
 Valor de retorno: *objeto JSON*.
 Veja plataformas suportadas.

Parâmetros:

- **dir** - o caminho absoluto para o diretório;
- **regex incl** - uma expressão regular descrevendo o padrão de nome da entidade (arquivo, diretório, link simbólico) a ser incluída; inclui todas se vazio (valor padrão);
- **regex excl** - uma expressão regular descrevendo o padrão de nome da entidade (arquivo, diretório, link simbólico) a ser excluída; não exclui nenhuma se vazio (valor padrão);

- **types incl** - tipos de entradas de diretório a serem listadas, valores possíveis: *file* - arquivo regular, *dir* - subdiretório, *sym* - link simbólico, *sock* - socket, *bdev* - dispositivo de bloco, *cdev* - dispositivo de caractere, *fifo* - FIFO, *dev* - sinônimo de "bdev,cdev", *all* - todos os tipos (padrão), ou seja, "file,dir,sym,sock,bdev,cdev,fifo". Vários tipos devem ser separados por vírgula e entre aspas.
- **types excl** - tipos de entradas de diretório (veja `types incl`) que NÃO devem ser listadas. Se algum tipo de entrada estiver em ambos `types incl` e `types excl`, entradas desse tipo NÃO serão listadas.
- **max depth** - a profundidade máxima de subdiretórios a serem percorridos:
-1 (padrão) - ilimitado,
0 - não desce em subdiretórios.
- **min size** - o tamanho mínimo (em bytes) para o arquivo ser listado. Arquivos menores não serão listados. Sufixos de memória podem ser usados.
- **max size** - o tamanho máximo (em bytes) para o arquivo ser listado. Arquivos maiores não serão listados. Sufixos de memória podem ser usados.
- **min age** - a idade mínima (em segundos) da entrada de diretório a ser listada. Entradas mais recentes não serão listadas. Sufixos de tempo podem ser usados.
- **max age** - a idade máxima (em segundos) da entrada de diretório a ser listada. Entradas tão antigas e mais antigas não serão listadas (tempo de modificação). Sufixos de tempo podem ser usados.
- **regex excl dir** - uma expressão regular descrevendo o padrão de nome do diretório a ser excluído. Todo o conteúdo do diretório será excluído (em contraste com `regex excl`)

Comentários:

- No Windows, este item segue caminhos UNC, o que pode ser um risco de segurança;
- Variáveis de ambiente, por exemplo, %APP_HOME%, $HOME e %TEMP% não são suportadas;
- Pseudo-diretórios "." e ".." nunca são listados;
- Links simbólicos nunca são seguidos para percorrer diretórios;
- Tanto `regex incl` quanto `regex excl` são aplicados a arquivos e diretórios ao gerar a lista de entradas, mas são ignorados ao escolher subdiretórios para percorrer (se `regex incl` for "(?i)^.+\.zip$" e `max depth` não for definido, então todos os subdiretórios serão percorridos, mas apenas os arquivos do tipo zip serão contados).
- O tempo de execução é limitado pelo valor de timeout na configuração do agent. Como a varredura de diretórios grandes pode levar mais tempo do que isso, nenhum dado será retornado e o item ficará como não suportado. A lista parcial não será retornada.
- Ao filtrar por tamanho, apenas arquivos regulares têm tamanhos significativos. No Linux e BSD, diretórios também têm tamanhos diferentes de zero (normalmente alguns Kb). Dispositivos têm tamanho zero, por exemplo, o tamanho de **/dev/sda1** não reflete o tamanho da partição respectiva. Portanto, ao usar `min size` e `max size`, é aconselhável especificar `types incl` como "*file*", para evitar surpresas.

Exemplos:

```
vfs.dir.get[/dev] #recupera a lista de dispositivos em /dev (Linux)
vfs.dir.get["C:\Users\ADMINI~1\AppData\Local\Temp"] #recupera a lista de arquivos em um diretório temporár
```

vfs.dir.size[dir,<regex incl>,<regex excl>,<mode>,<max depth>,<regex excl dir>]

 O tamanho do diretório (em bytes).
 Valor de retorno: *Integer*.
 Plataformas suportadas: Linux, Windows. O item pode funcionar em outras plataformas do tipo UNIX.

Parâmetros:

- **dir** - o caminho absoluto para o diretório;
- **regex incl** - uma expressão regular descrevendo o padrão de nome da entidade (arquivo, diretório, link simbólico) a ser incluída; inclui todos se vazio (valor padrão);
- **regex excl** - uma expressão regular descrevendo o padrão de nome da entidade (arquivo, diretório, link simbólico) a ser excluída; não exclui nenhum se vazio (valor padrão);
- **mode** - valores possíveis: *apparent* (padrão) - obtém tamanhos aparentes de arquivos em vez do uso do disco (atua como `du -sb dir`), *disk* - obtém o uso do disco (atua como `du -s -B1 dir`). Diferente do comando du, o item vfs.dir.size leva em consideração arquivos ocultos ao calcular o tamanho do diretório (atua como `du -sb .[^.]* *` dentro do dir).
- **max depth** - a profundidade máxima de subdiretórios a serem percorridos: **-1** (padrão) - ilimitado, **0** - não desce em subdiretórios.
- **regex excl dir** - uma expressão regular descrevendo o padrão de nome do diretório a ser excluído. Todo o conteúdo do diretório será excluído (em contraste com `regex excl`)

Comentários:

- No Windows, este item segue caminhos UNC, o que pode ser um risco de segurança;
- Apenas diretórios com pelo menos permissão de leitura para o usuário *zabbix* são calculados. Para diretórios apenas com permissão de leitura, o tamanho do próprio diretório é calculado. Diretórios com permissões de leitura e execução são calculados incluindo o conteúdo.

- Com diretórios grandes ou discos lentos, este item pode atingir o tempo limite devido à configuração Timeout nos arquivos de configuração do agent e do server/proxy. Aumente os valores de timeout conforme necessário.
- O limite de tamanho do arquivo depende do suporte a arquivos grandes.

Exemplos:

```
vfs.dir.size[/tmp,log] #calcula o tamanho de todos os arquivos em /tmp que contenham 'log' em seus nomes
vfs.dir.size[/tmp,log,^.+\.old$] #calcula o tamanho de todos os arquivos em /tmp que contenham 'log' em se
```

vfs.file.cksum[file,<mode>]

 O checksum do arquivo, calculado pelo algoritmo UNIX cksum.
 Valor de retorno: *Integer* - com mode como *crc32*, *String* - com mode como *md5*, *sha256*.
 Veja plataformas suportadas.

Parâmetros:

- **file** - o caminho completo para o arquivo;
- **mode** - *crc32* (padrão), *md5* ou *sha256*.

Comentários:

- No Windows, este item segue caminhos UNC, o que pode ser um risco de segurança;
- O limite de tamanho do arquivo depende do suporte a arquivos grandes.

Exemplo:

```
vfs.file.cksum[/etc/passwd]
```

Exemplos de valores retornados (crc32/md5/sha256 respectivamente):

```
675436101
9845acf68b73991eb7fd7ee0ded23c44
ae67546e4aac995e5c921042d0cf0f1f7147703aa42bfbfb65404b30f238f2dc
```

vfs.file.contents[file,<encoding>]

 Recupera o conteúdo de um arquivo[7].
 Valor de retorno: *Texto*.
 Veja plataformas suportadas.

Parâmetros:

- **file** - o caminho completo para o arquivo;
- **encoding** - o identificador da página de código.

Comentários:

- No Windows, este item segue caminhos UNC, o que pode ser um risco de segurança;
- O valor de retorno é limitado a 16MB (incluindo espaços em branco à direita que são truncados); limites do banco de dados também se aplicam;
- Uma string vazia é retornada se o arquivo estiver vazio ou contiver apenas caracteres LF/CR;
- A marca de ordem de bytes (BOM) é excluída da saída.

Exemplo:

```
vfs.file.contents[/etc/passwd]
```

vfs.file.exists[file,<types incl>,<types excl>]

 Verifica se o arquivo existe.
 Valor de retorno: 0 - não encontrado; 1 - arquivo do tipo especificado existe.
 Veja as plataformas suportadas.

Parâmetros:

- **file** - o caminho completo para o arquivo;
- **types incl** - a lista de tipos de arquivo a serem incluídos, valores possíveis: *file* (arquivo regular, padrão (se types_excl não estiver definido)), *dir* (diretório), *sym* (link simbólico), *sock* (socket), *bdev* (dispositivo de bloco), *cdev* (dispositivo de caractere), *fifo* (FIFO), *dev* (sinônimo de "bdev,cdev"), *all* (todos os tipos mencionados, padrão se types_excl estiver definido).
- **types excl** - a lista de tipos de arquivo a serem excluídos, veja types_incl para valores possíveis (por padrão, nenhum tipo é excluído)

Comentários:

- No Windows, este item segue caminhos UNC, o que pode ser um risco de segurança;
- Vários tipos devem ser separados por vírgula e todo o conjunto entre aspas "";
- Se o mesmo tipo estiver em <types_incl> e <types_excl>, arquivos desse tipo serão excluídos;
- O limite de tamanho do arquivo depende do suporte a arquivos grandes.

Exemplos:

```
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]

 Retorna informações sobre um arquivo.
 Valor de retorno: *objeto JSON*.
 Veja plataformas suportadas.

Parâmetro:

- **file** - o caminho completo para o arquivo

Comentários:

- No Windows, este item segue caminhos UNC, o que pode ser um risco de segurança;
- Tipos de arquivos suportados em sistemas UNIX-like: arquivo regular, diretório, link simbólico, socket, bloco de dispositivo, dispositivo de caractere, FIFO.
- O limite de tamanho do arquivo depende do suporte a arquivos grandes.

Exemplo:

```
vfs.file.get[/etc/passwd] #retorna um JSON com informações sobre o arquivo /etc/passwd (tipo, usuário, per
```

vfs.file.md5sum[file]

 O checksum MD5 do arquivo.
 Valor de retorno: String de caracteres (hash MD5 do arquivo).
 Veja as plataformas suportadas.

Parâmetro:

- **file** - o caminho completo para o arquivo

Comentários:

- No Windows, este item segue caminhos UNC, o que pode ser um risco de segurança;
- O limite de tamanho do arquivo depende do suporte a arquivos grandes.

Exemplo:

```
vfs.file.md5sum[/usr/local/etc/zabbix_agentd.conf]
```

Exemplo de valor retornado:

```
b5052decb577e0fffd622d6ddc017e82
```

vfs.file.owner[file,<ownertype>,<resulttype>]

 Recupera o proprietário de um arquivo.
 Valor de retorno: *String*.
 Veja as plataformas suportadas.

Parâmetros:

- **file** - o caminho completo para o arquivo;
- **ownertype** - *user* (padrão) ou *group* (somente Unix);
- **resulttype** - *name* (padrão) ou *id*; para id - retorna uid/gid no Unix, SID no Windows.

Comentários:

- No Windows, este item segue caminhos UNC, o que pode ser um risco de segurança;
- O limite de tamanho do arquivo depende do suporte a arquivos grandes.

Exemplos:

```
vfs.file.owner[/tmp/zabbix_server.log] #retorna o proprietário do arquivo /tmp/zabbix_server.log
vfs.file.owner[/tmp/zabbix_server.log,,id] #retorna o ID do proprietário do arquivo /tmp/zabbix_server.log
```

vfs.file.permissions[file]

 Retorna uma string de 4 dígitos contendo o número octal com as permissões UNIX.
 Valor de retorno: *String*.
 Plataformas suportadas: Linux. O item pode funcionar em outras plataformas do tipo UNIX.

Parâmetros:

- **file** - o caminho completo para o arquivo

O limite de tamanho do arquivo depende do suporte a arquivos grandes.

Exemplo:

```
vfs.file.permissions[/etc/passwd] #retorna as permissões de /etc/passwd, por exemplo, '0644'
```

vfs.file.regexp[file,regexp,<encoding>,<start line>,<end line>,<output>]

 Recupera uma string no arquivo[7].
 Valor de retorno: A linha contendo a string correspondente ou conforme especificado pelo parâmetro opcional `output`.
 Veja as plataformas suportadas.

Parâmetros:

- **file** - o caminho completo para o arquivo;
- **regexp** - uma expressão regular descrevendo o padrão necessário;
- **encoding** - o identificador da página de código;
- **start line** - o número da primeira linha a ser pesquisada (primeira linha do arquivo por padrão);
- **end line** - o número da última linha a ser pesquisada (última linha do arquivo por padrão);
- **output** - um modelo de formatação de saída opcional. A sequência de escape **\0** é substituída pela parte do texto correspondente (do primeiro caractere onde a correspondência começa até o caractere onde termina), enquanto uma sequência de escape **\N** (onde N=1...9) é substituída pelo N-ésimo grupo correspondente (ou uma string vazia se N exceder o número de grupos capturados).

Comentários:

- No Windows, este item segue caminhos UNC, o que pode ser um risco de segurança;
- O limite de tamanho do arquivo depende do suporte a arquivos grandes.
- Apenas a primeira linha correspondente é retornada;
- Uma string vazia é retornada se nenhuma linha corresponder à expressão;
- A marca de ordem de bytes (BOM) é excluída da saída;
- A extração de conteúdo usando o parâmetro `output` ocorre no agent.

Exemplos:

```
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] → obtendo o ID do usuário *zabbix*
```

vfs.file.regmatch[file,regexp,<encoding>,<start line>,<end line>]

 Procura uma string no arquivo[7].
 Valores de retorno: 0 - correspondência não encontrada; 1 - encontrada.
 Veja as plataformas suportadas.

Parâmetros:

- **file** - o caminho completo para o arquivo;
- **regexp** - uma expressão regular descrevendo o padrão necessário;
- **encoding** - o identificador da página de código;
- **start line** - o número da primeira linha a ser pesquisada (primeira linha do arquivo por padrão);
- **end line** - o número da última linha a ser pesquisada (última linha do arquivo por padrão).

Comentários:

- No Windows, este item segue caminhos UNC, o que pode ser um risco de segurança;
- O limite de tamanho do arquivo depende do suporte a arquivos grandes.
- A marca de ordem de bytes (BOM) é ignorada.

Exemplo:

```
vfs.file.regmatch[/var/log/app.log,error]
```

vfs.file.size[file,<mode>]

 O tamanho do arquivo (em bytes).
 Valor de retorno: *Integer*.
 Veja as plataformas suportadas.

Parâmetros:

- **file** - o caminho completo para o arquivo;
- **mode** - valores possíveis: *bytes* (padrão) ou *lines* (linhas em branco também são contadas).

Comentários:

- No Windows, este item segue caminhos UNC, o que pode ser um risco de segurança;
- O arquivo deve ter permissões de leitura para o usuário *zabbix*;

- O limite de tamanho do arquivo depende do suporte a arquivos grandes.

Exemplo:

```
vfs.file.size[/var/log/syslog]
```

vfs.file.time[file,<mode>]

 A informação de tempo do arquivo.
 Valor de retorno: *Integer* (timestamp Unix).
 Veja as plataformas suportadas.

Parâmetros:

- **file** - o caminho completo para o arquivo;
- **mode** - valores possíveis:
modify (padrão) - o último tempo de modificação do conteúdo do arquivo,
access - o último tempo de leitura do arquivo,
change - o último tempo de alteração das propriedades do arquivo

Comentários:

- No Windows, este item segue caminhos UNC, o que pode ser um risco de segurança;
- O limite de tamanho do arquivo depende do suporte a arquivos grandes.

Exemplo:

```
vfs.file.time[/etc/passwd,modify]
```

vfs.fs.discovery

 A lista de sistemas de arquivos montados com seu tipo e opções de montagem. Usado para descoberta de baixo nível.
 Valor de retorno: *objeto JSON*.
 Plataformas suportadas: Linux, FreeBSD, Solaris, HP-UX, AIX, MacOS X, OpenBSD, NetBSD, Windows.

vfs.fs.get

 A lista de sistemas de arquivos montados com seu tipo, espaço em disco disponível, estatísticas de inode e opções de montagem. Pode ser usado para descoberta de baixo nível.
 Valor de retorno: *objeto JSON*.
 Plataformas suportadas: Linux, FreeBSD, Solaris, HP-UX, AIX, MacOS X, OpenBSD, NetBSD, Windows.

Comentários:

- Sistemas de arquivos com contagem de inodes igual a zero, o que pode ocorrer em sistemas de arquivos com inodes dinâmicos (por exemplo, btrfs), também são relatados;
- Veja também: Descoberta de sistemas de arquivos montados.

vfs.fs.inode[fs,<mode>]

 O número ou a porcentagem de inodes.
 Valor de retorno: *Integer* - para número; *Float* - para porcentagem.
 Veja as plataformas suportadas.

Parâmetros:

- **fs** - o sistema de arquivos;
- **mode** - valores possíveis: *total* (padrão), *free*, *used*, *pfree* (livre, porcentagem) ou *pused* (usado, porcentagem).

Se a contagem de inodes for igual a zero, o que pode ser o caso para sistemas de arquivos com inodes dinâmicos (por exemplo, btrfs), os valores de pfree/pused serão reportados como ”100” e ”0”, respectivamente.

Exemplo:

```
vfs.fs.inode[/,pfree]
```

vfs.fs.size[fs,<mode>]

 O espaço em disco em bytes ou em porcentagem do total.
 Valor de retorno: *Integer* - para bytes; *Float* - para porcentagem.
 Veja as plataformas suportadas.

Parâmetros:

- **fs** - o sistema de arquivos;
- **mode** - valores possíveis: *total* (padrão), *free*, *used*, *pfree* (livre, porcentagem) ou *pused* (usado, porcentagem).

Comentários:

- Se o sistema de arquivos não estiver montado, retorna o tamanho de um sistema de arquivos local onde o ponto de montagem está localizado;
- O espaço reservado de um sistema de arquivos é levado em consideração e não é incluído ao usar o modo *free*.

Exemplo:

```
vfs.fs.size[/tmp,free]
```

vm.memory.size[<mode>]

 O tamanho da memória em bytes ou em porcentagem do total.
 Valor de retorno: *Integer* - para bytes; *Float* - para porcentagem.
 Veja as plataformas suportadas.

Parâmetro:

- **mode** - valores possíveis: *total* (padrão), *active*, *anon*, *buffers*, *cached*, *exec*, *file*, *free*, *inactive*, *pinned*, *shared*, *slab*, *wired*, *used*, *pused* (usado, porcentagem), *available* ou *pavailable* (disponível, porcentagem).

Comentários:

- Este item aceita três categorias de parâmetros:
  - *total* - quantidade total de memória;
  - tipos de memória específicos da plataforma: *active*, *anon*, *buffers*, *cached*, *exec*, *file*, *free*, *inactive*, *pinned*, *shared*, *slab*, *wired*;
  - estimativas em nível de usuário de quanto de memória está em uso e disponível: *used*, *pused*, *available*, *pavailable*;
- O parâmetro de modo *active* é suportado apenas no FreeBSD, HP-UX, MacOS X, OpenBSD, NetBSD;
- Os parâmetros de modo *anon*, *exec*, *file* são suportados apenas no NetBSD;
- O parâmetro de modo *buffers* é suportado apenas no Linux, FreeBSD, OpenBSD, NetBSD;
- O parâmetro de modo *cached* é suportado apenas no Linux, FreeBSD, AIX, OpenBSD, NetBSD;
- Os parâmetros de modo *inactive*, *wired* são suportados apenas no FreeBSD, MacOS X, OpenBSD, NetBSD;
- O parâmetro de modo *pinned* é suportado apenas no AIX;
- O parâmetro de modo *shared* é suportado apenas no Linux 2.4, FreeBSD, OpenBSD, NetBSD;
- Veja também detalhes adicionais para este item.

Exemplo:

```
vm.memory.size[pavailable]
```

web.page.get[host,<path>,<port>]

 Obtém o conteúdo de uma página web.
 Valor de retorno: Fonte da página web como texto (incluindo cabeçalhos).
 Veja as plataformas suportadas.

Parâmetros:

- **host** - o nome do host ou URL (como `scheme://host:port/path`, onde apenas *host* é obrigatório). Esquemas de URL permitidos: *http*, *https*[4]. Um esquema ausente será tratado como *http*. Se uma URL for especificada, `path` e `port` devem estar vazios. Especificar nome de usuário/senha ao conectar-se a servidores que exigem autenticação, por exemplo: `http://user:password@www.example.com` só é possível com suporte a cURL [4]. Punycode é suportado em nomes de host.
- **path** - o caminho para um documento HTML (o padrão é /);
- **port** - o número da porta (o padrão é 80 para HTTP)

Comentários:

- Este item se torna não suportado se o recurso especificado em `host` não existir ou estiver indisponível;
- `host` pode ser um nome de host, nome de domínio, endereço IPv4 ou IPv6. Mas para endereço IPv6 o agent Zabbix deve ser compilado com suporte a IPv6 habilitado.

Exemplos:

```
web.page.get[www.example.com,index.php,80]
web.page.get[https://www.example.com]
web.page.get[https://blog.example.com/?s=zabbix]
web.page.get[localhost:80]
web.page.get["[::1]/server-status"]
```

web.page.perf[host,<path>,<port>]

 O tempo de carregamento de uma página web completa (em segundos).
 Valor de retorno: *Float*.
 Veja as plataformas suportadas.

Parâmetros:

- **host** - o nome do host ou URL (como `scheme://host:port/path`, onde apenas *host* é obrigatório). Os esquemas de URL permitidos: *http*, *https*[4]. Um esquema ausente será tratado como *http*. Se uma URL for especificada, `path` e `port` devem estar vazios. Especificar nome de usuário/senha ao conectar-se a servidores que exigem autenticação, por exemplo:

`http://user:password@www.example.com` só é possível com suporte a cURL [4]. Punycode é suportado em nomes de host.
- **path** - o caminho para um documento HTML (o padrão é /);
- **port** - o número da porta (o padrão é 80 para HTTP)

Comentários:

- Este item se torna não suportado se o recurso especificado em `host` não existir ou não estiver disponível;
- `host` pode ser um nome de host, nome de domínio, endereço IPv4 ou IPv6. Mas para endereço IPv6 o agent do Zabbix deve ser compilado com suporte a IPv6 habilitado.

Exemplos:

```
web.page.perf[www.example.com,index.php,80]
web.page.perf[https://www.example.com]
```

web.page.regexp[host,<path>,<port>,regexp,<length>,<output>]

 Procura uma string na página web.
 Valor de retorno: A string correspondente ou conforme especificado pelo parâmetro opcional `output`.
 Veja as plataformas suportadas.

Parâmetros:

- **host** - o nome do host ou URL (como `scheme://host:port/path`, onde apenas *host* é obrigatório). Esquemas de URL permitidos: *http*, *https*[4]. Um esquema ausente será tratado como *http*. Se uma URL for especificada, `path` e `port` devem estar vazios. Especificar nome de usuário/senha ao conectar-se a servidores que exigem autenticação, por exemplo: `http://user:password@www.example.com` só é possível com suporte a cURL [4]. Punycode é suportado em nomes de host.
- **path** - o caminho para um documento HTML (o padrão é /);
- **port** - o número da porta (o padrão é 80 para HTTP)
- **regexp** - uma expressão regular descrevendo o padrão necessário;
- **length** - o número máximo de caracteres a serem retornados;
- **output** - um modelo de formatação de saída opcional. A sequência de escape **\0** é substituída pela parte correspondente do texto (do primeiro caractere onde a correspondência começa até o caractere onde a correspondência termina), enquanto uma sequência de escape **\N** (onde N=1...9) é substituída pelo N-ésimo grupo correspondente (ou uma string vazia se N exceder o número de grupos capturados).

Comentários:

- Este item se torna não suportado se o recurso especificado em `host` não existir ou estiver indisponível;
- `host` pode ser um nome de host, nome de domínio, endereço IPv4 ou IPv6. Mas para endereço IPv6 o agent Zabbix deve ser compilado com suporte a IPv6 habilitado.
- A extração de conteúdo usando o parâmetro `output` ocorre no agent.

Exemplos:

```
web.page.regexp[www.example.com,index.php,80,OK,2]
web.page.regexp[https://www.example.com,,,OK,2]
```

agent.hostmetadata

 A metadata do host do agent.
 Valor de retorno: *String*.
 Veja as plataformas suportadas.

Retorna o valor dos parâmetros HostMetadata ou HostMetadataItem, ou uma string vazia se nenhum deles estiver definido.

agent.hostname

 O nome do host do agent.
 Valor de retorno: *String*.
 Veja plataformas suportadas.

Retorna:

- Como checagem passiva - o nome do primeiro host listado no parâmetro Hostname do arquivo de configuração do agent;
- Como checagem ativa - o nome do hostname atual.

agent.ping

 A verificação de disponibilidade do agent.
 Valor de retorno: Nada - indisponível; 1 - disponível.
 Veja plataformas suportadas.

Use a função de trigger **nodata()** para verificar a indisponibilidade do host.

agent.variant

 A variante do agent Zabbix (agent Zabbix ou agent Zabbix 2).
 Valor de retorno: 1 - agent Zabbix; 2 - agent Zabbix 2.
 Veja plataformas suportadas.

agent.version

 A versão do agent Zabbix.
 Valor de retorno: *String*.
 Veja plataformas suportadas.

Exemplo de valor retornado:

```
6.0.3
```

zabbix.stats[<ip>,<port>]

 Retorna um conjunto de métricas internas do Zabbix server ou proxy. Se `ip` e `port` forem especificados, as métricas são recuperadas da instância remota; caso contrário, da instância local.
 Valor de retorno: *objeto JSON*.
 Veja as plataformas suportadas.

Parâmetros:

- **ip** - a lista de IP/DNS/máscara de rede dos servers/proxies a serem consultados remotamente (padrão é 127.0.0.1);
- **port** - a porta do server/proxy a ser consultado remotamente (padrão é 10051)

Comentários:

- Um conjunto selecionado de métricas internas é retornado por este item. Para detalhes, veja Monitoramento remoto de estatísticas do Zabbix;
- Observe que a solicitação de estatísticas só será aceita dos endereços listados no parâmetro 'StatsAllowedIP' do server/proxy na instância de destino.

zabbix.stats[<ip>,<port>,queue,<from>,<to>]

 Retorna o número de items monitorados na fila que estão atrasados no Zabbix server ou proxy. Se `ip` e `port` forem especificados, as métricas são recuperadas da instância remota; caso contrário, da instância local.
 Valor de retorno: *objeto JSON*.
 Veja as plataformas suportadas.

Parâmetros:

- **ip** - a lista de IP/DNS/máscara de rede dos servers/proxies a serem consultados remotamente (o padrão é 127.0.0.1);
- **port** - a porta do server/proxy a ser consultado remotamente (o padrão é 10051)
- **queue** - constante (para ser usada como está)
- **from** - atrasado por pelo menos (o padrão é 6 segundos)
- **to** - atrasado por no máximo (o padrão é infinito)

Observe que a solicitação de estatísticas só será aceita dos endereços listados no parâmetro 'StatsAllowedIP' do server/proxy na instância de destino.

Notas de rodapé

[1] Nota específica para Linux. O agent Zabbix deve ter acesso somente leitura ao sistema de arquivos */proc*. Patches de kernel do www.grsecurity.org limitam os direitos de acesso de usuários não privilegiados.

[2] `vfs.dev.read[]`, `vfs.dev.write[]`: O agent Zabbix encerrará conexões de dispositivos "obsoletas" se os valores do item não forem acessados por mais de 3 horas. Isso pode acontecer se um sistema tiver dispositivos com caminhos que mudam dinamicamente ou se um dispositivo for removido manualmente. Observe também que esses itens, se usarem um intervalo de atualização de 3 horas ou mais, sempre retornarão '0'.

[3] `vfs.dev.read[]`, `vfs.dev.write[]`: Se o padrão *all* for usado para o primeiro parâmetro, a chave retornará estatísticas resumidas, incluindo todos os dispositivos de bloco como sda, sdb e suas partições (sda1, sda2, sdb3...) e múltiplos dispositivos (MD raid) baseados nesses dispositivos/partições de bloco e volumes lógicos (LVM) baseados nesses dispositivos/partições de bloco. Nesses casos, os valores retornados devem ser considerados apenas como valores relativos (dinâmicos no tempo), mas não como valores absolutos.

[4] SSL (HTTPS) é suportado apenas se o agent for compilado com suporte a cURL. Caso contrário, o item ficará como não suportado.

[5] Os valores `bytes` e `errors` não são suportados para interfaces loopback em sistemas Solaris até e incluindo o Solaris 10 6/06, pois as estatísticas de bytes, erros e utilização não são armazenadas e/ou relatadas pelo kernel. No entanto, se você estiver monitorando um sistema Solaris via net-snmp, valores podem ser retornados, pois o net-snmp possui código legado do cmu-snmp datado de 1997 que, ao falhar em ler valores de bytes das estatísticas da interface, retorna o contador de pacotes (que existe em interfaces loopback) multiplicado por um valor arbitrário de 308. Isso pressupõe que o comprimento médio de um pacote é de 308 octetos, o que é uma estimativa muito grosseira, já que o limite de MTU em sistemas Solaris para interfaces loopback é de 8892 bytes. Esses valores não devem ser considerados corretos ou mesmo próximos da precisão. Eles são estimativas. O agent Zabbix não faz nenhum tipo de estimativa, mas o net-snmp retornará um valor para esses campos.

[6] A linha de comando no Solaris, obtida de /proc/pid/psinfo, é limitada a 80 bytes e contém a linha de comando conforme estava quando o processo foi iniciado.

[7] Os itens `vfs.file.contents[]`, `vfs.file.regexp[]`, `vfs.file.regmatch[]` podem ser usados para recuperar o conteúdo de arquivos. Se você deseja restringir o acesso a arquivos específicos com informações confidenciais, execute o agent Zabbix sob um usuário que não tenha permissões de acesso para visualizar esses arquivos.

Uso com utilitários de linha de comando

Observe que, ao testar ou usar chaves de item com zabbix_agentd ou zabbix_get a partir da linha de comando, você também deve considerar a sintaxe do shell.

Por exemplo, se um determinado parâmetro da chave precisar ser colocado entre aspas duplas, você deverá escapar explicitamente as aspas duplas, caso contrário, elas serão removidas pelo shell como caracteres especiais e não serão passadas para o utilitário Zabbix.

Exemplos:

```
zabbix_agentd -t 'vfs.dir.count[/var/log,,,"file,dir",,0]'

zabbix_agentd -t vfs.dir.count[/var/log,,,\"file,dir\",,0]
```

Configurações de codificação

Para garantir que os dados adquiridos não sejam corrompidos, você pode especificar a codificação correta para processar a verificação (por exemplo, 'vfs.file.contents') no parâmetro `encoding`. A lista de codificações suportadas (identificadores de code page) pode ser encontrada na documentação da libiconv (Projeto GNU) ou na documentação do Microsoft Windows SDK para "Code Page Identifiers". Observe que a Microsoft às vezes marca algumas code pages como "disponíveis apenas para aplicativos gerenciados" — ou seja, expostas apenas via runtime .NET e não disponíveis através das APIs nativas do Win32. O agent do Zabbix implementa sua própria lógica de conversão de codificação, portanto, essas code pages são suportadas pelo agent do Zabbix mesmo quando as funções nativas do Windows não as fornecem.

Se nenhuma codificação for especificada no parâmetro `encoding`, as seguintes estratégias de resolução são aplicadas:

- Se a codificação não for especificada (ou for uma string vazia), assume-se que é UTF-8, os dados são processados "como estão";
- Análise de BOM - aplicável para items 'vfs.file.contents', 'vfs.file.regexp', 'vfs.file.regmatch'. É feita uma tentativa de determinar a codificação correta usando a marca de ordem de bytes (BOM) no início do arquivo. Se o BOM não estiver presente, a resolução padrão (veja acima) será aplicada.

Solução de problemas de itens do agent

No caso de verificações passivas, para evitar que o item não obtenha nenhum valor porque a solicitação do server ao agent atinge o tempo limite primeiro, o seguinte deve ser observado:

- Quando a versão do agent for mais antiga que a versão do server, o valor de *Timeout* na configuração do item (ou timeout global) pode precisar ser maior do que o valor de `Timeout` no arquivo de configuração do agent.
- Quando a versão do agent for mais recente que a versão do server, o valor de `Timeout` no arquivo de configuração do server pode precisar ser maior do que o valor de `Timeout` no arquivo de configuração do agent.

1 Agente Zabbix 2

O agent Zabbix 2 oferece suporte a todas as chaves de item suportadas para o agent Zabbix em Unix e Windows. Esta página fornece detalhes sobre as chaves de item adicionais, que você pode usar apenas com o agent Zabbix 2, agrupadas pelo plugin ao qual pertencem.

As chaves de item são listadas sem parâmetros e informações adicionais. Clique na chave de item para ver todos os detalhes.

Chave do item	Descrição	Plugin
ceph.df.details	O uso de dados do cluster e a distribuição entre os pools.	Ceph
ceph.osd.stats	Estatísticas agregadas e por OSD.	
ceph.osd.discovery	A lista de OSDs descobertos.	
ceph.osd.dump	Os limites de uso e status dos OSDs.	
ceph.ping	Testa se uma conexão com o Ceph pode ser estabelecida.	
ceph.pool.discovery	A lista de pools descobertos.	
ceph.status	O status geral do cluster.	
docker.container_info	Informações de baixo nível sobre um container.	Docker
docker.container_stats	As estatísticas de uso de recursos do container.	

Chave do item	Descrição	Plugin
docker.containers	Retorna a lista de containers.	
docker.containers.discovery	Retorna a lista de containers. Usado para descoberta de baixo nível.	
docker.data.usage	Informações sobre o uso atual de dados.	
docker.images	Retorna a lista de imagens.	
docker.images.discovery	Retorna a lista de imagens. Usado para descoberta de baixo nível.	
docker.info	As informações do sistema.	
docker.ping	Testa se o daemon Docker está ativo ou não.	
ember.get	Retorna o resultado do dispositivo solicitado.	Ember+
memcached.ping	Testa se uma conexão está ativa ou não.	Memcached
memcached.stats	Obtém a saída do comando STATS.	
mongodb.collection.stats	Retorna uma variedade de estatísticas de armazenamento para uma determinada coleção.	MongoDB
mongodb.collections.discovery	Retorna uma lista de coleções descobertas.	
mongodb.collections.usage	Retorna as estatísticas de uso das coleções.	
mongodb.connpool.stats	Retorna informações sobre as conexões de saída abertas da instância atual do banco de dados para outros membros do cluster fragmentado ou conjunto de réplicas.	
mongodb.db.stats	Retorna as estatísticas que refletem o estado do sistema do banco de dados.	
mongodb.db.discovery	Retorna uma lista de bancos de dados descobertos.	
mongodb.jumbo_chunks.count	Retorna a contagem de jumbo chunks.	
mongodb.oplog.stats	Retorna o status do conjunto de réplicas, usando dados coletados do oplog.	
mongodb.ping	Testa se uma conexão está ativa ou não.	
mongodb.rs.config	Retorna a configuração atual do conjunto de réplicas.	
mongodb.rs.status	Retorna o status do conjunto de réplicas do ponto de vista do membro onde o método é executado.	
mongodb.server.status	Retorna o estado do banco de dados.	
mongodb.sh.discovery	Retorna a lista de shards descobertos presentes no cluster.	
mongodb.version	Retorna a versão do servidor de banco de dados.	
mqtt.get	Assina um tópico ou tópicos específicos (com curingas) do broker fornecido e aguarda publicações.	MQTT
mssql.availability.group.get	Retorna grupos de disponibilidade.	MSSQL
mssql.custom.query	Retorna o resultado de uma consulta personalizada.	
mssql.db.get	Retorna todos os bancos de dados MSSQL disponíveis.	
mssql.job.status.get	Retorna o status dos jobs.	
mssql.last.backup.get	Retorna o horário do último backup de todos os bancos de dados.	
mssql.local.db.get	Retorna bancos de dados que estão participando de um grupo de disponibilidade Always On e réplica (primária ou secundária) e estão localizados no servidor ao qual a conexão foi estabelecida.	
mssql.mirroring.get	Retorna informações de espelhamento.	
mssql.nonlocal.db.get	Retorna bancos de dados que estão participando de um grupo de disponibilidade Always On e réplica (primária ou secundária) localizados em outros servidores (o banco de dados não é local para a instância do SQL Server à qual a conexão foi estabelecida).	
mssql.perfcounter.get	Retorna os contadores de desempenho.	
mssql.ping	Testa se uma conexão está ativa ou não.	
mssql.quorum.get	Retorna as informações de quórum.	
mssql.quorum.member.get	Retorna os membros do quórum.	
mssql.replica.get	Retorna as réplicas.	
mssql.version	Retorna a versão do MSSQL.	
mysql.custom.query	Retorna o resultado de uma consulta personalizada.	MySQL
mysql.db.discovery	Retorna a lista de bancos de dados MySQL.	
mysql.db.size	O tamanho do banco de dados em bytes.	
mysql.get_status_variables	Valores das variáveis de status global.	
mysql.ping	Testa se uma conexão está ativa ou não.	
mysql.replication.discovery	Retorna a lista de replicações MySQL.	
mysql.replication.get_slave_status	O status da replicação.	
mysql.version	A versão do MySQL.	
net.dns.get	Executa uma consulta DNS e retorna informações detalhadas do registro DNS.	Rede
oracle.diskgroups.stats	Retorna as estatísticas dos grupos de discos Automatic Storage Management (ASM).	Oracle
oracle.diskgroups.discovery	Retorna a lista de grupos de discos ASM.	

Chave do item	Descrição	Plugin
oracle.archive.info	As estatísticas dos logs de arquivamento.	
oracle.cdb.info	As informações dos Container Databases (CDBs).	
oracle.custom.query	O resultado de uma consulta personalizada.	
oracle.datafiles.stats	Retorna as estatísticas dos arquivos de dados.	
oracle.db.discovery	Retorna a lista de bancos de dados.	
oracle.fra.stats	Retorna as estatísticas da Fast Recovery Area (FRA).	
oracle.instance.info	As estatísticas da instância.	
oracle.pdb.info	As informações dos Pluggable Databases (PDBs).	
oracle.pdb.discovery	Retorna a lista de PDBs.	
oracle.pga.stats	Retorna as estatísticas da Program Global Area (PGA).	
oracle.ping	Testa se uma conexão com o Oracle pode ser estabelecida.	
oracle.proc.stats	Retorna as estatísticas dos processos.	
oracle.redolog.info	As informações do arquivo de log do arquivo de controle.	
oracle.sga.stats	Retorna as estatísticas da System Global Area (SGA).	
oracle.sessions.stats	Retorna as estatísticas das sessões.	
oracle.sys.metrics	Retorna um conjunto de valores de métricas do sistema.	
oracle.sys.params	Retorna um conjunto de valores de parâmetros do sistema.	
oracle.ts.stats	Retorna as estatísticas dos tablespaces.	
oracle.ts.discovery	Retorna uma lista de tablespaces.	
oracle.user.info	Retorna informações do usuário Oracle.	
oracle.version	Retorna a versão do servidor de banco de dados.	
pgsql.autovacuum.count	O número de workers autovacuum.	PostgreSQL
pgsql.archive	As informações sobre arquivos arquivados.	
pgsql.bgwriter	O número combinado de checkpoints para o cluster de banco de dados, dividido por tipo de checkpoint.	
pgsql.cache.hit	A taxa de acerto do buffer cache do PostgreSQL.	
pgsql.connections	Retorna conexões por tipo.	
pgsql.custom.query	Retorna o resultado de uma consulta personalizada.	
pgsql.db.age	A idade do FrozenXID mais antigo do banco de dados.	
pgsql.db.bloating_tables	O número de tabelas inchadas por banco de dados.	
pgsql.db.discovery	A lista de bancos de dados PostgreSQL.	
pgsql.db.size	O tamanho do banco de dados em bytes.	
pgsql.dbstat	Coleta as estatísticas por banco de dados.	
pgsql.dbstat.sum	Os dados resumidos para todos os bancos de dados em um cluster.	
pgsql.locks	As informações sobre locks concedidos por banco de dados.	
pgsql.oldest.xid	A idade do XID mais antigo.	
pgsql.ping	Testa se uma conexão está ativa ou não.	
pgsql.queries	Métricas de consulta por tempo de execução.	
pgsql.replication.count	O número de servidores standby.	
pgsql.replication.process	O flush lag, write lag e replay lag por cada processo de envio.	
pgsql.replication.process.discovery	A descoberta do nome do processo de replicação.	
pgsql.replication.recovery_role	O status de recuperação.	
pgsql.replication.status	O status da replicação.	
pgsql.replication_lag.b	O atraso de replicação em bytes.	
pgsql.replication_lag.sec	O atraso de replicação em segundos.	
pgsql.uptime	O tempo de atividade do PostgreSQL em milissegundos.	
pgsql.version	Retorna a versão do PostgreSQL.	
pgsql.wal.stat	As estatísticas do WAL.	
redis.config	Obtém os parâmetros de configuração de uma instância Redis que correspondem ao padrão.	Redis
redis.info	Obtém a saída do comando INFO.	
redis.ping	Testa se uma conexão está ativa ou não.	
redis.slowlog.count	O número de entradas de log lento desde que o Redis foi iniciado.	
smart.attribute.discovery	Retorna uma lista de atributos de dispositivos S.M.A.R.T.	S.M.A.R.T.
smart.disk.discovery	Retorna uma lista de dispositivos S.M.A.R.T.	
smart.disk.get	Retorna todas as propriedades disponíveis dos dispositivos S.M.A.R.T.	
systemd.unit.get	Retorna todas as propriedades de uma unidade systemd.	Systemd
systemd.unit.info	Informações da unidade systemd.	
systemd.unit.discovery	A lista de unidades systemd e seus detalhes.	
web.certificate.get	Valida os certificados e retorna detalhes do certificado.	Certificados Web

Veja também:

- Plugins embutidos
- Plugins carregáveis

Detalhes da chave do item

Parâmetros sem colchetes angulares são obrigatórios. Parâmetros marcados com colchetes angulares **< >** são opcionais.

ceph.df.details[connString,<user>,<apikey>]

 O uso de dados do cluster e a distribuição entre pools.
 Valor de retorno: *objeto JSON*.

Parâmetros:

- **connString** - o URI ou nome da sessão;
- **user, apikey** - o nome de usuário e a chave de API para conectar ao Ceph.

ceph.osd.stats[connString,<user>,<apikey>]

 Estatísticas agregadas e por OSD.
 Valor de retorno: *objeto JSON*.

Parâmetros:

- **connString** - o URI ou nome da sessão;
- **user, apikey** - o nome de usuário e a chave de API para conectar ao Ceph.

ceph.osd.discovery[connString,<user>,<apikey>]

 A lista de OSDs descobertos. Usado para descoberta de baixo nível.
 Valor de retorno: *objeto JSON*.

Parâmetros:

- **connString** - o URI ou nome da sessão;
- **user, apikey** - o nome de usuário e a chave de API para conectar ao Ceph.

ceph.osd.dump[connString,<user>,<apikey>]

 Os limites de uso e status dos OSDs.
 Valor de retorno: *objeto JSON*.

Parâmetros:

- **connString** - o URI ou nome da sessão;
- **user, apikey** - o nome de usuário e a chave de API para conectar ao Ceph.

ceph.ping[connString,<user>,<apikey>]

 Testa se uma conexão com o Ceph pode ser estabelecida.
 Valor de retorno: *0* - a conexão está quebrada (se houver qualquer erro apresentado, incluindo problemas de AUTENTICAÇÃO e configuração); *1* - a conexão foi bem-sucedida.

Parâmetros:

- **connString** - o URI ou nome da sessão;
- **user, apikey** - o nome de usuário e a chave de API para conectar ao Ceph.

ceph.pool.discovery[connString,<user>,<apikey>]

 A lista de pools descobertos. Usado para descoberta de baixo nível.
 Valor de retorno: *objeto JSON*.

Parâmetros:

- **connString** - o URI ou nome da sessão;
- **user, apikey** - o nome de usuário e a chave de API para conectar ao Ceph.

ceph.status[connString,<user>,<apikey>]

 O status geral do cluster.
 Valor de retorno: *objeto JSON*.

Parâmetros:

- **connString** - o URI ou nome da sessão;
- **user, apikey** - o nome de usuário e a chave de API para conectar ao Ceph.

docker.container_info[<ID>,<info>]

 Informações de baixo nível sobre um container.
 Valor de retorno: A saída da chamada de API ContainerInspect serializada como JSON.

Parâmetros:

- **ID** - o ID ou nome do container;
- **info** - a quantidade de informações retornadas. Valores suportados: *short* (padrão) ou *full*.

O usuário do Agent 2 (’zabbix’) deve ser adicionado ao grupo ’docker’ para privilégios suficientes. Caso contrário, a verificação falhará.

docker.container_stats[<ID>]

 As estatísticas de uso de recursos do container.
 Valor de retorno: A saída da chamada de API ContainerStats e a porcentagem de uso da CPU serializada como JSON.

Parâmetro:

- **ID** - o ID ou nome do container.

O usuário do Agent 2 (’zabbix’) deve ser adicionado ao grupo ’docker’ para privilégios suficientes. Caso contrário, a verificação falhará.

docker.containers

 A lista de containers.
 Valor de retorno: A saída da chamada da API ContainerList serializada como JSON.

O usuário do Agent 2 (’zabbix’) deve ser adicionado ao grupo ’docker’ para privilégios suficientes. Caso contrário, a verificação falhará.

docker.containers.discovery[<options>]

 Retorna a lista de containers. Usado para descoberta de baixo nível.
 Valor de retorno: *objeto JSON*.

Parâmetro:

- **options** - especifica se todos os containers ou apenas os em execução devem ser descobertos. Valores suportados: *true* - retorna todos os containers; *false* - retorna apenas os containers em execução (padrão).

O usuário do Agent 2 (’zabbix’) deve ser adicionado ao grupo ’docker’ para privilégios suficientes. Caso contrário, a verificação falhará.

docker.data.usage

 Informações sobre o uso atual de dados.
 Valor de retorno: A saída da chamada de API SystemDataUsage serializada como JSON.

O usuário do Agent 2 (’zabbix’) deve ser adicionado ao grupo ’docker’ para privilégios suficientes. Caso contrário, a verificação falhará.

docker.images

 Retorna a lista de imagens.
 Valor de retorno: A saída da chamada de API ImageList serializada como JSON.

O usuário do Agent 2 (’zabbix’) deve ser adicionado ao grupo ’docker’ para privilégios suficientes. Caso contrário, a verificação falhará.

docker.images.discovery

 Retorna a lista de imagens. Usado para descoberta de baixo nível.
 Valor de retorno: *objeto JSON*.

O usuário do Agent 2 (’zabbix’) deve ser adicionado ao grupo ’docker’ para privilégios suficientes. Caso contrário, a verificação falhará.

docker.info

 As informações do sistema.
 Valor de retorno: A saída da chamada de API SystemInfo serializada como JSON.

O usuário do Agent 2 (’zabbix’) deve ser adicionado ao grupo ’docker’ para privilégios suficientes. Caso contrário, a verificação falhará.

docker.ping

 Testa se o daemon do Docker está ativo ou não.
 Valor de retorno: *1* - a conexão está ativa; *0* - a conexão está quebrada.

O usuário do Agent 2 (’zabbix’) deve ser adicionado ao grupo ’docker’ para privilégios suficientes. Caso contrário, a verificação falhará.

ember.get[<uri>,<path>]

 Retorna o resultado do dispositivo solicitado.
 Valor de retorno: *objeto JSON*.

Parâmetros:

- **uri** - URI do dispositivo Ember+. Padrão: 127.0.0.1:9998;
- **path** - Caminho OID para o dispositivo. Vazio por padrão, retorna os dados da coleção raiz.

memcached.ping[connString,<user>,<password>]

 Testa se uma conexão está ativa ou não.
 Valor de retorno: *1* - a conexão está ativa; *0* - a conexão está quebrada (se houver qualquer erro apresentado, incluindo AUTH e problemas de configuração).

Parâmetros:

- **connString** - o URI ou nome da sessão;
- **user, password** - as credenciais de login do Memcached.

memcached.stats[connString,<user>,<password>,<type>]

 Obtém a saída do comando STATS.
 Valor de retorno: *JSON* - a saída é serializada como JSON.

Parâmetros:

- **connString** - o URI ou nome da sessão;
- **user, password** - as credenciais de login do Memcached;
- **type** - tipo de estatística a ser retornada: *items*, *sizes*, *slabs* ou *settings* (vazio por padrão, retorna estatísticas gerais).

mongodb.collection.stats[connString,<user>,<password>,<database>,collection]

 Retorna várias estatísticas de armazenamento para uma determinada coleção.
 Valor de retorno: *objeto JSON*.

Parâmetros:

- **connString** - o URI ou nome da sessão;
- **user, password** - as credenciais de login do MongoDB;
- **database** - o nome do banco de dados (padrão: admin);
- **collection** - o nome da coleção.

mongodb.collections.discovery[connString,<user>,<password>]

 Retorna uma lista de coleções descobertas. Usado para descoberta de baixo nível.
 Valor de retorno: *objeto JSON*.

Parâmetros:

- **connString** - o URI ou nome da sessão;
- **user, password** - as credenciais de login do MongoDB.

mongodb.collections.usage[connString,<user>,<password>]

 Retorna as estatísticas de uso para as coleções.
 Valor de retorno: *objeto JSON*.

Parâmetros:

- **connString** - o URI ou nome da sessão;
- **user, password** - as credenciais de login do MongoDB.

mongodb.connpool.stats[connString,<user>,<password>]

 Retorna informações sobre as conexões de saída abertas da instância de banco de dados atual para outros membros do cluster sharded ou do conjunto de réplicas.
 Valor de retorno: *objeto JSON*.

Parâmetros:

- **connString** - o URI ou nome da sessão;
- **user, password** - as credenciais de login do MongoDB.

mongodb.db.stats[connString,<user>,<password>,<database>]

 Retorna as estatísticas que refletem o estado do sistema do banco de dados especificado.
 Valor de retorno: *objeto JSON*.

Parâmetros:

- **connString** - o URI ou nome da sessão;
- **user, password** - as credenciais de login do MongoDB;
- **database** - o nome do banco de dados (padrão: admin).

mongodb.db.discovery[connString,<user>,<password>]

 Retorna uma lista de bancos de dados descobertos. Usado para descoberta de baixo nível.
 Valor de retorno: *objeto JSON*.

Parâmetros:

- **connString** - o URI ou nome da sessão;
- **user, password** - as credenciais de login do MongoDB.

mongodb.jumbo_chunks.count[connString,<user>,<password>]

 Retorna a contagem de jumbo chunks.
 Valor de retorno: *objeto JSON*.

Parâmetros:

- **connString** - o URI ou nome da sessão;
- **user, password** - as credenciais de login do MongoDB.

mongodb.oplog.stats[connString,<user>,<password>]

 Retorna o status do conjunto de réplicas, usando dados coletados do oplog.
 Valor de retorno: *objeto JSON*.

Parâmetros:

- **connString** - o URI ou nome da sessão;
- **user, password** - as credenciais de login do MongoDB.

mongodb.ping[connString,<user>,<password>]

 Testa se uma conexão está ativa ou não.
 Valor de retorno: *1* - a conexão está ativa; *0* - a conexão está quebrada (se houver qualquer erro apresentado, incluindo problemas de AUTENTICAÇÃO e configuração).

Parâmetros:

- **connString** - o URI ou nome da sessão;
- **user, password** - as credenciais de login do MongoDB.

mongodb.rs.config[connString,<user>,<password>]

 Retorna a configuração atual do conjunto de réplicas.
 Valor de retorno: *objeto JSON*.

Parâmetros:

- **connString** - o URI ou nome da sessão;
- **user, password** - as credenciais de login do MongoDB.

mongodb.rs.status[connString,<user>,<password>]

 Retorna o status do replica set do ponto de vista do membro onde o método é executado.
 Valor de retorno: *objeto JSON*.

Parâmetros:

- **connString** - o URI ou nome da sessão;
- **user, password** - as credenciais de login do MongoDB.

mongodb.server.status[connString,<user>,<password>]

 Retorna o estado do banco de dados.
 Valor de retorno: *objeto JSON*.

Parâmetros:

- **connString** - o URI ou nome da sessão;
- **user, password** - as credenciais de login do MongoDB.

mongodb.sh.discovery[connString,<user>,<password>]

 Retorna a lista de shards descobertos presentes no cluster.
 Valor de retorno: *objeto JSON*.

Parâmetros:

- **connString** - o URI ou nome da sessão;
- **user, password** - as credenciais de login do MongoDB.

mongodb.version[connString,<user>,<password>]

 Retorna a versão do servidor de banco de dados.
 Valor de retorno: *String*.

Parâmetros:

- **connString** - o URI ou nome da sessão;
- **user, password** - as credenciais de login do MongoDB.

mqtt.get[<broker url>,topic,<user>,<password>]

 Assina um tópico específico ou tópicos (com curingas) do broker fornecido e aguarda publicações.
 Valor de retorno: Dependendo do conteúdo do tópico. Se curingas forem usados, retorna o conteúdo do tópico como JSON.

Parâmetros:

- **broker url** - a URL do broker MQTT no formato `protocol://host:port` sem parâmetros de consulta (protocolos suportados: `tcp`, `ssl`, `ws`). Se nenhum valor for especificado, o agent usará `tcp://localhost:1883`. Se um protocolo ou porta forem omitidos, o protocolo padrão (`tcp`) ou porta (1883) serão usados;
- **topic** - o tópico MQTT (obrigatório). Curingas (+,#) são suportados;
- **user, password** - as credenciais de autenticação (se necessário).

Comentários:

- O item deve ser configurado como uma verificação ativa (tipo de item 'Zabbix agent (active)');
- Certificados de criptografia TLS podem ser usados salvando-os em um local padrão (por exemplo, diretório `/etc/ssl/certs/` para Ubuntu). Para TLS, use o esquema `tls://`.

mssql.availability.group.get[URI,<user>,<password>]

 Retorna grupos de disponibilidade.
 Valor de retorno: *objeto JSON*.

Parâmetros:

- **URI** - URI do servidor MSSQL (o único esquema suportado é `sqlserver://`). Credenciais embutidas serão ignoradas. Desde o Zabbix 7.0.6, é possível especificar um nome de instância como parte da URI, ou seja: `sqlserver://localhost/InstanceName` (sem porta). Se a porta for especificada, o nome da instância será ignorado;
- **user, password** - nome de usuário, senha para enviar ao servidor MSSQL protegido.

Para mais informações, consulte o readme do plugin MSSQL.

mssql.custom.query[URI,<user>,<password>,queryName,<args...>]

 Retorna o resultado de uma consulta personalizada.
 Valor de retorno: *objeto JSON*.

Parâmetros:

- **URI** - URI do servidor MSSQL (o único esquema suportado é `sqlserver://`). Credenciais embutidas serão ignoradas. Desde o Zabbix 7.0.6, é possível especificar um nome de instância como parte da URI, ou seja: `sqlserver://localhost/InstanceName` (sem porta). Se a porta for especificada, o nome da instância será ignorado;
- **user, password** - nome de usuário, senha para enviar ao servidor MSSQL protegido;
- **queryName** - nome de uma consulta personalizada configurada em `Plugins.MSSQL.CustomQueriesDir` sem a extensão .sql;
- **args** - um ou vários argumentos separados por vírgula para passar para a consulta.

Para mais informações, consulte o readme do plugin MSSQL.

mssql.db.get

 Retorna todos os bancos de dados MSSQL disponíveis.
 Valor de retorno: *objeto JSON*.

Para mais informações, consulte o readme do plugin MSSQL.

mssql.job.status.get

 Retorna o status dos jobs.
 Valor de retorno: *objeto JSON*.

Para mais informações, consulte o readme do plugin MSSQL.

mssql.last.backup.get

 Retorna o horário do último backup de todos os bancos de dados.
 Valor de retorno: *objeto JSON*.

Para mais informações, consulte o readme do plugin MSSQL.

mssql.local.db.get

 Retorna bancos de dados que estão participando de um grupo de disponibilidade Always On e réplica (primária ou secundária) e estão localizados no servidor ao qual a conexão foi estabelecida.
 Valor de retorno: *objeto JSON*.

Para mais informações, consulte o readme do plugin MSSQL.

mssql.mirroring.get

 Retorna informações de espelhamento.
 Valor de retorno: *objeto JSON*.

Para mais informações, consulte o readme do plugin MSSQL.

mssql.nonlocal.db.get

 Retorna bancos de dados que estão participando de um grupo de disponibilidade Always On e réplica (primária ou secundária) localizada em outros servidores (o banco de dados não é local para a instância do SQL Server à qual a conexão foi estabelecida).
 Valor de retorno: *objeto JSON*.

Para mais informações, consulte o readme do plugin MSSQL.

mssql.perfcounter.get

 Retorna os contadores de desempenho.
 Valor de retorno: *objeto JSON*.

Para mais informações, consulte o readme do plugin MSSQL.

mssql.ping

 Ping no banco de dados. Teste se a conexão está corretamente configurada.
 Valor de retorno: *1* - ativo, *0* - inativo.

Para mais informações, consulte o readme do plugin MSSQL.

mssql.quorum.get

 Retorna as informações do quorum.
 Valor de retorno: *objeto JSON*.

Para mais informações, consulte o readme do plugin MSSQL.

mssql.quorum.member.get

 Retorna os membros do quorum.
 Valor de retorno: *objeto JSON*.

Para mais informações, consulte o readme do plugin MSSQL.

mssql.replica.get

 Retorna as réplicas.
 Valor de retorno: *objeto JSON*.

Para mais informações, consulte o readme do plugin MSSQL.

mssql.version

 Retorna a versão do MSSQL.
 Valor de retorno: *String*.

Para mais informações, consulte o readme do plugin MSSQL.

mysql.custom.query[connString,<user>,<password>,queryName,<args...>]

 Retorna o resultado de uma consulta personalizada.
 Valor de retorno: *objeto JSON*.

Parâmetros:

- **connString** - URI ou nome da sessão;
- **user, password** - credenciais de login do MySQL;
- **queryName** - nome de uma consulta personalizada, deve corresponder ao nome do arquivo SQL sem extensão;
- **args** - um ou vários argumentos separados por vírgula para passar para a consulta.

Para mais informações, consulte o readme do plugin MySQL, seção *Consultas personalizadas*.

mysql.db.discovery[connString,<user>,<password>]

 Retorna a lista de bancos de dados MySQL. Usado para descoberta de baixo nível.
 Valor de retorno: O resultado da consulta SQL ”show databases” no formato JSON LLD.

Parâmetros:

- **connString** - o URI ou nome da sessão;
- **user, password** - as credenciais de login do MySQL.

mysql.db.size[connString,<user>,<password>,<database name>]

 O tamanho do banco de dados em bytes.
 Valor de retorno: Resultado da consulta SQL ”select coalesce(sum(data_length + index_length),0) as size from information_schema.tables where table_schema=?” para um banco de dados específico em bytes.

Parâmetros:

- **connString** - o URI ou nome da sessão;
- **user, password** - as credenciais de login do MySQL;
- **database name** - o nome do banco de dados.

mysql.get_status_variables[connString,<user>,<password>]

 Valores das variáveis de status globais.
 Valor de retorno: Resultado da consulta SQL ”show global status” em formato JSON.

Parâmetros:

- **connString** - o URI ou nome da sessão;
- **user, password** - as credenciais de login do MySQL.

mysql.ping[connString,<user>,<password>]

 Testa se uma conexão está ativa ou não.
 Valor de retorno: *1* - a conexão está ativa; *0* - a conexão está quebrada (se houver qualquer erro apresentado, incluindo problemas de AUTENTICAÇÃO e configuração).

Parâmetros:

- **connString** - o URI ou nome da sessão;
- **user, password** - as credenciais de login do MySQL.

mysql.replication.discovery[connString,<user>,<password>]

 Retorna a lista de replicações do MySQL. Usado para descoberta de baixo nível.
 Valor de retorno: o resultado da consulta SQL `SHOW SLAVE STATUS` ou `SHOW REPLICA STATUS` no formato JSON LLD.

Parâmetros:

- **connString** - o URI ou nome da sessão;
- **user, password** - as credenciais de login do MySQL.

mysql.replication.get_slave_status[connString,<user>,<password>,<master host>]

 O status da replicação.
 Valor de retorno: resultado da consulta SQL `SHOW SLAVE STATUS` ou `SHOW REPLICA STATUS` em formato JSON.

Parâmetros:

- **connString** - o URI ou nome da sessão;
- **user, password** - as credenciais de login do MySQL;
- **master host** - o nome do host mestre de replicação. Se nenhum for encontrado, um erro é retornado. Se este parâmetro não for especificado, todos os hosts serão retornados.

mysql.version[connString,<user>,<password>]

 A versão do MySQL.
 Valor de retorno: *String* (com a versão da instância do MySQL).

Parâmetros:

- **connString** - o URI ou nome da sessão;
- **user, password** - as credenciais de login do MySQL.

net.dns.get[<ip>,name,<type>,<timeout>,<count>,<protocol>,"<flags>"]

Executa uma consulta DNS e retorna informações detalhadas do registro DNS.
 Este item é uma versão estendida do item de agent Zabbix `net.dns.record` com mais tipos de registros e flags personalizáveis suportadas.
 Valores de retorno: *objeto JSON*

Parâmetros:

- **ip** - o endereço IP do servidor DNS (deixe em branco para o servidor DNS padrão);
- **name** - o nome DNS a ser consultado;
- **type** - o tipo de registro a ser consultado (o padrão é *SOA*);
- **timeout** - o tempo limite para a solicitação em segundos (o padrão é 1 segundo);
- **count** - o número de tentativas para a solicitação (o padrão é 2);
- **protocol** - o protocolo usado para executar consultas DNS: *udp* (padrão) ou *tcp*;
- **flags** - um ou mais argumentos separados por vírgula para passar para uma consulta.

Comentários:

- Os valores possíveis para `type` são: *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*. Observe que os valores devem estar apenas em maiúsculas; valores em minúsculas ou mistos não são suportados.
- Para pesquisas de DNS reverso (quando `type` é definido como *PTR*), você pode fornecer o nome DNS tanto no formato invertido quanto no não invertido (veja exemplos abaixo). Observe que quando o registro PTR é solicitado, o nome DNS é na verdade um endereço IP.

- Os valores possíveis para `flags` são: *cdflag* ou *nocdflag* (padrão), *rdflag* (padrão) ou *nordflag*, *dnssec* ou *nodnssec* (padrão), *nsid* ou *nonsid* (padrão), *edns0* (padrão) ou *noedns0*, *aaflag* ou *noaaflag* (padrão), *adflag* ou *noadflag* (padrão). As `flags` *dnssec* e *nsid* não podem ser usadas junto com *noedns0*, pois ambas requerem *edns0*. Observe que os valores devem estar apenas em minúsculas; valores em maiúsculas ou mistos não são suportados.
- Nomes de domínio internacionalizados não são suportados, use nomes codificados em IDNA.
- A saída é um objeto contendo informações do registro DNS com base nos parâmetros fornecidos (veja mais detalhes).

Exemplos:

```
net.dns.get[192.0.2.0,example.com,DNSKEY,3,3,tcp,"cdflag,rdflag,nsid"]

net.dns.get[,198.51.100.1,PTR,,,,"cdflag,rdflag,nsid"]
net.dns.get[,1.100.51.198.in-addr.arpa,PTR,,,,"cdflag,rdflag,nsid"]

net.dns.get[,2a00:1450:400f:800::200e,PTR,,,,"cdflag,rdflag,nsid"]
net.dns.get[,e.0.0.2.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.0.f.0.0.4.0.5.4.1.0.0.a.2.ip6.arpa,PTR,,,,"cdflag,rdfla
```

oracle.diskgroups.stats[connString,<user>,<password>,<service>,<diskgroup>]

 Retorna as estatísticas dos grupos de disco do Automatic Storage Management (ASM).
 Valor de retorno: *objeto JSON*.

Parâmetros:

- **connString** - pode ser:
  - o URI - se nenhuma sessão com o nome fornecido for encontrada e `Plugins.Oracle.ResolveTNS` estiver definido como false. Se contiver um esquema (por exemplo, ”tcp://”), porta (por exemplo, 1521) ou ambos, a opção ResolveTNS não será considerada e, de qualquer forma, será tratada como URI;
  - nome da sessão - se tal nome for encontrado no arquivo de configuração do plugin;
  - chave TNS (suportada desde o Zabbix 7.0.19) - se a opção ResolveTNS estiver definida como true e nenhuma das condições acima se aplicar;
  - valor TNS (suportado desde o Zabbix 7.0.19) - se começar com o parêntese aberto “(“ (espaços à esquerda ignorados). Neste caso, a opção ResolveTNS não é considerada.
- **user** - o nome de usuário do Oracle, suporta a adição de uma das opções de login `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` ou `as sysrac` no formato `user as sysdba` (uma opção de login não diferencia maiúsculas de minúsculas e não deve conter um espaço à direita). Observe que apenas as opções de login `as sysdba`, `as sysoper` ou `as sysasm` são suportadas antes do Zabbix 7.0.18;
- **password** - a senha do Oracle;
- **service** - o nome do serviço Oracle;
- **diskgroup** - o nome do grupo de disco ASM a ser consultado.

oracle.diskgroups.discovery[connString,<user>,<password>,<service>]

 Retorna a lista de grupos de disco ASM. Usado para descoberta de baixo nível.
 Valor de retorno: *objeto JSON*.

Parâmetros:

- **connString** - pode ser:
  - o URI - se nenhuma sessão com o nome fornecido for encontrada e `Plugins.Oracle.ResolveTNS` estiver definido como false. Se contiver um esquema (por exemplo, ”tcp://”), porta (por exemplo, 1521) ou ambos, a opção ResolveTNS não será considerada e, de qualquer forma, será tratada como URI;
  - nome da sessão - se tal nome for encontrado no arquivo de configuração do plugin;
  - chave TNS (suportada desde o Zabbix 7.0.19) - se a opção ResolveTNS estiver definida como true e nenhuma das condições acima se aplicar;
  - valor TNS (suportado desde o Zabbix 7.0.19) - se começar com o parêntese aberto “(“ (espaços à esquerda ignorados). Neste caso, a opção ResolveTNS não é considerada.
- **user** - o nome de usuário do Oracle, suporta anexar uma das opções de login `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` ou `as sysrac` no formato `user as sysdba` (uma opção de login não diferencia maiúsculas de minúsculas e não deve conter um espaço à direita). Observe que apenas as opções de login `as sysdba`, `as sysoper` ou `as sysasm` são suportadas antes do Zabbix 7.0.18;
- **password** - a senha do Oracle;
- **service** - o nome do serviço Oracle.

oracle.archive.info[connString,<user>,<password>,<service>,<destination>]

 As estatísticas dos logs de arquivamento.
 Valor de retorno: *objeto JSON*.

Parâmetros:

- **connString** - pode ser:

    - o URI - se nenhuma sessão com o nome fornecido for encontrada e `Plugins.Oracle.ResolveTNS` estiver definido como false. Se contiver um esquema (por exemplo, "tcp://"), porta (por exemplo, 1521) ou ambos, a opção ResolveTNS não será considerada e, de qualquer forma, será tratada como URI;
    - nome da sessão - se tal nome for encontrado no arquivo de configuração do plugin;
    - chave TNS (suportado desde o Zabbix 7.0.19) - se a opção ResolveTNS estiver definida como true e nenhuma das condições acima se aplicar;
    - valor TNS (suportado desde o Zabbix 7.0.19) - se começar com o parêntese aberto "(" (espaços à esquerda ignorados). Neste caso, a opção ResolveTNS não é considerada.
- **user** - o nome de usuário do Oracle, suporta a adição de uma das opções de login `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` ou `as sysrac` no formato `user as sysdba` (uma opção de login não diferencia maiúsculas de minúsculas, não deve conter um espaço à direita). Observe que apenas as opções de login `as sysdba`, `as sysoper` ou `as sysasm` são suportadas antes do Zabbix 7.0.18;
- **password** - a senha do Oracle;
- **service** - o nome do serviço Oracle;
- **destination** - o nome do destino a ser consultado.

oracle.cdb.info[connString,<user>,<password>,<service>,<database>]

 Informações dos Container Databases (CDBs).
 Valor de retorno: *objeto JSON*.

Parâmetros:

- **connString** - pode ser:
    - o URI - se nenhuma sessão com o nome fornecido for encontrada e `Plugins.Oracle.ResolveTNS` estiver definido como false. Se contiver um esquema (por exemplo, "tcp://"), porta (por exemplo, 1521) ou ambos, a opção ResolveTNS não será considerada e, de qualquer forma, será tratada como URI;
    - nome da sessão - se tal nome for encontrado no arquivo de configuração do plugin;
    - chave TNS (suportado desde o Zabbix 7.0.19) - se a opção ResolveTNS estiver definida como true e nenhuma das condições acima se aplicar;
    - valor TNS (suportado desde o Zabbix 7.0.19) - se começar com o parêntese aberto "(" (espaços à esquerda ignorados). Neste caso, a opção ResolveTNS não é considerada.
- **user** - o nome de usuário do Oracle, suporta a adição de uma das opções de login `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` ou `as sysrac` no formato `user as sysdba` (uma opção de login não diferencia maiúsculas de minúsculas e não deve conter um espaço à direita). Observe que apenas as opções de login `as sysdba`, `as sysoper` ou `as sysasm` são suportadas antes do Zabbix 7.0.18;
- **password** - a senha do Oracle;
- **service** - o nome do serviço Oracle;
- **database** - o nome do banco de dados a ser consultado.

oracle.custom.query[connString,<user>,<password>,<service>,queryName,<args...>]

 O resultado de uma consulta personalizada.
 Valor de retorno: *objeto JSON*.

Parâmetros:

- **connString** - pode ser:
    - o URI - se nenhuma sessão com o nome fornecido for encontrada e `Plugins.Oracle.ResolveTNS` estiver definido como false. Se contiver um esquema (por exemplo, "tcp://"), porta (por exemplo, 1521) ou ambos, a opção ResolveTNS não será considerada e, de qualquer forma, será tratada como URI;
    - nome da sessão - se tal nome for encontrado no arquivo de configuração do plugin;
    - chave TNS (suportado desde o Zabbix 7.0.19) - se a opção ResolveTNS estiver definida como true e nenhuma das condições acima se aplicar;
    - valor TNS (suportado desde o Zabbix 7.0.19) - se começar com o parêntese aberto "(" (espaços à esquerda ignorados). Neste caso, a opção ResolveTNS não é considerada.
- **user** - o nome de usuário do Oracle, suporta a adição de uma das opções de login `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` ou `as sysrac` no formato `user as sysdba` (uma opção de login não diferencia maiúsculas de minúsculas e não deve conter um espaço à direita). Observe que apenas as opções de login `as sysdba`, `as sysoper` ou `as sysasm` são suportadas antes do Zabbix 7.0.18;
- **password** - a senha do Oracle;
- **service** - o nome do serviço Oracle;
- **queryName** - o nome de uma consulta personalizada, deve corresponder ao nome do arquivo SQL sem extensão;
- **args** - um ou vários argumentos separados por vírgula para passar para a consulta.

Comentários: - Os dados retornados são convertidos automaticamente em JSON. - Evite retornar JSON diretamente das consultas, pois ele será corrompido quando o plugin tentar convertê-lo novamente em JSON.

oracle.datafiles.stats[connString,<user>,<password>,<service>]

 Retorna as estatísticas dos arquivos de dados.
 Valor de retorno: *objeto JSON*.

Parâmetros:

- **connString** - pode ser:
  - o URI - se nenhuma sessão com o nome fornecido for encontrada e `Plugins.Oracle.ResolveTNS` estiver definido como false. Se contiver um esquema (por exemplo, ”tcp://”), porta (por exemplo, 1521) ou ambos, a opção ResolveTNS não será considerada e, de qualquer forma, será tratada como URI;
  - nome da sessão - se tal nome for encontrado no arquivo de configuração do plugin;
  - chave TNS (suportada desde o Zabbix 7.0.19) - se a opção ResolveTNS estiver definida como true e nenhuma das condições acima se aplicar;
  - valor TNS (suportado desde o Zabbix 7.0.19) - se começar com o parêntese aberto “(“ (espaços à esquerda ignorados). Neste caso, a opção ResolveTNS não é considerada.
- **user** - o nome de usuário do Oracle, suporta a adição de uma das opções de login `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` ou `as sysrac` no formato `user as sysdba` (uma opção de login não diferencia maiúsculas de minúsculas e não deve conter um espaço à direita). Observe que apenas as opções de login `as sysdba`, `as sysoper` ou `as sysasm` são suportadas antes do Zabbix 7.0.18;
- **password** - a senha do Oracle;
- **service** - o nome do serviço Oracle.

oracle.db.discovery[connString,<user>,<password>,<service>]

 Retorna a lista de bancos de dados. Usado para descoberta de baixo nível.
 Valor de retorno: *objeto JSON*.

Parâmetros:

- **connString** - pode ser:
  - o URI - se nenhuma sessão com o nome fornecido for encontrada e `Plugins.Oracle.ResolveTNS` estiver definido como false. Se contiver um esquema (por exemplo, ”tcp://”), porta (por exemplo, 1521) ou ambos, a opção ResolveTNS não será considerada e, de qualquer forma, será tratada como URI;
  - nome da sessão - se tal nome for encontrado no arquivo de configuração do plugin;
  - chave TNS (suportado desde o Zabbix 7.0.19) - se a opção ResolveTNS estiver definida como true e nenhuma das condições acima se aplicar;
  - valor TNS (suportado desde o Zabbix 7.0.19) - se começar com o parêntese de abertura “(“ (espaços à esquerda ignorados). Neste caso, a opção ResolveTNS não é considerada.
- **user** - o nome de usuário do Oracle, suporta a adição de uma das opções de login `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` ou `as sysrac` no formato `user as sysdba` (uma opção de login não diferencia maiúsculas de minúsculas e não deve conter um espaço à direita). Observe que apenas as opções de login `as sysdba`, `as sysoper` ou `as sysasm` são suportadas antes do Zabbix 7.0.18;
- **password** - a senha do Oracle;
- **service** - o nome do serviço Oracle.

oracle.fra.stats[connString,<user>,<password>,<service>]

 Retorna as estatísticas da Fast Recovery Area (FRA).
 Valor de retorno: *objeto JSON*.

Parâmetros:

- **connString** - pode ser:
  - o URI - se nenhuma sessão com o nome fornecido for encontrada e `Plugins.Oracle.ResolveTNS` estiver definido como false. Se contiver um esquema (por exemplo, ”tcp://”), porta (por exemplo, 1521) ou ambos, a opção ResolveTNS não será considerada e, de qualquer forma, será tratada como URI;
  - nome da sessão - se tal nome for encontrado no arquivo de configuração do plugin;
  - chave TNS (suportado desde o Zabbix 7.0.19) - se a opção ResolveTNS estiver definida como true e nenhuma das condições acima se aplicar;
  - valor TNS (suportado desde o Zabbix 7.0.19) - se começar com o parêntese aberto “(“ (espaços à esquerda ignorados). Neste caso, a opção ResolveTNS não é considerada.
- **user** - o nome de usuário do Oracle, suporta a adição de uma das opções de login `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` ou `as sysrac` no formato `user as sysdba` (uma opção de login não diferencia maiúsculas de minúsculas e não deve conter um espaço à direita). Observe que apenas as opções de login `as sysdba`, `as sysoper` ou `as sysasm` são suportadas antes do Zabbix 7.0.18;
- **password** - a senha do Oracle;
- **service** - o nome do serviço Oracle.

oracle.instance.info[connString,<user>,<password>,<service>]

 As estatísticas da instância.
 Valor de retorno: *objeto JSON*.

Parâmetros:

- **connString** - pode ser:
  - o URI - se nenhuma sessão com o nome fornecido for encontrada e `Plugins.Oracle.ResolveTNS` estiver definido como false. Se contiver um esquema (por exemplo, "tcp://"), porta (por exemplo, 1521) ou ambos, a opção ResolveTNS não será considerada e, de qualquer forma, será tratada como URI;
  - nome da sessão - se tal nome for encontrado no arquivo de configuração do plugin;
  - chave TNS (suportado desde o Zabbix 7.0.19) - se a opção ResolveTNS estiver definida como true e nenhuma das condições acima se aplicar;
  - valor TNS (suportado desde o Zabbix 7.0.19) - se começar com o parêntese aberto "(" (espaços à esquerda ignorados). Neste caso, a opção ResolveTNS não é considerada.
- **user** - o nome de usuário do Oracle, suporta a adição de uma das opções de login `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` ou `as sysrac` no formato `user as sysdba` (uma opção de login não diferencia maiúsculas de minúsculas e não deve conter um espaço à direita). Observe que apenas as opções de login `as sysdba`, `as sysoper` ou `as sysasm` são suportadas antes do Zabbix 7.0.18;
- **password** - a senha do Oracle;
- **service** - o nome do serviço Oracle.

oracle.pdb.info[connString,<user>,<password>,<service>,<database>]

 Informações sobre Pluggable Databases (PDBs).
 Valor de retorno: *objeto JSON*.

Parâmetros:

- **connString** - pode ser:
  - o URI - se nenhuma sessão com o nome fornecido for encontrada e `Plugins.Oracle.ResolveTNS` estiver definido como false. Se contiver um esquema (por exemplo, "tcp://"), porta (por exemplo, 1521) ou ambos, a opção ResolveTNS não será considerada e, de qualquer forma, será tratada como URI;
  - nome da sessão - se tal nome for encontrado no arquivo de configuração do plugin;
  - chave TNS (suportado desde o Zabbix 7.0.19) - se a opção ResolveTNS estiver definida como true e nenhuma das condições acima se aplicar;
  - valor TNS (suportado desde o Zabbix 7.0.19) - se começar com o parêntese aberto "(" (espaços à esquerda ignorados). Neste caso, a opção ResolveTNS não é considerada.
- **user** - o nome de usuário do Oracle, suporta a adição de uma das opções de login `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` ou `as sysrac` no formato `user as sysdba` (uma opção de login não diferencia maiúsculas de minúsculas e não deve conter um espaço à direita). Observe que apenas as opções de login `as sysdba`, `as sysoper` ou `as sysasm` são suportadas antes do Zabbix 7.0.18;
- **password** - a senha do Oracle;
- **service** - o nome do serviço do Oracle;
- **database** - o nome do banco de dados a ser consultado.

oracle.pdb.discovery[connString,<user>,<password>,<service>]

 Retorna a lista de PDBs. Usado para descoberta de baixo nível.
 Valor de retorno: *objeto JSON*.

Parâmetros:

- **connString** - pode ser:
  - o URI - se nenhuma sessão com o nome fornecido for encontrada e `Plugins.Oracle.ResolveTNS` estiver definido como false. Se contiver um esquema (por exemplo, "tcp://"), porta (por exemplo, 1521) ou ambos, a opção ResolveTNS não será considerada e, de qualquer forma, será tratada como URI;
  - nome da sessão - se tal nome for encontrado no arquivo de configuração do plugin;
  - chave TNS (suportado desde o Zabbix 7.0.19) - se a opção ResolveTNS estiver definida como true e nenhuma das condições acima se aplicar;
  - valor TNS (suportado desde o Zabbix 7.0.19) - se começar com o parêntese aberto "(" (espaços à esquerda ignorados). Neste caso, a opção ResolveTNS não é considerada.
- **user** - o nome de usuário do Oracle, suporta a adição de uma das opções de login `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` ou `as sysrac` no formato `user as sysdba` (uma opção de login não diferencia maiúsculas de minúsculas e não deve conter um espaço à direita). Observe que apenas as opções de login `as sysdba`, `as sysoper` ou `as sysasm` são suportadas antes do Zabbix 7.0.18;
- **password** - a senha do Oracle;
- **service** - o nome do serviço Oracle.

oracle.pga.stats[connString,<user>,<password>,<service>]

 Retorna as estatísticas da Program Global Area (PGA).
 Valor de retorno: *objeto JSON*.

Parâmetros:

- **connString** - pode ser:

    - o URI - se nenhuma sessão com o nome fornecido for encontrada e `Plugins.Oracle.ResolveTNS` estiver definido como false. Se contiver um esquema (por exemplo, "tcp://"), porta (por exemplo, 1521) ou ambos, a opção ResolveTNS não será considerada e, de qualquer forma, será tratada como URI;
    - nome da sessão - se tal nome for encontrado no arquivo de configuração do plugin;
    - chave TNS (suportado desde o Zabbix 7.0.19) - se a opção ResolveTNS estiver definida como true e nenhuma das condições acima se aplicar;
    - valor TNS (suportado desde o Zabbix 7.0.19) - se começar com o parêntese aberto "(" (espaços à esquerda ignorados). Neste caso, a opção ResolveTNS não é considerada.
- **user** - o nome de usuário do Oracle, suporta a adição de uma das opções de login `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` ou `as sysrac` no formato `user as sysdba` (uma opção de login não diferencia maiúsculas de minúsculas e não deve conter um espaço à direita). Observe que apenas as opções de login `as sysdba`, `as sysoper` ou `as sysasm` são suportadas antes do Zabbix 7.0.18;
- **password** - a senha do Oracle;
- **service** - o nome do serviço Oracle.

oracle.ping[connString,<user>,<password>,<service>]

 Testa se uma conexão com o Oracle pode ser estabelecida.
 Valor de retorno: *1* - a conexão foi bem-sucedida; *0* - a conexão falhou (se houver qualquer erro apresentado, incluindo problemas de AUTENTICAÇÃO e configuração).

Parâmetros:

- **connString** - pode ser:
    - o URI - se nenhuma sessão com o nome fornecido for encontrada e `Plugins.Oracle.ResolveTNS` estiver definido como false. Se contiver um esquema (por exemplo, "tcp://"), porta (por exemplo, 1521) ou ambos, a opção ResolveTNS não será considerada e será tratada como URI;
    - nome da sessão - se tal nome for encontrado no arquivo de configuração do plugin;
    - chave TNS (suportado desde o Zabbix 7.0.19) - se a opção ResolveTNS estiver definida como true e nenhuma das condições acima se aplicar;
    - valor TNS (suportado desde o Zabbix 7.0.19) - se começar com o parêntese aberto "(" (espaços à esquerda são ignorados). Neste caso, a opção ResolveTNS não é considerada.
- **user** - o nome de usuário do Oracle, suporta a adição de uma das opções de login `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` ou `as sysrac` no formato `user as sysdba` (a opção de login não diferencia maiúsculas de minúsculas e não deve conter espaço à direita). Observe que apenas as opções de login `as sysdba`, `as sysoper` ou `as sysasm` são suportadas antes do Zabbix 7.0.18;
- **password** - a senha do Oracle;
- **service** - o nome do serviço Oracle.

Exemplos:

```
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>]

 Retorna as estatísticas de processos.
 Valor de retorno: *objeto JSON*.

Parâmetros:

- **connString** - pode ser:
    - o URI - se nenhuma sessão com o nome fornecido for encontrada e `Plugins.Oracle.ResolveTNS` estiver definido como false. Se contiver um esquema (por exemplo, "tcp://"), porta (por exemplo, 1521) ou ambos, a opção ResolveTNS não será considerada e, de qualquer forma, será tratada como URI;
    - nome da sessão - se tal nome for encontrado no arquivo de configuração do plugin;
    - chave TNS (suportado desde o Zabbix 7.0.19) - se a opção ResolveTNS estiver definida como true e nenhuma das condições acima se aplicar;
    - valor TNS (suportado desde o Zabbix 7.0.19) - se começar com o parêntese aberto "(" (espaços à esquerda são ignorados). Neste caso, a opção ResolveTNS não é considerada.
- **user** - o nome de usuário do Oracle, suporta a adição de uma das opções de login `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` ou `as sysrac` no formato `user as sysdba` (uma opção de login não diferencia maiúsculas de minúsculas e não deve conter um espaço à direita). Observe que apenas as opções de login `as sysdba`, `as sysoper` ou `as sysasm` são suportadas antes do Zabbix 7.0.18;
- **password** - a senha do Oracle;
- **service** - o nome do serviço Oracle.

oracle.redolog.info[connString,<user>,<password>,<service>]

 As informações do arquivo de log do arquivo de controle.
 Valor de retorno: *objeto JSON*.

Parâmetros:

- **connString** - pode ser:
  - o URI - se nenhuma sessão com o nome fornecido for encontrada e `Plugins.Oracle.ResolveTNS` estiver definido como false. Se contiver um esquema (por exemplo, ”tcp://”), porta (por exemplo, 1521) ou ambos, a opção ResolveTNS não será considerada e, de qualquer forma, será tratada como URI;
  - nome da sessão - se tal nome for encontrado no arquivo de configuração do plugin;
  - chave TNS (suportado desde o Zabbix 7.0.19) - se a opção ResolveTNS estiver definida como true e nenhuma das condições acima se aplicar;
  - valor TNS (suportado desde o Zabbix 7.0.19) - se começar com o parêntese aberto “(“ (espaços à esquerda ignorados). Neste caso, a opção ResolveTNS não é considerada.
- **user** - o nome de usuário do Oracle, suporta a adição de uma das opções de login `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` ou `as sysrac` no formato `user as sysdba` (uma opção de login não diferencia maiúsculas de minúsculas e não deve conter um espaço à direita). Observe que apenas as opções de login `as sysdba`, `as sysoper` ou `as sysasm` são suportadas antes do Zabbix 7.0.18;
- **password** - a senha do Oracle;
- **service** - o nome do serviço Oracle.

oracle.sga.stats[connString,<user>,<password>,<service>]

 Retorna as estatísticas da System Global Area (SGA).
 Valor de retorno: *objeto JSON*.

Parâmetros:

- **connString** - pode ser:
  - o URI - se nenhuma sessão com o nome fornecido for encontrada e `Plugins.Oracle.ResolveTNS` estiver definido como false. Se contiver um esquema (por exemplo, ”tcp://”), porta (por exemplo, 1521) ou ambos, a opção ResolveTNS não será considerada e, de qualquer forma, será tratada como URI;
  - nome da sessão - se tal nome for encontrado no arquivo de configuração do plugin;
  - chave TNS (suportada desde o Zabbix 7.0.19) - se a opção ResolveTNS estiver definida como true e nenhuma das condições acima se aplicar;
  - valor TNS (suportado desde o Zabbix 7.0.19) - se começar com o parêntese de abertura “(“ (espaços à esquerda ignorados). Neste caso, a opção ResolveTNS não é considerada.
- **user** - o nome de usuário do Oracle, suporta a adição de uma das opções de login `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` ou `as sysrac` no formato `user as sysdba` (uma opção de login não diferencia maiúsculas de minúsculas e não deve conter um espaço à direita). Observe que apenas as opções de login `as sysdba`, `as sysoper` ou `as sysasm` são suportadas antes do Zabbix 7.0.18;
- **password** - a senha do Oracle;
- **service** - o nome do serviço Oracle.

oracle.sessions.stats[connString,<user>,<password>,<service>,<lockMaxTime>]

 Retorna as estatísticas das sessões.
 Valor de retorno: *objeto JSON*.

Parâmetros:

- **connString** - pode ser:
  - o URI - se nenhuma sessão com o nome fornecido for encontrada e `Plugins.Oracle.ResolveTNS` estiver definido como false. Se contiver um esquema (por exemplo, ”tcp://”), porta (por exemplo, 1521) ou ambos, a opção ResolveTNS não será considerada e, de qualquer forma, será tratada como URI;
  - nome da sessão - se tal nome for encontrado no arquivo de configuração do plugin;
  - chave TNS (suportada desde o Zabbix 7.0.19) - se a opção ResolveTNS estiver definida como true e nenhuma das condições acima se aplicar;
  - valor TNS (suportado desde o Zabbix 7.0.19) - se começar com o parêntese aberto “(“ (espaços à esquerda ignorados). Neste caso, a opção ResolveTNS não é considerada.
- **user** - o nome de usuário do Oracle, suporta a adição de uma das opções de login `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` ou `as sysrac` no formato `user as sysdba` (uma opção de login não diferencia maiúsculas de minúsculas e não deve conter um espaço à direita). Observe que apenas as opções de login `as sysdba`, `as sysoper` ou `as sysasm` são suportadas antes do Zabbix 7.0.18;
- **password** - a senha do Oracle;
- **service** - o nome do serviço do Oracle;
- **lockMaxTime** - a duração máxima do bloqueio da sessão em segundos para contar a sessão como bloqueada por tempo prolongado. Padrão: 600 segundos.

oracle.sys.metrics[connString,<user>,<password>,<service>,<duration>]

 Retorna um conjunto de valores de métricas do sistema.
 Valor de retorno: *objeto JSON*.

Parâmetros:

- **connString** - pode ser:
  - o URI - se nenhuma sessão com o nome fornecido for encontrada e `Plugins.Oracle.ResolveTNS` estiver definido como false. Se contiver um esquema (por exemplo, ”tcp://”), porta (por exemplo, 1521) ou ambos, a opção ResolveTNS não será considerada e, de qualquer forma, será tratada como URI;
  - nome da sessão - se tal nome for encontrado no arquivo de configuração do plugin;
  - chave TNS (suportado desde o Zabbix 7.0.19) - se a opção ResolveTNS estiver definida como true e nenhuma das condições acima se aplicar;
  - valor TNS (suportado desde o Zabbix 7.0.19) - se começar com o parêntese de abertura “(“ (espaços à esquerda ignorados). Neste caso, a opção ResolveTNS não é considerada.
- **user** - o nome de usuário do Oracle, suporta a adição de uma das opções de login `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` ou `as sysrac` no formato `user as sysdba` (uma opção de login não diferencia maiúsculas de minúsculas, não deve conter um espaço à direita). Observe que apenas as opções de login `as sysdba`, `as sysoper` ou `as sysasm` são suportadas antes do Zabbix 7.0.18;
- **password** - a senha do Oracle;
- **service** - o nome do serviço Oracle;
- **duration** - o intervalo de captura (em segundos) dos valores das métricas do sistema. Valores possíveis: *60* — duração longa (padrão), *15* — duração curta.

oracle.sys.params[connString,<user>,<password>,<service>]

 Retorna um conjunto de valores de parâmetros do sistema.
 Valor de retorno: *objeto JSON*.

Parâmetros:

- **connString** - pode ser:
  - o URI - se nenhuma sessão com o nome fornecido for encontrada e `Plugins.Oracle.ResolveTNS` estiver definido como false. Se contiver um esquema (por exemplo, ”tcp://”), porta (por exemplo, 1521) ou ambos, a opção ResolveTNS não será considerada e, de qualquer forma, será tratada como URI;
  - nome da sessão - se tal nome for encontrado no arquivo de configuração do plugin;
  - chave TNS (suportado desde o Zabbix 7.0.19) - se a opção ResolveTNS estiver definida como true e nenhuma das condições acima se aplicar;
  - valor TNS (suportado desde o Zabbix 7.0.19) - se começar com o parêntese aberto “(“ (espaços à esquerda ignorados). Neste caso, a opção ResolveTNS não é considerada.
- **user** - o nome de usuário do Oracle, suporta a adição de uma das opções de login `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` ou `as sysrac` no formato `user as sysdba` (uma opção de login não diferencia maiúsculas de minúsculas, não deve conter um espaço à direita). Observe que apenas as opções de login `as sysdba`, `as sysoper` ou `as sysasm` são suportadas antes do Zabbix 7.0.18;
- **password** - a senha do Oracle;
- **service** - o nome do serviço Oracle.

oracle.ts.stats[connString,<user>,<password>,<service>,<tablespace>,<type>,<conname>]

 Retorna as estatísticas das tablespaces.
 Valor de retorno: *objeto JSON*.

Parâmetros:

- **connString** - pode ser:
  - o URI - se nenhuma sessão com o nome fornecido for encontrada e `Plugins.Oracle.ResolveTNS` estiver definido como false. Se contiver um esquema (por exemplo, ”tcp://”), porta (por exemplo, 1521) ou ambos, a opção ResolveTNS não será considerada e, de qualquer forma, será tratada como URI;
  - nome da sessão - se tal nome for encontrado no arquivo de configuração do plugin;
  - chave TNS (suportado desde o Zabbix 7.0.19) - se a opção ResolveTNS estiver definida como true e nenhuma das condições acima se aplicar;
  - valor TNS (suportado desde o Zabbix 7.0.19) - se começar com o parêntese aberto “(“ (espaços à esquerda ignorados). Neste caso, a opção ResolveTNS não é considerada.
- **user** - o nome de usuário do Oracle, suporta a adição de uma das opções de login `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` ou `as sysrac` no formato `user as sysdba` (uma opção de login não diferencia maiúsculas de minúsculas e não deve conter um espaço à direita). Observe que apenas as opções de login `as sysdba`, `as sysoper` ou `as sysasm` são suportadas antes do Zabbix 7.0.18;
- **password** - a senha do Oracle;
- **service** - o nome do serviço Oracle;
- **tablespace** - nome da tablespace a ser consultada. Padrão (se deixado em branco e `type` estiver definido):
  - ”TEMP” (se `type` estiver definido como ”TEMPORARY”);
  - ”USERS” (se `type` estiver definido como ”PERMANENT”).
- **type** - o tipo de tablespace a ser consultada. Padrão (se `tablespace` estiver definido): ”PERMANENT”.

- **conname** - nome do container para o qual a informação é necessária.

Se `tablespace`, `type` ou `conname` forem omitidos, o item retornará as estatísticas das tablespaces para todos os containers correspondentes (incluindo PDBs e CDB).

oracle.ts.discovery[connString,<user>,<password>,<service>]

 Retorna uma lista de tablespaces. Usado para descoberta de baixo nível.
 Valor de retorno: *objeto JSON*.

Parâmetros:

- **connString** - pode ser:
  - o URI - se nenhuma sessão com o nome fornecido for encontrada e `Plugins.Oracle.ResolveTNS` estiver definido como false. Se contiver um esquema (por exemplo, "tcp://"), porta (por exemplo, 1521) ou ambos, a opção ResolveTNS não será considerada e, de qualquer forma, será tratada como URI;
  - nome da sessão - se tal nome for encontrado no arquivo de configuração do plugin;
  - chave TNS (suportado desde o Zabbix 7.0.19) - se a opção ResolveTNS estiver definida como true e nenhuma das condições acima se aplicar;
  - valor TNS (suportado desde o Zabbix 7.0.19) - se começar com o parêntese aberto "(" (espaços à esquerda ignorados). Neste caso, a opção ResolveTNS não é considerada.
- **user** - o nome de usuário do Oracle, suporta a adição de uma das opções de login `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` ou `as sysrac` no formato `user as sysdba` (uma opção de login não diferencia maiúsculas de minúsculas e não deve conter um espaço à direita). Observe que apenas as opções de login `as sysdba`, `as sysoper` ou `as sysasm` são suportadas antes do Zabbix 7.0.18;
- **password** - a senha do Oracle;
- **service** - o nome do serviço Oracle.

oracle.user.info[connString,<user>,<password>,<service>,<username>]

 Retorna informações do usuário Oracle.
 Valor de retorno: *objeto JSON*.

Parâmetros:

- **connString** - pode ser:
  - o URI - se nenhuma sessão com o nome fornecido for encontrada e `Plugins.Oracle.ResolveTNS` estiver definido como false. Se contiver um esquema (por exemplo, "tcp://"), porta (por exemplo, 1521) ou ambos, a opção ResolveTNS não será considerada e, de qualquer forma, será tratada como URI;
  - nome da sessão - se tal nome for encontrado no arquivo de configuração do plugin;
  - chave TNS (suportado desde o Zabbix 7.0.19) - se a opção ResolveTNS estiver definida como true e nenhuma das condições acima se aplicar;
  - valor TNS (suportado desde o Zabbix 7.0.19) - se começar com o parêntese aberto "(" (espaços à esquerda ignorados). Neste caso, a opção ResolveTNS não é considerada.
- **user** - o nome de usuário do Oracle, suporta a adição de uma das opções de login `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` ou `as sysrac` no formato `user as sysdba` (uma opção de login não diferencia maiúsculas de minúsculas, não deve conter um espaço à direita). Observe que apenas as opções de login `as sysdba`, `as sysoper` ou `as sysasm` são suportadas antes do Zabbix 7.0.18;
- **password** - a senha do Oracle;
- **service** - o nome do serviço Oracle;
- **username** - o nome de usuário para o qual as informações são necessárias. Nomes de usuário em minúsculas não são suportados. Padrão: usuário atual.

oracle.version[connString,<user>,<password>,<service>]

 Retorna a versão do servidor de banco de dados.
 Valor de retorno: *String*.

Parâmetros:

- **connString** - pode ser:
  - o URI - se nenhuma sessão com o nome fornecido for encontrada e `Plugins.Oracle.ResolveTNS` estiver definido como false. Se contiver um esquema (por exemplo, "tcp://"), porta (por exemplo, 1521) ou ambos, a opção ResolveTNS não será considerada e, de qualquer forma, será tratada como URI;
  - nome da sessão - se tal nome for encontrado no arquivo de configuração do plugin;
  - chave TNS (suportado desde o Zabbix 7.0.19) - se a opção ResolveTNS estiver definida como true e nenhuma das condições acima se aplicar;
  - valor TNS (suportado desde o Zabbix 7.0.19) - se começar com o parêntese aberto "(" (espaços à esquerda são ignorados). Neste caso, a opção ResolveTNS não é considerada.
- **user** - o nome de usuário do Oracle, suporta a adição de uma das opções de login `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` ou `as sysrac` no formato `user as sysdba` (uma opção de login não diferencia

maiúsculas de minúsculas e não deve conter um espaço à direita). Observe que apenas as opções de login `as sysdba`, `as sysoper` ou `as sysasm` são suportadas antes do Zabbix 7.0.18;
- **password** - a senha do Oracle;
- **service** - o nome do serviço do Oracle.

pgsql.autovacuum.count[uri,<username>,<password>,<database name>]

 O número de workers de autovacuum.
 Valor de retorno: *Integer*.

Parâmetros:

- **uri** - o URI ou nome da sessão;
- **username, password** - as credenciais do PostgreSQL;
- **database name** - o nome do banco de dados (se omitido, o valor de Plugins.PostgreSQL.Default.Database de postgresql.conf é usado).

pgsql.archive[uri,<username>,<password>,<database name>]

 As informações sobre arquivos arquivados.
 Valor de retorno: *objeto JSON*.

Parâmetros:

- **uri** - o URI ou nome da sessão;
- **username, password** - as credenciais do PostgreSQL;
- **database name** - o nome do banco de dados (se omitido, o valor de Plugins.PostgreSQL.Default.Database de postgresql.conf é usado).

pgsql.bgwriter[uri,<username>,<password>,<database name>]

 O número combinado de checkpoints para o cluster de banco de dados, dividido por tipo de checkpoint.
 Valor de retorno: *objeto JSON*.

Parâmetros:

- **uri** - o URI ou nome da sessão;
- **username, password** - as credenciais do PostgreSQL;
- **database name** - o nome do banco de dados (se omitido, o valor de Plugins.PostgreSQL.Default.Database de postgresql.conf é usado).

pgsql.cache.hit[uri,<username>,<password>,<database name>]

 A taxa de acerto do buffer cache do PostgreSQL.
 Valor de retorno: *Float*.

Parâmetros:

- **uri** - o URI ou nome da sessão;
- **username, password** - as credenciais do PostgreSQL;
- **database name** - o nome do banco de dados (se omitido, o valor de Plugins.PostgreSQL.Default.Database do postgresql.conf é usado).

pgsql.connections[uri,<username>,<password>,<database name>]

 Retorna conexões por tipo.
 Valor de retorno: *objeto JSON*.

Parâmetros:

- **uri** - o URI ou nome da sessão;
- **username, password** - as credenciais do PostgreSQL;
- **database name** - o nome do banco de dados (se omitido, o valor de Plugins.PostgreSQL.Default.Database do postgresql.conf é usado).

pgsql.custom.query[uri,<username>,<password>,queryName,<args...>]

 Retorna o resultado de uma consulta personalizada.
 Valor de retorno: *objeto JSON*.

Parâmetros:

- **uri** - o URI ou nome da sessão;
- **username, password** - as credenciais do PostgreSQL;
- **queryName** - o nome de uma consulta personalizada, deve corresponder ao nome do arquivo SQL sem extensão;
- **args** - um ou vários argumentos separados por vírgula para passar para a consulta.

pgsql.db.age[uri,<username>,<password>,<database name>]

 A idade do FrozenXID mais antigo do banco de dados.
 Valor de retorno: *Inteiro*.

Parâmetros:

- **uri** - o URI ou nome da sessão;
- **username, password** - as credenciais do PostgreSQL;
- **database name** - o nome do banco de dados (se omitido, o valor de Plugins.PostgreSQL.Default.Database do postgresql.conf é usado).

pgsql.db.bloating_tables[uri,<username>,<password>,<database name>]

 O número de tabelas inchadas por banco de dados.
 Valor de retorno: *Integer*.

Parâmetros:

- **uri** - o URI ou nome da sessão;
- **username, password** - as credenciais do PostgreSQL;
- **database name** - o nome do banco de dados (se omitido, o valor de Plugins.PostgreSQL.Default.Database do postgresql.conf é usado).

pgsql.db.discovery[uri,<username>,<password>,<database name>]

 A lista de bancos de dados do PostgreSQL. Usado para descoberta de baixo nível.
 Valor de retorno: *objeto JSON*.

Parâmetros:

- **uri** - o URI ou nome da sessão;
- **username, password** - as credenciais do PostgreSQL;
- **database name** - o nome do banco de dados (se omitido, o valor de Plugins.PostgreSQL.Default.Database de postgresql.conf é usado).

pgsql.db.size[uri,<username>,<password>,<database name>]

 O tamanho do banco de dados em bytes.
 Valor de retorno: *Integer*.

Parâmetros:

- **uri** - o URI ou nome da sessão;
- **username, password** - as credenciais do PostgreSQL;
- **database name** - o nome do banco de dados (se omitido, é usado o valor de Plugins.PostgreSQL.Default.Database do postgresql.conf).

pgsql.dbstat[uri,<username>,<password>,<database name>]

 Coleta as estatísticas por banco de dados. Usado para descoberta de baixo nível.
 Valor de retorno: *objeto JSON*.

Parâmetros:

- **uri** - o URI ou nome da sessão;
- **username, password** - as credenciais do PostgreSQL;
- **database name** - o nome do banco de dados (se omitido, o valor de Plugins.PostgreSQL.Default.Database do postgresql.conf é usado).

pgsql.dbstat.sum[uri,<username>,<password>,<database name>]

 Os dados resumidos para todos os bancos de dados em um cluster.
 Valor de retorno: *objeto JSON*.

Parâmetros:

- **uri** - o URI ou nome da sessão;
- **username, password** - as credenciais do PostgreSQL;
- **database name** - o nome do banco de dados (se omitido, o valor de Plugins.PostgreSQL.Default.Database do postgresql.conf é usado).

pgsql.locks[uri,<username>,<password>,<database name>]

 Informações sobre bloqueios concedidos por banco de dados. Usado para descoberta de baixo nível.
 Valor de retorno: *objeto JSON*.

Parâmetros:

- **uri** - o URI ou nome da sessão;
- **username, password** - as credenciais do PostgreSQL;
- **database name** - o nome do banco de dados (se omitido, o valor de Plugins.PostgreSQL.Default.Database do postgresql.conf é usado).

pgsql.oldest.xid[uri,<username>,<password>,<database name>]

 A idade do XID mais antigo.
 Valor de retorno: *Inteiro*.

Parâmetros:

- **uri** - o URI ou nome da sessão;
- **username, password** - as credenciais do PostgreSQL;
- **database name** - o nome do banco de dados (se omitido, o valor de Plugins.PostgreSQL.Default.Database do postgresql.conf é usado).

pgsql.ping[uri,<username>,<password>,<database name>]

 Testa se uma conexão está ativa ou não.
 Valor de retorno: *1* - a conexão está ativa; *0* - a conexão está quebrada (se houver qualquer erro apresentado, incluindo problemas de AUTENTICAÇÃO e configuração).

Parâmetros:

- **uri** - o URI ou nome da sessão;
- **username, password** - as credenciais do PostgreSQL;
- **database name** - o nome do banco de dados (se omitido, o valor de Plugins.PostgreSQL.Default.Database do postgresql.conf é usado).

pgsql.queries[uri,<username>,<password>,<database name>,período de tempo]

 Métricas de consultas por tempo de execução.
 Valor de retorno: *objeto JSON*.

Parâmetros:

- **uri** - o URI ou nome da sessão;
- **username, password** - as credenciais do PostgreSQL;
- **database name** - o nome do banco de dados (se omitido, o valor de Plugins.PostgreSQL.Default.Database do postgresql.conf é usado);
- **período de tempo** - o limite de tempo de execução para a contagem de consultas lentas (deve ser um inteiro positivo).

pgsql.replication.count[uri,<username>,<password>]

 O número de servidores standby.
 Valor de retorno: *Inteiro*.

Parâmetros:

- **uri** - o URI ou nome da sessão;
- **username, password** - as credenciais do PostgreSQL.

pgsql.replication.process[uri,<username>,<password>]

 O atraso de flush, atraso de gravação e atraso de reprodução por cada processo de envio.
 Valor de retorno: *objeto JSON*.

Parâmetros:

- **uri** - o URI ou nome da sessão;
- **username, password** - as credenciais do PostgreSQL.

pgsql.replication.process.discovery[uri,<username>,<password>]

 A descoberta do nome do processo de replicação.
 Valor de retorno: *objeto JSON*.

Parâmetros:

- **uri** - o URI ou nome da sessão;
- **username, password** - as credenciais do PostgreSQL.

pgsql.replication.recovery_role[uri,<username>,<password>]

 O status de recuperação.
 Valor de retorno: *0* - modo master; *1* - a recuperação ainda está em andamento (modo standby).

Parâmetros:

- **uri** - o URI ou nome da sessão;
- **username, password** - as credenciais do PostgreSQL.

pgsql.replication.status[uri,<username>,<password>]

 O status da replicação.
 Valor de retorno: *0* - streaming está inativo; *1* - streaming está ativo; *2* - modo master.

Parâmetros:

- **uri** - o URI ou nome da sessão;
- **username, password** - as credenciais do PostgreSQL.

pgsql.replication_lag.b[uri,<username>,<password>]

 O atraso de replicação em bytes.
 Valor de retorno: *Integer*.

Parâmetros:

- **uri** - o URI ou nome da sessão;
- **username, password** - as credenciais do PostgreSQL.

pgsql.replication_lag.sec[uri,<username>,<password>]

 O atraso de replicação em segundos.
 Valor de retorno: *Integer*.

Parâmetros:

- **uri** - o URI ou nome da sessão;
- **username, password** - as credenciais do PostgreSQL.

pgsql.uptime[uri,<username>,<password>,<database name>]

 O tempo de atividade do PostgreSQL em milissegundos.
 Valor de retorno: *Float*.

Parâmetros:

- **uri** - o URI ou nome da sessão;
- **username, password** - as credenciais do PostgreSQL;
- **database name** - o nome do banco de dados (se omitido, o valor de Plugins.PostgreSQL.Default.Database do postgresql.conf é usado).

pgsql.version[uri,<username>,<password>,<database name>]

 Retorna a versão do PostgreSQL.
 Valor de retorno: *String*.

Parâmetros:

- **uri** - o URI ou nome da sessão;
- **username, password** - as credenciais do PostgreSQL;
- **database name** - o nome do banco de dados (se omitido, o valor de Plugins.PostgreSQL.Default.Database do postgresql.conf é usado).

pgsql.wal.stat[uri,<username>,<password>,<database name>]

 As estatísticas do WAL.
 Valor de retorno: *objeto JSON*.

Parâmetros:

- **uri** - o URI ou nome da sessão;
- **username, password** - as credenciais do PostgreSQL;
- **database name** - o nome do banco de dados (se omitido, o valor de Plugins.PostgreSQL.Default.Database do postgresql.conf é usado).

redis.config[connString,<password>,<pattern>]

 Obtém os parâmetros de configuração de uma instância Redis que correspondem ao padrão.
 Valor de retorno: *JSON* - se um padrão de estilo glob foi usado; valor único - se um padrão não contiver nenhum caractere curinga.

Parâmetros:

- **connString** - o URI ou nome da sessão;
- **password** - a senha do Redis;
- **pattern** - um padrão de estilo glob (* por padrão).

Comentário:

- Aviso de segurança: `redis.config` executa o comando `CONFIG GET` do Redis e pode retornar parâmetros de configuração sensíveis (por exemplo, `requirepass`, `masterauth`, configurações relacionadas a TLS e outros segredos). Usar o padrão padrão (`*`) ou padrões amplos pode expor esses segredos a qualquer pessoa com permissão para visualizar valores de item no Zabbix.

redis.info[connString,<password>,<section>]

 Obtém a saída do comando INFO.
 Valor de retorno: *JSON* - a saída é serializada como JSON.

Parâmetros:

- **connString** - o URI ou nome da sessão;
- **password** - a senha do Redis;
- **section** - a seção de informações (*default* por padrão).

redis.ping[connString,<password>]

 Testa se uma conexão está ativa ou não.
 Valor de retorno: *1* - a conexão está ativa; *0* - a conexão está quebrada (se houver qualquer erro apresentado, incluindo problemas de AUTH e configuração).

Parâmetros:

- **connString** - o URI ou nome da sessão;
- **password** - a senha do Redis.

redis.slowlog.count[connString,<password>]

 O número de entradas de log lento desde que o Redis foi iniciado.
 Valor de retorno: *Integer*.

Parâmetros:

- **connString** - o URI ou nome da sessão;
- **password** - a senha do Redis.

smart.attribute.discovery

 Retorna uma lista de atributos do dispositivo S.M.A.R.T.
 Valor de retorno: *objeto JSON*.

Comentários:

- As seguintes macros e seus valores são retornados: {#NAME}, {#DISKTYPE}, {#ID}, {#ATTRNAME}, {#THRESH};
- Os tipos de unidade HDD, SSD e NVME são suportados. As unidades podem estar sozinhas ou combinadas em um RAID. {#NAME} terá um complemento em caso de RAID, por exemplo: {"{#NAME}": "/dev/sda cciss,2"}.

smart.disk.discovery[<type>]

 Retorna uma lista de dispositivos S.M.A.R.T.
 Valor de retorno: *objeto JSON*.

Parâmetro:

- **type** - especifica um valor para verificar os discos. Valores possíveis: *id*, *name* (padrão). *id* não é suportado no Windows e retornará um erro se selecionado

Comentários:

- As seguintes macros e seus valores são retornados: {#NAME}, {#DISKTYPE}, {#MODEL}, {#SN}, {#PATH}, {#ATTRIBUTES}, {#RAIDTYPE};
- Os tipos de unidade HDD, SSD e NVME são suportados. Se uma unidade não pertencer a um RAID, o {#RAIDTYPE} estará vazio. {#NAME} terá um complemento no caso de RAID, por exemplo: {"{#NAME}": "/dev/sda cciss,2"}.

smart.disk.get[<path>,<raid type>]

 Retorna todas as propriedades disponíveis dos dispositivos S.M.A.R.T.
 Valor de retorno: *objeto JSON*.

Parâmetros:

- **path** - o caminho do disco, a macro {#PATH} pode ser usada como valor;
- **raid_type** - o tipo de RAID, a macro {#RAID} pode ser usada como valor

Comentários:

- Os tipos de unidade HDD, SSD e NVME são suportados. As unidades podem ser isoladas ou combinadas em um RAID;
- Os dados incluem a versão do smartctl e argumentos de chamada, e campos adicionais:
disk_name - contém o nome com os complementos necessários para a descoberta de RAID, por exemplo: {"disk_name": "/dev/sda cciss,2"}
disk_type - contém o tipo de disco HDD, SSD ou NVME, por exemplo: {"disk_type": "ssd"};
- Se nenhum parâmetro for especificado, o item retornará informações sobre todos os discos.

systemd.unit.get[unit name,<interface>]

 Retorna todas as propriedades de uma unidade systemd.
 Valor de retorno: *objeto JSON*.

Parâmetros:

- **unit name** - o nome da unidade (você pode querer usar a macro {#UNIT.NAME} no protótipo de item para descobrir o nome);
- **interface** - o tipo de interface da unidade, valores possíveis: *Unit* (padrão), *Service*, *Socket*, *Device*, *Mount*, *Automount*, *Swap*, *Target*, *Path*.

Comentários:

- Este item é suportado apenas na plataforma Linux;
- LoadState, ActiveState e UnitFileState para a interface Unit são retornados como texto e inteiro: `"ActiveState":{"state":1,"text`

systemd.unit.info[nome da unidade,<propriedade>,<interface>]

 Informação da unidade systemd.
 Valor de retorno: *String*.

Parâmetros:

- **nome da unidade** - o nome da unidade (você pode querer usar a macro {#UNIT.NAME} no protótipo de item para descobrir o nome);
- **propriedade** - propriedade da unidade (por exemplo, ActiveState (padrão), LoadState, Description);
- **interface** - o tipo de interface da unidade (por exemplo, Unit (padrão), Socket, Service).

Comentários:

- Este item é suportado apenas na plataforma Linux;
- Este item permite recuperar uma propriedade específica de um tipo específico de interface, conforme descrito na API dbus.

Exemplos:

```
systemd.unit.info["{#UNIT.NAME}"] #coleta informações do estado ativo (active, reloading, inactive, failed
systemd.unit.info["{#UNIT.NAME}",LoadState] #coleta informações do estado de carregamento das unidades sys
systemd.unit.info[mysqld.service,Id] #recupera o nome técnico do serviço (mysqld.service)
systemd.unit.info[mysqld.service,Description] #recupera a descrição do serviço (MySQL Server)
systemd.unit.info[mysqld.service,ActiveEnterTimestamp] #recupera a última vez que o serviço entrou no esta
systemd.unit.info[dbus.socket,NConnections,Socket] #coleta o número de conexões desta unidade de socket
```

systemd.unit.discovery[<type>]

 Lista de unidades systemd e seus detalhes. Usado para descoberta de baixo nível.
 Valor de retorno: *objeto JSON*.

Parâmetro:

- **type** - valores possíveis: *all*, *automount*, *device*, *mount*, *path*, *service* (padrão), *socket*, *swap*, *target*.

Este item é suportado apenas na plataforma Linux.

web.certificate.get[hostname,<port>,<address>]

 Valida os certificados e retorna detalhes do certificado.
 Valor de retorno: *objeto JSON*.

Parâmetros:

- **hostname** - pode ser IP ou DNS.
Pode conter o esquema da URL (*https* apenas), caminho (será ignorado) e porta.
Se uma porta for fornecida tanto no primeiro quanto no segundo parâmetro, seus valores devem coincidir.
Se o endereço (o 3º parâmetro) for especificado, o hostname será usado apenas para SNI e verificação do hostname;
- **port** - o número da porta (o padrão é 443 para HTTPS);
- **address** - pode ser IP ou DNS. Se especificado, será usado para conexão, e o hostname (o 1º parâmetro) será usado para SNI e verificação do host. Caso o 1º parâmetro seja um IP e o 3º parâmetro seja DNS, o 1º parâmetro será usado para conexão, e o 3º parâmetro será usado para SNI e verificação do host.

Comentários:

- Este item se torna não suportado se o destino especificado na configuração do host não existir, estiver indisponível ou se o handshake TLS falhar com qualquer erro, exceto um certificado inválido;
- Atualmente, a extensão X.509 AIA (Authority Information Access), CRLs e OCSP (incluindo OCSP stapling) e Certificate Transparency não são suportados;
- Campos de resposta JSON:
  - *x509*: contém os detalhes do certificado X.509.
    - *version*: a versão X.509 (por exemplo, 3).
    - *serial_number*: o número de série do certificado.
    - *signature_algorithm*: o algoritmo usado para assinar o certificado (por exemplo, `SHA256-RSA`).
    - *issuer*: o emissor do certificado.
    - *not_before*: a data de início da validade do certificado.
    - *not_after*: a data de expiração do certificado.
    - *subject*: o sujeito do certificado.
    - *public_key_algorithm*: o algoritmo usado para a chave pública (por exemplo, `RSA`).
    - *alternative_names*: nomes alternativos do sujeito (se presentes), caso contrário `null`.
  - result: contém o resultado da validação.
    - *value*: o status da validação (veja os valores possíveis abaixo).

        * *message*: mensagem detalhada da validação (por exemplo, "`certificate verified successfully`").
    - *sha1_fingerprint*: a impressão digital SHA-1 do certificado.
    - *sha256_fingerprint*: a impressão digital SHA-256 do certificado.
- O campo `$.result.value` indica o resultado da validação do certificado. Valores possíveis incluem:
    - *valid* – o certificado é válido e confiável.
    - *valid-but-self-signed* – o certificado é válido, mas autoassinado, ou seja, seu sujeito corresponde ao seu emissor.
    - *invalid* – o certificado é inválido devido a um problema como expiração, hostname incorreto ou uma autoridade de assinatura desconhecida.

Exemplo:

```
web.certificate.get[example.com,443]
```

Resposta JSON:


```
{
  "x509": {
    "version": 3,
    "serial_number": "0ad893bafa68b0b7fb7a404f06ecaf9a",
    "signature_algorithm": "ECDSA-SHA384",
    "issuer": "CN=DigiCert Global G3 TLS ECC SHA384 2020 CA1,O=DigiCert Inc,C=US",
    "not_before": {
      "value": "Jan 15 00:00:00 2025 GMT",
      "timestamp": 1736899200
    },
    "not_after": {
      "value": "Jan 15 23:59:59 2026 GMT",
      "timestamp": 1768521599
    },
    "subject": "CN=*.example.com,O=Internet Corporation for Assigned Names and Numbers,L=Los Angeles,ST=Ca
    "public_key_algorithm": "ECDSA",
    "alternative_names": [
      "*.example.com",
      "example.com"
    ]
  },
  "result": {
    "value": "valid",
    "message": "certificate verified successfully"
  },
  "sha1_fingerprint": "310db7af4b2bc9040c8344701aca08d0c69381e3",
  "sha256_fingerprint": "455943cf819425761d1f950263ebf54755d8d684c25535943976f488bc79d23b"
}
```


2 Agent Zabbix para Windows

Visão geral

Os items do agent Zabbix para Windows são apresentados em duas listas:

- Itens compartilhados - as chaves de item que são compartilhadas com o agent Zabbix para UNIX;
- Itens específicos do Windows - as chaves de item que são suportadas **apenas** no Windows.

Observe que todas as chaves de item suportadas pelo agent Zabbix no Windows também são suportadas pela nova geração do agent Zabbix 2. Veja as chaves de item adicionais que você pode usar apenas com o agent 2.

Veja também: Permissões mínimas para itens do Windows

Itens compartilhados

A tabela abaixo lista os itens do agent Zabbix que são suportados no Windows e são compartilhados com o agent Zabbix UNIX:

- A chave do item é um link para detalhes completos do item do agent Zabbix UNIX
- Comentários relevantes para o Windows estão incluídos

Chave do item	Descrição	Grupo de itens
log	O monitoramento de um arquivo de log. Este item não é suportado para o Log de Eventos do Windows. O parâmetro `persistent_dir` não é suportado no Windows.	Monitoramento de log
log.count	A contagem de linhas correspondentes em um arquivo de log monitorado. Este item não é suportado para o Log de Eventos do Windows. O parâmetro `persistent_dir` não é suportado no Windows.	
logrt	O monitoramento de um arquivo de log que é rotacionado. Este item não é suportado para o Log de Eventos do Windows. O parâmetro `persistent_dir` não é suportado no Windows.	
logrt.count	A contagem de linhas correspondentes em um arquivo de log monitorado que é rotacionado. Este item não é suportado para o Log de Eventos do Windows. O parâmetro `persistent_dir` não é suportado no Windows.	
modbus.get	Lê dados Modbus.	Modbus
net.dns	Verifica se o serviço DNS está ativo. Os parâmetros `ip`, `timeout` e `count` são ignorados no Windows, a menos que esteja usando o agent Zabbix 2.	Rede
net.dns.perf	Verifica o desempenho de um serviço DNS. Os parâmetros `ip`, `timeout` e `count` são ignorados no Windows, a menos que esteja usando o agent Zabbix 2.	
net.dns.record	Executa uma consulta DNS. Os parâmetros `ip`, `timeout` e `count` são ignorados no Windows, a menos que esteja usando o agent Zabbix 2.	
net.if.discovery	A lista de interfaces de rede. Algumas versões do Windows (por exemplo, Server 2008) podem exigir as atualizações mais recentes instaladas para suportar caracteres não ASCII nos nomes das interfaces.	
net.if.in	As estatísticas de tráfego de entrada em uma interface de rede. No Windows, o item obtém valores de contadores de 64 bits, se disponíveis. Os contadores de estatísticas de interface de 64 bits foram introduzidos no Windows Vista e Windows Server 2008. Se os contadores de 64 bits não estiverem disponíveis, o agent usa contadores de 32 bits. Nomes de interface multibyte no Windows são suportados. Você pode obter descrições de interfaces de rede no Windows com os itens net.if.discovery ou net.if.list.	
net.if.out	As estatísticas de tráfego de saída em uma interface de rede. No Windows, o item obtém valores de contadores de 64 bits, se disponíveis. Os contadores de estatísticas de interface de 64 bits foram introduzidos no Windows Vista e Windows Server 2008. Se os contadores de 64 bits não estiverem disponíveis, o agent usa contadores de 32 bits. Nomes de interface multibyte no Windows são suportados. Você pode obter descrições de interfaces de rede no Windows com os itens net.if.discovery ou net.if.list.	
net.if.total	A soma das estatísticas de tráfego de entrada e saída em uma interface de rede. No Windows, o item obtém valores de contadores de 64 bits, se disponíveis. Os contadores de estatísticas de interface de 64 bits foram introduzidos no Windows Vista e Windows Server 2008. Se os contadores de 64 bits não estiverem disponíveis, o agent usa contadores de 32 bits. Você pode obter descrições de interfaces de rede no Windows com os itens net.if.discovery ou net.if.list.	
net.tcp.listen	Verifica se esta porta TCP está no estado LISTEN.	
net.tcp.port	Verifica se é possível fazer uma conexão TCP com a porta especificada.	
net.tcp.service	Verifica se um serviço está em execução e aceitando conexões TCP. A verificação de LDAP e HTTPS no Windows só é suportada pelo agent Zabbix 2.	
net.tcp.service.perf	Verifica o desempenho de um serviço TCP. A verificação de LDAP e HTTPS no Windows só é suportada pelo agent Zabbix 2.	
net.tcp.socket.count	Retorna o número de sockets TCP que correspondem aos parâmetros. Este item é suportado no Linux pelo agent Zabbix, mas no Windows é suportado apenas pelo agent Zabbix 2 no Windows 64 bits.	
net.udp.service	Verifica se um serviço está em execução e respondendo a solicitações UDP.	
net.udp.service.perf	Verifica o desempenho de um serviço UDP.	

Chave do item	Descrição	Grupo de itens
net.udp.socket.count	Retorna o número de sockets UDP que correspondem aos parâmetros. Este item é suportado no Linux pelo agent Zabbix, mas no Windows é suportado apenas pelo agent Zabbix 2 no Windows 64 bits.	
proc.get	A lista de processos do SO e seus parâmetros. O parâmetro `cmdline` não é suportado no Windows.	Processos
proc.num	O número de processos. No Windows, apenas os parâmetros `name` e `user` são suportados.	
system.cpu.discovery	A lista de CPUs/núcleos de CPU detectados.	Sistema
system.cpu.load	A carga da CPU. Quando um processo coletor é iniciado no agent Zabbix, os seguintes contadores de desempenho são inicializados e posteriormente usados para este item: `\System\Processor Queue Length`	
system.cpu.num	O número de CPUs.	
system.cpu.util	A porcentagem de utilização da CPU. O valor é adquirido usando o contador de desempenho *Processor Time*. Observe que, desde o Windows 8, o Gerenciador de Tarefas mostra a utilização da CPU com base no contador de desempenho *Processor Utility*, enquanto nas versões anteriores era o contador *Processor Time* (veja mais detalhes). *system* é o único parâmetro `type` suportado no Windows.	
system.hostname	O nome do host do sistema. O valor é adquirido pelas funções GetComputerName() (para **netbios**), GetComputerNameExA() (para **fqdn**) ou gethostname() (para **host**) no Windows. Veja também uma descrição mais detalhada.	
system.localtime	A hora do sistema.	
system.run	Executa o comando especificado no host.	
system.sw.arch	As informações de arquitetura de software.	
system.sw.os	As informações do sistema operacional.	
system.sw.os.get	Informações detalhadas sobre o sistema operacional (versão, tipo, nome da distribuição, versão menor e maior, etc).	
system.swap.size	O tamanho do espaço de swap em bytes ou em porcentagem do total. O parâmetro de tipo `pused` é suportado no Linux pelo agent Zabbix, mas no Windows é suportado apenas pelo agent Zabbix 2. Observe que esta chave pode relatar tamanho/porcentagem de swap incorretos em plataformas Windows virtualizadas (VMware ESXi, VirtualBox). Neste caso, você pode usar a chave `perf_counter[\700(_Total)\702]` para obter a porcentagem correta de swap.	
system.uname	Identificação do sistema. No Windows, o valor para este item é obtido das classes WMI Win32_OperatingSystem e Win32_Processor. O nome do SO (incluindo edição) pode ser traduzido para o idioma de exibição do usuário. Em algumas versões do Windows, contém símbolos de marca registrada e espaços extras.	
system.uptime	O tempo de atividade do sistema em segundos.	
vfs.dir.count	A contagem de entradas do diretório. No Windows, links simbólicos de diretório são ignorados e hard links são contados apenas uma vez.	Sistemas de arquivos virtuais
vfs.dir.get	A lista de entradas do diretório. No Windows, links simbólicos de diretório são ignorados e hard links são contados apenas uma vez.	
vfs.dir.size	O tamanho do diretório. No Windows, qualquer link simbólico é ignorado e hard links são considerados apenas uma vez.	
vfs.file.cksum	O checksum do arquivo, calculado pelo algoritmo UNIX cksum.	
vfs.file.contents	Recuperando o conteúdo de um arquivo.	
vfs.file.exists	Verifica se o arquivo existe. No Windows, as aspas duplas devem ser escapadas com barra invertida '\' e toda a chave do item deve ser colocada entre aspas duplas ao usar o utilitário de linha de comando para chamar zabbix_get.exe ou agent2. Observe que o item pode se tornar não suportado no Windows se um diretório for pesquisado dentro de um diretório inexistente, por exemplo, `vfs.file.exists[C:\no\dir,dir]` (onde 'no' não existe).	

Chave do item	Descrição	Grupo de itens
vfs.file.get	Retorna informações sobre um arquivo. Tipos de arquivo suportados no Windows: arquivo regular, diretório, link simbólico	
vfs.file.md5sum	O checksum MD5 do arquivo.	
vfs.file.owner	Recupera o proprietário de um arquivo.	
vfs.file.regexp	Recupera uma string no arquivo.	
vfs.file.regmatch	Encontra uma string no arquivo.	
vfs.file.size	O tamanho do arquivo.	
vfs.file.time	As informações de tempo do arquivo. No Windows XP, `vfs.file.time[file,change]` pode ser igual a `vfs.file.time[file,access]`.	
vfs.fs.discovery	A lista de sistemas de arquivos montados com seu tipo e opções de montagem. A macro `{#FSLABEL}` é suportada no Windows.	
vfs.fs.get	A lista de sistemas de arquivos montados com seu tipo, espaço em disco disponível, estatísticas de inode e opções de montagem. A macro `{#FSLABEL}` é suportada no Windows.	
vfs.fs.size	O espaço em disco em bytes ou em porcentagem do total.	
vm.memory.size	O tamanho da memória em bytes ou em porcentagem do total.	Memória virtual
web.page.get	Obtém o conteúdo de uma página web.	Monitoramento web
web.page.perf	O tempo de carregamento de uma página web completa.	
web.page.regexp	Encontra uma string na página web.	
agent.hostmetadata	Os metadados do host do agent.	Zabbix
agent.hostname	O nome do host do agent.	
agent.ping	A verificação de disponibilidade do agent.	
agent.variant	A variante do agent Zabbix (agent Zabbix ou agent Zabbix 2).	
agent.version	A versão do agent Zabbix.	
zabbix.stats	Retorna um conjunto de métricas internas do server ou proxy Zabbix remotamente.	
zabbix.stats	Retorna o número de itens monitorados na fila que estão atrasados no server ou proxy Zabbix remotamente.	

Itens específicos do Windows

A tabela fornece detalhes sobre as chaves de item que são suportadas **apenas** pelo agent Zabbix para Windows.

Itens específicos do Windows às vezes são equivalentes aproximados de um item de agent semelhante, por exemplo, `proc_info`, suportado no Windows, corresponde aproximadamente ao item `proc.mem`, não suportado no Windows.

A chave do item é um link para detalhes completos da chave do item.

Chave do item	Descrição	Grupo de item
eventlog	Monitoramento do log de eventos do Windows.	Monitoramento de log
eventlog.count	A contagem de linhas no log de eventos do Windows.	
net.if.list	A lista de interfaces de rede (inclui tipo de interface, status, endereço IPv4, descrição).	Rede
perf_counter	O valor de qualquer contador de desempenho do Windows.	Contadores de desempenho
perf_counter_en	O valor de qualquer contador de desempenho do Windows em inglês.	
perf_instance.discovery	A lista de instâncias de objeto dos contadores de desempenho do Windows.	
perf_instance_en.discovery	A lista de instâncias de objeto dos contadores de desempenho do Windows, descobertas usando os nomes dos objetos em inglês.	
proc_info	Várias informações sobre processos específicos.	Processos
registry.data	Retorna dados para o nome de valor especificado na chave do Registro do Windows.	Registro
registry.get	A lista de valores ou chaves do Registro do Windows localizados na chave fornecida.	
service.discovery	A lista de serviços do Windows.	Serviços

Chave do item	Descrição	Grupo de item
service.info	Informações sobre um serviço.	
services	A listagem de serviços.	
vm.vmemory.size	O tamanho da memória virtual em bytes ou em porcentagem do total.	Memória virtual
wmi.get	Executa uma consulta WMI e retorna o primeiro objeto selecionado.	WMI
wmi.getall	Executa uma consulta WMI e retorna toda a resposta.	

Detalhes da chave do item

Parâmetros sem colchetes angulares são obrigatórios. Parâmetros marcados com colchetes angulares **< >** são opcionais.

eventlog[name,<regexp>,<severity>,<source>,<eventid>,<maxlines>,<mode>]

 Monitoramento de log de eventos.
 Valor de retorno: *Log*.

Parâmetros:

- **name** - o nome do canal do log de eventos (*Nome do Log* na interface do Visualizador de Eventos);
- **regexp** - uma expressão regular descrevendo o padrão necessário (diferencia maiúsculas de minúsculas);
- **severity** - uma expressão regular descrevendo a severidade (não diferencia maiúsculas de minúsculas). Este parâmetro aceita uma expressão regular baseada nos seguintes valores: ”Information”, ”Warning”, ”Error”, ”Critical”, ”Verbose” (executando no Windows Vista ou mais recente).
- **source** - uma expressão regular descrevendo o identificador da fonte (não diferencia maiúsculas de minúsculas);
- **eventid** - uma expressão regular descrevendo o(s) identificador(es) do evento (diferencia maiúsculas de minúsculas);
- **maxlines** - o número máximo de novas linhas por segundo que o agent enviará para o Zabbix server ou proxy. Este parâmetro sobrescreve o valor de ’MaxLinesPerSecond’ em zabbix_agentd.conf.
- **mode** - valores possíveis: *all* (padrão) ou *skip* - ignora o processamento de dados antigos (afeta apenas itens recém-criados).

Comentários:

- O item deve ser configurado como uma verificação ativa;
- O agent não consegue enviar eventos do log ”Forwarded events”;
- O Windows Eventing 6.0 é suportado;
- Selecionar um tipo de informação diferente de Log para este item resultará na perda do timestamp local, bem como das informações de severidade e fonte do log;
- Veja também informações adicionais sobre monitoramento de logs.

Exemplos:

```
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] #aqui uma expressão regular personalizada chamada `TWOSHORT` é referenciada
```

eventlog.count[name,<regexp>,<severity>,<source>,<eventid>,<maxproclines>,<mode>]

 A contagem de linhas no log de eventos do Windows.
 Valor de retorno: *Integer*.

Parâmetros:

- **name** - o nome do canal do log de eventos (*Nome do Log* na interface do Visualizador de Eventos);
- **regexp** - uma expressão regular descrevendo o padrão necessário (diferencia maiúsculas de minúsculas);
- **severity** - uma expressão regular descrevendo a severidade (não diferencia maiúsculas de minúsculas). Este parâmetro aceita uma expressão regular baseada nos seguintes valores: ”Information”, ”Warning”, ”Error”, ”Critical”, ”Verbose” (executando no Windows Vista ou mais recente).
- **source** - uma expressão regular descrevendo o identificador da fonte (não diferencia maiúsculas de minúsculas);
- **eventid** - uma expressão regular descrevendo o(s) identificador(es) do evento (diferencia maiúsculas de minúsculas);
- **maxproclines** - o número máximo de novas linhas por segundo que o agent irá analisar (não pode exceder 10000). O valor padrão é 10*’MaxLinesPerSecond’ em zabbix_agentd.conf.
- **mode** - valores possíveis: *all* (padrão) ou *skip* - ignora o processamento de dados antigos (afeta apenas os itens recém-criados).

Comentários:

- O item deve ser configurado como uma verificação ativa;
- O agent não consegue enviar eventos do log "Forwarded events";
- O Windows Eventing 6.0 é suportado;
- Selecionar um tipo de informação diferente de Log para este item levará à perda do timestamp local, bem como das informações de severidade e fonte do log;
- Veja também informações adicionais sobre monitoramento de logs.

Exemplos:

```
eventlog.count[System,,"Warning|Error"]
eventlog.count[Windows PowerShell,,,,,,skip]
```

net.if.list

 A lista de interfaces de rede (inclui tipo de interface, status, endereço IPv4, descrição).
 Valor de retorno: *Texto*.

Comentários:

- Nomes de interface multibyte são suportados;
- Interfaces desabilitadas não são listadas;
- Ativar/desativar alguns componentes pode alterar a ordem no nome da interface do Windows;
- Algumas versões do Windows (por exemplo, Server 2008) podem exigir as últimas atualizações instaladas para suportar caracteres não ASCII nos nomes das interfaces.

perf_counter[counter,<interval>]

 O valor de qualquer contador de desempenho do Windows.
 Valor de retorno: *Integer*, *float*, *string* ou *text* (dependendo da solicitação).

Parâmetros:

- **counter** - o caminho para o contador;
- **interval** - os últimos N segundos para armazenar o valor médio. O `interval` deve estar entre 1 e 900 segundos (inclusive) e o valor padrão é 1.

Comentários:

- Este item segue caminhos UNC, o que pode ser um risco de segurança;
- `interval` é usado para contadores que requerem mais de uma amostra (como utilização de CPU), portanto, a verificação retorna um valor médio para os últimos "interval" segundos a cada vez;
- O Performance Monitor pode ser usado para obter a lista de contadores disponíveis.
- Veja também: Contadores de desempenho do Windows.

perf_counter_en[counter,<interval>]

 O valor de qualquer contador de desempenho do Windows em inglês.
 Valor de retorno: *Integer*, *float*, *string* ou *text* (dependendo da solicitação).

Parâmetros:

- **counter** - o caminho para o contador em inglês;
- **interval** - os últimos N segundos para armazenar o valor médio. O `interval` deve estar entre 1 e 900 segundos (inclusive) e o valor padrão é 1.

Comentários:

- Este item segue caminhos UNC, o que pode ser um risco de segurança;
- `interval` é usado para contadores que requerem mais de uma amostra (como utilização de CPU), então a verificação retorna um valor médio para os últimos "interval" segundos a cada vez;
- Este item é suportado apenas no **Windows Server 2008/Vista** e superior;
- Você pode encontrar a lista de strings em inglês visualizando a seguinte chave de registro: `HKEY_LOCAL_MACHINE\SOFTWARE\Micros NT\CurrentVersion\Perflib\009`.

perf_instance.discovery[object]

 A lista de instâncias de objetos dos contadores de desempenho do Windows. Usado para descoberta de baixo nível.
 Valor de retorno: *objeto JSON*.

Parâmetro:

- **object** - o nome do objeto (localizado).

perf_instance_en.discovery[object]

 A lista de instâncias de objeto dos contadores de desempenho do Windows, descobertas usando os nomes de objeto em inglês. Usado para descoberta de baixo nível.
 Valor de retorno: *objeto JSON*.

Parâmetro:

- **object** - o nome do objeto (em inglês).

proc_info[processo,<atributo>,<tipo>]

 Várias informações sobre processo(s) específico(s).
 Valor de retorno: *Float*.

Parâmetros:

- **processo** - o nome do processo;
- **atributo** - o atributo do processo solicitado;
- **tipo** - o tipo de representação (significativo quando mais de um processo com o mesmo nome existe)

Comentários:

- Os seguintes `atributos` são suportados:
vmsize (padrão) - tamanho da memória virtual do processo em Kbytes
wkset - tamanho do conjunto de trabalho do processo (quantidade de memória física usada pelo processo) em Kbytes
pf - número de page faults
ktime - tempo de kernel do processo em milissegundos
utime - tempo de usuário do processo em milissegundos
io_read_b - número de bytes lidos pelo processo durante operações de I/O
io_read_op - número de operações de leitura realizadas pelo processo
io_write_b - número de bytes gravados pelo processo durante operações de I/O
io_write_op - número de operações de gravação realizadas pelo processo
io_other_b - número de bytes transferidos pelo processo durante operações diferentes de leitura e gravação
io_other_op - número de operações de I/O realizadas pelo processo, exceto leitura e gravação
gdiobj - número de objetos GDI usados pelo processo
userobj - número de objetos USER usados pelo processo;
- Os `tipos` válidos são:
avg (padrão) - valor médio para todos os processos chamados <processo>
min - valor mínimo entre todos os processos chamados <processo>
max - valor máximo entre todos os processos chamados <processo>
sum - soma dos valores para todos os processos chamados <processo>;
- Em um sistema 64 bits, é necessário um agent Zabbix 64 bits para que este item funcione corretamente.

Exemplos:

```
proc_info[iexplore.exe,wkset,sum] #recupera a quantidade de memória física usada por todos os processos do
proc_info[iexplore.exe,pf,avg] #recupera o número médio de page faults para processos do Internet Explorer
```

registry.data[key,<value name>]

 Retorna os dados para o nome de valor especificado na chave do Registro do Windows.
 Valor de retorno: *Integer*, *string* ou *text* (dependendo do tipo de valor)

Parâmetros:

- **key** - a chave do registro incluindo a chave raiz; abreviações de raiz (por exemplo, HKLM) são permitidas;
- **value name** - o nome do valor do registro na chave (string vazia "" por padrão). O valor padrão é retornado se o nome do valor não for fornecido.

Comentários:

- Abreviações de raiz suportadas:
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
- Chaves com espaços devem ser colocadas entre aspas duplas.

Exemplos:

```
registry.data["HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\Windows Error Reporting"] #retorna os dados d
registry.data["HKLM\SOFTWARE\Microsoft\Windows\Windows Error Reporting","EnableZip"] #retorna os dados do
```

registry.get[key,<mode>,<name regexp>]

 A lista de valores ou chaves do Registro do Windows localizados na chave fornecida.
 Valor de retorno: *objeto JSON*.

Parâmetros:

- **key** - a chave do registro incluindo a chave raiz; abreviações de raiz (por exemplo, HKLM) são permitidas (veja os comentários para registry.data[] para ver a lista completa de abreviações);
- **mode** - valores possíveis:
values (padrão) ou *keys*;
- **name regexp** - descobre apenas valores com nomes que correspondam à expressão regular (padrão - descobre todos os valores). Permitido apenas com *values* como `mode`.

Chaves com espaços devem ser colocadas entre aspas duplas.

Exemplos:

```
registry.get[HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall,values,"^DisplayName|DisplayVersion$
O JSON incluirá detalhes da chave, última subchave, nome do valor, tipo do valor e dados do valor.
registry.get[HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall,values] #retorna os dados de todos o
O JSON incluirá detalhes da chave, última subchave, nome do valor, tipo do valor e dados do valor.
registry.get[HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall,keys] #retorna todas as subchaves de
O JSON incluirá detalhes da chave e da última subchave.
```

service.discovery

 A lista de serviços do Windows. Usado para descoberta de baixo nível.
 Valor de retorno: *objeto JSON*.

service.info[service,<param>]

 Informações sobre um serviço.
 Valor de retorno: *Integer* - com `param` como *state*, *startup*; *String* - com `param` como *displayname*, *path*, *user*; *Text* - com `param` como *description*
Especificamente para *state*: 0 - em execução, 1 - pausado, 2 - inicialização pendente, 3 - pausa pendente, 4 - continuação pendente, 5 - parada pendente, 6 - parado, 7 - desconhecido, 255 - serviço inexistente
Especificamente para *startup*: 0 - automático, 1 - automático atrasado, 2 - manual, 3 - desabilitado, 4 - desconhecido, 5 - inicialização automática por trigger, 6 - inicialização automática atrasada por trigger, 7 - inicialização manual por trigger

Parâmetros:

- **service** - nome real do serviço ou seu nome de exibição conforme visto no MMC Services snap-in;
- **param** - *state* (padrão), *displayname*, *path*, *user*, *startup* ou *description*.

Comentários:

- Items como `service.info[service,state]` e `service.info[service]` retornarão a mesma informação;
- Somente com `param` como *state* este item retorna um valor para serviços inexistentes (255).

Exemplos:

```
service.info[SNMPTRAP] - estado do serviço SNMPTRAP;
service.info[SNMP Trap] - estado do mesmo serviço, mas com o nome de exibição especificado;
service.info[EventLog,startup] - o tipo de inicialização do serviço EventLog
```

services[<type>,<state>,<exclude>]

 A listagem de serviços.
 Valor de retorno: *0* - se vazio; *Texto* - a lista de serviços separados por nova linha.

Parâmetros:

- **type** - *all* (padrão), *automatic*, *manual* ou *disabled*;
- **state** - *all* (padrão), *stopped*, *started*, *start_pending*, *stop_pending*, *running*, *continue_pending*, *pause_pending* ou *paused*;
- **exclude** - os serviços a serem excluídos do resultado. Os serviços excluídos devem ser listados entre aspas duplas, separados por vírgula, sem espaços.

Exemplos:

```
services[,started] #retorna a lista de serviços iniciados;
services[automatic, stopped] #retorna a lista de serviços parados que deveriam estar em execução;
services[automatic, stopped, "service1,service2,service3"] #retorna a lista de serviços parados que deveri
```

vm.vmemory.size[<type>]

 O tamanho da memória virtual em bytes ou em porcentagem do total.
 Valor de retorno: *Integer* - para bytes; *float* - para porcentagem.

Parâmetro:

- **type** - valores possíveis: *available* (memória virtual disponível), *pavailable* (memória virtual disponível, em porcentagem), *pused* (memória virtual usada, em porcentagem), *total* (memória virtual total, padrão), ou *used* (memória virtual usada)

Comentários:

- O monitoramento das estatísticas de memória virtual é baseado em:
  - Memória virtual total no Windows (física total + tamanho do arquivo de paginação);
  - A quantidade máxima de memória que o agent Zabbix pode comprometer;
  - O limite atual de memória comprometida para o sistema ou agent Zabbix, o que for menor.

Exemplo:

```
vm.vmemory.size[pavailable] #retorna a memória virtual disponível, em porcentagem
```

wmi.get[<namespace>,<query>]

 Executa uma consulta WMI e retorna o primeiro objeto selecionado.
 Valor de retorno: *Integer*, *float*, *string* ou *text* (dependendo da solicitação).

Parâmetros:

- **namespace** - o namespace WMI;
- **query** - a consulta WMI retornando um único objeto.

As consultas WMI são realizadas com WQL.

Exemplo:

```
wmi.get[root\cimv2,select status from Win32_DiskDrive where Name like '%PHYSICALDRIVE0%'] #retorna o statu
```

wmi.getall[<namespace>,<query>]

 Executa uma consulta WMI e retorna toda a resposta. Pode ser usado para descoberta de baixo nível.
 Valor de retorno: *objeto JSON*

Parâmetros:

- **namespace** - o namespace WMI;
- **query** - a consulta WMI.

Comentários:

- As consultas WMI são realizadas com WQL.
- O pré-processamento JSONPath pode ser usado para apontar para valores mais específicos no JSON retornado.

Exemplo:

```
wmi.getall[root\cimv2,select * from Win32_DiskDrive where Name like '%PHYSICALDRIVE%'] #retorna informaçõe
```

Monitorando serviços do Windows

Este tutorial fornece instruções passo a passo para configurar o monitoramento de serviços do Windows. Pressupõe-se que o server e o agent do Zabbix estejam configurados e operacionais.

Passo 1

Obtenha o nome do serviço.

Você pode obter o nome do serviço acessando o snap-in Serviços do MMC e exibindo as propriedades do serviço. Na guia *Geral* você deve ver um campo chamado "Nome do serviço". O valor que segue é o nome que você usará ao configurar um item para monitoramento. Por exemplo, se você quiser monitorar o serviço "workstation", então seu serviço pode ser: **lanmanworkstation**.

Passo 2

Configure um item para monitorar o serviço.

O item `service.info[service,<param>]` recupera informações sobre um serviço específico. Dependendo da informação que você precisa, especifique a opção `param`, que aceita os seguintes valores: *displayname*, *state*, *path*, *user*, *startup* ou *description*. O valor padrão é *state* se `param` não for especificado (`service.info[service]`).

O tipo de valor de retorno depende do `param` escolhido: inteiro para *state* e *startup*; string de caracteres para *displayname*, *path* e *user*; texto para *description*.

Exemplo:

- *Chave:* `service.info[lanmanworkstation]`
- *Tipo de informação:* Numérico (sem sinal)

O item `service.info[lanmanworkstation]` irá recuperar informações sobre o estado do serviço como um valor numérico. Para mapear um valor numérico para uma representação de texto no frontend ("0" como "Em execução", "1" como "Pausado", etc.), você pode configurar o mapeamento de valores no host em que o item está configurado. Para isso, vincule o template *Windows services by Zabbix agent* ou *Windows services by Zabbix agent active* ao host, ou configure no host um novo mapa de valores baseado no mapa de valores *Windows service state* configurado nos templates mencionados.

Observe que ambos os templates mencionados possuem uma regra de descoberta configurada que irá descobrir serviços automaticamente. Se você não quiser isso, você pode desabilitar a regra de descoberta no nível do host assim que o template for vinculado ao host.

Descoberta de serviços do Windows

A descoberta de baixo nível fornece uma maneira de criar automaticamente items, triggers e gráficos para diferentes entidades em um computador. O Zabbix pode iniciar automaticamente o monitoramento dos serviços do Windows em sua máquina, sem a necessidade de saber o nome exato de um serviço ou criar items para cada serviço manualmente. Um filtro pode ser usado para gerar items, triggers e gráficos reais apenas para os serviços de interesse.

2 Agente SNMP

Visão geral

Você pode querer usar o monitoramento SNMP em dispositivos como impressoras, switches de rede, roteadores ou nobreaks que geralmente possuem SNMP habilitado e nos quais seria impraticável tentar configurar sistemas operacionais completos e agents Zabbix.

Para poder recuperar dados fornecidos por agents SNMP nesses dispositivos, o server Zabbix deve ser configurado inicialmente com suporte a SNMP especificando a flag `--with-net-snmp`. Recomenda-se também instalar arquivos MIB para garantir que os valores dos items sejam exibidos no formato correto. Sem os arquivos MIB, podem ocorrer problemas de formatação, como exibir valores em HEX em vez de UTF-8 ou vice-versa.

As verificações SNMP são realizadas apenas pelo protocolo UDP.

Os daemons server e proxy do Zabbix registram linhas semelhantes às seguintes se receberem uma resposta SNMP incorreta:

```
SNMP response from host "gateway" does not contain all of the requested variable bindings
```

Embora não cubram todos os casos problemáticos, são úteis para identificar dispositivos SNMP individuais para os quais as solicitações combinadas devem ser desabilitadas.

O server/proxy Zabbix tentará novamente até 5 vezes (desde o Zabbix 7.0.14) para items SNMP `walk` e `get`. O mecanismo de repetição não se aplica a falhas de resolução de DNS.

Para verificações SNMP legadas (número OID único ou string), o server/proxy Zabbix tentará pelo menos uma vez após uma tentativa de consulta malsucedida: seja por meio do mecanismo de repetição da biblioteca SNMP ou por meio do mecanismo interno de processamento combinado.

Warning:
Se estiver monitorando dispositivos SNMPv3, certifique-se de que msgAuthoritativeEngineID (também conhecido como snmpEngineID ou "Engine ID") nunca seja compartilhado por dois dispositivos. De acordo com a RFC 2571 (seção 3.1.1.1), ele deve ser exclusivo para cada dispositivo.

Warning:
A RFC3414 exige que os dispositivos SNMPv3 persistam seus engineBoots. Alguns dispositivos não fazem isso, o que resulta em suas mensagens SNMP sendo descartadas como desatualizadas após serem reiniciadas. Nessa situação, o cache SNMP precisa ser limpo manualmente em um server/proxy (usando -R snmp_cache_reload) ou o server/proxy precisa ser reiniciado.

Configurando o monitoramento SNMP

Para começar a monitorar um dispositivo via SNMP, as seguintes etapas devem ser realizadas:

Passo 1

Descubra a string SNMP (ou OID) do item que você deseja monitorar.

Para obter uma lista de strings SNMP, use o comando **snmpwalk** (parte do software net-snmp, que deve ter sido instalado como parte da instalação do Zabbix) ou uma ferramenta equivalente:

```
snmpwalk -v 2c -c public <host IP> .
```

Como '2c' aqui representa a versão do SNMP, você também pode substituí-lo por '1', para indicar a Versão 1 do SNMP no dispositivo.

Isso deve fornecer uma lista de strings SNMP e seus últimos valores. Se não fornecer, é possível que a 'comunidade' SNMP seja diferente da padrão 'public', caso em que você precisará descobrir qual é.

Você pode então percorrer a lista até encontrar a string que deseja monitorar, por exemplo, se você quiser monitorar os bytes que entram em sua switch na porta 3, você usaria a string `IF-MIB::ifHCInOctets.3` desta linha:

```
IF-MIB::ifHCInOctets.3 = Counter64: 3409739121
```

Agora você pode usar o comando **snmpget** para descobrir o OID numérico para 'IF-MIB::ifHCInOctets.3':

```
snmpget -v 2c -c public -On <host IP> IF-MIB::ifHCInOctets.3
```

Observe que o último número na string é o número da porta que você deseja monitorar. Veja também: Índices dinâmicos.

Isso deve fornecer algo como o seguinte:

```
.1.3.6.1.2.1.31.1.1.1.6.3 = Counter64: 3472126941
```

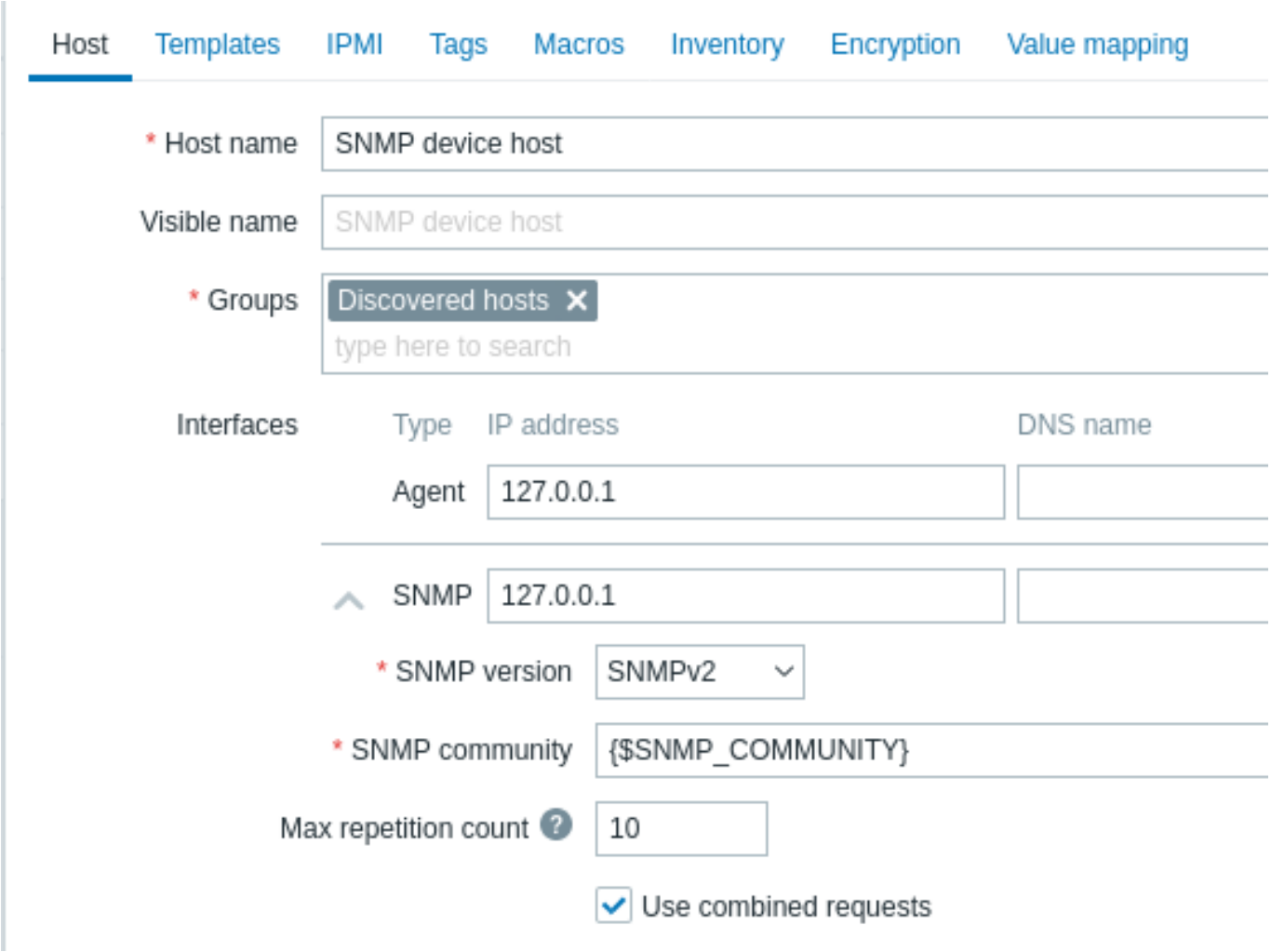
Novamente, o último número no OID é o número da porta.

Note:
Alguns dos OIDs SNMP mais usados são traduzidos automaticamente para uma representação numérica pelo Zabbix.

No último exemplo acima, o tipo de valor é "Counter64", que internamente corresponde ao tipo ASN_COUNTER64. A lista completa de tipos suportados é ASN_COUNTER, ASN_COUNTER64, ASN_UINTEGER, ASN_UNSIGNED64, ASN_INTEGER, ASN_INTEGER64, ASN_FLOAT, ASN_DOUBLE, ASN_TIMETICKS, ASN_GAUGE, ASN_IPADDRESS, ASN_OCTET_STR e ASN_OBJECT_ID. Esses tipos correspondem aproximadamente a "Counter32", "Counter64", "UInteger32", "INTEGER", "Float", "Double", "Timeticks", "Gauge32", "IpAddress", "OCTET STRING", "OBJECT IDENTIFIER" na saída do **snmpget**, mas também podem ser exibidos como "STRING", "Hex-STRING", "OID" e outros, dependendo da presença de uma dica de exibição.

Passo 2

Crie um host correspondente a um dispositivo.



Adicione uma interface SNMP para o host:

- Insira o endereço IP/nome DNS e o número da porta.
- Selecione a *versão SNMP* no menu suspenso.
- Adicione as credenciais da interface dependendo da versão SNMP selecionada:
  - SNMPv1, v2 requerem apenas a comunidade (geralmente 'public').
  - SNMPv3 requer opções mais específicas (veja abaixo).
- Especifique o valor máximo de repetição (padrão: 10) para requisições nativas SNMP bulk (GetBulkRequest-PDUs); apenas para itens `discovery[]` e `walk[]` em SNMPv2 e v3. Observe que definir esse valor muito alto pode causar timeout na

verificação do agent SNMP.

- Marque a caixa *Usar requisições combinadas* para permitir o processamento combinado de requisições SNMP (não relacionado às requisições nativas SNMP bulk "walk" e "get").

Parâmetro SNMPv3	Descrição
Nome do contexto	Insira o nome do contexto para identificar o item na sub-rede SNMP. Macros de usuário são resolvidas neste campo.
Nome de segurança	Insira o nome de segurança. Macros de usuário são resolvidas neste campo.
Nível de segurança	Selecione o nível de segurança: **noAuthNoPriv** - nem protocolos de autenticação nem de privacidade são usados **AuthNoPriv** - protocolo de autenticação é usado, protocolo de privacidade não **AuthPriv** - ambos os protocolos de autenticação e privacidade são usados
Protocolo de autenticação	Selecione o protocolo de autenticação - *MD5*, *SHA1*; com net-snmp 5.8 e mais recente *SHA224*, *SHA256*, *SHA384* ou *SHA512*.
Senha de autenticação	Insira a senha de autenticação. Macros de usuário são resolvidas neste campo.
Protocolo de privacidade	Selecione o protocolo de privacidade - *DES*, *AES128*, *AES192*, *AES256*, *AES192C* (Cisco) ou *AES256C* (Cisco). Veja as notas sobre suporte ao protocolo de privacidade
Senha de privacidade	Insira a senha de privacidade. Macros de usuário são resolvidas neste campo.

No caso de credenciais SNMPv3 incorretas (nome de segurança, protocolo/senha de autenticação, protocolo de privacidade):

- O Zabbix recebe um ERRO do net-snmp, exceto para *Senha de privacidade* incorreta, caso em que o Zabbix recebe um erro de TIMEOUT do net-snmp.
- A disponibilidade da interface SNMP mudará para vermelho (indisponível).

Note:
Alterações em *Protocolo de autenticação*, *Senha de autenticação*, *Protocolo de privacidade* ou *Senha de privacidade*, feitas sem alterar o *Nome de segurança*, normalmente são aplicadas automaticamente quando a interface SNMPv3 correspondente é atualizada no Zabbix. Nos casos em que o *Nome de segurança* também é alterado, todos os parâmetros serão atualizados imediatamente.

Você pode usar um dos templates SNMP fornecidos que adicionará automaticamente um conjunto de items. Antes de usar um template, verifique se ele é compatível com o host.

Clique em *Adicionar* para salvar o host.

Suporte ao protocolo de privacidade

Dependendo do seu sistema operacional e da configuração do net-snmp, alguns protocolos de privacidade podem não estar disponíveis:

- Em alguns sistemas operacionais mais recentes (por exemplo, RHEL9), o suporte ao DES foi removido do pacote net-snmp.
- Protocolos de criptografia AES192 e superiores não são suportados por padrão em sistemas operacionais mais antigos que RHEL 8, CentOS 8, Oracle Linux 8, Debian 12, Ubuntu LTS 22.04, openSUSE Leap 15.5.

Para verificar se a biblioteca net-snmp suporta AES192+, use uma das seguintes opções:

1. `net-snmp-config`:

```
net-snmp-config --configure-options
```

Se a saída contiver `--enable-blumenthal-aes`, AES192+ é suportado.

Observe que o net-snmp-config faz parte do pacote de desenvolvimento para SNMP (libsnmp-dev para Debian/Ubuntu, net-snmp-devel para CentOS/RHEL/OL/SUSE) e pode não estar instalado por padrão.

2. `snmpget`:

```
snmpget -v 3 -x AES-256
```

Se a saída contiver `Invalid privacy protocol specified after -3x flag: AES-256`, AES192+ não é suportado. Se a saída contiver `No hostname specified.`, AES192+ não é suportado.

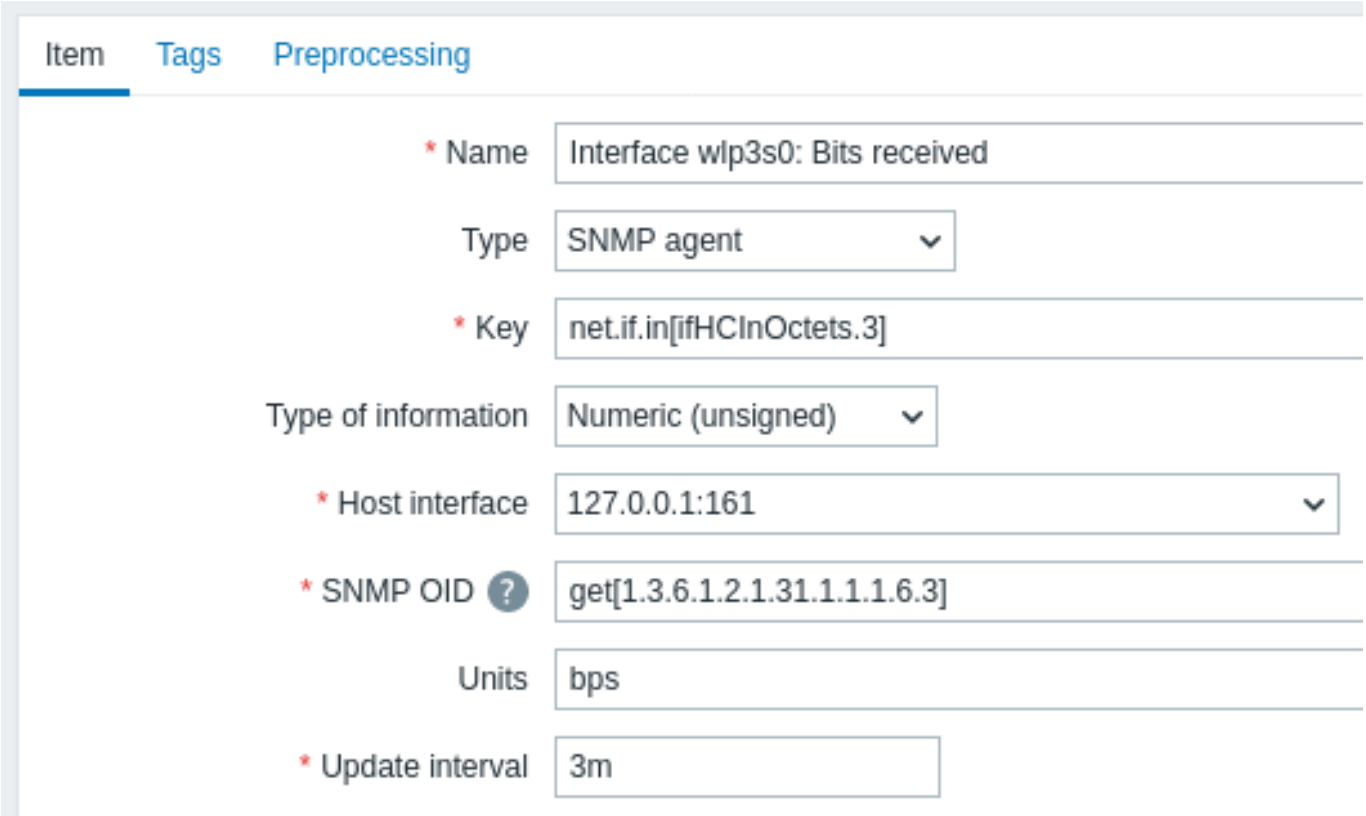
Se sua biblioteca net-snmp não suportar protocolos AES192 e superiores, recompile o net-snmp com a opção `--enable-blumenthal-aes` e, em seguida, recompile o Zabbix server especificando a opção `--with-net-snmp=/home/user/yourcustomnetsnmp/bin/net-snmp-`

Passo 3

Crie um item para monitoramento.

Agora volte ao Zabbix e clique em *Itens* para o host SNMP que você criou anteriormente. Dependendo se você usou um template ou não ao criar seu host, você terá uma lista de itens SNMP associados ao seu host ou apenas uma lista vazia. Vamos trabalhar com a suposição de que você irá criar o item você mesmo usando as informações que acabou de coletar usando snmpwalk e snmpget, então clique em *Criar item*.

Preencha os parâmetros necessários no novo formulário de item:



Parâmetro	Descrição
Nome	Insira o nome do item.
Tipo	Selecione **SNMP agent** aqui.
Chave	Insira a chave como algo significativo.
Interface do host	Certifique-se de selecionar a interface SNMP, por exemplo, do seu switch/roteador.

Parâmetro	Descrição
SNMP OID	Use um dos formatos suportados para inserir o(s) valor(es) do OID: **walk[OID1,OID2,...]** - recupera uma subárvore de valores. Por exemplo: `walk[1.3.6.1.2.1.2.2.1.2,1.3.6.1.2.1.2.2.1.3]`. Esta opção faz uso de requisições SNMP bulk nativas (GetBulkRequest-PDUs) de forma assíncrona. As configurações de timeout para este item podem ser definidas no formulário de configuração do item. Considere definir um valor de timeout baixo para evitar longos atrasos se o dispositivo estiver inacessível, pois até 5 tentativas (desde o Zabbix 7.0.14) serão feitas se as anteriores expirarem ou falharem (por exemplo, um timeout de 3 segundos pode resultar em um tempo de espera de 15 segundos). Você pode usar este item como item mestre, com itens dependentes que extraem dados do mestre usando pré-processamento. É possível especificar vários OIDs em um único snmp walk, como `walk[OID1,OID2,...]` para processar um OID por vez de forma assíncrona. Se a requisição bulk não retornar resultados, será tentado recuperar um único registro sem requisição bulk. Nomes de MIBs são suportados como parâmetros; assim, `walk[1.3.6.1.2.1.2.2.1.2]` e `walk[ifDescr]` retornarão a mesma saída. Se vários OIDs/MIBs forem especificados, ou seja, `walk[ifDescr,ifType,ifPhysAddress]`, então a saída é uma lista concatenada. Requisições GetBulk são usadas com interfaces SNMPv2 e v3 e GetNext para interfaces SNMPv1; o número máximo de repetições para requisições bulk é configurado no nível da interface. O parâmetro de repetições máximas afeta as requisições bulk determinando o número máximo de OIDs retornados em uma única resposta bulk. Um valor mais alto resulta em respostas bulk maiores, reduzindo o número de transmissões necessárias. No entanto, nem todos os dispositivos podem suportar valores muito altos, o que pode causar problemas. Este item retorna a saída do utilitário snmpwalk com os parâmetros -Oe -Ot -On. Você pode usar este item como item mestre em descoberta SNMP. **get[OID]** - recupera um *único* valor de forma assíncrona. Por exemplo: `get[1.3.6.1.2.1.31.1.1.1.6.3]` As configurações de timeout para este item podem ser definidas no formulário de configuração do item. Considere definir um valor de timeout baixo para evitar longos atrasos se o dispositivo estiver inacessível, pois até 5 tentativas (desde o Zabbix 7.0.14) serão feitas se as anteriores expirarem ou falharem (por exemplo, um timeout de 3 segundos pode resultar em um tempo de espera de 15 segundos). **OID** - (legado) insira um único OID textual ou numérico para recuperar um único valor de forma síncrona, opcionalmente combinado com outros valores. Por exemplo: `1.3.6.1.2.1.31.1.1.1.6.3`. Para esta opção, o timeout da checagem do item será igual ao valor definido no arquivo de configuração do server. É **recomendado** usar os itens `walk[OID]` e `get[OID]` para melhor desempenho. Todos os itens `walk[OID]` e `get[OID]` são executados de forma assíncrona - não é necessário receber a resposta de uma requisição antes que outras checagens sejam iniciadas. A resolução de DNS também é assíncrona. A concorrência máxima de checagens assíncronas é 1000 (definida por MaxConcurrentChecksPerPoller). O número de pollers SNMP assíncronos é definido pelo parâmetro StartSNMPPollers. Observe que para estatísticas de tráfego de rede, retornadas por qualquer um dos métodos, uma etapa *Alteração por segundo* deve ser adicionada na guia *Pré-processamento*; caso contrário, você obterá o valor acumulado do dispositivo SNMP em vez da última alteração.

Todos os campos obrigatórios estão marcados com um asterisco vermelho.

Agora salve o item e vá para *Monitoramento* > *Últimos dados* para seus dados SNMP.

Exemplo 1

Exemplo geral:

Parâmetro	Descrição
OID	1.2.3.45.6.7.8.0 (ou .1.2.3.45.6.7.8.0)
Key	<String única a ser usada como referência para triggers> Por exemplo, ”my_param”.

Observe que o OID pode ser fornecido em formato numérico ou string. No entanto, em alguns casos, o OID em string deve ser convertido para representação numérica. A ferramenta snmpget pode ser usada para esse propósito:

```
snmpget -On localhost public enterprises.ucdavis.memory.memTotalSwap.0
```

Exemplo 2

Monitoramento do tempo de atividade:

Parâmetro	Descrição
OID	MIB::sysUpTime.0
Chave	router.uptime
Tipo de valor	Float
Unidades	uptime
Etapa de pré-processamento: Multiplicador personalizado	0.01

Requisições SNMP bulk nativas

O **walk[OID1,OID2,...]** item permite usar a funcionalidade nativa do SNMP para requisições bulk (GetBulkRequest-PDUs), disponível nas versões 2/3 do SNMP.

Uma requisição GetBulk no SNMP executa múltiplas requisições GetNext e retorna o resultado em uma única resposta. Isso pode ser usado tanto para itens SNMP regulares quanto para descoberta SNMP, a fim de minimizar as idas e vindas na rede.

O item SNMP **walk[OID1,OID2,...]** pode ser usado como o item mestre que coleta dados em uma única requisição, com itens dependentes que analisam a resposta conforme necessário usando pré-processamento.

Observe que o uso de requisições SNMP bulk nativas não está relacionado à opção de combinar requisições SNMP, que é a própria maneira do Zabbix de combinar múltiplas requisições SNMP (veja a próxima seção).

Até cinco tentativas (desde o Zabbix 7.0.14) serão feitas para itens SNMP bulk para evitar falhas caso um dos pacotes seja perdido. O timeout para itens SNMP com `get` e `walk` (definido no formulário de configuração do item) é definido para toda a sessão. O timeout é aplicado independentemente de os dados serem recuperados completamente; se os dados forem recebidos parcialmente (por exemplo, os dados são coletados com sucesso apenas para um dos múltiplos OIDs), então o item se torna não suportado com a mensagem ”Only partial data received”. Se o timeout for atingido, uma nova tentativa será feita, o timeout será redefinido e a última requisição será reenviada, permitindo continuar a sessão a partir da última requisição caso um único pacote seja perdido ou chegue tarde demais. Considere definir um valor de timeout baixo para evitar longos atrasos caso o dispositivo esteja inacessível, pois até 5 tentativas serão feitas se as anteriores expirarem ou falharem (por exemplo, um timeout de 3 segundos pode resultar em um tempo de espera de 15 segundos).

Funcionamento interno do processamento combinado

O Zabbix server e o proxy podem consultar dispositivos SNMP para múltiplos valores em uma única requisição. Isso afeta vários tipos de items SNMP:

- items SNMP regulares
- items SNMP com índices dinâmicos
- regras de descoberta de baixo nível SNMP

Todos os items SNMP em uma única interface com parâmetros idênticos são agendados para serem consultados ao mesmo tempo. Os dois primeiros tipos de items são processados pelos pollers em lotes de no máximo 128 items, enquanto as regras de descoberta de baixo nível são processadas individualmente, como antes.

Em um nível mais baixo, existem dois tipos de operações realizadas para consultar valores: obter múltiplos objetos especificados e percorrer uma árvore OID.

Para ”obter”, um GetRequest-PDU é usado com no máximo 128 variáveis. Para ”percorrer”, um GetNextRequest-PDU é usado para SNMPv1 e GetBulkRequest com o campo ”max-repetitions” de no máximo 128 é usado para SNMPv2 e SNMPv3.

Assim, os benefícios do processamento combinado para cada tipo de item SNMP são descritos abaixo:

- items SNMP regulares se beneficiam das melhorias de ”obtenção”;
- items SNMP com índices dinâmicos se beneficiam tanto das melhorias de ”obtenção” quanto de ”percorrer”: ”obtenção” é usada para verificação de índice e ”percorrer” para construir o cache;
- regras de descoberta de baixo nível SNMP se beneficiam das melhorias de ”percorrer”.

No entanto, existe um problema técnico em que nem todos os dispositivos são capazes de retornar 128 valores por requisição. Alguns sempre retornam uma resposta adequada, mas outros respondem com um erro ”tooBig(1)” ou não respondem de forma alguma quando a resposta potencial ultrapassa um determinado limite.

Para encontrar um número ideal de objetos para consultar em um determinado dispositivo, o Zabbix usa a seguinte estratégia. Ele começa cautelosamente consultando 1 valor em uma requisição. Se isso for bem-sucedido, ele consulta 2 valores em uma requisição. Se isso for bem-sucedido novamente, ele consulta 3 valores em uma requisição e continua de forma semelhante, multiplicando o número de objetos consultados por 1,5, resultando na seguinte sequência de tamanhos de requisição: 1, 2, 3, 4, 6, 9, 13, 19, 28, 42, 63, 94, 128.

No entanto, uma vez que um dispositivo se recusa a fornecer uma resposta adequada (por exemplo, para 42 variáveis), o Zabbix faz duas coisas.

Primeiro, para o lote atual de items, ele reduz pela metade o número de objetos em uma única requisição e consulta 21 variáveis. Se o dispositivo estiver ativo, a consulta deve funcionar na grande maioria dos casos, porque 28 variáveis eram conhecidas por funcionar e 21 é significativamente menos do que isso. No entanto, se ainda falhar, o Zabbix volta a consultar os valores um por um. Se ainda falhar neste ponto, então o dispositivo definitivamente não está respondendo e o tamanho da requisição não é um problema.

A segunda coisa que o Zabbix faz para os próximos lotes de items é começar com o último número bem-sucedido de variáveis (28 em nosso exemplo) e continuar incrementando os tamanhos das requisições em 1 até atingir o limite. Por exemplo, supondo que o maior tamanho de resposta seja 32 variáveis, as requisições subsequentes serão de tamanhos 29, 30, 31, 32 e 33. A última requisição falhará e o Zabbix nunca mais fará uma requisição de tamanho 33. A partir desse ponto, o Zabbix consultará no máximo 32 variáveis para esse dispositivo.

Se grandes consultas falharem com esse número de variáveis, isso pode significar uma de duas coisas. Os critérios exatos que um dispositivo usa para limitar o tamanho da resposta não podem ser conhecidos, mas tentamos aproximar isso usando o número de variáveis. Portanto, a primeira possibilidade é que esse número de variáveis esteja próximo do limite real de tamanho de resposta do dispositivo no caso geral: às vezes a resposta é menor que o limite, às vezes é maior. A segunda possibilidade é que um pacote UDP em qualquer direção simplesmente se perdeu. Por esses motivos, se o Zabbix receber uma consulta com falha, ele reduz o número máximo de variáveis para tentar chegar mais fundo na faixa confortável do dispositivo, mas apenas até duas vezes.

No exemplo acima, se uma consulta com 32 variáveis falhar, o Zabbix reduzirá a contagem para 31. Se isso também falhar, o Zabbix reduzirá a contagem para 30. No entanto, o Zabbix não reduzirá a contagem abaixo de 30, porque assumirá que falhas adicionais são devido à perda de pacotes UDP, e não ao limite do dispositivo.

Se, no entanto, um dispositivo não puder lidar com requisições combinadas corretamente por outros motivos e a heurística descrita acima não funcionar, existe uma configuração ”Usar requisições combinadas” para cada interface que permite desabilitar requisições combinadas para esse dispositivo.

Se as requisições combinadas causarem respostas parciais ou malformadas que resultem em cálculos incorretos por segundo (delta) (por exemplo, picos aparentes em contadores de interface), desabilite *Usar requisições combinadas* para a interface afetada para forçar consultas separadas por item; isso geralmente evita picos falsos. Como alternativa, considere usar items assíncronos `get[]` ou `walk[]`, que são executados de forma assíncrona e não estão sujeitos ao agrupamento por interface *Usar requisições combinadas* — eles podem ser usados em vez das verificações OID síncronas legadas para evitar problemas relacionados a requisições combinadas. Procure por entradas de log do server/proxy semelhantes à mostrada na seção Visão geral para ajudar a identificar os dispositivos afetados.

Além disso, se a interface frequentemente se tornar indisponível, pode ser necessário aumentar o parâmetro `UnavailableDelay` nos arquivos de configuração do Zabbix server ou do Zabbix proxy para reduzir a frequência das requisições. Os items podem se tornar não suportados se dados parciais forem recebidos durante a descoberta ou percorrendo OIDs.

1 Índices dinâmicos

Visão geral

Embora você possa encontrar o número de índice necessário (por exemplo, de uma interface de rede) entre os OIDs SNMP, às vezes você pode não confiar completamente que o número de índice sempre permanecerá o mesmo.

Os números de índice podem ser dinâmicos - eles podem mudar ao longo do tempo e seu item pode parar de funcionar como consequência.

Para evitar esse cenário, é possível definir um OID que leve em consideração a possibilidade de um número de índice mudar.

Por exemplo, se você precisar recuperar o valor do índice para anexar ao **ifInOctets** que corresponde à interface **GigabitEthernet0/1** em um dispositivo Cisco, use o seguinte OID:

```
ifInOctets["index","ifDescr","GigabitEthernet0/1"]
```

A sintaxe

Uma sintaxe especial para OID é usada:

<OID dos dados>["index","<OID base do índice>","<string a ser pesquisada>"]

Parâmetro	Descrição
OID dos dados	OID principal a ser usado para a recuperação de dados no item.
index	Método de processamento. Atualmente, um método é suportado: **index** – procura pelo índice e o adiciona ao OID de dados
OID base do índice	Este OID será consultado para obter o valor do índice correspondente à string.
string a ser pesquisada	A string a ser usada para uma correspondência exata com um valor ao fazer a pesquisa. Sensível a maiúsculas e minúsculas.

Exemplo

Obtendo o uso de memória do processo *apache*.

Se usar esta sintaxe OID:

```
HOST-RESOURCES-MIB::hrSWRunPerfMem["index","HOST-RESOURCES-MIB::hrSWRunPath", "/usr/sbin/apache2"]
```

o número do índice será procurado aqui:

```
...
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"
...
```

Agora temos o índice, 5388. O índice será anexado ao OID de dados para receber o valor que nos interessa:

```
HOST-RESOURCES-MIB::hrSWRunPerfMem.5388 = INTEGER: 31468 KBytes
```

Cache de pesquisa de índice

Quando um item de índice dinâmico é solicitado, o Zabbix recupera e armazena em cache toda a tabela SNMP sob o OID base para o índice, mesmo que uma correspondência seja encontrada antes. Isso é feito para o caso de outro item referenciar o mesmo OID base posteriormente - o Zabbix procuraria o índice no cache, em vez de consultar o host monitorado novamente. Observe que cada processo poller usa um cache separado.

Em todas as operações subsequentes de recuperação de valor, apenas o índice encontrado é verificado. Se ele não mudou, o valor é solicitado. Se ele mudou, o cache é reconstruído - cada poller que encontra um índice alterado percorre novamente a tabela SNMP do índice.

2 OIDs Especiais

Alguns dos OIDs SNMP mais usados são traduzidos automaticamente para uma representação numérica pelo Zabbix. Por exemplo, **ifIndex** é traduzido para **1.3.6.1.2.1.2.2.1.1**, **ifIndex.0** é traduzido para **1.3.6.1.2.1.2.2.1.1.0**.

A tabela contém uma lista dos OIDs especiais.

OID especial	Identificador	Descrição
ifIndex	1.3.6.1.2.1.2.2.1.1	Um valor único para cada interface.
ifDescr	1.3.6.1.2.1.2.2.1.2	Uma string textual contendo informações sobre a interface. Esta string deve incluir o nome do fabricante, o nome do produto e a versão da interface de hardware.
ifType	1.3.6.1.2.1.2.2.1.3	O tipo de interface, distinguido de acordo com o(s) protocolo(s) físico(s)/de enlace imediatamente 'abaixo' da camada de rede na pilha de protocolos.

OID especial	Identificador	Descrição
ifMtu	1.3.6.1.2.1.2.2.1.4	O tamanho do maior datagrama que pode ser enviado/recebido na interface, especificado em octetos.
ifSpeed	1.3.6.1.2.1.2.2.1.5	Uma estimativa da largura de banda atual da interface em bits por segundo.
ifPhysAddress	1.3.6.1.2.1.2.2.1.6	O endereço da interface na camada de protocolo imediatamente 'abaixo' da camada de rede na pilha de protocolos.
ifAdminStatus	1.3.6.1.2.1.2.2.1.7	O estado administrativo atual da interface.
ifOperStatus	1.3.6.1.2.1.2.2.1.8	O estado operacional atual da interface.
ifInOctets	1.3.6.1.2.1.2.2.1.10	O número total de octetos recebidos na interface, incluindo caracteres de enquadramento.
ifInUcastPkts	1.3.6.1.2.1.2.2.1.11	O número de pacotes unicast de sub-rede entregues a um protocolo de camada superior.
ifInNUcastPkts	1.3.6.1.2.1.2.2.1.12	O número de pacotes não-unicast (ou seja, broadcast de sub-rede ou multicast de sub-rede) entregues a um protocolo de camada superior.
ifInDiscards	1.3.6.1.2.1.2.2.1.13	O número de pacotes recebidos que foram descartados mesmo que nenhum erro tenha sido detectado para impedir que fossem entregues a um protocolo de camada superior. Uma possível razão para descartar tal pacote pode ser liberar espaço no buffer.
ifInErrors	1.3.6.1.2.1.2.2.1.14	O número de pacotes recebidos que continham erros que impediram sua entrega a um protocolo de camada superior.
ifInUnknownProtos	1.3.6.1.2.1.2.2.1.15	O número de pacotes recebidos pela interface que foram descartados devido a um protocolo desconhecido ou não suportado.
ifOutOctets	1.3.6.1.2.1.2.2.1.16	O número total de octetos transmitidos pela interface, incluindo caracteres de enquadramento.
ifOutUcastPkts	1.3.6.1.2.1.2.2.1.17	O número total de pacotes que protocolos de nível superior solicitaram que fossem transmitidos e que não foram endereçados a um endereço multicast ou broadcast nesta subcamada, incluindo aqueles que foram descartados ou não enviados.
ifOutNUcastPkts	1.3.6.1.2.1.2.2.1.18	O número total de pacotes que protocolos de nível superior solicitaram que fossem transmitidos e que foram endereçados a um endereço multicast ou broadcast nesta subcamada, incluindo aqueles que foram descartados ou não enviados.
ifOutDiscards	1.3.6.1.2.1.2.2.1.19	O número de pacotes enviados que foram descartados mesmo que nenhum erro tenha sido detectado para impedir sua transmissão. Uma possível razão para descartar tal pacote pode ser liberar espaço no buffer.
ifOutErrors	1.3.6.1.2.1.2.2.1.20	O número de pacotes enviados que não puderam ser transmitidos devido a erros.
ifOutQLen	1.3.6.1.2.1.2.2.1.21	O comprimento da fila de pacotes de saída (em pacotes).

3 Arquivos MIB

Introdução

MIB significa Base de Informações de Gerenciamento. Os arquivos MIB permitem usar a representação textual de um OID (Identificador de Objeto). É possível usar OIDs brutos ao monitorar dispositivos SNMP com o Zabbix, mas se você se sentir mais confortável usando a representação textual, precisará instalar arquivos MIB.

Por exemplo,

`ifHCOutOctets`

é a representação textual do OID

`1.3.6.1.2.1.31.1.1.1.10`

Instalando arquivos MIB

Em sistemas baseados em Debian:

```
apt install snmp-mibs-downloader
download-mibs
```

Em sistemas baseados em RedHat:

```
dnf install net-snmp-libs
```

Habilitando arquivos MIB

Em sistemas baseados em RedHat, os arquivos MIB devem estar habilitados por padrão. Em sistemas baseados em Debian, você deve editar o arquivo `/etc/snmp/snmp.conf` e comentar a linha que diz `mibs :`

```
# Como os pacotes snmp vêm sem arquivos MIB por motivos de licença, o carregamento
# de MIBs é desabilitado por padrão. Se você adicionou os MIBs, pode reabilitar
# o carregamento deles comentando a linha a seguir.
mibs :
```

Testando arquivos MIB

O teste de MIBs SNMP pode ser feito usando o utilitário `snmpwalk`. Se você não o tiver instalado, use as instruções a seguir.

Em sistemas baseados em Debian:

```
apt install snmp
```

Em sistemas baseados em RedHat:

```
dnf install net-snmp-utils
```

Depois disso, o seguinte comando não deve apresentar erro ao consultar um dispositivo de rede:

```
$ snmpwalk -v 2c -c public <IP DO DISPOSITIVO DE REDE> 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
[...]
```

Usando MIBs no Zabbix

O mais importante a ter em mente é que os processos do Zabbix não são informados sobre as alterações feitas nos arquivos MIB. Portanto, após cada alteração, você deve reiniciar o Zabbix server ou proxy, por exemplo:

```
systemctl restart zabbix-server
```

Após isso, as alterações feitas nos arquivos MIB entram em vigor.

Usando arquivos MIB personalizados

Existem arquivos MIB padrão que acompanham todas as distribuições GNU/Linux. Mas alguns fornecedores de dispositivos fornecem os seus próprios.

Vamos supor que você queira usar o arquivo MIB CISCO-SMI. As instruções a seguir irão baixá-lo e instalá-lo:

```
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
```

Agora você deve ser capaz de usá-lo. Tente traduzir o nome do objeto *ciscoProducts* do arquivo MIB para OID:

```
snmptranslate -IR -On CISCO-SMI::ciscoProducts
.1.3.6.1.4.1.9.1
```

Se você receber erros em vez do OID, certifique-se de que todos os comandos anteriores não retornaram nenhum erro.

A tradução do nome do objeto funcionou, você está pronto para usar o arquivo MIB personalizado. Observe o prefixo do nome MIB (*CISCO-SMI::*) usado na consulta. Você precisará disso ao usar ferramentas de linha de comando, bem como o Zabbix.

Não se esqueça de reiniciar o Zabbix server/proxy antes de usar este arquivo MIB no Zabbix.

> **Attention:**
> Lembre-se de que os arquivos MIB podem ter dependências. Ou seja, um MIB pode exigir outro. Para satisfazer essas dependências, você deve instalar todos os arquivos MIB afetados.

3 Traps SNMP

Visão geral

Receber traps SNMP é o oposto de consultar dispositivos habilitados para SNMP.

Neste caso, as informações são enviadas de um dispositivo habilitado para SNMP para o snmptrapd e são coletadas ou ”capturadas” pelo Zabbix server ou Zabbix proxy a partir de um arquivo.

Normalmente, traps são enviadas após alguma alteração de condição e o agent conecta-se ao server na porta 162 (em oposição à porta 161 no lado do agent que é usada para consultas). O uso de traps pode detectar alguns problemas rápidos que ocorrem durante o intervalo de consulta e podem ser perdidos pelos dados de consulta.

O recebimento de traps SNMP no Zabbix foi projetado para funcionar com o **snmptrapd** e um dos mecanismos para passar as traps para o Zabbix - seja um script Bash ou Perl ou SNMPTT.

> **Note:**
> A maneira mais simples de configurar o monitoramento de traps após configurar o Zabbix é usar a solução de script Bash, pois Perl e SNMPTT geralmente não estão presentes em distribuições modernas e exigem uma configuração mais complexa. No entanto, esta solução usa um script configurado como `traphandle`. Para melhor desempenho em sistemas de produção, use a solução Perl embutida (seja script com a opção do `perl` ou SNMPTT).

O fluxo de trabalho para receber uma trap:

1. O `snmptrapd` recebe uma trap
2. O `snmptrapd` passa a trap para o script receptor (Bash, Perl) ou SNMPTT
3. O receptor analisa, formata e grava a trap em um arquivo
4. O Zabbix SNMP trapper lê e analisa o arquivo de trap
5. Para cada trap, o Zabbix encontra todos os items ”SNMP trapper” com interfaces de host correspondentes ao endereço da trap recebida. Observe que apenas o ”IP” ou ”DNS” selecionado na interface do host é usado durante a correspondência.
6. Para cada item encontrado, a trap é comparada ao regexp em `snmptrap[regexp]`. A trap é definida como o valor de **todos** os items correspondentes. Se nenhum item correspondente for encontrado e houver um item `snmptrap.fallback`, a trap é definida como o valor desse item.
7. Se a trap não foi definida como valor de nenhum item, o Zabbix, por padrão, registra a trap não correspondida. (Isso é configurado por ”Registrar traps SNMP não correspondidas” em Administração > Geral > Outros.)

Notas sobre failover de HA

Durante a troca de nó em alta disponibilidade (HA), o Zabbix continuará o processamento após o último registro dentro do último timestamp ISO 8601; se o mesmo registro não for encontrado, apenas o timestamp será usado para identificar a última posição.

Configurando traps SNMP

Este tipo de item requer a seguinte configuração no frontend.

1. Crie uma interface SNMP para seu host

- Em *Coleta de dados* > *Hosts*, crie/edite o host e, no campo *Interfaces*, adicione o tipo de interface ”SNMP”, especificando o endereço IP ou DNS.

O endereço de cada trap recebida será comparado aos endereços IP e DNS de todas as interfaces SNMP para encontrar os hosts correspondentes.

2. Configure o item

- Em *Coleta de dados* > *Hosts*, crie/edite o item necessário.
- No campo *Chave*, use uma das chaves de trap SNMP:

Chave		
Descrição	Valor de retorno	Comentários
snmptrap[regexp] Captura todas as traps SNMP que correspondem à expressão regular especificada em **regexp**. Se regexp não for especificado, captura qualquer trap.	Trap SNMP	Este item só pode ser definido para interfaces SNMP. Macros de usuário e expressões regulares globais são suportadas no parâmetro desta chave de item.
snmptrap.fallback		

Chave		
Captura todas as traps SNMP que não foram capturadas por nenhum dos itens snmptrap[] para essa interface.	Trap SNMP	Este item só pode ser definido para interfaces SNMP.

Note:
A correspondência de expressões regulares multilinha não é suportada no momento.

- Defina o *Tipo de informação* como "Log" para que os timestamps sejam analisados. Outros formatos, como "Numérico", também são aceitáveis, mas podem exigir um manipulador de trap personalizado.

Configurando o monitoramento de traps SNMP

Configurando o Zabbix server/proxy

Para ler os traps, o Zabbix server ou proxy deve ser configurado para iniciar o processo SNMP trapper e apontar para o arquivo de trap que está sendo escrito pelo SNMPTT ou por um receptor de trap Bash/Perl. Para isso, edite o arquivo de configuração (zabbix_server.conf ou zabbix_proxy.conf):

```
StartSNMPTrapper=1
SNMPTrapperFile=[TRAP FILE]
```

Warning:
Se o parâmetro systemd **PrivateTmp** for usado, é improvável que este arquivo funcione em */tmp*.

Configurando o receptor de trap Bash

Requisitos: apenas snmptrapd.

Um script receptor de trap Bash pode ser usado para passar traps para o Zabbix server a partir do snmptrapd usando o arquivo trapper. Para configurá-lo, adicione a opção `traphandle` ao arquivo de configuração do snmptrapd (`snmptrapd.conf`), veja o exemplo.

Note:
O snmptrapd pode precisar ser reiniciado para aplicar as alterações em sua configuração.

Configurando o receptor de traps Perl

Requisitos: Perl, Net-SNMP compilado com --enable-embedded-perl (feito por padrão desde o Net-SNMP 5.4)

Um receptor de traps Perl (procure por misc/snmptrap/zabbix_trap_receiver.pl) pode ser usado para passar traps diretamente para o Zabbix server a partir do snmptrapd. Para configurá-lo:

- adicione o script Perl ao arquivo de configuração do snmptrapd (snmptrapd.conf), por exemplo:

```
perl do "[CAMINHO COMPLETO PARA O SCRIPT RECEPTOR]";
```

- configure o receptor, por exemplo:

```
$SNMPTrapperFile = '[ARQUIVO DE TRAP]';
$DateTimeFormat = '[FORMATO DE DATA E HORA]';
```

Note:
O snmptrapd pode precisar ser reiniciado para aplicar as alterações em sua configuração.

Note:
Se o nome do script não estiver entre aspas, o snmptrapd se recusará a iniciar com mensagens semelhantes a estas:


```
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
```

Configurando o SNMPTT

Primeiro, o snmptrapd deve ser configurado para usar o SNMPTT.

> **Note:**
> Para melhor desempenho, o SNMPTT deve ser configurado como um daemon usando **snmptthandler-embedded** para passar os traps para ele. Veja instruções para configurar o SNMPTT.

Quando o SNMPTT estiver configurado para receber os traps, configure o `snmptt.ini`:

1. habilite o uso do módulo Perl do pacote NET-SNMP:

```
net_snmp_perl_enable = 1
```

2. registre os traps no arquivo de traps que será lido pelo Zabbix:

```
log_enable = 1
log_file = [TRAP FILE]
```

3. defina o formato de data e hora:

```
date_time_format = %Y-%m-%dT%H:%M:%S%z
```

> **Warning:**
> O pacote "net-snmp-perl" foi removido no RHEL 8.0-8.2; re-adicionado no RHEL 8.3. Para mais informações, veja os problemas conhecidos.

Agora formate os traps para que o Zabbix possa reconhecê-los (edite o snmptt.conf):

1. Cada instrução FORMAT deve começar com "ZBXTRAP [address]", onde [address] será comparado aos endereços IP e DNS das interfaces SNMP no Zabbix. Exemplo:

```
EVENT coldStart .1.3.6.1.6.3.1.1.5.1 "Status Events" Normal
FORMAT ZBXTRAP $aA Device reinitialized (coldStart)
```

2. Veja mais sobre o formato do trap SNMP abaixo.

> **Attention:**
> Não use traps desconhecidos - o Zabbix não será capaz de reconhecê-los. Traps desconhecidos podem ser tratados definindo um evento geral no snmptt.conf:

> `EVENT general .* "General event" Normal`

Formato do SNMP trap

Todos os receptores de trap Perl personalizados e a configuração de trap SNMPTT devem formatar o trap da seguinte maneira:

```
[timestamp] [o trap, parte 1] ZBXTRAP [endereço] [o trap, parte 2]
```

onde

- [timestamp] - o timestamp no formato "%Y-%m-%dT%H:%M:%S%z"
- ZBXTRAP - cabeçalho que indica que um novo trap começa nesta linha
- [endereço] - endereço IP usado para encontrar o host para este trap

Observe que "ZBXTRAP" e "[endereço]" serão removidos da mensagem durante o processamento. Se o trap for formatado de outra forma, o Zabbix pode analisar os traps de forma inesperada.

Exemplo de trap:

```
2024-01-11T15:28:47+0200 .1.3.6.1.6.3.1.1.5.3 Normal "Status Events" localhost - ZBXTRAP 192.168.1.1 Link
```

Isso resultará no seguinte trap para a interface SNMP com IP=192.168.1.1:

```
2024-01-11T15:28:47+0200 .1.3.6.1.6.3.1.1.5.3 Normal "Status Events"
localhost - Link down on interface 2. Admin state: 1. Operational state: 2
```

Requisitos do sistema

Note:
Recomenda-se instalar arquivos MIB para garantir que os valores dos items sejam exibidos no formato correto. Sem os arquivos MIB, podem ocorrer problemas de formatação, como exibir valores em HEX em vez de UTF-8 ou vice-versa.

Suporte a arquivos grandes

O Zabbix possui suporte a arquivos grandes para arquivos SNMP trapper. O tamanho máximo de arquivo que o Zabbix pode ler é 2^63 (8 EiB). Observe que o sistema de arquivos pode impor um limite inferior ao tamanho do arquivo.

Rotação de logs

O Zabbix não fornece nenhum sistema de rotação de logs - isso deve ser tratado pelo usuário. A rotação de logs deve primeiro renomear o arquivo antigo e só depois excluí-lo para que nenhum trap seja perdido:

1. O Zabbix abre o arquivo de trap na última localização conhecida e vai para o passo 3
2. O Zabbix verifica se o arquivo atualmente aberto foi rotacionado comparando o número do inode com o número do inode do arquivo de trap definido. Se não houver arquivo aberto, o Zabbix redefine a última localização e vai para o passo 1.
3. O Zabbix lê os dados do arquivo atualmente aberto e define a nova localização.
4. Os novos dados são processados. Se este era o arquivo rotacionado, o arquivo é fechado e volta para o passo 2.
5. Se não houver novos dados, o Zabbix aguarda 1 segundo e volta para o passo 2.

Sistema de arquivos

Devido à implementação do arquivo trap, o Zabbix precisa que o sistema de arquivos suporte inodes para diferenciar arquivos (a informação é adquirida por uma chamada stat()).

Exemplos de configuração usando diferentes versões do protocolo SNMP

Este exemplo usa o snmptrapd e um script Bash receptor para passar traps para o Zabbix server.

Configuração:

1. Configure o Zabbix para iniciar o SNMP trapper e defina o arquivo de trap. Adicione ao `zabbix_server.conf`:

```
StartSNMPTrapper=1
SNMPTrapperFile=/var/lib/zabbix/snmptraps/snmptraps.log
```

2. Baixe o script Bash para `/usr/sbin/zabbix_trap_handler.sh`:

```
curl -o /usr/sbin/zabbix_trap_handler.sh https://raw.githubusercontent.com/zabbix/zabbix-docker/7.0/Docker
```

Se necessário, ajuste a variável ZABBIX_TRAPS_FILE no script. Para usar o valor padrão, crie primeiro o diretório pai:

```
mkdir -p /var/lib/zabbix/snmptraps
```

3. Adicione o seguinte ao `snmtrapd.conf` (consulte o exemplo funcional)

```
traphandle default /bin/bash /usr/sbin/zabbix_trap_handler.sh
```

Note:
Pode ser necessário reiniciar o snmptrapd para que ele reconheça as alterações em sua configuração.

4. Crie um item SNMP TEST (lembre-se dos requisitos de configuração iniciais):

   Tipo: SNMP trap
 Tipo de informação: Log
 Interface do host: SNMP 127.0.0.1
 Chave: `snmptrap["linkUp"]`
 Formato de hora do log: yyyyMMdd.hhmmss

Observe que o formato de data e hora ISO 8601 é usado.

5. Em seguida, vamos configurar o `snmptrapd` para a versão do protocolo SNMP escolhida e enviar traps de teste usando o utilitário `snmptrap`.

SNMPv1, SNMPv2

Os protocolos SNMPv1 e SNMPv2 dependem da autenticação por "community string". No exemplo abaixo, usaremos "secret" como community string. Ela deve ser definida com o mesmo valor nos emissores de SNMP trap.

Observe que, embora ainda seja amplamente utilizado em ambientes de produção, o SNMPv2 não oferece nenhuma criptografia e autenticação real do remetente. Os dados são enviados como texto simples e, portanto, essas versões do protocolo só devem ser usadas em ambientes seguros, como redes privadas, e nunca devem ser usadas em qualquer rede pública ou de terceiros.

A versão 1 do SNMP não é realmente usada atualmente, pois não suporta contadores de 64 bits e é considerada um protocolo legado.

Para habilitar a aceitação de traps SNMPv1 ou SNMPv2, adicione a seguinte linha ao `snmptrapd.conf`. Substitua "secret" pela community string SNMP configurada nos emissores de SNMP trap:

```
authCommunity log,execute,net secret
```

Em seguida, podemos enviar um trap de teste usando o `snmptrap`. Usaremos o OID comum "link up" neste exemplo:

```
snmptrap -v 2c -c secret localhost '' linkUp.0
```

SNMPv3

O SNMPv3 resolve os problemas de segurança do SNMPv1/v2 e fornece autenticação e criptografia. Você pode usar os métodos de autenticação MD5 ou múltiplos SHA e DES/múltiplos AES como cifra.

Para habilitar a aceitação de SNMPv3, adicione as seguintes linhas ao `snmptrapd.conf`:

```
createUser -e 0x8000000001020304 traptest SHA mypassword AES
authuser log,execute traptest
```

Attention:
Observe a palavra-chave "execute" que permite executar scripts para este modelo de segurança de usuário.

```
snmptrap -v 3 -n "" -a SHA -A mypassword -x AES -X mypassword -l authPriv -u traptest -e 0x800000000102030
```

Warning:
Se você deseja usar métodos de criptografia fortes, como AES192 ou AES256, use o net-snmp a partir da versão 5.8. Pode ser necessário recompilá-lo com a opção `configure`: `--enable-blumenthal-aes`. Versões mais antigas do net-snmp não suportam AES192/AES256. Veja também: Strong Authentication or Encryption.

Verificação

Em ambos os exemplos, você verá linhas semelhantes em seu `/var/lib/zabbix/snmptraps/snmptraps.log`:

```
2024-01-30T10:04:23+0200 ZBXTRAP 127.0.0.1
UDP: [127.0.0.1]:56585->[127.0.0.1]:162
DISMAN-EVENT-MIB::sysUpTimeInstance = 2538834
SNMPv2-MIB::snmpTrapOID.0 = IF-MIB::linkUp.0
```

O valor do item no Zabbix será:

```
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
```

Exemplo com 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
```

Veja também

- Artigo do blog Zabbix sobre traps SNMP
- Configurando snmptrapd (documentação oficial do net-snmp)
- Configurando snmptrapd para receber notificações SNMPv3 (documentação oficial do net-snmp)

4 Verificações IPMI

Visão geral

Você pode monitorar a integridade e a disponibilidade de dispositivos Intelligent Platform Management Interface (IPMI) no Zabbix. Para realizar verificações IPMI, o Zabbix server deve ser inicialmente configurado com suporte a IPMI.

IPMI é uma interface padronizada para gerenciamento remoto ”lights-out” ou ”out-of-band” de sistemas computacionais. Ela permite monitorar o status do hardware diretamente dos chamados cartões de gerenciamento ”out-of-band”, independentemente do sistema operacional ou se a máquina está ligada.

O monitoramento IPMI do Zabbix funciona apenas para dispositivos com suporte a IPMI (HP iLO, DELL DRAC, IBM RSA, Sun SSP, etc).

Um processo gerenciador IPMI agenda as verificações IPMI por meio de pollers IPMI. Um host é sempre consultado por apenas um poller IPMI por vez, reduzindo o número de conexões abertas aos controladores BMC. Assim, é seguro aumentar o número de pollers IPMI sem se preocupar com a sobrecarga do controlador BMC. O processo gerenciador IPMI é iniciado automaticamente quando pelo menos um poller IPMI é iniciado.

Veja também problemas conhecidos para verificações IPMI.

Configuração

Configuração do host

Um host deve ser configurado para processar verificações IPMI. Uma interface IPMI deve ser adicionada, com os respectivos números de IP e porta, e os parâmetros de autenticação IPMI devem ser definidos.

Consulte a configuração de hosts para obter mais detalhes.

Configuração do servidor

Por padrão, o servidor Zabbix não está configurado para iniciar nenhum poller IPMI, portanto, quaisquer itens IPMI adicionados não funcionarão. Para alterar isso, abra o arquivo de configuração do servidor Zabbix (zabbix_server.conf) como root e procure a seguinte linha:

```
# StartIPMIPollers=0
```

Descomente-a e defina a contagem de pollers para, por exemplo, 3, para que fique assim:

```
StartIPMIPollers=3
```

Salve o arquivo e reinicie o zabbix_server em seguida.

Configuração de item

Ao configurar um item no nível do host:

- Selecione ’IPMI agent’ como o *Tipo*
- Insira uma chave de item que seja única dentro do host (por exemplo, ipmi.fan.rpm)
- Para *Interface do host*, selecione a interface IPMI relevante (IP e porta). Observe que uma interface IPMI deve existir no host.
- Especifique o *Sensor IPMI* (por exemplo, ’FAN MOD 1A RPM’ em um Dell Poweredge) de onde a métrica será obtida. Por padrão, o ID do sensor deve ser especificado. Também é possível usar prefixos antes do valor:
  - `id:` - para especificar o ID do sensor;
  - `name:` - para especificar o nome completo do sensor. Isso pode ser útil em situações em que os sensores só podem ser diferenciados especificando o nome completo.
- Selecione o respectivo tipo de informação (’Numérico (float)’ neste caso; para sensores discretos - ’Numérico (sem sinal)’), unidades (provavelmente ’rpm’) e quaisquer outros atributos de item necessários

Verificações suportadas

O agent IPMI suporta o item interno **ipmi.get**, que retorna informações relacionadas ao sensor IPMI e pode ser usado para a descoberta de sensores IPMI.
 Valor de retorno: *objeto JSON*

Tempo limite e término da sessão

Os tempos limite de mensagens IPMI e as contagens de tentativas são definidos na biblioteca OpenIPMI. Devido ao design atual do OpenIPMI, não é possível tornar esses valores configuráveis no Zabbix, nem no nível da interface nem no nível do item.

O tempo limite de inatividade da sessão IPMI para LAN é de 60 +/-3 segundos. Atualmente, não é possível implementar o envio periódico do comando Activate Session com o OpenIPMI. Se não houver verificações de item IPMI do Zabbix para um BMC específico por mais tempo do que o tempo limite de sessão configurado no BMC, então a próxima verificação IPMI após o tempo limite expirar irá expirar devido a tempos limite de mensagens individuais, tentativas ou erro de recebimento. Após isso, uma nova sessão é aberta e uma nova varredura completa do BMC é iniciada. Uma nova porta UDP pode ser aberta para gerenciar a nova

sessão. A inatividade é definida pela ausência de solicitações de saída e respostas de entrada. Se você quiser evitar varreduras desnecessárias do BMC, é aconselhável definir o intervalo de coleta do item IPMI abaixo do tempo limite de inatividade da sessão IPMI configurado no BMC.

Notas sobre sensores discretos IPMI

Para encontrar sensores em um host, inicie o Zabbix server com **DebugLevel=4** habilitado. Aguarde alguns minutos e encontre registros de descoberta de sensores no arquivo de log do Zabbix server:

```
$ grep 'Added sensor' zabbix_server.log
8358:20130318:111122.170 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:7 id:'CATERR' reading_type:
8358:20130318:111122.170 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:15 id:'CPU Therm Trip' read
8358:20130318:111122.171 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:17 id:'System Event Log' re
8358:20130318:111122.171 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:17 id:'PhysicalSecurity' re
8358:20130318:111122.171 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:14 id:'IPMI Watchdog' readi
8358:20130318:111122.171 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:16 id:'Power Unit Stat' rea
8358:20130318:111122.171 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:16 id:'P1 Therm Ctrl %' rea
8358:20130318:111122.172 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:16 id:'P1 Therm Margin' rea
8358:20130318:111122.172 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:13 id:'System Fan 2' readin
8358:20130318:111122.172 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:13 id:'System Fan 3' readin
8358:20130318:111122.172 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:14 id:'P1 Mem Margin' readi
8358:20130318:111122.172 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:17 id:'Front Panel Temp' re
8358:20130318:111122.173 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:15 id:'Baseboard Temp' read
8358:20130318:111122.173 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:9 id:'BB +5.0V' reading_typ
8358:20130318:111122.173 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:14 id:'BB +3.3V STBY' readi
8358:20130318:111122.173 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:9 id:'BB +3.3V' reading_typ
8358:20130318:111122.173 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:17 id:'BB +1.5V P1 DDR3' re
8358:20130318:111122.173 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:17 id:'BB +1.1V P1 Vccp' re
8358:20130318:111122.174 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:14 id:'BB +1.05V PCH' readi
```

Para decodificar tipos e estados de sensores IPMI, uma cópia das especificações IPMI 2.0 está disponível (observe que não estão previstos novos updates para a especificação IPMI).

O primeiro parâmetro a ser analisado é ”reading_type”. Use a ”Tabela 42-1, Event/Reading Type Code Ranges” das especificações para decodificar o código ”reading_type”. A maioria dos sensores em nosso exemplo tem ”reading_type:0x1”, o que significa sensor ”threshold”. A ”Tabela 42-3, Sensor Type Codes” mostra que ”type:0x1” significa sensor de temperatura, ”type:0x2” - sensor de voltagem, ”type:0x4” - Fan, etc. Sensores threshold às vezes são chamados de sensores ”analógicos”, pois medem parâmetros contínuos como temperatura, voltagem, rotações por minuto.

Outro exemplo - um sensor com ”reading_type:0x3”. A ”Tabela 42-1, Event/Reading Type Code Ranges” diz que códigos de tipo de leitura 02h-0Ch significam sensor ”Generic Discrete”. Sensores discretos têm até 15 possíveis estados (em outras palavras - até 15 bits significativos). Por exemplo, para o sensor ’CATERR’ com ”type:0x7”, a ”Tabela 42-3, Sensor Type Codes” mostra que esse tipo significa ”Processor” e o significado dos bits individuais é: 00h (o bit menos significativo) - IERR, 01h - Thermal Trip, etc.

Há alguns sensores com ”reading_type:0x6f” em nosso exemplo. Para esses sensores, a ”Tabela 42-1, Event/Reading Type Code Ranges” recomenda usar a ”Tabela 42-3, Sensor Type Codes” para decodificar o significado dos bits. Por exemplo, o sensor ’Power Unit Stat’ tem o tipo ”type:0x9”, que significa ”Power Unit”. O offset 00h significa ”PowerOff/Power Down”. Em outras palavras, se o bit menos significativo for 1, então o servidor está desligado. Para testar esse bit, a função `bitand` com máscara ’1’ pode ser usada. A expressão do trigger pode ser como

```
bitand(last(/www.example.com/Power Unit Stat,#1),1)=1
```

para avisar sobre um desligamento do servidor.

Notas sobre nomes de sensores discretos no OpenIPMI-2.0.16, 2.0.17, 2.0.18 e 2.0.19

Os nomes dos sensores discretos no OpenIPMI-2.0.16, 2.0.17 e 2.0.18 frequentemente têm um ”0” adicional (ou algum outro dígito ou letra) adicionado ao final. Por exemplo, enquanto o `ipmitool` e o OpenIPMI-2.0.19 exibem os nomes dos sensores como ”`PhysicalSecurity`” ou ”`CATERR`”, no OpenIPMI-2.0.16, 2.0.17 e 2.0.18 os nomes são ”`PhysicalSecurity0`” ou ”`CATERR0`”, respectivamente.

Ao configurar um item IPMI com o Zabbix server usando o OpenIPMI-2.0.16, 2.0.17 e 2.0.18, use esses nomes terminados com ”0” no campo *IPMI sensor* dos itens do agente IPMI. Quando seu Zabbix server for atualizado para uma nova distribuição Linux, que utiliza o OpenIPMI-2.0.19 (ou posterior), os itens com esses sensores discretos IPMI se tornarão ”NÃO SUPORTADOS”. Você deve alterar os nomes dos *IPMI sensor* (remover o ’0’ no final) e aguardar algum tempo até que eles voltem a ficar ”Habilitados”.

Observações sobre a disponibilidade simultânea de sensores de limite e sensores discretos

Alguns agents IPMI fornecem tanto um sensor de limite quanto um sensor discreto com o mesmo nome. A preferência é sempre dada ao sensor de limite.

Observações sobre a terminação da conexão

Se as verificações IPMI não forem realizadas (por qualquer motivo: todos os itens IPMI do host desativados/não suportados, host desativado/excluído, host em manutenção, etc.), a conexão IPMI será encerrada pelo Zabbix server ou proxy em 3 a 4 horas, dependendo do momento em que o Zabbix server/proxy foi iniciado.

5 Verificações simples

Visão geral

As verificações simples são normalmente usadas para verificações remotas sem agent de serviços.

Observe que o agent Zabbix não é necessário para verificações simples. O server/proxy Zabbix é responsável pelo processamento das verificações simples (realizando conexões externas, etc).

Exemplos de uso de verificações simples:

```
net.tcp.service[ftp,,155]
net.tcp.service[http]
net.tcp.service.perf[http,,8080]
net.udp.service.perf[ntp]
```

Note:
Os campos *Nome de usuário* e *Senha* (limitados a 255 caracteres) na configuração do item de verificação simples são usados para itens de monitoramento VMware; ignorados caso contrário.

Verificações suportadas

As chaves de item são listadas sem parâmetros opcionais e informações adicionais. Clique na chave do item para ver os detalhes completos.

Veja também chaves de item de monitoramento VMware.

Chave do item	Descrição
icmpping	Acessibilidade do host por ICMP ping.
icmppingloss	A porcentagem de pacotes perdidos.
icmppingsec	O tempo de resposta do ICMP ping.
net.tcp.service	Verifica se um serviço está em execução e aceitando conexões TCP.
net.tcp.service.perf	Verifica o desempenho de um serviço TCP.
net.udp.service	Verifica se um serviço está em execução e respondendo a solicitações UDP.
net.udp.service.perf	Verifica o desempenho de um serviço UDP.

Detalhes da chave do item

Parâmetros sem colchetes angulares são obrigatórios. Parâmetros marcados com colchetes angulares **< >** são opcionais.

icmpping[<target>,<packets>,<interval>,<size>,<timeout>,<options>]

 Acessibilidade do host por ICMP ping.
 Valor de retorno: *0* - ICMP ping falhou; *1* - ICMP ping bem-sucedido.

Parâmetros:

- **target** - o IP do host ou nome DNS;
- **packets** - o número de pacotes;
- **interval** - o tempo entre pacotes sucessivos em milissegundos;
- **size** - o tamanho do pacote em bytes;
- **timeout** - o tempo limite em milissegundos;
- **options** - usado para permitir redirecionamento: se vazio (valor padrão), respostas redirecionadas são tratadas como host de destino inativo; se definido como *allow_redirect*, respostas redirecionadas são tratadas como host de destino ativo.

Veja também a tabela de valores padrão.

Exemplo:

```
icmpping[,4] #Se pelo menos um pacote dos quatro for retornado, o item retornará 1.
```

icmppingloss[<target>,<packets>,<interval>,<size>,<timeout>,<options>]

 A porcentagem de pacotes perdidos.
 Valor de retorno: *Float*.

Parâmetros:

- **target** - o IP ou nome DNS do host;
- **packets** - o número de pacotes;
- **interval** - o tempo entre pacotes sucessivos em milissegundos;
- **size** - o tamanho do pacote em bytes;
- **timeout** - o tempo limite em milissegundos;
- **options** - usado para permitir redirecionamento: se vazio (valor padrão), respostas redirecionadas são tratadas como host de destino inativo; se definido como *allow_redirect*, respostas redirecionadas são tratadas como host de destino ativo.

Veja também a tabela de valores padrão.

icmppingsec[<target>,<packets>,<interval>,<size>,<timeout>,<mode>,<options>]

 O tempo de resposta do ping ICMP (em segundos).
 Valor de retorno: *Float*.

Parâmetros:

- **target** - o IP ou nome DNS do host;
- **packets** - o número de pacotes;
- **interval** - o tempo entre pacotes sucessivos em milissegundos;
- **size** - o tamanho do pacote em bytes;
- **timeout** - o tempo limite em milissegundos;
- **mode** - valores possíveis: *min*, *max* ou *avg* (padrão);
- **options** - usado para permitir redirecionamento: se vazio (valor padrão), respostas redirecionadas são tratadas como host de destino inativo; se definido como *allow_redirect*, respostas redirecionadas são tratadas como host de destino ativo.

Comentários:

- Pacotes que são perdidos ou atingem o tempo limite não são usados no cálculo;
- Se o host não estiver disponível (tempo limite atingido), o item retornará 0;
- Se o valor de retorno for menor que 0,0001 segundos, o valor será definido como 0,0001 segundos;
- Veja também a tabela de valores padrão.

net.tcp.service[service,<ip>,<port>]

 Verifica se um serviço está em execução e aceitando conexões TCP.
 Valor de retorno: *0* - o serviço está inativo; *1* - o serviço está em execução.

Parâmetros:

- **service** - valores possíveis: *ssh*, *ldap*, *smtp*, *ftp*, *http*, *pop*, *nntp*, *imap*, *tcp*, *https*, *telnet* (veja detalhes);
- **ip** - o endereço IP ou nome DNS (por padrão, o IP/DNS do host é usado);
- **port** - o número da porta (por padrão, o número da porta padrão do serviço é usado).

Comentários:

- Observe que com o serviço *tcp* é obrigatório indicar a porta;
- Essas verificações podem resultar em mensagens adicionais nos arquivos de log do daemon do sistema (sessões SMTP e SSH geralmente são registradas);
- A verificação de protocolos criptografados (como IMAP na porta 993 ou POP na porta 995) atualmente não é suportada. Como alternativa, use `net.tcp.service[tcp,<ip>,port]` para verificações como essas.

Exemplo:

```
net.tcp.service[ftp,,45] #Este item pode ser usado para testar a disponibilidade do servidor FTP na porta
```

Attention:
Se o SELinux estiver em modo de reforço, verificações simples personalizadas de TCP/UDP podem ser bloqueadas pela política. Para verificar e permitir a nova conexão de saída, revise as negações de auditoria: com `grep denied /var/log/audit/audit.log`

net.tcp.service.perf[service,<ip>,<port>]

 Verifica o desempenho de um serviço TCP.
 Valor de retorno: *Float*: *0.000000* - o serviço está inativo; *segundos* - o número de segundos gastos ao conectar-se ao serviço.

Parâmetros:

- **service** - valores possíveis: *ssh*, *ldap*, *smtp*, *ftp*, *http*, *pop*, *nntp*, *imap*, *tcp*, *https*, *telnet* (veja detalhes);
- **ip** - o endereço IP ou nome DNS (por padrão, o IP/DNS do host é usado);
- **port** - o número da porta (por padrão, o número da porta padrão do serviço é usado).

Comentários:

- Observe que com o serviço *tcp* a indicação da porta é obrigatória;
- A verificação de protocolos criptografados (como IMAP na porta 993 ou POP na porta 995) atualmente não é suportada. Como alternativa, use `net.tcp.service[tcp,<ip>,port]` para verificações como essas.

Exemplo:

```
net.tcp.service.perf[ssh] #Este item pode ser usado para testar a velocidade da resposta inicial do servid
```

net.udp.service[service,<ip>,<port>]

 Verifica se um serviço está em execução e respondendo a solicitações UDP.
 Valor de retorno: *0* - o serviço está inativo; *1* - o serviço está em execução.

Parâmetros:

- **service** - valores possíveis: *ntp* (veja detalhes);
- **ip** - o endereço IP ou nome DNS (por padrão, o IP/DNS do host é usado);
- **port** - o número da porta (por padrão, o número da porta padrão do serviço é usado).

Exemplo:

```
net.udp.service[ntp,,45] #Este item pode ser usado para testar a disponibilidade do serviço NTP na porta U
```

net.udp.service.perf[service,<ip>,<port>]

 Verifica o desempenho de um serviço UDP.
 Valor de retorno: *Float*: *0.000000* - o serviço está inativo; *segundos* - o número de segundos gastos aguardando a resposta do serviço.

Parâmetros:

- **service** - valores possíveis: *ntp* (veja detalhes);
- **ip** - o endereço IP ou nome DNS (por padrão, o IP/DNS do host é usado);
- **port** - o número da porta (por padrão, o número da porta padrão do serviço é usado).

Exemplo:

```
net.udp.service.perf[ntp] #Este item pode ser usado para testar o tempo de resposta do serviço NTP.
```

> **Attention:**
> Para suporte a SourceIP em verificações simples de LDAP (por exemplo, `net.tcp.service[ldap]`), é necessário o OpenLDAP versão 2.6.1 ou superior.

Processamento de timeout

O Zabbix não processará uma verificação simples por mais do que os segundos definidos em *Timeout* no formulário de configuração do item. Para itens VMware, o Zabbix não processará uma verificação simples por mais do que os segundos definidos em `Timeout` no arquivo de configuração do Zabbix server ou proxy. Para itens icmpping*, os valores de timeout e tentativas são especificados diretamente na chave do item e não são influenciados pelo parâmetro global `Timeout`. Certifique-se de que esses valores estejam configurados adequadamente na chave do item. Observe que um tempo limite máximo possível codificado de 600 segundos é aplicado aos itens icmpping* (desde o Zabbix 7.0.22).

Pings ICMP

O Zabbix usa o utilitário externo **fping** para processar pings ICMP (**icmpping**, **icmppingloss**, **icmppingsec**).

Instalação

fping não está incluído no Zabbix e precisa ser instalado separadamente:

- Várias plataformas baseadas em Unix possuem o pacote fping em seus repositórios padrão, mas ele não é pré-instalado. Neste caso, você pode usar o gerenciador de pacotes para instalar o fping.
- O Zabbix fornece pacotes fping para RHEL. Observe que esses pacotes são fornecidos sem suporte oficial.
- O fping também pode ser compilado a partir do código-fonte.

Configuração

Especifique o local do fping no parâmetro *FpingLocation* do arquivo de configuração do Zabbix server/proxy (ou o parâmetro *Fping6Location* para usar endereços IPv6).

O fping deve ser executável pelo usuário que executa o Zabbix server/proxy e este usuário deve ter direitos suficientes.

Veja também: Problemas conhecidos para o processamento de verificações simples com versões do fping abaixo de 3.10.

Valores padrão

Valores padrão, limites e descrição dos valores para parâmetros de checagem ICMP:

Parâmetro	Unidade	Descrição	Flag do fping	Padrões definidos por		Limites permitidos pelo Zabbix	
				fping	**Zabbix**	**min**	**max**
packets	número	número de pacotes de requisição enviados para um destino	-C		3	1	10000
interval	milissegundos	tempo de espera entre pacotes sucessivos para um destino individual	-p	1000		20	ilimitado
size	bytes	tamanho do pacote em bytes 56 bytes em x86, 68 bytes em x86_64	-b	56 ou 68		24	65507
timeout	milissegundos	**fping v3.x** - tempo limite para aguardar após o último pacote enviado, afetado pela flag *-C* **fping v4.x** - tempo limite individual para cada pacote	-t	**fping v3.x** - 500 **fping v4.x** e mais recentes - herdado da flag *-p*, mas não mais que 2000		50	ilimitado

Os valores padrão podem diferir ligeiramente dependendo da plataforma e versão.

Além disso, o Zabbix usa as opções do fping *-i intervalo ms* (não confunda com o parâmetro do item *interval* mencionado na tabela acima, que corresponde à opção do fping *-p*) e *-S endereço IP de origem* (ou *-I* em versões mais antigas do fping). Essas opções são detectadas automaticamente executando checagens com diferentes combinações de opções. O Zabbix tenta detectar o valor mínimo em milissegundos que o fping permite usar com *-i* tentando 3 valores: 0, 1 e 10. O valor que primeiro for bem-sucedido é então usado para as checagens ICMP subsequentes. Esse processo é feito por cada processo ICMP pinger individualmente.

As opções do fping detectadas automaticamente são invalidadas a cada hora e detectadas novamente na próxima tentativa de realizar a checagem ICMP. Defina DebugLevel>=4 para visualizar detalhes desse processo no arquivo de log do server ou proxy.

O Zabbix grava os endereços IP a serem verificados por qualquer uma das três chaves *icmpping** em um arquivo temporário, que é então passado para o fping. Se os itens tiverem parâmetros de chave diferentes, apenas aqueles com parâmetros de chave idênticos são gravados em um único arquivo. Todos os endereços IP gravados no mesmo arquivo serão verificados pelo fping em paralelo, então o processo ICMP pinger do Zabbix gastará uma quantidade fixa de tempo, desconsiderando o número de endereços IP no arquivo.

1 Chaves de item de monitoramento do VMware

A lista de chaves de item de monitoramento do VMware foi movida para a seção Monitoramento do VMware.

6 Monitoramento de arquivos de log

Visão geral

O Zabbix pode ser usado para monitoramento centralizado e análise de arquivos de log com/sem suporte à rotação de logs.

Notificações podem ser usadas para alertar os usuários quando um arquivo de log contém determinadas strings ou padrões de string.

Para monitorar um arquivo de log, você deve ter:

- O agent do Zabbix em execução no host

- item de monitoramento de log configurado

Attention:
O limite de tamanho de um arquivo de log monitorado depende do suporte a arquivos grandes.

Configuração

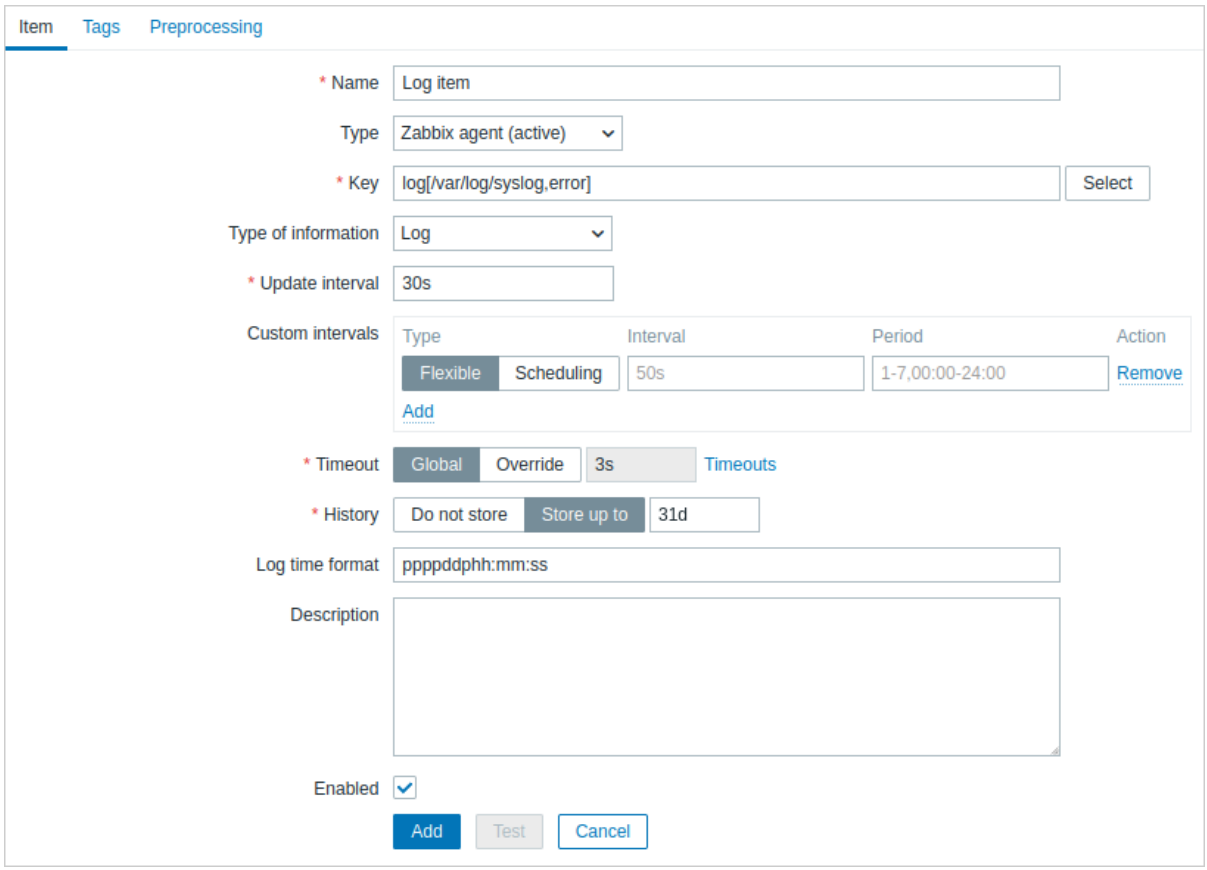
Verifique os parâmetros do agent

Certifique-se de que no arquivo de configuração do agent:

- O parâmetro `Hostname` corresponde ao nome do host no frontend.
- Os servidores no parâmetro `ServerActive` estão especificados para o processamento de verificações ativas.

Configuração do item

Configure um item de monitoramento de log.



Todos os campos obrigatórios estão marcados com um asterisco vermelho.

Especificamente para itens de monitoramento de log, você deve informar:

Tipo	Selecione **Zabbix agent (active)** aqui.
Chave	Use uma das seguintes chaves de item: **log[]** ou **logrt[]**: Essas duas chaves de item permitem monitorar logs e filtrar entradas de log pelo conteúdo da expressão regular, se presente. Por exemplo: `log[/var/log/syslog,error]`. Certifique-se de que o arquivo tenha permissões de leitura para o usuário 'zabbix', caso contrário, o status do item será definido como 'não suportado'. **log.count[]** ou **logrt.count[]**: Essas duas chaves de item permitem retornar apenas o número de linhas correspondentes. Consulte a seção de chaves de item do Zabbix agent suportadas para obter detalhes sobre o uso dessas chaves de item e seus parâmetros.

Tipo de informação	Preenchido automaticamente: Para itens log[] ou logrt[] - `Log`; Para itens log.count[] ou logrt.count[] - `Numérico (sem sinal)`. Se estiver usando opcionalmente o parâmetro `output`, você pode selecionar manualmente o tipo de informação apropriado diferente de `Log`. Observe que escolher um tipo de informação diferente de Log levará à perda do timestamp local.
Intervalo de atualização (em seg)	O parâmetro define com que frequência o Zabbix agent verificará se há alterações no arquivo de log. Definir como 1 segundo garantirá que você receba novos registros o mais rápido possível.
Formato de data/hora do log	Neste campo, você pode especificar opcionalmente o padrão para analisar o timestamp da linha do log. Espaços reservados suportados: * **y**: *Ano (1970-2038)* * **M**: *Mês (01-12)* * **d**: *Dia (01-31)* * **h**: *Hora (00-23)* * **m**: *Minuto (00-59)* * **s**: *Segundo (00-59)* Se deixado em branco, o timestamp será definido como 0 em tempo Unix, representando 1º de janeiro de 1970. Por exemplo, considere a seguinte linha do arquivo de log do Zabbix agent: " 23480:20100328:154718.045 Zabbix agent started. Zabbix 1.8.2 (revision 11211)." Ela começa com seis posições de caracteres para o PID, seguidas por data, hora e o restante da mensagem. O formato de data/hora do log para esta linha seria "pppppp:yyyyMMdd:hhmmss". Observe que os caracteres "p" e ":" são espaços reservados e podem ser qualquer caractere, exceto "yMdhms".

Notas importantes

- O server e o agent mantêm o controle do tamanho do log monitorado e do horário da última modificação (para logrt) em dois contadores. Adicionalmente:
  - O agent também usa internamente números de inode (em UNIX/GNU/Linux), índices de arquivo (em Microsoft Windows) e somas MD5 dos primeiros 512 bytes do arquivo de log para melhorar as decisões quando os arquivos de log são truncados e rotacionados.
  - Em sistemas UNIX/GNU/Linux, presume-se que os sistemas de arquivos onde os arquivos de log estão armazenados reportam números de inode, que podem ser usados para rastrear arquivos.
  - No Microsoft Windows, o agent do Zabbix determina o tipo de sistema de arquivos onde os arquivos de log residem e usa:
    * Em sistemas de arquivos NTFS, índices de arquivos de 64 bits.
    * Em sistemas de arquivos ReFS (apenas a partir do Microsoft Windows Server 2012), IDs de arquivos de 128 bits.
    * Em sistemas de arquivos onde os índices de arquivos mudam (por exemplo, FAT32, exFAT), um algoritmo alternativo é usado para adotar uma abordagem sensata em condições incertas quando a rotação do arquivo de log resulta em vários arquivos de log com o mesmo horário de última modificação.
  - Os números de inode, índices de arquivo e somas MD5 são coletados internamente pelo agent do Zabbix. Eles não são transmitidos ao server do Zabbix e são perdidos quando o agent do Zabbix é parado.
  - Não modifique o horário de última modificação de um arquivo de log (por exemplo, com `touch`) e não substitua um arquivo de log monitorado copiando um arquivo de volta para seu nome original (isso cria um novo inode). Em ambos os casos, o Zabbix pode tratar o arquivo como um arquivo diferente e relê-lo desde o início, o que pode produzir alertas duplicados.
  - Se houver vários arquivos de log correspondentes para o item `logrt[]` e o agent do Zabbix estiver seguindo o mais recente deles e este arquivo de log mais recente for excluído, uma mensagem de aviso `"there are no files matching "<regexp mask>" in "<directory>"` será registrada. O agent do Zabbix ignora arquivos de log com horário de modificação menor que o horário de modificação mais recente visto pelo agent para o item `logrt[]` que está sendo verificado.
- O agent começa a ler o arquivo de log a partir do ponto em que parou da última vez.
- O número de bytes já analisados (o contador de tamanho) e o horário da última modificação (o contador de tempo) são armazenados no banco de dados do Zabbix e enviados ao agent para garantir que o agent comece a ler o arquivo de log a partir desse ponto nos casos em que o agent acabou de ser iniciado ou recebeu items que estavam anteriormente desabilitados ou não suportados. No entanto, se o agent recebeu um contador de tamanho diferente de zero do server, mas o item logrt[] ou logrt.count[] não consegue encontrar arquivos correspondentes, o contador de tamanho é redefinido para 0 para analisar desde o início se os arquivos aparecerem posteriormente.
- Sempre que o arquivo de log se torna menor que o contador de tamanho conhecido pelo agent, o contador é redefinido para zero e o agent começa a ler o arquivo de log desde o início, levando em consideração o contador de tempo.

- Se houver vários arquivos correspondentes com o mesmo horário de última modificação no diretório, o agent tenta analisar corretamente todos os arquivos de log com o mesmo horário de modificação e evitar pular dados ou analisar os mesmos dados duas vezes, embora isso não possa ser garantido em todas as situações. O agent não assume nenhum esquema específico de rotação de arquivos de log nem determina um. Quando apresentados vários arquivos de log com o mesmo horário de última modificação, o agent os processará em ordem lexicográfica decrescente. Assim, para alguns esquemas de rotação, os arquivos de log serão analisados e relatados em sua ordem original. Para outros esquemas de rotação, a ordem original do arquivo de log não será respeitada, o que pode levar ao relato de registros de arquivos de log correspondentes em ordem alterada (o problema não ocorre se os arquivos de log tiverem horários de última modificação diferentes).
- O agent do Zabbix processa novos registros de um arquivo de log uma vez a cada *Intervalo de atualização* segundos.
- O agent do Zabbix não envia mais do que **maxlines** de um arquivo de log por segundo. O limite evita a sobrecarga de recursos de rede e CPU e substitui o valor padrão fornecido pelo parâmetro **MaxLinesPerSecond** no arquivo de configuração do agent.
- Para encontrar a string necessária, o Zabbix processará 10 vezes mais novas linhas do que definido em MaxLinesPerSecond. Assim, por exemplo, se um item `log[]` ou `logrt[]` tiver *Intervalo de atualização* de 1 segundo, por padrão o agent analisará no máximo 200 registros de arquivo de log e enviará no máximo 20 registros correspondentes ao server do Zabbix em uma verificação. Aumentando **MaxLinesPerSecond** no arquivo de configuração do agent ou definindo o parâmetro **maxlines** na chave do item, o limite pode ser aumentado para até 10000 registros de arquivo de log analisados e 1000 registros correspondentes enviados ao server do Zabbix em uma verificação. Se o *Intervalo de atualização* for definido para 2 segundos, os limites para uma verificação serão definidos 2 vezes maiores do que com *Intervalo de atualização* de 1 segundo.
- Além disso, os valores de log e log.count são sempre limitados a 50% do tamanho do buffer de envio do agent, mesmo que não haja valores de log nele. Portanto, para que os valores de **maxlines** sejam enviados em uma conexão (e não em várias conexões), o parâmetro BufferSize do agent deve ser pelo menos maxlines x 2. O agent do Zabbix pode fazer upload de dados durante a coleta de logs e, assim, liberar o buffer, enquanto o agent do Zabbix 2 interromperá a coleta de logs até que os dados sejam enviados e o buffer seja liberado, o que é feito de forma assíncrona.
- Na ausência de items de log, todo o tamanho do buffer do agent é usado para valores que não são de log. Quando os valores de log chegam, eles substituem os valores mais antigos que não são de log conforme necessário, até os 50% designados.
- Para registros de arquivo de log com mais de 256kB, apenas os primeiros 256kB são comparados com a expressão regular e o restante do registro é ignorado. No entanto, se o agent do Zabbix for interrompido enquanto estiver lidando com um registro longo, o estado interno do agent será perdido e o registro longo poderá ser analisado novamente e de forma diferente após o agent ser iniciado novamente.
- Nota especial para separadores de caminho "\": se file_format for "file\.log", então não deve haver um diretório "file", pois não é possível definir de forma inequívoca se "." está escapado ou é o primeiro símbolo do nome do arquivo.
- Expressões regulares para `logrt` são suportadas apenas no nome do arquivo, a correspondência de expressão regular no diretório não é suportada.
- Em plataformas UNIX, um item `logrt[]` se torna NOTSUPPORTED se um diretório onde os arquivos de log devem ser encontrados não existir.
- No Microsoft Windows, se um diretório não existir, o item não se tornará NOTSUPPORTED (por exemplo, se o diretório estiver com erro de digitação na chave do item).
- A ausência de arquivos de log para o item `logrt[]` não o torna NOTSUPPORTED. Erros de leitura de arquivos de log para o item `logrt[]` são registrados como avisos no arquivo de log do agent do Zabbix, mas não tornam o item NOTSUPPORTED.
- O arquivo de log do agent do Zabbix pode ser útil para descobrir por que um item `log[]` ou `logrt[]` se tornou NOTSUPPORTED. O Zabbix pode monitorar seu arquivo de log do agent, exceto quando em DebugLevel=4 ou DebugLevel=5.
- Procurar um ponto de interrogação usando uma expressão regular, por exemplo, `\?` pode resultar em falsos positivos se o arquivo de texto contiver símbolos NUL, pois estes são substituídos por "?" pelo Zabbix para continuar processando a linha até o caractere de nova linha.

Extraindo parte correspondente da expressão regular

Às vezes, podemos querer extrair apenas o valor interessante de um arquivo de destino em vez de retornar toda a linha quando uma correspondência de expressão regular for encontrada.

Itens de log têm a capacidade de extrair valores desejados de linhas correspondentes. Isso é realizado pelo parâmetro adicional **output** em itens `log` e `logrt`.

Usar o parâmetro 'output' permite indicar o "grupo de captura" da correspondência que pode nos interessar.

Assim, por exemplo

```
log[/path/to/the/file,"large result buffer allocation.*Entries: ([0-9]+)",,,,\1]
```

deve permitir retornar a contagem de entradas conforme encontrada no conteúdo de:

```
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
```

Apenas o número será retornado porque **\1** refere-se ao primeiro e único grupo de captura: **([0-9]+)**.

E, com a capacidade de extrair e retornar um número, o valor pode ser usado para definir triggers.

Usando o parâmetro maxdelay

O parâmetro `maxdelay` em items de log permite ignorar algumas linhas mais antigas dos arquivos de log para que as linhas mais recentes sejam analisadas dentro dos segundos definidos em `maxdelay`.

Warning:
Especificar 'maxdelay' > 0 pode levar a **ignorar registros importantes do arquivo de log e alertas perdidos**. Use com cuidado, por sua conta e risco, somente quando necessário.

Por padrão, os items para monitoramento de log acompanham todas as novas linhas que aparecem nos arquivos de log. No entanto, existem aplicações que, em algumas situações, começam a escrever um número enorme de mensagens em seus arquivos de log. Por exemplo, se um banco de dados ou servidor DNS estiver indisponível, essas aplicações inundam os arquivos de log com milhares de mensagens de erro quase idênticas até que a operação normal seja restaurada. Por padrão, todas essas mensagens serão devidamente analisadas e as linhas correspondentes enviadas para o server conforme configurado nos items `log` e `logrt`.

A proteção interna contra sobrecarga consiste em um parâmetro configurável `maxlines` (protege o server de muitas linhas de log correspondentes recebidas) e um limite de 10*'maxlines' (protege a CPU e o I/O do host de sobrecarga pelo agent em uma verificação). Ainda assim, existem 2 problemas com a proteção interna. Primeiro, um grande número de mensagens potencialmente pouco informativas são reportadas ao server e consomem espaço no banco de dados. Segundo, devido ao número limitado de linhas analisadas por segundo, o agent pode ficar atrasado em relação aos registros mais recentes do log por horas. Muito provavelmente, você pode preferir ser informado mais rapidamente sobre a situação atual nos arquivos de log em vez de analisar registros antigos por horas.

A solução para ambos os problemas é usar o parâmetro `maxdelay`. Se `maxdelay` > 0 for especificado, durante cada verificação o número de bytes processados, o número de bytes restantes e o tempo de processamento são medidos. A partir desses números, o agent calcula um atraso estimado - quantos segundos levaria para analisar todos os registros restantes em um arquivo de log.

Se o atraso não exceder `maxdelay`, o agent continua analisando o arquivo de log normalmente.

Se o atraso for maior que `maxdelay`, o agent **ignora um trecho do arquivo de log "pulando" sobre ele** para uma nova posição estimada, de modo que as linhas restantes possam ser analisadas dentro dos segundos definidos em `maxdelay`.

Observe que o agent nem mesmo lê as linhas ignoradas para o buffer, mas calcula uma posição aproximada para pular no arquivo.

O fato de pular linhas do arquivo de log é registrado no arquivo de log do agent assim:

```
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
```

O número "to byte" é aproximado porque, após o "pulo", o agent ajusta a posição no arquivo para o início de uma linha de log, que pode estar mais à frente ou mais atrás no arquivo.

Dependendo de como a velocidade de crescimento se compara com a velocidade de análise do arquivo de log, você pode não ver "pulos", ver "pulos" raros ou frequentes, grandes ou pequenos, ou até mesmo um pequeno "pulo" em cada verificação. Flutuações na carga do sistema e na latência da rede também afetam o cálculo do atraso e, portanto, o "pulo" à frente para acompanhar o parâmetro "maxdelay".

Definir `maxdelay` < `intervalo de atualização` não é recomendado (pode resultar em "pulos" pequenos e frequentes).

Notas sobre o tratamento da rotação de arquivos de log 'copytruncate'

`logrt` com a opção `copytruncate` assume que arquivos de log diferentes possuem registros diferentes (pelo menos seus timestamps são diferentes), portanto, os hashes MD5 dos blocos iniciais (até os primeiros 512 bytes) serão diferentes. Dois arquivos com os mesmos hashes MD5 dos blocos iniciais significam que um deles é o original e o outro é uma cópia.

`logrt` com a opção `copytruncate` faz um esforço para processar corretamente as cópias dos arquivos de log sem relatar duplicatas. No entanto, coisas como produzir várias cópias de arquivos de log com o mesmo timestamp, rotação de arquivos de log mais frequente do que o intervalo de atualização do item logrt[], reinicialização frequente do agent não são recomendadas. O agent tenta lidar com todas essas situações de forma razoável, mas bons resultados não podem ser garantidos em todas as circunstâncias.

Notas sobre arquivos persistentes para itens log*[]

Objetivo dos arquivos persistentes

Quando o agent Zabbix é iniciado, ele recebe uma lista de verificações ativas do server ou proxy Zabbix. Para as métricas log[], ele recebe o tamanho do log processado e o horário de modificação para determinar de onde começar o monitoramento do arquivo

de log. Dependendo do tamanho real do arquivo de log e do horário de modificação relatados pelo sistema de arquivos, o agent decide se continua o monitoramento do arquivo de log a partir do tamanho processado ou se reanalisa o arquivo de log desde o início.

Um agent em execução mantém um conjunto maior de atributos para rastrear todos os arquivos de log monitorados entre as verificações. Esse estado em memória é perdido quando o agent é interrompido.

O novo parâmetro opcional **persistent_dir** especifica um diretório para armazenar esse estado dos itens log[], log.count[], logrt[] ou logrt.count[] em um arquivo. O estado do item de log é restaurado a partir do arquivo persistente após o agent Zabbix ser reiniciado.

O principal caso de uso é o monitoramento de um arquivo de log localizado em um sistema de arquivos espelhado. Até certo momento, o arquivo de log é gravado em ambos os espelhos. Depois, os espelhos são separados. Na cópia ativa, o arquivo de log continua crescendo, recebendo novos registros. O agent Zabbix o analisa e envia o tamanho dos logs processados e o horário de modificação para o server. Na cópia passiva, o arquivo de log permanece o mesmo, bem atrás da cópia ativa. Posteriormente, o sistema operacional e o agent Zabbix são reiniciados a partir da cópia passiva. O tamanho do log processado e o horário de modificação que o agent Zabbix recebe do server podem não ser válidos para a situação na cópia passiva. Para continuar o monitoramento do arquivo de log a partir do ponto em que o agent parou no momento da divisão do espelho do sistema de arquivos, o agent restaura seu estado a partir do arquivo persistente.

Operação do agent com arquivo persistente

Na inicialização, o agent do Zabbix não sabe nada sobre arquivos persistentes. Somente após receber uma lista de verificações ativas do Zabbix server (proxy), o agent percebe que alguns itens de log devem ser respaldados por arquivos persistentes em diretórios especificados.

Durante a operação do agent, os arquivos persistentes são abertos para escrita (com fopen(filename, ”w”)) e sobrescritos com os dados mais recentes. A chance de perder dados do arquivo persistente se a sobrescrita e a divisão do espelhamento do sistema de arquivos acontecerem ao mesmo tempo é muito pequena, não há tratamento especial para isso. A escrita no arquivo persistente NÃO é seguida por sincronização forçada com a mídia de armazenamento (fsync() não é chamado).

A sobrescrita com os dados mais recentes é feita após o relatório bem-sucedido do registro correspondente do arquivo de log ou metadados (tamanho do log processado e hora da modificação) para o Zabbix server. Isso pode acontecer com a mesma frequência de cada verificação de item se o arquivo de log continuar mudando.

Nenhuma ação especial durante o desligamento do agent.

Após receber uma lista de verificações ativas, o agent marca arquivos persistentes obsoletos para remoção. Um arquivo persistente se torna obsoleto se: - O item de log correspondente não for mais monitorado; - Um item de log for reconfigurado com um local **persistent_dir** diferente do anterior.

A remoção é feita com um atraso de 24 horas porque arquivos de log em estado NOTSUPPORTED não são incluídos na lista de verificações ativas, mas podem se tornar SUPPORTED posteriormente e seus arquivos persistentes serão úteis.

Se o agent for parado antes de expirar as 24 horas, os arquivos obsoletos não serão excluídos, pois o agent Zabbix não está mais recebendo informações sobre sua localização do server Zabbix.

> **Warning:**
> Reconfigurar o **persistent_dir** de um item de log de volta para o local antigo enquanto o agent está parado, sem excluir o arquivo persistente antigo pelo usuário - fará com que o estado do agent seja restaurado a partir do arquivo persistente antigo, resultando em mensagens perdidas ou alertas falsos.

Nomeação e localização de arquivos persistentes

O agent do Zabbix diferencia as verificações ativas por suas chaves. Por exemplo, logrt[/home/zabbix/test.log] e logrt[/home/zabbix/test.log,] são itens diferentes. Modificar o item logrt[/home/zabbix/test.log„,10] no frontend para logrt[/home/zabbix/test.log„,20] resultará na exclusão do item logrt[/home/zabbix/test.log„,10] da lista de verificações ativas do agent e na criação do item logrt[/home/zabbix/test.log„,20] (alguns atributos são mantidos durante a modificação no frontend/server, mas não no agent).

O nome do arquivo é composto pela soma MD5 da chave do item com o comprimento da chave do item anexado para reduzir a possibilidade de colisões. Por exemplo, o estado do item logrt[/home/zabbix50/test.log„„„„/home/zabbix50/agent_private] será mantido no arquivo persistente c963ade4008054813bbc0a650bb8e09266.

Vários itens de log podem usar o mesmo valor de **persistent_dir**.

persistent_dir é especificado levando em consideração layouts específicos de sistema de arquivos, pontos de montagem e opções de montagem e configuração de espelhamento de armazenamento - o arquivo persistente deve estar no mesmo sistema de arquivos espelhado que o arquivo de log monitorado.

Se o diretório **persistent_dir** não puder ser criado ou não existir, ou se os direitos de acesso para o agent do Zabbix não permitirem criar/gravar/ler/excluir arquivos, o item de log se tornará NOTSUPPORTED.

Se os direitos de acesso aos arquivos de armazenamento persistente forem removidos durante a operação do agent ou outros erros ocorrerem (por exemplo, disco cheio), os erros serão registrados no arquivo de log do agent, mas o item de log não se tornará NÃO SUPORTADO.

Carga em I/O

O arquivo persistente do item é atualizado após o envio bem-sucedido de cada lote de dados (contendo os dados do item) para o server. Por exemplo, o `BufferSize` padrão é 100. Se um item de log encontrou 70 registros correspondentes, os primeiros 50 registros serão enviados em um lote, o arquivo persistente será atualizado, então os 20 registros restantes serão enviados (talvez com algum atraso quando mais dados forem acumulados) no segundo lote, e o arquivo persistente será atualizado novamente.

Ações se a comunicação falhar entre o agent e o server

Cada linha correspondente dos itens `log[]` e `logrt[]` e o resultado de cada verificação dos itens `log.count[]` e `logrt.count[]` requerem um slot livre na área designada de 50% no buffer de envio do agent. Os elementos do buffer são enviados regularmente para o server (ou proxy) e os slots do buffer ficam livres novamente.

Enquanto houver slots livres na área de log designada no buffer de envio do agent e a comunicação falhar entre o agent e o server (ou proxy), os resultados do monitoramento de log são acumulados no buffer de envio. Isso ajuda a mitigar falhas de comunicação de curta duração.

Durante falhas de comunicação mais longas, todos os slots de log são ocupados e as seguintes ações são tomadas:

- As verificações dos itens `log[]` e `logrt[]` são interrompidas. Quando a comunicação é restabelecida e há slots livres no buffer, as verificações são retomadas a partir da posição anterior. Nenhuma linha correspondente é perdida, elas apenas são relatadas posteriormente.
- As verificações dos itens `log.count[]` e `logrt.count[]` são interrompidas se `maxdelay = 0` (padrão). O comportamento é semelhante aos itens `log[]` e `logrt[]` conforme descrito acima. Observe que isso pode afetar os resultados de `log.count[]` e `logrt.count[]`: por exemplo, uma verificação conta 100 linhas correspondentes em um arquivo de log, mas como não há slots livres no buffer, a verificação é interrompida. Quando a comunicação é restabelecida, o agent conta as mesmas 100 linhas correspondentes e também 70 novas linhas correspondentes. O agent agora envia count = 170 como se tivessem sido encontradas em uma única verificação.
- As verificações dos itens `log.count[]` e `logrt.count[]` com `maxdelay > 0`: se não houve "salto" durante a verificação, o comportamento é semelhante ao descrito acima. Se ocorreu um "salto" sobre linhas do arquivo de log, então a posição após o "salto" é mantida e o resultado contado é descartado. Assim, o agent tenta acompanhar o crescimento do arquivo de log mesmo em caso de falha de comunicação.

Manipulação de erros de compilação e execução de expressões regulares

Se uma expressão regular usada em um item `log[]`, `logrt[]`, `log.count[]` ou `logrt.count[]` não puder ser compilada pela biblioteca PCRE ou PCRE2, o item entra no estado NÃO SUPORTADO com uma mensagem de erro. Para continuar monitorando o item de log, a expressão regular deve ser corrigida.

Se a expressão regular for compilada com sucesso, mas falhar em tempo de execução (em alguns ou em todos os registros de log), o item de log permanece suportado e o monitoramento continua. O erro de tempo de execução é registrado no arquivo de log do agent Zabbix (sem o registro do arquivo de log).

A taxa de registro é limitada a um erro de tempo de execução por verificação para permitir que o agent Zabbix monitore seu próprio arquivo de log. Por exemplo, se 10 registros forem analisados e 3 registros falharem com um erro de tempo de execução de regexp, um registro será produzido no log do agent.

Exceção: se `MaxLinesPerSecond=1` e o intervalo de atualização=1 (apenas 1 registro é permitido analisar por verificação), os erros de tempo de execução de regexp não são registrados.

O zabbix_agentd registra a chave do item em caso de erro de tempo de execução, o zabbix_agent2 registra o ID do item para ajudar a identificar qual item de log possui erros de tempo de execução. É recomendável redesenhar a expressão regular em caso de erros de tempo de execução.

7 Items calculados

Visão geral

Um item calculado permite criar um cálculo com base nos valores de alguns itens existentes. Por exemplo, você pode querer calcular a média horária de algum valor de item ou calcular o valor total para um grupo de itens. É para isso que serve um item calculado.

Os cálculos podem usar tanto:

- valores únicos de itens individuais
- um filtro complexo para selecionar vários itens para agregação (consulte cálculos agregados para obter detalhes)

Itens calculados são uma forma de criar fontes de dados virtuais. Todos os cálculos são feitos apenas pelo Zabbix server. Os valores são periodicamente calculados com base na expressão aritmética utilizada.

Os dados resultantes são armazenados no banco de dados do Zabbix como para qualquer outro item; tanto os valores de histórico quanto de tendência são armazenados e gráficos podem ser gerados.

Note:
Se o resultado do cálculo for um valor float, ele será convertido para inteiro se o tipo de informação do item calculado for *Numérico (sem sinal)*.
Além disso, se não houver dados recentes no cache e não houver período de consulta definido na função, o Zabbix, por padrão, irá buscar no passado até uma semana para consultar o banco de dados por valores históricos.

Itens calculados compartilham sua sintaxe com expressões de trigger. A comparação com strings é permitida em itens calculados. Itens calculados podem ser referenciados por macros ou outras entidades, assim como qualquer outro tipo de item.

Para usar itens calculados, escolha o tipo de item **Calculado**.

Campos configuráveis

A **key** é um identificador único do item (por host). Você pode criar qualquer nome de key usando os símbolos suportados.

A definição do cálculo deve ser inserida no campo **Fórmula**. Não há conexão entre a fórmula e a key. Os parâmetros da key não são usados de nenhuma forma na fórmula.

A sintaxe de uma fórmula simples é:

`function(/host/key,<parâmetro1>,<parâmetro2>,...)`

onde:

function	Uma das funções suportadas: last, min, max, avg, count, etc
host	Host do item que é usado para o cálculo. O host atual pode ser omitido (ou seja, como em `function(//key,parameter,...)`).
key	Key do item que é usado para o cálculo. Itens que retornam valores com o tipo de dado binário não são suportados.
parameter(s)	Parâmetros da função, se necessário. Sufixos de tempo e sufixos de tamanho de memória são suportados.

Attention:
Macros de usuário na fórmula serão expandidas se usadas para referenciar um parâmetro de função, parâmetro de filtro de item ou uma constante. Macros de usuário NÃO serão expandidas se referenciando uma função, nome de host, key de item, parâmetro de key de item ou operador.

Uma fórmula mais complexa pode usar uma combinação de funções, operadores e parênteses. Você pode usar todas as funções e operadores suportados em expressões de trigger. A lógica e a precedência dos operadores são exatamente as mesmas.

Ao contrário das expressões de trigger, o Zabbix processa itens calculados de acordo com o intervalo de atualização do item, e não ao receber um novo valor.

Todos os itens que são referenciados por funções de histórico na fórmula do item calculado devem existir e estar coletando dados. Além disso, se você alterar a key do item de um item referenciado, deverá atualizar manualmente todas as fórmulas que usam essa key.

Um item calculado pode se tornar não suportado em vários casos:

- item(s) referenciado(s)
  - não é encontrado
  - está desabilitado
  - pertence a um host desabilitado
  - não é suportado (exceto com a função nodata() e operadores com valores desconhecidos)
- sem dados para calcular uma função
- divisão por zero
- sintaxe incorreta usada

Exemplos de uso

Exemplo 1

Calculando a porcentagem de espaço livre em disco em '/'.

Uso da função **last**:

```
100*last(//vfs.fs.size[/,free])/last(//vfs.fs.size[/,total])
```

O Zabbix pegará os valores mais recentes para os espaços livres e totais em disco e calculará a porcentagem de acordo com a fórmula fornecida.

Exemplo 2

Calculando uma média de 10 minutos do número de valores processados pelo Zabbix.

Uso da função **avg**:

```
avg(/Zabbix Server/zabbix[wcache,values],10m)
```

Observe que o uso extensivo de items calculados com longos períodos de tempo pode afetar o desempenho do Zabbix server.

Exemplo 3

Calculando a largura de banda total em eth0.

Soma de duas funções:

```
last(//net.if.in[eth0,bytes])+last(//net.if.out[eth0,bytes])
```

Exemplo 4

Calculando a porcentagem do tráfego de entrada.

Expressão mais complexa:

```
100*last(//net.if.in[eth0,bytes])/(last(//net.if.in[eth0,bytes])+last(//net.if.out[eth0,bytes]))
```

Veja também: Exemplos de cálculos agregados

1 Cálculos agregados

Visão geral

Os cálculos agregados são um tipo de item calculado que permite coletar informações de vários items pelo Zabbix server e, em seguida, calcular um agregado, dependendo da função de agregação utilizada.

Os cálculos agregados não exigem nenhum agent em execução no host monitorado.

Sintaxe

Para recuperar agregados, use uma das funções de agregação suportadas: `avg`, `max`, `min`, `sum`, etc. Em seguida, adicione a função **foreach** como o único parâmetro e seu filtro de item para selecionar os itens necessários:

```
aggregate_function(function_foreach(/host/key?[group="host group"],timeperiod))
```

Uma função **foreach** (por exemplo, *avg_foreach*, *count_foreach*, etc.) retorna um valor agregado para cada item selecionado. Os itens são selecionados usando o filtro de item (`/host/key?[group="host group"]`), a partir do histórico do item. Para mais detalhes, consulte funções foreach.

Se alguns dos itens não tiverem dados para o período solicitado, eles serão ignorados no cálculo. Se nenhum item tiver dados, a função retornará um erro.

Alternativamente, você pode listar vários itens como parâmetros para agregação:

```
aggregate_function(function(/host/key,parameter),function(/host2/key2,parameter),...)
```

Observe que `function` aqui deve ser uma função de histórico/tendência.

> **Note:**
> Se o agregado resultar em um valor float, ele será convertido para inteiro se o tipo de informação do item agregado for *Numérico (sem sinal)*.

Macros de usuário e macros de descoberta de baixo nível são suportadas em:

- parâmetros da chave do item
- parâmetros da função
- condições de filtro (nome do grupo de hosts e nome da tag)
- constantes de expressão

Um cálculo agregado pode se tornar não suportado se:

- nenhum dos itens referenciados for encontrado (o que pode acontecer se a chave do item estiver incorreta, nenhum dos itens existir ou todos os grupos incluídos estiverem incorretos)
- não houver dados para calcular uma função

Exemplos de uso

Exemplos de chaves para cálculos agregados.

Exemplo 1

Espaço total em disco do grupo de hosts 'MySQL Servers'.

```
sum(last_foreach(/*/vfs.fs.size[/,total]?[group="MySQL Servers"]))
```

Exemplo 2

Soma dos valores mais recentes de todos os items que correspondem a net.if.in[*] no host.

```
sum(last_foreach(/host/net.if.in[*]))
```

Exemplo 3

Carga média do processador do grupo de hosts 'MySQL Servers'.

```
avg(last_foreach(/*/system.cpu.load[,avg1]?[group="MySQL Servers"]))
```

Exemplo 4

Média de 5 minutos do número de consultas por segundo para o grupo de hosts 'MySQL Servers'.

```
avg(avg_foreach(/*/mysql.qps?[group="MySQL Servers"],5m))
```

Exemplo 5

Carga média da CPU em todos os hosts em vários grupos de hosts que possuem tags específicas.

```
avg(last_foreach(/*/system.cpu.load?[(group="Servers A" or group="Servers B" or group="Servers C") and (ta
```

Exemplo 6

Cálculo usado nas somas dos valores mais recentes dos items de todo um grupo de hosts.

```
sum(last_foreach(/*/net.if.out[eth0,bytes]?[group="video"])) / sum(last_foreach(/*/nginx_stat.sh[active]?[
```

Exemplo 7

O número total de items não suportados no grupo de hosts 'Zabbix servers'.

```
sum(last_foreach(/*/zabbix[host,,items_unsupported]?[group="Zabbix servers"]))
```

Exemplo 8

Soma dos últimos resultados numéricos das verificações de DNS em todos os hosts. A forma do item mostrada é `net.dns[192.0.2.0,examp` como exemplo de uma possível chave.

```
sum(last_foreach(/*/net.dns[*,*,*]))
```

Observe que os curingas devem corresponder ao número de parâmetros na chave (aqui `net.dns` possui três parâmetros: ip, nome, tipo).

Exemplos de sintaxe correta/incorreta

Expressões (incluindo chamadas de funções) não podem ser usadas como parâmetros de histórico, tendência ou foreach das funções. No entanto, essas funções podem ser usadas em outros parâmetros de função (não históricos).

Expressão	Exemplo
Válido	`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)))`

Inválido	`sum(/host/key,10+2)`
	`sum(/host/key, avg(10,2))`
	`sum(/host/key,last(/host/key2))`

Observe que, em uma expressão como:

```
sum(sum_foreach(//resptime[*],5m))/sum(count_foreach(//resptime[*],5m))
```

não é possível garantir que ambas as partes da equação sempre terão o mesmo conjunto de valores. Enquanto uma parte da expressão está sendo avaliada, um novo valor para o período solicitado pode chegar e, então, a outra parte da expressão terá um conjunto diferente de valores.

8 Verificações internas

Visão geral

As verificações internas permitem monitorar os processos internos do Zabbix. Em outras palavras, você pode monitorar o que acontece com o Zabbix server ou o Zabbix proxy.

As verificações internas são calculadas:

- no Zabbix server - se o host for monitorado pelo server
- no Zabbix proxy - se o host for monitorado pelo proxy

As verificações internas são processadas pelo server ou proxy independentemente do status de manutenção do host.

Para usar este item, escolha o tipo de item **Zabbix internal**.

Note:
As verificações internas são processadas pelos pollers do Zabbix.

Desempenho

O uso de alguns itens internos pode afetar negativamente o desempenho. Esses itens são:

- `zabbix[host,,items]`
- `zabbix[host,,items_unsupported]`
- `zabbix[hosts]`
- `zabbix[items]`
- `zabbix[items_unsupported]`
- `zabbix[queue,,]`
- `zabbix[requiredperformance]`
- `zabbix[stats,,,queue,,]`
- `zabbix[triggers]`

As seções do frontend Informações do sistema e Fila também são afetadas.

Checagens suportadas

As chaves de item são listadas sem parâmetros personalizáveis e informações adicionais. Clique na chave do item para ver os detalhes completos.

Chave do item	Descrição
zabbix[boottime]	O tempo de inicialização do processo do Zabbix server ou Zabbix proxy em segundos.
zabbix[cluster,discovery,nodes]	Descobre os nós do cluster de alta disponibilidade.
zabbix[connector_queue]	A contagem de valores enfileirados na fila do conector.
zabbix[discovery_queue]	A contagem de checagens de rede enfileiradas na fila de descoberta.
zabbix[host,,items]	O número de items habilitados (suportados e não suportados) no host.
zabbix[host,,items_unsupported]	O número de items habilitados e não suportados no host.
zabbix[host,,maintenance]	O status de manutenção atual do host.
zabbix[host,active_agent,available]	A disponibilidade das checagens de agent ativo no host.

Chave do item	Descrição
zabbix[host,discovery,interfaces]	Os detalhes de todas as interfaces configuradas do host no frontend do Zabbix.
zabbix[host„available]	A disponibilidade da interface principal de um tipo específico de checagem no host.
zabbix[hosts]	O número de hosts monitorados.
zabbix[items]	O número de items habilitados (suportados e não suportados).
zabbix[items_unsupported]	O número de items não suportados.
zabbix[java„]	As informações sobre o Zabbix Java gateway.
zabbix[lld_queue]	A contagem de valores enfileirados na fila de processamento de descoberta de baixo nível.
zabbix[preprocessing]	Estatísticas dos valores recebidos pelo gerenciador de pré-processamento.
zabbix[preprocessing_queue]	A contagem de valores enfileirados na fila de pré-processamento.
zabbix[process„,]	A porcentagem de tempo que um processo específico do Zabbix ou um grupo de processos (identificados por <type> e <mode>) passou em <state>.
zabbix[proxy„]	As informações sobre o Zabbix proxy.
zabbix[proxy,discovery]	A lista de proxies do Zabbix.
zabbix[proxy group„available]	O número de proxies online em um grupo de proxy.
zabbix[proxy group„pavailable]	A porcentagem de proxies online em um grupo de proxy.
zabbix[proxy group„proxies]	A lista de proxies do Zabbix em um grupo de proxy.
zabbix[proxy group„state]	O estado de um grupo de proxy.
zabbix[proxy group,discovery]	Retorna uma lista de grupos de proxy com dados de configuração e dados em tempo real.
zabbix[proxy_buffer,buffer,]	Retorna as estatísticas de uso do buffer de memória do proxy.
zabbix[proxy_buffer,state,changes]	Retorna o número de mudanças de estado entre os modos de buffer disco/memória desde o início.
zabbix[proxy_buffer,state,current]	Retorna o estado de trabalho atual onde os novos dados estão sendo armazenados.
zabbix[proxy_history]	O número de valores na tabela de histórico do proxy aguardando para serem enviados ao server.
zabbix[queue„]	O número de items monitorados na fila que estão atrasados pelo menos <from> segundos, mas menos de <to> segundos.
zabbix[rcache„]	As estatísticas de disponibilidade do cache de configuração do Zabbix.
zabbix[requiredperformance]	O desempenho necessário do Zabbix server ou Zabbix proxy, em novos valores por segundo esperados.
zabbix[stats„]	Retorna as métricas internas de um Zabbix server ou proxy. Se <ip> e <port> forem especificados, as métricas são recuperadas da instância remota; caso contrário, da instância local.
zabbix[stats„,queue„]	Retorna as métricas internas da fila de um Zabbix server ou proxy. Se <ip> e <port> forem especificados, as métricas são recuperadas da instância remota; caso contrário, da instância local.
zabbix[tcache„]	As estatísticas de efetividade do cache de função de tendência do Zabbix.
zabbix[triggers]	O número de triggers habilitados no banco de dados do Zabbix, com todos os items habilitados em hosts habilitados.
zabbix[uptime]	O tempo de atividade do processo do Zabbix server ou proxy em segundos.
zabbix[vcache,buffer,]	As estatísticas de disponibilidade do cache de valores do Zabbix.
zabbix[vcache,cache,]	As estatísticas de efetividade do cache de valores do Zabbix.
zabbix[version]	A versão do Zabbix server ou proxy.
zabbix[vmware,buffer,]	As estatísticas de disponibilidade do cache vmware do Zabbix.
zabbix[vps,written]	O número total de valores de histórico gravados no banco de dados.
zabbix[wcache„]	As estatísticas e disponibilidade do cache de escrita do Zabbix.

Detalhes da chave do item

- Parâmetros sem colchetes angulares são obrigatórios e devem ser usados *como estão* (por exemplo, ”host” e ”available” em `zabbix[host,<type>,available]`).
- Parâmetros com colchetes angulares **< >** devem ser substituídos por um valor válido. Se um parâmetro tiver um valor padrão, ele pode ser omitido.
- Valores para itens e parâmetros de item rotulados como ”não suportado em proxy” só podem ser recuperados se o host for monitorado pelo server. Da mesma forma, valores ”não suportados em server” só podem ser recuperados se o host for monitorado por proxy.

zabbix[boottime]

 O tempo de inicialização do processo do Zabbix server ou Zabbix proxy em segundos.
 Valor de retorno: *Inteiro*.

zabbix[cluster,discovery,nodes]

 Descobre os nós do cluster de alta disponibilidade.
 Valor de retorno: *objeto JSON*.

Comentários:

- Este item pode ser usado em descoberta de baixo nível.
- Os campos retornados correspondem principalmente às propriedades do objeto de nó de alta disponibilidade.
- Campos adicionais:
  - `db_timestamp` — hora atual do server (timestamp Unix) incluída na resposta
  - `lastaccess_age` — segundos desde o último heartbeat do nó (db_timestamp - lastaccess)
  - `status` — estado do nó:
    - 0 — standby
    - 1 — parado manualmente
    - 2 — indisponível
    - 3 — ativo

Exemplo de valores de retorno:

```
[
  {
    "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]

 A contagem de valores enfileirados na fila do conector.
 Valor de retorno: *Inteiro*.

zabbix[discovery_queue]

 A contagem de verificações de rede enfileiradas na fila de descoberta.
 Valor de retorno: *Inteiro*.

zabbix[host,,items]

 O número de items habilitados (suportados e não suportados) no host.
 Valor de retorno: *Inteiro*.

zabbix[host,,items_unsupported]

 O número de items habilitados e não suportados no host.
 Valor de retorno: *Integer*.

zabbix[host,„maintenance]

 O status de manutenção atual do host.
 Valores de retorno: *0* - estado normal; *1* - manutenção com coleta de dados; *2* - manutenção sem coleta de dados.

Comentários:

- Este item é sempre processado pelo Zabbix server, independentemente da localização do host (no server ou proxy). O proxy não receberá este item com os dados de configuração.
- O segundo parâmetro deve estar vazio e é reservado para uso futuro.

zabbix[host,active_agent,available]

 A disponibilidade de verificações de agent ativo no host.
 Valores de retorno: *0* - desconhecido; *1* - disponível; *2* - não disponível.

zabbix[host,discovery,interfaces]

 Os detalhes de todas as interfaces configuradas do host no frontend do Zabbix.
 Valor de retorno: *objeto JSON*.

Comentários:

- Este item pode ser usado na descoberta de baixo nível.
- Este item não é suportado no proxy Zabbix.

zabbix[host,<type>,available]

 A disponibilidade da interface principal de um determinado tipo de checagem no host.
 Valores de retorno: *0* - desconhecido; *1* - disponível; *2* - não disponível.

Parâmetros:

- **type** - *agent*, *snmp*, *ipmi* ou *jmx*.

Comentários:

- O valor do item é calculado de acordo com os parâmetros de configuração relacionados à inacessibilidade/indisponibilidade do host.

zabbixhosts

 O número de hosts monitorados.
 Valor de retorno: *Inteiro*.

zabbixitems

 O número de items habilitados (suportados e não suportados).
 Valor de retorno: *Inteiro*.

zabbix[items_unsupported]

 O número de items não suportados.
 Valor de retorno: *Integer*.

zabbix[java,„<param>]

 A informação sobre o Zabbix Java gateway.
 Valores de retorno: *1* - se <param> for *ping*; *versão do Java gateway* - se <param> for *version* (por exemplo: ”7.0.0”).

Parâmetros:

- **param** - *ping* ou *version*.

Comentários:

- Este item pode ser usado para verificar a disponibilidade do Java gateway usando a função de trigger `nodata()`.
- O segundo parâmetro deve estar vazio e é reservado para uso futuro.

zabbix[lld_queue]

 A contagem de valores enfileirados na fila de processamento de descoberta de baixo nível.
 Valor de retorno: *Integer*.

Comentários:

- Este item pode ser usado para monitorar o comprimento da fila de processamento de descoberta de baixo nível.

zabbix[preprocessing]

 Estatísticas dos valores recebidos pelo gerenciador de pré-processamento:

- *queued* - o número e o tamanho dos valores em fila que requerem pré-processamento (contador)
- *direct* - o número e o tamanho dos valores em fila que não requerem pré-processamento (contador)
- *queue* - a contagem de valores enfileirados na fila de pré-processamento (igual a `zabbix[preprocessing_queue]`)

Valor de retorno: *JSON*.

Este item é suportado desde o Zabbix 7.0.12.

Exemplo de valores de retorno:

```
{"data":
    {
    "queued": {
        "count": 106,
        "size": 58620
    },
    "direct": {
        "count": 395,
        "size": 33843
    },
    "queue": 0
    }
}
```

zabbix[preprocessing_queue]

 A contagem de valores enfileirados na fila de pré-processamento.
 Valor de retorno: *Integer*.

Comentários:

- Este item pode ser usado para monitorar o comprimento da fila de pré-processamento.

zabbix[process,<type>,<mode>,<state>]

 A porcentagem de tempo que um processo Zabbix específico ou um grupo de processos (identificados por <type> e <mode>) passou em <state>. É calculado apenas para o último minuto.
 Valor de retorno: *Float*.

Parâmetros:

- **type** - para processos do server: *agent poller*, *alert manager*, *alert syncer*, *alerter*, *availability manager*, *browser poller*, *configuration syncer*, *configuration syncer worker*, *connector manager*, *connector worker*, *discovery manager*, *discovery worker*, *escalator*, *ha manager* (*desde o Zabbix 7.0.5*), *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*;
para processos do proxy: *agent poller*, *availability manager*, *browser poller*, *configuration syncer*, *data sender*, *discovery manager*, *discovery worker*, *history syncer*, *housekeeper*, *http agent poller*, *http poller*, *icmp pinger*, *internal poller*, *ipmi manager*, *ipmi poller*, *java poller*, *odbc poller*, *poller*, *preprocessing manager*, *preprocessing worker*, *self-monitoring*, *snmp poller*, *snmp trapper*, *task manager*, *trapper*, *unreachable poller*, *vmware collector*;
- **mode** - *avg* - valor médio para todos os processos de um determinado tipo (padrão);
count - retorna o número de forks para um determinado tipo de processo, <state> não deve ser especificado;
max - valor máximo;
min - valor mínimo;
<*process number*> - número do processo (entre 1 e o número de instâncias pré-forked; por exemplo, se 4 trappers estiverem em execução, o valor estará entre 1 e 4);
- **state** - *busy* - o processo está em estado ocupado, por exemplo, processando uma solicitação (padrão);
idle - o processo está em estado ocioso, sem fazer nada.

Comentários:

- Se <mode> for um número de processo Zabbix que não está em execução (por exemplo, com 5 pollers em execução, <mode> é especificado como 6), tal item se tornará não suportado.
- Mínimo e máximo referem-se à porcentagem de uso para um único processo. Portanto, se em um grupo de 3 pollers as porcentagens de uso por processo forem 2, 18 e 66, min retornará 2 e max retornará 66.
- Os processos relatam o que estão fazendo na memória compartilhada e o processo de auto-monitoramento resume esses dados a cada segundo. Mudanças de estado (busy/idle) são registradas na mudança - assim, um processo que se torna ocupado é registrado como tal e não muda ou atualiza o estado até que se torne ocioso. Isso garante que mesmo processos totalmente travados sejam corretamente registrados como 100% ocupados.
- Atualmente, ”busy” significa ”não dormindo”, mas no futuro estados adicionais podem ser introduzidos - aguardando locks, realizando consultas ao banco de dados, etc. Observe que, desde o Zabbix 7.0.6, os pollers assíncronos são considerados ocupados se atingirem o limite definido pelo parâmetro de configuração MaxConcurrentChecksPerPoller do server/proxy.
- No Linux e na maioria dos outros sistemas, a resolução é de 1/100 de segundo.

Exemplos:

```
zabbix[process,poller,avg,busy] #tempo médio dos processos poller fazendo algo durante o último minuto
zabbix[process,"icmp pinger",max,busy] #tempo máximo gasto fazendo algo por qualquer processo ICMP pinger
zabbix[process,"history syncer",2,busy] #tempo gasto fazendo algo pelo history syncer número 2 durante o ú
zabbix[process,trapper,count] #quantidade de processos trapper atualmente em execução
```

zabbix[proxy,<name>,<param>]

 Informação sobre o proxy Zabbix.
 Valor de retorno: *Integer*.

Parâmetros:

- **name** - o nome do proxy;
- **param** - *lastaccess* - o timestamp da última mensagem de heartbeat recebida do proxy;
delay - quanto tempo os valores coletados não foram enviados; calculado como ”proxy delay” + (”current server time” - ”proxy lastaccess”), onde ”proxy delay” é a diferença entre o tempo atual do proxy e o timestamp do valor não enviado mais antigo no proxy.

Comentários:

- Este item é sempre processado pelo Zabbix server, independentemente da localização do host (no server ou proxy).
- A função `fuzzytime()` pode ser usada para verificar a disponibilidade do proxy.

Exemplo:

```
zabbix[proxy,"Germany",lastaccess] #o timestamp da última mensagem de heartbeat recebida do proxy "Germany
```

zabbix[proxy,discovery]

 A lista de proxies Zabbix com nome, modo, criptografia, compressão, versão, última vez visto, contagem de hosts, contagem de items, valores necessários por segundo (vps), status da versão (atual/desatualizado/não suportado), timeouts por tipo de item, nome do grupo do proxy (se o proxy pertence a um grupo), estado (desconhecido/offline/online).
 Valor de retorno: *objeto JSON*.

zabbix[proxy group,<name>,available]

 O número de proxies online em um grupo de proxies.
 Valor de retorno: *Integer*.

Parâmetros:

- **name** - o nome do grupo de proxies.

zabbix[proxy group,<name>,pavailable]

 A porcentagem de proxies online em um grupo de proxies.
 Valor de retorno: *Float*.

Parâmetros:

- **name** - o nome do grupo de proxies.

zabbix[proxy group,<name>,proxies]

 A lista de proxies Zabbix em um grupo de proxies com nome, modo, criptografia, compactação, versão, última visualização, contagem de hosts, contagem de items, valores necessários por segundo (vps), status da versão (atual/desatualizada/não suportada), timeouts, nome do grupo de proxies, estado (desconhecido/offline/online).
 Valor de retorno: *JSON*.

Parâmetros:

- **name** - o nome do grupo de proxies.

zabbix[proxy group,<name>,state]

 O estado de um grupo de proxy.
 Valor de retorno: *0* - desconhecido; *1* - offline; *2* - recuperando; *3* - online; *4* - degradando.

Parâmetros:

- **name** - o nome do grupo de proxy.

zabbix[proxy group,discovery]

 Retorna uma lista de grupos de proxy com dados de configuração e dados em tempo real. Os dados de configuração incluem o nome do grupo de proxy, o atraso de failover e o número mínimo de proxies online necessários. Os dados em tempo real incluem o estado do grupo de proxy (veja os comentários para detalhes), o número de proxies online e a porcentagem de proxies online.
 Valor de retorno: *JSON*.

Comentários:

- Este item não retorna proxies sem grupo.

- Se houver um valor inválido para ”failover_delay” ou ”min_online”, então um valor especial *-1* é relatado para indicar isso. Valores inválidos podem ocorrer se macros forem usadas para configuração e as macros não puderem ser expandidas para um valor válido.
- O estado do grupo de proxy é relatado como um inteiro: *0* - desconhecido; *1* - offline; *2* - recuperando; *3* - online; *4* - degradando.

Exemplo de valores de retorno:

```
{
    "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>]

 As estatísticas de uso do buffer de memória do proxy.
 Valores de retorno: *Integer* (para tamanho); *Float* (para porcentagem).

Parâmetros:

- **mode**: *total* - o tamanho total do buffer (pode ser usado para verificar se o buffer de memória está habilitado);
free - o tamanho do buffer livre;
pfree - a porcentagem do buffer livre;
used - o tamanho do buffer usado;
pused - a porcentagem do buffer usado.

Comentários:

- Retorna um erro ’Proxy memory buffer is disabled’ quando o buffer de memória está desabilitado;
- Este item não é suportado no Zabbix server.

zabbix[proxy_buffer,state,changes]

 Retorna o número de mudanças de estado entre os modos de buffer de disco/memória desde o início.
 Valores de retorno: *Inteiro*; *0* - o buffer de memória está desabilitado.

Comentários:

- Mudanças frequentes de estado indicam que o tamanho ou a idade do buffer de memória devem ser aumentados.
- Se o estado do buffer de memória for monitorado com pouca frequência (por exemplo, uma vez por minuto), o buffer pode alternar seu estado sem que isso seja registrado.

zabbix[proxy_buffer,state,current]

 Retorna o estado de trabalho atual onde os novos dados estão sendo armazenados.
 Valores de retorno: *0* - disco; *1* - memória.

Comentários:

- ”0” também é retornado quando o buffer de memória está desabilitado.

zabbix[proxy_history]

 O número de valores na tabela de histórico do proxy aguardando para serem enviados ao server.
 Valores de retorno: *Inteiro*.

Comentários:

- Este item não é suportado no Zabbix server.

zabbix[queue,<from>,<to>]

 O número de items monitorados na fila que estão atrasados pelo menos <from> segundos, mas menos de <to> segundos.
 Valor de retorno: *Integer*.

Parâmetros:

- **from** - atrasado por pelo menos (o padrão é 6 segundos);
- **to** - atrasado por no máximo (o padrão é infinito).

Comentários:

- Sufixos de tempo (s,m,h,d,w) são suportados nos parâmetros.

Attention:
O Zabbix proxy não tem conhecimento dos períodos de manutenção; veja Cálculo de filas durante a manutenção para detalhes.

zabbix[rcache,<cache>,<mode>]

 As estatísticas de disponibilidade do cache de configuração do Zabbix.
 Valores de retorno: *Integer* (para tamanho); *Float* (para porcentagem).

Parâmetros:

- **cache** - *buffer*;
- **mode** - *total* - o tamanho total do buffer;
free - o tamanho do buffer livre;
pfree - a porcentagem do buffer livre;
used - o tamanho do buffer usado;
pused - a porcentagem do buffer usado.

zabbix[requiredperformance]

 O desempenho necessário do Zabbix server ou Zabbix proxy, em novos valores por segundo esperados.
 Valor de retorno: *Float*.

Comentários:

- Correlaciona-se aproximadamente com "Desempenho necessário do server, novos valores por segundo" em *Relatórios* > *Informações do sistema*.

zabbix[stats,<ip>,<port>]

 Retorna as métricas internas de um Zabbix server ou proxy. Se <ip> e <port> forem especificados, as métricas são recuperadas da instância remota; caso contrário, da instância local.
 Valores de retorno: *objeto JSON*.

Parâmetros:

- **ip** - a lista de IP/DNS/máscara de rede dos servers/proxies a serem consultados remotamente (o padrão é 127.0.0.1);
- **port** - a porta do server/proxy a ser consultado remotamente (o padrão é 10051).

Comentários:

- A solicitação de stats só será aceita dos endereços listados no parâmetro 'StatsAllowedIP' do server/proxy na instância de destino.
- Um conjunto selecionado de métricas internas é retornado por este item. Para detalhes, consulte Monitoramento remoto de estatísticas do Zabbix.

zabbix[stats,<ip>,<port>,queue,<from>,<to>]

 Retorna as métricas internas da fila (veja `zabbix[queue,<from>,<to>]`) de um Zabbix server ou proxy. Se <ip> e <port> forem especificados, as métricas são obtidas da instância remota; caso contrário, da instância local.
 Valores de retorno: *objeto JSON*.

Parâmetros:

- **ip** - a lista de IP/DNS/máscara de rede dos servers/proxies a serem consultados remotamente (o padrão é 127.0.0.1);
- **port** - a porta do server/proxy a ser consultado remotamente (o padrão é 10051);
- **from** - atrasado por pelo menos (o padrão é 6 segundos);
- **to** - atrasado por no máximo (o padrão é infinito).

Comentários:

- A solicitação de estatísticas só será aceita dos endereços listados no parâmetro 'StatsAllowedIP' do server/proxy na instância de destino.
- Um conjunto selecionado de métricas internas é retornado por este item. Para detalhes, veja Monitoramento remoto das estatísticas do Zabbix.

Attention:
O Zabbix proxy não tem conhecimento dos períodos de manutenção; veja Cálculo das filas durante a manutenção para detalhes.

zabbix[tcache,cache,<parameter>]

 As estatísticas de eficácia do cache de função de tendência do Zabbix.
 Valores de retorno: *Integer* (para tamanho); *Float* (para porcentagem).

Parâmetros:

- **parameter** - *all* - solicitações totais de cache (padrão);
hits - acertos de cache;
phits - porcentagem de acertos de cache;
misses - falhas de cache;
pmisses - porcentagem de falhas de cache;
items - o número de items em cache;
requests - o número de solicitações em cache;
pitems - porcentagem de items em cache de items em cache + solicitações. Uma porcentagem baixa provavelmente significa que o tamanho do cache pode ser reduzido.

Comentários:

- Este item não é suportado no proxy Zabbix.

zabbixtriggers

 O número de triggers habilitados no banco de dados do Zabbix, com todos os items habilitados em hosts habilitados.
 Valor de retorno: *Inteiro*.

Comentários:

- Este item não é suportado no proxy Zabbix.

zabbix[uptime]

 O tempo de atividade do processo do Zabbix server ou proxy em segundos.
 Valor de retorno: *Inteiro*.

zabbix[vcache,buffer,<mode>]

 As estatísticas de disponibilidade do cache de valores do Zabbix.
 Valores de retorno: *Integer* (para tamanho); *Float* (para porcentagem).

Parâmetros:

- **mode** - *total* - o tamanho total do buffer;
free - o tamanho do buffer livre;
pfree - a porcentagem do buffer livre;
used - o tamanho do buffer usado;
pused - a porcentagem do buffer usado.

Comentários:

- Este item não é suportado no proxy Zabbix.

zabbix[vcache,cache,<parameter>]

 As estatísticas de eficácia do cache de valores do Zabbix.
 Valor de retorno: *Integer*. Se <parameter> for definido como *mode*, o valor de retorno será: *0* - modo normal; *1* - modo de pouca memória.

Parâmetros:

- **parameter** - *requests* - o número total de solicitações;
hits - o número de acertos no cache (valores históricos obtidos do cache);
misses - o número de falhas no cache (valores históricos obtidos do banco de dados);
mode - o modo de operação do cache de valores.

Comentários:

- Uma vez que o modo de pouca memória tenha sido ativado, o cache de valores permanecerá nesse estado por 24 horas, mesmo que o problema que acionou esse modo seja resolvido antes.
- Você pode usar essa chave com a etapa de pré-processamento *Change per second* para obter estatísticas de valores por segundo.
- Este item não é suportado no Zabbix proxy.

zabbix[version]

 A versão do Zabbix server ou proxy.
 Valor de retorno: *String*. Por exemplo: `7.0.0`.

zabbix[vmware,buffer,<mode>]

 As estatísticas de disponibilidade do cache vmware do Zabbix.
 Valores de retorno: *Integer* (para tamanho); *Float* (para porcentagem).

Parâmetros:

- **mode** - *total* - o tamanho total do buffer;
free - o tamanho do buffer livre;
pfree - a porcentagem do buffer livre;
used - o tamanho do buffer usado;
pused - a porcentagem do buffer usado.

zabbix[vps,written]

 O número total de valores históricos gravados no banco de dados.
 Valor de retorno: *Inteiro*.

zabbix[wcache,<cache>,<mode>]

 As estatísticas e disponibilidade do cache de gravação do Zabbix.
 Valores de retorno: *Integer* (para número/tamanho); *Float* (para porcentagem).

Parâmetros:

- **cache** - *values*, *history*, *index* ou *trend*;
- **mode** - (com *values*) *all* (padrão) - o número total de valores processados pelo Zabbix server/proxy, exceto itens não suportados (contador);
float - o número de valores float processados (contador);
uint - o número de valores inteiros sem sinal processados (contador);
str - o número de valores de caracteres/string processados (contador);
log - o número de valores de log processados (contador);
text - o número de valores de texto processados (contador);
bin - o número de valores binários processados (contador);
not supported - o número de vezes que o processamento do item resultou em item tornando-se não suportado ou permanecendo nesse estado (contador);
(com *history*, *index*, *trend* cache) *pfree* (padrão) - a porcentagem de buffer livre;
total - o tamanho total do buffer;
free - o tamanho do buffer livre;
used - o tamanho do buffer usado;
pused - a porcentagem do buffer usado.

Comentários:

- O parâmetro de cache `trend` não é suportado com o Zabbix proxy.
- O cache de histórico é usado para armazenar valores de item. Um número baixo indica problemas de desempenho no lado do banco de dados.
- O cache de índice de histórico é usado para indexar os valores armazenados no cache de histórico.
- Após o cache de histórico ser preenchido e então limpo, o cache de índice de histórico ainda manterá alguns dados. Esse comportamento é esperado e ajuda o sistema a funcionar de forma mais eficiente, evitando o processamento extra necessário para redimensionar constantemente a memória.
- O cache de tendência armazena o agregado para a hora atual para todos os itens que recebem dados.
- Você pode usar a chave zabbix[wcache,values] com a etapa de pré-processamento *Change per second* para obter estatísticas de valores por segundo.

9 Verificações SSH

Visão geral

As verificações SSH são realizadas como monitoramento sem agent. O agent Zabbix não é necessário para verificações SSH.

Para realizar verificações SSH, o server Zabbix deve ser inicialmente configurado com suporte a SSH2 (libssh ou libssh2). Veja também: Requisitos.

> **Attention:**
> A partir do RHEL 8, apenas libssh é suportado. Para outras distribuições, libssh é sugerido em vez de libssh2.

Configuração

Autenticação por senha

As verificações SSH fornecem dois métodos de autenticação - um par usuário/senha e baseado em arquivo de chave.

Se você não pretende usar chaves, nenhuma configuração adicional é necessária, além de vincular libssh ou libssh2 ao Zabbix, se você estiver compilando a partir do código-fonte.

Autenticação por arquivo de chave

Para usar a autenticação baseada em chave para items SSH, certas alterações na configuração do server são necessárias.

Abra o arquivo de configuração do Zabbix server (*zabbix_server.conf*) como `root` e procure pela seguinte linha:

```
##### SSHKeyLocation=
```

Descomente-a e defina o caminho completo para a pasta onde as chaves públicas e privadas serão localizadas:

```
SSHKeyLocation=/home/zabbix/.ssh
```

Salve o arquivo e reinicie o Zabbix server em seguida.

O caminho */home/zabbix* aqui é o diretório home para a conta de usuário *zabbix*, e *.ssh* é um diretório onde, por padrão, as chaves públicas e privadas serão geradas por um comando ssh-keygen dentro do diretório home.

Normalmente, os pacotes de instalação do Zabbix server de diferentes distribuições de SO criam a conta de usuário *zabbix* com um diretório home em outro local, por exemplo, */var/lib/zabbix* (como para contas de sistema).

Antes de gerar as chaves, você pode realocar o diretório home para */home/zabbix*, para que ele corresponda ao parâmetro de configuração do Zabbix server `SSHKeyLocation` mencionado acima.

Note:
As etapas a seguir podem ser ignoradas se a conta *zabbix* foi adicionada manualmente de acordo com a seção de instalação. Nesse caso, o diretório home para a conta *zabbix* provavelmente já é */home/zabbix*.

Para alterar o diretório home da conta de usuário *zabbix*, todos os processos em execução que o utilizam devem ser parados:

```
systemctl stop zabbix-agent
systemctl stop zabbix-server
```

Para alterar o local do diretório home com uma tentativa de movê-lo (se ele existir), o seguinte comando deve ser executado:

```
usermod -m -d /home/zabbix zabbix
```

Também é possível que um diretório home não existisse no local antigo, então ele deve ser criado no novo local. Uma tentativa segura de fazer isso é:

```
test -d /home/zabbix || mkdir /home/zabbix
```

Para garantir que tudo está seguro, comandos adicionais podem ser executados para definir permissões para o diretório home:

```
chown zabbix:zabbix /home/zabbix
chmod 700 /home/zabbix
```

Os processos parados anteriormente agora podem ser iniciados novamente:

```
systemctl start zabbix-agent
systemctl start zabbix-server
```

Agora, as etapas para gerar as chaves públicas e privadas podem ser realizadas com os seguintes comandos (para melhor legibilidade, os prompts dos comandos estão comentados):

```
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):
<Deixe em branco>
##### Enter same passphrase again:
<Deixe em branco>
##### 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:
As chaves pública e privada (*id_rsa.pub* e *id_rsa*) foram geradas por padrão no diretório */home/zabbix/.ssh*, que corresponde ao parâmetro de configuração `SSHKeyLocation` do Zabbix server.

Attention:
Tipos de chave diferentes de "rsa" podem ser suportados pela ferramenta ssh-keygen e por servidores SSH, mas podem não ser suportados pela libssh2 usada pelo Zabbix.

Formulário de configuração do Shell

Esta etapa deve ser executada apenas uma vez para cada host que será monitorado por verificações SSH.

Usando os seguintes comandos, o arquivo de chave **pública** pode ser instalado em um host remoto *10.10.10.10*, para que as verificações SSH possam ser realizadas com uma conta *root* (para melhor legibilidade, os prompts de comando estão comentados):

```
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.
```

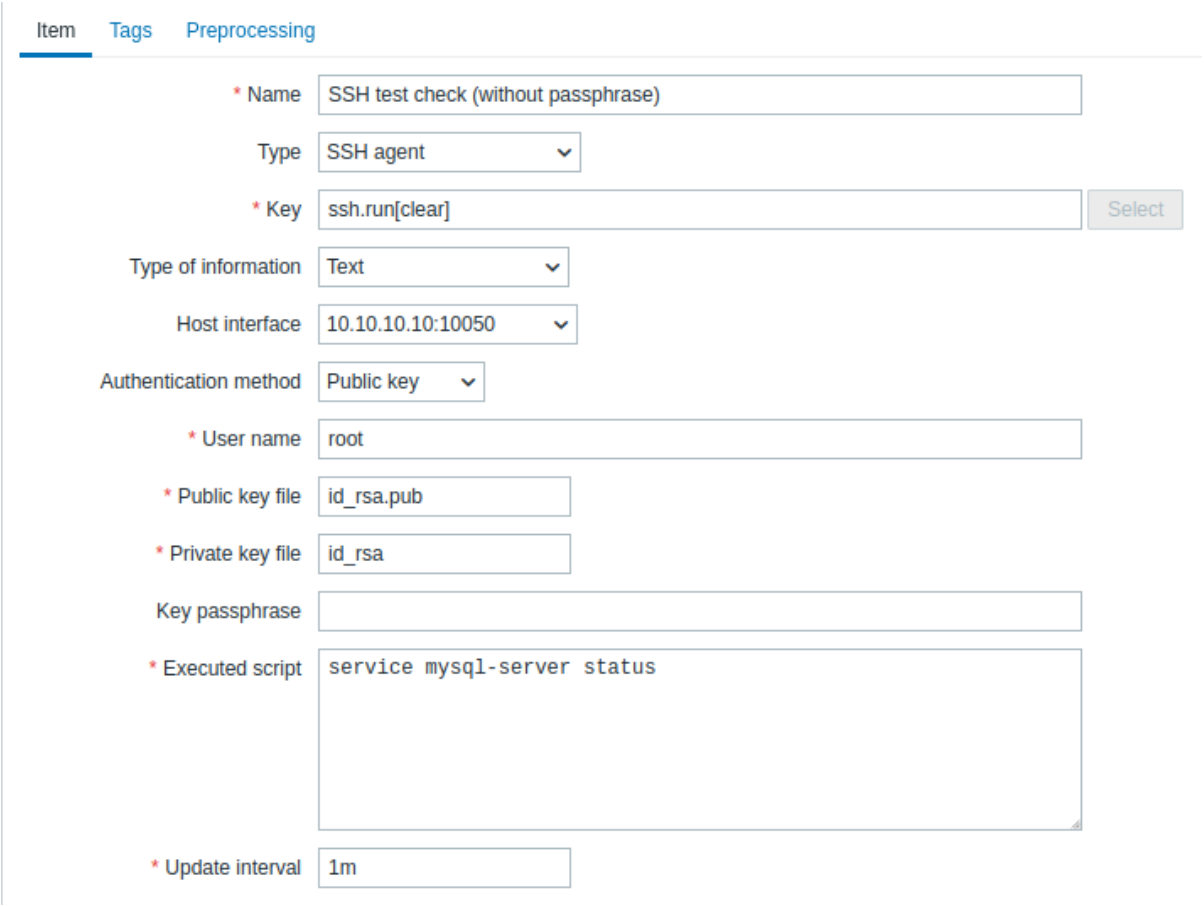
Agora é possível verificar o login SSH usando a chave privada padrão (*/home/zabbix/.ssh/id_rsa*) para a conta de usuário *zabbix*:

```
sudo -u zabbix ssh root@10.10.10.10
```

Se o login for bem-sucedido, a parte de configuração no shell estará concluída e a sessão SSH remota poderá ser fechada.

Configuração do item

O(s) comando(s) real(is) a ser(em) executado(s) devem ser colocados no campo *Script executado* na configuração do item. Vários comandos podem ser executados um após o outro, colocando-os em uma nova linha. Neste caso, os valores retornados também serão formatados como multilinhas.



Todos os campos obrigatórios estão marcados com um asterisco vermelho.

Os campos que exigem informações específicas para itens SSH são:

Parâmetro	Descrição	Comentários
Tipo	Selecione **SSH agent** aqui.	

Parâmetro	Descrição	Comentários

Chave	Chave de item única (por host) no formato **ssh.run[descrição curta única,<ip>,<porta>,<codificação>,<opções ssh>]**	**descrição curta única** é obrigatória e deve ser única para cada item SSH por host.
		
		A porta padrão é 22, não a porta especificada na interface à qual este item está atribuído.
		
		opções ssh permitem passar opções SSH adicionais no formato *chave1=valor1;chave2=valor2,valor3*. Vários valores para uma chave podem ser passados separados por vírgula (neste caso, o parâmetro deve ser entre aspas); várias chaves de opção podem ser passadas separadas por ponto e vírgula.
		
		As seguintes chaves de opção são suportadas: `KexAlgorithms`, `HostkeyAlgorithms`, `Ciphers`, `MACs`. O suporte à chave de opção e ao valor depende da biblioteca SSH; se uma opção não for suportada, um erro será retornado e o item se tornará não suportado.
		
		Observe que o sinal "+" para anexar configurações de cifra e "!" para desabilitar configurações de cifra específicas (como no GnuTLS e OpenSSL) não são suportados.
		
		Exemplos:
		=>
		`ssh.run[KexAlgorithms,127.0.0.1,,,Ciphers=aes128`
		=>
		`ssh.run[KexAlgorithms,,,,"KexAlgorithms=diffie-h`

Método de autenticação	Um dos "Senha" ou "Chave pública".	
Nome de usuário	Nome de usuário (até 255 caracteres) para autenticar no host remoto. Obrigatório.	
Arquivo de chave pública	Nome do arquivo da chave pública se o *Método de autenticação* for "Chave pública". Obrigatório.	Exemplo: *id_rsa.pub* - nome padrão do arquivo de chave pública gerado por um comando ssh-keygen.
Arquivo de chave privada	Nome do arquivo da chave privada se o *Método de autenticação* for "Chave pública". Obrigatório.	Exemplo: *id_rsa* - nome padrão do arquivo de chave privada.
Senha ou *Senha da chave*	Senha (até 255 caracteres) para autenticação ou Senha **se** ela foi usada para a chave privada.	Deixe o campo *Senha da chave* vazio se a senha não foi usada. Veja também problemas conhecidos sobre o uso de senha.
Script executado	Comando(s) de shell executado(s) usando sessão remota SSH.	O valor de retorno do(s) comando(s) de shell executado(s) é limitado a 16MB (incluindo espaços em branco finais que são truncados); limites do banco de dados também se aplicam. Observe que a biblioteca libssh2 pode truncar scripts executáveis para ~32kB. Exemplos: *date +%s* *systemctl status mysql-server* *ps auxww | grep httpd | wc -l*

10 Verificações Telnet

Visão geral

As verificações Telnet são realizadas como monitoramento sem agent. O agent Zabbix não é necessário para verificações Telnet.

Campos configuráveis

O(s) comando(s) real(is) a ser(em) executado(s) devem ser colocados no campo **Script executado** na configuração do item.
Vários comandos podem ser executados um após o outro, colocando-os em uma nova linha. Neste caso, o valor retornado também será formatado como multilinhas.

Caracteres suportados com os quais o prompt do shell pode terminar:

- $
- #
- >
- %

Note:
Uma linha de prompt do telnet que termina com um desses caracteres será removida do valor retornado, mas apenas para o primeiro comando na lista de comandos, ou seja, apenas no início da sessão telnet.

Chave	Descrição
telnet.run[<descrição curta única>,<ip>,<porta>,<codificação>]	Executa um comando em um dispositivo remoto usando conexão telnet

Attention:
Se uma verificação telnet retornar um valor com caracteres não-ASCII e em codificação diferente de UTF8, então o parâmetro *<codificação>* da chave deve ser especificado corretamente. Veja a página codificação dos valores retornados para mais detalhes.

11 Verificações externas

Visão geral

A verificação externa é uma verificação executada pelo Zabbix server por meio da execução de um script shell ou de um binário. No entanto, quando os hosts são monitorados por um Zabbix proxy, as verificações externas são executadas pelo proxy.

As verificações externas não exigem que nenhum agent esteja em execução no host monitorado.

A sintaxe da chave do item é:

`script[<parâmetro1>,<parâmetro2>,...]`

Onde:

ARGUMENTO	DEFINIÇÃO
script	Nome de um script shell ou de um binário.
parâmetro(s)	Parâmetros de linha de comando opcionais.

Se você não quiser passar nenhum parâmetro para o script, poderá usar:

```
script[] ou
script
```

O Zabbix server ou proxy pesquisará o diretório especificado para scripts externos e executará o comando (consulte o parâmetro `ExternalScripts` no arquivo de configuração do Zabbix server/proxy). O comando será executado com o mesmo usuário do Zabbix server/proxy, portanto, quaisquer permissões de acesso ou variáveis de ambiente devem ser tratadas em um script wrapper, se necessário. As permissões no comando também devem permitir que esse usuário o execute. Somente comandos no diretório especificado estão disponíveis para execução.

Warning:
Não use excessivamente as verificações externas, pois cada script exige o início de um processo fork pelo Zabbix server/proxy, e a execução de muitos scripts pode diminuir significativamente o desempenho do Zabbix.

Exemplo de uso

Executando o script **check_oracle.sh** com o primeiro parâmetro '-h'. O segundo parâmetro será substituído pelo endereço IP ou nome DNS, dependendo da seleção nas propriedades do host.

```
check_oracle.sh["-h","{HOST.CONN}"]
```

Assumindo que o host está configurado para usar o endereço IP, o Zabbix server/proxy executará:

```
check_oracle.sh '-h' '192.168.1.4'
```

Resultado da verificação externa

O valor de retorno de uma verificação externa é uma saída padrão juntamente com um erro padrão produzido pela verificação.

Attention:
Um item que retorna texto (caractere, log ou tipo de informação de texto) não ficará não suportado em caso de saída de erro padrão.

O valor de retorno é limitado a 16MB (incluindo espaços em branco à direita que são truncados); limites do banco de dados também se aplicam.

Se o script solicitado não for encontrado ou o Zabbix server/proxy não tiver permissões para executá-lo, o item ficará não suportado e uma mensagem de erro correspondente será exibida.

Em caso de timeout, o item ficará não suportado, uma mensagem de erro correspondente será exibida e o processo bifurcado para o script será encerrado.

12 Itens trapper

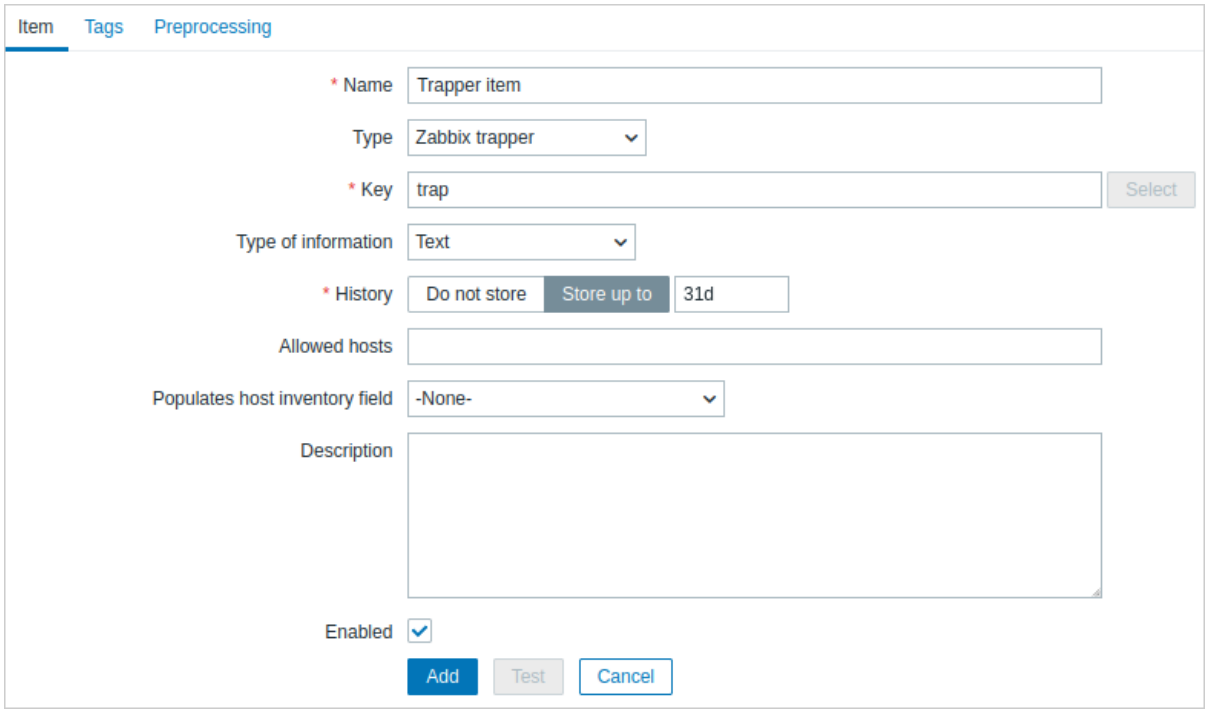
Visão geral

Itens do tipo trapper aceitam dados recebidos em vez de consultá-los. Isso é útil para qualquer dado que você queira enviar ao Zabbix.

Configuração

Para configurar um item trapper:

1. Vá para *Coleta de dados* → *Hosts*.
2. Clique em *Itens* na linha do host.
3. Clique em *Criar item*.
4. Insira os parâmetros do item no formulário de configuração.



Todos os campos obrigatórios estão marcados com um asterisco vermelho.

Os campos que requerem informações específicas para itens trapper são:

Tipo	Selecione ”Zabbix trapper”.
Chave	Insira uma chave que será usada para reconhecer o item ao enviar dados para o Zabbix server.
Tipo de informação	Selecione o tipo de informação que corresponderá ao formato dos dados que serão enviados.
Hosts permitidos	Lista de endereços IP separados por vírgula (opcionalmente na notação CIDR) ou nomes DNS. Se especificado, conexões recebidas serão aceitas apenas dos hosts listados aqui. Se o suporte a IPv6 estiver habilitado, então ’127.0.0.1’, ’::127.0.0.1’, ’::ffff:127.0.0.1’ são tratados igualmente e ’::/0’ permitirá qualquer endereço IPv4 ou IPv6. ’0.0.0.0/0’ pode ser usado para permitir qualquer endereço IPv4. Observe que ”endereços IPv6 compatíveis com IPv4” (prefixo 0000::/96) são suportados, mas obsoletos pelo RFC4291. Exemplo: 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} Espaços, macros de usuário e macros de host {HOST.HOST}, {HOST.NAME}, {HOST.IP}, {HOST.DNS}, {HOST.CONN} são suportados.

Note:
Antes de enviar valores, pode ser necessário aguardar até 60 segundos após salvar o item até que o Zabbix server atualize o cache de configuração.

Enviando dados

O envio de dados para o Zabbix server ou proxy é possível usando a ferramenta Zabbix sender ou o protocolo do Zabbix sender. O envio de dados para o Zabbix server também é possível usando o método de API `history.push`.

Zabbix sender

Para enviar dados para o Zabbix server ou proxy usando o utilitário Zabbix sender, você pode executar o seguinte comando para enviar o ”test value”:

```
zabbix_sender -z <endereço IP do server> -p 10051 -s "New host" -k trap -o "test value"
```

Para enviar o ”test value”, as seguintes opções de comando são usadas:

- `-z` para especificar o endereço IP do Zabbix server
- `-p` para especificar o número da porta do Zabbix server (10051 por padrão)
- `-s` para especificar o host (certifique-se de usar o nome técnico em vez do nome visível do host)
- `-k` para especificar a chave do item configurado no item trapper
- `-o` para especificar o valor a ser enviado

Attention:
O processo Zabbix trapper não expande macros usadas na chave do item para verificar a existência da chave do item correspondente para o host de destino.

Para mais informações sobre a comunicação entre o Zabbix sender e o Zabbix server ou proxy, consulte *Protocolo do Zabbix sender*.

history.push

Para enviar dados para o Zabbix server usando o método de API `history.push`, você pode fazer a seguinte solicitação HTTP POST contendo alguns valores de teste:

```
curl --request POST \
  --url 'https://example.com/zabbix/api_jsonrpc.php' \
  --header 'Authorization: Bearer 0424bd59b807674191e7d77572075f33' \
  --header 'Content-Type: application/json-rpc' \
  --data '{"jsonrpc":"2.0","method":"history.push","params":[{"itemid":10600,"value":"test value 1"},{"ite
```

Se a solicitação estiver correta, a resposta retornada pela API pode ser semelhante a:

```
{
    "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
}
```


Os erros nos dados de resposta indicam que o envio de dados para itens específicos falhou na validação pelo Zabbix server. Isso pode acontecer pelos seguintes motivos:

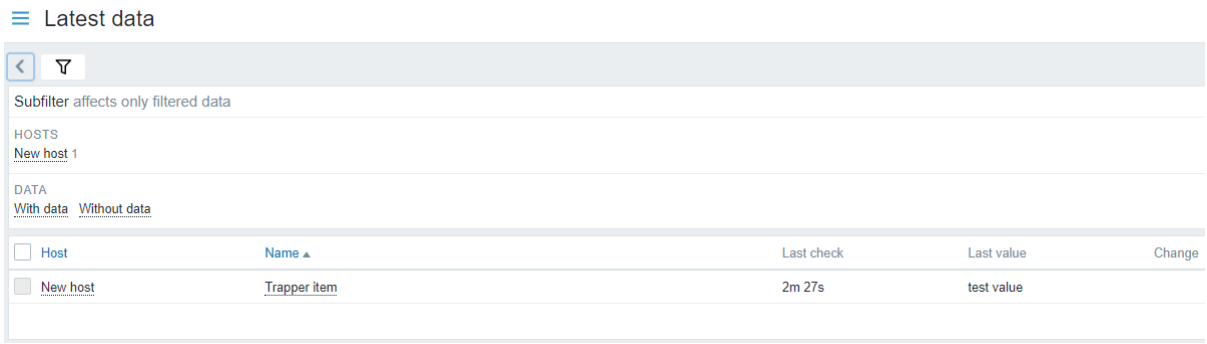
- o usuário que está enviando os dados não tem permissão de *leitura* para o host do item;
- o host está desabilitado ou em manutenção sem coleta de dados;
- o item não existe ou ainda não está incluído no cache de configuração do server;
- o item está desabilitado ou seu tipo é diferente de Zabbix trapper ou HTTP agent (com trapping habilitado);
- o IP ou DNS do usuário não está definido na lista de *Allowed hosts* do item;
- outro item possui um valor com timestamp duplicado no nível de nanossegundos.

A ausência de erros indica que os valores enviados foram aceitos para processamento, o que inclui pré-processamento (se houver), processamento de trigger e salvamento no banco de dados. Observe que o processamento de um valor aceito também pode falhar (por exemplo, durante o pré-processamento), resultando no descarte do valor.

Para mais informações sobre como trabalhar com a API do Zabbix, consulte API.

Exibindo dados

Depois que os dados são enviados, você pode navegar até *Monitoramento* → *Últimos dados* para ver o resultado:



Note:
Se um único valor numérico for enviado, o gráfico de dados mostrará uma linha horizontal à esquerda e à direita do ponto de tempo do valor.

13 Monitoramento JMX

Visão geral

O monitoramento JMX pode ser usado para monitorar contadores JMX de uma aplicação Java.

O monitoramento JMX possui suporte nativo no Zabbix na forma de um daemon Zabbix chamado "Zabbix Java gateway".

Para recuperar o valor de um contador JMX específico em um host, o Zabbix server consulta o **Java gateway** do Zabbix, que por sua vez usa a API de gerenciamento JMX para consultar remotamente a aplicação de interesse.

Para mais detalhes e configuração, consulte a seção Zabbix Java gateway.

> **Warning:**
> A comunicação entre o Java gateway e a aplicação JMX monitorada não deve ser bloqueada por firewall.

Habilitando o monitoramento remoto JMX para aplicações Java

Uma aplicação Java não precisa de nenhum software adicional instalado, mas precisa ser iniciada com as opções de linha de comando especificadas abaixo para ter suporte ao monitoramento remoto JMX.

No mínimo, se você deseja apenas começar monitorando uma aplicação Java simples em um host local sem segurança aplicada, inicie-a com estas opções:

```
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
```

Isso faz com que o Java escute conexões JMX na porta 12345, apenas do host local, e diz para não exigir autenticação ou SSL.

Se você quiser permitir conexões em outra interface, defina o parâmetro -Djava.rmi.server.hostname para o IP dessa interface.

Se você deseja ser mais rigoroso quanto à segurança, há muitas outras opções Java disponíveis para você. Por exemplo, o próximo exemplo inicia a aplicação com um conjunto de opções mais versátil e a abre para uma rede mais ampla, não apenas o host local.

```
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
```

A maioria (senão todas) dessas configurações pode ser especificada em `$JRE/lib/management/management.properties` (ou onde quer que esse arquivo esteja em seu sistema).

Observe que, se você deseja usar SSL, é necessário modificar o script startup.sh adicionando as opções `-Djavax.net.ssl.*` ao Java gateway, para que ele saiba onde encontrar os arquivos de chave e trust stores.

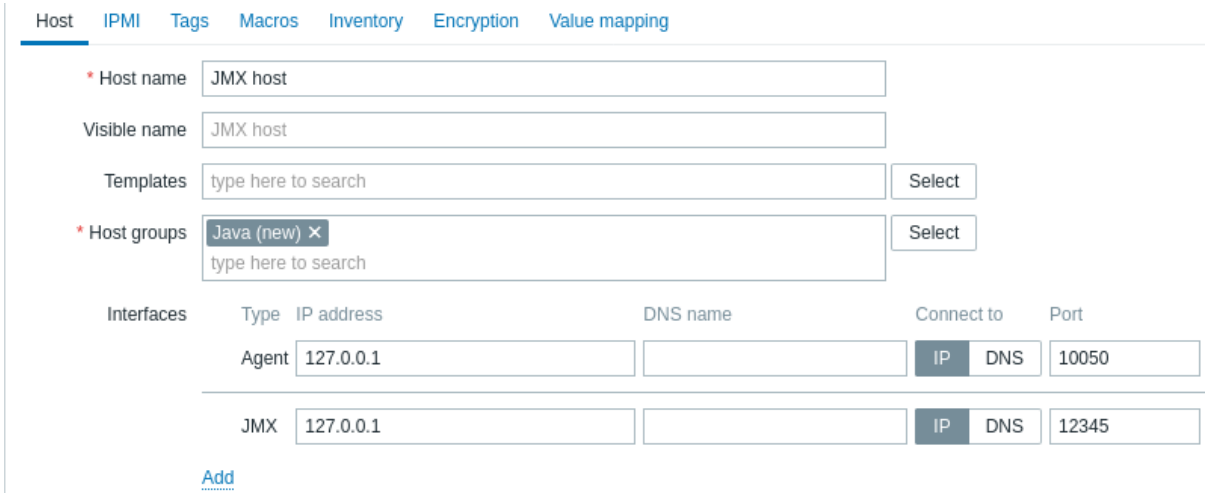
Veja Monitoring and Management Using JMX para uma descrição detalhada.

Configurando interfaces e items JMX no frontend do Zabbix

Com o Java gateway em execução, o server sabendo onde encontrá-lo e um aplicativo Java iniciado com suporte para monitoramento remoto JMX, é hora de configurar as interfaces e items na interface gráfica do Zabbix.

Configurando a interface JMX

Você começa criando uma interface do tipo JMX no host de interesse.

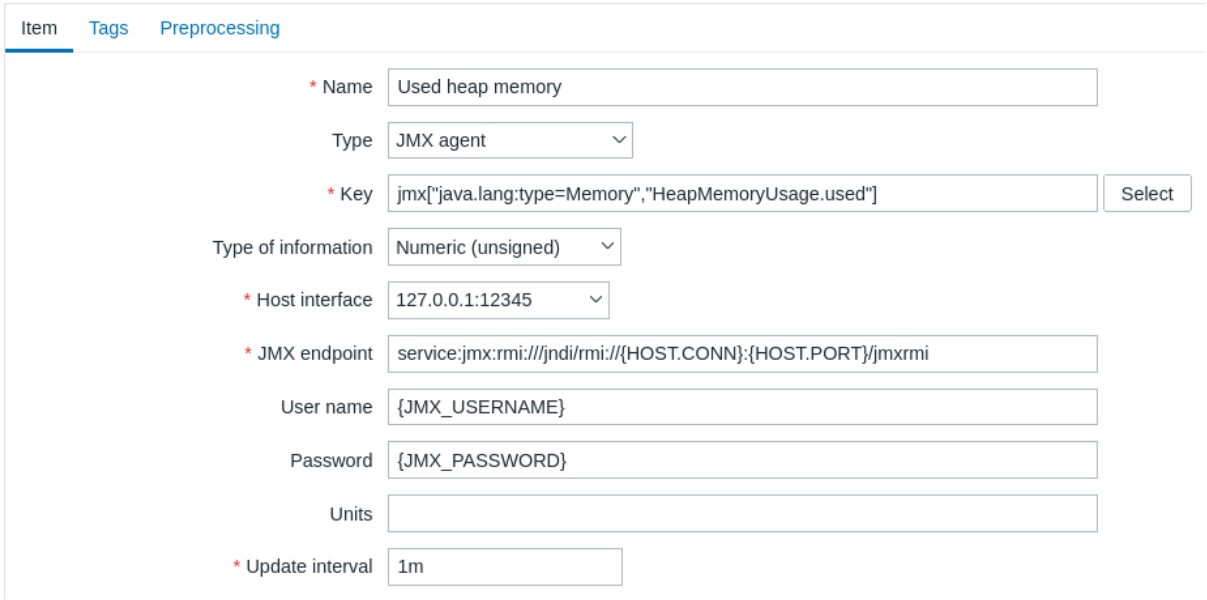


Todos os campos obrigatórios estão marcados com um asterisco vermelho.

Adicionando um item de agent JMX

Para cada contador JMX que você deseja monitorar, adicione um item de **agent JMX** vinculado àquela interface.

A chave na captura de tela abaixo diz `jmx["java.lang:type=Memory","HeapMemoryUsage.used"]`.



Todos os campos obrigatórios estão marcados com um asterisco vermelho.

Os campos que exigem informações específicas para itens JMX são:

Tipo	Defina **agent JMX** aqui.
Chave	A chave do item `jmx[]` contém três parâmetros: **nome do objeto** - o nome do objeto de um MBean **nome do atributo** - um nome de atributo de MBean com nomes de campos de dados compostos opcionais separados por pontos **descrição curta exclusiva** - uma descrição exclusiva que permite vários itens JMX com o mesmo nome de objeto e nome de atributo no host (opcional) Veja abaixo mais detalhes sobre as chaves dos itens JMX. Você pode descobrir MBeans e atributos de MBean usando um item de descoberta de baixo nível jmx.discovery[].
Endpoint JMX	Você pode especificar um endpoint JMX personalizado. Certifique-se de que os parâmetros de conexão do endpoint JMX correspondam à interface JMX. Isso pode ser feito usando macros {HOST.*} como feito no endpoint JMX padrão. As macros {HOST.*} e macros de usuário são suportadas.
Nome de usuário	Especifique o nome de usuário (até 255 caracteres), se você configurou autenticação em seu aplicativo Java. Macros de usuário são suportadas.

Senha	Especifique a senha (até 255 caracteres), se você configurou autenticação em seu aplicativo Java. Macros de usuário são suportadas.

Se você deseja monitorar um contador Booleano que seja ”true” ou ”false”, então especifique o tipo de informação como ”Numérico (sem sinal)” e selecione a etapa de pré-processamento ”Booleano para decimal” na aba Pré-processamento. O server armazenará valores Booleanos como 1 ou 0, respectivamente.

Chaves de item JMX em mais detalhes

Atributos simples

Um nome de objeto MBean nada mais é do que uma string que você define em sua aplicação Java. Um nome de atributo, por outro lado, pode ser mais complexo. Caso um atributo retorne um tipo de dado primitivo (um inteiro, uma string, etc.), não há com o que se preocupar, a chave será assim:

```
jmx[com.example:Type=Hello,weight]
```

Neste exemplo, o nome do objeto é ”com.example:Type=Hello”, o nome do atributo é ”weight” e o tipo de valor retornado provavelmente deve ser ”Numérico (flutuante)”.

Atributos que retornam dados compostos

Fica mais complicado quando seu atributo retorna dados compostos. Por exemplo: o nome do seu atributo é ”apple” e ele retorna um hash representando seus parâmetros, como ”weight”, ”color” etc. Sua chave pode ser assim:

```
jmx[com.example:Type=Hello,apple.weight]
```

É assim que o nome do atributo e a chave do hash são separados, usando um ponto. Da mesma forma, se um atributo retorna dados compostos aninhados, as partes são separadas por um ponto:

```
jmx[com.example:Type=Hello,fruits.apple.weight]
```

Atributos que retornam dados tabulares

Atributos de dados tabulares consistem em um ou vários atributos compostos. Se tal atributo for especificado no parâmetro de nome do atributo, então este valor de item retornará a estrutura completa do atributo no formato JSON. Os valores dos elementos individuais dentro do atributo de dados tabulares podem ser recuperados usando pré-processamento.

Exemplo de atributo de dados tabulares:

```
jmx[com.example:type=Hello,foodinfo]
```

Valor do item:

```
[
  {
    "a": "apple",
    "b": "banana",
    "c": "cherry"
  },
  {
    "a": "potato",
    "b": "lettuce",
    "c": "onion"
  }
]
```

Problema com pontos

Até agora tudo bem. Mas e se um nome de atributo ou uma chave de hash contiver um ponto? Aqui está um exemplo:

```
jmx[com.example:Type=Hello,all.fruits.apple.weight]
```

Isso é um problema. Como dizer ao Zabbix que o nome do atributo é ”all.fruits”, e não apenas ”all”? Como distinguir um ponto que faz parte do nome de um ponto que separa um nome de atributo e chaves de hash?

Isso é possível, tudo o que você precisa fazer é escapar os pontos que fazem parte do nome com uma barra invertida:

```
jmx[com.example:Type=Hello,all\.fruits.apple.weight]
```

Da mesma forma, se sua chave de hash contiver um ponto, você deve escapá-lo:

```
jmx[com.example:Type=Hello,all\.fruits.apple.total\.weight]
```

Outros problemas

Um caractere de barra invertida em um nome de atributo deve ser escapado:

```
jmx[com.example:type=Hello,c:\\documents]
```

Para lidar com quaisquer outros caracteres especiais na chave do item JMX, consulte a seção de formato de chave de item section.

Isso é tudo o que há para isso. Feliz monitoramento JMX!

Tipos de dados não primitivos

É possível trabalhar com MBeans personalizados retornando tipos de dados não primitivos, que substituem o método **toString()**.

Usando endpoint personalizado com JBoss EAP 6.4

Endpoints personalizados permitem trabalhar com diferentes protocolos de transporte além do RMI padrão.

Para ilustrar essa possibilidade, vamos tentar configurar o monitoramento do JBoss EAP 6.4 como exemplo. Primeiro, vamos fazer algumas suposições:

- Você já instalou o Zabbix Java gateway. Caso não tenha feito isso, siga a documentação.
- O Zabbix server e o Java gateway estão instalados com o prefixo `/usr/local/`
- O JBoss já está instalado em `/opt/jboss-eap-6.4/` e está rodando em modo standalone
- Vamos assumir que todos esses componentes estão funcionando no mesmo host
- O firewall e o SELinux estão desativados (ou configurados adequadamente)

Vamos fazer algumas configurações simples no zabbix_server.conf:

```
JavaGateway=127.0.0.1
StartJavaPollers=5
```

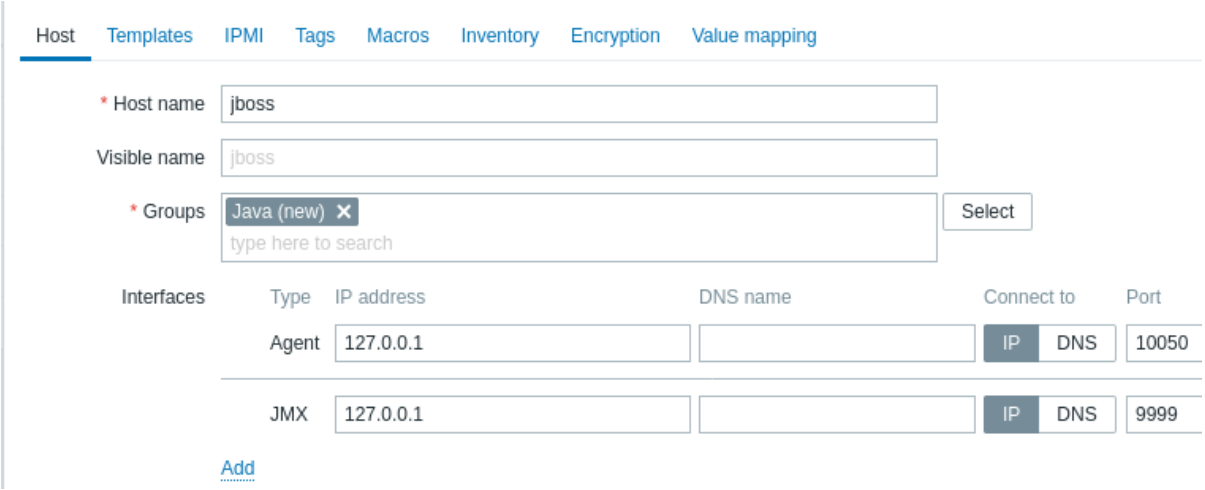
E no arquivo de configuração `zabbix_java/settings.sh` (ou `zabbix_java_gateway.conf`):

```
START_POLLERS=5
```

Verifique se o JBoss está ouvindo em sua porta de gerenciamento padrão:

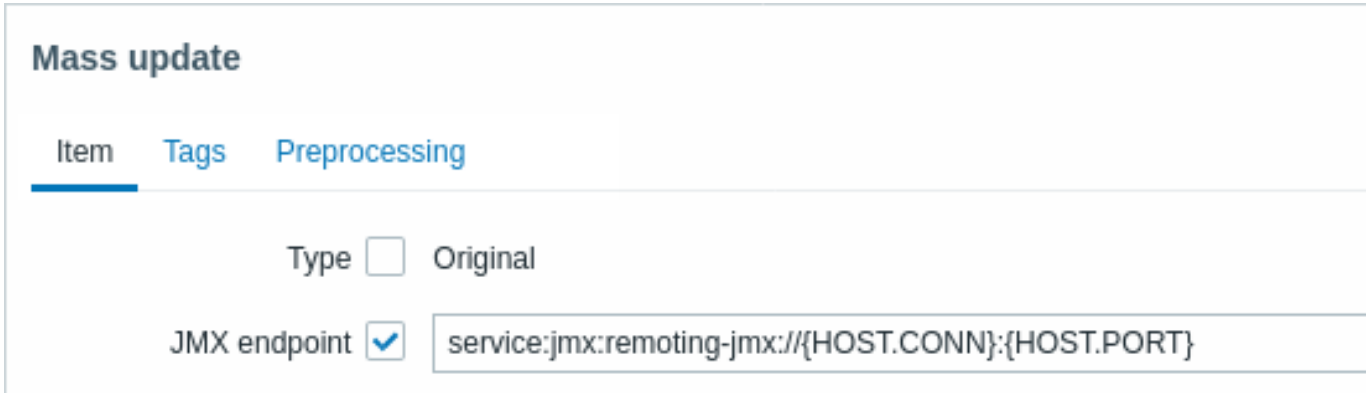
```
$ netstat -natp | grep 9999
tcp        0      0 127.0.0.1:9999          0.0.0.0:*               LISTEN      10148/java
```

Agora vamos criar um host com a interface JMX 127.0.0.1:9999 no Zabbix.



Como sabemos que esta versão do JBoss usa o protocolo JBoss Remoting em vez do RMI, podemos atualizar em massa o parâmetro endpoint JMX para os items em nosso template JMX da seguinte forma:

```
service:jmx:remoting-jmx://{HOST.CONN}:{HOST.PORT}
```



Vamos atualizar o cache de configuração:

```
/usr/local/sbin/zabbix_server -R config_cache_reload
```

Observe que você pode encontrar um erro inicialmente.

```
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" significa que o Java gateway não sabe como trabalhar com o protocolo especificado. Isso pode ser corrigido criando um arquivo `~/needed_modules.txt` com o seguinte conteúdo:

```
jboss-as-remoting
jboss-logging
jboss-logmanager
jboss-marshalling
jboss-remoting
jboss-sasl
jcl-over-slf4j
jul-to-slf4j-stub
log4j-jboss-logmanager
remoting-jmx
slf4j-api
xnio-api
xnio-nio
```

e então executando o comando:

```
for i in $(cat ~/needed_modules.txt); do find /opt/jboss-eap-6.4 -iname "${i}*.jar" -exec cp '{}' /usr/loc
```

Assim, o Java gateway terá todos os módulos necessários para trabalhar com jmx-remoting. O que resta é reiniciar o Java gateway, aguardar um pouco e, se você fez tudo certo, verá que os dados de monitoramento JMX começam a chegar no Zabbix (veja também: Últimos dados).

14 Monitoramento ODBC

Visão geral

O monitoramento ODBC corresponde ao tipo de item **Database monitor** no frontend do Zabbix.

ODBC é uma API middle-ware da linguagem de programação C para acessar sistemas de gerenciamento de banco de dados (DBMS). O conceito ODBC foi desenvolvido pela Microsoft e posteriormente portado para outras plataformas.

O Zabbix pode consultar qualquer banco de dados que seja suportado pelo ODBC. Para isso, o Zabbix não se conecta diretamente aos bancos de dados, mas usa a interface ODBC e os drivers configurados no ODBC. Isso permite um monitoramento mais eficiente de diferentes bancos de dados para múltiplos propósitos (por exemplo, verificar filas específicas do banco de dados, estatísticas de uso, etc.).

O Zabbix suporta unixODBC, que é uma das implementações de API ODBC de código aberto mais utilizadas.

Attention:
Veja também: problemas conhecidos para verificações ODBC.

Instalando o unixODBC

A maneira sugerida de instalar o unixODBC é usar os repositórios de pacotes padrão do sistema operacional Linux. Nas distribuições Linux mais populares, o unixODBC está incluído no repositório de pacotes por padrão. Se os pacotes não estiverem disponíveis, os arquivos-fonte podem ser obtidos na página inicial do unixODBC: http://www.unixodbc.org/download.html.

Para instalar o unixODBC, use o gerenciador de pacotes para o sistema de sua escolha:

```
##### Para sistemas Ubuntu/Debian:
apt install unixodbc unixodbc-dev

##### Para sistemas baseados em RedHat/Fedora:
dnf install unixODBC unixODBC-devel

##### Para sistemas baseados em SUSE:
zypper in unixODBC-devel
```

Attention:
O pacote `unixodbc-dev` ou `unixODBC-devel` é necessário para compilar o Zabbix com suporte ao unixODBC. Para habilitar o suporte ODBC, o Zabbix deve ser compilado com a seguinte opção de configuração:

`--with-unixodbc[=ARG] # Use ODBC driver against unixODBC package.`

Instalando drivers unixODBC

O driver de banco de dados unixODBC deve ser instalado para o banco de dados que será monitorado. Para uma lista de bancos de dados e drivers suportados, consulte a página inicial do unixODBC: http://www.unixodbc.org/drivers.html.

Note:
Em algumas distribuições Linux, os drivers de banco de dados estão incluídos nos repositórios de pacotes.

MySQL

Para instalar o driver de banco de dados MySQL unixODBC, use o gerenciador de pacotes para o sistema de sua escolha:

```
##### Para sistemas Ubuntu/Debian:
apt install odbc-mariadb

##### Para sistemas baseados em RedHat/Fedora:
dnf install mariadb-connector-odbc

##### Para sistemas baseados em SUSE:
zypper install mariadb-connector-odbc
```

Para instalar o driver de banco de dados sem um gerenciador de pacotes, consulte a documentação do MySQL para `mysql-connector-odbc`, ou a documentação do MariaDB para `mariadb-connector-odbc`.

PostgreSQL

Para instalar o driver de banco de dados unixODBC do PostgreSQL, use o gerenciador de pacotes para o sistema de sua escolha:

```
##### Para sistemas Ubuntu/Debian:
apt install odbc-postgresql

##### Para sistemas baseados em RedHat/Fedora:
dnf install postgresql-odbc

##### Para sistemas baseados em SUSE:
zypper install psqlODBC
```

Para instalar o driver de banco de dados sem um gerenciador de pacotes, consulte a documentação do PostgreSQL.

Oracle

Para instalar o driver de banco de dados unixODBC, consulte a documentação da Oracle.

MSSQL

Para instalar o driver de banco de dados MSSQL unixODBC, use o gerenciador de pacotes do sistema de sua escolha:

```
##### Para sistemas Ubuntu/Debian:
apt install tdsodbc

##### Para sistemas baseados em RedHat/Fedora (pacotes EPEL: https://docs.fedoraproject.org/en-US/epel/):
dnf install epel-release
dnf install freetds

##### Para sistemas baseados em SUSE:
zypper install libtdsodbc0
```

Para instalar o driver de banco de dados sem um gerenciador de pacotes, consulte o Guia do usuário do FreeTDS.

Configurando o unixODBC

Para configurar o unixODBC, você deve editar os arquivos `odbcinst.ini` e `odbc.ini`. Você pode verificar a localização desses arquivos executando o seguinte comando:

```
odbcinst -j
```

O resultado do comando deve conter informações semelhantes ao seguinte:

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

odbcinst.ini

O arquivo `odbcinst.ini` lista os drivers de banco de dados ODBC instalados. Se `odbcinst.ini` estiver ausente, é necessário criá-lo manualmente.

```
[TEST_MYSQL]
Description=ODBC for MySQL
Driver=/usr/lib/libmyodbc5.so
FileUsage=1
```

Parâmetro	Descrição
TEST_MYSQL	Nome do driver do banco de dados.
Description	Descrição do driver do banco de dados.
Driver	Localização da biblioteca do driver do banco de dados.
FileUsage	Determina se o driver do banco de dados suporta conectar-se a um servidor de banco de dados sem suporte para acessar arquivos locais (0); suporta leitura de dados de arquivos (1); suporta gravação de dados em arquivos (2).
Threading	Nível de serialização de threads. Suportado para PostgreSQL. Desde 1.6, se o gerenciador de driver for compilado com suporte a threads, você pode adicionar outra entrada de driver.

odbc.ini

O arquivo `odbc.ini` é usado para configurar fontes de dados. Observe que a lista de parâmetros suportados depende do driver do banco de dados (por exemplo, bancos de dados Oracle podem usar ServerName em vez de Server, etc.).

```
[TEST_MYSQL]
Description=MySQL Test Database
Driver=mysql
Server=127.0.0.1
User=root
Password=
Port=3306
Socket=
Database=zabbix
```

Parâmetro	Descrição
TEST_MYSQL	Nome da fonte de dados (DSN).
Description	Descrição da fonte de dados.
Driver	Nome do driver do banco de dados (conforme especificado em `odbcinst.ini`).
Server	IP/DNS do servidor do banco de dados.
User	Usuário do banco de dados para conexão.
Password	Senha do usuário do banco de dados.
Port	Porta de conexão do banco de dados.
Socket	Socket de conexão do banco de dados.
Database	Nome do banco de dados.

Para outras opções possíveis de parâmetros de configuração, consulte a documentação do MySQL.

O arquivo `odbc.ini` para PostgreSQL pode conter parâmetros adicionais:

```
[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=7.4+
ShowOidColumn=No
FakeOidIndex=No
RowVersioning=No
ShowSystemTables=No
Fetch=Yes
BoolsAsChar=Yes
SSLmode=Require
ConnSettings=
```

Parâmetro	Descrição
ReadOnly	Especifica se a conexão com o banco de dados permite apenas operações de leitura (consultas `SELECT`) e restringe modificações (comandos `INSERT`, `UPDATE` e `DELETE`); útil para cenários em que os dados devem permanecer inalterados.
Protocol	Versão do protocolo backend do PostgreSQL (ignorado ao usar conexões SSL).
ShowOidColumn	Especifica se deve incluir o Object ID (OID) em SQLColumns.
FakeOidIndex	Especifica se deve criar um índice exclusivo falso no OID.
RowVersioning	Especifica se deve permitir que aplicativos detectem se os dados foram modificados por outros usuários enquanto você está tentando atualizar uma linha. Observe que este parâmetro pode acelerar o processo de atualização, pois, para atualizar uma linha, não é necessário especificar cada coluna individualmente na cláusula `WHERE`.
ShowSystemTables	Especifica se o driver do banco de dados deve tratar tabelas do sistema como tabelas regulares em SQLTables; útil para acessibilidade, permitindo visibilidade nas tabelas do sistema.
Fetch	Especifica se o driver deve usar automaticamente declare cursor/fetch para lidar com instruções `SELECT` e manter um cache de 100 linhas.

Parâmetro	Descrição
BoolsAsChar	Controla o mapeamento de tipos Booleanos. Se definido como "Yes", Bools são mapeados para `SQL_CHAR`; caso contrário, são mapeados para `SQL_BIT`.
SSLmode	Especifica o modo SSL para a conexão.
ConnSettings	Configurações adicionais enviadas para o backend na conexão.

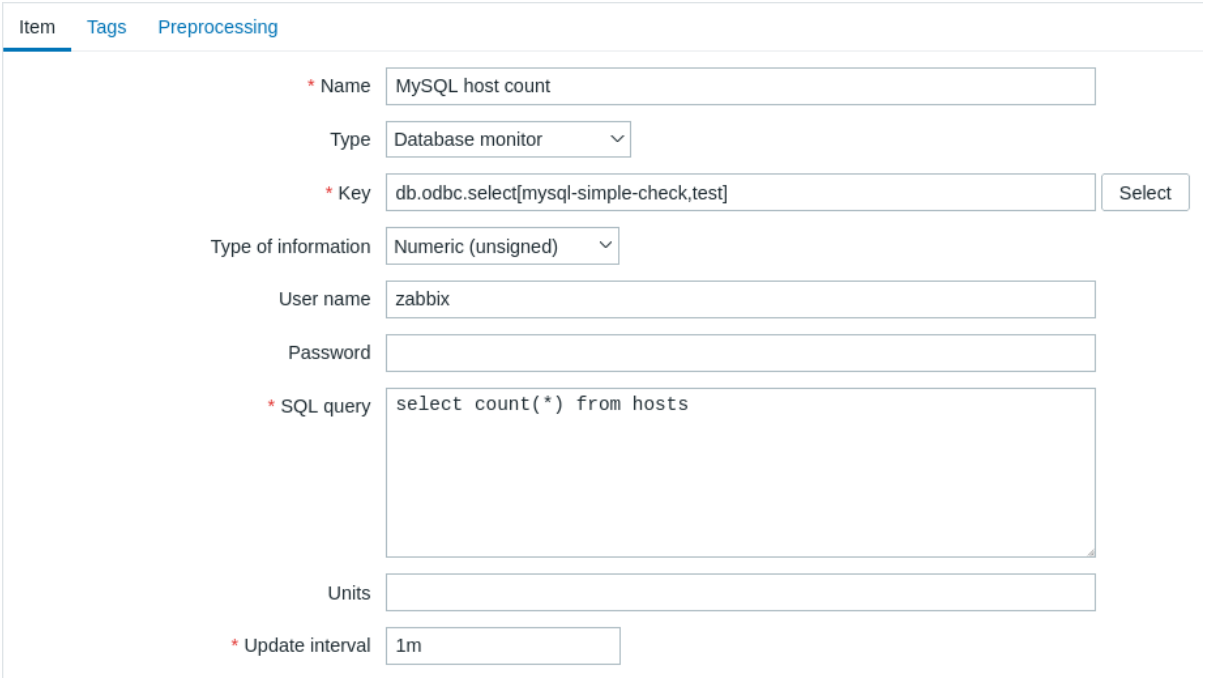
Testando a conexão ODBC

Para testar se a conexão ODBC está funcionando corretamente, você pode usar o utilitário `isql` (incluído no pacote `unixODBC`):

```
isql test
+---------------------------------------+
| Connected!                            |
|                                       |
| sql-statement                         |
| help [tablename]                      |
| quit                                  |
|                                       |
+---------------------------------------+
```

Configuração de item no frontend do Zabbix

Configure um item de **monitoramento de banco de dados**.



Todos os campos obrigatórios estão marcados com um asterisco vermelho.

Para itens de monitoramento de banco de dados, você deve especificar:

Tipo	Selecione "Monitoramento de banco de dados" aqui.
Chave	Insira uma das chaves de item suportadas: **db.odbc.select[]** - este item retorna um valor (a primeira coluna da primeira linha do resultado da consulta SQL); **db.odbc.get[]** - este item retorna várias linhas/colunas em formato JSON; **db.odbc.discovery[]** - este item retorna dados de descoberta de baixo nível.
Nome de usuário	Insira o nome de usuário do banco de dados (até 255 caracteres). Este parâmetro é opcional se o nome de usuário do banco de dados for especificado no arquivo `odbc.ini`. Se uma string de conexão for usada e o campo *Nome de usuário* não estiver vazio, ele será adicionado à string de conexão como `UID=<user>`.

Senha	Insira a senha do usuário do banco de dados (até 255 caracteres). Este parâmetro é opcional se a senha for especificada no arquivo `odbc.ini`. Se uma string de conexão for usada e o campo *Senha* não estiver vazio, ele será adicionado à string de conexão como `PWD=<password>`. Desde o Zabbix 7.0.3, caracteres especiais são suportados neste campo. Antes do Zabbix 7.0.3, se a senha contivesse um ponto e vírgula, ela deveria ser envolvida por chaves, por exemplo, `{P?;)*word}`. Após o 7.0.3, envolver a senha neste caso ainda é suportado, mas não é obrigatório. A senha será adicionada à string de conexão após o nome de usuário como `UID=<username>;PWD={P?;)*word}`. Para testar a string resultante, você pode executar o seguinte comando: `isql -v -k 'Driver=libmaodbc.so;Database=zabbix;UID=zabbix;PWD={P?;)*word}'`
Consulta SQL	Insira a consulta SQL. Observe que com `db.odbc.select[]`, a consulta deve retornar apenas um valor.
Tipo de informação	Selecione aqui o tipo de informação que será retornada pela consulta. Se o tipo de informação for selecionado incorretamente, o item se tornará não suportado.

Notas importantes

- Os itens de monitoramento de banco de dados se tornarão não suportados se nenhum processo *odbc poller* for iniciado na configuração do server ou proxy. Para ativar os ODBC pollers, defina o parâmetro `StartODBCPollers` no arquivo de configuração do Zabbix server ou, para verificações realizadas pelo proxy, no arquivo de configuração do Zabbix proxy.
- O valor do parâmetro *Timeout* no formulário de configuração do item é usado como o tempo limite de login ODBC e o tempo limite de execução da consulta. Observe que essas configurações de tempo limite podem ser ignoradas se o driver ODBC instalado não as suportar.
- O comando SQL deve retornar um conjunto de resultados como qualquer consulta usando a instrução `select`. A sintaxe da consulta dependerá do SGBDR que irá processá-la. A sintaxe da requisição para um procedimento armazenado deve ser iniciada com a palavra-chave `call`.

Detalhes da chave do item

Parâmetros sem colchetes angulares são obrigatórios. Parâmetros marcados com colchetes angulares **< >** são opcionais.

db.odbc.select[<descrição curta única>,<dsn>,<string de conexão>]

 Retorna um valor, ou seja, a primeira coluna da primeira linha do resultado da consulta SQL.
 Valor de retorno: Dependendo da consulta SQL.

Parâmetros:

- **descrição curta única** - uma descrição curta única para identificar o item (para uso em triggers, etc.);
- **dsn** - o nome da fonte de dados (conforme especificado em `odbc.ini`);
- **string de conexão** - a string de conexão (pode conter argumentos específicos do driver).

Comentários:

- Embora `dsn` e `string de conexão` sejam parâmetros opcionais, pelo menos um deles é obrigatório; se ambos forem definidos, `dsn` será ignorado.
- Se uma consulta retornar mais de uma coluna, apenas a primeira coluna será lida. Se uma consulta retornar mais de uma linha, apenas a primeira linha será lida.

db.odbc.get[<descrição curta exclusiva>,<dsn>,<string de conexão>]

 Transforma o resultado da consulta SQL em um array JSON.
 Valor de retorno: *objeto JSON*.

Parâmetros:

- **descrição curta exclusiva** - uma descrição curta exclusiva para identificar o item (para uso em triggers, etc.);
- **dsn** - o nome da fonte de dados (conforme especificado em `odbc.ini`);
- **string de conexão** - a string de conexão (pode conter argumentos específicos do driver).

Comentários:

- Embora `dsn` e `string de conexão` sejam parâmetros opcionais, pelo menos um deles é obrigatório; se ambos forem definidos, `dsn` será ignorado.
- Várias linhas/colunas em formato JSON podem ser retornadas. Este item pode ser usado como um item mestre que coleta todos os dados em uma chamada de sistema, enquanto o pré-processamento JSONPath pode ser usado em itens dependentes

para extrair valores individuais. Para mais informações, veja um exemplo do formato retornado, usado em descoberta de baixo nível.

Exemplo:

```
# Conexão para o driver MySQL ODBC 5:
db.odbc.get[MySQL example,,"Driver=/usr/local/lib/libmyodbc5a.so;Database=master;Server=127.0.0.1;Port=330
```

db.odbc.discovery[<descrição curta exclusiva>,<dsn>,<string de conexão>]

 Transforma o resultado da consulta SQL em um array JSON, usado para descoberta de baixo nível. Os nomes das colunas do resultado da consulta são transformados em nomes de macros de descoberta de baixo nível emparelhados com os valores dos campos descobertos. Essas macros podem ser usadas na criação de protótipos de item, trigger, etc.
 Valor de retorno: *objeto JSON*.

Parâmetros:

- **descrição curta exclusiva** - uma descrição curta exclusiva para identificar o item (para uso em triggers, etc.);
- **dsn** - o nome da fonte de dados (conforme especificado em `odbc.ini`);
- **string de conexão** - a string de conexão (pode conter argumentos específicos do driver).

Comentários:

- Embora `dsn` e `string de conexão` sejam parâmetros opcionais, pelo menos um deles é obrigatório; se ambos forem definidos, `dsn` será ignorado.

Mensagens de erro

As mensagens de erro do ODBC são estruturadas em campos para fornecer informações detalhadas. Por exemplo, uma mensagem de erro pode ser assim:

```
Cannot execute ODBC query: [SQL_ERROR]:[42601][7][ERROR: syntax error at or near ";"; Error while executin
```

- "`Cannot execute ODBC query`" - Mensagem do Zabbix
- "`[SQL_ERROR]`" - Código de retorno do ODBC
- "`[42601]`" - SQLState
- "`[7]`" - Código de erro nativo
- "`[ERROR: syntax error at or near ";"; Error while executing the query]`" - Mensagem de erro nativa

Observe que o comprimento da mensagem de erro é limitado a 2048 bytes, portanto, a mensagem pode ser truncada. Se houver mais de um registro de diagnóstico ODBC, o Zabbix tenta concatená-los (separados por |) até o limite de comprimento permitir.

15 Items dependentes

Visão geral

Existem situações em que um item coleta várias métricas ao mesmo tempo ou até faz mais sentido coletar métricas relacionadas simultaneamente, por exemplo:

- Utilização da CPU de núcleos individuais
- Tráfego de rede de entrada/saída/total

Para permitir a coleta em massa de métricas e o uso simultâneo em vários itens relacionados, o Zabbix suporta itens dependentes. Itens dependentes dependem do item mestre que coleta seus dados simultaneamente, em uma única consulta. Um novo valor para o item mestre preenche automaticamente os valores dos itens dependentes. Itens dependentes não podem ter um intervalo de atualização diferente do item mestre.

As opções de pré-processamento do Zabbix podem ser usadas para extrair a parte necessária para o item dependente a partir dos dados do item mestre.

O pré-processamento é gerenciado por um processo `preprocessing manager`, juntamente com threads de trabalho que executam as etapas de pré-processamento. Todos os valores com pré-processamento (antes do Zabbix 7.0.17, todos os valores), recebidos de diferentes coletores de dados, passam pelo preprocessing manager antes de serem adicionados ao cache de histórico. A comunicação IPC baseada em socket é usada entre os coletores de dados (pollers, trappers, etc) e o processo de pré-processamento.

O Zabbix server ou o Zabbix proxy (se o host for monitorado por proxy) executam as etapas de pré-processamento e processam os itens dependentes.

Itens de qualquer tipo, até mesmo itens dependentes, podem ser definidos como itens mestres. Níveis adicionais de itens dependentes podem ser usados para extrair partes menores do valor de um item dependente existente.

Limitações

- Apenas dependências do mesmo host (template) são permitidas
- Um protótipo de item pode depender de outro protótipo de item ou de um item regular do mesmo host
- A contagem máxima de itens dependentes para um item mestre é limitada a 29999 (independentemente do número de níveis de dependência)
- Máximo de 3 níveis de dependência permitidos
- O item dependente em um host com item mestre de um template não será exportado para XML

Configuração de item

Um item dependente depende de seu item mestre para obter dados. É por isso que o **item mestre** deve ser configurado (ou existir) primeiro:

- Vá para: *Coleta de dados* → *Hosts*
- Clique em *Itens* na linha do host
- Clique em *Criar item*
- Insira os parâmetros do item no formulário

Todos os campos obrigatórios estão marcados com um asterisco vermelho.

Clique em *Adicionar* para salvar o item mestre.

Em seguida, você pode configurar um **item dependente**.

Todos os campos obrigatórios estão marcados com um asterisco vermelho.

Os campos que exigem informações específicas para itens dependentes são:

Tipo	Selecione **Item dependente** aqui.
Chave	Insira uma chave que será usada para reconhecer o item.
Item mestre	Selecione o item mestre. O valor do item mestre será usado para preencher o valor do item dependente.
Tipo de informação	Selecione o tipo de informação que corresponderá ao formato dos dados que serão armazenados.

Você pode usar o pré-processamento do valor do item para extrair a parte necessária do valor do item mestre.

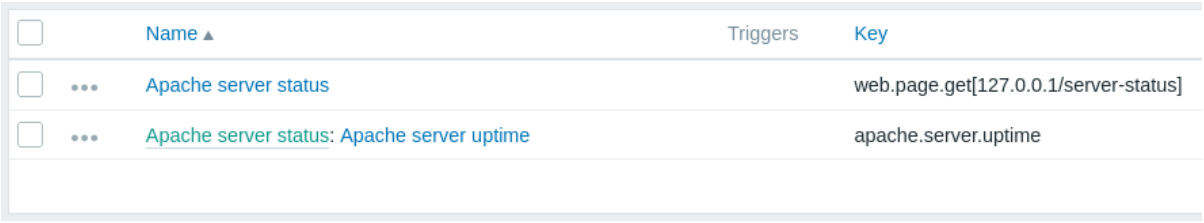
Sem pré-processamento, o valor do item dependente será exatamente o mesmo que o valor do item mestre.

Clique em *Adicionar* para salvar o item dependente.

Um atalho para criar um item dependente mais rapidamente pode ser acessado clicando no botão na lista de itens e selecionando *Criar item dependente*.

Exibição

Na lista de items, os items dependentes são exibidos com o nome do item mestre como prefixo.



Se um item mestre for excluído, todos os seus items dependentes também serão excluídos.

16 Agente HTTP

Visão geral

Este tipo de item permite a coleta de dados usando o protocolo HTTP/HTTPS. A captura também é possível usando a utilidade Zabbix sender ou o protocolo do Zabbix sender (para enviar dados ao Zabbix server ou proxy), ou usando o método de API `history.push` (para enviar dados ao Zabbix server).

As verificações de item HTTP são executadas pelo Zabbix server. No entanto, quando os hosts são monitorados por um Zabbix proxy, as verificações de item HTTP são executadas pelo proxy.

As verificações de item HTTP não requerem nenhum agent em execução no host sendo monitorado.

O agent HTTP suporta tanto HTTP quanto HTTPS. O Zabbix seguirá redirecionamentos opcionalmente (veja a opção *Seguir redirecionamentos* abaixo). O número máximo de redirecionamentos é definido em 10 (usando a opção cURL CURLOPT_MAXREDIRS).

Attention:
O Zabbix server/proxy deve ser inicialmente configurado com suporte a cURL (libcurl).

As verificações HTTP são executadas de forma assíncrona - não é necessário receber a resposta de uma solicitação antes que outras verificações sejam iniciadas. A resolução de DNS também é assíncrona.

A concorrência máxima de verificações assíncronas é 1000 (definida por MaxConcurrentChecksPerPoller).

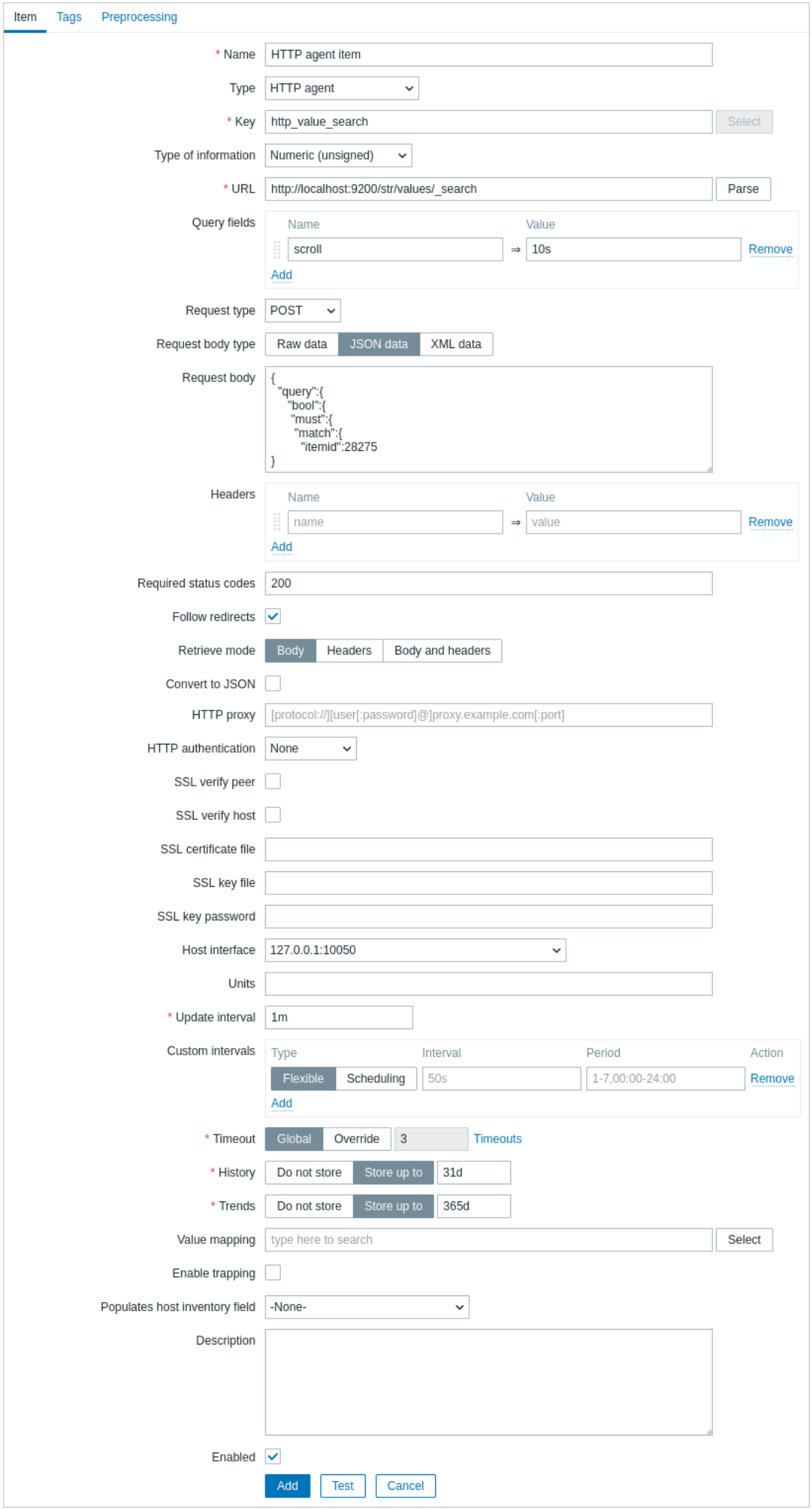
O número de pollers assíncronos de agent HTTP é definido pelo parâmetro StartHTTPAgentPollers.

O recurso de conexões persistentes do cURL foi adicionado às verificações de agent HTTP a partir do Zabbix 7.0.

Configuração

Para configurar um item HTTP:

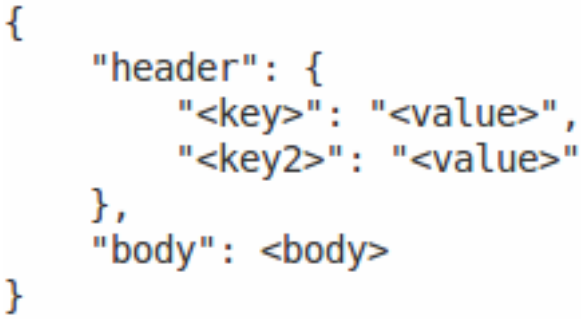
- Vá para: *Coleta de dados* → *Hosts*
- Clique em *Itens* na linha do host
- Clique em *Criar item*
- Insira os parâmetros do item no formulário


Item
Tags
Preprocessing
* Name
HTTP agent item
Type
HTTP agent
* Key
http_value_search
Select
Type of information
Numeric (unsigned)
* URL
http://localhost:9200/str/values/_search
Parse
Query fields
Name
Value
scroll
⇒
10s
Remove
Add
Request type
POST
Request body type
Raw data
JSON data
XML data
Request body
{
"query":{
"bool":{
"must":{
"match":{
"itemid":28275
}
Headers
Name
Value
name
⇒
value
Remove
Add
Required status codes
200
Follow redirects
Retrieve mode
Body
Headers
Body and headers
Convert to JSON
HTTP proxy
[protocol://][user[:password]@]proxy.example.com[:port]
HTTP authentication
None
SSL verify peer
SSL verify host
SSL certificate file
SSL key file
SSL key password
Host interface
127.0.0.1:10050
Units
* Update interval
1m
Custom intervals
Type
Interval
Period
Action
Flexible
Scheduling
50s
1-7,00:00-24:00
Remove
Add
* Timeout
Global
Override
3
Timeouts
* History
Do not store
Store up to
31d
* Trends
Do not store
Store up to
365d
Value mapping
type here to search
Select
Enable trapping
Populates host inventory field
-None-
Description
Enabled
Add
Test
Cancel

Todos os campos obrigatórios estão marcados com um asterisco vermelho.

Os campos que exigem informações específicas para itens HTTP são:

Parâmetro	Descrição
Tipo	Selecione **HTTP agent** aqui.
Chave	Insira uma chave de item exclusiva.
URL	URL para conectar e recuperar dados. Por exemplo: https://www.exemplo.com.br http://www.exemplo.com.br/download Nomes de domínio podem ser especificados em caracteres Unicode. Eles são automaticamente convertidos para ASCII (punycode) ao executar a verificação HTTP. O botão *Analisar* pode ser usado para separar campos de consulta opcionais (como ?name=Admin&password=mypassword) da URL, movendo os atributos e valores para *Campos de consulta* para codificação automática da URL. Limitado a 2048 caracteres. Macros suportadas: {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {ITEM.ID}, {ITEM.KEY}, {ITEM.KEY.ORIG}, macros de usuário, macros de descoberta de baixo nível. Isso define a opção cURL CURLOPT_URL.
Campos de consulta	Variáveis para a URL (veja acima). Especificadas como pares de atributo e valor. Os valores são codificados automaticamente na URL. Valores de macros são resolvidos e então codificados automaticamente na URL. Macros suportadas: {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {ITEM.ID}, {ITEM.KEY}, {ITEM.KEY.ORIG}, macros de usuário, macros de descoberta de baixo nível. Isso define a opção cURL CURLOPT_URL.
Tipo de requisição	Selecione o tipo de método de requisição: *GET*, *POST*, *PUT* ou *HEAD*
Tipo de corpo da requisição	Selecione o tipo de corpo da requisição: **Dados brutos** - corpo de requisição HTTP personalizado, macros são substituídas, mas nenhuma codificação é realizada **Dados JSON** - corpo de requisição HTTP em formato JSON. Macros podem ser usadas como string, número, true e false; macros usadas como strings devem estar entre aspas duplas. Valores de macros são resolvidos e então escapados automaticamente. Se "Content-Type" não for especificado nos cabeçalhos, será definido como "Content-Type: application/json" **Dados XML** - corpo de requisição HTTP em formato XML. Macros podem ser usadas como nó de texto, atributo ou seção CDATA. Valores de macros são resolvidos e então escapados automaticamente em um nó de texto e atributo. Se "Content-Type" não for especificado nos cabeçalhos, será definido como "Content-Type: application/xml" *Nota* que selecionar *Dados XML* requer libxml2.
Corpo da requisição	Insira o corpo da requisição. Macros suportadas: {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {ITEM.ID}, {ITEM.KEY}, {ITEM.KEY.ORIG}, macros de usuário, macros de descoberta de baixo nível.
Cabeçalhos	Cabeçalhos HTTP personalizados que serão enviados ao executar uma requisição. Especificados como pares de atributo e valor. Macros suportadas: {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {ITEM.ID}, {ITEM.KEY}, {ITEM.KEY.ORIG}, macros de usuário, macros de descoberta de baixo nível. Isso define a opção cURL CURLOPT_HTTPHEADER.
Códigos de status requeridos	Lista de códigos de status HTTP esperados. Se o Zabbix receber um código que não está na lista, o item se tornará não suportado. Se vazio, nenhuma verificação é realizada. Por exemplo: 200,201,210-299 Macros suportadas na lista: macros de usuário, macros de descoberta de baixo nível. Isso usa a opção cURL CURLINFO_RESPONSE_CODE.
Seguir redirecionamentos	Marque a caixa de seleção para seguir redirecionamentos HTTP. Isso define a opção cURL CURLOPT_FOLLOWLOCATION.
Modo de recuperação	Selecione a parte da resposta que deve ser recuperada: **Corpo** - apenas o corpo **Cabeçalhos** - apenas os cabeçalhos **Corpo e cabeçalhos** - corpo e cabeçalhos

Parâmetro	Descrição
Converter para JSON	Os cabeçalhos são salvos como pares de atributo e valor sob a chave "header" se o *Modo de recuperação* não estiver definido como *Corpo*. Se 'Content-Type: application/json' for encontrado, o corpo será salvo como um objeto, caso contrário, será armazenado como string, por exemplo: 
Proxy HTTP	Você pode especificar um proxy HTTP para usar, usando o formato `[protocol://][username[:password]@]proxy.exemplo.com.br[:port]`. O prefixo opcional `protocol://` pode ser usado para especificar protocolos de proxy alternativos (por exemplo, https, socks4, socks5; veja a documentação; o suporte ao prefixo de protocolo foi adicionado no cURL 7.21.7). Sem protocolo especificado, o proxy será tratado como um proxy HTTP. Se você especificar o protocolo errado, a conexão falhará e o item se tornará não suportado. Por padrão, a porta 1080 será usada. Se especificado, o proxy substituirá as variáveis de ambiente relacionadas a proxy, como http_proxy, HTTPS_PROXY. Se não especificado, o proxy não substituirá as variáveis de ambiente relacionadas a proxy. O valor inserido é passado "como está", sem verificação de sanidade. *Nota* que apenas autenticação simples é suportada com proxy HTTP. Macros suportadas: {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {ITEM.ID}, {ITEM.KEY}, {ITEM.KEY.ORIG}, macros de usuário, macros de descoberta de baixo nível. Isso define a opção cURL CURLOPT_PROXY.
Autenticação HTTP	Selecione a opção de autenticação: **Nenhuma** - nenhuma autenticação usada; **Básica** - autenticação básica é usada; **NTLM** - autenticação NTLM (Windows NT LAN Manager) é usada; **Kerberos** - autenticação Kerberos é usada (veja também: Configurando Kerberos com Zabbix); **Digest** - autenticação Digest é usada. Isso define a opção cURL CURLOPT_HTTPAUTH.
Nome de usuário	Insira o nome de usuário (até 255 caracteres). Este campo está disponível se *Autenticação HTTP* estiver definido como Básica, NTLM, Kerberos ou Digest. Macros de usuário e macros de descoberta de baixo nível são suportadas.
Senha	Insira a senha do usuário (até 255 caracteres). Este campo está disponível se *Autenticação HTTP* estiver definido como Básica, NTLM, Kerberos ou Digest. Macros de usuário e macros de descoberta de baixo nível são suportadas.
Verificar SSL peer	Marque a caixa de seleção para verificar o certificado SSL do servidor web. O certificado do servidor será automaticamente obtido do local da autoridade certificadora (CA) do sistema. Você pode substituir o local dos arquivos CA usando o parâmetro de configuração do Zabbix server ou proxy SSLCALocation. Isso define a opção cURL CURLOPT_SSL_VERIFYPEER.
Verificar host SSL	Marque a caixa de seleção para verificar se o campo Common Name ou o campo Subject Alternate Name do certificado do servidor web corresponde. Isso define a opção cURL CURLOPT_SSL_VERIFYHOST.
Arquivo de certificado SSL	Nome do arquivo de certificado SSL usado para autenticação do cliente. O arquivo de certificado deve estar no formato PEM[1]. Se o arquivo de certificado também contiver a chave privada, deixe o campo de arquivo de chave SSL vazio. Se a chave estiver criptografada, especifique a senha no campo de senha da chave SSL. O diretório que contém este arquivo é especificado pelo parâmetro de configuração do Zabbix server ou proxy SSLCertLocation. Macros suportadas: {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {ITEM.ID}, {ITEM.KEY}, {ITEM.KEY.ORIG}, macros de usuário, macros de descoberta de baixo nível. Isso define a opção cURL CURLOPT_SSLCERT.

Parâmetro	Descrição
Arquivo de chave SSL	Nome do arquivo de chave privada SSL usado para autenticação do cliente. O arquivo de chave privada deve estar no formato PEM[1]. O diretório que contém este arquivo é especificado pelo parâmetro de configuração do Zabbix server ou proxy SSLKeyLocation. Macros suportadas: {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {ITEM.ID}, {ITEM.KEY}, {ITEM.KEY.ORIG}, macros de usuário, macros de descoberta de baixo nível. Isso define a opção cURL CURLOPT_SSLKEY.
Senha da chave SSL	Senha do arquivo de chave privada SSL. Macros suportadas: macros de usuário, macros de descoberta de baixo nível. Isso define a opção cURL CURLOPT_KEYPASSWD.
Timeout	O Zabbix não gastará mais do que o tempo definido no processamento da URL (1-600 segundos). Na verdade, este parâmetro define o tempo máximo para fazer uma conexão com a URL e o tempo máximo para executar uma requisição HTTP. Portanto, o Zabbix não gastará mais do que 2 x *Timeout* segundos em uma verificação. Isso define a opção cURL CURLOPT_TIMEOUT. Para mais informações sobre o parâmetro *Timeout*, veja atributos gerais do item.
Habilitar trapping	Com esta caixa de seleção marcada, o item também funcionará como um item trapper e aceitará dados enviados ao Zabbix server ou proxy usando a utilidade Zabbix sender ou o protocolo do Zabbix sender, ou aceitará dados enviados ao Zabbix server usando o método de API `history.push`. Para mais informações sobre envio de dados, veja: Itens trapper.
Hosts permitidos	Visível apenas se a caixa de seleção *Habilitar trapping* estiver marcada. Lista de endereços IP separados por vírgula, opcionalmente em notação CIDR, ou nomes DNS. Se especificado, conexões recebidas serão aceitas apenas dos hosts listados aqui. Se o suporte a IPv6 estiver habilitado, então '127.0.0.1', '::127.0.0.1', '::ffff:127.0.0.1' são tratados igualmente e '::/0' permitirá qualquer endereço IPv4 ou IPv6. '0.0.0.0/0' pode ser usado para permitir qualquer endereço IPv4. Observe que "endereços IPv6 compatíveis com IPv4" (prefixo 0000::/96) são suportados, mas obsoletos pelo RFC4291. Exemplo: 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.exemplo.com.br, {HOST.HOST} Espaços e macros de usuário são permitidos neste campo. Macros de host: {HOST.HOST}, {HOST.NAME}, {HOST.IP}, {HOST.DNS}, {HOST.CONN} são permitidas neste campo.

Note:
Se o campo *Proxy HTTP* for deixado em branco, outra maneira de usar um proxy HTTP é definir variáveis de ambiente relacionadas a proxy.
Para HTTP - defina a variável de ambiente `http_proxy` para o usuário do Zabbix server. Por exemplo:
 `http_proxy=http://proxy_ip:proxy_port`.
Para HTTPS - defina a variável de ambiente `HTTPS_PROXY`. Por exemplo:
 `HTTPS_PROXY=http://proxy_ip:proxy_port`.
Mais detalhes estão disponíveis executando o comando de shell: *# man curl*.

Attention:
[1] O Zabbix suporta arquivos de certificado e chave privada apenas no formato PEM. Caso você tenha seu certificado e dados de chave privada em um arquivo no formato PKCS #12 (normalmente com extensão *.p12 ou *.pfx), você pode gerar o arquivo PEM a partir dele usando os seguintes comandos:

```
openssl pkcs12 -in ssl-cert.p12 -clcerts -nokeys -out ssl-cert.pem
openssl pkcs12 -in ssl-cert.p12 -nocerts -nodes  -out ssl-cert.key
```

Exemplos

Exemplo 1

Envie solicitações GET simples para recuperar dados de serviços como Elasticsearch:

- Crie um item GET com URL: `localhost:9200/?pretty`
- Observe a resposta:

```
{
  "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"
}
```

- Agora extraia o número da versão usando uma etapa de pré-processamento JSONPath: `$.version.number`

Exemplo 2

Envie solicitações POST simples para recuperar dados de serviços como Elasticsearch:

- Crie um item POST com a URL: `http://localhost:9200/str/values/_search?scroll=10s`
- Configure o seguinte corpo POST para obter a carga do processador (média de 1 min por núcleo)

```
{
    "query": {
        "bool": {
            "must": [{
                "match": {
                    "itemid": 28275
                }
            }],
            "filter": [{
                "range": {
                    "clock": {
                        "gt": 1517565836,
                        "lte": 1517566137
                    }
                }
            }]
        }
    }
}
```

- Recebido:

```
{
    "_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
                }
            }]
        }
    }
```

- Agora use uma etapa de pré-processamento JSONPath para obter o valor do item: `$.hits.hits[0]._source.value`

Exemplo 3

Verificando se a API do Zabbix está ativa, usando apiinfo.version.

- Configuração do item:

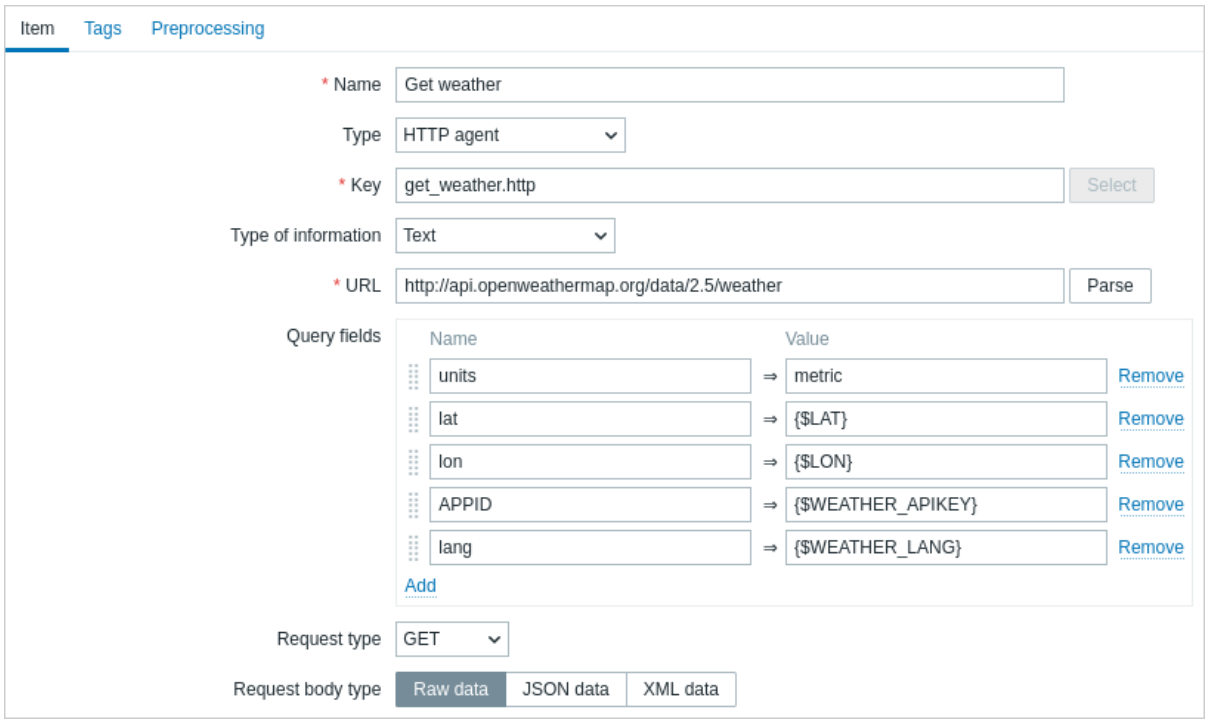
Observe o uso do método POST com dados JSON, definindo os cabeçalhos da solicitação e solicitando o retorno apenas dos cabeçalhos:

- Pré-processamento do valor do item com expressão regular para obter o código HTTP:
- Verificando o resultado em *Últimos dados*:

Exemplo 4

Recuperando informações meteorológicas conectando-se ao serviço público Openweathermap.

- Configure um item mestre para coleta de dados em massa em um único JSON:



Observe o uso de macros nos campos de consulta. Consulte a API do Openweathermap para saber como preenchê-los.

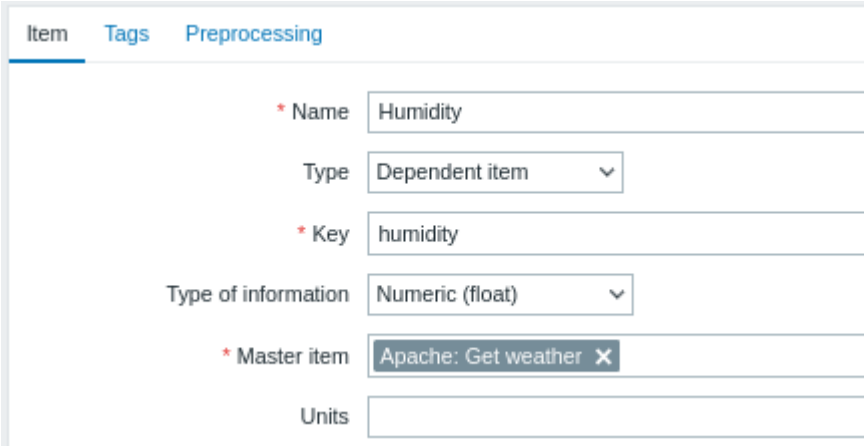
Exemplo de JSON retornado em resposta ao agente HTTP:

```
{
    "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
    }
}
```

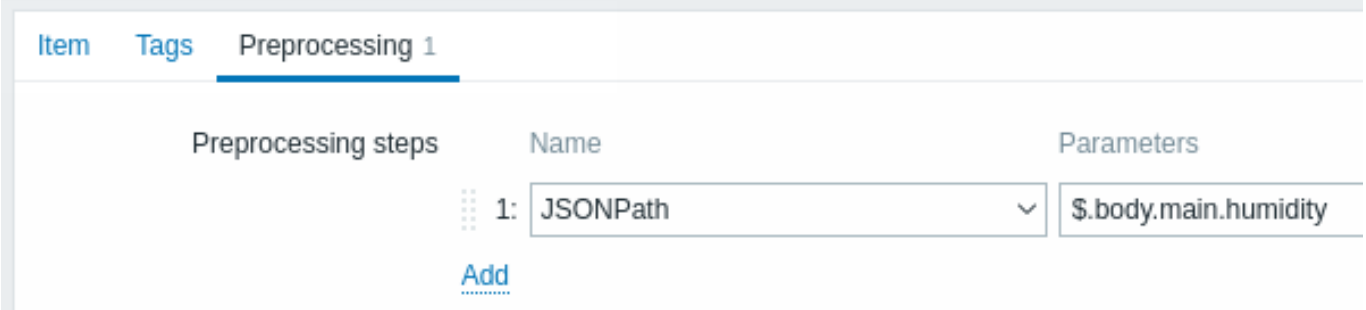

A próxima tarefa é configurar itens dependentes que extraem dados do JSON.

- Configure um item dependente de exemplo para umidade:

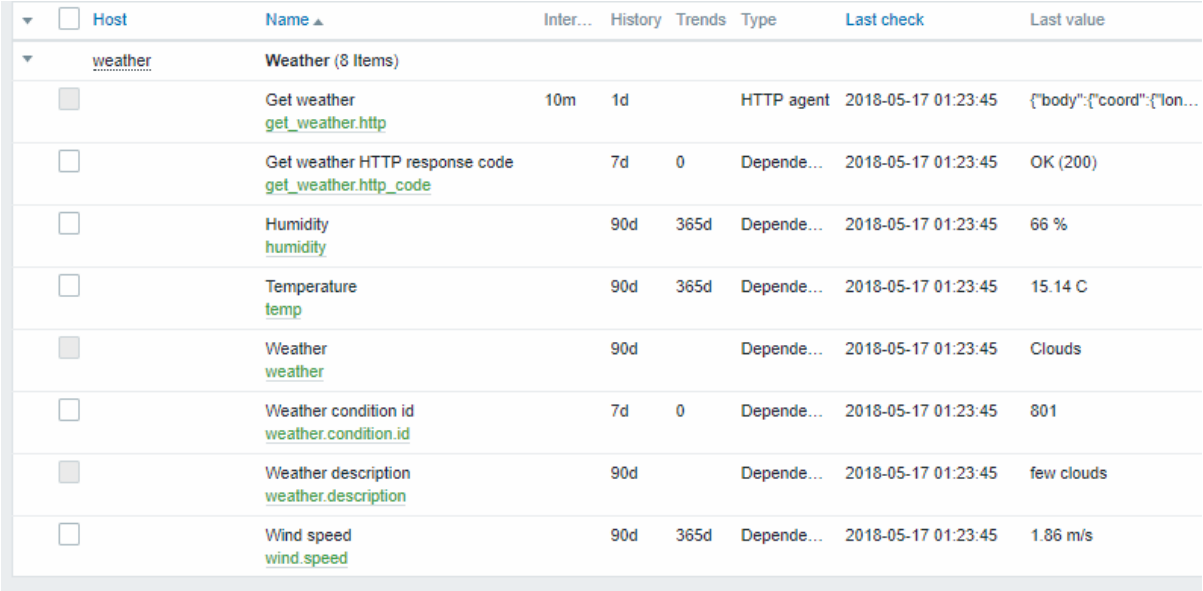


Outras métricas meteorológicas, como 'Temperatura', são adicionadas da mesma maneira.

- Pré-processamento de valor de item dependente de exemplo com JSONPath:



- Verifique o resultado dos dados meteorológicos em *Últimos dados*:

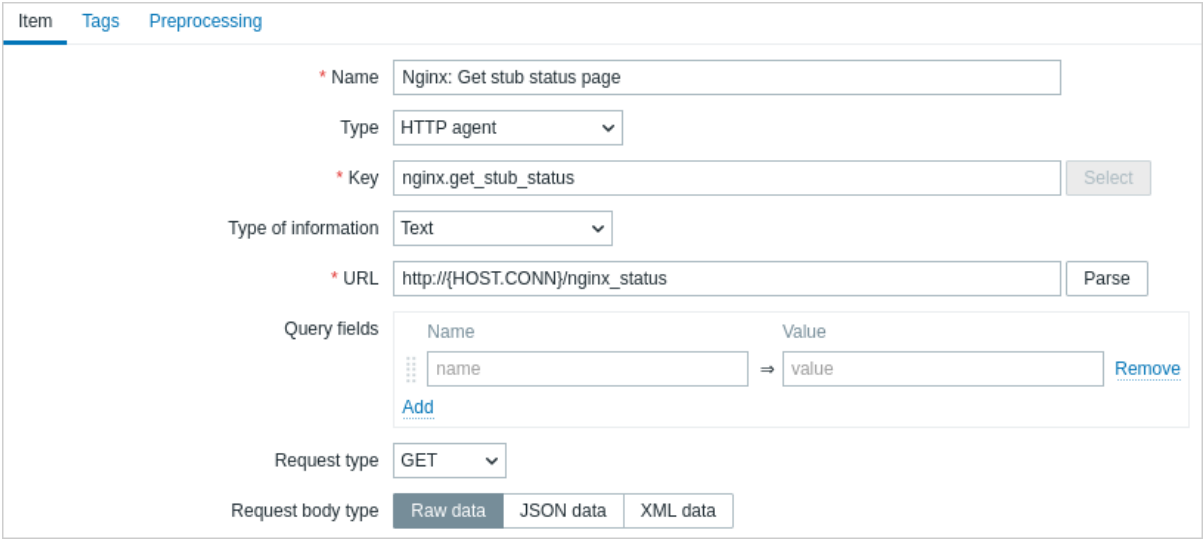


Host	Name ▲	Inter...	History	Trends	Type	Last check	Last value
weather	**Weather** (8 Items)						
	Get weather get_weather.http	10m	1d		HTTP agent	2018-05-17 01:23:45	{"body":{"coord":{"lon...
	Get weather HTTP response code get_weather.http_code		7d	0	Depende...	2018-05-17 01:23:45	OK (200)
	Humidity 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

Exemplo 5

Conectando-se à página de status do Nginx e obtendo suas métricas em lote.

- Configure o Nginx seguindo o guia oficial.
- Configure um item mestre para coleta de dados em lote:

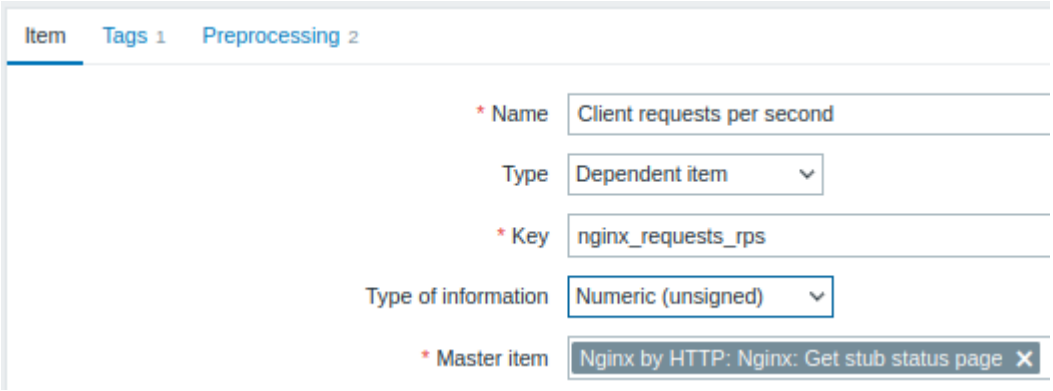


Exemplo de saída do status do Nginx:

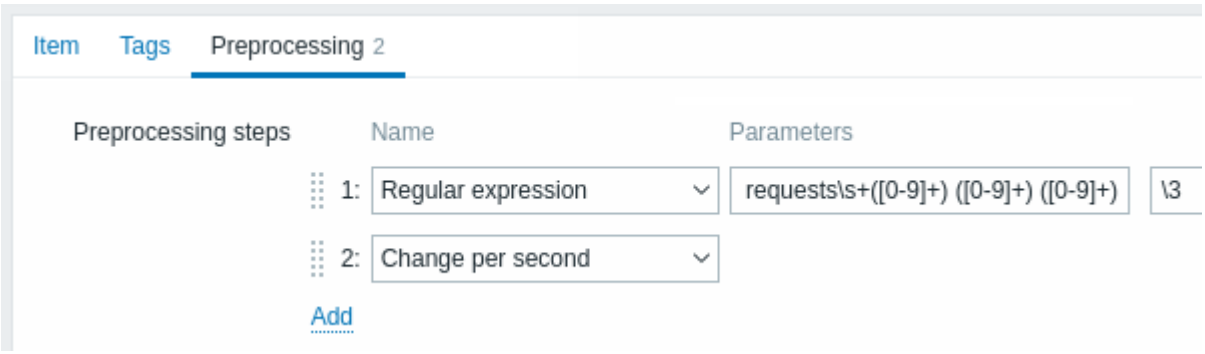
```
Active connections: 1 Active connections:
server accepts handled requests
 52 52 52
Reading: 0 Writing: 1 Waiting: 0
```

A próxima tarefa é configurar os itens dependentes que extraem os dados.

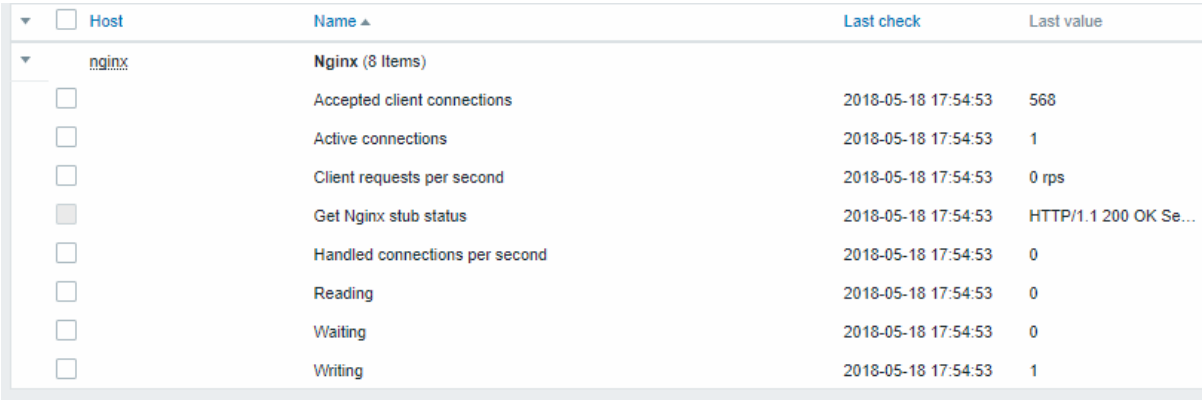
- Configure um item dependente de exemplo para requisições por segundo:



- Exemplo de pré-processamento de valor de item dependente com expressão regular `server accepts handled requests\s+([0-9]+) ([0-9]+) ([0-9]+)`:



- Verifique o resultado completo do módulo stub em *Últimos dados*:



17 Verificações do Prometheus

Visão geral

O Zabbix pode consultar métricas expostas no formato de linha do Prometheus.

São necessárias duas etapas para começar a coletar dados do Prometheus:

- um item mestre HTTP apontando para o endpoint de dados apropriado, por exemplo, `https://<prometheus host>/metrics`
- itens dependentes usando uma opção de pré-processamento Prometheus para consultar os dados necessários das métricas coletadas pelo item mestre

Existem duas opções de pré-processamento de dados do Prometheus:

- *Padrão Prometheus* - usado em itens normais para consultar dados do Prometheus
- *Prometheus para JSON* - usado em itens normais e para descoberta de baixo nível. Neste caso, os dados consultados do Prometheus são retornados em formato JSON.

Processamento em lote

O processamento em lote é suportado para items dependentes. Para habilitar o cache e a indexação, o pré-processamento *Prometheus pattern* deve ser a **primeira** etapa de pré-processamento. Quando *Prometheus pattern* é a primeira etapa de pré-processamento, os dados Prometheus analisados são armazenados em cache e indexados pela primeira condição

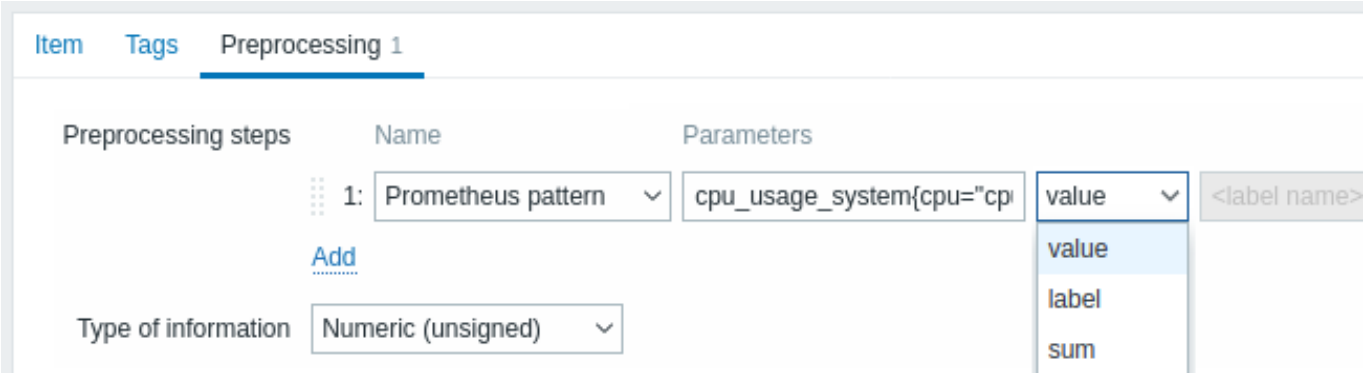
`<label>==<value>` na etapa de pré-processamento *Prometheus pattern*. Esse cache é reutilizado ao processar outros items dependentes neste lote. Para desempenho ideal, o primeiro rótulo deve ser aquele com mais valores diferentes.

Se houver outro pré-processamento a ser feito antes da primeira etapa, ele deve ser movido para o item mestre ou para um novo item dependente que seria usado como item mestre para os items dependentes.

Configuração

Desde que você tenha o item mestre HTTP configurado, é necessário criar um item dependente que utilize uma etapa de pré-processamento Prometheus:

- Insira os parâmetros gerais do item dependente no formulário de configuração
- Vá para a aba Pré-processamento
- Selecione uma opção de pré-processamento Prometheus (*Padrão Prometheus* ou *Prometheus para JSON*)



Os seguintes parâmetros são específicos para a opção de pré-processamento *Padrão Prometheus*:

Parâmetro	Descrição	Exemplos
Padrão	Para definir o padrão de dados necessário, você pode usar uma linguagem de consulta semelhante à linguagem de consulta do Prometheus (veja a tabela de comparação), por exemplo: <nome da métrica> - selecionar pelo nome da métrica {__name__="<nome da métrica>"} - selecionar pelo nome da métrica {__name__=~"<regex>"} - selecionar pelo nome da métrica que corresponda a uma expressão regular {<nome do rótulo>="<valor do rótulo>",...} - selecionar pelo nome do rótulo {<nome do rótulo>=~"<regex>",...} - selecionar pelo nome do rótulo que corresponda a uma expressão regular {__name__=~".*"}==<valor> - selecionar pelo valor da métrica Ou uma combinação dos anteriores: <nome da métrica>{<nome do rótulo1>="<valor do rótulo1>",<nome do rótulo2>=~"<regex>",...}==<valor> O valor do rótulo pode ser qualquer sequência de caracteres UTF-8, mas os caracteres barra invertida, aspas duplas e quebra de linha devem ser escapados como \\, \" e \n respectivamente; outros caracteres não devem ser escapados.	*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*

Parâmetro	Descrição	Exemplos
Processamento do resultado	Especifique se deseja retornar o valor, o rótulo ou aplicar a função apropriada (se o padrão corresponder a várias linhas e o resultado precisar ser agregado): **value** - retorna o valor da métrica (erro se várias linhas forem correspondidas) **label** - retorna o valor do rótulo especificado no campo *Rótulo* (erro se várias métricas forem correspondidas) **sum** - retorna a soma dos valores **min** - retorna o valor mínimo **max** - retorna o valor máximo **avg** - retorna o valor médio **count** - retorna a contagem de valores Este campo está disponível apenas para a opção *Padrão Prometheus*.	Veja também exemplos de uso dos parâmetros abaixo.
Saída	Defina o nome do rótulo (opcional). Neste caso, o valor correspondente ao nome do rótulo é retornado. Este campo está disponível apenas para a opção *Padrão Prometheus*, se 'Rótulo' for selecionado no campo *Processamento do resultado*.	

Exemplos de uso dos parâmetros

1. O caso de uso mais comum é retornar o **valor**. Para retornar o valor de `/var/db` de:

   `node_disk_usage_bytes{path="/var/cache"} 2.1766144e+09<br> node_disk_usage_bytes{path="/var/db"} 20480<br> node_disk_usage_bytes{path="/var/dpkg"} 8192<br> node_disk_usage_bytes{path="/var/empty"} 4096`

use os seguintes parâmetros:

- *Padrão* - `node_disk_usage_bytes{path="/var/db"}`
- *Processamento do resultado* - selecione 'value'

2. Você também pode se interessar pelo valor **médio** de todos os parâmetros `node_disk_usage_bytes`:

- *Padrão* - `node_disk_usage_bytes`
- *Processamento do resultado* - selecione 'avg'

3. Embora o Prometheus suporte apenas dados numéricos, é comum usar uma solução alternativa que permite retornar também a descrição textual relevante. Isso pode ser feito com um filtro e especificando o rótulo. Assim, para retornar o valor do rótulo 'color' de

   `elasticsearch_cluster_health_status{cluster="elasticsearch",color="green"} 1<br> elasticsearch_cluster 0<br> elasticsearch_cluster_health_status{cluster="elasticsearch",color="yellow"} 0`

use os seguintes parâmetros:

- *Padrão* - `elasticsearch_cluster_health_status {cluster="elasticsearch"} == 1`
- *Processamento do resultado* - selecione 'label'
- *Rótulo* - especifique 'color'

O filtro (baseado no valor numérico '1') irá corresponder à linha apropriada, enquanto o rótulo irá retornar a descrição do status de integridade (atualmente 'green'; mas potencialmente também 'red' ou 'yellow').

Prometheus para JSON

Os dados do Prometheus podem ser usados para descoberta de baixo nível. Nesse caso, os dados em formato JSON são necessários e a opção de pré-processamento *Prometheus para JSON* retornará exatamente isso.

Para mais detalhes, consulte Descoberta usando dados do Prometheus.

Comparação de linguagem de consulta

A tabela a seguir lista as diferenças e semelhanças entre o PromQL e a linguagem de consulta de pré-processamento do Prometheus no Zabbix.

PromQL instant vector selector		Pré-processamento do Prometheus no Zabbix
Diferenças		
Destino da consulta	Servidor Prometheus	Texto simples no formato de exposição do Prometheus
Retorno	Vetor instantâneo	Valor de métrica ou rótulo (padrão Prometheus) Array de métricas para valor único em JSON (Prometheus para JSON)
Operadores de correspondência de rótulo	**=**, **!=**, **=~**, **!~**	**=**, **!=**, **=~**, **!~**
Expressão regular usada na correspondência de nome de rótulo ou métrica	RE2	PCRE
Operadores de comparação	Veja lista	Apenas **==** (igual) é suportado para filtragem de valor
Semelhanças		
Selecionando por nome de métrica igual à string	<nome da métrica> ou {__name__=”<nome da métrica>”}	<nome da métrica> ou {__name__=”<nome da métrica>”}
Selecionando por nome de métrica que corresponde à expressão regular	{__name__=~”<regex>”}	{__name__=~”<regex>”}
Selecionando por valor de <nome do rótulo> igual à string	{<nome do rótulo>=”<valor do rótulo>”,...}	{<nome do rótulo>=”<valor do rótulo>”,...}
Selecionando por valor de <nome do rótulo> que corresponde à expressão regular	{<nome do rótulo>=~”<regex>”,...}	{<nome do rótulo>=~”<regex>”,...}
Selecionando por valor igual à string	{__name__=~”.*”} == <valor>	{__name__=~”.*”} == <valor>

18 Itens de script

Visão geral

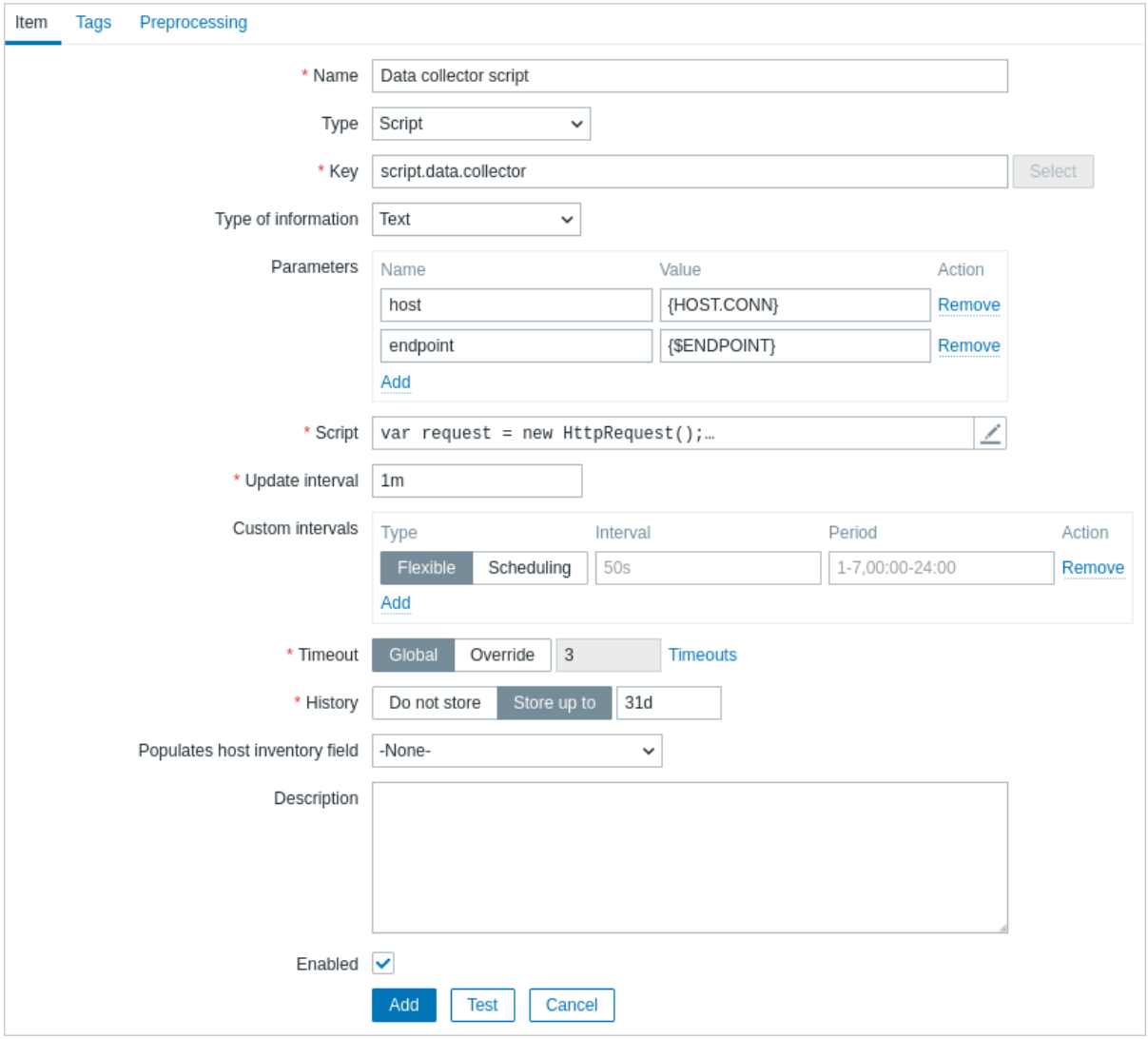
Itens de script podem ser usados para coletar dados executando um código JavaScript definido pelo usuário com a capacidade de recuperar dados via HTTP/HTTPS. Além do script, uma lista opcional de parâmetros (pares de nome e valor) e um tempo limite podem ser especificados.

Esse tipo de item pode ser útil em cenários de coleta de dados que exigem várias etapas ou lógica complexa. Como exemplo, um item de Script pode ser configurado para fazer uma chamada HTTP, processar os dados recebidos na primeira etapa de alguma forma e passar o valor transformado para a segunda chamada HTTP.

Itens de script são processados por pollers do Zabbix server ou proxy.

Configuração

No campo *Tipo* do formulário de configuração do item, selecione Script e preencha os campos obrigatórios.



Todos os campos obrigatórios estão marcados com um asterisco vermelho.

Os campos que exigem informações específicas para itens do tipo Script são:

Campo	Descrição
Chave	Insira uma chave única que será usada para identificar o item.
Parâmetros	Especifique as variáveis a serem passadas para o script como pares de atributo e valor. Macros de usuário são suportadas. Para ver quais macros internas são suportadas, faça uma busca por "Script-type item" na tabela de macros suportadas.
Script	Insira o código JavaScript no editor modal que é aberto ao clicar no campo de parâmetro ou no ícone de lápis ao lado dele. Este código deve fornecer a lógica para retornar o valor da métrica. O código tem acesso a todos os parâmetros e a todos os objetos JavaScript adicionais adicionados pelo Zabbix. Veja também: Guia JavaScript.
Timeout	Tempo limite de execução do JavaScript (1-600s; se excedido, retornará um erro). Observe que, dependendo do script, pode levar mais tempo para o timeout ser acionado. Para mais informações sobre o parâmetro *Timeout*, consulte atributos gerais do item.

Exemplos

Coleta de dados simples

Colete o conteúdo de *https://www.example.com/release_notes*:

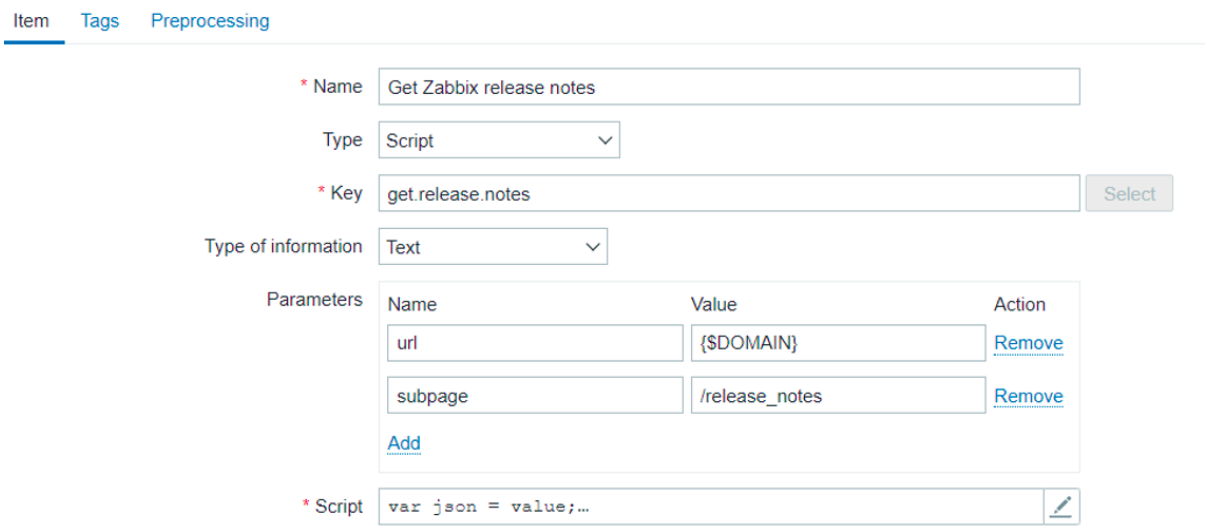
- Crie um item com o tipo "Script".
- No campo *Script*, insira:

```
var request = new HttpRequest();
return request.get("https://www.example.com/release_notes");
```

Coleta de dados com parâmetros

Colete o conteúdo de uma página específica e faça uso de parâmetros:

- Crie um item com o tipo ”Script” e dois parâmetros:
  - **url : {$DOMAIN}** (a macro de usuário {$DOMAIN} deve ser definida, preferencialmente no nível do host)
  - **subpage : /release_notes**



- No campo *Script*, insira:

```
var obj = JSON.parse(value);
var url = obj.url;
var subpage = obj.subpage;
var request = new HttpRequest();
return request.get(url + subpage);
```

Várias requisições HTTP

Coletar o conteúdo de *https://www.example.com* e *https://www.example.com/release_notes*:

- Crie um item com o tipo ”Script”.
- No campo *Script*, insira:

```
var request = new HttpRequest();
return request.get("https://www.example.com") + request.get("https://www.example.com/release_notes");
```

Registro em log

Adicione a entrada ”Log test” ao log do Zabbix server e receba o valor do item ”1” como retorno:

- Crie um item com o tipo ”Script”.
- No campo *Script*, insira:

```
Zabbix.log(3, 'Log test');
return 1;
```

19 Itens de navegador

Visão geral

Os itens do navegador permitem monitorar sites e aplicativos web complexos usando um navegador.

> **Attention:**
> O suporte a itens do navegador é atualmente experimental.

Os itens do navegador coletam dados executando um código JavaScript definido pelo usuário e recuperando dados via HTTP/HTTPS. Este item pode simular ações relacionadas ao navegador, como clicar, inserir texto, navegar por páginas da web e outras interações do usuário com sites ou aplicativos web.

Além do script, uma lista opcional de parâmetros (pares de nome e valor) e tempo limite pode ser especificada.

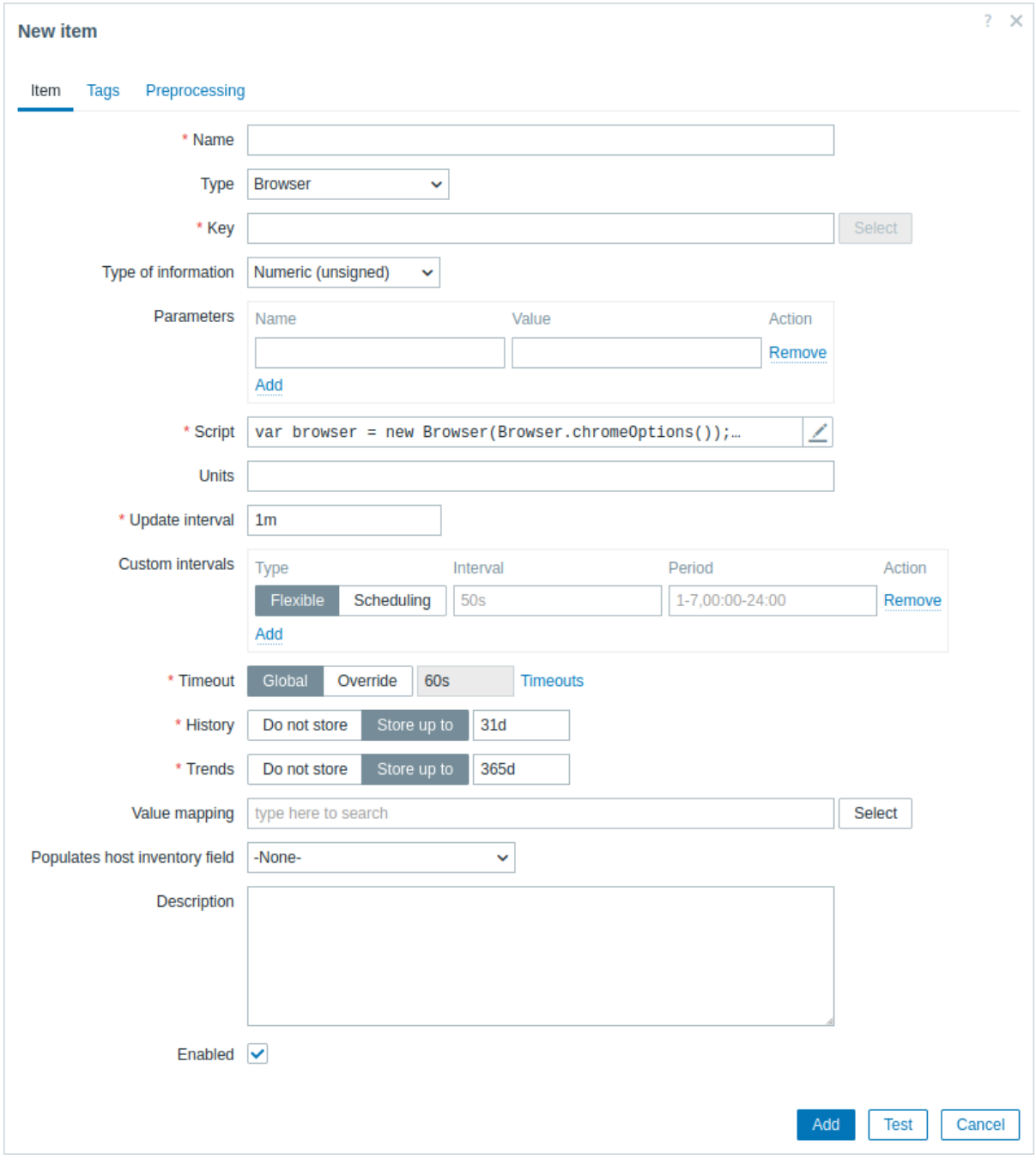
Attention:
O item implementa parcialmente o padrão W3C WebDriver com Selenium Server ou um WebDriver simples (por exemplo, ChromeDriver) como endpoint de teste web. Para que o item funcione, defina o endpoint no parâmetro `WebDriverURL` do arquivo de configuração do Zabbix server/proxy (se estiver usando o ChromeDriver, consulte Considerações de Segurança). Para melhor desempenho, considere usar um servidor dedicado para o ambiente de teste web.

As verificações de itens do navegador são executadas e processadas pelos pollers de navegador do Zabbix server ou proxy. Se necessário, você pode ajustar o número de instâncias pré-criadas de pollers de navegador no parâmetro `StartBrowserPollers` do arquivo de configuração do Zabbix server/proxy.

Para monitorar sites e aplicativos web complexos, o template Website by Browser está disponível como um template pronto para uso.

Configuração

No campo *Tipo* do formulário de configuração do item, selecione Navegador e preencha os campos obrigatórios.



Todos os campos obrigatórios estão marcados com um asterisco vermelho.

Os campos que exigem informações específicas para itens do tipo Navegador são:

Campo	Descrição
Chave	Insira uma chave exclusiva que será usada para identificar o item.
Parâmetros	Especifique as variáveis a serem passadas para o script como pares de atributo e valor. Macros de usuário são suportadas. Para ver quais macros internas são suportadas, faça uma busca por "item do tipo Navegador" na tabela de macros suportadas.
Script	Insira o código JavaScript no editor modal que é aberto ao clicar no campo de parâmetro ou no ícone de lápis ao lado dele. Este código deve fornecer a lógica para retornar o valor da métrica. O código tem acesso a todos os parâmetros, todos os objetos JavaScript adicionais e objetos JavaScript de item do tipo Navegador adicionados pelo Zabbix. Veja também: Guia JavaScript.
Timeout	Tempo limite de execução do JavaScript (1-600s; se excedido, retornará um erro). Observe que, dependendo do script, pode levar mais tempo para o timeout ser acionado. Para mais informações sobre o parâmetro *Timeout*, consulte atributos gerais do item.

Exemplos

Para um exemplo de como configurar o Zabbix para monitorar sites usando o template *Website by Browser*, consulte Monitorar sites com itens Browser.

Script padrão

O seguinte script:

1. Inicializa uma sessão do navegador.
2. Navega até uma URL especificada.
3. Coleta entradas de desempenho e estatísticas da sessão e as retorna como uma string JSON.

No campo *Script*, insira:

```
var browser = new Browser(Browser.chromeOptions());

try {
    browser.navigate("http://example.com");
    browser.collectPerfEntries();
}
finally {
    return JSON.stringify(browser.getResult());
}
```

Inicializar o navegador com capacidades personalizadas

O seguinte script:

1. Inicializa uma sessão do navegador para o navegador disponível com base no primeiro navegador correspondente na ordem especificada no script.
2. Define as capacidades do navegador, incluindo a estratégia de carregamento da página e opções específicas para cada navegador, como o modo headless para os navegadores Chrome, Firefox e Microsoft Edge.

No campo *Script*, insira:

```
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"
            }
        ]
    }
});
```

Inicializar navegador com GUI

Por padrão, as sessões do navegador (exceto Safari) são inicializadas no modo headless, o que significa que a interface gráfica do navegador (GUI) não é exibida.

O script a seguir inicializa uma sessão do navegador com a GUI ativada.

Observe que, se o WebDriver não conseguir localizar o binário do navegador, você pode especificar o caminho manualmente.

```
var opts = Browser.chromeOptions();
opts.capabilities.alwaysMatch['goog:chromeOptions'].args = [];

// Para inicializar uma sessão do Firefox com GUI, descomente as linhas a seguir:
// var opts = Browser.firefoxOptions();
// opts.capabilities.alwaysMatch['moz:firefoxOptions'].binary = 'usr/bin/firefox';
// opts.capabilities.alwaysMatch['moz:firefoxOptions'].args = [];

// Para inicializar uma sessão do Microsoft Edge com GUI, descomente as linhas a seguir:
// var opts = Browser.edgeOptions();
// opts.capabilities.alwaysMatch['ms:edgeOptions'].binary = 'usr/bin/microsoft-edge';
// opts.capabilities.alwaysMatch['ms:edgeOptions'].args = [];

var browser = new Browser(opts);
```

> **Note:**
> Se seus testes estiverem sendo executados em um servidor remoto ou em um contêiner, você pode usar um cliente Virtual Network Computing (VNC) para se conectar ao servidor VNC da máquina. Isso permite que você visualize e interaja com a GUI do navegador remotamente.

Tirar capturas de tela

O seguinte script:

1. Inicializa uma sessão do navegador.
2. Define o tamanho da janela do navegador para determinar o tamanho da captura de tela (especificado como parâmetros, veja abaixo).
3. Navega até uma URL (especificada como parâmetro, veja abaixo).
4. Coleta estatísticas da sessão, captura uma captura de tela e a adiciona às estatísticas coletadas.
5. Lida com erros capturando mensagens de erro e uma captura de tela.
6. Retorna os resultados coletados como uma string JSON.

O script também usa parâmetros do formulário de configuração do item:

- webURL - http://example.com
- width - 1920

- height - 1080

No campo *Script*, insira:

```
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);
}
```

Verificar login no Zabbix

O seguinte script:

1. Inicializa uma sessão do navegador.
2. Navega para uma página (especificada como um parâmetro, veja abaixo).
3. Insere o nome de usuário e a senha (especificados como parâmetros, veja abaixo).
4. Localiza e clica no botão de login.
5. Localiza e clica no botão de logout.
6. Coleta dados de desempenho antes e depois do login, bem como após o logout.
7. Lida com erros capturando mensagens de erro e uma captura de tela.
8. Retorna os resultados coletados como uma string JSON.

O script também utiliza parâmetros do formulário de configuração do item:

- webURL - http://{HOST.CONN}/index.php
- username - {$USERNAME}
- password - {$PASSWORD}

No campo *Script*, insira:

```
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);
}
```

Encontrar links

O seguinte script:

1. Inicializa uma sessão do navegador.
2. Define uma função para remover elementos duplicados de um array (veja o passo 5).
3. Navega até uma página (especificada como parâmetros, veja abaixo).
4. Encontra links na página.
5. Remove links duplicados para garantir que sejam únicos.
6. Extrai apenas os links que começam com "http".
7. Formata os links extraídos para uma estrutura específica.
8. Lida com erros capturando mensagens de erro e uma captura de tela.
9. Retorna os resultados coletados como uma string JSON.

O script também usa parâmetros do formulário de configuração do item:

- scheme - {$WEBSITE.SCHEME}
- domain - {$WEBSITE.DOMAIN}
- path - {$WEBSITE.PATH}

No campo *Script*, insira:

```
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);
}
```

1 Criando um item

Visão geral

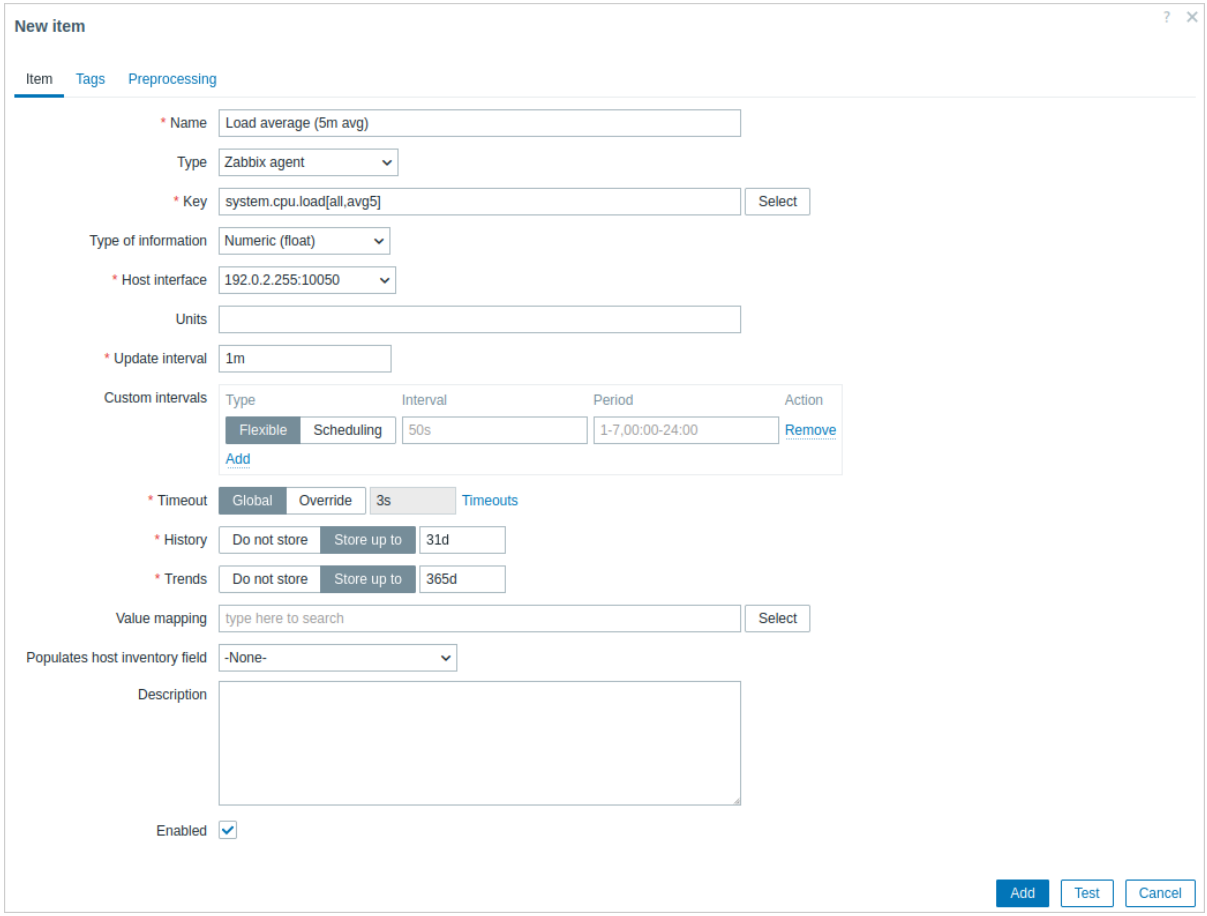
Para criar um item no frontend do Zabbix, faça o seguinte:

- Vá para: *Coleta de dados* > *Hosts*
- Clique em *Itens* na linha do host
- Clique em *Criar item* no canto superior direito da tela
- Insira os parâmetros do item no formulário

Você também pode criar um item abrindo um já existente, pressionando o botão *Clonar* e salvando com um nome diferente.

Configuração

A aba **Item** contém os atributos gerais do item.



Todos os campos obrigatórios estão marcados com um asterisco vermelho.

Parâmetro	Descrição
Nome	Nome do item. Macros de usuário são suportadas.
Tipo	Tipo do item. Veja as seções individuais de tipos de item.
Chave	Chave do item (até 2048 caracteres). As chaves de item suportadas podem ser encontradas nas seções de tipos de item. A chave deve ser única dentro de um único host. Se o tipo de chave for 'Zabbix agent', 'Zabbix agent (active)' ou 'Simple check', o valor da chave deve ser suportado pelo Zabbix agent ou Zabbix server. Veja também: o formato correto da chave.
Tipo de informação	Tipo de dado conforme armazenado no banco de dados após conversões, se houver. **Numérico (sem sinal)** - inteiro sem sinal de 64 bits. Observe que um valor de ponto flutuante, se recebido para um item inteiro, será truncado da parte decimal. **Numérico (float)** - número de ponto flutuante de 64 bits. Este tipo permitirá precisão de aproximadamente 15 dígitos e intervalo de aproximadamente -1,79E+308 a 1,79E+308 (com exceção do PostgreSQL 11 e versões anteriores). O recebimento de valores em notação científica também é suportado. Ex.: 1.23E+7, 1e308, 1.1E-4. **Caractere** - dados de texto curtos. **Log** - dados de texto longos com propriedades opcionais relacionadas a log (timestamp, source, severity, logeventid). **Texto** - dados de texto longos. Veja também limites de dados de texto. **Binário** - número binário (suportado apenas para itens dependentes). Itens que retornam valores com o tipo de dado binário não são suportados em fórmulas de item calculado ou expressões de trigger. Para chaves de item que retornam dados apenas em um formato específico, o tipo de informação correspondente é selecionado automaticamente.
Interface do host	Selecione a interface do host. Este campo está disponível ao editar um item no nível do host.

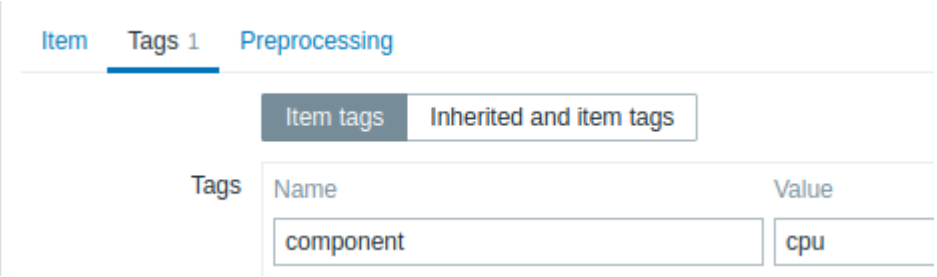
Parâmetro	Descrição
Unidades	Se um símbolo de unidade for definido, o Zabbix aplica pós-processamento ao valor recebido do item e o exibe com o sufixo de unidade especificado. Unidades suportadas com formatação especial (e exemplos de valor recebido → valor exibido): **B** - bytes (1024 → 1 KB) **Bps** - bytes por segundo (1024 → 1 KBps) **s** - segundos, exibidos usando até três maiores unidades de tempo não nulas (881764 → 10d 4h 56m) **uptime** - tempo decorrido em hh:mm:ss ou N dias, hh:mm:ss (881764 → 10 dias, 04:56:04) **unixtime** - timestamp Unix, formatado como yyyy.mm.dd hh:mm:ss (881764 → 1970-01-11 04:56:04 AM); para formatação correta, o valor recebido do item deve ser *Numérico (sem sinal)*. Para outras unidades (como Hz, W, etc.), se o valor recebido exceder 1000, ele será dividido por 1000 e exibido com um prefixo correspondente (5000 → 5 KHz, 881764 → 881.76 KW). Se o símbolo da unidade for prefixado com `!`, a conversão e o prefixo da unidade são desabilitados (1024 !B → 1024 B, 61 !s → 61 s). Para mais exemplos e detalhes sobre símbolos de unidade e conversão de unidade, veja Sufixos de valor de item.
Intervalo de atualização	Recupera um novo valor para este item a cada N segundos. O intervalo máximo permitido é de 86400 segundos (1 dia). Sufixos de tempo são suportados, ex.: 30s, 1m, 2h, 1d. Macros de usuário são suportadas. Uma única macro deve preencher todo o campo. Múltiplas macros em um campo ou macros misturadas com texto não são suportadas. *Nota*: O intervalo de atualização só pode ser definido como '0' se existirem intervalos personalizados com valor diferente de zero. Se definido como '0', e existir um intervalo personalizado (flexível ou agendado) com valor diferente de zero, o item será consultado durante a duração do intervalo personalizado. *Nota* que a primeira consulta do item após o item se tornar ativo ou após a alteração do intervalo de atualização pode ocorrer antes do valor configurado. Novos itens serão verificados em até 60 segundos após sua criação, a menos que tenham Agendamento ou Intervalo de atualização flexível e o *Intervalo de atualização* esteja definido como 0. Um item passivo existente pode ser consultado imediatamente pressionando o botão *Executar agora*.
Intervalos personalizados	Você pode criar regras personalizadas para checar o item: **Flexível** - cria uma exceção ao *Intervalo de atualização* (intervalo com frequência diferente). **Agendamento** - cria um agendamento de consulta personalizado. Para informações detalhadas, veja Intervalos personalizados. Sufixos de tempo são suportados no campo *Intervalo*, ex.: 30s, 1m, 2h, 1d. Macros de usuário são suportadas. Uma única macro deve preencher todo o campo. Múltiplas macros em um campo ou macros misturadas com texto não são suportadas.
Timeout	Define o timeout da checagem do item (disponível para tipos de item suportados). Selecione a opção de timeout: **Global** - o timeout do proxy/global é usado (exibido no campo *Timeout* desabilitado). **Substituir** - um timeout personalizado é usado (definido no campo *Timeout*; intervalo permitido: 1 - 600s). Sufixos de tempo, ex.: 30s, 1m, e macros de usuário são suportados. Clicar no link *Timeouts* permite configurar timeouts do proxy ou globais (se um proxy não for usado). Observe que o link *Timeouts* é visível apenas para usuários do tipo *Super admin* com permissões para as seções do frontend *Administração* > *Geral* ou *Administração* > *Proxies*.

Parâmetro	Descrição
Histórico	Selecione: **Não armazenar** - o histórico do item não é armazenado. Útil para itens mestre se apenas itens dependentes precisarem manter histórico. Esta configuração não pode ser sobrescrita pelas configurações globais do housekeeper. **Armazenar até** - especifique a duração de armazenamento do histórico detalhado no banco de dados (1 hora a 25 anos). Dados mais antigos serão removidos pelo housekeeper. Armazenado em segundos. Sufixos de tempo são suportados, ex.: 2h, 1d. Macros de usuário são suportadas. O valor *Armazenar até* pode ser sobrescrito globalmente em *Administração* > *Housekeeping*. Se existir uma configuração global de sobrescrita, um ícone laranja é exibido. Se você posicionar o mouse sobre ele, uma mensagem de aviso será exibida, ex.: *Sobrescrito pelas configurações globais do housekeeper (1d)*. Recomenda-se manter os valores registrados pelo menor tempo possível para reduzir o tamanho do histórico de valores no banco de dados. Em vez de armazenar um longo histórico de valores, você pode armazenar dados de tendências por mais tempo. Veja também Histórico e tendências.
Tendências	Selecione: **Não armazenar** - tendências não são armazenadas. Esta configuração não pode ser sobrescrita pelas configurações globais do housekeeper. **Armazenar até** - especifique a duração de armazenamento do histórico agregado (mínimo, máximo, média, contagem por hora) no banco de dados (1 dia a 25 anos). Dados mais antigos serão removidos pelo housekeeper. Armazenado em segundos. Sufixos de tempo são suportados, ex.: 24h, 1d. Macros de usuário são suportadas. O valor *Armazenar até* pode ser sobrescrito globalmente em *Administração* > *Housekeeping*. Se existir uma configuração global de sobrescrita, um ícone laranja é exibido. Se você posicionar o mouse sobre ele, uma mensagem de aviso será exibida, ex.: *Sobrescrito pelas configurações globais do housekeeper (7d)*. *Nota:* O armazenamento de tendências não está disponível para dados não numéricos - caractere, log e texto. Veja também Histórico e tendências.
Mapeamento de valor	Aplica mapeamento de valor a este item. O mapeamento de valor não altera os valores recebidos, serve apenas para exibição dos dados. Funciona com itens *Numérico (sem sinal)*, *Numérico (float)* e *Caractere*. Por exemplo, ”Estados de serviço do Windows”.
Formato de tempo do log	Disponível apenas para itens do tipo **Log**. Placeholders suportados: **y**: *Ano (1970-2038)*. **M**: *Mês (01-12)*. **d**: *Dia (01-31)*. **h**: *Hora (00-23)*. **m**: *Minuto (00-59)*. **s**: *Segundo (00-59)*. Se deixado em branco, o timestamp será definido como 0 em tempo Unix, representando 1º de janeiro de 1970. Por exemplo, considere a seguinte linha do arquivo de log do Zabbix agent: ” 23480:20100328:154718.045 Zabbix agent started. Zabbix 1.8.2 (revision 11211).” Ela começa com seis posições de caractere para o PID, seguidas por data, hora e o restante da mensagem. O formato de tempo do log para esta linha seria ”ppppp:yyyyMMdd:hhmmss”. Observe que os caracteres ”p” e ”:” são placeholders e podem ser quaisquer caracteres, exceto ”yMdhms”.
Preenche campo de inventário do host	Você pode selecionar um campo de inventário do host que o valor do item irá preencher. Isso funcionará se o preenchimento automático do inventário estiver habilitado para o host. Este campo não está disponível se o *Tipo de informação* estiver definido como ’Log’.
Descrição	Digite uma descrição do item. Macros de usuário são suportadas.
Habilitado	Marque a caixa para habilitar o item para que ele seja processado.
Últimos dados	Clique no link para ver os últimos dados do item. Este link só está disponível ao editar um item já existente.

Note:
Campos específicos de tipo de item são descritos nas páginas correspondentes.

Note:
Ao editar um item de template existente no nível do host, vários campos são somente leitura. Você pode usar o link no cabeçalho do formulário para ir ao nível do template e editá-los lá, lembrando que as alterações no nível do template irão alterar o item para todos os hosts aos quais o template está vinculado.

A aba **Tags** permite definir tags no nível do item.



Pré-processamento de valor do item

A aba **Pré-processamento** permite definir regras de transformação para os valores recebidos.

Teste de item

Attention:
Para realizar o teste de item, certifique-se de que o horário do sistema no server e no proxy esteja sincronizado. No caso de o horário do server estar atrasado, o teste de item pode retornar a mensagem de erro "The task has been expired." No entanto, definir fusos horários diferentes no server e no proxy não afetará o resultado do teste.

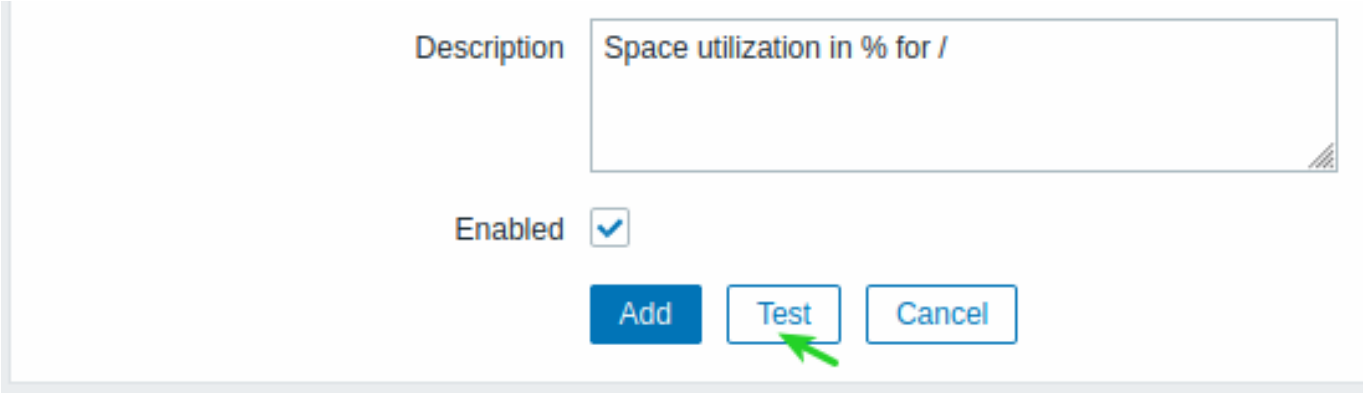
É possível testar um item e, se configurado corretamente, obter um valor real como retorno. O teste pode ocorrer mesmo antes de um item ser salvo.

O teste está disponível para itens de host e template, protótipos de item e regras de descoberta de baixo nível. O teste não está disponível para itens ativos.

O teste de item está disponível para os seguintes tipos de item passivo:

- Zabbix agent
- SNMP agent (v1, v2, v3)
- IPMI agent
- Verificações SSH
- Verificações Telnet
- JMX agent
- Verificações simples (exceto itens `icmpping*`, `vmware.*`)
- Zabbix internal
- Itens calculados
- Verificações externas
- Monitor de banco de dados
- HTTP agent
- Script
- Navegador

Para testar um item, clique no botão *Testar* na parte inferior do formulário de configuração do item. Observe que o botão *Testar* ficará desabilitado para itens que não podem ser testados (como verificações ativas, verificações simples excluídas).

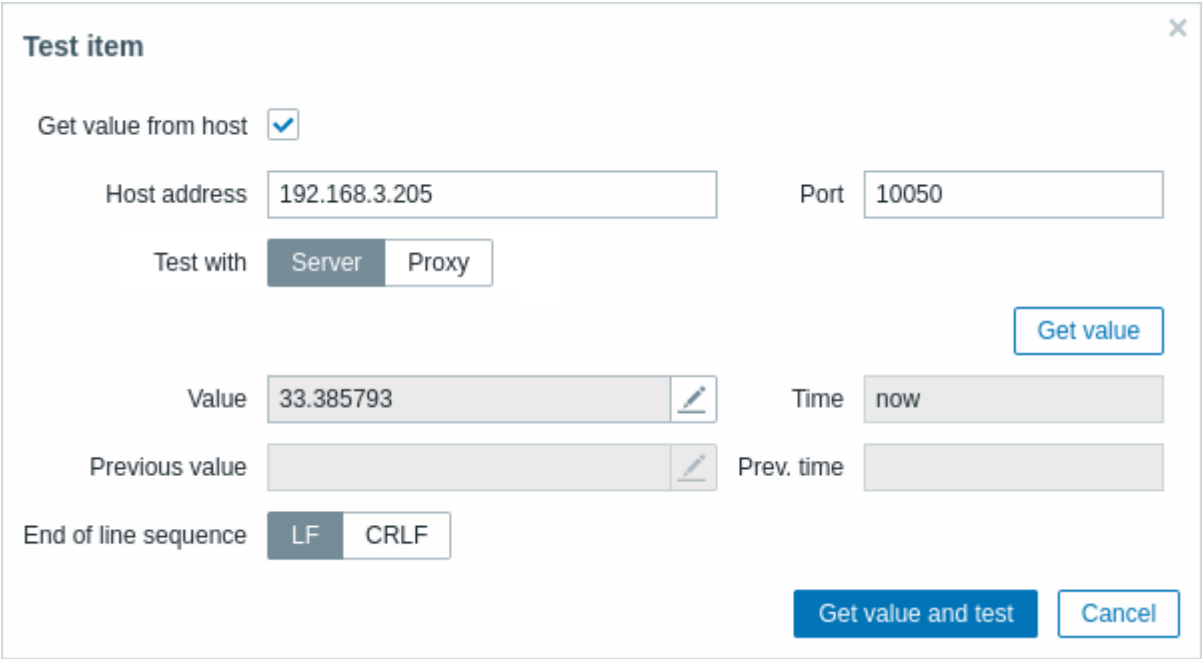


O formulário de teste de item possui campos para os parâmetros necessários do host (endereço do host, porta, testar com server/proxy (nome do proxy)) e detalhes específicos do item (como comunidade SNMPv2 ou credenciais de segurança SNMPv3). Esses campos são sensíveis ao contexto:

- Os valores são preenchidos automaticamente quando possível, ou seja, para itens que exigem um agent, obtendo as informações da interface de agent selecionada do host.
- Os valores devem ser preenchidos manualmente para itens de template.
- Os valores de macros em texto simples são resolvidos.
- Os campos em que o valor (ou parte do valor) é uma macro secreta ou macro do Vault ficam vazios e devem ser preenchidos manualmente. Se algum parâmetro do item contiver um valor de macro secreto, a seguinte mensagem de aviso será exibida: "Item contains user-defined macros with secret values. Values of these macros should be entered manually."
- Os campos são desabilitados quando não são necessários no contexto do tipo de item (por exemplo, o campo de endereço do host e o campo de proxy são desabilitados para itens calculados)

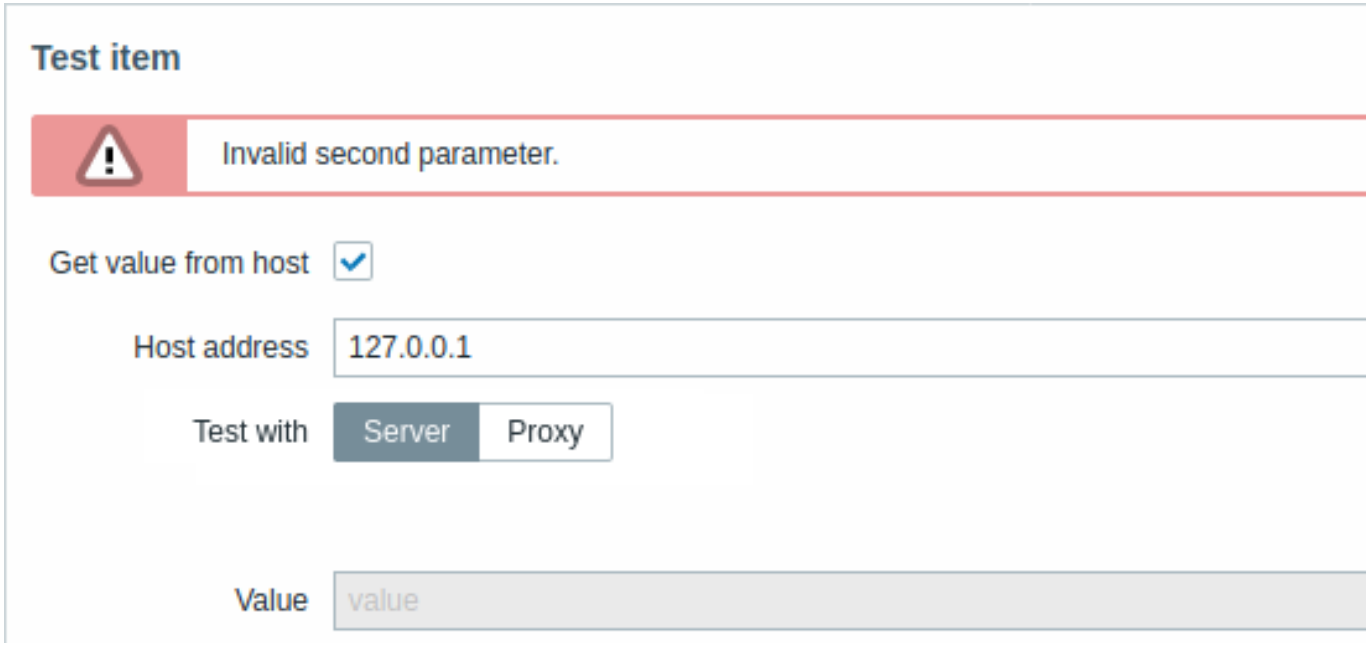
Para testar o item, clique em *Obter valor*. Se o valor for recuperado com sucesso, ele preencherá o campo *Valor*, movendo o valor atual (se houver) para o campo *Valor anterior* e também calculando o campo *Tempo ant.*, ou seja, a diferença de tempo entre os dois valores (cliques) e tentando detectar uma sequência EOL e alternar para CRLF se detectar "\n\r" no valor recuperado.

Clique em *Obter valor e testar* para testar o pré-processamento.



Os valores recuperados de um host e os resultados dos testes são truncados para um tamanho máximo de 512KB quando enviados para o frontend. Se um resultado for truncado, um ícone de aviso será exibido. A descrição do aviso é exibida ao passar o mouse. Observe que dados maiores que 512KB ainda são processados totalmente pelo Zabbix server.

Se a configuração estiver incorreta, uma mensagem de erro será exibida descrevendo a possível causa.



Um valor recuperado com sucesso do host também pode ser usado para testar as etapas de pré-processamento.

Botões do formulário

Os botões na parte inferior do formulário permitem realizar várias operações.

Add	Adicionar um item. Este botão está disponível apenas para novos items.
Update	Atualizar as propriedades de um item.
Clone	Criar outro item com base nas propriedades do item atual.
Execute now	Executar uma verificação para um novo valor de item imediatamente. Suportado apenas para verificações **passivas** (veja mais detalhes). *Observe* que ao verificar um valor imediatamente, o cache de configuração não é atualizado, portanto, o valor não refletirá alterações muito recentes na configuração do item.
Test	Testar se a configuração do item está correta obtendo um valor.
Clear history and trends	Excluir o histórico e as tendências do item.
Delete	Excluir o item.
Cancel	Cancelar a edição das propriedades do item.

Limites de dados de texto

Os limites de dados de texto dependem do banco de dados utilizado. Antes de armazenar valores de texto no banco de dados, eles são truncados para corresponder ao limite do tipo de valor do banco de dados:

Banco de dados	Tipo de informação		
	Character	**Log**	**Text**
MySQL	255 caracteres	65536 bytes	65536 bytes
PostgreSQL	255 caracteres	65536 caracteres	65536 caracteres
Oracle	255 caracteres	65536 caracteres	65536 caracteres
SQLite (apenas Zabbix proxy)	255 caracteres	65536 caracteres	65536 caracteres

Limite de script personalizado

O comprimento disponível do script personalizado depende do banco de dados utilizado:

Banco de dados	Limite em caracteres	Limite em bytes
MySQL	65535	65535
PostgreSQL	65535	sem limite
Oracle	2048	4000
SQLite (apenas Zabbix proxy)	65535	sem limite

Timeout do item

O timeout do item especifica por quanto tempo o Zabbix deve esperar antes de abortar a verificação como falha.

Quando o timeout é atingido, a verificação é abortada mesmo que a recuperação dos dados não tenha terminado. Se os dados forem recebidos parcialmente, o item se torna não suportado e uma mensagem de erro é registrada (por exemplo, se os dados forem coletados com sucesso apenas para um dos vários OIDs em uma verificação SNMP).

Para muitos tipos de item, você pode definir timeouts de item **flexíveis**:

- por item individual
- por tipo de item (no nível do proxy)
- por tipo de item (no nível global)

Um timeout personalizado *por item individual* é útil se você quiser especificar um timeout maior para um item específico, mantendo os timeouts para outros itens baixos.

Precedência do timeout

1. O timeout de item individual substitui qualquer outro timeout.
2. Os timeouts no nível do proxy substituem os timeouts globais.

O timeout da configuração do Zabbix server ou do proxy **não** tem papel nas verificações para as quais os timeouts flexíveis estão definidos.

Suporte a timeout flexível

Timeouts flexíveis de item são suportados para estes tipos de item:

- Zabbix agent (tanto checagens passivas quanto ativas)
- SNMP agent (exceto checagens SNMP legadas[1])
- Checagem simples (exceto itens `icmpping*` e VMware[2])
- SSH agent
- Telnet agent
- Checagem externa
- Monitor de banco de dados
- Script
- HTTP agent
- Browser

[1] Para checagens SNMP legadas (consulta de OID única), as configurações de timeout do server ou do proxy se aplicam.
 [2] Para itens `icmpping*`, o valor de timeout é especificado diretamente na chave do item. Itens de monitoramento VMware usam seu próprio parâmetro VMwareTimeout da configuração do server ou do proxy.

Items não suportados

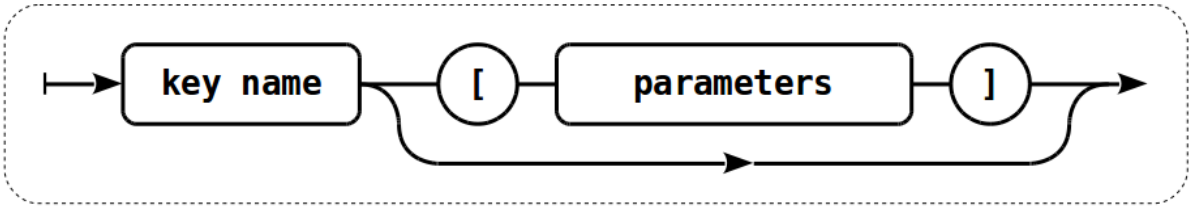
Itens não suportados são relatados como tendo o status *Não suportado* e ainda são verificados novamente no seu *Intervalo de atualização* padrão.

Um item se torna não suportado se seu valor não puder ser recuperado por algum motivo (por exemplo, erro de conexão, nenhum poller configurado para processar o item). Além disso, itens que não recebem dados históricos permanecem no status *Não suportado*. Este é o comportamento esperado—um item muda para o status *Normal* somente quando recebe novos dados históricos válidos.

Assim como os problemas, os itens não suportados são reavaliados apenas quando novos dados são recebidos—mesmo que não haja mais dados históricos disponíveis para esse item. Em outras palavras, itens e triggers mudam seu estado apenas ao receber novos dados. No entanto, se uma expressão de trigger incluir uma função de data e hora e/ou nodata(), o trigger será recalculado a cada 30 segundos (veja tempo de cálculo dos triggers para detalhes).

1 Formato da chave do item

O formato da chave do item, incluindo os parâmetros da chave, deve seguir as regras de sintaxe. As ilustrações a seguir mostram a sintaxe suportada. Os elementos e caracteres permitidos em cada ponto podem ser determinados seguindo as setas - se algum bloco puder ser alcançado pela linha, ele é permitido, caso contrário, não é permitido.



Para construir uma chave de item válida, começa-se especificando o nome da chave, depois há a escolha de ter parâmetros ou não - conforme mostrado pelas duas linhas que podem ser seguidas.

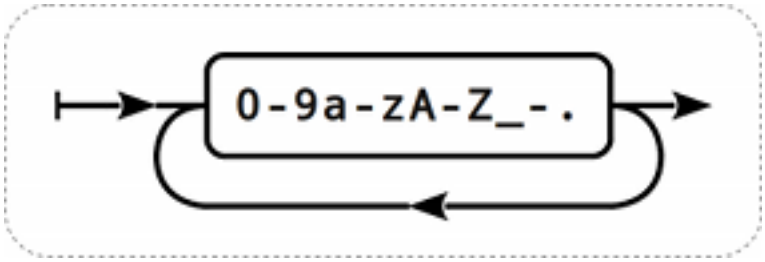
Nome da chave

O nome da chave em si tem um intervalo limitado de caracteres permitidos, que simplesmente seguem uns aos outros. Os caracteres permitidos são:

`0-9a-zA-Z_-.`

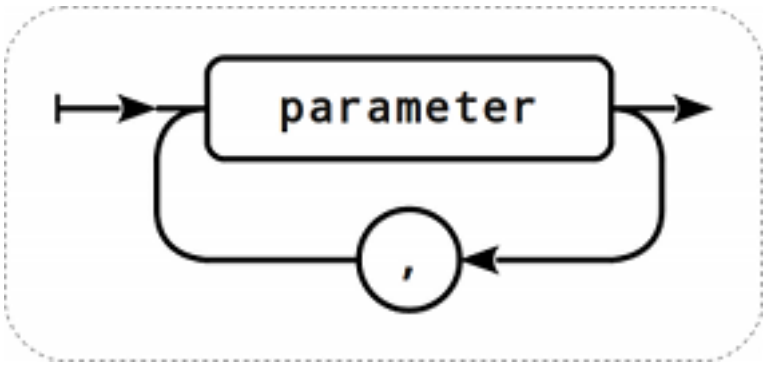
O que significa:

- todos os números;
- todas as letras minúsculas;
- todas as letras maiúsculas;
- sublinhado;
- traço;
- ponto.

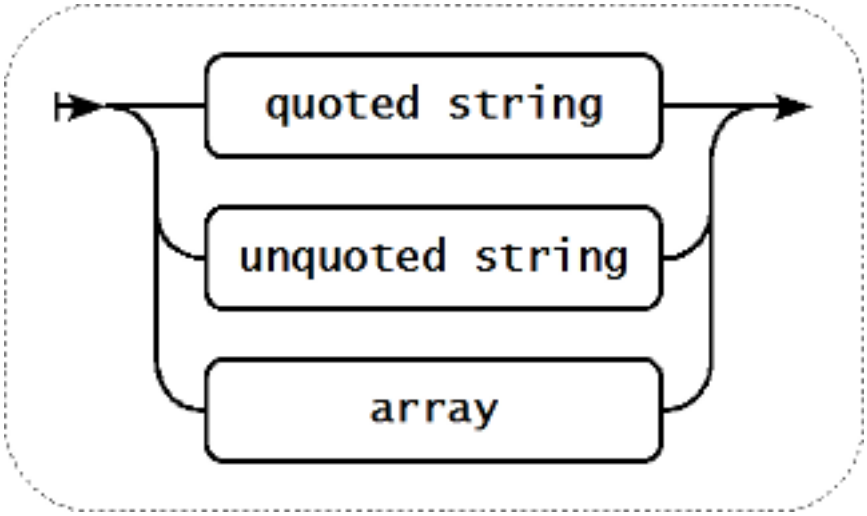


Parâmetros de chave

Uma chave de item pode ter vários parâmetros separados por vírgulas.



Cada parâmetro de chave pode ser uma string entre aspas, uma string sem aspas ou um array.

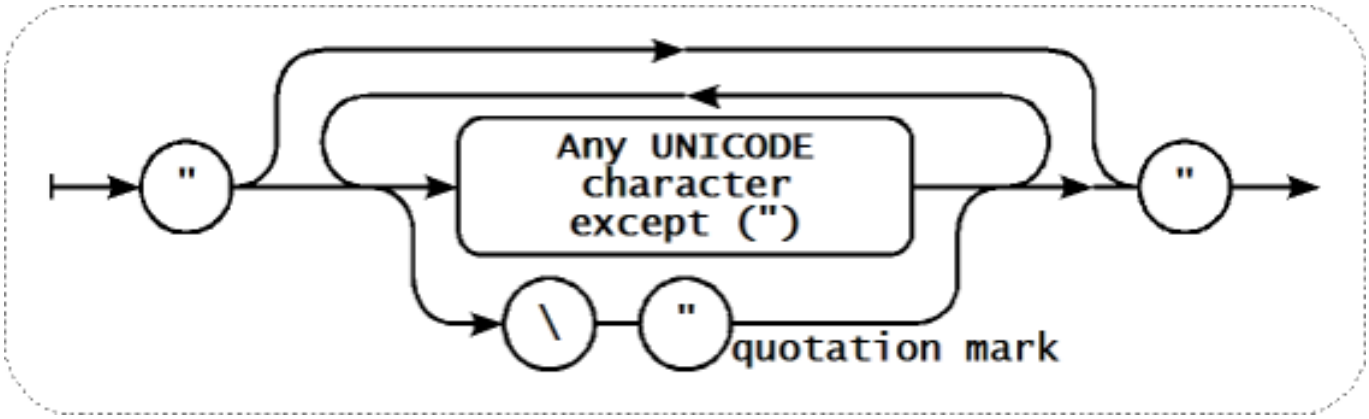


O parâmetro também pode ser deixado em branco, usando assim o valor padrão. Nesse caso, o número apropriado de vírgulas deve ser adicionado se quaisquer outros parâmetros forem especificados. Por exemplo, a chave de item **icmpping[„200„500]** especificaria que o intervalo entre pings individuais é de 200 milissegundos, o tempo limite - 500 milissegundos, e todos os outros parâmetros permanecem com seus valores padrão.

É possível incluir macros nos parâmetros. Elas podem ser macros de usuário ou algumas das macros internas. Para ver quais macros internas específicas são suportadas nos parâmetros de chave de item, procure na página Macros suportadas por ”item key parameters”.

Parâmetro - string entre aspas

Se o parâmetro da chave for uma string entre aspas, qualquer caractere Unicode é permitido. Se a string do parâmetro da chave contiver uma aspa, esse parâmetro deverá ser colocado entre aspas e cada aspa que fizer parte da string do parâmetro deverá ser escapada com um caractere de barra invertida (\). Se a string do parâmetro da chave contiver vírgula, esse parâmetro deverá ser colocado entre aspas.

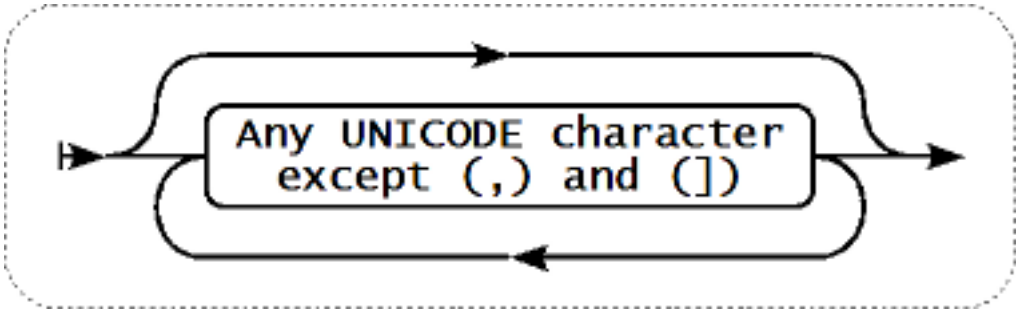


Warning:
Para colocar parâmetros de chave de item entre aspas, use apenas aspas duplas. Aspas simples não são suportadas.

Attention:
Arrays de parâmetros multinível, por exemplo, `[a,[b,[c,d]],e]`, não são permitidos.

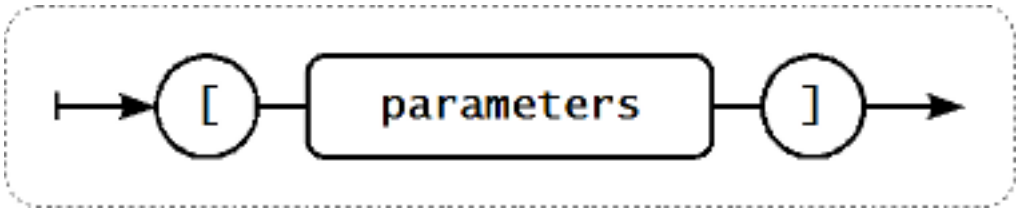
Parâmetro - string sem aspas

Se o parâmetro da chave for uma string sem aspas, qualquer caractere Unicode é permitido, exceto vírgula e colchete direito (]). O parâmetro sem aspas não pode começar com colchete esquerdo ([).



Parâmetro - array

Se o parâmetro da chave for um array, ele será novamente colocado entre colchetes, onde os parâmetros individuais seguem as regras e sintaxe de especificação de múltiplos parâmetros.



2 Intervalos personalizados

Visão geral

É possível criar regras personalizadas em relação aos horários em que um item é verificado. Os dois métodos para isso são *Intervalos flexíveis*, que permitem redefinir o intervalo de atualização padrão, e *Agendamento*, pelo qual uma verificação de item pode ser executada em um horário específico ou sequência de horários.

Intervalos flexíveis

Intervalos flexíveis permitem redefinir o intervalo de atualização padrão para períodos de tempo específicos. Um intervalo flexível é definido com *Interval* e *Period*, onde:

- *Interval* – o intervalo de atualização para o período de tempo especificado. Sufixos de tempo são suportados, por exemplo, 30s, 1m, 2h, 1d.
- *Period* – o período de tempo em que o intervalo flexível está ativo (veja períodos de tempo para uma descrição detalhada do formato de *Period*)

Se vários intervalos flexíveis se sobrepuserem, o menor valor de *Interval* será usado para o período de sobreposição. Observe que, se o menor valor dos intervalos flexíveis sobrepostos for '0', nenhuma coleta será realizada. Fora dos intervalos flexíveis, o intervalo de atualização padrão é usado.

Observe que, se o intervalo flexível for igual ao comprimento do período, o item será verificado exatamente uma vez. Se o intervalo flexível for maior que o período, o item pode ser verificado uma vez ou pode não ser verificado (portanto, tal configuração não é recomendada). Se o intervalo flexível for menor que o período, o item será verificado pelo menos uma vez.

Se o intervalo flexível for definido como '0', o item não será coletado durante o período do intervalo flexível e retomará a coleta de acordo com o *Update interval* padrão assim que o período terminar. Exemplos:

Interval	Period	Descrição
10	1-5,09:00-18:00	O item será verificado a cada 10 segundos durante o horário comercial.
0	1-7,00:00-7:00	O item não será verificado durante a noite.
0	7-7,00:00-24:00	O item não será verificado aos domingos.
60	1-7,12:00-12:01	O item será verificado às 12:00 todos os dias. Observe que isso era usado como uma solução alternativa para verificações agendadas e recomenda-se usar intervalos de agendamento para tais verificações.

Intervalos de agendamento

Os intervalos de agendamento são usados para verificar items em horários específicos. Enquanto os intervalos flexíveis são projetados para redefinir o intervalo padrão de atualização do item, os intervalos de agendamento são usados para especificar um agendamento de verificação independente, que é executado em paralelo.

Um intervalo de agendamento é definido como: `md<filter>wd<filter>h<filter>m<filter>s<filter>` onde:

- **md** - dias do mês
- **wd** - dias da semana
- **h** - horas
- **m** - minutos
- **s** – segundos

<filter> é usado para especificar valores para seu prefixo (dias, horas, minutos, segundos) e é definido como: [<from>[-<to>]][/<step>
onde:

- <from> e <to> definem o intervalo de valores correspondentes (incluídos). Se <to> for omitido, o filtro corresponderá a um intervalo <from> - <from>. Se <from> também for omitido, o filtro corresponderá a todos os valores possíveis.
- <step> define os saltos do valor numérico ao longo do intervalo. Por padrão, <step> tem o valor 1, o que significa que todos os valores do intervalo definido são correspondidos.

Embora as definições de filtro sejam opcionais, pelo menos um filtro deve ser usado. Um filtro deve ter um intervalo ou o valor <*step*> definido.

Um filtro vazio corresponde a '0' se nenhum filtro de nível inferior for definido ou a todos os valores possíveis caso contrário. Por exemplo, se o filtro de hora for omitido, apenas a hora '0' será correspondida, desde que os filtros de minuto e segundo também sejam omitidos, caso contrário, um filtro de hora vazio corresponderá a todos os valores de hora.

Valores válidos de <from> e <to> para seus respectivos prefixos de filtro são:

Prefixo	Descrição	<*from*>	<*to*>
md	Dias do mês	1-31	1-31
wd	Dias da semana	1-7	1-7
h	Horas	0-23	0-23
m	Minutos	0-59	0-59
s	Segundos	0-59	0-59

O valor <from> deve ser menor ou igual ao valor <to>. O valor <step> deve ser maior ou igual a 1 e menor ou igual a <to> - <from>.

Valores de dias do mês, horas, minutos e segundos de um único dígito podem ser prefixados com 0. Por exemplo, md01-31 e h/02 são intervalos válidos, mas md01-031 e wd01-07 não são.

No frontend do Zabbix, vários intervalos de agendamento são inseridos em linhas separadas. Na API do Zabbix, eles são concatenados em uma única string com um ponto e vírgula ; como separador.

Se um horário for correspondido por vários intervalos, ele será executado apenas uma vez. Por exemplo, wd1h9;h9 será executado apenas uma vez na segunda-feira às 9h.

Exemplos:

Intervalo	Será executado
m0-59	a cada minuto
h9-17/2	a cada 2 horas começando às 9:00 (9:00, 11:00 ...)
m0,30 ou m/30	de hora em hora às hh:00 e hh:30
m0,5,10,15,20,25,30,35,40,45,50,55 ou m/5	a cada cinco minutos
wd1-5h9	toda segunda a sexta-feira às 9:00
wd1-5h9-18	toda segunda a sexta-feira às 9:00,10:00,...,18:00
h9,10,11 ou h9-11	todos os dias às 9:00, 10:00 e 11:00
md1h9m30	todo dia 1º de cada mês às 9:30
md1wd1h9m30	todo dia 1º de cada mês às 9:30 se for segunda-feira
h9m/30	todos os dias às 9:00, 9:30
h9m0-59/30	todos os dias às 9:00, 9:30
h9,10m/30	todos os dias às 9:00, 9:30, 10:00, 10:30
h9-10m30	todos os dias às 9:30, 10:30
h9m10-40/30	todos os dias às 9:10, 9:40
h9,10m10-40/30	todos os dias às 9:10, 9:40, 10:10, 10:40
h9-10m10-40/30	todos os dias às 9:10, 9:40, 10:10, 10:40
h9m10-40	todos os dias às 9:10, 9:11, 9:12, ... 9:40
h9m10-40/1	todos os dias às 9:10, 9:11, 9:12, ... 9:40
h9-12,15	todos os dias às 9:00, 10:00, 11:00, 12:00, 15:00

Intervalo	Será executado
h9-12,15m0	todos os dias às 9:00, 10:00, 11:00, 12:00, 15:00
h9-12,15m0s30	todos os dias às 9:00:30, 10:00:30, 11:00:30, 12:00:30, 15:00:30
h9-12s30	todos os dias às 9:00:30, 9:01:30, 9:02:30 ... 12:58:30, 12:59:30
h9m/30;h10 (*sintaxe específica da API*)	todos os dias às 9:00, 9:30, 10:00
h9m/30 h10 (*adicione isso como outra linha no frontend*)	todos os dias às 9:00, 9:30, 10:00

Alinhando fusos horários para proxies e agent

Observe que proxies e agent do Zabbix usam seus fusos horários locais ao processar intervalos de agendamento.

Por esse motivo, quando intervalos de agendamento são aplicados a items monitorados por proxy Zabbix ou items ativos de agent, é recomendável definir o fuso horário dos respectivos proxies ou agent igual ao do server Zabbix, caso contrário a fila pode relatar atrasos de items incorretamente.

O fuso horário para proxy Zabbix ou agent pode ser definido usando a variável de ambiente TZ no arquivo de unidade `systemd`:

```
[Service]
...
Environment="TZ=Europe/Amsterdam"
```

2 Pré-processamento de valor de item

Visão geral

O pré-processamento permite aplicar transformações aos valores dos items recebidos antes de salvá-los no banco de dados. Essas transformações/etapas de pré-processamento são realizadas pelo Zabbix server ou proxy (se os items forem monitorados por proxy).

Esse recurso suporta uma variedade de casos de uso, como:

- converter bytes em bits (por exemplo, multiplicando valores de tráfego de rede por ”8”);
- calcular estatísticas por segundo para valores que aumentam incrementalmente;
- aplicar expressões regulares para extrair ou modificar valores;
- executar scripts personalizados nos valores;
- descartar valores inalterados para otimizar o armazenamento no banco de dados.

Uma ou mais etapas de pré-processamento podem ser configuradas para um item. Essas etapas são executadas na ordem em que são configuradas.

> **Attention:**
> Se uma etapa de pré-processamento falhar, um item se tornará não suportado. Isso pode ser evitado pelo tratamento de erro *Personalizado em caso de falha* (disponível para a maioria das transformações), permitindo descartar valores ou definir valores personalizados.

 Para items de log, os metadados do log (sem valor) sempre redefinirão o estado de item não suportado, tornando-o suportado novamente. Isso acontece mesmo que o erro inicial tenha ocorrido após o recebimento de um valor de log do agent.

Todos os valores passados para o pré-processamento são inicialmente tratados como strings. A conversão para o tipo de valor desejado (conforme definido na configuração do item) é realizada ao final do pipeline de pré-processamento. No entanto, etapas específicas de pré-processamento podem acionar conversões anteriores, se necessário. Para informações técnicas detalhadas, consulte Detalhes do pré-processamento.

Para garantir que sua configuração de pré-processamento funcione conforme o esperado, você pode testá-la.

Veja também: Exemplos de pré-processamento

Configuração

As etapas de pré-processamento são definidas na guia **Pré-processamento** do formulário de configuração do item.

Clique em *Adicionar* para selecionar uma transformação suportada.

O campo *Tipo de informação* é exibido na parte inferior da guia quando pelo menos uma etapa de pré-processamento é definida. Se necessário, é possível alterar o tipo de informação sem sair da guia *Pré-processamento*. Veja Criando um item para a descrição detalhada do parâmetro.

Transformações suportadas

Todas as transformações suportadas estão listadas abaixo. Clique no nome da transformação para ver detalhes completos sobre ela.

Nome	Descrição	Tipo
Expressão regular	Corresponde o valor à expressão regular e substitui pela saída desejada.	Texto
Substituir	Encontra a string de pesquisa e a substitui por outra (ou nada).	
Remover espaços	Remove os caracteres especificados do início e do final do valor.	
Remover espaços à direita	Remove os caracteres especificados do final do valor.	
Remover espaços à esquerda	Remove os caracteres especificados do início do valor.	
XML XPath	Extrai valor ou fragmento de dados XML usando a funcionalidade XPath.	Dados estruturados
JSON Path	Extrai valor ou fragmento de dados JSON usando a funcionalidade JSONPath.	
CSV para JSON	Converte dados de arquivo CSV para o formato JSON.	
XML para JSON	Converte dados em formato XML para JSON.	
Valor SNMP walk	Extrai valor pelo OID/nome MIB especificado e aplica opções de formatação.	SNMP
SNMP walk para JSON	Converte valores SNMP para JSON.	
Valor SNMP get	Aplica opções de formatação ao valor SNMP get.	
Multiplicador personalizado	Multiplica o valor pelo valor inteiro ou de ponto flutuante especificado.	Aritmética
Mudança simples	Calcula a diferença entre o valor atual e o anterior.	Mudança
Mudança por segundo	Calcula a velocidade da mudança de valor (diferença entre o valor atual e o anterior) por segundo.	
Booleano para decimal	Converte o valor do formato booleano para decimal.	Sistemas numéricos
Octal para decimal	Converte o valor do formato octal para decimal.	
Hexadecimal para decimal	Converte o valor do formato hexadecimal para decimal.	
JavaScript	Insira o código JavaScript.	Scripts personalizados
No intervalo	Define um intervalo no qual o valor deve estar.	Validação
Corresponde à expressão regular	Especifica uma expressão regular à qual o valor deve corresponder.	
Não corresponde à expressão regular	Especifica uma expressão regular à qual o valor não deve corresponder.	
Verificar erro em JSON	Verifica uma mensagem de erro em nível de aplicação localizada no JSONPath.	
Verificar erro em XML	Verifica uma mensagem de erro em nível de aplicação localizada no XPath.	
Verificar erro usando expressão regular	Verifica uma mensagem de erro em nível de aplicação usando uma expressão regular.	
Verificar valor não suportado	Verifica se nenhum valor de item pôde ser recuperado.	
Descartar inalterado	Descarta um valor se ele não foi alterado.	Limitação
Descartar inalterado com heartbeat	Descarta um valor se ele não foi alterado dentro do período de tempo definido.	
Padrão Prometheus	Usa a seguinte consulta para extrair os dados necessários das métricas do Prometheus.	Prometheus
Prometheus para JSON	Converte as métricas Prometheus necessárias para JSON.	

Observe que, para as etapas de pré-processamento *Mudança* e *Limitação*, o Zabbix precisa lembrar o último valor para calcular/comparar o novo valor conforme necessário. Esses valores anteriores são tratados pelo gerenciador de pré-processamento. Se o Zabbix server ou proxy for reiniciado ou houver qualquer alteração feita nas etapas de pré-processamento, o último valor do item correspondente será redefinido, resultando em:

- para as etapas *Mudança simples*, *Mudança por segundo* - o próximo valor será ignorado porque não há valor anterior para calcular a mudança;
- para as etapas *Descartar inalterado*, *Descartar inalterado com heartbeat* - o próximo valor nunca será descartado, mesmo que devesse ter sido devido às regras de descarte.

Expressão regular

Corresponde o valor à expressão regular e substitui pela saída necessária.

Parâmetros:

- **pattern** - a expressão regular;
- **output** - o modelo de formatação de saída. Uma sequência de escape \N (onde N=1...9) é substituída pelo N-ésimo grupo capturado. Uma sequência de escape \0 é substituída pelo texto correspondente.

Comentários:

- Uma falha ao corresponder o valor de entrada tornará o item não suportado;
- A expressão regular suporta a extração de no máximo 10 grupos capturados com a sequência \N;
- Se você marcar a caixa *Personalizado em caso de falha*, é possível especificar opções personalizadas de tratamento de erro: descartar o valor, definir um valor especificado ou definir uma mensagem de erro especificada. Em caso de falha na etapa de pré-processamento, o item não se tornará não suportado se a opção de descartar o valor ou definir um valor especificado for selecionada.
- Consulte a seção expressões regulares para alguns exemplos existentes.

Replace

Encontre a string de pesquisa e substitua-a por outra (ou nada).

Parâmetros:

- **string de pesquisa** - a string a ser encontrada e substituída, diferencia maiúsculas de minúsculas (obrigatório);
- **substituição** - a string para substituir a string de pesquisa. A string de substituição também pode estar vazia, permitindo efetivamente excluir a string de pesquisa quando encontrada.

Comentários:

- Todas as ocorrências da string de pesquisa serão substituídas;
- É possível usar sequências de escape para pesquisar ou substituir quebras de linha, retorno de carro, tabulações e espaços ”\n \r \t \s”; a barra invertida pode ser escapada como ”\\” e as sequências de escape podem ser escapadas como ”\\n”;
- O escape de quebras de linha, retorno de carro e tabulações é feito automaticamente durante a descoberta de baixo nível.

Trim

Remove os caracteres especificados do início e do final do valor.

Remover à direita

Remove os caracteres especificados do final do valor.

Left trim

Remove os caracteres especificados do início do valor.

XML XPath

Extrai um valor ou fragmento de dados XML usando a funcionalidade XPath.

Comentários:

- Para que esta opção funcione, o Zabbix server (ou Zabbix proxy) deve ser compilado com suporte a libxml;
- Namespaces não são suportados;
- Se você marcar a caixa *Personalizado em caso de falha*, é possível especificar opções personalizadas de tratamento de erro: descartar o valor, definir um valor especificado ou definir uma mensagem de erro especificada. Em caso de falha na etapa de pré-processamento, o item não ficará como não suportado se a opção de descartar o valor ou definir um valor especificado for selecionada.

Exemplos:

```
number(/document/item/value) #irá extrair '10' de <document><item><value>10</value></item></document>
number(/document/item/@attribute) #irá extrair '10' de <document><item attribute="10"></item></document>
/document/item #irá extrair '<item><value>10</value></item>' de <document><item><value>10</value></item></
```

JSON Path

Extrai um valor ou fragmento de dados JSON usando a funcionalidade JSONPath.

Se você marcar a caixa *Personalizado em caso de falha*, é possível especificar opções personalizadas de tratamento de erro: descartar o valor, definir um valor especificado ou definir uma mensagem de erro especificada. Em caso de falha na etapa de pré-processamento, o item não ficará como não suportado se a opção de descartar o valor ou definir um valor especificado for selecionada.

CSV para JSON

Converte dados de arquivo CSV para o formato JSON.

Para mais informações, consulte: Pré-processamento CSV para JSON.

XML para JSON

Converte dados no formato XML para JSON.

Para mais informações, consulte: Regras de serialização.

Se você marcar a caixa *Personalizado em caso de falha*, é possível especificar opções personalizadas de tratamento de erro: descartar o valor, definir um valor especificado ou definir uma mensagem de erro especificada. Em caso de falha na etapa de pré-processamento, o item não ficará como não suportado se a opção de descartar o valor ou definir um valor especificado for selecionada.

Valor SNMP walk

Extrai o valor pelo OID/nome MIB especificado e aplica opções de formatação:

- **Inalterado** - retorna hex-string como string hexadecimal sem escape (*note* que dicas de exibição ainda são aplicadas);
- **UTF-8 de hex-STRING** - converte hex-string para string UTF-8;
- **MAC de hex-STRING** - valida hex-string como endereço MAC e retorna uma string de endereço MAC adequada (onde ' ' são substituídos por ':');
- **Inteiro de BITS** - converte os primeiros 8 bytes de uma string de bits expressa como uma sequência de caracteres hexadecimais (por exemplo, "1A 2B 3C 4D") em um inteiro sem sinal de 64 bits. Em strings de bits mais longas que 8 bytes, os bytes subsequentes serão ignorados.

Se você marcar a caixa *Personalizado em caso de falha*, é possível especificar opções personalizadas de tratamento de erro: descartar o valor, definir um valor especificado ou definir uma mensagem de erro especificada. Em caso de falha em uma etapa de pré-processamento, o item não se tornará não suportado se a opção de descartar o valor ou definir um valor especificado for selecionada.

SNMP walk para JSON

Converte valores SNMP para JSON.

Especifique um nome de campo no JSON e o caminho OID SNMP correspondente. Os valores dos campos serão preenchidos pelos valores no caminho OID SNMP especificado.

Comentários:

- Opções de formatação de valor semelhantes às da etapa *Valor SNMP walk* estão disponíveis;
- Você pode usar esta etapa de pré-processamento para descoberta de OID SNMP;
- Se você marcar a caixa *Personalizar em caso de falha*, é possível especificar opções personalizadas de tratamento de erro: descartar o valor, definir um valor especificado ou definir uma mensagem de erro especificada. Em caso de falha na etapa de pré-processamento, o item não ficará como não suportado se a opção de descartar o valor ou definir um valor especificado for selecionada.

Valor SNMP get

Aplique opções de formatação ao valor SNMP get:

- **UTF-8 de Hex-STRING** - converte hex-string para string UTF-8;
- **MAC de Hex-STRING** - valida hex-string como endereço MAC e retorna uma string de endereço MAC adequada (onde ' ' são substituídos por ':');
- **Inteiro de BITS** - converte os primeiros 8 bytes de uma string de bits expressa como uma sequência de caracteres hexadecimais (por exemplo, "1A 2B 3C 4D") em um inteiro sem sinal de 64 bits. Em strings de bits mais longas que 8 bytes, os bytes subsequentes serão ignorados.

Se você marcar a caixa *Personalizado em caso de falha*, é possível especificar opções personalizadas de tratamento de erro: descartar o valor, definir um valor especificado ou definir uma mensagem de erro especificada. Em caso de falha em uma etapa de pré-processamento, o item não ficará como não suportado se a opção de descartar o valor ou definir um valor especificado for selecionada.

Multiplicador personalizado

Multiplica o valor pelo valor inteiro ou de ponto flutuante especificado.

Comentários:

- Use esta opção para converter valores recebidos em KB, MBps, etc., em B, Bps. Caso contrário, o Zabbix não poderá definir corretamente os prefixos (K, M, G, etc.).
- Observe que, se o tipo de informação do item for *Numérico (sem sinal)*, os valores recebidos com uma parte fracionária serão truncados (ou seja, '0.9' se tornará '0') antes que o multiplicador personalizado seja aplicado;

- Se você usar um multiplicador personalizado ou armazenar o valor como *Alteração por segundo* para itens com o tipo de informação definido como *Numérico (sem sinal)* e o valor calculado resultante for realmente um número float, o valor calculado ainda será aceito como correto, truncando a parte decimal e armazenando o valor como um inteiro;
- Suportado: notação científica, por exemplo, 1e+70; macros de usuário e macros LLD; strings que incluem macros, por exemplo, {#MACRO}e+10, {$MACRO1}e+{$MACRO2}. As macros devem ser resolvidas para um número inteiro ou de ponto flutuante.
- Se você marcar a caixa de seleção *Personalizado em caso de falha*, é possível especificar opções personalizadas de tratamento de erro: descartar o valor, definir um valor especificado ou definir uma mensagem de erro especificada. Em caso de falha em uma etapa de pré-processamento, o item não se tornará não suportado se a opção de descartar o valor ou definir um valor especificado for selecionada.

Mudança simples

Calcula a diferença entre o valor atual e o valor anterior.

Comentários:

- Esta etapa pode ser útil para medir um valor que está constantemente crescendo;
- Avaliado como **value**-**prev_value**, onde *value* - o valor atual; *prev_value* - o valor recebido anteriormente;
- Apenas uma operação de mudança por item (”Mudança simples” ou ”Mudança por segundo”) é permitida;
- Se o valor atual for menor que o valor anterior, o Zabbix descarta essa diferença (não armazena nada) e aguarda outro valor;
- Se você marcar a caixa *Personalizar em caso de falha*, é possível especificar opções personalizadas de tratamento de erro: descartar o valor, definir um valor especificado ou definir uma mensagem de erro especificada. Em caso de falha em uma etapa de pré-processamento, o item não ficará como não suportado se a opção de descartar o valor ou definir um valor especificado for selecionada.

Mudança por segundo

Calcula a velocidade da mudança de valor (diferença entre o valor atual e o anterior) por segundo.

Comentários:

- Esta etapa é útil para calcular a velocidade por segundo de um valor que está constantemente crescendo;
- Como esse cálculo pode produzir números de ponto flutuante, recomenda-se definir o ’Tipo de informação’ como *Numérico (float)*, mesmo que os valores brutos recebidos sejam inteiros. Isso é especialmente relevante para números pequenos, onde a parte decimal é importante. Se os valores de ponto flutuante forem grandes e puderem exceder o comprimento do campo ’float’, caso em que todo o valor pode ser perdido, na verdade é sugerido usar *Numérico (sem sinal)* e, assim, cortar apenas a parte decimal.
- Avaliado como (**valor**-**valor_anterior**)/(**tempo**-**tempo_anterior**), onde *valor* - o valor atual; *valor_anterior* - o valor recebido anteriormente; *tempo* - o timestamp atual; *tempo_anterior* - o timestamp do valor anterior;
- Apenas uma operação de mudança por item (”Mudança simples” ou ”Mudança por segundo”) é permitida;
- Se o valor atual for menor que o valor anterior, o Zabbix descarta essa diferença (não armazena nada) e aguarda outro valor. Isso ajuda a funcionar corretamente com, por exemplo, um estouro de contadores SNMP de 32 bits.
- Se você marcar a caixa *Personalizado em falha*, é possível especificar opções personalizadas de tratamento de erro: descartar o valor, definir um valor especificado ou definir uma mensagem de erro especificada. Em caso de falha na etapa de pré-processamento, o item não ficará como não suportado se a opção de descartar o valor ou definir um valor especificado for selecionada.

Boolean para decimal

Converte o valor do formato booleano para decimal.

Comentários:

- A representação textual é traduzida para 0 ou 1. Assim, ’TRUE’ é armazenado como 1 e ’FALSE’ é armazenado como 0. Todos os valores são comparados de forma case-insensitive. Atualmente, os valores reconhecidos são, para *TRUE* - true, t, yes, y, on, up, running, enabled, available, ok, master; para *FALSE* - false, f, no, n, off, down, unused, disabled, unavailable, err, slave. Além disso, qualquer valor numérico diferente de zero é considerado como TRUE e zero é considerado como FALSE.
- Se você marcar a caixa *Personalizar em caso de falha*, é possível especificar opções personalizadas de tratamento de erro: descartar o valor, definir um valor especificado ou definir uma mensagem de erro especificada. Em caso de falha em uma etapa de pré-processamento, o item não ficará como não suportado se a opção de descartar o valor ou definir um valor especificado for selecionada.

Octal para decimal

Converte o valor do formato octal para decimal.

Se você marcar a caixa de seleção *Personalizado em caso de falha*, é possível especificar opções personalizadas de tratamento de erro: descartar o valor, definir um valor especificado ou definir uma mensagem de erro especificada. Em caso de falha na etapa de pré-processamento, o item não ficará como não suportado se a opção de descartar o valor ou definir um valor especificado for selecionada.

Hexadecimal para decimal

Converte o valor do formato hexadecimal para decimal.

Se você marcar a caixa de seleção *Personalizado em caso de falha*, é possível especificar opções personalizadas de tratamento de erro: descartar o valor, definir um valor especificado ou definir uma mensagem de erro especificada. Em caso de falha na etapa de pré-processamento, o item não ficará como não suportado se a opção de descartar o valor ou definir um valor especificado for selecionada.

JavaScript

Digite o código JavaScript no editor modal que é aberto ao clicar no campo de parâmetro ou no ícone de lápis ao lado dele.

> **Attention:**
> Não use atribuições não declaradas no JavaScript de pré-processamento. Use `var` para declarar variáveis locais.

Comentários:

- O comprimento do JavaScript disponível depende do banco de dados usado;
- Para mais informações, consulte: Pré-processamento Javascript.

No intervalo

Defina um intervalo em que um valor deve estar, especificando valores mínimos/máximos (inclusive).

Comentários:

- Valores numéricos são aceitos (incluindo qualquer número de dígitos, parte decimal opcional e parte exponencial opcional, valores negativos);
- O valor mínimo deve ser menor que o máximo;
- Pelo menos um valor deve existir;
- Macros de usuário e macros de descoberta de baixo nível podem ser usadas;
- Se você marcar a caixa *Personalizar em caso de falha*, é possível especificar opções personalizadas de tratamento de erro: descartar o valor, definir um valor especificado ou definir uma mensagem de erro especificada. Em caso de falha na etapa de pré-processamento, o item não ficará como não suportado se a opção de descartar o valor ou definir um valor especificado for selecionada.

Corresponde à expressão regular

Especifique uma expressão regular que um valor deve corresponder.

Se você marcar a caixa de seleção *Personalizado em caso de falha*, é possível especificar opções personalizadas de tratamento de erro: descartar o valor, definir um valor especificado ou definir uma mensagem de erro especificada. Em caso de falha na etapa de pré-processamento, o item não ficará como não suportado se a opção de descartar o valor ou definir um valor especificado for selecionada.

Não corresponde à expressão regular

Especifique uma expressão regular que um valor não deve corresponder.

Se você marcar a caixa de seleção *Personalizado em falha*, é possível especificar opções personalizadas de tratamento de erro: descartar o valor, definir um valor especificado ou definir uma mensagem de erro especificada. Em caso de falha na etapa de pré-processamento, o item não ficará como não suportado se a opção de descartar o valor ou definir um valor especificado for selecionada.

Verificar erro no JSON

Verifica por uma mensagem de erro em nível de aplicação localizada no JSONPath. Interrompe o processamento se for bem-sucedido e a mensagem não estiver vazia; caso contrário, continua o processamento com o valor que estava antes desta etapa de pré-processamento.

Comentários:

- Esses erros de serviço externo são relatados ao usuário como estão, sem adicionar informações da etapa de pré-processamento;
- Nenhum erro será relatado em caso de falha ao analisar um JSON inválido;

- Se você marcar a caixa *Personalizado em caso de falha*, é possível especificar opções personalizadas de tratamento de erro: descartar o valor, definir um valor especificado ou definir uma mensagem de erro especificada. Em caso de falha na etapa de pré-processamento, o item não ficará como não suportado se a opção de descartar o valor ou definir um valor especificado for selecionada.

Verificar erro no XML

Verifica por uma mensagem de erro em nível de aplicação localizada no XPath. Interrompe o processamento se for bem-sucedido e a mensagem não estiver vazia; caso contrário, continua o processamento com o valor que estava antes desta etapa de pré-processamento.

Comentários:

- Esses erros de serviço externo são relatados ao usuário como estão, sem adicionar informações da etapa de pré-processamento;
- Nenhum erro será relatado em caso de falha ao analisar um XML inválido;
- Se você marcar a caixa *Personalizado em caso de falha*, é possível especificar opções personalizadas de tratamento de erro: descartar o valor, definir um valor especificado ou definir uma mensagem de erro especificada. Em caso de falha na etapa de pré-processamento, o item não ficará como não suportado se a opção de descartar o valor ou definir um valor especificado for selecionada.

Verificar erro usando uma expressão regular

Verifique uma mensagem de erro em nível de aplicação usando uma expressão regular. Pare o processamento se for bem-sucedido e a mensagem não estiver vazia; caso contrário, continue o processamento com o valor que estava antes desta etapa de pré-processamento.

Parâmetros:

- **pattern** - a expressão regular;
- **output** - o modelo de formatação de saída. Uma sequência de escape \N (onde N=1…9) é substituída pelo N-ésimo grupo correspondente. Uma sequência de escape \0 é substituída pelo texto correspondente.

Comentários:

- Esses erros de serviço externo são relatados ao usuário como estão, sem adicionar informações da etapa de pré-processamento;
- Se você marcar a caixa *Personalizado em caso de falha*, é possível especificar opções personalizadas de tratamento de erro: descartar o valor, definir um valor especificado ou definir uma mensagem de erro especificada. Em caso de falha na etapa de pré-processamento, o item não ficará como não suportado se a opção de descartar o valor ou definir um valor especificado for selecionada.

Verificar valor não suportado

Verifica se nenhum valor de item pôde ser recuperado. Especifique como a falha deve ser processada, com base na inspeção da mensagem de erro retornada.

Parâmetros:

- **scope** - selecione o escopo de processamento de erro:
qualquer erro - qualquer erro;
erro corresponde - apenas o erro que corresponde à expressão regular especificada em *pattern*;
erro não corresponde - apenas o erro que não corresponde à expressão regular especificada em *pattern*
- **pattern** - a expressão regular para comparar com o erro. Se *qualquer erro* for selecionado no parâmetro scope, este campo não será exibido. Se exibido, este campo é obrigatório.

Comentários:

- Normalmente, a ausência/falha ao recuperar um valor faria com que o item se tornasse não suportado. Esta etapa de pré-processamento permite modificar esse comportamento. Se você marcar a caixa *Personalizado em caso de falha* (sempre marcada e desabilitada para esta etapa de pré-processamento), é possível especificar opções personalizadas de tratamento de erro: descartar o valor, definir um valor especificado ou definir uma mensagem de erro especificada. Em caso de falha na etapa de pré-processamento, o item não se tornará não suportado se a opção de descartar o valor ou definir um valor especificado for selecionada.
- Esta etapa de pré-processamento apenas verifica se nenhum valor de item pôde ser recuperado. Ela não verifica, por exemplo, se o tipo do valor recuperado (por exemplo, string) corresponde ao tipo de informação do item (por exemplo, numérico); para detalhes, consulte Exemplos de pré-processamento. Se houver incompatibilidade de tipo, o item ainda poderá se tornar não suportado após a execução de todas as etapas de pré-processamento. Para verificar incompatibilidade de tipo, você pode usar, por exemplo, a etapa de pré-processamento *Multiplicador personalizado*; consulte Exemplos de pré-processamento.
- A captura de grupos de expressão regular é suportada nos campos *Definir valor como* ou *Definir erro como*. Use \N (onde N=1…9) para recuperar o N-ésimo grupo correspondente; use \0 para recuperar o texto correspondente;

- Essas etapas são sempre executadas como as primeiras etapas de pré-processamento e são colocadas acima de todas as outras após salvar as alterações no item;
- Várias etapas *Verificar valor não suportado* são suportadas, na ordem especificada. Uma etapa para *qualquer erro* será automaticamente colocada como a última etapa neste grupo.

Descartar inalterado

Descarta um valor se ele não foi alterado.

Comentários:

- Se um valor for descartado, ele não será salvo no banco de dados e o Zabbix server não terá conhecimento de que esse valor foi recebido. Nenhuma expressão de trigger será avaliada, como resultado, nenhum problema para triggers relacionadas será criado/resolvido. Funções funcionarão apenas com base nos dados que estão realmente salvos no banco de dados. Como as tendências são construídas com base nos dados do banco de dados, se não houver valor salvo para uma hora, também não haverá dados de tendências para essa hora.
- Apenas uma opção de limitação pode ser especificada por item.

Descartar inalterado com heartbeat

Descarta um valor se ele não tiver sido alterado dentro do período de tempo definido (em segundos).

Comentários:

- Valores inteiros positivos são suportados para especificar os segundos (mínimo - 1 segundo);
- Sufixos de tempo podem ser usados (por exemplo, 30s, 1m, 2h, 1d);
- Macros de usuário e macros de descoberta de baixo nível podem ser usadas;
- Se um valor for descartado, ele não será salvo no banco de dados e o Zabbix server não terá conhecimento de que esse valor foi recebido. Nenhuma expressão de trigger será avaliada, como resultado, nenhum problema para triggers relacionadas será criado/resolvido. Funções funcionarão apenas com base nos dados que estão realmente salvos no banco de dados. Como as tendências são construídas com base nos dados do banco de dados, se não houver valor salvo para uma hora, também não haverá dados de tendências para essa hora.
- Apenas uma opção de limitação pode ser especificada por item.

Padrão Prometheus

Use a seguinte consulta para extrair os dados necessários das métricas do Prometheus.

Consulte Verificações do Prometheus para mais detalhes.

Prometheus para JSON

Converta as métricas Prometheus necessárias para JSON.

Consulte Verificações Prometheus para mais detalhes.

Suporte a macros

Macros de usuário e macros de usuário com contexto são suportadas em:

- parâmetros de etapas de pré-processamento, incluindo código JavaScript;
- parâmetros de tratamento de erro personalizado (campos *Definir valor como* e *Definir erro como*).

Note:
O contexto da macro é ignorado quando uma macro é substituída por seu valor. O valor da macro é inserido no código como está, não é possível adicionar escape adicional antes de colocar o valor no código JavaScript. Esteja ciente de que isso pode causar erros de JavaScript em alguns casos.

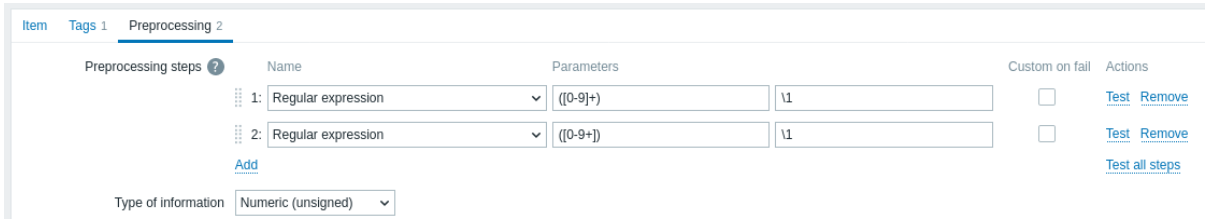
Testando

Veja testando o pré-processamento.

1 Teste de pré-processamento

Testando

Testar as etapas de pré-processamento é útil para garantir que pipelines de pré-processamento complexos produzam os resultados esperados, sem precisar esperar que o valor do item seja recebido e pré-processado.

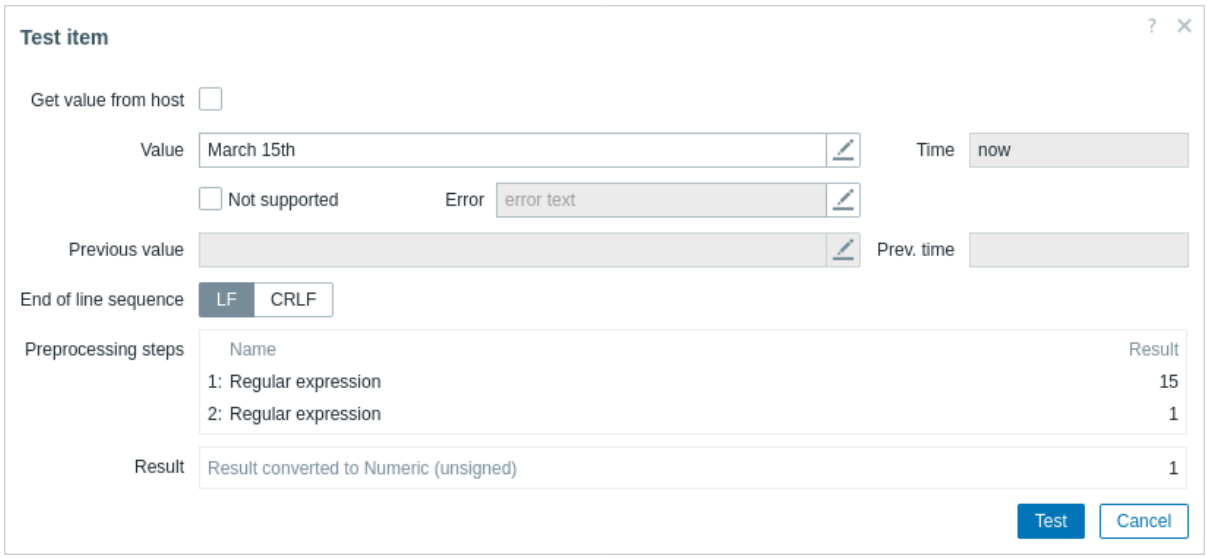


É possível testar:

- contra um valor hipotético
- contra um valor real de um host

Cada etapa de pré-processamento pode ser testada individualmente, assim como todas as etapas podem ser testadas juntas. Ao clicar no botão *Testar* ou *Testar todas as etapas* respectivamente no bloco de Ações, uma janela de teste é aberta.

Testando valor hipotético



Parâmetro	Descrição
Obter valor do host	Se você deseja testar um valor hipotético, deixe esta caixa de seleção desmarcada. Veja também: Testando valor real.
Valor	Insira o valor de entrada para testar. Clicar no campo do parâmetro ou no botão de visualizar/editar abrirá uma janela de área de texto para inserir o valor ou bloco de código.
Não suportado	Marque esta caixa de seleção para testar um valor não suportado. Esta opção é útil para testar a etapa de pré-processamento *Verificar valor não suportado*.
Erro	Insira o texto do erro. Este campo é habilitado quando *Obter valor do host* está desmarcado, mas *Não suportado* está marcado. Se *Obter valor do host* estiver marcado, este campo será preenchido com a mensagem de erro real (somente leitura) do host.
Hora	A hora do valor de entrada é exibida: `agora` (somente leitura).
Valor anterior	Insira um valor de entrada anterior para comparar. Somente para as etapas de pré-processamento *Alteração* e *Throttling*.
Hora anterior	Insira a hora do valor de entrada anterior para comparar. Somente para as etapas de pré-processamento *Alteração* e *Throttling*. O valor padrão é baseado no valor do campo 'Intervalo de atualização' do item (se '1m', então este campo é preenchido com `agora-1m`). Se nada for especificado ou o usuário não tiver acesso ao host, o padrão é `agora-30s`.
Macros	Se houver macros utilizadas, elas são listadas juntamente com seus valores. Os valores são editáveis para fins de teste, mas as alterações só serão salvas dentro do contexto de teste.
Sequência de fim de linha	Selecione a sequência de fim de linha para valores de entrada multilinha: **LF** - sequência LF (line feed) **CRLF** - sequência CRLF (carriage-return line-feed).

Parâmetro	Descrição
Etapas de pré-processamento	As etapas de pré-processamento são listadas; o resultado do teste é exibido para cada etapa após clicar no botão *Testar*. Os resultados dos testes são truncados para um tamanho máximo de 512KB quando enviados para o frontend. Se um resultado for truncado, um ícone de aviso será exibido. A descrição do aviso é exibida ao passar o mouse. Observe que dados maiores que 512KB ainda são processados totalmente pelo Zabbix server. Se a etapa falhar no teste, um ícone de erro será exibido. A descrição do erro é exibida ao passar o mouse. Caso "Personalizado em falha" seja especificado para a etapa e essa ação seja executada, uma nova linha aparece logo após a linha da etapa de teste de pré-processamento, mostrando qual ação foi realizada e qual resultado produziu (erro ou valor).
Resultado	O resultado final do teste das etapas de pré-processamento é exibido em todos os casos quando todas as etapas são testadas juntas (quando você clica no botão *Testar todas as etapas*). O tipo de conversão para o tipo de valor do item também é exibido, por exemplo `Resultado convertido para Numérico (sem sinal)`. Os resultados dos testes são truncados para um tamanho máximo de 512KB quando enviados para o frontend. Se um resultado for truncado, um ícone de aviso será exibido. A descrição do aviso é exibida ao passar o mouse. Observe que dados maiores que 512KB ainda são processados totalmente pelo Zabbix server.

Clique em *Testar* para ver o resultado após cada etapa de pré-processamento.

Os valores de teste são armazenados entre as sessões de teste para etapas individuais ou todas as etapas, permitindo que o usuário altere as etapas de pré-processamento ou a configuração do item e, em seguida, retorne à janela de teste sem precisar inserir as informações novamente. No entanto, os valores são perdidos ao atualizar a página.

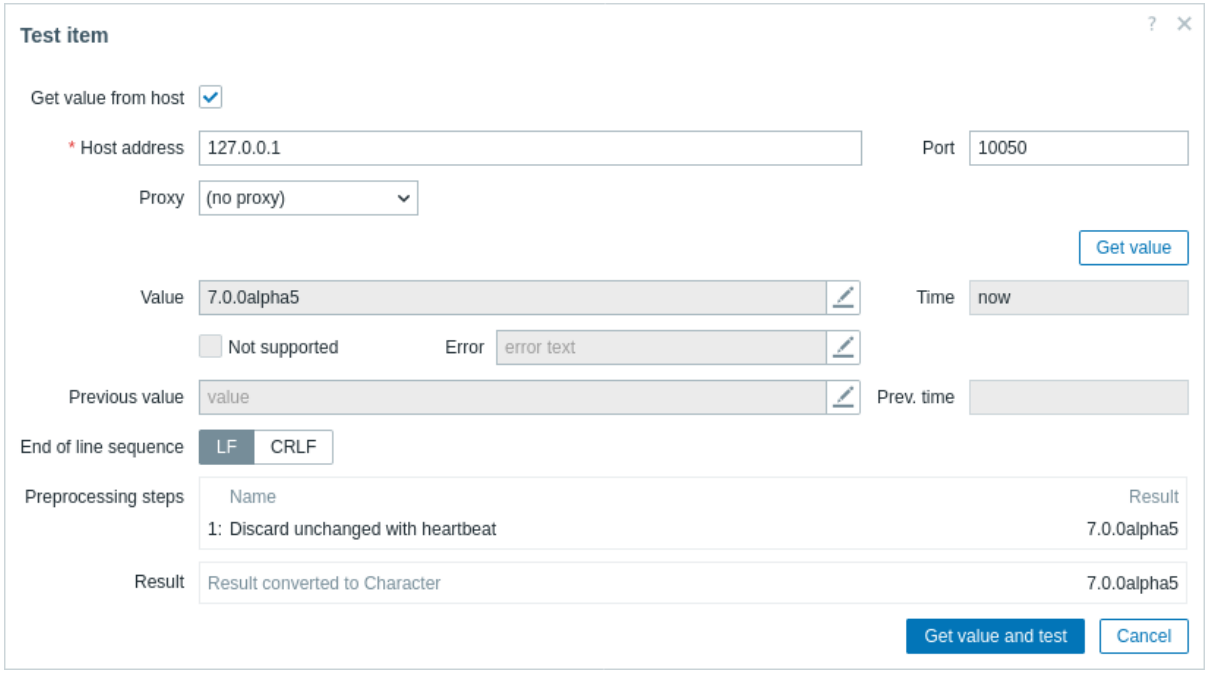
O teste é realizado pelo Zabbix server. O frontend envia uma solicitação correspondente ao server e aguarda o resultado. A solicitação contém o valor de entrada e as etapas de pré-processamento (com macros de usuário expandidas). Para as etapas *Alteração* e *Throttling*, um valor e hora anteriores opcionais podem ser especificados. O server responde com os resultados para cada etapa de pré-processamento.

Todos os erros técnicos ou erros de validação de entrada são exibidos na caixa de erro na parte superior da janela de teste.

Testando valor real

Para testar o pré-processamento com um valor real:

- Marque a caixa de seleção *Obter valor do host*
- Insira ou verifique os parâmetros do host (endereço do host, porta, nome do proxy/sem proxy) e detalhes específicos do item (como comunidade SNMPv2 ou credenciais de segurança SNMPv3). Esses campos são sensíveis ao contexto:
  - Os valores são preenchidos automaticamente quando possível, ou seja, para itens que exigem um agent, obtendo as informações da interface agent selecionada do host
  - Os valores devem ser preenchidos manualmente para itens de template
  - Valores de macros em texto simples são resolvidos
  - Quando o valor do campo (ou parte do valor) for uma macro secreta ou macro do Vault, o campo ficará vazio e deverá ser preenchido manualmente. Se qualquer parâmetro do item contiver um valor de macro secreto, a seguinte mensagem de aviso será exibida: "O item contém macros definidas pelo usuário com valores secretos. Os valores dessas macros devem ser inseridos manualmente."
  - Os campos são desabilitados quando não são necessários no contexto do tipo de item (por exemplo, os campos de endereço do host e proxy são desabilitados para itens calculados)
- Clique em *Obter valor e testar* para testar o pré-processamento



Se você especificou um mapeamento de valor no formulário de configuração do item (campo 'Exibir valor'), a caixa de diálogo de teste do item mostrará outra linha após o resultado final, chamada 'Resultado com mapeamento de valor aplicado'.

Parâmetros específicos para obter um valor real de um host:

Parâmetro	Descrição
Obter valor do host	Marque esta caixa de seleção para obter um valor real do host.
Endereço do host	Insira o endereço do host. Este campo é preenchido automaticamente pelo endereço da interface do host do item.
Porta	Insira a porta do host. Este campo é preenchido automaticamente pela porta da interface do host do item.
Campos adicionais para interfaces SNMP (Versão SNMP, Comunidade SNMP, Nome do contexto, etc.)	Veja Configurando o monitoramento SNMP para detalhes adicionais sobre a configuração de uma interface SNMP (v1, v2 e v3). Esses campos são preenchidos automaticamente a partir da interface do host do item.
Proxy	Especifique o proxy se o host for monitorado por um proxy. Este campo é preenchido automaticamente pelo proxy do host (se houver).
Valor	Valor recuperado do host. Clicar no campo de parâmetro ou no botão de visualizar/editar abrirá uma janela de área de texto do valor ou bloco de código. Os valores são truncados para um tamanho máximo de 512KB e apenas no frontend. Se um resultado for truncado, um ícone de aviso será exibido. A descrição do aviso é exibida ao passar o mouse. Observe que dados maiores que 512KB ainda são processados totalmente pelo Zabbix server.

Para os demais parâmetros, consulte Testando valor hipotético acima.

2 Detalhes do pré-processamento

Visão geral

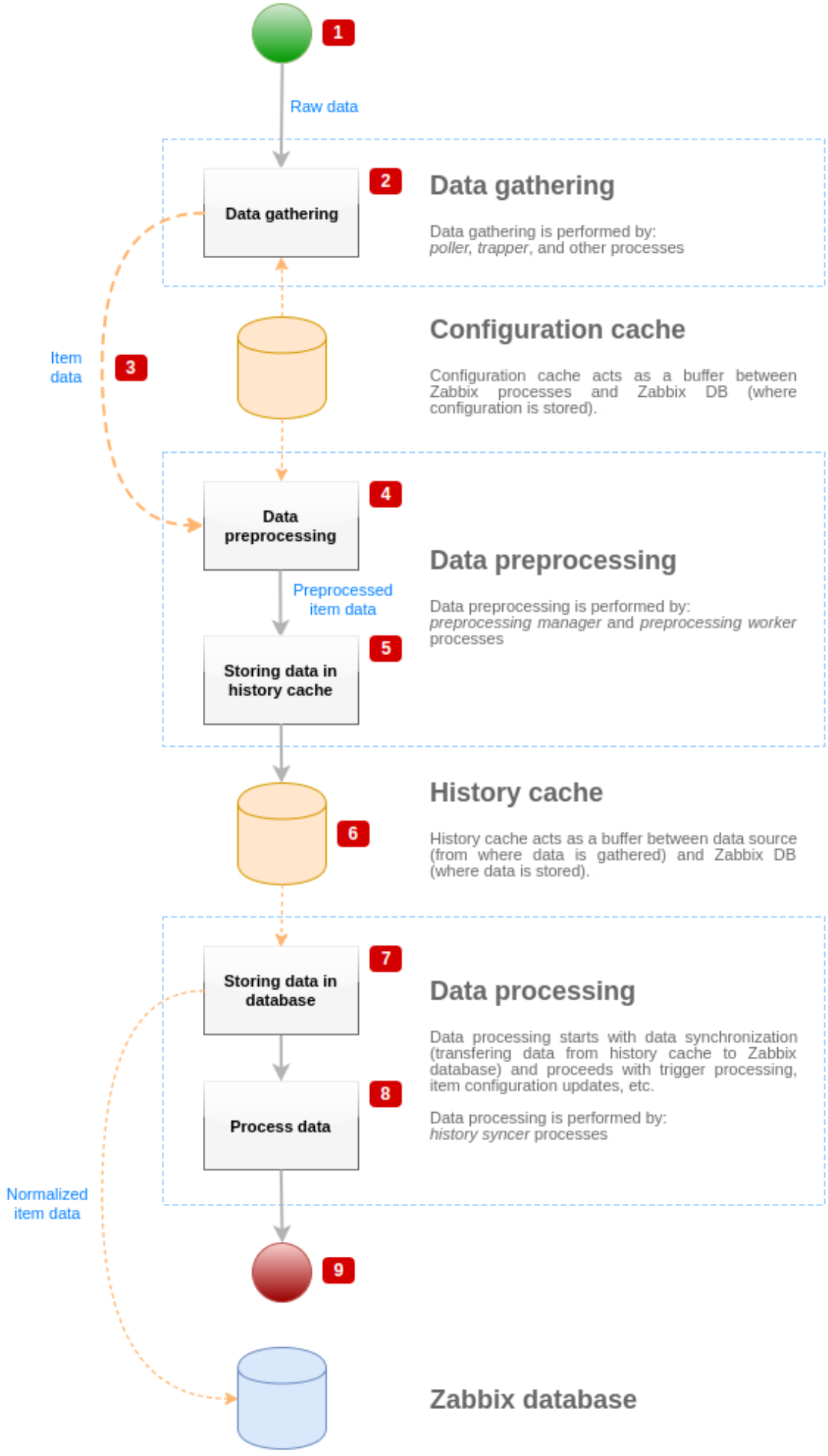
Esta seção fornece detalhes sobre o pré-processamento de valores de item. O pré-processamento de valores de item permite definir e executar regras de transformação para os valores de item recebidos.

O pré-processamento é gerenciado pelo processo de gerenciamento de pré-processamento juntamente com os workers de pré-processamento que executam as etapas de pré-processamento. Todos os valores com pré-processamento (antes do Zabbix 7.0.17, todos os valores), recebidos de diferentes coletores de dados, passam pelo gerenciador de pré-processamento antes de serem adicionados ao cache de histórico. A comunicação IPC baseada em socket é usada entre os coletores de dados (pollers, trappers,

etc.) e o processo de pré-processamento. O pré-processamento é realizado pelo Zabbix server ou pelo Zabbix proxy (para os itens monitorados pelo proxy).

Processamento de valor de item

Para visualizar o fluxo de dados desde a fonte de dados até o banco de dados do Zabbix, podemos usar o seguinte diagrama simplificado:


1
Raw data
Data gathering
2
Data gathering
Data gathering is performed by:
poller, trapper, and other processes
Item data
3
Configuration cache
Configuration cache acts as a buffer between Zabbix processes and Zabbix DB (where configuration is stored).
Data preprocessing
4
Data preprocessing
Data preprocessing is performed by:
preprocessing manager and preprocessing worker processes
Preprocessed item data
Storing data in history cache
5
History cache
6
History cache acts as a buffer between data source (from where data is gathered) and Zabbix DB (where data is stored).
Storing data in database
7
Data processing
Data processing starts with data synchronization (transfering data from history cache to Zabbix database) and proceeds with trigger processing, item configuration updates, etc.
Process data
8
Data processing is performed by:
history syncer processes
Normalized item data
9
Zabbix database

O diagrama acima mostra apenas processos, objetos e ações relacionados ao processamento de valores de item de forma **simplificada**. O diagrama não mostra mudanças condicionais de direção, tratamento de erros ou loops. O cache de dados local do gerenciador de pré-processamento também não é mostrado porque não afeta o fluxo de dados diretamente. O objetivo deste diagrama é mostrar os processos envolvidos no processamento de valores de item e a forma como eles interagem.

- A coleta de dados começa com dados brutos de uma fonte de dados. Neste ponto, os dados contêm apenas ID, timestamp e valor (podem ser múltiplos valores também).
- Não importa qual tipo de coletor de dados é usado, a ideia é a mesma para checagens ativas ou passivas, para items trapper, etc., pois isso apenas muda o formato dos dados e o iniciador da comunicação (ou o coletor de dados está esperando por uma conexão e dados, ou o coletor de dados inicia a comunicação e solicita os dados). Os dados brutos são validados, a configuração do item é recuperada do cache de configuração (os dados são enriquecidos com os dados de configuração).
- Um mecanismo IPC baseado em socket é usado para passar dados dos coletores de dados para o gerenciador de pré-processamento. Neste ponto, o coletor de dados continua a coletar dados sem esperar pela resposta do gerenciador de pré-processamento.
- O pré-processamento dos dados é realizado. Isso inclui a execução das etapas de pré-processamento e o processamento de items dependentes.

Note:
Um item pode mudar seu estado para NÃO SUPORTADO enquanto o pré-processamento é realizado se qualquer uma das etapas de pré-processamento falhar.

- Os dados históricos do cache de dados local do gerenciador de pré-processamento estão sendo transferidos para o cache de histórico.
- Neste ponto, o fluxo de dados para até a próxima sincronização do cache de histórico (quando o processo de sincronização de histórico realiza a sincronização dos dados).
- O processo de sincronização começa com a normalização dos dados antes de armazená-los no banco de dados do Zabbix. A normalização dos dados realiza conversões para o tipo de item desejado (tipo definido na configuração do item), incluindo o truncamento de dados textuais com base nos tamanhos predefinidos permitidos para esses tipos (HISTORY_STR_VALUE_LEN para string, HISTORY_TEXT_VALUE_LEN para texto e HISTORY_LOG_VALUE_LEN para valores de log). Os dados são enviados para o banco de dados do Zabbix após a normalização ser concluída.

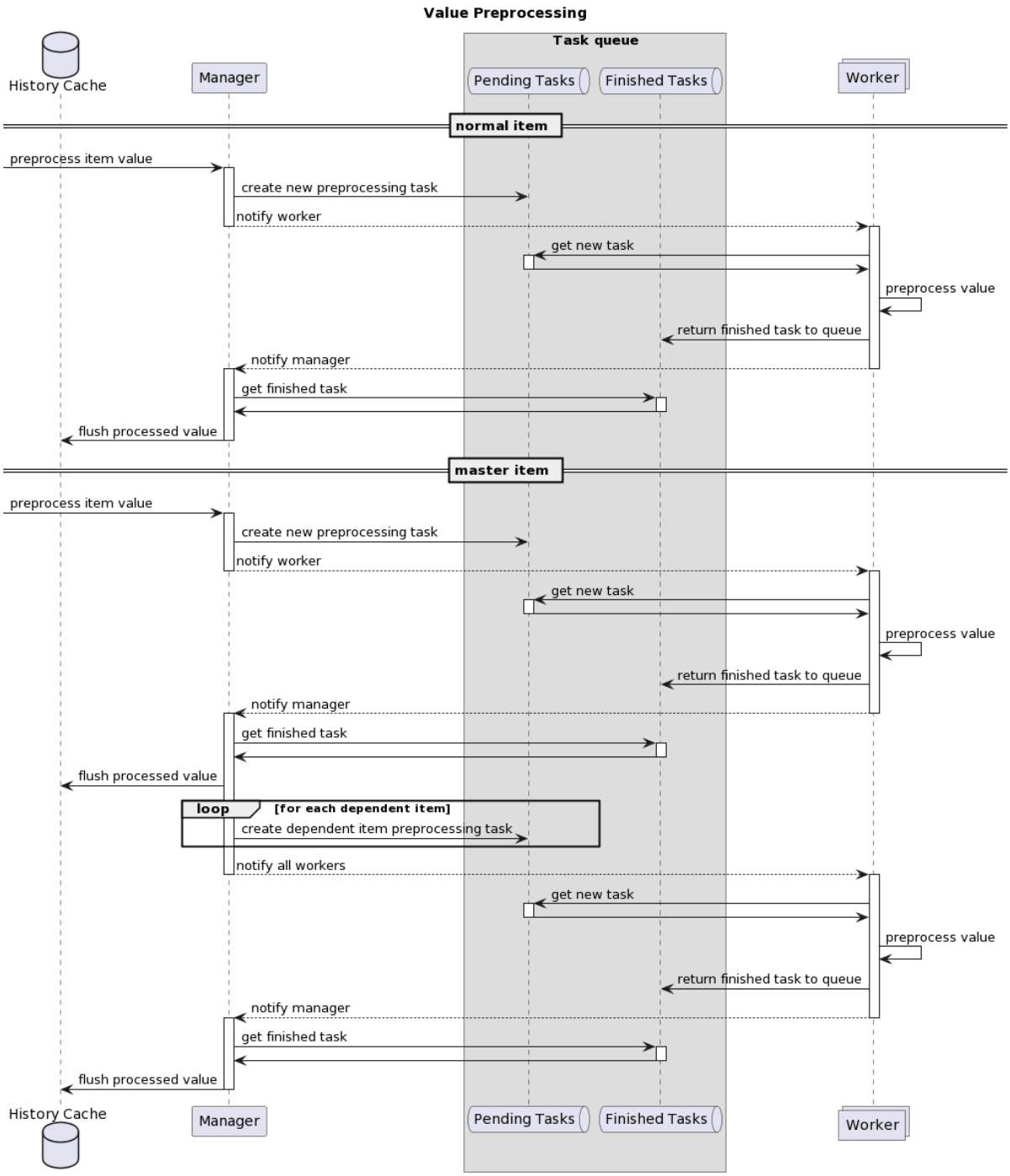
Note:
Um item pode mudar seu estado para NÃO SUPORTADO se a normalização dos dados falhar (por exemplo, quando um valor textual não pode ser convertido para número).

- Os dados coletados estão sendo processados - triggers são verificadas, a configuração do item é atualizada se o item se tornar NÃO SUPORTADO, etc.
- Isso é considerado o fim do fluxo de dados do ponto de vista do processamento de valores de item.

Pré-processamento do valor do item

O pré-processamento dos dados é realizado nas seguintes etapas:

- Se o item não possui pré-processamento nem itens dependentes, seu valor é adicionado ao cache de histórico ou enviado ao gerenciador LLD. Caso contrário, o valor do item é passado para o gerenciador de pré-processamento usando um mecanismo IPC baseado em socket UNIX (antes do Zabbix 7.0.17, todos os valores são passados pelo gerenciador de pré-processamento antes de serem adicionados ao cache de histórico ou enviados ao gerenciador LLD).
- Uma tarefa de pré-processamento é criada e adicionada à fila, e os workers de pré-processamento são notificados sobre a nova tarefa.
- Neste ponto, o fluxo de dados é interrompido até que haja pelo menos um worker de pré-processamento desocupado (ou seja, que não esteja executando nenhuma tarefa).
- Quando um worker de pré-processamento está disponível, ele pega a próxima tarefa da fila.
- Após a conclusão do pré-processamento (tanto em caso de falha quanto de sucesso na execução das etapas de pré-processamento), o valor pré-processado é adicionado à fila de tarefas concluídas e o gerenciador é notificado sobre uma nova tarefa concluída.
- O gerenciador de pré-processamento converte o resultado para o formato desejado (definido pelo tipo de valor do item) e o adiciona ao cache de histórico ou o envia ao gerenciador LLD.
- Se houver itens dependentes para o item processado, então os itens dependentes são adicionados à fila de pré-processamento com o valor do item mestre já pré-processado. Os itens dependentes são enfileirados ignorando as solicitações normais de pré-processamento de valor, mas apenas para itens mestres com o valor definido e que não estejam em estado NÃO SUPORTADO.



Observe que, no diagrama, o pré-processamento do item mestre está ligeiramente simplificado, ignorando o cache de pré-processamento.

Fila de pré-processamento

A fila de pré-processamento é organizada da seguinte forma:

- a lista de tarefas pendentes:
  - tarefas criadas diretamente a partir de solicitações de pré-processamento de valores na ordem em que foram recebidas
- a lista de tarefas imediatas (processadas antes das tarefas pendentes):
  - tarefas de teste (criadas em resposta a solicitações de teste de item/pré-processamento pelo frontend)
  - tarefas de item dependente
  - tarefas de sequência (tarefas que devem ser executadas em uma ordem estrita):
    - tendo etapas de pré-processamento usando o último valor:
      - change
      - throttling
      - JavaScript (cache de bytecode)
    - cache de pré-processamento de item dependente
- a lista de tarefas finalizadas

Cache de pré-processamento

O cache de pré-processamento foi introduzido para melhorar o desempenho do pré-processamento para vários items dependentes que possuem etapas de pré-processamento semelhantes (o que é um resultado comum do LLD).

O cache é feito pré-processando um item dependente e reutilizando alguns dos dados internos de pré-processamento para o restante dos items dependentes. O cache de pré-processamento é suportado apenas para a primeira etapa de pré-processamento dos seguintes tipos:

- Padrão Prometheus (indexa a entrada por métricas)
- JSONPath (analisa os dados em uma árvore de objetos e indexa a primeira expressão `[?(@.path == "value")]`)

Trabalhadores de pré-processamento

O arquivo de configuração do Zabbix server permite que os usuários definam a quantidade de threads de trabalhadores de pré-processamento. O parâmetro de configuração StartPreprocessors deve ser usado para definir o número de instâncias pré-iniciadas de trabalhadores de pré-processamento, que deve pelo menos corresponder ao número de núcleos de CPU disponíveis.

Se as tarefas de pré-processamento não forem limitadas pela CPU e envolverem solicitações de rede frequentes, é recomendada a configuração de trabalhadores adicionais. O número ideal de trabalhadores de pré-processamento pode ser determinado por muitos fatores, incluindo a contagem de items "pré-processáveis" (items que exigem a execução de qualquer etapa de pré-processamento), a contagem de processos de coleta de dados, a média de etapas para o pré-processamento de items, etc. Trabalhadores insuficientes podem levar a um alto uso de memória. Para solucionar o uso excessivo de memória em sua instalação do Zabbix, consulte Perfilando o uso excessivo de memória com tcmalloc.

Mas assumindo que não há operações pesadas de pré-processamento, como análise de grandes blocos de XML/JSON, o número de trabalhadores de pré-processamento pode corresponder ao número total de coletores de dados. Dessa forma, haverá na maioria das vezes (exceto nos casos em que os dados do coletor chegam em massa) pelo menos um trabalhador de pré-processamento desocupado para os dados coletados.

Warning:
Muitos processos de coleta de dados (pollers, unreachable pollers, ODBC pollers, HTTP pollers, Java pollers, pingers, trappers, proxypollers) juntamente com o IPMI manager, SNMP trapper e trabalhadores de pré-processamento podem esgotar o limite de descritores de arquivos por processo para o gerente de pré-processamento.

 Esgotar o limite de descritores de arquivos por processo fará com que o Zabbix server pare, normalmente logo após a inicialização, mas às vezes levando mais tempo. Para evitar tais problemas, revise o arquivo de configuração do Zabbix server para otimizar o número de verificações e processos simultâneos. Além disso, se necessário, certifique-se de que o limite de descritores de arquivos esteja definido suficientemente alto, verificando e ajustando os limites do sistema.

Pipeline de processamento de valor

O processamento do valor do item é executado em várias etapas (ou fases) por vários processos. Isso pode causar:

- Um item dependente pode receber valores, enquanto o valor mestre não pode. Isso pode ser alcançado usando o seguinte caso de uso:
  - O item mestre tem o tipo de valor `UINT` (um item trapper pode ser usado), o item dependente tem o tipo de valor `TEXT`.
  - Nenhuma etapa de pré-processamento é necessária para os itens mestre e dependente.
  - Um valor textual (por exemplo, "abc") deve ser passado para o item mestre.
  - Como não há etapas de pré-processamento a serem executadas, o gerenciador de pré-processamento verifica se o item mestre não está no estado NÃO SUPORTADO e se o valor está definido (ambos são verdadeiros) e coloca o item dependente na fila com o mesmo valor do item mestre (já que não há etapas de pré-processamento).
  - Quando ambos os itens mestre e dependente chegam à fase de sincronização do histórico, o item mestre se torna NÃO SUPORTADO devido ao erro de conversão de valor (dados textuais não podem ser convertidos para inteiro sem sinal).

Como resultado, o item dependente recebe um valor, enquanto o item mestre muda seu estado para NÃO SUPORTADO.

- Um item dependente recebe um valor que não está presente no histórico do item mestre. O caso de uso é muito semelhante ao anterior, exceto pelo tipo do item mestre. Por exemplo, se o tipo `CHAR` for usado para o item mestre, então o valor do item mestre será truncado na fase de sincronização do histórico, enquanto os itens dependentes receberão seus valores do valor inicial (não truncado) do item mestre.

3 Exemplos de pré-processamento

Visão geral

Esta seção apresenta exemplos de uso de etapas de pré-processamento para realizar algumas tarefas práticas.

Filtrando registros do log de eventos do VMware

Este exemplo usa a etapa de pré-processamento Corresponde à expressão regular para filtrar eventos desnecessários do log de eventos do VMware.

1. Em um host VMware Hypervisor em funcionamento, verifique se o item vmware.eventlog está presente e funcionando corretamente. Observe que o item de log de eventos já pode estar presente no hypervisor se um template VMware foi vinculado durante a criação do host.

2. No host VMware Hypervisor, crie um item dependente do tipo *Log* e defina o item de log de eventos como seu item mestre.

3. Na guia *Pré-processamento* do item dependente, clique em *Adicionar* para criar uma etapa de pré-processamento e selecione *Corresponde à expressão regular* no menu suspenso. Em seguida, especifique um dos seguintes padrões:

- Para filtrar todos os eventos de log:

```
.* logged in .*
```

- Para filtrar linhas contendo nomes de usuário após "User":

```
\bUser\s+\K\S+
```

Attention:
Se a expressão regular não for correspondida, o item dependente se tornará não suportado com uma mensagem de erro correspondente. Para evitar isso, marque a caixa *Personalizado em caso de falha* e selecione uma opção como descartar o valor ou definir um valor personalizado. Observe que valores descartados não são armazenados no banco de dados; como resultado, triggers não são avaliados e dados de tendência não são gerados.

Como alternativa, você pode usar a etapa de pré-processamento Expressão regular para extrair grupos correspondentes e controlar a saída:

- Para extrair e exibir todo o evento de log contendo "logged in", especifique os seguintes parâmetros:

```
Padrão: .*logged in.*
Saída: \0
```

- Para extrair e exibir nomes de usuário após "User":

```
Padrão: User (.*?)(?=\ )
Saída: \1
```

Verificando o tipo de valor recuperado

Este exemplo usa a etapa de pré-processamento Multiplicador personalizado para verificar se o tipo de valor recuperado do item é numérico.

Na aba *Pré-processamento* de um item, selecione a etapa de pré-processamento *Multiplicador personalizado* e especifique o seguinte parâmetro (multiplica o valor recuperado por 1):

```
1
```

Attention:
Se o pré-processamento falhar (por exemplo, se a entrada não for numérica), o item se tornará não suportado com uma mensagem de erro correspondente. Para evitar isso, marque a caixa *Personalizar em caso de falha* e selecione uma opção, como descartar o valor ou definir um valor personalizado. Observe que valores descartados não são armazenados no banco de dados; como resultado, triggers não são avaliados e dados de tendência não são gerados.

Verificando valor não suportado

Este exemplo utiliza a etapa de pré-processamento Verificar valor não suportado para verificar se o valor do item não pôde ser recuperado.

Quando um processo de poller do Zabbix server/proxy tenta coletar o valor de um item, ele pode:

- Retornar um resultado válido.
- Retornar um resultado que inicialmente parece válido, mas pode se tornar não suportado posteriormente (por exemplo, devido a uma incompatibilidade de tipo de valor após o pré-processamento).
- Retornar um erro ao coletar o valor, fazendo com que o item se torne não suportado. As causas mais comuns incluem:
  - Chave de item desconhecida (para Zabbix agent, Verificação simples ou itens internos do Zabbix)
  - OID desconhecido (SNMP agent), sensor desconhecido (IPMI agent) ou métrica JMX ausente (JMX agent)
  - Não é possível ler o arquivo de trap (SNMP trap)
  - Script não encontrado (Verificação externa)
  - URL inexistente (HTTP agent, Browser)

- Falha no login (SSH agent, TELNET agent)
- Sintaxe de fórmula inválida (Calculado), erro de sintaxe JavaScript (Script) ou SQL inválido (Monitor de banco de dados)

Para detectar e lidar com erros na coleta de valores de itens, você pode usar a etapa de pré-processamento *Verificar valor não suportado*. Observe que esta etapa é sempre executada primeiro e detecta apenas erros que ocorrem antes do início do pré-processamento.

Na guia *Pré-processamento* de um item, selecione a etapa de pré-processamento *Verificar valor não suportado* e especifique um dos seguintes parâmetros:

- Para quaisquer erros:

```
Parâmetro: any error
```

- Para erros contendo ”cannot connect”:

```
Parâmetro: error matches
Padrão: (?i)cannot connect
```

Em seguida, use a opção *Personalizar em caso de falha* para descartar o valor (neste caso, o erro), definir um valor personalizado ou retornar uma mensagem de erro personalizada. Observe que valores descartados não são armazenados no banco de dados; como resultado, triggers não são avaliados e dados de tendência não são gerados.

4 Funcionalidade JSONPath

Visão geral

Esta seção descreve a funcionalidade JSONPath suportada nas etapas de pré-processamento de valores de item.

JSONPath é composto por segmentos separados por pontos. Um segmento pode assumir a forma de uma palavra simples, representando um nome de valor JSON, o caractere curinga (*) ou uma construção mais complexa entre colchetes. O ponto antes de um segmento entre colchetes é opcional e pode ser omitido.

Exemplo de JSONPath	Descrição
`$.object.name`	Retorna o conteúdo de `object.name`.
`$.object['name']`	Retorna o conteúdo de `object.name`.
`$.object.['name']`	Retorna o conteúdo de `object.name`.
`$["object"]['name']`	Retorna o conteúdo de `object.name`.
`$.['object'].["name"]`	Retorna o conteúdo de `object.name`.
`$.object.history.length()`	Retorna o número de elementos do array `object.history`.
`$[?(@.name == 'Object')].price.first()`	Retorna o valor da propriedade `price` do primeiro objeto chamado ”Object”.
`$[?(@.name == 'Object')].history.first().length()`	Retorna o número de elementos do array history do primeiro objeto chamado ”Object”.
`$[?(@.price > 10)].length()`	Retorna o número de objetos com preço maior que 10.

Veja também: Escapando caracteres especiais de valores de macros LLD em JSONPath.

Segmentos suportados

Segmento	Descrição
`<name>`	Corresponde à propriedade do objeto pelo nome.
`*`	Corresponde a todas as propriedades do objeto.
`['<name>']`	Corresponde à propriedade do objeto pelo nome.
`['<name>', '<name>', ...]`	Corresponde à propriedade do objeto por qualquer um dos nomes listados.
`[<index>]`	Corresponde ao elemento do array pelo índice.
`[<number>, <number>, ...]`	Corresponde ao elemento do array por qualquer um dos índices listados.
`[*]`	Corresponde a todas as propriedades do objeto ou elementos do array.

Segmento	Descrição
`[<start>:<end>]`	Corresponde aos elementos do array pelo intervalo definido: **<start>** - o primeiro índice a ser correspondido (inclusivo); se não especificado, corresponde a todos os elementos do array desde o início; se negativo, especifica o deslocamento inicial a partir do final do array; **<end>** - o último índice a ser correspondido (exclusivo); se não especificado, corresponde a todos os elementos do array até o final; se negativo, especifica o deslocamento inicial a partir do final do array.
`[?(<expression>)]`	Corresponde a objetos/elementos do array aplicando uma expressão de filtro.

Para encontrar um segmento correspondente ignorando sua ancestralidade (segmento destacado), ele deve ser prefixado com dois pontos (`..`). Por exemplo, `$..name` ou `$..['name']` retornam valores de todas as propriedades `name`.

Os nomes dos elementos correspondentes podem ser extraídos adicionando um sufixo til (~) ao JSONPath. Ele retorna o nome do objeto correspondente ou um índice em formato de string do item do array correspondente. O formato de saída segue as mesmas regras de outras consultas JSONPath - resultados de caminho definido são retornados 'como estão', e resultados de caminho indefinido são retornados em um array. No entanto, há pouco valor em extrair o nome de um elemento que corresponde a um caminho definitivo, pois ele já é conhecido.

Expressão de filtro

A expressão de filtro é uma expressão aritmética em notação infixa.

Operandos suportados:

Operando	Descrição

`"<texto>"`
`'<texto>'`
	Constante de texto.
	
	Exemplo:
	`'value: \\'1\\''`
	`"value: '1'"`

`<número>`	Constante numérica com suporte à notação científica. Exemplo: 123
`<jsonpath começando com $>`	Valor referenciado pelo JSONPath a partir do nó raiz do documento de entrada; apenas caminhos definidos são suportados. Exemplo: `$.object.name`
`<jsonpath começando com @>`	Valor referenciado pelo JSONPath a partir do objeto/elemento atual; apenas caminhos definidos são suportados. Exemplo: `@.name`

Operadores suportados:

Operador	Tipo	Descrição	Resultado
-	Binário	Subtração	Número
+	Binário	Adição	Número
/	Binário	Divisão	Número
*	Binário	Multiplicação	Número
==	Binário	Igualdade	Booleano (1/0)
!=	Binário	Desigualdade	Booleano (1/0)
	Binário	Menor que	Booleano (1/0)
<=	Binário	Menor ou igual a	Booleano (1/0)
>	Binário	Maior que	Booleano (1/0)
>=	Binário	Maior ou igual a	Booleano (1/0)
=~	Binário	Corresponde à expressão regular	Booleano (1/0)
!	Unário	NÃO booleano	Booleano (1/0)
||	Binário	OU booleano	Booleano (1/0)
&&	Binário	E booleano	Booleano (1/0)

Funções

Funções podem ser usadas no final do JSONPath. Múltiplas funções podem ser encadeadas se a função anterior retornar um valor que seja aceito pela função seguinte.

Funções suportadas:

Função	Descrição	Entrada	Saída
`avg`	Valor médio dos números em um array de entrada	Array de números	Número
`min`	Valor mínimo dos números em um array de entrada	Array de números	Número
`max`	Valor máximo dos números em um array de entrada	Array de números	Número
`sum`	Soma dos números em um array de entrada	Array de números	Número
`length`	Número de elementos em um array de entrada	Array	Número
`first`	O primeiro elemento de um array	Array	Uma construção JSON (objeto, array, valor) dependendo do conteúdo do array de entrada

Funções de agregação JSONPath aceitam valores numéricos entre aspas. Esses valores são automaticamente convertidos de strings para tipos numéricos quando a agregação é necessária. Entradas incompatíveis farão com que a função gere um erro.

Valor de saída

Os JSONPaths podem ser divididos em caminhos definidos e indefinidos. Um caminho definido pode retornar apenas nulo ou uma única correspondência. Um caminho indefinido pode retornar várias correspondências: JSONPaths com listas de nomes/índices destacadas, fatias de array ou segmentos de expressão. No entanto, quando uma função é usada, o JSONPath se torna definido, pois as funções sempre retornam um único valor.

Um caminho definido retorna o objeto/array/valor ao qual faz referência. Em contraste, um caminho indefinido retorna um array dos objetos/arrays/valores correspondentes.

> **Attention:**
> A ordem das propriedades nos resultados da consulta JSONPath pode não corresponder à ordem original das propriedades do JSON devido a métodos internos de otimização. Por exemplo, o JSONPath `$.books[1]["author", "title"]` pode retornar `["title", "author"]`. Se for essencial preservar a ordem original das propriedades, métodos alternativos de pós-processamento da consulta devem ser considerados.

Regras de formatação de caminho

Espaços em branco (espaço, caractere de tabulação) podem ser usados em segmentos de notação de colchetes e expressões, por exemplo: `$[ 'a' ][ 0 ][ ?( $.b == 'c' ) ][ : -1 ].first( )`.

Strings devem ser delimitadas por aspas simples (`'`) ou duplas (`"`). Dentro das strings, aspas simples ou duplas (dependendo de quais são usadas para delimitá-las) e barras invertidas (`\`) são escapadas com o caractere barra invertida (`\`).

Exemplo

```
{
  "books": [
    {
      "category": "reference",
      "author": "Nigel Rees",
      "title": "Sayings of the Century",
      "price": 8.95,
      "id": 1
    },
    {
      "category": "fiction",
      "author": "Evelyn Waugh",
      "title": "Sword of Honour",
      "price": 12.99,
      "id": 2
    },
```

```
    {
      "category": "fiction",
      "author": "Herman Melville",
      "title": "Moby Dick",
      "isbn": "0-553-21311-3",
      "price": 8.99,
      "id": 3
    },
    {
      "category": "fiction",
      "author": "J. R. R. Tolkien",
      "title": "The Lord of the Rings",
      "isbn": "0-395-19395-8",
      "price": 22.99,
      "id": 4
    }
  ],
  "services": {
    "delivery": {
      "servicegroup": 1000,
      "description": "Entrega no dia seguinte na cidade local",
      "active": true,
      "price": 5
    },
    "bookbinding": {
      "servicegroup": 1001,
      "description": "Impressão e montagem do livro no formato A5",
      "active": true,
      "price": 154.99
    },
    "restoration": {
      "servicegroup": 1002,
      "description": "Vários métodos de restauração",
      "active": false,
      "methods": [
        {
          "description": "Limpeza química",
          "price": 46
        },
        {
          "description": "Prensagem de páginas danificadas por umidade",
          "price": 24.5
        },
        {
          "description": "Reencadernação de livro rasgado",
          "price": 99.49
        }
      ]
    }
  },
  "filters": {
    "price": 10,
    "category": "fiction",
    "no filters": "no \"filters\""
  },
  "closed message": "A loja está fechada",
  "tags": [
    "a",
    "b",
    "c",
    "d",
    "e"
```

```
  ]
}
```

JSONPath	Tipo	Resultado
`$.filters.price`	definido	10
`$.filters.category`	definido	fiction
`$.filters['no filters']`	definido	no ”filters”
`$.filters`	definido	{ ”price”: 10, ”category”: ”fiction”, ”no filters”: ”no \”filters\”” }
`$.books[1].title`	definido	Sword of Honour
`$.books[-1].author`	definido	J. R. R. Tolkien
`$.books.length()`	definido	4
`$.tags[:]`	indefinido	[”a”, ”b”, ”c”, ”d”, ”e”]
`$.tags[2:]`	indefinido	[”c”, ”d”, ”e”]
`$.tags[:3]`	indefinido	[”a”, ”b”, ”c”]
`$.tags[1:4]`	indefinido	[”b”, ”c”, ”d”]
`$.tags[-2:]`	indefinido	[”d”, ”e”]
`$.tags[:-3]`	indefinido	[”a”, ”b”]
`$.tags[:-3].length()`	definido	2
`$.books[0, 2].title`	indefinido	[”Moby Dick”, ”Sayings of the Century”]
`$.books[1]['author', "title"]`	indefinido	[”Sword of Honour”, ”Evelyn Waugh”]
`$..id`	indefinido	[1, 2, 3, 4]
`$.services..price`	indefinido	[154.99, 5, 46, 24.5, 99.49]
`$.books[?(@.id == 4 - 0.4 * 5)].title`	indefinido	[”Sword of Honour”] Nota: Esta consulta mostra que operações aritméticas podem ser usadas em consultas; pode ser simplificada para `$.books[?(@.id == 2)].title`
`$.books[?(@.id == 2 || @.id == 4)].title`	indefinido	[”Sword of Honour”, ”The Lord of the Rings”]
`$.books[?(!(@.id == 2))].title`	indefinido	[”Sayings of the Century”, ”Moby Dick”, ”The Lord of the Rings”]
`$.books[?(@.id != 2)].title`	indefinido	[”Sayings of the Century”, ”Moby Dick”, ”The Lord of the Rings”]
`$.books[?(@.title =~ " of ")].title`	indefinido	[”Sayings of the Century”, ”Sword of Honour”, ”The Lord of the Rings”]
`$.books[?(@.price > 12.99)].title`	indefinido	[”The Lord of the Rings”]
`$.books[?(@.author > "Herman Melville")].title`	indefinido	[”Sayings of the Century”, ”The Lord of the Rings”]
`$.books[?(@.price > $.filters.price)].title`	indefinido	[”Sword of Honour”, ”The Lord of the Rings”]
`$.books[?(@.category == $.filters.category)].title`	indefinido	[”Sword of Honour”,”Moby Dick”,”The Lord of the Rings”]
`$.books[?(@.category == "fiction" && @.price < 10)].title`	indefinido	[”Moby Dick”]

JSONPath	Tipo	Resultado

`$..[?(@.id)]`	indefinido	[
		{
		”price”: 8.95,
		”id”: 1,
		”category”: ”reference”,
		”author”: ”Nigel Rees”,
		”title”: ”Sayings of the Century”
		},
		{
		”price”: 12.99,
		”id”: 2,
		”category”: ”fiction”,
		”author”: ”Evelyn Waugh”,
		”title”: ”Sword of Honour”
		},
		{
		”price”: 8.99,
		”id”: 3,
		”category”: ”fiction”,
		”author”: ”Herman Melville”,
		”title”: ”Moby Dick”,
		”isbn”: ”0-553-21311-3”
		},
		{
		”price”: 22.99,
		”id”: 4,
		”category”: ”fiction”,
		”author”: ”J. R. R. Tolkien”,
		”title”: ”The Lord of the Rings”,
		”isbn”: ”0-395-19395-8”
		}
		]

`$.services..[?(@.price > 50)].description`	indefinido	[”Impressão e montagem do livro no formato A5”, ”Reencadernação de livro rasgado”]
`$..id.length()`	definido	4
`$.books[?(@.id == 2)].title.first()`	definido	Sword of Honour
`$..tags.first().length()`	definido	5 Nota: `$..tags` é um caminho indefinido, então retorna um array de elementos correspondentes, ou seja, `[["a", "b", "c", "d", "e"]]`; `first()` retorna o primeiro elemento, ou seja, `["a", "b", "c", "d", "e"]`; `length()` calcula o comprimento do elemento, ou seja, 5.
`$.books[*].price.min()`	definido	8.95
`$..price.max()`	definido	154.99
`$.books[?(@.category == "fiction")].price.avg()`	definido	14.99
`$.books[?(@.category == $.filters.xyz)].title`	indefinido	Nota: Uma consulta sem correspondência retorna NULL para caminhos definidos e indefinidos.
`$.services[?(@.active=="true")].servicegroup`	indefinido	[1001,1000] Nota: Constantes de texto devem ser usadas em comparações de valores booleanos.
`$.services[?(@.active=="false")].servicegroup`	indefinido	[1002] Nota: Constantes de texto devem ser usadas em comparações de valores booleanos.
`$.services[?(@.servicegroup=='1002')]~.first()`	definido	location

1 Escape de caracteres especiais de valores de macro LLD em JSONPath

Quando macros de descoberta de baixo nível são usadas no pré-processamento de JSONPath e seus valores são resolvidos, as seguintes regras de escape de caracteres especiais são aplicadas:

- apenas os caracteres barra invertida (\) e aspas duplas (”) são considerados para escape;
- se o valor da macro resolvida contiver esses caracteres, cada um deles é escapado com uma barra invertida;
- se eles já estiverem escapados com uma barra invertida, isso não é considerado como escape e tanto a barra invertida quanto os caracteres especiais seguintes são escapados novamente.

Por exemplo:

JSONPath	Valor da macro LLD	Após substituição
$.[?(@.value == ”{#MACRO}”)]	special ”value”	$.[?(@.value == ”special \”value\””)]
	c:\temp	$.[?(@.value == ”c:\\temp”)]
	a\\b	$.[?(@.value == ”a\\\\b”)]

Quando usada na expressão, a macro que pode ter caracteres especiais deve ser colocada entre aspas duplas:

JSONPath	Valor da macro LLD	Após substituição	Resultado
$.[?(@.value == ”{#MACRO}”)]	special ”value”	$.[?(@.value == ”special \”value\””)]	OK
$.[?(@.value == {#MACRO})]		$.[?(@.value == special \”value\”)]	**Expressão JSONPath inválida**

Quando usada no caminho, a macro que pode ter caracteres especiais deve ser colocada entre colchetes **e** aspas duplas:

JSONPath	Valor da macro LLD	Após substituição	Resultado
$.[”{#MACRO}”].value	c:\temp	$.[”c:\\temp”].value	OK
$.{#MACRO}.value		$.c:\\temp.value	**Expressão JSONPath inválida**

5 Pré-processamento JavaScript

Visão geral

Esta seção fornece detalhes do pré-processamento por JavaScript.

> **Attention:**
> Não use atribuições não declaradas no JavaScript de pré-processamento. Use `var` para declarar variáveis locais.

Pré-processamento JavaScript

O pré-processamento JavaScript é feito invocando uma função JavaScript com um único parâmetro ’value’ e o corpo da função fornecido pelo usuário. O resultado da etapa de pré-processamento é o valor retornado por essa função, por exemplo, para realizar a conversão de Fahrenheit para Celsius, insira:

```
return (value - 32)  * 5 / 9
```

nos parâmetros de pré-processamento JavaScript, que serão encapsulados em uma função JavaScript pelo server:

```
function (value)
{
   return (value - 32) * 5 / 9
}
```

O parâmetro de entrada ’value’ é sempre passado como uma string. O valor de retorno é automaticamente convertido para string via método toString() (se falhar, o erro é retornado como valor de string), com algumas exceções:

- retornar o valor undefined resultará em erro;
- retornar o valor null fará com que o valor de entrada seja descartado, assim como o pré-processamento ’Descartar valor’ na ação ’Personalizado em caso de falha’.

Erros podem ser retornados lançando valores/objetos (normalmente strings ou objetos Error).

Por exemplo:

```
if (value == 0)
    throw "Zero input value"
return 1/value
```

Cada script tem um tempo limite de execução de 10 segundos (dependendo do script, pode levar mais tempo para o tempo limite ser acionado); excedê-lo retornará erro. Um limite de heap de 512 megabytes é imposto.

O bytecode da etapa de pré-processamento JavaScript é armazenado em cache e reutilizado quando a etapa é aplicada na próxima vez. Quaisquer alterações nas etapas de pré-processamento do item farão com que o script em cache seja redefinido e recompilado posteriormente.

Falhas consecutivas em tempo de execução (3 seguidas) farão com que o mecanismo seja reinicializado para mitigar a possibilidade de um script quebrar o ambiente de execução para os próximos scripts (essa ação é registrada com DebugLevel 4 ou superior).

O pré-processamento JavaScript é implementado com o mecanismo JavaScript Duktape.

Veja também: Objetos JavaScript adicionais e funções globais

Usando macros em scripts

É possível usar macros de usuário (assim como macros LLD no contexto de descoberta de baixo nível) em código JavaScript. Se um script contiver macros, essas macros são resolvidas pelo server/proxy antes de executar etapas específicas de pré-processamento. Observe que, ao testar etapas de pré-processamento no frontend, os valores das macros não serão buscados e precisarão ser inseridos manualmente.

> **Note:**
> O contexto é ignorado quando uma macro é substituída por seu valor. O valor da macro é inserido no código como está, não é possível adicionar escape adicional antes de colocar o valor no código JavaScript. Por favor, esteja ciente de que isso pode causar erros de JavaScript em alguns casos.

No exemplo abaixo, se o valor recebido exceder o valor da macro {$THRESHOLD}, o valor do limite (se presente) será retornado:

```
var threshold = '{$THRESHOLD}';
return (!isNaN(threshold) && value > threshold) ? threshold : value;
```

Exemplos

Os exemplos a seguir ilustram como você pode usar o pré-processamento em JavaScript.

Cada exemplo contém uma breve descrição, um corpo de função para os parâmetros de pré-processamento em JavaScript e o resultado da etapa de pré-processamento - valor retornado pela função.

Exemplo 1: Converter um número (notação científica para inteiro)

Converta o número ”2.62128e+07” da notação científica para um inteiro.

```
return (Number(value))
```

Valor retornado pela função: 26212800.

Exemplo 2: Converter um número (binário para decimal)

Converta o número binário ”11010010” para um número decimal.

```
return(parseInt(value,2))
```

Valor retornado pela função: 210.

Exemplo 3: Arredondar um número

Arredonde o número ”18.2345” para 2 dígitos.

```
return(Math.round(value* 100) / 100)
```

Valor retornado pela função: 18.23.

Exemplo 4: Contar letras em uma string

Conta o número de letras na string ”Zabbix”.

```
return (value.length)
```

Valor retornado pela função: 6.

Exemplo 5: Obter o tempo restante

Obtenha o tempo restante (em segundos) até a data de expiração de um certificado (12 de fevereiro de 2022 12:33:56 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);
```

Valor retornado pela função: 44380233.

Exemplo 6: Remover propriedades JSON

Modifique a estrutura de dados JSON removendo quaisquer propriedades com a chave "data_size" ou "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)
```

Valor aceito pela função:

```
[
    {
        "table_name":"history",
        "data_size":"326.05",
        "index_size":"174.34"
    },
    {
        "table_name":"history_log",
        "data_size":"6.02",
        "index_size":"3.45"
    }
]
```

Valor retornado pela função:

```
[
    {
        "table_name":"history"
    },
    {
        "table_name":"history_log"
    }
]
```

Exemplo 7: Converter status do Apache para JSON

Converta o valor recebido de um item do agent Zabbix web.page.get (por exemplo, web.page.get[http://127.0.0.1:80/server-status?auto]) para um objeto JSON.

```
// Converter status do Apache para JSON

// Divide o valor em substrings e coloca essas substrings em um array
var lines = value.split('\n');

// Cria um objeto vazio "output"
var output = {};

// Cria um objeto "workers" com propriedades predefinidas
```

```
var workers = {
    '_': 0, 'S': 0, 'R': 0, 'W': 0,
    'K': 0, 'D': 0, 'C': 0, 'L': 0,
    'G': 0, 'I': 0, '.': 0
};

// Adiciona as substrings do array "lines" ao objeto "output" como propriedades (pares chave-valor)
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]);
    }
}

// Métricas multiversão
output.ServerUptimeSeconds = output.ServerUptimeSeconds || output.Uptime;
output.ServerVersion = output.ServerVersion || output.Server;

// Analisa a propriedade "Scoreboard" para obter a contagem de workers
if (typeof output.Scoreboard === 'string') {
    for (var i = 0; i < output.Scoreboard.length; i++) {
        var char = output.Scoreboard[i];

        workers[char]++;
    }
}

// Adiciona dados de worker ao objeto "output"
output.Workers = {
    waiting: workers['_'], starting: workers['S'], reading: workers['R'],
    sending: workers['W'], keepalive: workers['K'], dnslookup: workers['D'],
    closing: workers['C'], logging: workers['L'], finishing: workers['G'],
    cleanup: workers['I'], slot: workers['.']
};

// Retorna string JSON
return JSON.stringify(output);
```

Valor aceito pela função:

```
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__.....................................................................................
```

Valor retornado pela função:


```
{
    "Date": "Mon, 27 Mar 2023 11:08:39 GMT",
    "Server": "Apache/2.4.52 (Ubuntu)",
    "Vary": "Accept-Encoding",
    "Encoding": "gzip",
    "Length": 405,
    "Type": "text/plain; charset=ISO-8859-1",
    "ServerVersion": "Apache/2.4.52 (Ubuntu)",
    "ServerMPM": "prefork",
    "Server Built": "2023-03-08T17:32:01",
    "CurrentTime": "Monday, 27-Mar-2023 14:08:39 EEST",
    "RestartTime": "Monday, 27-Mar-2023 12:19:59 EEST",
    "ParentServerConfigGeneration": 1,
    "ParentServerMPMGeneration": 0,
    "ServerUptimeSeconds": 6520,
    "ServerUptime": "1 hour 48 minutes 40 seconds",
    "Load1": 0.56,
    "Load5": 0.33,
    "Load15": 0.28,
    "Total Accesses": 2476,
    "Total kBytes": 8370,
    "Total Duration": 52718,
    "CPUUser": 8.16,
    "CPUSystem": 3.44,
    "CPUChildrenUser": 0,
    "CPUChildrenSystem": 0,
    "CPULoad": 0.177914,
    "Uptime": 6520,
    "ReqPerSec": 0.379755,
    "BytesPerSec": 1314.55,
    "BytesPerReq": 3461.58,
    "DurationPerReq": 21.2916,
    "BusyWorkers": 2,
    "IdleWorkers": 6,
    "Scoreboard": "____KW__.....................................................................................
    "Workers": {
        "waiting": 6,
        "starting": 0,
        "reading": 0,
        "sending": 1,
        "keepalive": 1,
        "dnslookup": 0,
        "closing": 0,
        "logging": 0,
        "finishing": 0,
```

```
        "cleanup": 0,
        "slot": 142
    }
}
```

1 Objetos JavaScript adicionais

Visão geral

Esta seção descreve as adições do Zabbix à linguagem JavaScript implementada com Duktape e as funções globais do JavaScript suportadas.

Attention:
Não use atribuições não declaradas no pré-processamento do JavaScript. Use `var` para declarar variáveis locais.

Objetos internos

Zabbix

O objeto Zabbix fornece interação com a funcionalidade interna do Zabbix.

Método	Descrição
`log(loglevel, message)`	Grava <message> no log do Zabbix usando o nível de log <loglevel> (veja o parâmetro DebugLevel no arquivo de configuração).

Exemplo:

```
Zabbix.log(3, "esta é uma entrada de log gravada com o nível de log 'Warning'")
```

Você pode usar os seguintes aliases:

Alias	Alias para
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:
O tamanho total de todas as mensagens registradas é limitado a 8 MB por execução de script.

Método	Descrição
`sleep(delay)`	Atrasar a execução do JavaScript por `delay` milissegundos.

Exemplo (atrasar a execução por 15 segundos):

```
Zabbix.sleep(15000)
```

HttpRequest

Este objeto encapsula o manipulador cURL permitindo fazer requisições HTTP simples. Erros são lançados como exceções.

Attention:
A inicialização de múltiplos objetos `HttpRequest` é limitada a 10 por execução de script.

Método	Descrição
`addHeader(value)`	Adiciona um campo de cabeçalho HTTP. Este campo é usado para todas as requisições seguintes até ser limpo com o método `clearHeader()`. O comprimento total dos campos de cabeçalho que podem ser adicionados a um único objeto `HttpRequest` é limitado a 128 Kbytes (caracteres especiais e nomes de cabeçalho incluídos).
`clearHeader()`	Limpa o cabeçalho HTTP. Se nenhum campo de cabeçalho estiver definido, o `HttpRequest` definirá Content-Type como application/json se os dados enviados estiverem no formato JSON; text/plain caso contrário.
`connect(url)`	Envia uma requisição HTTP CONNECT para a URL e retorna a resposta.
`customRequest(method, url, data)`	Permite especificar qualquer método HTTP no primeiro parâmetro. Envia a requisição do método para a URL com o *data* opcional e retorna a resposta.
`delete(url, data)`	Envia uma requisição HTTP DELETE para a URL com o *data* opcional e retorna a resposta.
`getHeaders(<asArray>)`	Retorna o objeto dos campos de cabeçalho HTTP recebidos. O parâmetro `asArray` pode ser definido como "true" (por exemplo, `getHeaders(true)`), "false" ou indefinido. Se definido como "true", os valores dos campos de cabeçalho HTTP recebidos serão retornados como arrays; isso deve ser usado para recuperar os valores dos campos de múltiplos cabeçalhos com o mesmo nome. Se não definido ou definido como "false", os valores dos campos de cabeçalho HTTP recebidos serão retornados como strings.
`get(url, data)`	Envia uma requisição HTTP GET para a URL com o *data* opcional e retorna a resposta.
`head(url)`	Envia uma requisição HTTP HEAD para a URL e retorna a resposta.
`options(url)`	Envia uma requisição HTTP OPTIONS para a URL e retorna a resposta.
`patch(url, data)`	Envia uma requisição HTTP PATCH para a URL com o *data* opcional e retorna a resposta.
`put(url, data)`	Envia uma requisição HTTP PUT para a URL com o *data* opcional e retorna a resposta.
`post(url, data)`	Envia uma requisição HTTP POST para a URL com o *data* opcional e retorna a resposta.
`getStatus()`	Retorna o código de status da última requisição HTTP.
`setProxy(proxy)`	Define o proxy HTTP para o valor "proxy". Se este parâmetro estiver vazio, nenhum proxy será usado.

`setHttpAuth(bitmask, username, password)`	Define os métodos de autenticação HTTP habilitados (HTTPAUTH_BASIC, HTTPAUTH_DIGEST, HTTPAUTH_NEGOTIATE, HTTPAUTH_NTLM, HTTPAUTH_NONE) no parâmetro 'bitmask'.
	A flag HTTPAUTH_NONE permite desabilitar a autenticação HTTP.
	Exemplos:
	`request.setHttpAuth(HTTPAUTH_NTLM | HTTPAUTH_BASIC, username, password)`
	`request.setHttpAuth(HTTPAUTH_NONE)`

`trace(url, data)`	Envia uma requisição HTTP TRACE para a URL com o *data* opcional e retorna a resposta.

Exemplo:

```
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

O objeto XML permite o processamento de dados XML no pré-processamento de item e descoberta de baixo nível e webhooks.

> **Attention:**
> Para usar o objeto XML, o server/proxy deve ser compilado com suporte a libxml2.

Método	Descrição
`XML.query(data, expression)`	Recupera o conteúdo do nó usando XPath. Retorna null se o nó não for encontrado. **expression** - uma expressão XPath; **data** - dados XML como uma string.
`XML.toJson(data)`	Converte dados no formato XML para JSON.
`XML.fromJson(object)`	Converte dados no formato JSON para XML.

Exemplo:

Entrada:

```
<menu>
    <food type = "breakfast">
        <name>Chocolate</name>
        <price>$5.95</price>
        <description></description>
        <calories>650</calories>
    </food>
</menu>
```

Saída:

```
{
    "menu": {
        "food": {
            "@type": "breakfast",
            "name": "Chocolate",
            "price": "$5.95",
            "description": null,
            "calories": "650"
        }
    }
}
```

Regras de serialização

A conversão de XML para JSON será processada de acordo com as seguintes regras (para conversões de JSON para XML, as regras reversas são aplicadas):

1. Os atributos XML serão convertidos em chaves que têm seus nomes precedidos por '@'.

Exemplo:

Entrada:

```
<xml foo="FOO">
  <bar>
    <baz>BAZ</baz>
  </bar>
</xml>
```

Saída:

```
{
  "xml": {
    "@foo": "FOO",
    "bar": {
      "baz": "BAZ"
    }
  }
}
```

2. Elementos autocontidos (<foo/>) serão convertidos como tendo valor 'null'.

Exemplo:

Entrada:

```
<xml>
  <foo/>
</xml>
```

Saída:

```
{
  "xml": {
    "foo": null
  }
}
```

3. Atributos vazios (com valor "") serão convertidos como tendo valor de string vazia ('').

Exemplo:

Entrada:

```
<xml>
  <foo bar="" />
</xml>
```

Saída:

```
{
  "xml": {
    "foo": {
      "@bar": ""
    }
  }
}
```

4. Múltiplos nós filhos com o mesmo nome de elemento serão convertidos em uma única chave que tem um array de valores como seu valor.

Exemplo:

Entrada:

```
<xml>
  <foo>BAR</foo>
  <foo>BAZ</foo>
  <foo>QUX</foo>
</xml>
```

Saída:

```
{
  "xml": {
    "foo": ["BAR", "BAZ", "QUX"]
  }
}
```

5. Se um elemento de texto não tiver atributos nem filhos, ele será convertido como uma string.

Exemplo:

Entrada:

```
<xml>
    <foo>BAZ</foo>
</xml>
```

Saída:

```
{
  "xml": {
    "foo": "BAZ"
   }
}
```

6. Se um elemento de texto não tiver filhos, mas tiver atributos, o conteúdo do texto será convertido em um elemento com a chave '#text' e o conteúdo como valor; os atributos serão convertidos conforme descrito na regra de serialização 1.

Exemplo:

Entrada:

```
<xml>
  <foo bar="BAR">
    BAZ
  </foo>
</xml>
```

Saída:

```
{
  "xml": {
    "foo": {
      "@bar": "BAR",
      "#text": "BAZ"
    }
  }
}
```

Funções globais do JavaScript

Funções globais adicionais do JavaScript foram implementadas com o Duktape:

- `btoa(data)` - codifica os dados em uma string Base64.
- `atob(base64_string)` - desde a versão 7.0.1, decodifica a string Base64 como um buffer Uint8Array (no Zabbix 7.0.0 esta função retorna uma string decodificada).

```
try {
    b64 = btoa("test string");
    buffer = atob(b64);

    // Observe que a lógica de decodificação depende do formato dos dados do buffer.
    decoded = String.fromCharCode.apply(this, [].slice.call(buffer));
}
catch (error) {
    return {'error.name' : error.name, 'error.message' : error.message};
}
```

- `md5(data)` - calcula o hash MD5 dos dados.
- `sha256(data)` - calcula o hash SHA256 dos dados.

- `hmac('<hash type>',key,data)` - retorna um hash HMAC como uma string em formato hexadecimal; `md5` e `sha256` são suportados como `hash type`; os parâmetros `key` e `data` suportam dados binários.

  Exemplos:

  - `hmac('md5',key,data)`
  - `hmac('sha256',key,data)`

- `sign(hash,key,data)` - retorna a assinatura calculada (assinatura RSA com SHA-256) como uma string, onde:
 hash - apenas `sha256` é permitido, caso contrário, um erro será lançado.
 key - a chave privada. Deve estar em conformidade com o padrão PKCS#1 ou PKCS#8. A chave pode ser fornecida em diferentes formas:

  - com espaços em vez de quebras de linha
  - com `\n` escapado ou não escapado em vez de quebras de linha
  - sem nenhuma quebra de linha como uma string de linha única
  - como uma string formatada em JSON

  A chave também pode ser carregada de uma macro de usuário/macro secreta/vault.

  data - os dados que serão assinados. Pode ser uma string (dados binários também são suportados) ou buffer (Uint8Array/ArrayBuffer).

  Exemplo:

  - `sign('sha256',key,data)`

  OpenSSL ou GnuTLS é usado para calcular as assinaturas. Se o Zabbix foi compilado sem nenhuma dessas bibliotecas de criptografia, um erro será lançado ('missing OpenSSL or GnuTLS library').

2 Objetos JavaScript de item de navegador

Visão geral

Esta seção descreve as adições do Zabbix à linguagem JavaScript implementada com Duktape para uso no script do item Browser. Essas adições complementam os objetos JavaScript descritos na página Objetos JavaScript adicionais.

Browser

O objeto `Browser` gerencia sessões do WebDriver, inicializando uma sessão ao ser criado e encerrando-a ao ser destruído. Um único script pode suportar até quatro objetos `Browser`.

Para construir um objeto `Browser`, use a sintaxe `new Browser(options)`. O parâmetro `options` (*objeto JSON*) especifica as opções do navegador, geralmente o resultado do método de opções do WebDriver (por exemplo, `Browser.chromeOptions()`).

Os seguintes métodos são suportados com o objeto `Browser`.

Método	Descrição
`navigate(url)`	Navega para a URL especificada. Parâmetros: `url` - (string) URL para navegar.
`getUrl()`	Retorna uma string da URL da página aberta.
`getPageSource()`	Retorna uma string do código-fonte da página aberta.
`findElement(strategy, selector)`	Retorna um objeto `Element` com um elemento na página aberta (ou retorna `null` se nenhum elemento corresponder a `strategy` e `selector`). Parâmetros: `strategy` - (string, seletor css/texto do link/texto parcial do link/nome da tag/xpath) Estratégia de localização; `selector` - (string) Seletor do elemento usando a estratégia de localização especificada.

Método	Descrição
`findElements(strategy, target)`	Retorna um array de objetos `Element` com vários elementos na página aberta (ou retorna um array vazio se nenhum elemento corresponder à estratégia de localização e ao alvo). Parâmetros: `strategy` - (string, seletor css/texto do link/texto parcial do link/nome da tag/xpath) Estratégia de localização; `target` - (string) Seletor do elemento usando a estratégia de localização especificada.
`getCookies()`	Retorna um array de objetos `Cookie`.
`addCookie(cookie)`	Define cookies. Parâmetros: `cookie` - (objeto `Cookie`) Cookie a ser definido.
`getScreenshot()`	Retorna uma string codificada em Base64 representando uma imagem da viewport do navegador.
`setScreenSize(x,y)`	Define o tamanho da viewport do navegador. Parâmetros: `x` - (string) Largura em pixels; `y` - (string) Altura em pixels.
`setScriptTimeout(timeout)`	Define o tempo limite de carregamento do script. Parâmetros: `timeout` - (inteiro) Valor do tempo limite em milissegundos.
`setSessionTimeout(timeout)`	Define o tempo limite da sessão (carregamento da página). Parâmetros: `timeout` - (inteiro) Valor do tempo limite em milissegundos.
`setElementWaitTimeout(timeout)`	Define o tempo limite da estratégia de localização do elemento (implícito). Parâmetros: `timeout` - (inteiro) Valor do tempo limite em milissegundos.
`collectPerfEntries(mark)`	Coleta entradas de desempenho para recuperar com o método `getResult()`. Parâmetros: `mark` - (string, opcional) Marca do snapshot de desempenho.
`getRawPerfEntries()`	Retorna um array de objetos de entrada de desempenho.
`getResult()`	Retorna um objeto `Result` com estatísticas da sessão do navegador (informações de erro, snapshots de desempenho, etc.).
`getError()`	Retorna um objeto `BrowserError` com erros do navegador (ou retorna `null` se não houver erros do navegador).
`setError(message)`	Define uma mensagem de erro personalizada a ser incluída no objeto `Result`. Parâmetros: `message` - (string) Mensagem de erro.
`discardError()`	Descarta o erro a ser retornado no objeto `Result`.
`getAlert()`	Retorna um objeto `Alert` com alertas do navegador (ou retorna `null` se não houver alertas do navegador).
`chromeOptions()`	Retorna um objeto `chromeOptions` com opções predefinidas do navegador Chrome.
`firefoxOptions()`	Retorna um objeto `firefoxOptions` com opções predefinidas do navegador Firefox.
`safariOptions()`	Retorna um objeto `safariOptions` com opções predefinidas do navegador Safari.
`edgeOptions()`	Retorna um objeto `edgeOptions` com opções predefinidas do navegador Edge.
`switchFrame(target)`	Alterna para o frame especificado. Suportado desde a versão 7.0.4. Parâmetros: `target` - (elemento do navegador ou inteiro, opcional) Frame de destino. Para selecionar um frame por elemento, passe o elemento. Para selecionar um frame por índice, passe o número. Se deixado em branco, alternará para o contexto de navegação de nível superior.

Todos os métodos do Browser podem lançar os seguintes erros:

- `BrowserError` - derivado do objeto `Error` que é lançado se o construtor do `Browser` falhar; contém uma propriedade adicional `browser` com um objeto `Browser` que lançou este `BrowserError`.
- `WebdriverError` - derivado de `BrowserError`; contém as mesmas propriedades do objeto `BrowserError`, que indicam se o erro foi gerado em resposta a um erro na resposta do WebDriver.

Elemento

O objeto `Element` é retornado pelos métodos `findElement()`/`findElements()` do objeto `Browser` e não pode ser construído diretamente.

O objeto `Element` representa um elemento na página da web e fornece métodos para interagir com ele.

Os seguintes métodos são suportados com o objeto `Element`.

Método	Descrição
`getAttribute(name)`	Retorna uma string com o valor do atributo do elemento (ou retorna `null` se o atributo especificado não for encontrado). Parâmetros: `name` - (string) Nome do atributo.
`getProperty(name)`	Retorna uma string com o valor da propriedade do elemento (ou retorna `null` se a propriedade especificada não for encontrada). Parâmetros: `name` - (string) Nome da propriedade.
`getText()`	Retorna uma string com o valor do texto do elemento.
`click()`	Clica em um elemento.
`clear()`	Limpa o conteúdo de um elemento editável.
`sendKeys(keys)`	Envia teclas. Parâmetros: `keys` - (string) Teclas a serem enviadas.

Cookie

O objeto `Cookie` é retornado pelo método `getCookies()` do objeto `Browser` e passado para o método `addCookie()`.

Embora o objeto `Cookie` não tenha nenhum método, ele pode conter as seguintes propriedades:

Propriedade	Tipo	Descrição
name	string	Nome do cookie.
value	string	Valor do cookie.
path	string	Caminho para o qual o cookie é válido. O padrão é `"/"` se omitido ao adicionar um cookie.
domain	string	Domínio para o qual o cookie é visível. O padrão é o domínio da URL do documento ativo do contexto de navegação atual da sessão, se omitido ao adicionar um cookie.
secure	boolean	Indica se o cookie é seguro. O padrão é `false` se omitido ao adicionar um cookie.
httpOnly	boolean	Indica se o cookie é HTTP-only. O padrão é `false` se omitido ao adicionar um cookie.
expiry	integer	Tempo de expiração do cookie (em segundos desde a Época Unix). Não deve ser definido se omitido ao adicionar um cookie.
sameSite	string	O atributo `sameSite` do cookie, que controla se o cookie deve ser restrito a um contexto de primeira parte ou mesmo site. Pode ser definido como `"Lax"` ou `"Strict"`. O padrão é `"None"` se omitido ao adicionar um cookie.

Alert

O objeto `Alert` representa um alerta de página web, é retornado pelo método `getAlert()` do objeto `Browser` e não pode ser construído diretamente.

O objeto `Alert` contém a propriedade `text` com o texto do alerta (ou `null` se não houver alertas).

Os seguintes métodos são suportados com o objeto `Alert`.

Método	Descrição
`accept()`	Aceita o alerta.
`dismiss()`	Descarta o alerta.

Result

O objeto `Result` contém estatísticas de sessão e é retornado pelo método `getResult()` do objeto `Browser`.

Normalmente, o objeto `Result` é convertido em string e retornado do script, e então analisado em valores de item dependente por meio de pré-processamento.

Embora o objeto `Result` não tenha nenhum método, ele pode conter as seguintes propriedades.

Propriedade	Tipo	Descrição
duration	string	Duração da sessão desde a criação da sessão até a recuperação do resultado.
error	object	Informações de erro.
http_status	integer	Status HTTP retornado pelo WebDriver (ou 0 se não houver erros do WebDriver).
error_code	string	Erro retornado pelo WebDriver (ou string vazia se não houver erros do WebDriver).
message	string	Mensagem de erro do WebDriver (ou string vazia se não houver erros do WebDriver).
performance_data	object	Estatísticas de desempenho.
summary	object	Resumo de desempenho.
navigation	object	Resumo de navegação.
resource	object	Resumo de recursos.
details	array de objetos	Estatísticas de desempenho após cada operação que poderia ter resultado em navegação.
mark	string	*(opcional)* Marca de snapshot de desempenho especificada com o método `collectPerfEntries()`.
navigation	object	Estatísticas de navegação.
resource	object	Resumo de recursos para esta etapa.
user	array de objetos	Array de estatísticas do tipo mark/measure.
marks	array de objetos	Índices de snapshot de desempenho marcados.
name	string	Nome da marca de snapshot de desempenho.
index	integer	Índice de snapshot de desempenho no array de detalhes.

6 Pré-processamento de CSV para JSON

Visão geral

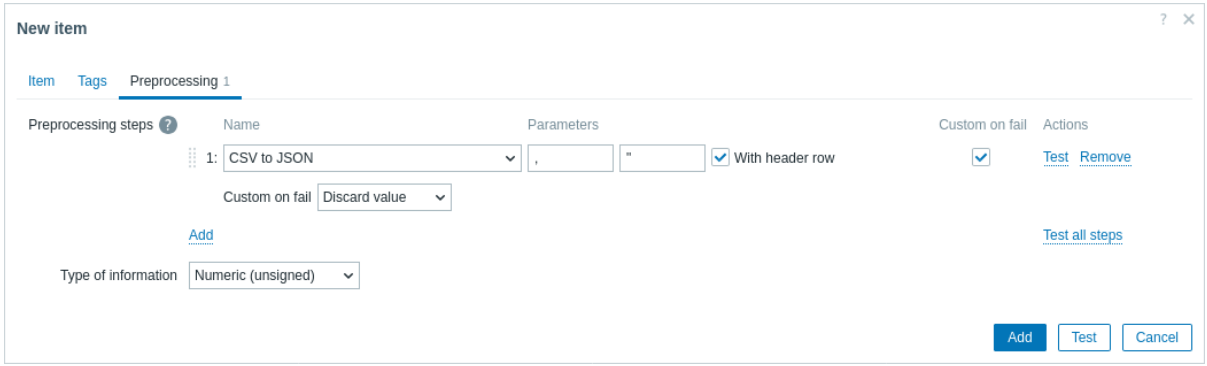
Nesta etapa de pré-processamento é possível converter arquivo de dado CSV para formato JSON. É suportado em:

- itens (protótipos de item)
- regras de descoberta de baixo-nível

Configuração

Para configurar uma etapa de pré-processamento CSV para JSON:

- Vá até a aba de Pré-processamento na configuração item/regra de descoberta
- Clique em *Adicionar*
- Selecione a opção *CSV para JSON*



O primeiro parâmetro permite definir um delimitador customizado. Note que se a primeira linha da entrada CSV começa com "Sep=" e é seguida por um caracter UTF-8 único esse caracter será usado como o delimitador no caso de o primeiro parâmetro não estar definido. Se o primeiro parâmetro não estiver definido e um delimitar não for recuperado da linha "Sep=", então uma vírgula é usada como separador.

O segundo parâmetro opcional permite definir um símbolo para quotação.

Se a caixa *Com linha de cabeçalho* estiver marcada, o valor da linha de cabeçalho será interpretada como nomes de coluna (veja Processamento de cabeçalho para mais informações).

Se a caixa *Customizado em caso de falha* estiver marcada, o item não se tornará não suportado no caso de uma etapa de pré-processamento falha. Adicionalmente opções de manipulação de erro podem ser definidas: descarte o valor, defina um valor especificado ou defina uma mensagem de erro especificada.

Processamento de cabeçalho

A linha de cabeçalho do arquivo CSV pode ser processada em duas formas diferentes:

- Se a caixa *Com linha de cabeçalho* estiver marcada - os valores da linha são interpretados como nomes de coluna. Neste caso os nomes de coluna devem ser únicos e a linha de dado não deve conter mais colunas do que a linha de cabeçalho;
- Se a caixa *Com linha de cabeçalho* não estiver marcada - a linha de cabeçalho é interpretada como dado. Os nomes de coluna são gerados automaticamente (1,2,3,4...)

Exemplo de arquivo CSV:

```
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:
Um caracter de quotação dentro de um campo quotado na entrada deve ser escapado precedendo-o com outro caracter de quotação.

Processando linha de cabeçalho

Saída JSON quando uma linha de cabeçalho é esperada:

```
[
   {
      "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"
```

```
    }
]
```

Sem processamento de linha de cabeçalho

Saída JSON quando uma linha de cabeçalho não é esperada:

```
[
    {
        "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"
    }
]
```

4 História e tendências

Visão geral

Histórico e tendências são as duas formas de armazenar dados coletados no Zabbix.

Enquanto o histórico mantém cada valor coletado, as tendências mantêm informações médias por hora e, portanto, consomem menos recursos.

Mantendo o histórico

Você pode definir por quantos dias o histórico será mantido:

- nas propriedades do item formulário
- ao atualizar itens em massa
- ao configurar tarefas do housekeeper

Qualquer dado mais antigo será removido pelo housekeeper.

O conselho geral é manter o histórico pelo menor número possível de dias e, assim, não sobrecarregar o banco de dados com muitos valores históricos.

Em vez de manter um histórico longo, você pode manter dados de tendências por mais tempo. Por exemplo, você pode manter o histórico por 14 dias e as tendências por 5 anos.

Você pode ter uma boa ideia de quanto espaço é necessário para o histórico em comparação com os dados de tendências consultando a página de dimensionamento do banco de dados.

Ao manter um histórico mais curto, você ainda poderá revisar dados antigos em gráficos, pois os gráficos usarão valores de tendências para exibir dados mais antigos.

Attention:
Se o histórico for definido como '0', o item atualizará apenas os itens dependentes e o inventário. Nenhuma função de trigger será avaliada porque a avaliação de trigger é baseada apenas em dados históricos.

Note:
Como uma forma alternativa de preservar o histórico, considere usar a funcionalidade de exportação de histórico de módulos carregáveis.

Mantendo tendências

Tendências são um mecanismo interno de redução de dados históricos que armazena o valor mínimo, máximo, médio e o número total de valores por hora para tipos de dados numéricos.

Você pode definir por quantos dias as tendências serão mantidas:

- nas propriedades do item
- ao atualizar itens em massa
- ao configurar tarefas do Housekeeper

As tendências geralmente podem ser mantidas por muito mais tempo do que o histórico. Quaisquer dados mais antigos serão removidos pelo housekeeper.

O servidor Zabbix acumula dados de tendências em tempo de execução no cache de tendências, à medida que os dados chegam. O servidor grava as tendências da **hora anterior** de cada item no banco de dados (onde o frontend pode encontrá-las) nas seguintes situações:

- o servidor recebe o primeiro valor da hora atual do item
- faltam 5 minutos ou menos para o final da hora atual e ainda não há valores da hora atual do item
- o servidor é parado

Para ver tendências em um gráfico, você precisa esperar pelo menos até o início da próxima hora (se o item for atualizado com frequência) e no máximo até o final da próxima hora (se o item for atualizado raramente), o que é no máximo 2 horas.

Quando o servidor grava o cache de tendências e já existem tendências no banco de dados para essa hora (por exemplo, o servidor foi reiniciado no meio da hora), o servidor precisa usar comandos de atualização em vez de simples inserções. Portanto, em uma instalação maior, se for necessário reiniciar, é desejável parar o servidor no final de uma hora e iniciar no início da próxima hora para evitar sobreposição de dados de tendências.

As tabelas de histórico não participam da geração de tendências de nenhuma forma.

Attention:
Se as tendências forem definidas como '0', o servidor Zabbix não calcula nem armazena tendências.

Note:
As tendências são calculadas e armazenadas com o mesmo tipo de dado dos valores originais. Como resultado, os cálculos de valor médio de valores do tipo de dado sem sinal são arredondados e quanto menor o intervalo de valores, menos preciso será o resultado. Por exemplo, se o item tiver valores 0 e 1, o valor médio será 0, não 0,5.
Além disso, reiniciar o servidor pode resultar em perda de precisão nos cálculos de valor médio do tipo de dado sem sinal para a hora atual.

5 Parâmetros do usuário

Visão geral

Às vezes, você pode querer executar uma verificação de agent que não vem predefinida no Zabbix. É aí que os parâmetros de usuário ajudam.

Você pode escrever um comando que recupere os dados necessários e incluí-lo no parâmetro de usuário no arquivo de configuração do agent (parâmetro de configuração 'UserParameter').

Um parâmetro de usuário tem a seguinte sintaxe:

```
UserParameter=<chave>,<comando>
```

Como você pode ver, um parâmetro de usuário também contém uma chave. A chave será necessária ao configurar um item. Digite uma chave de sua escolha que seja fácil de referenciar (deve ser única dentro de um host).

Reinicie o agent ou use a opção de controle em tempo de execução do agent para carregar o novo parâmetro, por exemplo:

```
zabbix_agentd -R userparameter_reload
```

Em seguida, ao configurar um item, insira a chave para referenciar o comando do parâmetro de usuário que você deseja executar.

Os parâmetros de usuário são comandos executados pelo agent do Zabbix. Observe que até 16MB de dados podem ser retornados antes das etapas de pré-processamento do valor do item.

/bin/sh é usado como interpretador de linha de comando em sistemas operacionais UNIX. Os parâmetros de usuário obedecem ao tempo limite de verificação do agent; se o tempo limite for atingido, o processo do parâmetro de usuário bifurcado será encerrado.

Veja também:

- Tutorial passo a passo sobre como usar parâmetros de usuário
- Execução de comandos

Exemplos de parâmetros de usuário simples

Um comando simples:

```
UserParameter=ping,echo 1
```

O agente sempre retornará '1' para um item com a chave 'ping'.

Um exemplo mais complexo:

```
UserParameter=mysql.ping,mysqladmin -uroot ping | grep -c vivo
```

O agente retornará '1', se o servidor MySQL estiver ativo, '0' - caso contrário.

Parâmetros de usuário flexíveis

Parâmetros de usuário flexíveis aceitam parâmetros com a chave. Desta forma, um parâmetro de usuário flexível pode ser a base para criar vários items.

Parâmetros de usuário flexíveis têm a seguinte sintaxe:

```
UserParameter=key[*],command
```

Parâmetro	Descrição
Key	Chave de item única. O [*] define que esta chave aceita parâmetros dentro dos colchetes. Os parâmetros são fornecidos ao configurar o item.
Command	Comando a ser executado para avaliar o valor da chave. *Apenas para parâmetros de usuário flexíveis*: Você pode usar referências posicionais $1...$9 no comando para se referir ao respectivo parâmetro na chave do item. O Zabbix analisa os parâmetros entre [ ] da chave do item e substitui $1,...,$9 no comando de acordo. $0 será substituído pelo comando original (antes da expansão de $0,...,$9) a ser executado. Referências posicionais são interpretadas independentemente de estarem entre aspas duplas (") ou simples ('). Para usar referências posicionais sem alteração, especifique um cifrão duplo - por exemplo, awk '{print $$2}'. Neste caso, $$2 será convertido em $2 ao executar o comando.

Attention:
Referências posicionais com o sinal $ são procuradas e substituídas pelo agent Zabbix apenas para parâmetros de usuário flexíveis. Para parâmetros de usuário simples, esse processamento de referência é ignorado e, portanto, qualquer citação do sinal $ não é necessária.

Attention:
Certos caracteres não são permitidos em parâmetros de usuário por padrão. Para a lista completa de caracteres, consulte UnsafeUserParameters.

Exemplo 1

Algo muito simples:

```
UserParameter=ping[*],echo $1
```

Podemos definir um número ilimitado de itens para monitoramento, todos com formato ping[algo].

- ping[0] - sempre retornará '0'
- ping[aaa] - sempre retornará 'aaa'

Exemplo 2

Vamos adicionar mais sentido!

```
UserParameter=mysql.ping[*],mysqladmin -u$1 -p$2 ping | grep -c vivo
```

Este parâmetro pode ser usado para monitorar a disponibilidade do MySQL base de dados. Podemos passar nome de usuário e senha:

```
mysql.ping[zabbix,our_password]
```

Exemplo 3

Quantas linhas correspondem a uma expressão regular em um arquivo?

```
UserParameter=wc[*],grep -c "$2" $1
```

Este parâmetro pode ser usado para calcular o número de linhas em um arquivo.

```
wc[/etc/passwd,root]
wc[/etc/services,zabbix]
```

Resultado do comando

O valor de retorno do comando é uma saída padrão juntamente com um erro padrão produzido pelo comando.

> **Attention:**
> Um item que retorna texto (caractere, log ou tipo de informação de texto) não ficará não suportado em caso de uma saída de erro padrão.

O valor de retorno é limitado a 16MB (incluindo espaços em branco à direita que são truncados); limites do banco de dados também se aplicam.

Parâmetros de usuário que retornam texto (caractere, log ou tipo de informação de texto) também podem retornar um espaço em branco. Em caso de resultado inválido, o item ficará não suportado.

1 Estendendo agentes Zabbix

Este tutorial fornece instruções passo a passo sobre como estender o funcionalidade do agente Zabbix com o uso de um user parâmetro.

Passo 1

Escreva um script ou linha de comando para recuperar o parâmetro necessário.

Por exemplo, podemos escrever o seguinte comando para obter o total número de consultas executadas por um servidor MySQL:

```
status mysqladmin -uroot | corte -f4 -d":" | corte -f1 -d"S"
```

Quando executado, o comando retorna o número total de consultas SQL.

Passo 2

Adicione o comando ao zabbix_agentd.conf:

```
UserParameter=mysql.questions, mysqladmin -uroot status | corte -f4 -d":" | corte -f1 -d"S"
```

mysql.questions é um identificador único. Pode ser qualquer chave válida identificador, por exemplo, *consultas*.

Teste este parâmetro usando o agente Zabbix com o sinalizador "-t" (se estiver executando em root, no entanto, observe que o agente pode ter permissões diferentes quando lançado como um daemon):

```
zabbix_agentd -t mysql.questions
```

Etapa 3

Recarregue os parâmetros do usuário do arquivo de configuração executando:

```
zabbix_agentd -R userparameter_reload
```

Você também pode reiniciar o agente em vez do comando de controle de tempo de execução.

Teste o parâmetro usando o utilitário zabbix_get.

Passo 4

Adicione um novo item com Key=mysql.questions ao host monitorado. Tipo de o item deve ser Zabbix Agent ou Zabbix Agent (ativo).

Esteja ciente de que o tipo de valores retornados deve ser definido corretamente no Zabbix servidor. Caso contrário, o Zabbix não os aceitará.

6 Contadores de desempenho do Windows

Visão geral

Você pode monitorar efetivamente os contadores de desempenho do Windows usando o *perf_counter[]* tecla.

Por exemplo:

```
perf_counter["\Processor(0)\Interrupções/s"]
```

ou

```
perf_counter["\Processor(0)\Interrupções/s", 10]
```

Para obter mais informações sobre como usar esta chave ou seu equivalente somente em inglês `perf_counter_en`, consulte item específico do Windows keys.

Para obter uma lista completa de contadores de desempenho disponíveis para monitoramento, você pode executar:

```
typeperf -qx
```

Você também pode usar a descoberta de baixo nível para descobrir vários object instâncias dos contadores de desempenho do Windows e automatizar a criação de perf_counter itens para vários objetos de instância.

Representação numérica

O Windows mantém representações numéricas (índices) para nomes de objetos e contadores de desempenho. O Zabbix suporta essas representações numéricas como parâmetros para as chaves de item `perf_counter`, `perf_counter_en` e nos parâmetros de configuração `PerfCounter`, `PerfCounterEn`.

No entanto, não é recomendado usá-las, a menos que você possa garantir que seus índices numéricos correspondam às strings corretas em hosts específicos. Se você precisar criar items portáteis que funcionem em diferentes hosts com várias versões do Windows localizadas, pode usar a chave `perf_counter_en` ou o parâmetro de configuração `PerfCounterEn`, que permitem usar nomes em inglês independentemente da localidade do sistema.

Para descobrir os equivalentes numéricos, execute o **regedit** e localize *Counter* em *HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Perflib\009*.

A entrada do registro contém informações como esta:

```
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
....
```

Aqui você pode encontrar os números correspondentes para cada parte da string do contador de desempenho, como em '\System\% Processor Time':

```
System → 2
% Processor Time → 6
```

Então você pode usar esses números para representar o caminho em números:

```
\2\6
```

Parâmetros do contador de desempenho

Você pode implantar alguns parâmetros PerfCounter para o monitoramento do Windows contadores de desempenho.

Por exemplo, você pode adicioná-los ao arquivo de configuração do agente Zabbix:

```
PerfCounter=UserPerfCounter1,"\Memory\Page Reads/s",30
ou
PerfCounter=UserPerfCounter2,"\4\24",30
```

Com esses parâmetros em vigor, você pode simplesmente usar *UserPerfCounter1* ou *UserPerfCounter2* como as chaves para criar o respectivos itens.

Lembre-se de reiniciar o agente Zabbix depois de fazer alterações no arquivo de configuração.

7 Atualização em massa

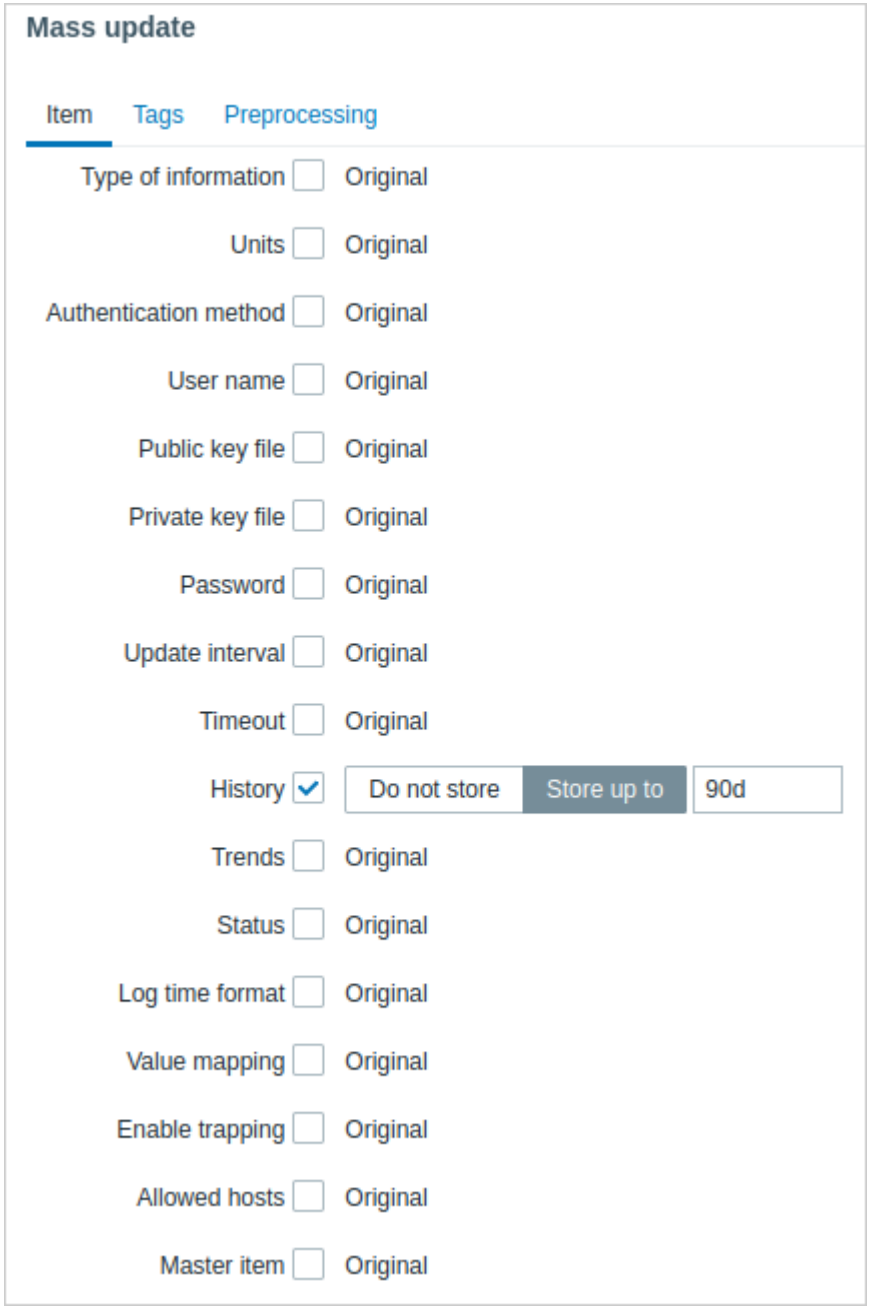
Visão geral

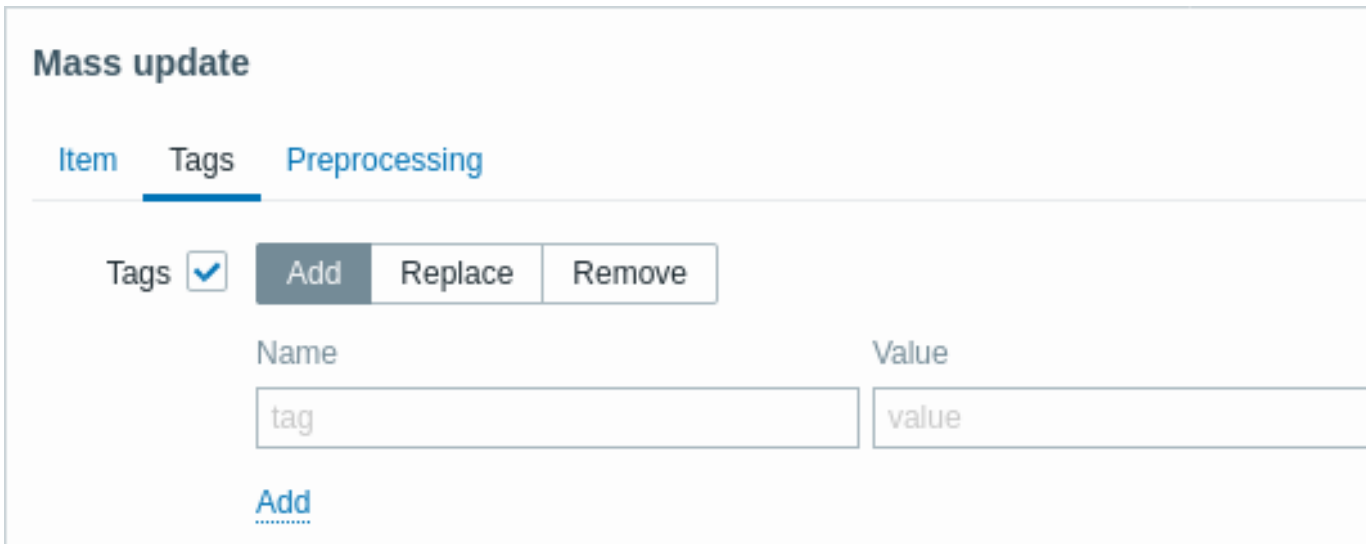
Algumas vezes você precisa alterar um atributo em um grande número de itens de uma só vez. Ao invés de alterar item a item, você pode utilizar o recurso de atualização em massa para fazer esta tarefa.

Usando atualização em massa

Para atualizar em massa alguns items, faça o seguinte:

- Marque as caixas de seleção dos items a serem atualizados na lista
- Clique em *Atualizar em massa* abaixo da lista
- Navegue até a aba com os atributos necessários (*Item*, *Tags* ou *Pré-processamento*)
- Marque as caixas de seleção dos atributos a serem atualizados
- Insira novos valores para os atributos



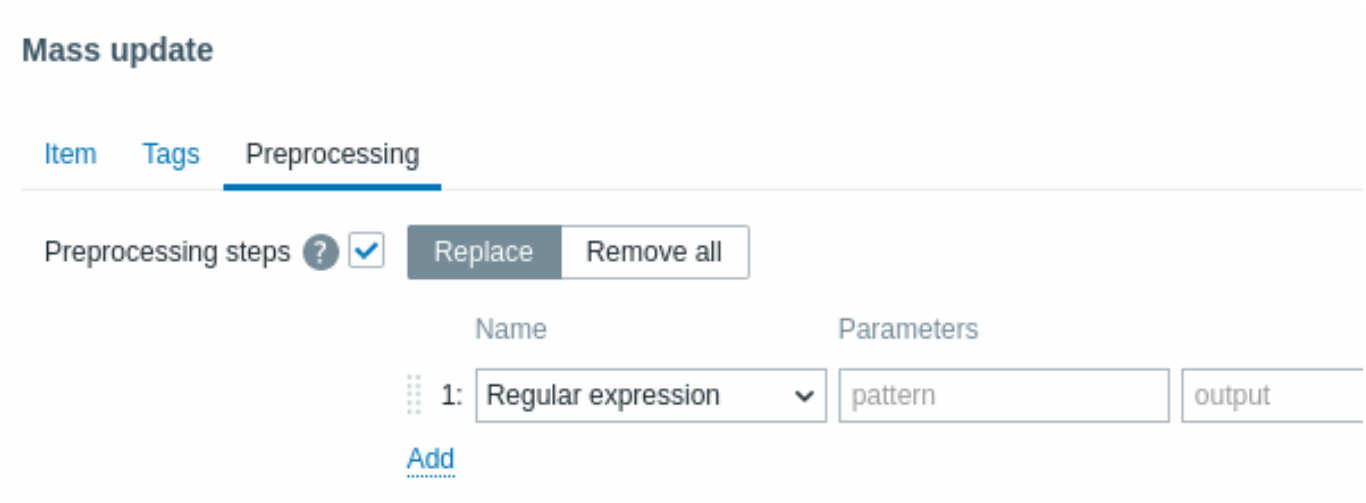


A opção *Tags* permite:

- *Adicionar* - adicionar as tags especificadas aos items (tags com o mesmo nome, mas valores diferentes não são consideradas 'duplicadas' e podem ser adicionadas ao mesmo item).
- *Substituir* - remover as tags especificadas e adicionar tags com novos valores

- *Remover* - remover as tags especificadas dos items

Macros de usuário, macros {INVENTORY.*}, {HOST.HOST}, {HOST.NAME}, {HOST.CONN}, {HOST.DNS}, {HOST.IP}, {HOST.PORT} e {HOST.ID} são suportadas em tags.



Marque a caixa ao lado de *Etapas de pré-processamento* para ativar a atualização em massa para etapas de pré-processamento.

A atualização em massa de *Pré-processamento* permite:

- *Substituir* - substituir as etapas de pré-processamento existentes nos items pelas etapas de pré-processamento especificadas abaixo
- *Remover todas* - remover todas as etapas de pré-processamento dos items

Quando terminar, clique em *Atualizar*.

8 Mapeamento de valores

Visão geral

O mapeamento de valores permite configurar uma representação mais amigável dos valores recebidos, usando mapeamentos entre valores numéricos/strings e representações em string.

Por exemplo, quando o valor de um item é ”0” ou ”1”, os mapeamentos de valores podem ser usados para representar esses valores de uma forma mais amigável:

- 0 → Não disponível
- 1 → Disponível

Mapeamentos de valores para tipos de backup de dados podem ser configurados da seguinte forma:

- F → Completo
- D → Diferencial
- I → Incremental

Intervalos de valores para tensão podem ser configurados da seguinte forma:

- <=209 → Baixa
- 210-230 → OK
- >=231 → Alta

O mapeamento de valores é usado no frontend do Zabbix e nas notificações enviadas pelos tipos de mídia.

Attention:
A substituição do valor recebido pela representação configurada é realizada tanto no frontend do Zabbix quanto no server; no entanto, o server realiza a substituição apenas nos seguintes casos:

- ao preencher o inventário do host;
- ao expandir as macros suportadas {ITEM.VALUE}, {ITEM.LASTVALUE}, {EVENT.OPDATA} e {EVENT.CAUSE.OPDATA}.

Os mapeamentos de valores são configurados em templates ou hosts. Uma vez configurados, eles ficam disponíveis para todos os items dentro do respectivo template ou host. Ao configurar items, especifique o nome de um mapeamento de valor previamente configurado no parâmetro *Mapeamento de valor*.

Note:
Não há herança de mapa de valores - hosts e templates não herdam mapeamentos de valores de templates vinculados. Os items de template em um host continuarão a usar os mapeamentos de valores configurados no template.

Note:
Os mapeamentos de valores podem ser usados com items que possuem os tipos de informação *Numérico (sem sinal)*, *Numérico (float)* e *Caracter*.

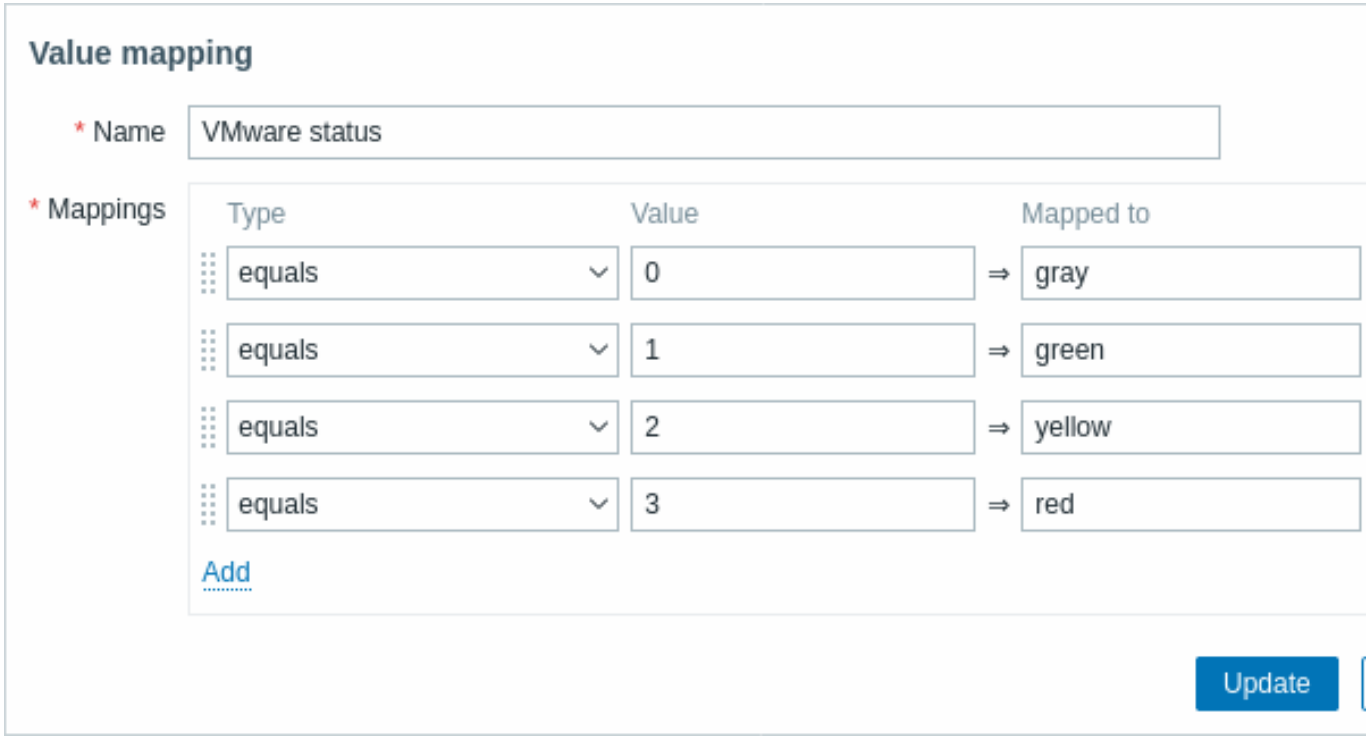
Os mapeamentos de valores são exportados/importados com o respectivo template ou host. Eles também podem ser atualizados em massa usando os formulários de atualização em massa de host e template.

Configuração

Para configurar um mapeamento de valores, siga estas etapas:

1. Abra o formulário de configuração do host ou template.

2. Na aba *Mapeamento de valores*, clique em *Adicionar* para adicionar um novo mapeamento de valores ou clique no nome de um mapeamento existente para editá-lo.



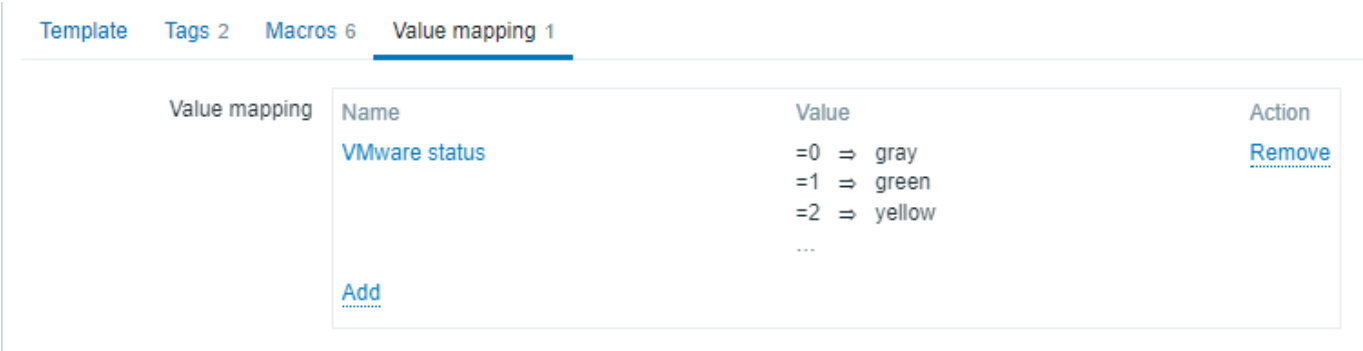
Parâmetros de um mapeamento de valores:

Parâmetro	Descrição
Nome	Nome exclusivo para o conjunto de mapeamentos de valores.
Mapeamentos	Regras individuais para mapear valores numéricos/string para representações em string. O mapeamento é aplicado na ordem das regras, que podem ser reordenadas arrastando.

Parâmetro	Descrição
Tipo	Tipo de mapeamento: **equals** - valores iguais serão mapeados; **is greater than or equals** - valores iguais ou maiores serão mapeados; **is less than or equals** - valores iguais ou menores serão mapeados; **in range** - valores dentro do intervalo serão mapeados; o intervalo é expresso como <número1>-<número2> ou <número>; múltiplos intervalos são suportados (por exemplo, 1-10,101-110,201); **regexp** - valores correspondentes à expressão regular serão mapeados (expressões regulares globais não são suportadas); **default** - todos os valores restantes serão mapeados, exceto aqueles com mapeamentos específicos. Para mapeamento de intervalos, apenas tipos de valores numéricos (*is greater than or equals*, *is less than or equals*, *in range*) são suportados.
Valor	Valor de entrada (pode conter um intervalo ou expressão regular, dependendo do tipo de mapeamento).
Mapeado para	Representação em string (até 64 caracteres) para o valor de entrada.

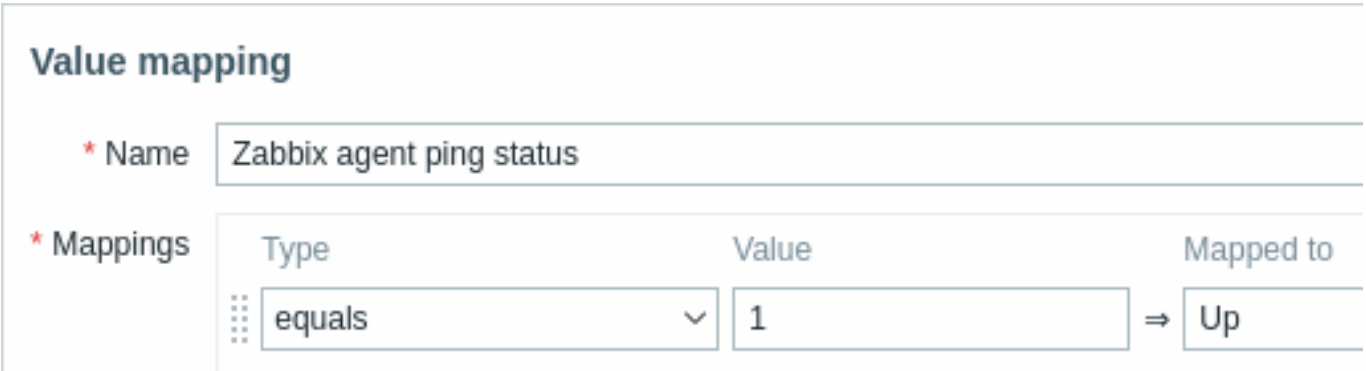
Todos os campos obrigatórios estão marcados com um asterisco vermelho.

Ao visualizar o mapeamento de valores na lista, apenas os três primeiros mapeamentos são visíveis, com três pontos indicando que existem mais mapeamentos.



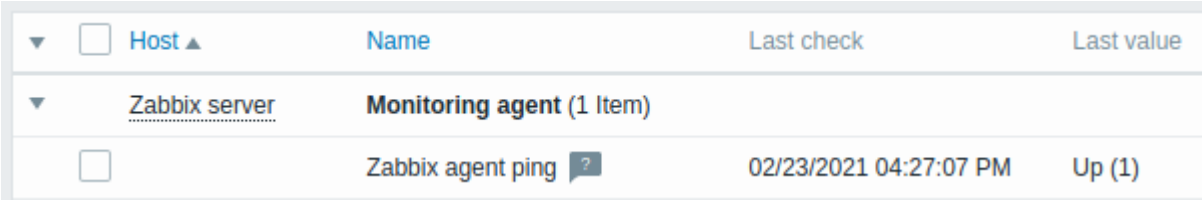
Exemplo de mapeamento de valor

Um dos itens de agent predefinidos, *Zabbix agent ping*, usa um mapeamento de valor em nível de template "Zabbix agent ping status" para exibir seus valores.



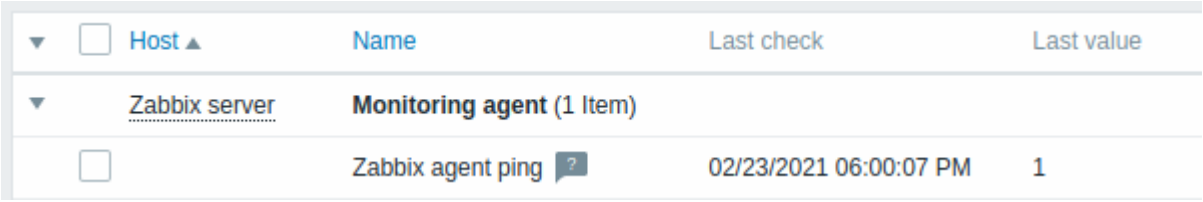
No formulário de configuração do item, você pode encontrar uma referência a esse mapeamento de valor no campo *Mapeamento de valor*:

Esse mapeamento é usado na seção *Monitoramento → Últimos dados* para exibir "Up" (com o valor bruto entre parênteses).



Note:
Na seção *Últimos dados*, os valores exibidos são reduzidos para 20 símbolos. Se o mapeamento de valor for usado, essa redução não é aplicada ao valor mapeado, mas apenas ao valor bruto (exibido entre parênteses).

Sem um mapeamento de valor predefinido, você veria apenas ”1”, o que pode ser difícil de entender.



9 Fila

Visão geral

A fila exibe os items que estão aguardando atualização. A fila é apenas uma representação **lógica** dos dados. Não existe uma fila IPC ou qualquer outro mecanismo de fila no Zabbix.

Items monitorados por proxies também são incluídos na fila - eles serão contados como em fila durante o período de atualização dos dados históricos do proxy.

Apenas items com tempos de atualização agendados são exibidos na fila. Isso significa que os seguintes tipos de items são excluídos da fila:

- items ativos de log, logrt e event log do agent Zabbix
- items SNMP trap
- items trapper
- items de monitoramento web
- items dependentes

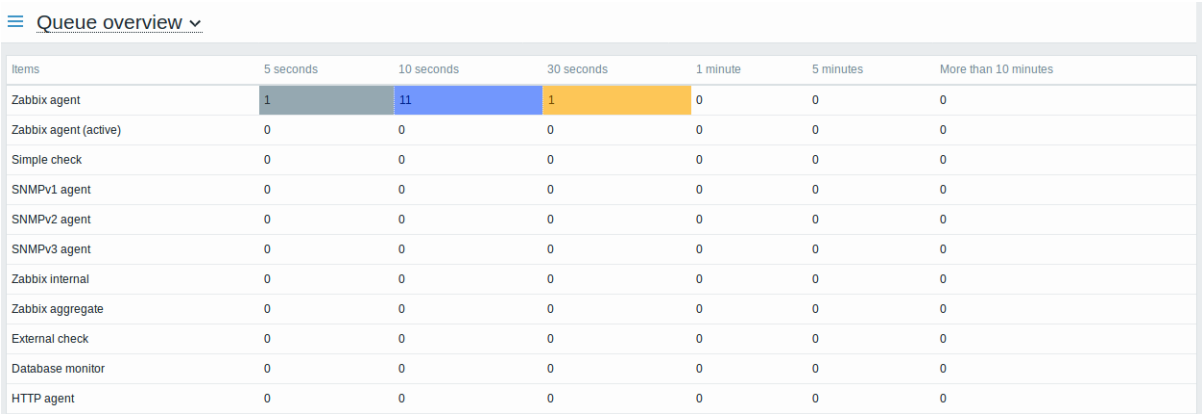
As estatísticas mostradas pela fila são um bom indicador do desempenho do Zabbix server.

A fila é recuperada diretamente do Zabbix server usando o protocolo JSON. A informação está disponível apenas se o Zabbix server estiver em execução.

Attention:
Os items não são contados na fila se a interface do item se tornar indisponível devido a problemas de conexão ou ao agent não estar funcionando corretamente.

Lendo a fila

Para ler a fila, vá para *Administração* > *Fila* > *Visão geral da fila*.



A imagem aqui está geralmente ”ok”, então podemos supor que o server está indo bem.

A fila mostra alguns items esperando até 30 segundos. Seria ótimo saber quais items são esses.

Para fazer isso, selecione *Detalhes da fila* no menu suspenso do título. Agora você pode ver uma lista desses items atrasados.

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

Com esses detalhes fornecidos, pode ser possível descobrir por que esses items podem estar atrasados.

Com um ou dois items atrasados, talvez não haja motivo para alarme. Eles podem ser atualizados em um segundo. No entanto, se você vir vários items sendo atrasados por muito tempo, pode haver um problema mais sério.

Veja também: Alinhando fusos horários ao usar intervalos de agendamento.

Queue overview

Items	5 seconds	10 seconds	30 seconds	1 minute	5 minutes	More than 10 minutes
Zabbix agent	0	1	1	26	0	0
Zabbix agent (active)	0	0	0	0	0	0
Simple check	0	0	0	0	0	0
SNMPv1 agent	0	0	0	0	0	0
SNMPv2 agent	0	0	0	0	0	0
SNMPv3 agent	0	0	0	0	0	0
Zabbix internal	0	0	0	0	0	0
Zabbix aggregate	0	0	0	0	0	0
External check	0	0	0	0	0	0
Database monitor	0	0	0	0	0	0
HTTP agent	0	0	0	0	0	0
IPMI agent	0	0	0	0	0	0
SSH agent	0	0	0	0	0	0
TELNET agent	0	0	0	0	0	0
JMX agent	0	0	0	0	0	0
Calculated	0	0	0	0	0	0

Item da fila

Existe um item especial de uso interno ao Zabbix que responde pela chave **zabbix[queue,<from>,<to>]** e pode ser utilizado para monitorar a saúde da fila do Zabbix. Ele irá retornar a quantidade de itens em atraso em um determinado tempo. Para maiores detalhes favor consultar o manual de itens internos.

10 Cache de valores

Visão geral

Para tornar o cálculo de expressões de trigger, items calculados e algumas macros muito mais rápido, uma opção de cache de valores é suportada pelo Zabbix server.

Este cache em memória pode ser usado para acessar dados históricos, em vez de fazer chamadas SQL diretas ao banco de dados. Se os valores históricos não estiverem presentes no cache, os valores ausentes são solicitados ao banco de dados e o cache é atualizado de acordo.

Os valores dos items permanecem no cache de valores até que:

- o item seja excluído (os valores em cache são excluídos após a próxima sincronização de configuração);
- o valor do item esteja fora do intervalo de tempo ou contagem especificado na expressão do trigger/item calculado (o valor em cache é removido quando um novo valor é recebido);
- o intervalo de tempo ou contagem especificado na expressão do trigger/item calculado seja alterado para que menos dados sejam necessários para o cálculo (valores em cache desnecessários são removidos após 24 horas).

Note:
O status do cache de valores pode ser observado usando a opção de controle de tempo de execução do server `diaginfo` (ou `diaginfo=valuecache`) e inspecionando a seção de informações de diagnóstico do cache de valores. Isso pode ser útil para determinar triggers ou items calculados mal configurados.

Para habilitar a funcionalidade de cache de valores, um parâmetro opcional **ValueCacheSize** é suportado pelo arquivo de configuração do Zabbix server.

Dois itens internos são suportados para monitorar o cache de valores: **zabbix[vcache,buffer,<mode>]** e **zabbix[vcache,cache,<paramet**
Veja mais detalhes em itens internos.

11 Executar agora

Visão geral

A opção *Executar agora* permite executar verificações passivas imediatamente.

A coleta de valores de item no Zabbix segue os intervalos de atualização configurados. Alguns itens (por exemplo, regras de descoberta de baixo nível) têm intervalos longos e pode ser necessário buscar um novo valor imediatamente — por exemplo, para detectar alterações em recursos descobertos.

A opção *Executar agora* é suportada para os seguintes tipos de item:

- Zabbix agent (passivo)
- Verificação simples
- SNMP agent (v1/v2/v3)
- Zabbix interno
- Verificação externa
- Monitor de banco de dados
- HTTP agent
- IPMI agent
- SSH agent
- TELNET agent
- JMX agent
- Calculado
- Item dependente
- Script
- Navegador

Se o item for do tipo ”Item dependente”, seu item mestre também deve ser de um dos tipos acima.

Configuração

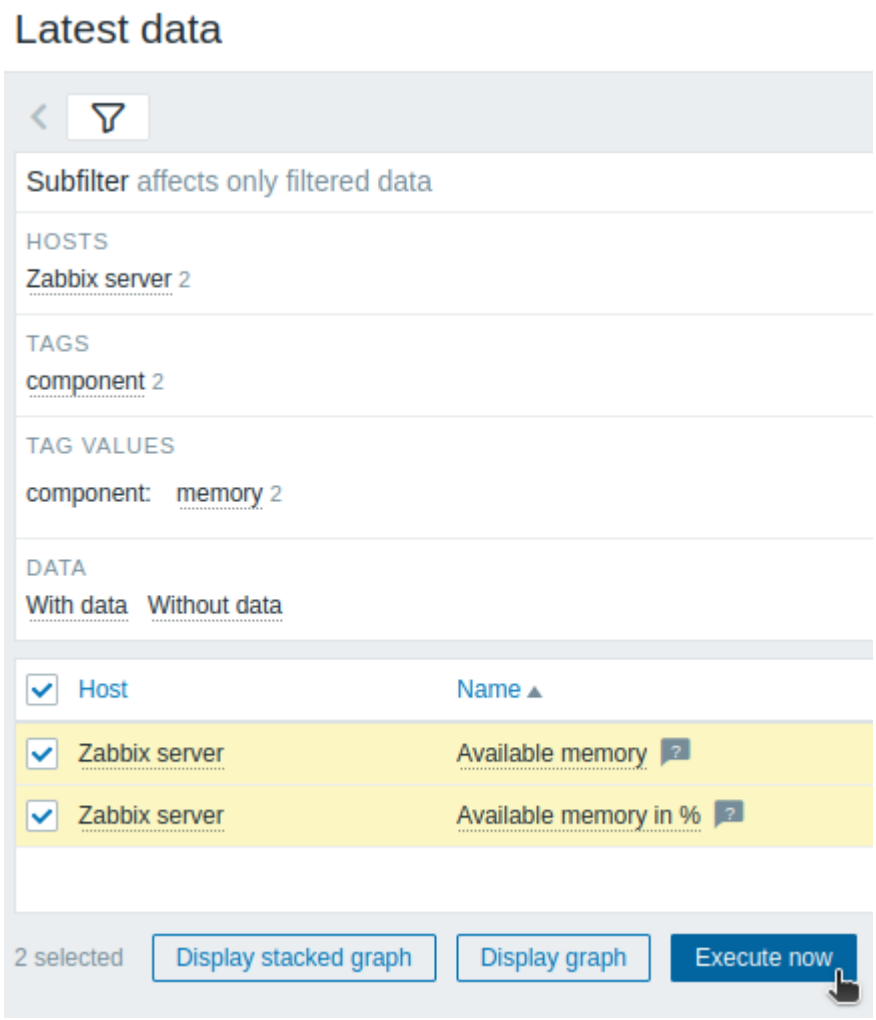
Você pode executar verificações passivas a partir de:

- *Monitoramento* > *Últimos dados*
- *Coleta de dados* > *Hosts* > *Itens* ou *Descoberta*
- O menu de item
- Ao editar items ou regras de descoberta de baixo nível (em seus formulários de configuração)

Attention:
A verificação deve existir no cache de configuração para ser executada; veja também CacheUpdateFrequency. O cache **não é atualizado** antes da execução da verificação, portanto, alterações recentes na configuração do item ou da regra de descoberta de baixo nível não serão consideradas. Para testar items ou regras LLD que estão sendo criadas ou recém-criadas, você pode usar a opção Testar.

Por exemplo, para executar uma verificação passiva em *Monitoramento* > *Últimos dados*:

1. Selecione os items; você pode executar a verificação para vários items de uma vez.
2. Clique em *Executar agora*.



Note:
Em *Últimos dados*, os usuários podem *Executar agora* items em hosts para os quais possuem permissão de *leitura-escrita* permissão, ou permissão de *leitura* e a ação *Permitir "Executar agora" em hosts somente leitura* ação habilitada para sua função.

12 Restringindo verificações do agent

Visão geral

Você pode controlar quais keys de item o agent Zabbix ou agent 2 está autorizado ou proibido de usar ao executar verificações de item, comandos remotos ou scripts.

Para isso, utilize estes parâmetros de configuração do agent para definir regras de permissão/negação:

- `AllowKey=<pattern>`
- `DenyKey=<pattern>`

O `<pattern>` deve conter uma única key de item e suporta curingas (`*`). O curinga corresponde a qualquer número de quaisquer caracteres em sua posição e pode ser usado para corresponder a keys de item ou parâmetros (por exemplo, `vfs.file.*[*]`).

Attention:
Para melhorar a segurança, recomenda-se usar keys de item exatas em vez de curingas. Para mais detalhes, consulte Protegendo regras de permissão/negação.

Ao contrário de outros parâmetros de configuração do agent, você pode especificar um número ilimitado de parâmetros `AllowKey` ou `DenyKey`.

Notas importantes

- Todos os itens `system.run` estão desabilitados por padrão (mesmo quando `DenyKey` está vazio), como se `DenyKey=system.run[*]` estivesse definido como a última regra. Por causa disso, você pode permitir itens `system.run` específicos sem negar

explicitamente outros itens `system.run`.

- Um item especificado em `AllowKey` também deve ser especificado em `DenyKey` (exceto para itens `system.run`); caso contrário, o agent Zabbix **não iniciará**.
- Se possível, use `AllowKey` para permitir apenas os itens necessários e negar todo o resto. Algumas chaves podem ser abusadas para ler arquivos não intencionais via path traversal (por exemplo, `vfs.file.contents["../../../../etc/passwd"]`), e novas versões do agent Zabbix podem introduzir chaves não cobertas por suas regras `DenyKey`.
- A configuração de `AllowKey` e `DenyKey` não afeta os parâmetros do agent `HostnameItem`, `HostMetadataItem` ou `HostInterfaceItem`.
- Itens negados tornam-se não suportados sem qualquer dica ou mensagem de erro; por exemplo:
  - O parâmetro de linha de comando do agent Zabbix `--print (-p)` não mostrará as chaves de itens negados.
  - O parâmetro de linha de comando do agent Zabbix `--test (-t)` retornará "Unsupported item key." para as chaves de itens negados.
  - Quando o log está ativado (`LogRemoteCommands=1`), o arquivo de log do agent Zabbix não registrará comandos remotos negados.

Ordem das regras de permitir/negar

Você pode especificar um número ilimitado de regras `AllowKey` ou `DenyKey`, embora a ordem delas seja importante.

- As regras são avaliadas uma a uma, de cima para baixo.
- Quando uma chave de item corresponde a uma regra, ela é permitida ou negada, e a avaliação das regras é interrompida.

Por exemplo, ao avaliar `vfs.file.contents[/etc/passwd]`, as regras são processadas da seguinte forma:

```
AllowKey=vfs.file.contents[/tmp/app.log]    # O padrão da chave do item não corresponde, o agent prossegue
AllowKey=vfs.file.contents[/etc/passwd]     # O padrão da chave do item corresponde; o agent permite a che
DenyKey=vfs.file.*[*]                       # O agent ignora a regra, pois a avaliação foi interrompida.
```

A seguinte ordem de regras irá negar a checagem do item:

```
DenyKey=vfs.file.*[*]                       # O padrão da chave do item corresponde; o agent nega a checag
AllowKey=vfs.file.contents[/etc/passwd]     # O agent ignora a regra, pois a avaliação foi interrompida.
AllowKey=vfs.file.contents[/tmp/app.log]    # O agent ignora a regra, pois a avaliação foi interrompida.
```

Exemplos

Os exemplos a seguir mostram padrões comuns de configuração para `AllowKey` e `DenyKey`.

Permitindo verificações e comandos específicos

Permitir apenas duas verificações de item `vfs.file` e dois comandos `system.run`:

```
AllowKey=vfs.file.contents[/tmp/app.log]
AllowKey=vfs.file.size[/tmp/app.log]
AllowKey=system.run[/usr/bin/uptime]
AllowKey=system.run[/usr/bin/df -h /]
DenyKey=vfs.file.*[*]
```

> **Note:**
> Definir `DenyKey=system.run[*]` é desnecessário, pois todos os outros comandos `system.run` são negados por padrão.

Permitindo scripts

Permita que o agent Zabbix execute scripts em hosts por todos os métodos disponíveis:

- Scripts globais que podem ser executados no frontend ou via API (este método sempre usa a chave `system.run[myscript.sh]`)
- Comandos remotos de operações de ação (este método sempre usa a chave `system.run[myscript.sh,nowait]`)
- Itens do agent Zabbix `system.run` com o script, por exemplo:
  - `system.run[myscript.sh]`
  - `system.run[myscript.sh,wait]`
  - `system.run[myscript.sh,nowait]`

```
AllowKey=system.run[myscript.sh,*]
```

Para controlar o parâmetro `wait/nowait`, você deve definir uma regra diferente. Por exemplo, você pode permitir apenas itens `system.run[myscript.sh,wait]`, excluindo assim outros métodos:

```
AllowKey=system.run[myscript.sh,wait]
```

Protegendo regras allow/deny

Este exemplo mostra como proteger regras `AllowKey` ou `DenyKey` excessivamente permissivas.

Considere as seguintes regras:

```
AllowKey=system.run["C:\Program^ Files\Zabbix^ Agent^ 2\scripts\test.bat*"]
DenyKey=vfs.file.*
DenyKey=system.cpu.load[*]
```

Note:
No Windows, você deve escapar os espaços no caminho usando um acento circunflexo (^).

Essas regras contêm um curinga (*), que pode ser mal utilizado:

- O script `test.bat` pode ser executado com quaisquer argumentos, incluindo argumentos não intencionais.
- O padrão `vfs.file.*` corresponde apenas a chaves de item sem parâmetros; no entanto, todos os itens `vfs.file` requerem parâmetros.
- O padrão `system.cpu.load[*]` corresponde apenas a chaves de item com parâmetros; no entanto, os itens `system.cpu.load` não requerem parâmetros.

Para proteger essas regras, permita explicitamente a execução de `test.bat` apenas com argumentos específicos e negue padrões corretos de chaves de item; por exemplo:

```
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[*]
```

Você pode testar as regras executando os seguintes comandos, que retornarão `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]
```

Exemplos de padrões

A tabela a seguir mostra como os padrões de chave de item são correspondidos:

- Uma chave corresponde ao padrão apenas se atender a **todas** as condições na coluna *Corresponde*.
- Os parâmetros devem estar totalmente entre colchetes (por exemplo, `vfs.file.contents[*` e `vfs.file.contents*utf8]` são padrões inválidos).

Padrão	Corresponde	Exemplos
*	Qualquer chave com ou sem parâmetros	

`vfs.file.*`	A chave começa com `vfs.file.`
	Sem parâmetros
		Corresponde:
		`vfs.file.size`
		`vfs.file.contents`
		
		Não corresponde:
		`vfs.file.contents[]`
		`vfs.file.size[/var/log/app.log]`

`vfs.*.contents`	A chave começa com `vfs.`
	A chave termina com `.contents`
	Sem parâmetros
		Corresponde:
		`vfs..contents`
		`vfs.mount.point.file.contents`
		
		Não corresponde:
		`vfs.contents`
		`vfs.file.contents[]`

Padrão	Corresponde	Exemplos
vfs.file.*[*]	A chave começa com vfs.file. Qualquer parâmetro ou vazio	Corresponde vfs.file.get.custom[] vfs.file.size[/var/log/app.log, utf8] Não corresponde: vfs.file.get.custom
vfs.file.contents	A chave é vfs.file.contents Sem parâmetros	Corresponde: vfs.file.contents Não corresponde: vfs.file.contents[/etc/passwd]
vfs.file.contents[]	A chave é vfs.file.contents[] Parâmetros vazios	Corresponde: vfs.file.contents[] Não corresponde: vfs.file.contents
vfs.file.contents[*]	A chave é vfs.file.contents Qualquer parâmetro ou vazio	Corresponde: vfs.file.contents[/path/to/file] Não corresponde: vfs.file.contents
vfs.file.contents[/etc/passwd,*]	A chave é vfs.file.contents O primeiro parâmetro é /etc/passwd Qualquer segundo parâmetro ou vazio	Corresponde: vfs.file.contents[/etc/passwd,] vfs.file.contents[/etc/passwd,utf8] Não corresponde: vfs.file.contents[] vfs.file.contents[/etc/passwd]
vfs.file.contents[*passwd*]	A chave é vfs.file.contents O primeiro parâmetro inclui passwd Sem segundo parâmetro	Corresponde: vfs.file.contents[/etc/passwd] Não corresponde: vfs.file.contents[/etc/passwd,] vfs.file.contents[/etc/passwd,utf8]
vfs.file.contents[*passwd*,*]	A chave é vfs.file.contents O primeiro parâmetro inclui passwd Qualquer segundo parâmetro ou vazio	Corresponde: vfs.file.contents[/etc/passwd,] vfs.file.contents[/etc/passwd,utf8] Não corresponde: vfs.file.contents[/etc/passwd] vfs.file.contents[/tmp/test]
vfs.file.contents[/etc/passwd,utf8]	A chave é vfs.file.contents O primeiro parâmetro é /etc/passwd O segundo parâmetro é utf8	Corresponde: vfs.file.contents[/etc/passwd,utf8] Não corresponde: vfs.file.contents[/etc/passwd,] vfs.file.contents[/etc/passwd,utf16]

3 Triggers

Visão geral

Triggers são expressões lógicas que ”avaliam” os dados coletados pelos items e representam o estado atual do sistema.

Enquanto os items são usados para coletar dados do sistema, é altamente impraticável acompanhar esses dados o tempo todo esperando por uma condição que seja alarmante ou mereça atenção. O trabalho de ”avaliar” os dados pode ser deixado para expressões de trigger.

As expressões de trigger permitem definir um limite de qual estado dos dados é ”aceitável”. Portanto, caso os dados recebidos

ultrapassem o estado aceitável, uma trigger é ”disparada” - ou muda seu status para PROBLEM.

Uma trigger pode ter os seguintes status:

Status	Descrição
OK	Este é o status normal da trigger.
Problem	Algo aconteceu. Por exemplo, a carga do processador está muito alta.
Unknown	O valor da trigger não pode ser calculado. Veja Status Unknown.

Em uma trigger simples, podemos querer definir um limite para uma média de cinco minutos de alguns dados, por exemplo, a carga da CPU. Isso é feito definindo uma expressão de trigger onde:

- a função ’avg’ é aplicada ao valor recebido na chave do item
- um período de cinco minutos para avaliação é usado
- um limite de ’2’ é definido

```
avg(/host/key,5m)>2
```

Esta trigger será ”disparada” (se tornará PROBLEM) se a média de cinco minutos for *maior* que 2.

Em uma trigger mais complexa, a expressão pode incluir uma **combinação** de múltiplas funções e múltiplos limites. Veja também: Expressão de trigger.

Triggers não podem ser criadas para items que retornam valores com o tipo de dado binário.

Note:
Após habilitar um trigger (alterando seu status de configuração de *Desabilitado* para *Habilitado*), a expressão do trigger é avaliada assim que um item nela recebe um valor ou chega o momento de processar as funções de data e hora e/ou nodata().

Most trigger functions are evaluated based on item value history data, while some trigger functions for long-term analytics, e.g. **trendavg()**, **trendcount()**, etc, use trend data.

Tempo de cálculo

Um trigger é recalculado toda vez que o Zabbix server recebe um novo valor que faz parte da expressão. Quando um novo valor é recebido, cada função incluída na expressão é recalculada (não apenas aquela que recebeu o novo valor).

Além disso, um trigger é recalculado toda vez que um novo valor é recebido **e** a cada 30 segundos se as funções de data e hora e/ou nodata() forem usadas na expressão.

As funções de data e hora e/ou nodata() são recalculadas a cada 30 segundos pelo processo history syncer do Zabbix.

Triggers que referenciam **apenas** funções de tendência são avaliados uma vez por o menor período de tempo na expressão. Veja também funções de tendência.

Período de avaliação

Um período de avaliação é usado em funções que fazem referência ao histórico do item. Ele permite especificar o intervalo de tempo que nos interessa. Pode ser especificado como um período de tempo (30s, 10m, 1h) ou como um intervalo de valores (#5 - para os cinco valores mais recentes).

O período de avaliação é medido até o ”agora” - onde ”agora” é o último tempo de recálculo do trigger (veja Tempo de cálculo acima); ”agora” não é o tempo ”agora” do server.

O período de avaliação especifica:

- Considerar todos os valores entre ”agora-período de tempo” e ”agora” (ou, com deslocamento de tempo, entre ”agora-deslocamento de tempo-período de tempo” e ”agora-deslocamento de tempo”)
- Considerar no máximo a quantidade de valores definida pelo parâmetro num do passado, até ”agora”
  - Se não houver valores disponíveis para o período de tempo ou quantidade especificada, o trigger ou item calculado que usa essa função se tornará não suportado

Observe que:

- Se apenas uma única função (referenciando o histórico de dados) for usada no trigger, ”agora” é sempre o valor mais recente recebido. Por exemplo, se o último valor foi recebido há uma hora, o período de avaliação será considerado até o valor mais recente de uma hora atrás.

- Um novo trigger é calculado assim que o primeiro valor é recebido (funções de histórico); ele será calculado em até 30 segundos para as funções de data e hora e nodata(). Assim, o trigger será calculado mesmo que talvez o período de avaliação definido (por exemplo, uma hora) ainda não tenha passado desde que o trigger foi criado. O trigger também será calculado após o primeiro valor, mesmo que o intervalo de avaliação tenha sido definido, por exemplo, para os dez valores mais recentes.

Status desconhecido

É possível que um operando desconhecido apareça em uma expressão de trigger se:

- um item não suportado for usado
- a avaliação da função para um item suportado resultar em erro

Neste caso, uma trigger geralmente é avaliada como ”desconhecida” (embora haja algumas exceções). Para mais detalhes, veja Expressões com operandos desconhecidos.

É possível ser notificado sobre triggers desconhecidas.

1 Configurando um trigger

Visão geral

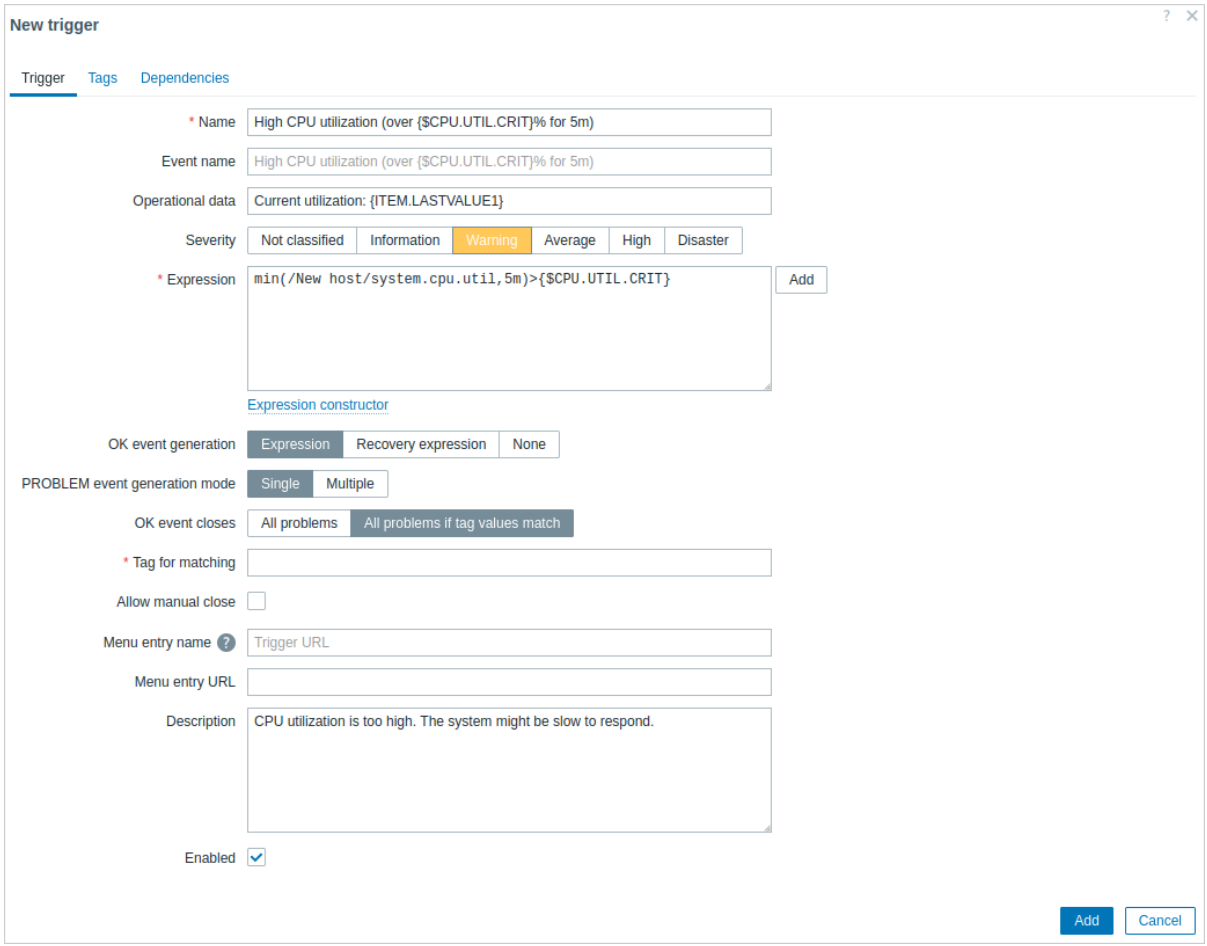
Para configurar um trigger, faça o seguinte:

- Vá para: *Coleta de dados* > *Hosts*
- Clique em *Triggers* na linha do host
- Clique em *Criar trigger* à direita (ou no nome do trigger para editar um trigger existente)
- Insira os parâmetros do trigger no formulário

Veja também informações gerais sobre triggers e seus tempos de cálculo.

Configuração

A aba **Trigger** contém todos os atributos essenciais do trigger.

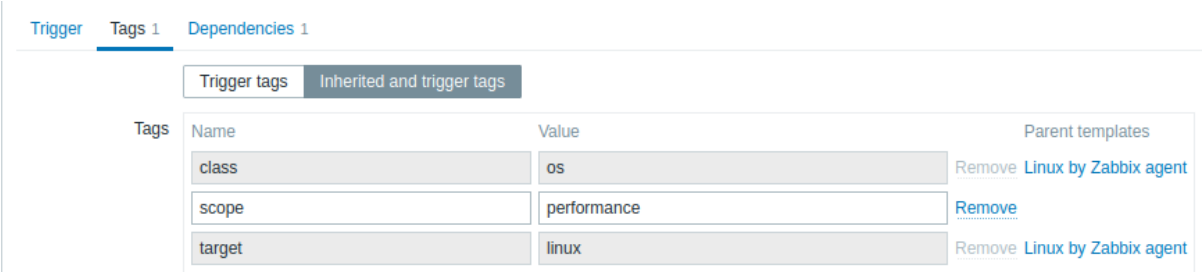


Todos os campos obrigatórios estão marcados com um asterisco vermelho.

Parâmetro	Descrição
Nome	Nome do trigger. As macros suportadas são: {HOST.HOST}, {HOST.NAME}, {HOST.PORT}, {HOST.CONN}, {HOST.DNS}, {HOST.IP}, {ITEM.VALUE}, {ITEM.LASTVALUE}, {ITEM.LOG.*} e macros de usuário {$MACRO}. As macros **$1, $2...$9** podem ser usadas para se referir à primeira, segunda...nona constante da expressão. *Nota*: as macros $1-$9 serão resolvidas corretamente se referenciando constantes em expressões relativamente simples e diretas. Por exemplo, o nome ”Carga do processador acima de $1 em {HOST.NAME}” mudará automaticamente para ”Carga do processador acima de 5 em Novo host” se a expressão for last(/Novo host/system.cpu.load[percpu,avg1])>5.
Nome do evento	Quando definido, este nome será usado para criar o nome do evento de problema. Por padrão, o nome do evento é o mesmo que o nome do trigger. O nome do evento pode ser usado para construir alertas significativos contendo dados do problema (veja exemplo). O mesmo conjunto de macros é suportado como no nome do trigger, além das macros de expressão {TIME} e {?EXPRESSION}.
Dados operacionais	Os dados operacionais permitem definir strings arbitrárias junto com macros. As macros serão resolvidas dinamicamente para dados em tempo real em *Monitoramento* > *Problemas*. Enquanto as macros no nome do trigger (veja acima) serão resolvidas para seus valores no momento em que um problema ocorre e se tornarão a base de um nome de problema estático, as macros nos dados operacionais mantêm a capacidade de exibir as informações mais recentes dinamicamente. Se nenhum dado operacional for configurado no nível do trigger, os valores mais recentes de todos os items da expressão serão exibidos. O mesmo conjunto de macros é suportado como no nome do trigger.
Severidade	Defina a severidade do trigger clicando nos botões.
Expressão	Expressão lógica usada para definir as condições de um problema. Sufixos de tempo e sufixos de tamanho de memória são suportados. Um problema é criado após todas as condições incluídas na expressão serem atendidas, ou seja, a expressão é avaliada como VERDADEIRA. O problema será resolvido assim que a expressão for avaliada como FALSA, a menos que condições adicionais de recuperação sejam especificadas em *Expressão de recuperação*.
Geração de evento OK	Opções de geração de evento OK: **Expressão** - eventos OK são gerados com base na mesma expressão que os eventos de problema; **Expressão de recuperação** - eventos OK são gerados se a expressão de problema for avaliada como FALSA e a expressão de recuperação for avaliada como VERDADEIRA; **Nenhum** - neste caso, o trigger nunca retornará ao estado OK por conta própria.
Expressão de recuperação	Expressão lógica (opcional) definindo condições adicionais que devem ser atendidas antes que o problema seja resolvido, após a expressão de problema original já ter sido avaliada como FALSA. A expressão de recuperação é útil para histerese de trigger. **Não** é possível resolver um problema apenas pela expressão de recuperação se a expressão de problema ainda for VERDADEIRA. Este campo só está disponível se ’Expressão de recuperação’ for selecionada para *Geração de evento OK*.
Modo de geração de evento PROBLEM	Modo para gerar eventos de problema: **Único** - um único evento é gerado quando um trigger entra no estado ’Problema’ pela primeira vez; **Múltiplo** - um evento é gerado a cada avaliação ’Problema’ do trigger.
Evento OK fecha	Selecione se o evento OK fecha: **Todos os problemas** - todos os problemas deste trigger; **Todos os problemas se os valores das tags coincidirem** - apenas aqueles problemas do trigger com valores de tag de evento correspondentes.
Tag para correspondência	Digite o nome da tag do evento a ser usada para correlação de eventos. Este campo é exibido se ’Todos os problemas se os valores das tags coincidirem’ for selecionado para a propriedade *Evento OK fecha* e é obrigatório neste caso.
Permitir fechamento manual	Marque para permitir o fechamento manual de eventos de problema gerados por este trigger. O fechamento manual é possível ao reconhecer eventos de problema.
Nome da entrada do menu	Se não estiver vazio, o nome inserido aqui (até 64 caracteres) é usado em vários locais do frontend como um rótulo para a URL do trigger especificada no parâmetro *URL da entrada do menu*. Se estiver vazio, o nome padrão *Trigger URL* será usado. O mesmo conjunto de macros é suportado como no nome do trigger, além de {EVENT.ID}, {HOST.ID} e {TRIGGER.ID}.

Parâmetro	Descrição
URL da entrada do menu	Se não estiver vazio, a URL inserida aqui (até 2048 caracteres) estará disponível como um link no menu de eventos em vários locais do frontend, por exemplo, ao clicar no nome do problema em *Monitoramento* > *Problemas* ou no widget de painel *Problemas*. O mesmo conjunto de macros é suportado como no nome do trigger, além de {EVENT.ID}, {HOST.ID} e {TRIGGER.ID}. Nota: macros de usuário com valores secretos não serão resolvidas na URL.
Descrição	Campo de texto usado para fornecer mais informações sobre este trigger. Pode conter instruções para corrigir um problema específico, detalhes de contato do responsável, etc. O mesmo conjunto de macros é suportado como no nome do trigger.
Habilitado	Desmarcar esta caixa desabilitará o trigger, se necessário. Problemas de um trigger desabilitado não são mais exibidos no frontend, mas não são excluídos.

A aba **Tags** permite definir tags no nível do trigger. Todos os problemas deste trigger serão marcados com os valores inseridos aqui.



Além disso, a opção *Tags herdadas e do trigger* permite visualizar tags definidas no nível do template se o trigger vier desse template. Se houver vários templates com a mesma tag, essas tags são exibidas uma vez e os nomes dos templates são separados por vírgulas. Um trigger não "herda" e exibe tags no nível do host.

Parâmetro	Descrição
Nome/Valor	Defina tags personalizadas para marcar eventos do trigger. As tags são um par de nome e valor. Você pode usar apenas o nome ou emparelhá-lo com um valor. Um trigger pode ter várias tags com o mesmo nome, mas valores diferentes. Macros de usuário, macros de usuário com contexto, macros de descoberta de baixo nível e funções de macro com `{{ITEM.VALUE}}`, `{{ITEM.LASTVALUE}}` são suportadas em tags de evento. Macros de descoberta de baixo nível podem ser usadas dentro do contexto da macro. A macro {TRIGGER.ID} é suportada em valores de tag de trigger. Pode ser útil para identificar triggers criados a partir de protótipos de trigger e, por exemplo, suprimir problemas desses triggers durante a manutenção. Se o comprimento total do valor expandido exceder 255, ele será cortado para 255 caracteres. Veja todas as macros suportadas para tags de evento. Tags de evento podem ser usadas para correlação de eventos, em condições de ação e também serão vistas em *Monitoramento* > *Problemas* ou no widget *Problemas*.

A aba **Dependências** contém todas as dependências do trigger.

Clique em *Adicionar* para adicionar uma nova dependência.

> **Note:**
> Você também pode configurar um trigger abrindo um existente, pressionando o botão *Clonar* e depois salvando com um nome diferente.

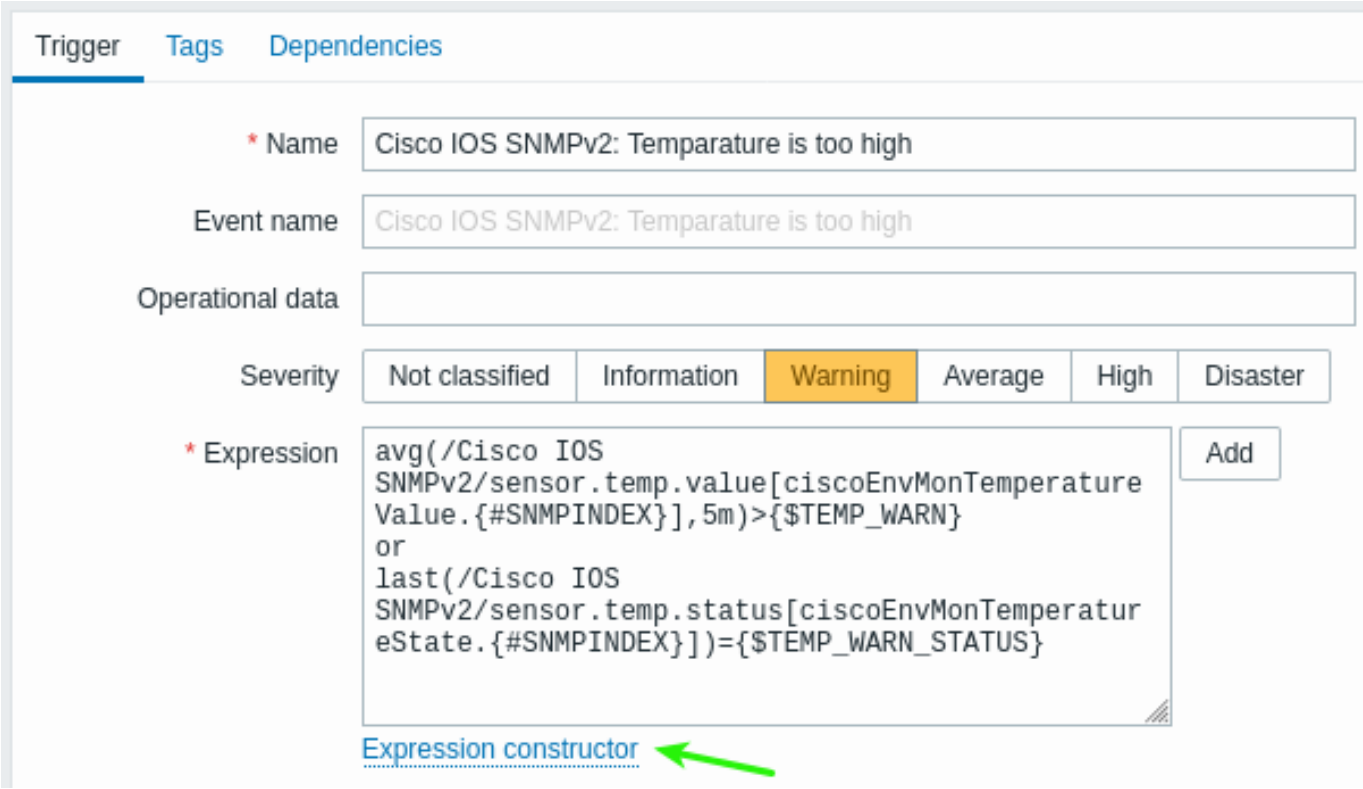
Testando expressões

É possível testar a expressão de trigger configurada para verificar qual seria o resultado da expressão dependendo do valor recebido.

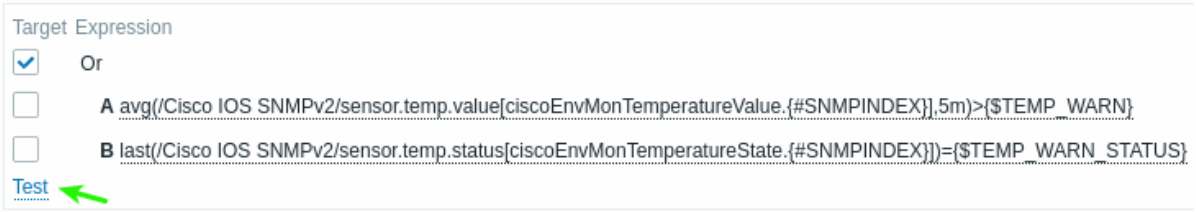
A seguinte expressão de um template oficial é usada como exemplo:

```
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}
```

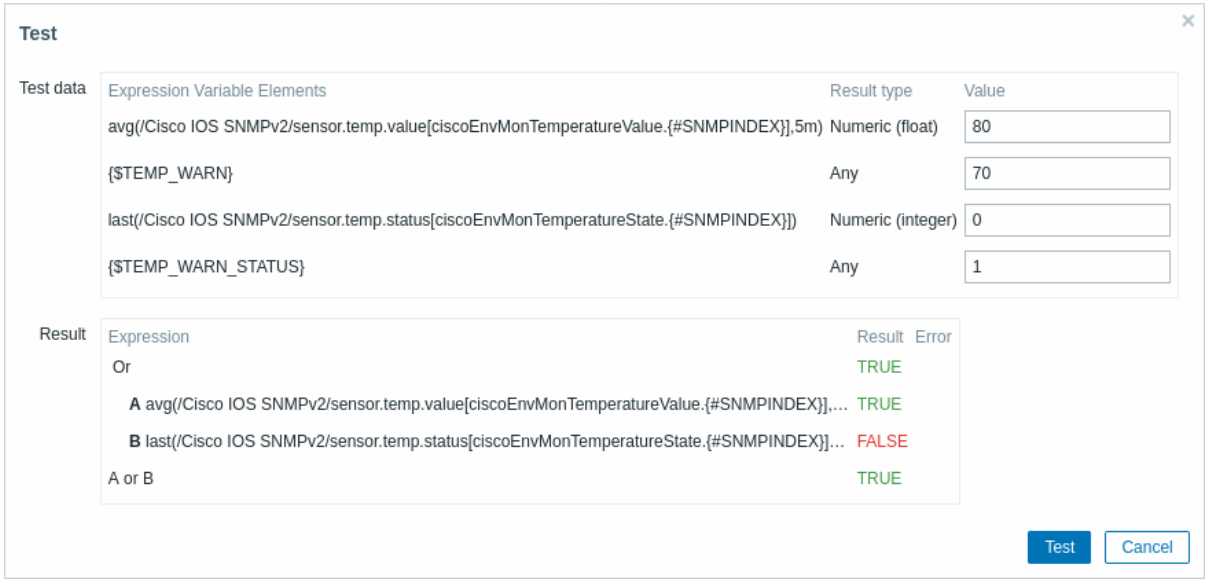
Para testar a expressão, clique em *Construtor de expressão* abaixo do campo de expressão.



No Construtor de expressão, todas as expressões individuais são listadas. Para abrir a janela de teste, clique em *Testar* abaixo da lista de expressões.



Na janela de teste, você pode inserir valores de exemplo (’80’, ’70’, ’0’, ’1’ neste exemplo) e então ver o resultado da expressão, clicando no botão *Testar*.



O resultado das expressões individuais, bem como da expressão completa, pode ser visto.

”TRUE” significa que a expressão especificada está correta. Neste caso específico A, ”80” é maior que o valor especificado em {$TEMP_WARN}, ”70” neste exemplo. Como esperado, aparece um resultado ”TRUE”.

”FALSE” significa que a expressão especificada está incorreta. Neste caso específico B, {$TEMP_WARN_STATUS} ”1” precisa ser

igual ao valor especificado, ”0” neste exemplo. Como esperado, aparece um resultado ”FALSE”.

O tipo de expressão escolhido é ”OR”. Se pelo menos uma das condições especificadas (A ou B neste caso) for TRUE, o resultado geral também será TRUE. Isso significa que o valor atual excede o valor de aviso e um problema ocorreu.

2 Expressão de trigger

Visão geral

As expressões usadas em triggers são muito flexíveis. Você pode usá-las para criar testes lógicos complexos em relação às estatísticas monitoradas.

Uma expressão simples usa uma **função** que é aplicada ao item com alguns parâmetros. A função retorna um resultado que é comparado ao limite, usando um operador e uma constante.

A sintaxe de uma expressão simples e útil é `function(/host/key,parameter)<operator><constant>`.

Por exemplo:

```
min(/Zabbix server/net.if.in[eth0,bytes],5m)>100K
```

será disparado se o número de bytes recebidos durante os últimos cinco minutos estiver sempre acima de 100 kilobytes.

Embora a sintaxe seja exatamente a mesma, do ponto de vista funcional existem dois tipos de expressões de trigger:

- expressão de problema - define as condições do problema
- expressão de recuperação (opcional) - define condições adicionais para a resolução do problema

Ao definir apenas uma expressão de problema, essa expressão será usada tanto como limite de problema quanto como limite de recuperação do problema. Assim que a expressão de problema for avaliada como VERDADEIRA, há um problema. Assim que a expressão de problema for avaliada como FALSA, o problema é resolvido.

Ao definir tanto a expressão de problema quanto a expressão de recuperação suplementar, a resolução do problema se torna mais complexa: não apenas a expressão de problema deve ser FALSA, mas também a expressão de recuperação deve ser VERDADEIRA. Isso é útil para criar histerese e evitar flapping de triggers.

> **Note:**
> É improdutivo usar a macro {TRIGGER.VALUE} em uma expressão de recuperação porque essa expressão só é avaliada quando o trigger está no estado ”Problema”. Consequentemente, {TRIGGER.VALUE} sempre será resolvido como ”1” (que indica um estado de ”Problema”) ao avaliar a expressão.

Funções

As funções permitem calcular os valores coletados (média, mínimo, máximo, soma), encontrar strings, referenciar o tempo atual e outros fatores.

Uma lista completa das funções suportadas está disponível.

Normalmente, as funções retornam valores numéricos para comparação. Ao retornar strings, a comparação é possível com os operadores **=** e **<>** (veja o exemplo).

Parâmetros da função

Os parâmetros da função permitem especificar:

- host e chave do item (funções que referenciam apenas o histórico do item do host)
- parâmetros específicos da função
- outras expressões (não disponível para funções que referenciam apenas o histórico do item do host, veja outras expressões para exemplos)

O host e a chave do item podem ser especificados como `/host/key`.
 A omissão do nome do host no primeiro parâmetro (ou seja, como em `function(//key,parameter,...)`) só é suportada em certos contextos:

- Na fórmula de items calculados
- Em macros de expressão, que podem ser usadas em:
  - O campo *Nome do evento*
  - Nome do gráfico
  - O rótulo dos elementos de mapa ”Host” e ”Trigger” elementos de mapa

Nesses contextos, você também pode usar a macro `{HOST.HOST}`. `{HOST.HOST<1-9>}` pode ser usado no caso do campo *Nome do evento* e do elemento de mapa ”Trigger” para se referir a um item específico na expressão do trigger. Quando o nome do host é omitido ou substituído por `{HOST.HOST}` nesses contextos, a referência aponta para o primeiro item na expressão do trigger ou

para o primeiro item no gráfico. Fora desses contextos suportados, omitir o nome do host em expressões de trigger resultará em erro. Veja o Exemplo 18 para uma ilustração do uso de barra dupla em macros de Nome do evento.

O item referenciado deve estar em um estado suportado (exceto para a função **nodata()**, que também é calculada para itens não suportados).

Enquanto outras expressões de trigger como parâmetros de função são limitadas a funções sem histórico em triggers, essa limitação não se aplica em items calculados.

Parâmetros específicos da função

Os parâmetros específicos da função são colocados após a chave do item e são separados da chave do item por uma vírgula. Veja as funções suportadas para uma lista completa desses parâmetros.

A maioria das funções numéricas aceita tempo como parâmetro. Você pode usar segundos ou sufixos de tempo para indicar o tempo. Precedido por um sinal de cerquilha, o parâmetro tem um significado diferente:

Expressão	Descrição
sum(/host/key,**10m)**	Soma dos valores nos últimos 10 minutos.
sum(/host/key,**#10)**	Soma dos últimos dez valores.

Parâmetros com um sinal de cerquilha têm um significado diferente com a função **last** - eles denotam o N-ésimo valor anterior, então, dados os valores 30, 70, 20, 60, 50 (do mais recente para o menos recente):

- `last(/host/key,#2)` retornaria '70'
- `last(/host/key,#5)` retornaria '50'

Deslocamento de tempo

Um deslocamento de tempo opcional é suportado com tempo ou contagem de valores como parâmetro da função. Este parâmetro permite referenciar dados de um período de tempo no passado.

O deslocamento de tempo começa com `now` - especificando o tempo atual, e é seguido por `+N<unidade de tempo>` ou `-N<unidade de tempo>` - para adicionar ou subtrair N unidades de tempo.

Por exemplo, `avg(/host/key,1h:now-1d)` retornará o valor médio para uma hora um dia atrás.

Attention:
O deslocamento de tempo especificado em meses (M) e anos (y) é suportado apenas para funções de tendência. Outras funções suportam segundos (s), minutos (m), horas (h), dias (d) e semanas (w).

Deslocamento de tempo com períodos de tempo absolutos

Períodos de tempo absolutos são suportados no parâmetro de deslocamento de tempo, por exemplo, meia-noite à meia-noite para um dia, segunda-feira a domingo para uma semana, primeiro dia ao último dia do mês para um mês.

O deslocamento de tempo para períodos de tempo absolutos começa com `now` - especificando o tempo atual, e é seguido por qualquer número de operações de tempo: `/<unidade de tempo>` - define o início e o fim da unidade de tempo, por exemplo, meia-noite à meia-noite para um dia, `+N<unidade de tempo>` ou `-N<unidade de tempo>` - para adicionar ou subtrair N unidades de tempo.

Observe que o valor do deslocamento de tempo pode ser maior ou igual a 0, enquanto o valor mínimo do período de tempo é 1.

Parâmetro	Descrição
1d:now/d	Ontem
1d:now/d+1d	Hoje
2d:now/d+1d	Últimos 2 dias
1w:now/w	Semana passada
1w:now/w+1w	Esta semana

Outras expressões

Os parâmetros da função podem conter outras expressões, conforme a seguinte sintaxe:

```
min(min(/host/key,1h),min(/host2/key2,1h)*10)
```

Observe que outras expressões não podem ser usadas se a função fizer referência ao histórico do item. Por exemplo, a seguinte sintaxe **não** é permitida:

~~`min(/host/key,#5*10)`~~

Operadores

Os seguintes operadores são suportados para triggers **(em ordem decrescente de prioridade de execução)**:

Prioridade	Operador	Definição	**Notas para valores desconhecidos**	Forçar conversão do operando para float [1]
1	**-**	Menos unário	**-**Desconhecido → Desconhecido	Sim
2	**not**	NOT lógico	**not** Desconhecido → Desconhecido	Sim
3	*	Multiplicação	0 * Desconhecido → Desconhecido (sim, Desconhecido, não 0 - para não perder Desconhecido em operações aritméticas) 1.2 * Desconhecido → Desconhecido	Sim
	/	Divisão	Desconhecido / 0 → erro Desconhecido / 1.2 → Desconhecido 0.0 / Desconhecido → Desconhecido	Sim
4	+	Soma aritmética	1.2 + Desconhecido → Desconhecido	Sim
	-	Subtração aritmética	1.2 - Desconhecido → Desconhecido	Sim
5	<	Menor que. O operador é definido como: A<B ⇔ (A<B-0.000001)	1.2 < Desconhecido → Desconhecido	Sim
	<=	Menor ou igual a. O operador é definido como: A<=B ⇔ (A≤B+0.000001)	Desconhecido <= Desconhecido → Desconhecido	Sim
	>	Maior que. O operador é definido como: A>B ⇔ (A>B+0.000001)		Sim
	>=	Maior ou igual a. O operador é definido como: A>=B ⇔ (A≥B-0.000001)		Sim
6	=	É igual. O operador é definido como: A=B ⇔ (A≥B-0.000001) e (A≤B+0.000001)		Não [1]

Prioridade	Operador	Definição	**Notas para valores desconhecidos**	Forçar conversão do operando para float [1]
	<>	Diferente. O operador é definido como: A<>B ⇔ (A<B-0.000001) ou (A>B+0.000001)		Não [1]
7	**and**	AND lógico	0 **and** Desconhecido → 0 1 **and** Desconhecido → Desconhecido Desconhecido **and** Desconhecido → Desconhecido	Sim
8	**or**	OR lógico	1 **or** Desconhecido → 1 0 **or** Desconhecido → Desconhecido Desconhecido **or** Desconhecido → Desconhecido	Sim

[1] O operando string ainda é convertido para numérico se:

- outro operando for numérico
- operador diferente de **=** ou **<>** for usado em um operando

Sobre operadores relacionais (<, <=, >, >=): se a conversão falhar (por exemplo, em ""), a expressão resulta em Desconhecido (sem fallback para comparação de strings).

Sobre operadores de igualdade (=, <>): se a conversão falhar, os operandos são comparados como strings brutas.

Os operadores **not**, **and** e **or** diferenciam maiúsculas de minúsculas e devem estar em minúsculas. Eles também devem ser cercados por espaços ou parênteses.

Todos os operadores, exceto o **-** unário e o **not**, têm associatividade da esquerda para a direita. O **-** unário e o **not** são não associativos (ou seja, **-(-1)** e **not (not 1)** devem ser usados em vez de **--1** e **not not 1**).

Resultado da avaliação:

- Os operadores **<**, **<=**, **>**, **>=**, **=**, **<>** devem resultar em '1' na expressão do trigger se a relação especificada for verdadeira e '0' se for falsa. Se pelo menos um operando for Desconhecido, o resultado será Desconhecido;
- **and** para operandos conhecidos deve resultar em '1' se ambos os operandos forem diferentes de '0'; caso contrário, resulta em '0'; para operandos desconhecidos, **and** resulta em '0' apenas se um operando for igual a '0'; caso contrário, resulta em 'Desconhecido';
- **or** para operandos conhecidos deve resultar em '1' se qualquer um dos operandos for diferente de '0'; caso contrário, resulta em '0'; para operandos desconhecidos, **or** resulta em '1' apenas se um operando for diferente de '0'; caso contrário, resulta em 'Desconhecido';
- O resultado do operador de negação lógica **not** para um operando conhecido é '0' se o valor do operando for diferente de '0'; '1' se o valor do operando for igual a '0'. Para operando desconhecido, **not** resulta em 'Desconhecido'.

Cache de valores

Os valores necessários para a avaliação do trigger são armazenados em cache pelo Zabbix server. Por causa disso, a avaliação do trigger causa uma carga maior no banco de dados por algum tempo após a reinicialização do server. O cache de valores não é limpo quando os valores do histórico do item são removidos (seja manualmente ou pelo housekeeper), então o server usará os valores em cache até que eles sejam mais antigos do que os períodos de tempo definidos nas funções do trigger ou até que o server seja reiniciado.

Note:
Se não houver dados recentes no cache e não houver um período de consulta definido na função, o Zabbix, por padrão, irá buscar no passado até uma semana para consultar o banco de dados por valores históricos.

Exemplos de triggers

Exemplo 1

A carga do processador está muito alta no Zabbix server.

```
last(/Zabbix server/system.cpu.load[all,avg1])>5
```

Ao usar a função 'last()', estamos referenciando o valor mais recente. /Zabbix server/system.cpu.load[all,avg1] fornece um nome curto para o parâmetro monitorado. Especifica que o host é 'Zabbix server' e a chave monitorada é 'system.cpu.load[all,avg1]'. Finalmente, >5 significa que o trigger está no estado PROBLEM sempre que a medição mais recente da carga do processador do Zabbix server for maior que 5.

Exemplo 2

www.example.com está sobrecarregado.

```
last(/www.example.com/system.cpu.load[all,avg1])>5 or min(/www.example.com/system.cpu.load[all,avg1],10m)>
```

A expressão é verdadeira quando a carga atual do processador é superior a 5 ou a carga do processador foi superior a 2 nos últimos 10 minutos.

Exemplo 3

/etc/passwd foi alterado.

```
last(/www.example.com/vfs.file.cksum[/etc/passwd],#1)<>last(/www.example.com/vfs.file.cksum[/etc/passwd],#
```

A expressão é verdadeira quando o valor anterior do checksum de /etc/passwd difere do mais recente.

Expressões semelhantes podem ser úteis para monitorar alterações em arquivos importantes, como /etc/passwd, /etc/inetd.conf, /kernel, etc.

Exemplo 4

Alguém está baixando um arquivo grande da Internet.

Uso da função min:

```
min(/www.example.com/net.if.in[eth0,bytes],5m)>100K
```

A expressão é verdadeira quando o número de bytes recebidos na eth0 é maior que 100 KB nos últimos 5 minutos.

Exemplo 5

Ambos os nós do servidor SMTP em cluster estão inativos.

Observe o uso de dois hosts diferentes em uma expressão:

```
last(/smtp1.example.com/net.tcp.service[smtp])=0 and last(/smtp2.example.com/net.tcp.service[smtp])=0
```

A expressão é verdadeira quando ambos os servidores SMTP estão inativos em smtp1.example.com e smtp2.example.com.

Exemplo 6

O agent do Zabbix precisa ser atualizado.

Uso da função find():

```
find(/example.example.com/agent.version,,"like","beta8")=1
```

A expressão é verdadeira se o agent do Zabbix tiver a versão beta8.

Exemplo 7

O servidor está inacessível.

```
count(/example.example.com/icmpping,30m,,"0")>5
```

A expressão é verdadeira se o host "example.example.com" estiver inacessível mais de 5 vezes nos últimos 30 minutos.

Exemplo 8

Sem heartbeats nos últimos 3 minutos.

Uso da função nodata():

```
nodata(/example.example.com/tick,3m)=1
```

Para utilizar este trigger, 'tick' deve ser definido como um item trapper do Zabbix. O host deve enviar periodicamente dados para este item usando o zabbix_sender. Se nenhum dado for recebido em 180 segundos, o valor do trigger se tornará PROBLEMA.

Observe que 'nodata' pode ser usado para qualquer tipo de item.

Exemplo 9

Atividade da CPU durante a noite.

Uso da função time():

```
min(/Zabbix server/system.cpu.load[all,avg1],5m)>2 and time()<060000
```

O trigger pode mudar seu estado para problema apenas durante a noite (00:00 - 06:00).

Exemplo 10

Atividade da CPU a qualquer momento, com exceção.

Uso da função time() e do operador **not**:

```
min(/zabbix/system.cpu.load[all,avg1],5m)>2
and not (dayofweek()=7 and time()>230000)
and not (dayofweek()=1 and time()<010000)
```

O trigger pode mudar seu estado para problema a qualquer momento, exceto por 2 horas na mudança da semana (domingo, 23:00 - segunda-feira, 01:00).

Exemplo 11

Verifique se a hora local do cliente está sincronizada com a hora do servidor Zabbix.

Uso da função fuzzytime():

```
fuzzytime(/MySQL_DB/system.localtime,10s)=0
```

O trigger mudará para o estado de problema caso a hora local no servidor MySQL_DB e no servidor Zabbix difira por mais de 10 segundos. Observe que 'system.localtime' deve ser configurado como uma verificação passiva para o agent Zabbix; no agent Zabbix 2 pode ser configurado como uma verificação ativa.

Exemplo 12

Comparando a carga média de hoje com a carga média do mesmo horário de ontem (usando deslocamento de tempo como `now-1d`).

```
avg(/server/system.cpu.load,1h)/avg(/server/system.cpu.load,1h:now-1d)>2
```

O trigger será disparado se a carga média da última hora ultrapassar a carga média da mesma hora de ontem em mais de duas vezes.

Exemplo 13

Usando o valor de outro item para obter um limite de trigger:

```
last(/Template PfSense/hrStorageFree[{#SNMPVALUE}])<last(/Template PfSense/hrStorageSize[{#SNMPVALUE}])*0.
```

O trigger será acionado se o armazenamento livre cair abaixo de 10 por cento.

Exemplo 14

Usando resultado da avaliação para obter o número de triggers acima de um limite:

```
(last(/server1/system.cpu.load[all,avg1])>5) + (last(/server2/system.cpu.load[all,avg1])>5) + (last(/serve
```

O trigger será disparado se pelo menos dois dos triggers na expressão estiverem em estado de problema.

Exemplo 15

Comparando valores de string de dois items - os operandos aqui são funções que retornam strings.

Problema: criar um alerta se a versão do Ubuntu for diferente em hosts diferentes

```
last(/NY Zabbix server/vfs.file.contents[/etc/os-release])<>last(/LA Zabbix server/vfs.file.contents[/etc/
```

Exemplo 16

Comparando dois valores de string - os operandos são:

- uma função que retorna uma string
- uma combinação de macros e strings

Problema: detectar alterações na consulta DNS

A chave do item é:

```
net.dns.record[192.0.2.0,{$WEBSITE_NAME},{$DNS_RESOURCE_RECORD_TYPE},2,1]
```

com macros definidas como

```
{$WEBSITE_NAME} = example.com
{$DNS_RESOURCE_RECORD_TYPE} = MX
```

e normalmente retorna:

```
example.com           MX       0 mail.example.com
```

Portanto, nossa expressão de trigger para detectar se o resultado da consulta DNS se desviou do resultado esperado é:

```
last(/Zabbix server/net.dns.record[192.0.2.0,{$WEBSITE_NAME},{$DNS_RESOURCE_RECORD_TYPE},2,1])<>"{$WEBSITE
```

Observe as aspas ao redor do segundo operando.

Exemplo 17

Comparando dois valores de string - os operandos são:

- uma função que retorna uma string
- uma constante de string com caracteres especiais \ e ”

Problema: detectar se o conteúdo do arquivo `/tmp/hello` é igual a:

```
\" //hello ?\"
```

Opção 1. Escreva a string diretamente:

```
last(/Zabbix server/vfs.file.contents[/tmp/hello])="\\\" //hello ?\\\""
```

Observe como os caracteres \ e ” são escapados quando a string é comparada diretamente.

Opção 2. Use uma macro

```
{$HELLO_MACRO} = \" //hello ?\"
```

na expressão:

```
last(/Zabbix server/vfs.file.contents[/tmp/hello])={$HELLO_MACRO}
```

Exemplo 18

Comparando períodos de longo prazo.

Problema: A carga do servidor Exchange aumentou mais de 10% no mês passado

```
trendavg(/Exchange/system.cpu.load,1M:now/M)>1.1*trendavg(/Exchange/system.cpu.load,1M:now/M-1M)
```

Você também pode usar o campo Nome do evento na configuração do trigger para construir uma mensagem de alerta significativa, por exemplo, para receber algo como

```
"A carga do servidor Exchange aumentou em 24% em julho (0,69) em comparação com junho (0,56)"
```

o nome do evento deve ser definido como:

```
Load of {HOST.HOST} server increased by {{?100*trendavg(//system.cpu.load,1M:now/M)/trendavg(//system.cpu.
```

Também é útil permitir o fechamento manual na configuração do trigger para esse tipo de problema.

Você tem um exemplo de trigger expressions que pode ser útil para outras pessoas? Utilize o formulário Example suggestion form para enviá-lo aos desenvolvedores do Zabbix.

Histerese

Às vezes, é necessário um intervalo entre os estados de problema e recuperação, em vez de um simples limiar. Por exemplo, se quisermos definir um trigger que relate um problema quando a temperatura da sala do servidor ultrapassar 20°C e quisermos que ele permaneça no estado de problema até que a temperatura caia abaixo de 15°C, um simples limiar de trigger em 20°C não será suficiente.

Em vez disso, precisamos definir primeiro uma expressão de trigger para o evento de problema (temperatura acima de 20°C). Em seguida, precisamos definir uma condição adicional de recuperação (temperatura abaixo de 15°C). Isso é feito definindo um parâmetro adicional *Expressão de recuperação* ao definir um trigger.

Neste caso, a recuperação do problema ocorrerá em duas etapas:

- Primeiro, a expressão de problema (temperatura acima de 20°C) terá que ser avaliada como FALSA
- Em segundo lugar, a expressão de recuperação (temperatura abaixo de 15°C) terá que ser avaliada como VERDADEIRA

A expressão de recuperação será avaliada somente quando o evento de problema for resolvido primeiro.

Warning:
A expressão de recuperação sendo VERDADEIRA sozinha não resolve um problema se a expressão de problema ainda for VERDADEIRA!

Exemplo 1

A temperatura na sala do servidor está muito alta.

Expressão do problema:

```
last(/server/temp)>20
```

Expressão de recuperação:

```
last(/server/temp)<=15
```

Exemplo 2

O espaço livre em disco está muito baixo.

Expressão do problema: é menor que 10GB nos últimos 5 minutos

```
max(/server/vfs.fs.size[/,free],5m)<10G
```

Expressão de recuperação: é maior que 40GB nos últimos 10 minutos

```
min(/server/vfs.fs.size[/,free],10m)>40G
```

Expressões com operandos desconhecidos

Geralmente, um operando desconhecido (como um item não suportado) na expressão fará com que o valor do trigger seja imediatamente definido como `Desconhecido`.

No entanto, em alguns casos, operandos desconhecidos (itens não suportados, erros de função) são admitidos na avaliação da expressão:

- A função `nodata()` é avaliada independentemente de o item referenciado ser suportado ou não.
- Expressões lógicas com OR e AND podem ser avaliadas para valores conhecidos em dois casos, independentemente de operandos desconhecidos:
  - **Caso 1**: "`1 or alguma_função(item_não_suportado1) or alguma_função(item_não_suportado2) or ...`" pode ser avaliado para um resultado conhecido ('1' ou "Problema"),
  - **Caso 2**: "`0 and alguma_função(item_não_suportado1) and alguma_função(item_não_suportado2) and ...`" pode ser avaliado para um resultado conhecido ('0' ou "OK").
 O Zabbix tenta avaliar tais expressões lógicas considerando itens não suportados como operandos desconhecidos. Nos dois casos acima, um valor conhecido será produzido ("Problema" ou "OK", respectivamente); em **todos os outros** casos, o trigger será avaliado como `Desconhecido`.
- Se a avaliação da função para um item suportado resultar em erro, o valor da função se torna `Desconhecido` e ela participa como operando desconhecido em avaliações posteriores da expressão.

Observe que operandos desconhecidos podem "desaparecer" apenas em expressões lógicas, conforme descrito acima. Em expressões aritméticas, operandos desconhecidos sempre levam ao resultado `Desconhecido` (exceto divisão por 0).

Attention:
Uma expressão que resulta em `Desconhecido` não altera o estado do trigger ("Problema/OK"). Portanto, se estava em "Problema" (veja o Caso 1), permanece no mesmo estado de problema mesmo que a parte conhecida seja resolvida ('1' se torne '0'), porque a expressão agora é avaliada como `Desconhecido` e isso não altera o estado do trigger.

Se uma expressão de trigger com vários itens não suportados for avaliada como `Desconhecido`, a mensagem de erro no frontend se refere ao último item não suportado avaliado.

3 Dependências de trigger

Visão geral

Às vezes, a disponibilidade de um host depende de outro. Um servidor que está atrás de um roteador ficará inacessível se o roteador cair. Com triggers configurados para ambos, você pode receber notificações sobre dois hosts inativos - enquanto apenas o roteador era o culpado.

É aqui que alguma dependência entre hosts pode ser útil. Com a dependência definida, as notificações dos dependentes podem ser retidas e apenas a notificação do problema raiz enviada.

Embora o Zabbix não suporte dependências entre hosts diretamente, elas podem ser definidas com outro método, mais flexível - dependências de trigger. Um trigger pode ter um ou mais triggers dos quais depende.

Assim, em nosso exemplo simples, abrimos o formulário de configuração do trigger do servidor e definimos que ele depende do respectivo trigger do roteador. Com essa dependência, o trigger do servidor não mudará seu estado enquanto o trigger do qual depende estiver no estado 'PROBLEM' - e, assim, nenhuma ação dependente será tomada e nenhuma notificação será enviada.

Se tanto o servidor quanto o roteador estiverem inativos, e uma dependência entre o trigger "servidor inativo" e o trigger "roteador inativo" for definida, o Zabbix não executará ações para o trigger dependente.

Enquanto o trigger pai estiver no estado PROBLEM, seus dependentes podem relatar valores que não podem ser confiáveis. Portanto, os triggers dependentes não serão reavaliados até que o trigger pai (o roteador no exemplo acima):

- volte do estado 'PROBLEM' para 'OK';
- mude seu estado de 'PROBLEM' para 'UNKNOWN';
- seja fechado manualmente, por correlação ou com a ajuda das funções de data e hora e/ou nodata();
- seja resolvido por um valor de um item não envolvido no trigger dependente;
- seja desabilitado, tenha um item desabilitado ou um host de item desabilitado

Em todos os casos mencionados acima, o trigger dependente (servidor) será reavaliado apenas quando uma nova métrica para ele for recebida. Isso significa que o trigger dependente pode não ser atualizado imediatamente.

Além disso:

- A dependência de trigger pode ser adicionada de qualquer trigger de host para qualquer outro trigger de host, desde que não resulte em uma dependência circular.
- A dependência de trigger pode ser adicionada de um template para outro. Se algum trigger do template A depender de algum trigger do template B, o template A só pode ser vinculado a um host (ou outro template) junto com o template B, mas o template B pode ser vinculado a um host (ou outro template) sozinho.
- A dependência de trigger pode ser adicionada de um trigger de template para um trigger de host. Nesse caso, vincular esse template a um host criará um trigger de host que depende do mesmo trigger de template do qual o trigger estava dependendo. Isso permite, por exemplo, ter um template onde alguns triggers dependem dos triggers do roteador (host). Todos os hosts vinculados a esse template dependerão desse roteador específico.
- A dependência de trigger não pode ser adicionada de um trigger de host para um trigger de template.
- A dependência de trigger pode ser adicionada de um protótipo de trigger para outro protótipo de trigger (dentro da mesma regra de descoberta de baixo nível) ou para um trigger real. Um protótipo de trigger não pode depender de um protótipo de trigger de uma regra LLD diferente ou de um trigger criado a partir de um protótipo de trigger. Um protótipo de trigger de host não pode depender de um trigger de um template.

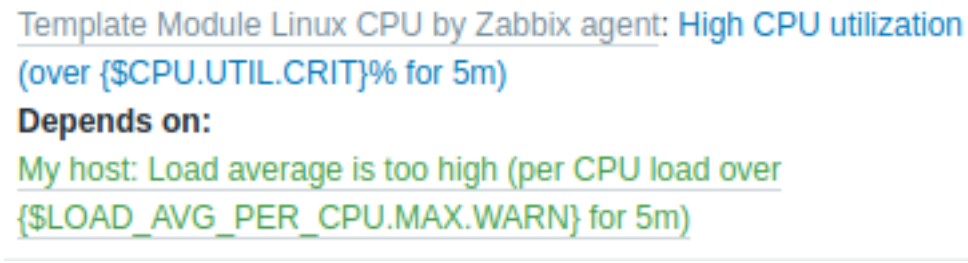
Configuração

Para definir uma dependência:

1. Abra a aba *Dependências* no formulário de configuração do trigger.
2. Clique em *Adicionar* na seção *Dependências* e selecione um ou mais triggers dos quais o trigger irá depender.

3. Clique em *Atualizar*.

Agora o trigger tem a indicação de sua dependência na lista.



Exemplo de várias dependências

Por exemplo, o Host está atrás do Router2 e o Router2 está atrás do Router1.

```
Zabbix - Router1 - Router2 - Host
```

Se o Router1 estiver inativo, então obviamente o Host e o Router2 também estarão inacessíveis, ainda assim, receber três notificações sobre o Host, o Router1 e o Router2 estarem todos inativos é excessivo.

Portanto, neste caso, definimos duas dependências:

```
o trigger 'Host está inativo' depende do trigger 'Router2 está inativo'
o trigger 'Router2 está inativo' depende do trigger 'Router1 está inativo'
```

Antes de alterar o status do trigger 'Host está inativo', o Zabbix irá verificar as dependências de trigger correspondentes. Se tais dependências forem encontradas e um desses triggers estiver no estado 'Problema', então o status do trigger não será alterado, as ações não serão executadas e nenhuma notificação será enviada.

O Zabbix realiza essa verificação de forma recursiva. Se o Router1 ou o Router2 estiverem inacessíveis, o trigger do Host não será atualizado.

4 Gravidade do trigger

A gravidade do trigger representa o nível de importância de um trigger.

O Zabbix suporta as seguintes gravidades de trigger por padrão.

Gravidade	Cor	Descrição
Não classificado	Cinza	Pode ser usada quando o nível de gravidade de um evento é desconhecido, não foi determinado, não faz parte do escopo regular de monitoramento, etc., por exemplo, durante a configuração inicial, como um espaço reservado para avaliação futura ou como parte de um processo de integração.
Informação	Azul claro	Pode ser usada para eventos informativos que não requerem atenção imediata, mas que ainda podem fornecer informações valiosas.
Aviso	Amarelo	Pode ser usada para indicar um possível problema que pode exigir investigação ou ação, mas que não é crítico.
Média	Laranja	Pode ser usada para indicar um problema significativo que deve ser resolvido relativamente em breve para evitar problemas adicionais.
Alta	Vermelho claro	Pode ser usada para indicar problemas críticos que precisam de atenção imediata para evitar interrupções significativas.
Desastre	Vermelho	Pode ser usada para indicar um incidente grave que requer ação imediata para evitar, por exemplo, paralisações do sistema ou perda de dados.

Note:
Os nomes e cores das gravidades dos triggers podem ser personalizados.

As gravidades dos triggers são usadas para:

- representação visual dos triggers - cores diferentes para diferentes gravidades;
- áudio em alarmes globais - áudios diferentes para diferentes gravidades;
- mídia do usuário - diferentes mídias (canal de notificação) para diferentes gravidades (por exemplo, SMS para triggers de gravidade *Alta* e *Desastre*, e E-mail para triggers de outras gravidades);
- limitar ações por condições baseadas nas gravidades dos triggers.

5 Personalizando as severidades dos triggers

Os nomes e cores das severidades dos triggers para elementos relacionados à severidade na interface gráfica podem ser configurados em *Administração → Geral → Opções de exibição de triggers*. As cores são compartilhadas entre todos os temas da interface.

Traduzindo nomes personalizados de severidade

> **Attention:**
> Se as traduções do frontend do Zabbix forem usadas, os nomes personalizados de severidade substituirão os nomes traduzidos por padrão.

Os nomes padrão de severidade de trigger estão disponíveis para tradução em todos os idiomas. Se um nome de severidade for alterado, um nome personalizado será usado em todos os idiomas e uma tradução manual adicional será necessária.

Procedimento para tradução de nomes personalizados de severidade:

- defina o nome personalizado de severidade necessário, por exemplo, 'Importante'
- edite <frontend_dir>/locale/<required_locale>/LC_MESSAGES/frontend.po
- adicione 2 linhas:

```
msgid "Importante"
msgstr "<string de tradução>"
```

e salve o arquivo.

- crie os arquivos .mo conforme descrito em <frontend_dir>/locale/README

Aqui **msgid** deve corresponder ao novo nome personalizado de severidade e **msgstr** deve ser a tradução para ele no idioma específico.

Este procedimento deve ser realizado após cada alteração no nome da severidade.

6 Atualização em massa

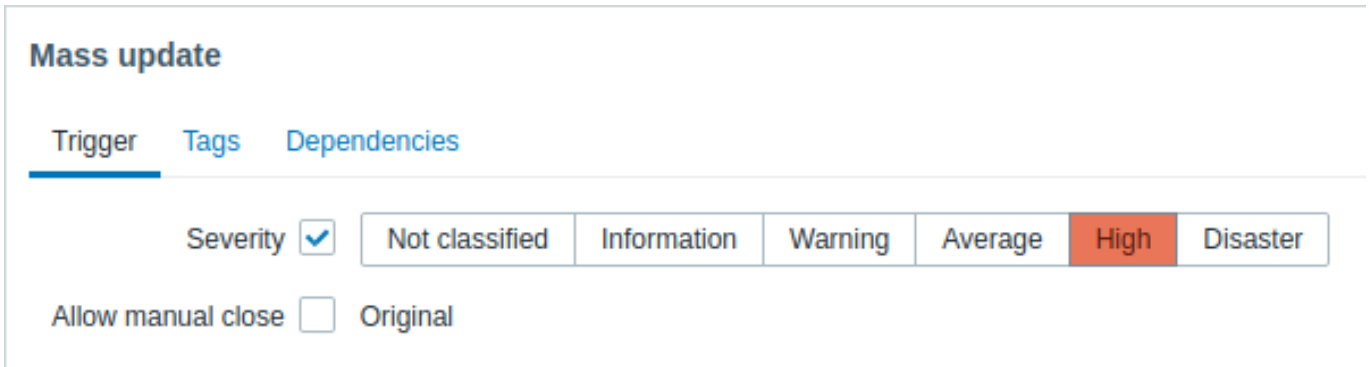
Visão geral

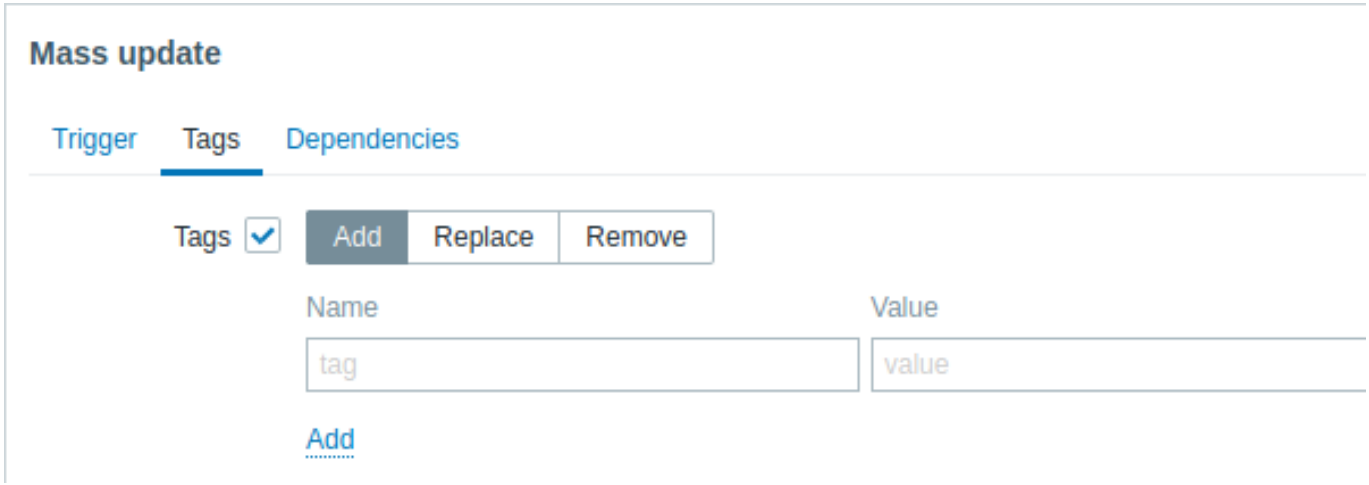
Com a atualização em massa, você pode alterar alguns atributos de vários triggers de uma só vez, economizando o trabalho de abrir cada trigger individualmente para edição.

Usando atualização em massa

Para atualizar vários triggers em massa, faça o seguinte:

- Marque as caixas de seleção dos triggers que deseja atualizar na lista
- Clique em *Atualização em massa* abaixo da lista
- Navegue até a guia com os atributos necessários (*Trigger*, *Tags* ou *Dependências*)
- Marque as caixas de seleção de qualquer atributo a ser atualizado

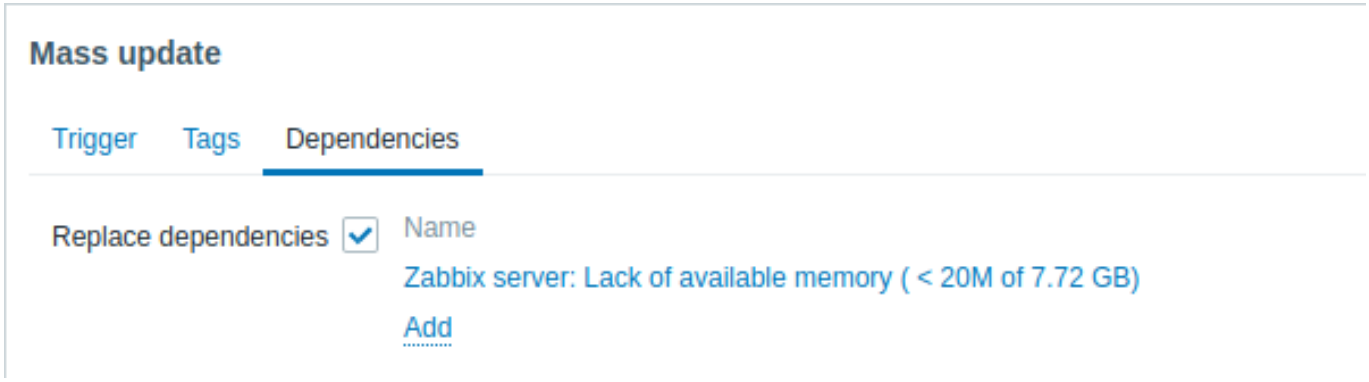




As seguintes opções estão disponíveis ao selecionar o respectivo botão para atualização de tags:

- *Adicionar* - permite adicionar novas tags aos triggers;
- *Substituir* - removerá quaisquer tags existentes do trigger e as substituirá pelas especificadas abaixo;
- *Remover* - removerá as tags especificadas dos triggers.

Observe que tags com o mesmo nome, mas valores diferentes, não são consideradas 'duplicadas' e podem ser adicionadas ao mesmo trigger.



Substituir dependências - removerá quaisquer dependências existentes do trigger e as substituirá pelas especificadas.

Clique em *Atualizar* para aplicar as alterações.

7 Funções preditivas de trigger

Visão geral

Às vezes, há sinais de um problema iminente. Esses sinais podem ser identificados para que ações possam ser tomadas com antecedência para evitar ou pelo menos minimizar o impacto do problema.

O Zabbix possui ferramentas para prever o comportamento futuro do sistema monitorado com base em dados históricos. Essas ferramentas são realizadas por meio de funções preditivas de trigger.

Funções

Antes de definir um trigger, é necessário definir o que é um estado de problema e quanto tempo é necessário para agir. Em seguida, existem duas maneiras de configurar um trigger sinalizando sobre uma situação potencialmente indesejada. Primeiro: o trigger deve disparar quando se espera que o sistema esteja em um estado de problema após o "tempo para agir". Segundo: o trigger deve disparar quando o sistema estiver prestes a atingir o estado de problema em menos de "tempo para agir". As funções de trigger correspondentes a serem usadas são **forecast** e **timeleft**. Observe que a análise estatística subjacente é basicamente idêntica para ambas as funções. Você pode configurar um trigger da maneira que preferir, com resultados semelhantes.

Parâmetros

Ambas as funções usam quase o mesmo conjunto de parâmetros. Use a lista de funções suportadas como referência.

Intervalo de tempo

Primeiramente, você deve especificar o período histórico que o Zabbix deve analisar para gerar a previsão. Você faz isso de uma maneira familiar, por meio do parâmetro `time period` e de um deslocamento de tempo opcional, assim como faz com as funções

avg, **count**, **delta**, **max**, **min** e **sum**.

Horizonte de previsão

(**forecast** apenas)
 O parâmetro `time` especifica até onde no futuro o Zabbix deve extrapolar as dependências encontradas nos dados históricos. Não importa se você usa `time_shift` ou não, `time` é sempre contado a partir do momento atual.

Limite a ser atingido

(**timeleft** apenas)
 O parâmetro `threshold` especifica um valor que o item analisado deve atingir, não importa se de cima para baixo ou de baixo para cima. Uma vez que determinamos f(t) (veja abaixo), devemos resolver a equação f(t) = `threshold` e retornar a raiz que está mais próxima do momento atual e à direita do momento atual ou 1.7976931348623158E+308 se não houver tal raiz.

Note:
Quando os valores do item se aproximam do limite e depois o cruzam, **timeleft** assume que a interseção já está no passado e, portanto, muda para a próxima interseção com o nível `threshold`, se houver. A melhor prática é usar previsões como complemento ao diagnóstico de problemas comum, não como substituição.[1]

Funções de ajuste

O `fit` padrão é a função *linear*. Mas se o seu sistema monitorado for mais complicado, você tem mais opções para escolher.

`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	$x = a*t^b$

Modos

(**forecast** apenas)
 Toda vez que uma função de trigger é avaliada, ela obtém dados do período de histórico especificado e ajusta uma função especificada aos dados. Assim, se os dados forem ligeiramente diferentes, a função ajustada será ligeiramente diferente. Se simplesmente calcularmos o valor da função ajustada em um momento especificado no futuro, você não saberá nada sobre como o item analisado deve se comportar entre agora e aquele momento no futuro. Para algumas opções de `fit` (como *polynomial*), um valor simples do futuro pode ser enganoso.

`mode`	resultado de **forecast**
value	f(agora + `time`)
max	$max_{agora <= t <= agora + time}$ f(t)
min	$min_{agora <= t <= agora + time}$ f(t)
delta	*max - min*
avg	média de f(t) (agora <= t <= agora + `time`) de acordo com a definição

Detalhes

Para evitar cálculos com números muito grandes, consideramos o timestamp do primeiro valor no período especificado mais 1 ns como um novo tempo zero (o tempo de época atual é da ordem de 10^9, a época ao quadrado é 10^{18}, a precisão dupla é cerca de 10^{-16}). 1 ns é adicionado para fornecer todos os valores de tempo positivos para os ajustes *logarítmico* e *potência*, que envolvem o cálculo de log(t). O deslocamento de tempo não afeta os ajustes *linear*, *polinomial*, *exponencial* (além de cálculos mais fáceis e precisos), mas altera a forma das funções *logarítmica* e *potência*.

Possíveis erros

As funções retornam -1 nas seguintes situações:

- o período de avaliação especificado não contém dados;
- o resultado da operação matemática não está definido[3];
- complicações numéricas (infelizmente, para alguns conjuntos de dados de entrada, o intervalo e a precisão do formato de ponto flutuante de dupla precisão tornam-se insuficientes)[4].

Note:
Nenhum aviso ou erro é sinalizado se o ajuste escolhido descrever mal os dados fornecidos ou se houver poucos dados para uma previsão precisa.

Exemplos e tratamento de erros

Para receber um aviso quando você estiver prestes a ficar sem espaço livre em disco no seu host, você pode usar uma expressão de trigger como esta:

```
timeleft(/host/vfs.fs.size[/,free],1h,0)}<1h
```

No entanto, o código de erro -1 pode entrar em ação e colocar seu trigger em estado de problema. Geralmente isso é bom porque você recebe um aviso de que suas previsões não estão funcionando corretamente e você deve analisá-las mais cuidadosamente para descobrir o motivo. Mas às vezes isso é ruim porque -1 pode simplesmente significar que não houve dados sobre o espaço livre em disco do host obtidos na última hora. Se você estiver recebendo muitos alertas falsos positivos, considere usar uma expressão de trigger mais complicada [5]:

```
timeleft(/host/vfs.fs.size[/,free],1h,0)<1h and timeleft(/host/vfs.fs.size[/,free],1h,0)<>-1
```

A situação é um pouco mais difícil com **forecast**. Antes de tudo, -1 pode ou não colocar o trigger em estado de problema dependendo de você ter uma expressão como `forecast(/host/item,(...))<...` ou como `forecast(/host/item,(...))>...`

Além disso, -1 pode ser uma previsão válida se for normal para o valor do item ser negativo. Mas a probabilidade dessa situação no mundo real é insignificante (veja como o operador **=** funciona). Portanto, adicione `... or forecast(/host/item,(...))=-1` ou `... and forecast(/host/item,(...))<>-1` se você quiser ou não quiser tratar -1 como um problema, respectivamente.

Notas de rodapé

[1] Por exemplo, um trigger simples como `timeleft(/host/item,1h,X) < 1h` pode entrar em estado de problema quando o valor do item se aproxima de X e então se recuperar repentinamente assim que o valor X for atingido. Se o problema for o valor do item estar abaixo de X, use: `last(/host/item) < X or timeleft(/host/item,1h,X) < 1h` Se o problema for o valor do item estar acima de X, use: `last(/host/item) > X or timeleft(/host/item,1h,X) < 1h`

[2] O grau do polinômio pode ser de 1 a 6, *polynomial1* é equivalente a *linear*. No entanto, use polinômios de grau mais alto com cautela. Se o período de avaliação contiver menos pontos do que o necessário para determinar os coeficientes do polinômio, o grau do polinômio será reduzido (por exemplo, *polynomial5* é solicitado, mas há apenas 4 pontos, portanto *polynomial3* será ajustado).

[3] Por exemplo, ajustar funções *exponential* ou *power* envolve calcular log() dos valores do item. Se os dados contiverem zeros ou números negativos, você receberá um erro, pois log() é definido apenas para valores positivos.

[4] Para ajustes *linear*, *exponential*, *logarithmic* e *power* todos os cálculos necessários podem ser escritos explicitamente. Para *polynomial* apenas *value* pode ser calculado sem etapas adicionais. Calcular *avg* envolve computar a antiderivada do polinômio (analiticamente). Calcular *max*, *min* e *delta* envolve computar a derivada do polinômio (analiticamente) e encontrar suas raízes (numericamente). Resolver f(t) = 0 envolve encontrar as raízes do polinômio (numericamente).

[5] Mas neste caso -1 pode fazer com que seu trigger se recupere do estado de problema. Para estar totalmente protegido, use: `timeleft(/host/vfs.fs.size[/,free],1h,0)<1h and ({TRIGGER.VALUE}=0 and timeleft(/host/vfs.fs.size[/,free], or {TRIGGER.VALUE}=1)`

4 Eventos

Visão geral

Existem vários tipos de eventos gerados no Zabbix:

- eventos de trigger - sempre que um trigger muda seu status (*OK→PROBLEM→OK*)
- eventos de serviço - sempre que um serviço muda seu status (*OK→PROBLEM→OK*)
- eventos de descoberta - quando hosts ou serviços são detectados
- eventos de autorregistro - quando agentes ativos são registrados automaticamente pelo server
- eventos internos - quando um item/regra de descoberta de baixo nível se torna não suportado ou um trigger entra em estado desconhecido

Os eventos são marcados com data e hora e podem ser a base de ações como envio de e-mail de notificação, etc.

Para visualizar detalhes dos eventos no frontend, vá para *Monitoramento → Problemas*. Lá você pode clicar na data e hora do evento para visualizar detalhes de um evento.

Mais informações estão disponíveis em:

- eventos de trigger
- outras fontes de eventos

1 Geração de evento de trigger

Visão geral

A alteração do status do trigger é a fonte mais frequente e importante de eventos. Cada vez que o trigger muda seu estado, um evento é gerado. O evento contém detalhes da alteração do estado do trigger - quando ocorreu e qual é o novo estado.

Dois tipos de eventos são criados por triggers - Problema e OK.

Eventos de problema

Um evento de problema é criado:

- quando uma expressão de trigger é avaliada como VERDADEIRA se o trigger estiver no estado OK;
- cada vez que uma expressão de trigger é avaliada como VERDADEIRA se a geração de múltiplos eventos de problema estiver habilitada para o trigger.

Eventos OK

Um evento OK fecha o(s) evento(s) de problema relacionado(s) e pode ser criado por 3 componentes:

- triggers - com base nas configurações de 'Geração de evento OK' e 'Evento OK fecha'
- correlação de eventos
- gerenciador de tarefas – quando um evento é fechado manualmente

Triggers

Os triggers possuem uma configuração de 'Geração de evento OK' que controla como os eventos OK são gerados:

- *Expressão* - um evento OK é gerado para um trigger em estado de problema quando sua expressão é avaliada como FALSO. Esta é a configuração mais simples, habilitada por padrão.
- *Expressão de recuperação* - um evento OK é gerado para um trigger em estado de problema quando sua expressão é avaliada como FALSO e a expressão de recuperação é avaliada como VERDADEIRO. Isso pode ser usado se os critérios de recuperação do trigger forem diferentes dos critérios de problema.
- *Nenhum* - um evento OK nunca é gerado. Isso pode ser usado em conjunto com a geração de múltiplos eventos de problema para simplesmente enviar uma notificação quando algo acontecer.

Além disso, os triggers possuem uma configuração 'Evento OK fecha' que controla quais eventos de problema são fechados:

- *Todos os problemas* - um evento OK fechará todos os problemas abertos criados pelo trigger
- *Todos os problemas se os valores das tags coincidirem* - um evento OK fechará os problemas abertos criados pelo trigger e que tenham pelo menos um valor de tag correspondente. A tag é definida pela configuração 'Tag para correspondência' do trigger. Se não houver eventos de problema para fechar, o evento OK não será gerado. Isso é frequentemente chamado de correlação de eventos em nível de trigger.

Correlação de eventos

A correlação de eventos (também chamada de correlação de eventos global) é uma forma de configurar regras personalizadas de fechamento de eventos (resultando na geração de eventos OK).

As regras definem como os novos eventos de problema são pareados com eventos de problema existentes e permitem fechar o novo evento ou os eventos correspondentes, gerando os respectivos eventos OK.

No entanto, a correlação de eventos deve ser configurada com muito cuidado, pois pode afetar negativamente o desempenho do processamento de eventos ou, se mal configurada, fechar mais eventos do que o pretendido (no pior dos casos, até mesmo todos os eventos de problema podem ser fechados). Algumas dicas de configuração:

1. sempre reduza o escopo da correlação definindo uma tag exclusiva para o evento de controle (o evento que é pareado com eventos antigos) e use a condição de correlação 'nova tag de evento'
2. não se esqueça de adicionar uma condição baseada no evento antigo ao usar a operação 'fechar evento antigo', ou todos os problemas existentes poderão ser fechados
3. evite usar nomes de tags comuns usados por diferentes configurações de correlação

Gerenciador de tarefas

Se a configuração 'Permitir fechamento manual' estiver ativada para o trigger, então é possível fechar manualmente eventos de problema gerados pelo trigger. Isso é feito no frontend ao atualizar um problema. O evento não é fechado diretamente – em vez disso, uma tarefa de 'fechar evento' é criada, que é tratada pelo gerenciador de tarefas em breve. O gerenciador de tarefas irá gerar um evento OK correspondente e o evento de problema será fechado.

2 Outras fontes de eventos

Eventos de serviço

Eventos de serviço são gerados apenas se as ações de serviço para esses eventos estiverem habilitadas. Nesse caso, cada alteração de status do serviço cria um novo evento:

- Evento de problema - quando o status do serviço é alterado de OK para PROBLEMA
- Evento de OK - quando o status do serviço é alterado de PROBLEMA para OK

O evento contém detalhes da alteração do estado do serviço - quando ocorreu e qual é o novo estado.

Eventos de descoberta

O Zabbix verifica periodicamente os intervalos de IP definidos nas regras de descoberta de rede. A frequência da verificação é configurável para cada regra individualmente. Uma vez que um host ou serviço é descoberto, um evento de descoberta (ou vários eventos) é gerado.

O Zabbix gera os seguintes eventos:

Evento	Quando é gerado
Serviço Ativo	Toda vez que o Zabbix detecta um serviço ativo.
Serviço Inativo	Toda vez que o Zabbix não consegue detectar o serviço.
Host Ativo	Se pelo menos um dos serviços estiver ATIVO para o IP.
Host Inativo	Se todos os serviços não estiverem respondendo.
Serviço Descoberto	Se o serviço voltar após uma indisponibilidade ou for descoberto pela primeira vez.
Serviço Perdido	Se o serviço for perdido após estar ativo.
Host Descoberto	Se o host voltar após uma indisponibilidade ou for descoberto pela primeira vez.
Host Perdido	Se o host for perdido após estar ativo.

Eventos de autorregistro de agent ativo

O autorregistro de agent ativo cria eventos no Zabbix.

Se configurado, um evento de autorregistro de agent ativo é criado quando um agent ativo anteriormente desconhecido solicita verificações ou se os metadados do host foram alterados. O server adiciona um novo host autorregistrado, usando o endereço IP e a porta recebidos do agent.

Para mais informações, consulte a página de autoregistro de agent ativo.

Eventos internos

Eventos internos ocorrem quando:

- um item muda de estado de 'normal' para 'não suportado'
- um item muda de estado de 'não suportado' para 'normal'
- uma regra de descoberta de baixo nível muda de estado de 'normal' para 'não suportado'
- uma regra de descoberta de baixo nível muda de estado de 'não suportado' para 'normal'
- um trigger muda de estado de 'normal' para 'desconhecido'
- um trigger muda de estado de 'desconhecido' para 'normal'

O objetivo de introduzir eventos internos é permitir que os usuários sejam notificados quando qualquer evento interno ocorrer, por exemplo, um item se tornar não suportado e parar de coletar dados.

Eventos internos só são criados quando ações internas para esses eventos estão habilitadas. Para interromper a geração de eventos internos (por exemplo, para itens que se tornam não suportados), desabilite todas as ações para eventos internos em Alertas → Ações → Ações internas.

Note:
Se as ações internas estiverem desabilitadas, enquanto um objeto estiver no estado 'não suportado', o evento de recuperação para esse objeto ainda será criado.
Se as ações internas estiverem habilitadas, enquanto um objeto estiver no estado 'não suportado', o evento de recuperação para esse objeto será criado, mesmo que o 'evento de problema' não tenha sido criado para o objeto.

Veja também: Recebendo notificação sobre itens não suportados

3 Encerramento manual de problemas

Visão geral

Embora geralmente os eventos de problema sejam resolvidos automaticamente quando o status do trigger vai de 'Problema' para 'OK', pode haver casos em que é difícil determinar se um problema foi resolvido por meio de uma expressão de trigger. Nesses casos, o problema precisa ser resolvido manualmente.

Por exemplo, o *syslog* pode relatar que alguns parâmetros do kernel precisam ser ajustados para um desempenho ideal. Nesse caso, o problema é relatado aos administradores do Linux, eles o corrigem e, em seguida, fecham o problema manualmente.

Os problemas podem ser fechados manualmente apenas para triggers com a opção *Permitir fechamento manual* ativada.

Quando um problema é "fechado manualmente", o Zabbix gera uma nova tarefa interna para o Zabbix server. Em seguida, o processo *task manager* executa essa tarefa e gera um evento OK, fechando assim o evento de problema.

Um problema fechado manualmente não significa que o trigger subjacente nunca entrará novamente em um estado de 'Problema'. A expressão do trigger é reavaliada e pode resultar em um problema:

- Quando novos dados chegam para qualquer item incluído na expressão do trigger (observe que os valores descartados por uma etapa de pré-processamento de limitação não são considerados como recebidos e não farão com que a expressão do trigger seja reavaliada);
- Quando as funções date and time e/ou nodata() são usadas na expressão.

Configuração

São necessárias duas etapas para fechar um problema manualmente.

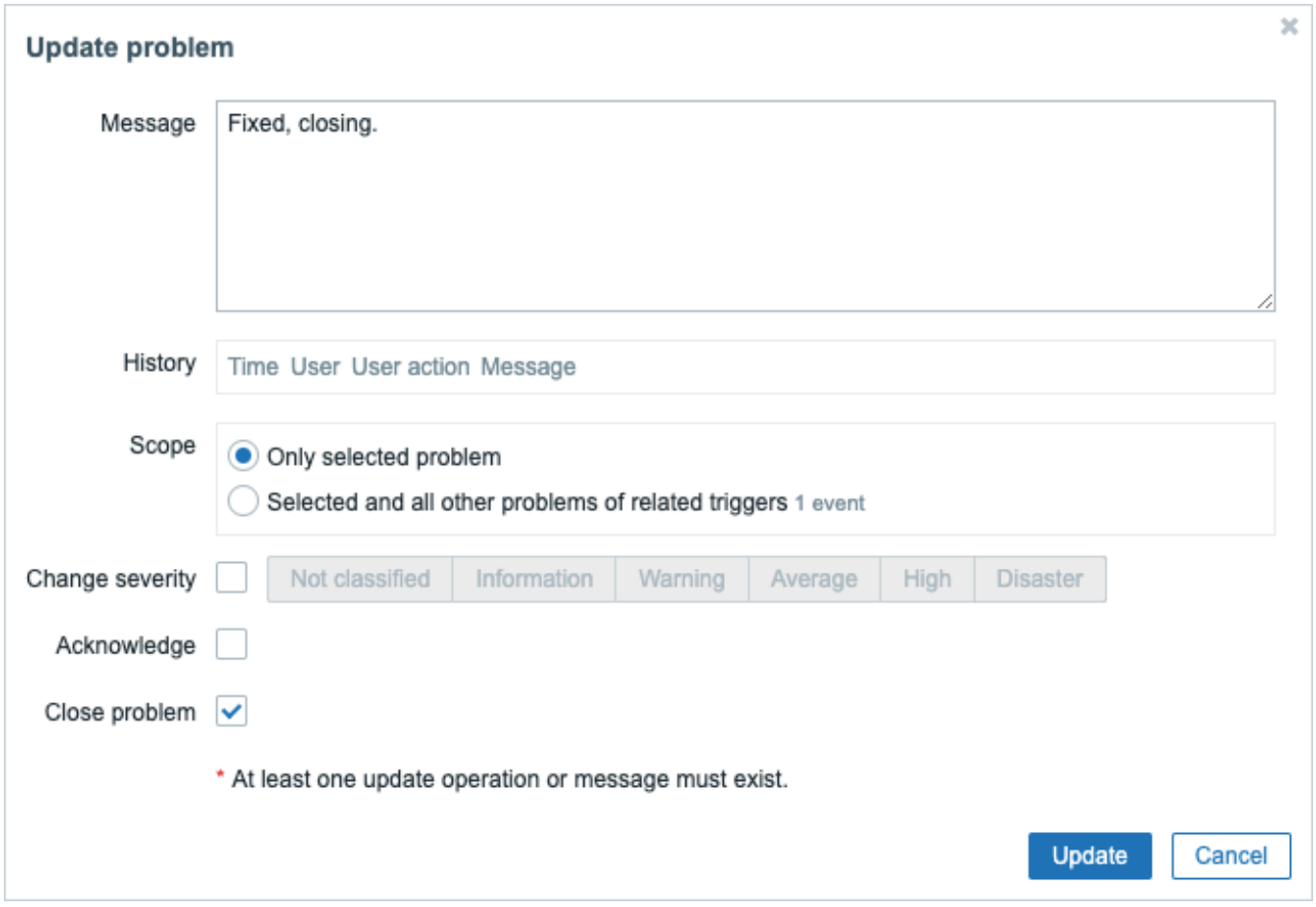
Configuração de trigger

Na configuração do trigger, habilite a opção *Permitir fechamento manual*.

Janela de atualização de problema

Se um problema surgir para um trigger com a flag *Fechamento manual*, você pode abrir a janela pop-up de atualização de problema desse problema e fechá-lo manualmente.

Para fechar o problema, marque a opção *Fechar problema* no formulário e clique em *Atualizar*.



Todos os campos obrigatórios são marcados com um asterisco vermelho.

A solicitação é processada pelo Zabbix server. Normalmente, levará alguns segundos para fechar o problema. Durante esse processo, *FECHANDO* é exibido em *Monitoramento* → *Problemas* como o status do problema.

Verificação

Pode-se verificar que um problema foi fechado manualmente:

- nos detalhes do evento, disponíveis em *Monitoramento* → *Problemas*;
- usando a macro {EVENT.UPDATE.HISTORY} em mensagens de notificação que fornecerá essa informação.

5 Correlação de eventos

Visão geral

A correlação de eventos permite correlacionar eventos de problema à sua resolução de uma forma muito precisa e flexível.

A correlação de eventos pode ser definida:

- no nível do trigger - um trigger pode ser usado para relacionar problemas separados à sua solução
- globalmente - problemas podem ser correlacionados à sua solução a partir de um trigger/método de coleta diferente usando regras de correlação global

1 Correlação de eventos baseada em trigger

Visão geral

A correlação de eventos baseada em trigger permite correlacionar problemas separados relatados por uma trigger.

Embora geralmente um evento OK possa fechar todos os eventos de problema criados por uma trigger, há casos em que uma abordagem mais detalhada é necessária. Por exemplo, ao monitorar arquivos de log, você pode querer descobrir certos problemas em um arquivo de log e fechá-los individualmente, em vez de todos juntos.

Este é o caso de triggers que têm o parâmetro *Modo de geração de evento de PROBLEMA* definido como *Múltiplo*. Essas triggers normalmente são usadas para monitoramento de logs, processamento de traps, etc.

É possível no Zabbix relacionar eventos de problema com base em tagging. As tags são usadas para extrair valores e criar identificação para eventos de problema. Aproveitando isso, os problemas também podem ser fechados individualmente com base na correspondência da tag.

Em outras palavras, a mesma trigger pode criar eventos separados identificados pela tag do evento. Portanto, os eventos de problema podem ser identificados um a um e fechados separadamente com base na identificação pela tag do evento.

Como funciona

No monitoramento de logs, você pode encontrar linhas semelhantes a estas:

```
Linha1: Serviço 1 parado
Linha2: Serviço 2 parado
Linha3: Serviço 1 foi reiniciado
Linha4: Serviço 2 foi reiniciado
```

A ideia da correlação de eventos é ser capaz de associar o evento de problema da Linha1 à resolução da Linha3 e o evento de problema da Linha2 à resolução da Linha4, e fechar esses problemas um por um:

```
Linha1: Serviço 1 parado
Linha3: Serviço 1 foi reiniciado #problema da Linha 1 fechado

Linha2: Serviço 2 parado
Linha4: Serviço 2 foi reiniciado #problema da Linha 2 fechado
```

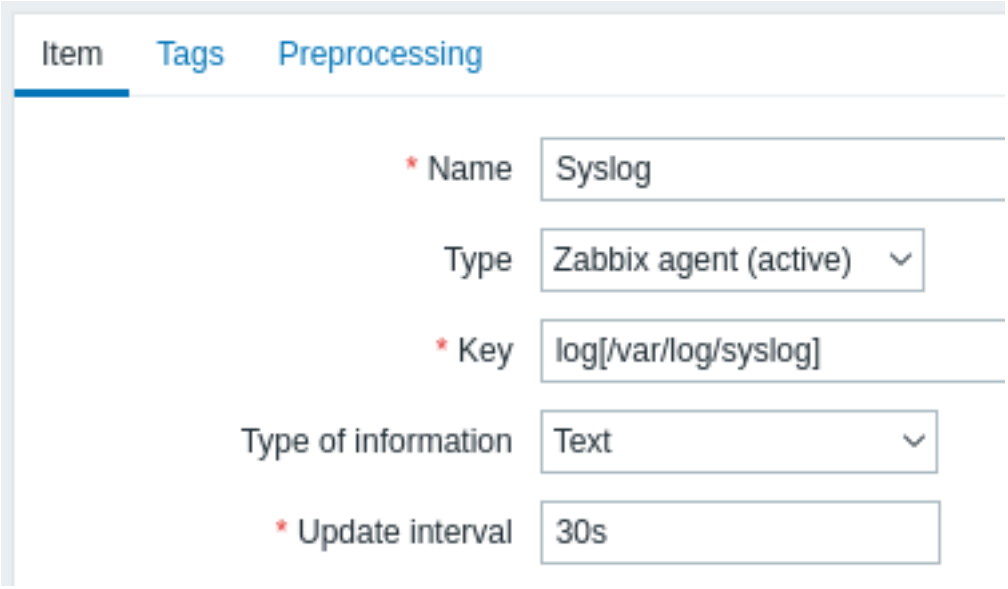
Para fazer isso, você precisa marcar esses eventos relacionados como, por exemplo, ”Serviço 1” e ”Serviço 2”. Isso pode ser feito aplicando uma expressão regular à linha do log para extrair o valor da tag. Então, quando os eventos são criados, eles são marcados como ”Serviço 1” e ”Serviço 2” respectivamente e o problema pode ser associado à resolução.

Configuração

Item

Para começar, você pode querer configurar um item que monitore um arquivo de log, por exemplo:

```
log[/var/log/syslog]
```



Com o item configurado, aguarde um minuto para que as alterações de configuração sejam aplicadas e, em seguida, vá para Últimos dados para garantir que o item começou a coletar dados.

Trigger

Com o item funcionando, você precisa configurar o trigger. É importante decidir quais entradas no arquivo de log merecem atenção. Por exemplo, a seguinte expressão de trigger irá procurar por uma string como 'Stopping' para sinalizar possíveis problemas:

```
find(/My host/log[/var/log/syslog],,"regexp","Stopping")=1
```

Attention:
Para garantir que cada linha contendo a string "Stopping" seja considerada um problema, também defina o *Modo de geração de evento de problema* na configuração do trigger como 'Múltiplos'.

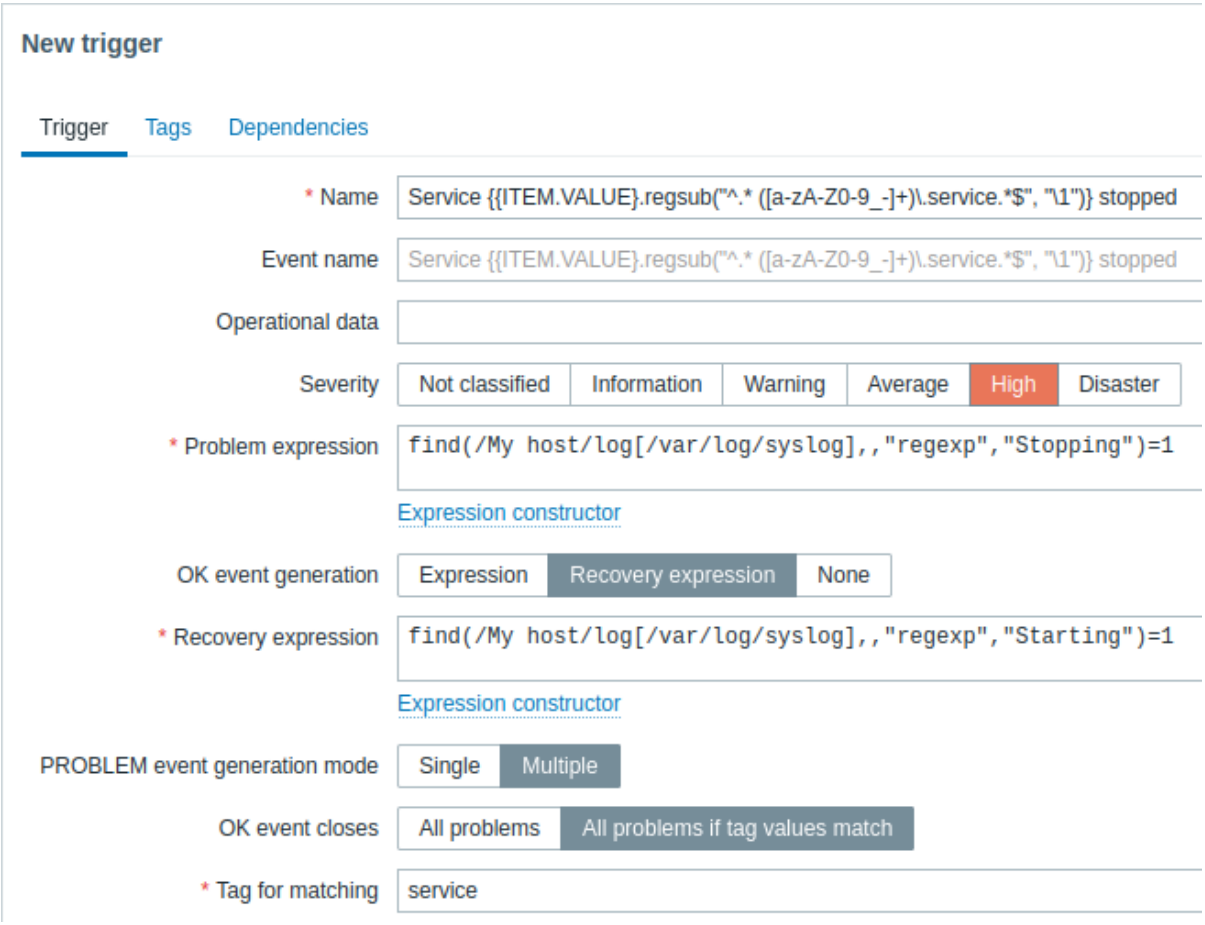
Em seguida, defina uma expressão de recuperação. A seguinte expressão de recuperação resolverá todos os problemas se uma linha de log for encontrada contendo a string "Starting":

```
find(/My host/log[/var/log/syslog],,"regexp","Starting")=1
```

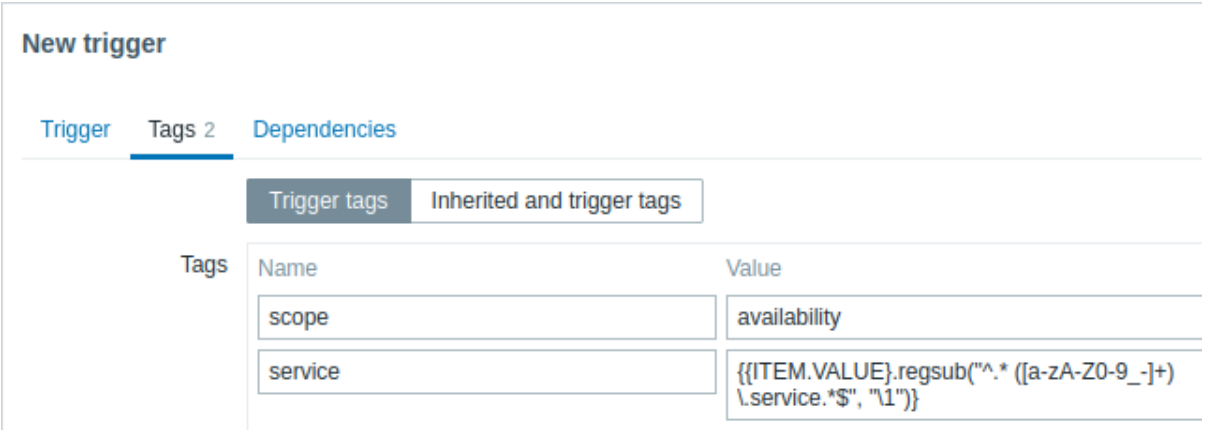
Como não queremos isso, é importante garantir de alguma forma que os problemas raiz correspondentes sejam encerrados, não apenas todos os problemas. É aí que a marcação pode ajudar.

Problemas e resoluções podem ser correspondidos especificando uma tag na configuração do trigger. As seguintes configurações devem ser feitas:

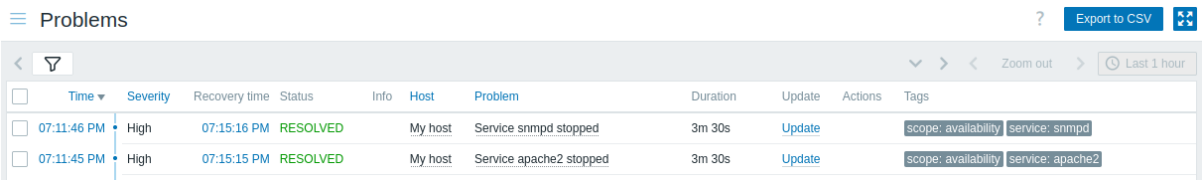
- *Modo de geração de evento de problema*: Múltiplos
- *Evento OK encerra*: Todos os problemas se os valores das tags corresponderem
- Insira o nome da tag para correspondência de eventos



- configure as tags para extrair valores de tags das linhas de log



Se configurado com sucesso, você poderá ver eventos de problema marcados por aplicação e correspondidos à sua resolução em *Monitoramento* → *Problemas*.



Warning:
Como é possível uma má configuração, quando tags de eventos semelhantes podem ser criadas para problemas **não relacionados**, por favor, revise os casos descritos abaixo!

- Macros indexadas sempre se referem ao campo *Expressão* da configuração do trigger, não à *Expressão de recuperação*. Por exemplo, em um evento de recuperação, **{ITEM.VALUE1}** será resolvido para o valor mais recente do primeiro item **na expressão de problema** no momento da recuperação. Se a expressão de recuperação for baseada em um item diferente e o valor do item da expressão de problema mudar até o momento da recuperação, os eventos receberão tags diferentes e

não serão correlacionados.

- Com dois aplicativos gravando mensagens de erro e recuperação no mesmo arquivo de log, um usuário pode decidir usar duas tags *service* no mesmo trigger com valores de tag diferentes, usando expressões regulares separadas nos valores das tags para extrair os nomes, por exemplo, do serviço A e do serviço B da macro {ITEM.VALUE} (por exemplo, quando os formatos das mensagens diferem). No entanto, isso pode não funcionar como planejado se não houver correspondência com as expressões regulares. Expressões regulares sem correspondência resultarão em valores de tag vazios e um único valor de tag vazio em ambos os eventos de problema e OK é suficiente para correlacioná-los. Assim, uma mensagem de recuperação do serviço A pode acidentalmente encerrar uma mensagem de erro do serviço B.
- As tags reais e os valores das tags só se tornam visíveis quando um trigger dispara. Se a expressão regular usada for inválida, ela será silenciosamente substituída por uma string *UNKNOWN*. Se o evento de problema inicial com um valor de tag *UNKNOWN* for perdido, podem aparecer eventos OK subsequentes com o mesmo valor de tag *UNKNOWN* que podem encerrar eventos de problema que não deveriam ter sido encerrados.
- Se um usuário usar a macro {ITEM.VALUE} sem funções de macro como valor da tag, a limitação de 255 caracteres se aplica. Quando as mensagens de log são longas e os primeiros 255 caracteres não são específicos, isso também pode resultar em tags de eventos semelhantes para problemas não relacionados.

2 Correlação global de eventos

Visão geral

A correlação global de eventos permite alcançar todas as métricas monitoradas pelo Zabbix e criar correlações.

É possível correlacionar eventos criados por triggers completamente diferentes e aplicar as mesmas operações a todos eles. Ao criar regras de correlação inteligentes, é possível se livrar de milhares de notificações repetitivas e focar nas causas-raiz de um problema!

A correlação global de eventos é um mecanismo poderoso, que permite desvincular-se da lógica de problema e resolução baseada em um único trigger. Até agora, um único evento de problema era criado por um trigger e dependíamos desse mesmo trigger para a resolução do problema. Não podíamos resolver um problema criado por um trigger com outro trigger. Mas com a correlação de eventos baseada em tags de eventos, podemos.

Por exemplo, um trigger de log pode relatar problemas de aplicação, enquanto um trigger de polling pode relatar que a aplicação está ativa e funcionando. Aproveitando as tags de eventos, você pode marcar o trigger de log como *status:down* enquanto marca o trigger de polling como *status:up*. Então, em uma regra de correlação global, você pode relacionar esses triggers e atribuir uma operação apropriada a essa correlação, como fechar os eventos antigos.

Em outro uso, a correlação global pode identificar triggers semelhantes e aplicar a mesma operação a eles. E se pudéssemos receber apenas um relatório de problema por problema de porta de rede? Não há necessidade de relatar todos eles. Isso também é possível com a correlação global de eventos.

A correlação global de eventos é configurada em **regras de correlação**. Uma regra de correlação define como os novos eventos de problema são pareados com eventos de problema existentes e o que fazer em caso de correspondência (fechar o novo evento, fechar eventos antigos correspondentes gerando os respectivos eventos OK). Se um problema for fechado por correlação global, isso será relatado na coluna *Info* de *Monitoramento* > *Problemas*.

A configuração de regras de correlação global está disponível apenas para usuários do nível Super Admin.

> **Attention:**
> A correlação de eventos deve ser configurada com muito cuidado, pois pode afetar negativamente o desempenho do processamento de eventos ou, se mal configurada, fechar mais eventos do que o pretendido (no pior caso, até mesmo todos os eventos de problema podem ser fechados).

Para configurar a correlação global **com segurança**, observe as seguintes dicas importantes:

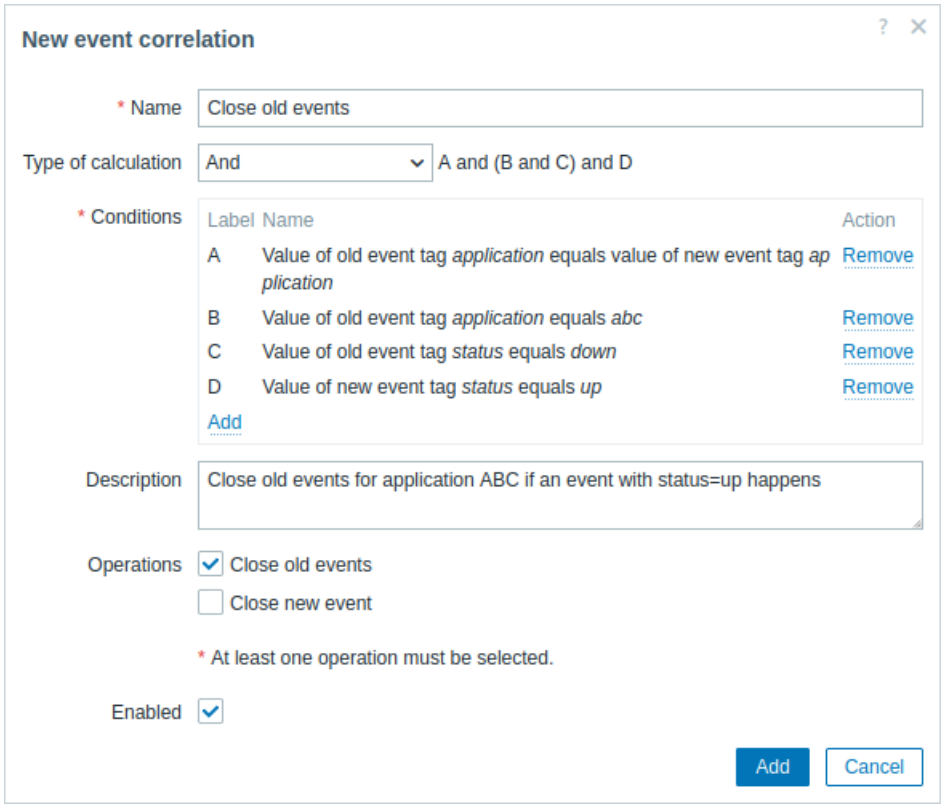
- Reduza o escopo da correlação. Sempre defina uma tag exclusiva para o novo evento que será pareado com eventos antigos e use a condição de correlação *Nome da tag do novo evento*.
- Adicione uma condição explícita de evento antigo ao usar **Fechar eventos antigos**. Sempre adicione pelo menos uma condição de *Evento antigo* (por exemplo, *Nome da tag do evento antigo*, *Valor da tag do evento antigo* ou *Par de tags de evento*) ao selecionar **Fechar eventos antigos** — caso contrário, a regra pode corresponder e fechar eventos de problema existentes não relacionados (no pior caso, todos os problemas). Prefira **Par de tags de evento** para corresponder valores de tempo de execução (host:porta, id de sessão, etc.), e restrinja ainda mais a correspondência por host ou grupo de hosts quando possível.
- Evite usar nomes de tags comuns que possam acabar sendo usados por diferentes configurações de correlação.
- Mantenha o número de regras de correlação limitado às que você realmente precisa.

Veja também: problemas conhecidos.

Configuração

Para configurar regras de correlação de eventos globalmente:

- Vá para *Coleta de dados* > *Correlação de eventos*
- Clique em *Criar correlação de eventos* à direita (ou no nome da correlação para editar uma regra existente)
- Insira os parâmetros da regra de correlação no formulário

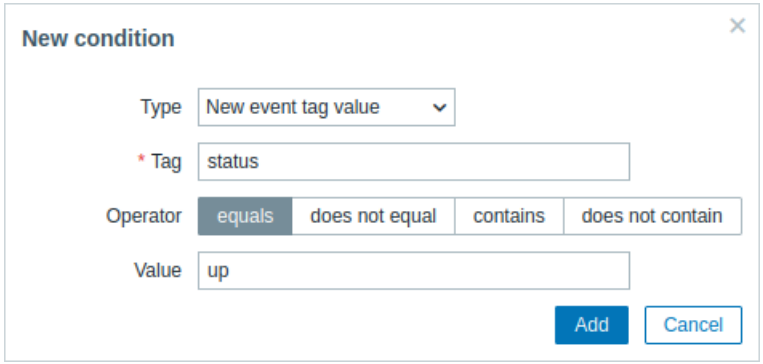


Todos os campos obrigatórios estão marcados com um asterisco vermelho.

Parâmetro	Descrição
Nome	Nome único da regra de correlação.
Tipo de cálculo	As seguintes opções de cálculo de condições estão disponíveis: **E** - todas as condições devem ser atendidas **Ou** - basta que uma condição seja atendida **E/Ou** - E com tipos de condição diferentes e Ou com o mesmo tipo de condição **Expressão personalizada** - uma fórmula de cálculo definida pelo usuário para avaliar as condições da ação. Deve incluir todas as condições (representadas por letras maiúsculas A, B, C, ...) e pode incluir espaços, tabulações, parênteses ( ), **and** (diferencia maiúsculas de minúsculas), **or** (diferencia maiúsculas de minúsculas), **not** (diferencia maiúsculas de minúsculas).
Condições	Lista de condições. Veja abaixo para detalhes sobre como configurar uma condição.
Descrição	Descrição da regra de correlação.

Parâmetro	Descrição
Operações	Marque a caixa de seleção da operação a ser executada quando o evento for correlacionado. As seguintes operações estão disponíveis: **Fechar eventos antigos** - fecha eventos antigos quando um novo evento ocorre. Sempre adicione uma condição baseada no evento antigo ao usar a operação *Fechar eventos antigos* ou todos os problemas existentes poderão ser fechados. **Fechar novo evento** - fecha o novo evento quando ele ocorre. **Atenção!** Não deixe as condições de evento antigo/novo vazias ao usar *Fechar eventos antigos*/*Fechar novo evento*. Se você selecionar a operação *Fechar eventos antigos* sem adicionar uma condição que corresponda ao evento antigo, o Zabbix pode corresponder a todos os eventos antigos existentes e fechá-los. Sempre adicione uma condição explícita de evento antigo (por exemplo, *Nome da tag do evento antigo* ou *Par de tags de evento*) ao usar *Fechar eventos antigos*. Por exemplo, uma regra que usa apenas uma *Condição de novo evento* e a *operação Fechar eventos antigos* corresponderá a todos os eventos antigos que atendam aos critérios de evento antigo (ausentes) - fechando efetivamente problemas antigos.
Habilitado	Se você marcar esta caixa de seleção, a regra de correlação será habilitada.

Para configurar os detalhes de uma nova condição, clique em Add no bloco Condições. Uma janela pop-up será aberta onde você pode editar os detalhes da condição.



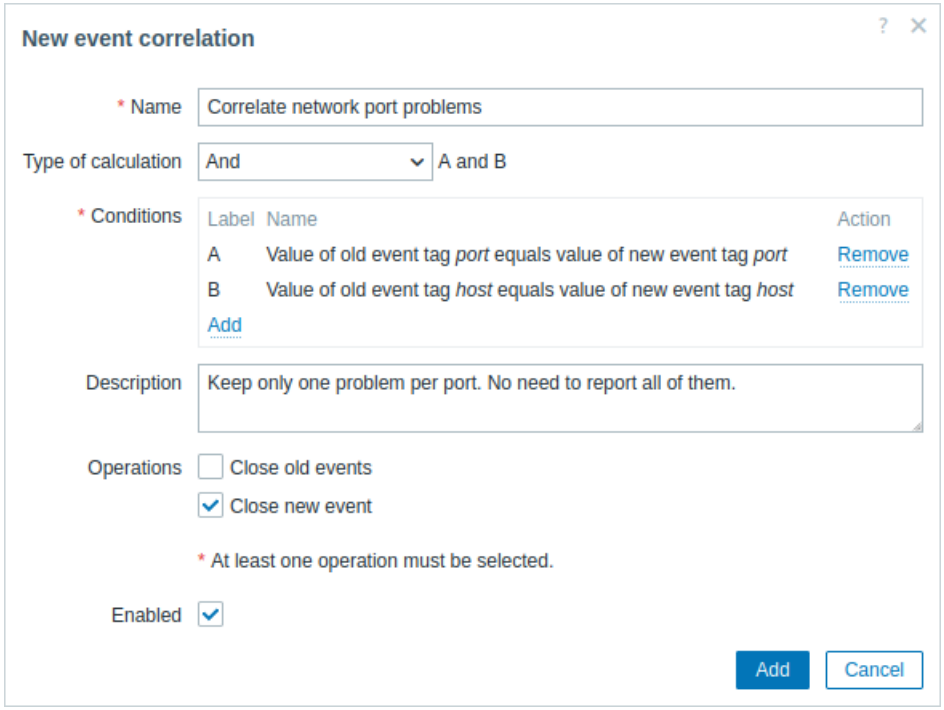
Parâmetro	Descrição
Nova condição	Selecione uma condição para correlacionar eventos. *Nota* que se nenhuma condição de evento antigo for especificada, todos os eventos antigos poderão ser correspondidos e fechados. Da mesma forma, se nenhuma condição de novo evento for especificada, todos os novos eventos poderão ser correspondidos e fechados. As seguintes condições estão disponíveis: **Nome da tag do evento antigo** - especifique o nome da tag do evento antigo para correspondência. **Nome da tag do novo evento** - especifique o nome da tag do novo evento para correspondência. **Grupo de hosts do novo evento** - especifique o grupo de hosts do novo evento para correspondência. **Par de tags de evento** - especifique o nome da tag do novo evento e o nome da tag do evento antigo para correspondência. Neste caso, haverá uma correspondência se os **valores** das tags em ambos os eventos coincidirem. Os *nomes* das tags não precisam coincidir. Esta opção é útil para corresponder valores em tempo de execução, que podem não ser conhecidos no momento da configuração (veja também Exemplo). **Valor da tag do evento antigo** - especifique o nome e o valor da tag do evento antigo para correspondência, usando os seguintes operadores: *igual* - possui o valor da tag do evento antigo *diferente de* - não possui o valor da tag do evento antigo *contém* - possui a string no valor da tag do evento antigo *não contém* - não possui a string no valor da tag do evento antigo **Valor da tag do novo evento** - especifique o nome e o valor da tag do novo evento para correspondência, usando os seguintes operadores: *igual* - possui o valor da tag do novo evento *diferente de* - não possui o valor da tag do novo evento *contém* - possui a string no valor da tag do novo evento *não contém* - não possui a string no valor da tag do novo evento

Warning:
Como é possível uma configuração incorreta, quando tags de evento semelhantes podem ser criadas para problemas **não relacionados**, revise os casos descritos abaixo!

- As tags reais e os valores das tags só se tornam visíveis quando um trigger dispara. Se a expressão regular usada for inválida, ela será silenciosamente substituída por uma string *UNKNOWN*. Se o evento inicial de problema com um valor de tag *UNKNOWN* for perdido, podem aparecer eventos OK subsequentes com o mesmo valor de tag *UNKNOWN* que podem fechar eventos de problema que não deveriam ter sido fechados.
- Se um usuário usar a macro {ITEM.VALUE} sem funções de macro como valor da tag, a limitação de 255 caracteres se aplica. Quando as mensagens de log são longas e os primeiros 255 caracteres não são específicos, isso também pode resultar em tags de evento semelhantes para problemas não relacionados.

Exemplo

Pare eventos de problema repetitivos da mesma porta de rede.



Esta regra de correlação global irá correlacionar problemas se os valores das tags *host* e *port* existirem no trigger e forem os mesmos no evento original e no novo.

A operação irá fechar novos eventos de problema na mesma porta de rede, mantendo apenas o problema original aberto.

6 Marcação

Visão geral

As tags consistem em um nome de tag e um valor de tag. Ao marcar entidades, você pode usar apenas o nome ou combiná-lo com um valor (por exemplo, `mysql`, `jira`, `target:mysql`, `service:jira`, etc.).

As tags podem ser definidas para várias entidades:

- Templates
- Hosts
- Items
- Cenários web
- Triggers
- Serviços
- Items e triggers de template
- Protótipos de host, item e trigger

Note:
Consulte as diretrizes oficiais do Zabbix para recomendações gerais sobre a definição de tags, juntamente com orientações específicas para templates, items, triggers e regras de descoberta de baixo nível.

As tags têm vários propósitos, principalmente marcar eventos. Quando as entidades são marcadas, qualquer novo evento relacionado a uma entidade marcada herdará suas tags. Por exemplo:

- com templates marcados - qualquer problema de host (criado por triggers do template) herdará as tags do template;
- com hosts marcados - qualquer problema de host herdará as tags do host;
- com items/cenários web marcados - qualquer problema de item/cenário web herdará as tags do item/cenário web;
- com triggers marcados - qualquer problema criado pelo trigger herdará as tags do trigger.

Um evento de problema herda todas as tags de toda a cadeia de entidades - templates, hosts, items/cenários web, triggers. Combinações idênticas de `tag:valor` (após a resolução de macros) são mescladas em uma só, evitando assim duplicação.

Tags de evento personalizadas oferecem mais flexibilidade. Por exemplo:

- correlação de eventos pode ser configurada com base em tags de eventos;
- condições de ação podem ser configuradas com base em tags de eventos;

- problemas de item podem ser agrupados com base em tags de eventos;
- tags de problema podem ser usadas para mapear problemas para serviços.

As entidades podem ser marcadas com o mesmo nome de tag, mas com valores de tag diferentes (por exemplo, `component:memory` e `component:storage`). Da mesma forma, uma entidade pode ter uma tag sem valor e a mesma tag com um valor (por exemplo, `database` e `database:postgresql`). Essas tags não são consideradas duplicadas.

Casos de uso

Alguns casos de uso comuns para marcação são os seguintes:

1. Marcar eventos de trigger:
   - Defina uma tag de trigger (por exemplo, `scope:performance`).
   - Os problemas criados por este trigger terão a tag de trigger.
2. Marcar problemas herdados de template:
   - Defina uma tag de template (por exemplo, `target:mysql`).
   - Os problemas criados por triggers deste template terão a tag de template.
3. Marcar problemas de host:
   - Defina uma tag de host (por exemplo, `service:jira`).
   - Os problemas criados por triggers deste host terão a tag de host.
4. Filtrar itens relacionados:
   - Defina uma tag de item (por exemplo, `component:cpu`).
   - Em *Monitoramento* → *Últimos dados*, os itens podem ser filtrados pela tag `component:cpu`.
5. Usar informações extraídas do valor do item como valor da tag:
   - Defina uma tag com uma macro como valor da tag (por exemplo, `tag-name:{{ITEM.VALUE<N>}.regsub()}`).
   - Em *Monitoramento* → *Problemas*, os problemas terão o valor da tag resolvido para os dados extraídos do valor do item.
6. Identificar problemas em um arquivo de log e fechá-los separadamente:
   - Defina uma tag de trigger para o trigger do item de monitoramento de log que irá extrair valores do valor do item usando uma macro (por exemplo, `service:{{ITEM.VALUE<N>}.regsub()}`).
   - Na configuração do trigger, configure a correlação de eventos:
     - defina o *Modo de geração de evento PROBLEM* como "Múltiplos";
     - defina *Evento OK fecha* como "Todos os problemas se os valores das tags corresponderem";
     - defina a tag para correspondência.
   - Os problemas criados pelo trigger do item de log terão a tag de trigger e serão fechados individualmente.
7. Filtrar notificações:
   - Defina tags de trigger (por exemplo, `scope:security` para trigger1 e `scope:availability` para trigger2).
   - Use o filtro de tags nas condições de ação para receber notificações apenas nos eventos que correspondam aos dados da tag.
8. Identificar problemas nas notificações:
   - Defina tags de trigger.
   - Use a macro {EVENT.TAGS} na notificação de problema.
   - A notificação de problema conterá as tags de trigger, facilitando a identificação de qual aplicação/serviço a notificação pertence.
9. Simplificar tarefas de configuração usando tags de template:
   - Defina uma tag de trigger de template.
   - Os triggers criados a partir deste trigger de template terão sua tag.
10. Criar triggers com tags a partir de descoberta de baixo nível (LLD):
    - Defina uma tag de protótipo de trigger com uma macro LLD no nome ou valor da tag (por exemplo, `scope:{#FSNAME}`).
    - Os triggers criados a partir do protótipo de trigger terão sua tag.
11. Corresponder serviços usando tags de serviço:
    - Defina tags de serviço.
    - Configure ações de serviço para serviços com tags correspondentes.
    - Além disso, use tags de serviço para vincular um serviço a um SLA para cálculos de SLA.
12. Vincular serviços a problemas usando tags de problema de serviço:
    - Defina uma tag de problema na configuração do serviço (por exemplo, `target:mysql`).
    - Os problemas com uma tag correspondente serão automaticamente correlacionados ao serviço, e o status do serviço mudará com base nas regras de cálculo de status do serviço configuradas.
13. Suprimir problemas quando um host estiver em modo de manutenção:
    - Defina tags na configuração do período de manutenção.
    - Os problemas com as tags definidas serão suprimidos.
14. Conceder acesso a grupos de usuários:
    - Defina tags na configuração do grupo de usuários.
    - Os usuários do grupo de usuários poderão visualizar apenas problemas com as tags definidas.

Configuração

As tags podem ser definidas em uma guia dedicada, por exemplo, na configuração do trigger:

Suporte a macros

Macros internas e macros de usuário em tags são resolvidas no momento do evento. Até que o evento ocorra, essas macros serão exibidas não resolvidas no frontend do Zabbix.

Macros de descoberta de baixo nível são resolvidas durante o processo de descoberta.

As seguintes macros podem ser usadas em nomes e valores de tags de trigger:

- {ITEM.VALUE}, {ITEM.LASTVALUE}, {HOST.HOST}, {HOST.NAME}, {HOST.CONN}, {HOST.DNS}, {HOST.IP}, {HOST.PORT} e {HOST.ID} macros internas
- {INVENTORY.*} macros internas (para referenciar valores de inventário de host de um ou vários hosts em uma expressão de trigger)
- Macros de usuário e macros de usuário com contexto (o contexto pode incluir macros de descoberta de baixo nível)
- Macros de descoberta de baixo nível (apenas em tags de protótipo de trigger)

As seguintes macros podem ser usadas em nomes e valores de tags de template, host e item/cenário web:

- {HOST.HOST}, {HOST.NAME}, {HOST.CONN}, {HOST.DNS}, {HOST.IP}, {HOST.PORT} e {HOST.ID} macros internas
- {INVENTORY.*} macros internas
- Macros de usuário
- Macros de descoberta de baixo nível (apenas em tags de host e protótipo de item)

As seguintes macros podem ser usadas em notificações baseadas em trigger:

- {EVENT.TAGS} e {EVENT.RECOVERY.TAGS} macros internas (essas macros serão resolvidas para uma lista separada por vírgulas de tags de evento ou tags de evento de recuperação)
- {EVENT.TAGSJSON} e {EVENT.RECOVERY.TAGSJSON} macros internas (essas macros serão resolvidas para um array JSON contendo objetos de tag de evento ou objetos de tag de evento de recuperação)

Extração de substring em tags de trigger

A extração de substring é suportada para preencher o nome da tag ou o valor da tag, usando uma função de macro. A função aplica uma expressão regular ao valor obtido pela macro suportada. Por exemplo:

```
{{ITEM.VALUE}.regsub(padrão, saída)}
{{ITEM.VALUE}.iregsub(padrão, saída)}

{{#LLDMACRO}.regsub(padrão, saída)}
{{#LLDMACRO}.iregsub(padrão, saída)}
```

Se o nome ou valor da tag exceder 255 caracteres após a resolução da macro, ele será truncado para 255 caracteres.

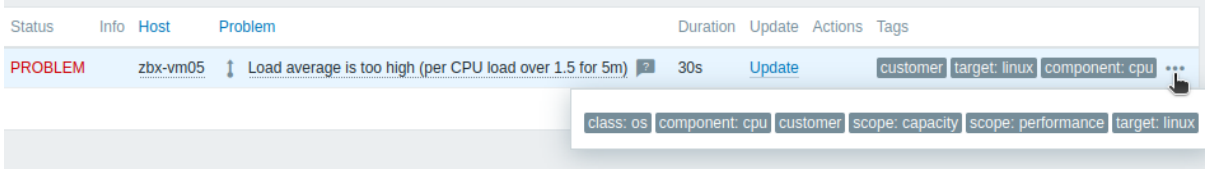
Veja também: Usando funções de macro em macros de descoberta de baixo nível para marcação de eventos.

Visualizando tags de eventos

As tags, se definidas, podem ser visualizadas com novos eventos em:

- *Monitoramento* → *Problemas*
- *Monitoramento* → *Problemas* → *Detalhes do evento*
- *Dashboards* → *Widget de Problemas*

A ordem e o número de tags exibidas são determinados pelas opções de filtragem *Prioridade de exibição da tag* e *Exibir tags* em *Monitoramento* → *Problemas* ou no widget de dashboard *Problemas*. Observe que no máximo três tags podem ser exibidas; se houver mais tags, ao passar o mouse sobre os três pontos, todas as tags serão exibidas em uma janela pop-up.



7 Visualização

Use a barra lateral para acessar o conteúdo da seção Visualização.

1 Gráficos

Visão geral

Com muitos dados fluindo para o Zabbix, torna-se muito mais fácil para os usuários se eles puderem visualizar uma representação gráfica do que está acontecendo, em vez de apenas números.

É aí que entram os gráficos. Os gráficos permitem compreender o fluxo de dados rapidamente, correlacionar problemas, descobrir quando algo começou ou fazer uma apresentação de quando algo pode se tornar um problema.

O Zabbix fornece aos usuários:

- gráficos simples integrados de dados de um item
- a possibilidade de criar gráficos personalizados mais complexos
- acesso à comparação de vários itens rapidamente em gráficos ad-hoc
- gráficos vetoriais modernos e personalizáveis e gráficos de pizza

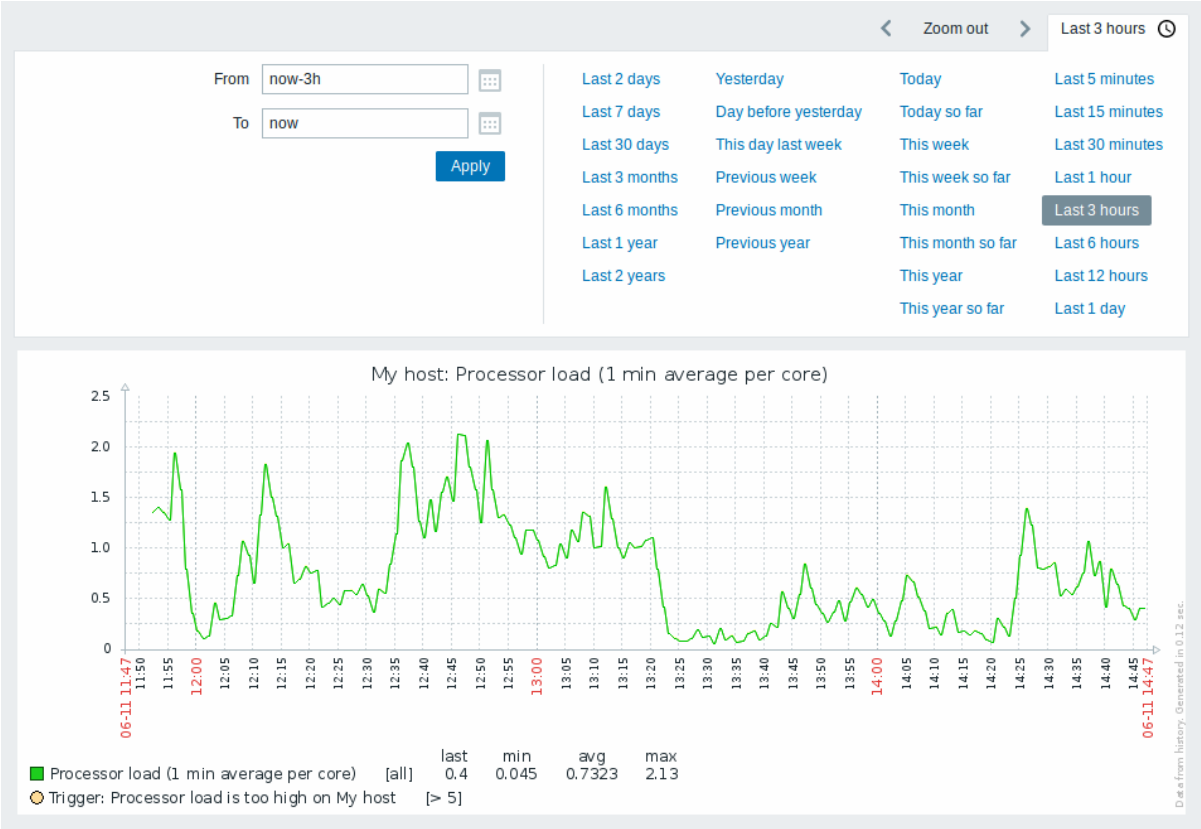
1 Gráficos simples

Visão geral

Gráficos simples são fornecidos para a visualização dos dados coletados pelos items.

Nenhum esforço de configuração é necessário por parte do usuário para visualizar gráficos simples. Eles são disponibilizados livremente pelo Zabbix.

Basta ir em *Monitoramento* > *Últimos dados* e clicar no link Gráfico para o respectivo item e um gráfico será exibido.



Note:
Gráficos simples são fornecidos para todos os items numéricos. Para items textuais, um link para o Histórico está disponível em *Monitoramento* > *Últimos dados*.

Seletor de período de tempo

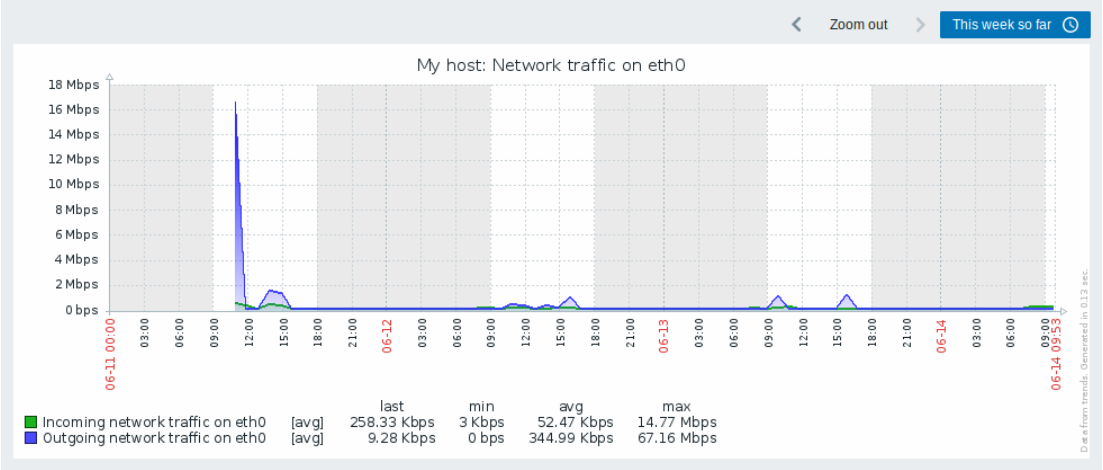
O seletor *Período de tempo* acima do gráfico permite selecionar períodos frequentemente necessários com um clique do mouse. Para mais informações, consulte *Seletor de período de tempo*.

Dados recentes vs períodos mais longos

Para dados muito recentes, uma **única** linha é desenhada conectando cada valor recebido. A linha única é desenhada enquanto houver pelo menos um pixel horizontal disponível para um valor.

Para dados que mostram um período mais longo, **três linhas** são desenhadas - uma verde escura mostra a média, enquanto uma linha rosa clara e uma linha verde clara mostram os valores máximos e mínimos naquele momento. O espaço entre os máximos e mínimos é preenchido com fundo amarelo.

O tempo de trabalho (dias úteis) é exibido nos gráficos como fundo branco, enquanto o tempo não útil é exibido em cinza (com o tema padrão *Original blue* do frontend).

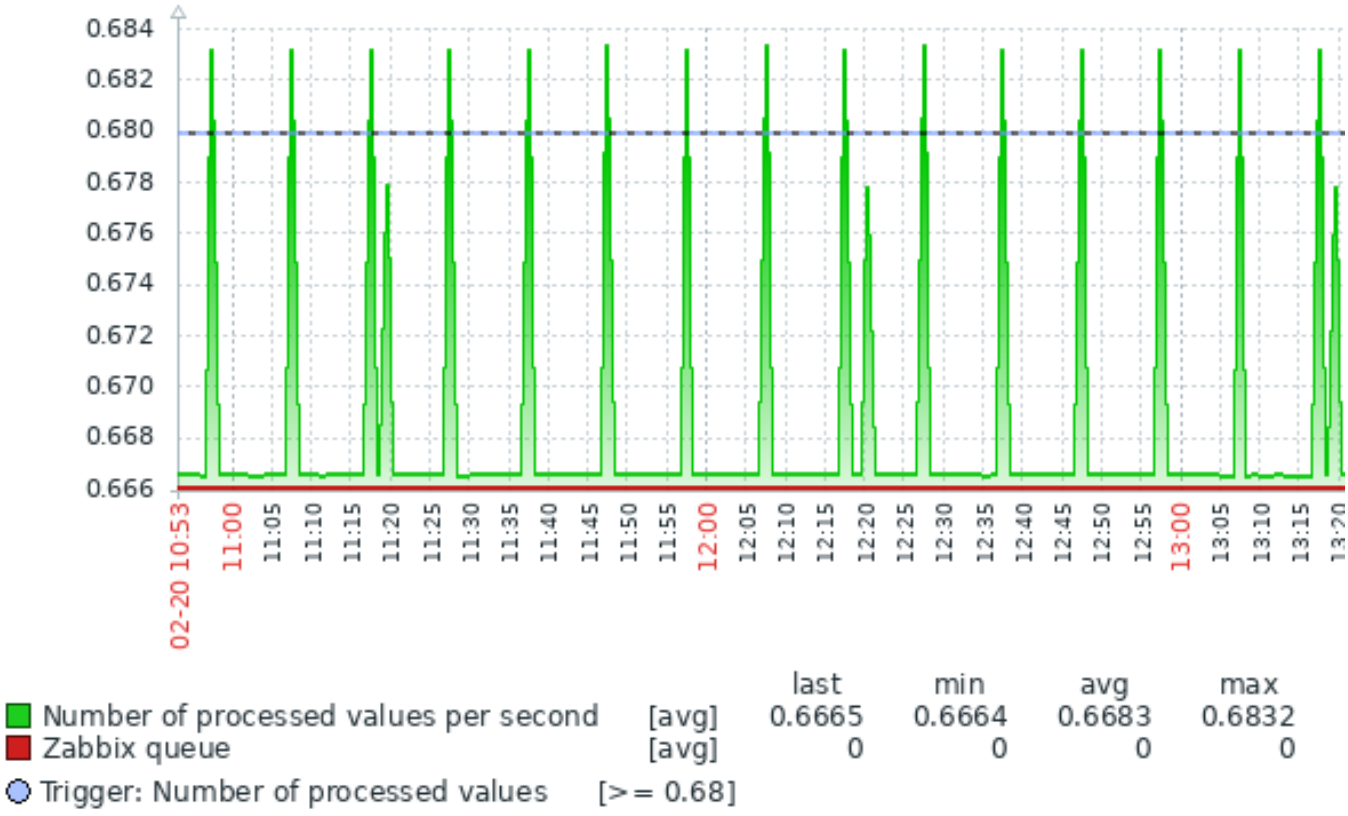


O tempo de trabalho é sempre exibido em gráficos simples, enquanto exibi-lo em gráficos personalizados é uma preferência do usuário.

O tempo de trabalho não é exibido se o gráfico mostrar mais de 3 meses.

Linhas de trigger

Triggers simples são exibidas como linhas com traços pretos sobre a cor de severidade do trigger - observe a linha azul no gráfico e as informações do trigger exibidas na legenda. Até 3 linhas de trigger podem ser exibidas no gráfico; se houver mais triggers, os triggers com menor severidade terão prioridade. Triggers são sempre exibidas em gráficos simples, enquanto a exibição delas em gráficos personalizados é uma preferência do usuário.



Gerando a partir de histórico/tendências

Os gráficos podem ser desenhados com base no histórico ou tendências do item.

Para os usuários que têm o modo de depuração do frontend ativado, uma legenda vertical cinza é exibida no canto inferior direito de um gráfico indicando de onde vêm os dados.

Vários fatores influenciam se o histórico ou as tendências do item são usados:

- longevidade do histórico do item. Por exemplo, o histórico do item pode ser mantido por 14 dias. Nesse caso, quaisquer dados mais antigos do que os quatorze dias virão das tendências.
- congestionamento de dados no gráfico. Se a quantidade de segundos a serem exibidos em um pixel horizontal do gráfico exceder 3600/16, os dados de tendências são exibidos (mesmo que o histórico do item ainda esteja disponível para o mesmo período).
- se as tendências estiverem desativadas, o histórico do item é usado para a construção do gráfico - se disponível para esse período.

Ausência de dados

Para items com um intervalo de atualização regular, nada é exibido no gráfico se os dados do item não forem coletados.

No entanto, para items trapper e items com um intervalo de atualização agendado (e intervalo de atualização regular definido como 0), uma linha reta é desenhada levando até o primeiro valor coletado e do último valor coletado até o final do gráfico; a linha está no nível do primeiro/último valor, respectivamente.

Alternando para valores brutos

Um menu suspenso no canto superior direito permite alternar do gráfico simples para as listagens *Valores/500 valores mais recentes*. Isso pode ser útil para visualizar os valores numéricos que compõem o gráfico.

Os valores representados aqui são brutos, ou seja, nenhuma unidade ou pós-processamento de valores é usado. No entanto, o mapeamento de valores é aplicado.

Problemas conhecidos

Consulte problemas conhecidos para obter gráficos.

2 Gráficos personalizados

Visão geral

Gráficos personalizados, como o nome sugere, oferecem capacidades de personalização.

Enquanto gráficos simples são bons para visualizar dados de um único item, eles não oferecem capacidades de configuração.

Assim, se você quiser alterar o estilo do gráfico ou a forma como as linhas são exibidas ou comparar vários items, por exemplo, tráfego de entrada e saída em um único gráfico, você precisa de um gráfico personalizado.

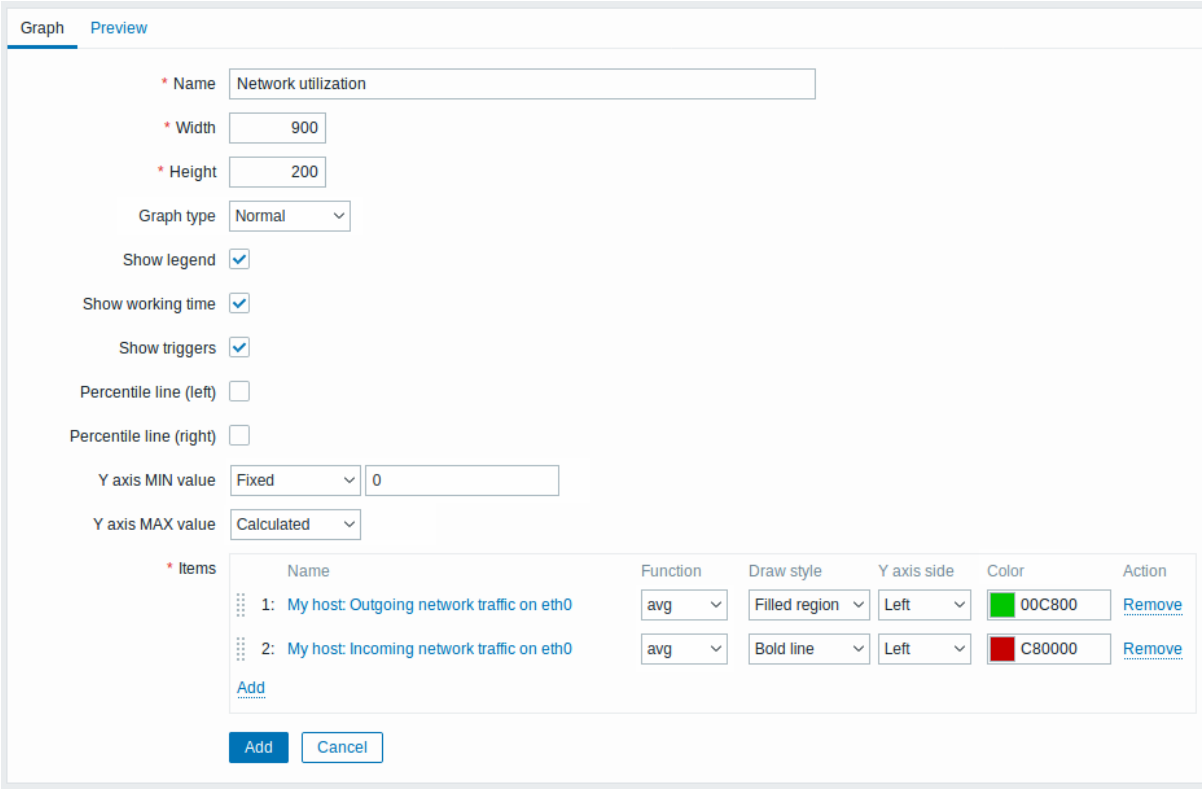
Gráficos personalizados são configurados manualmente.

Eles podem ser criados para um host ou vários hosts ou para um único template.

Configurando gráficos personalizados

Para criar um gráfico personalizado, faça o seguinte:

- Vá para *Coleta de dados > Hosts (ou Templates)*
- Clique em *Gráficos* na linha ao lado do host ou template desejado
- Na tela de Gráficos, clique em *Criar gráfico*
- Edite os atributos do gráfico



Todos os campos obrigatórios estão marcados com um asterisco vermelho.

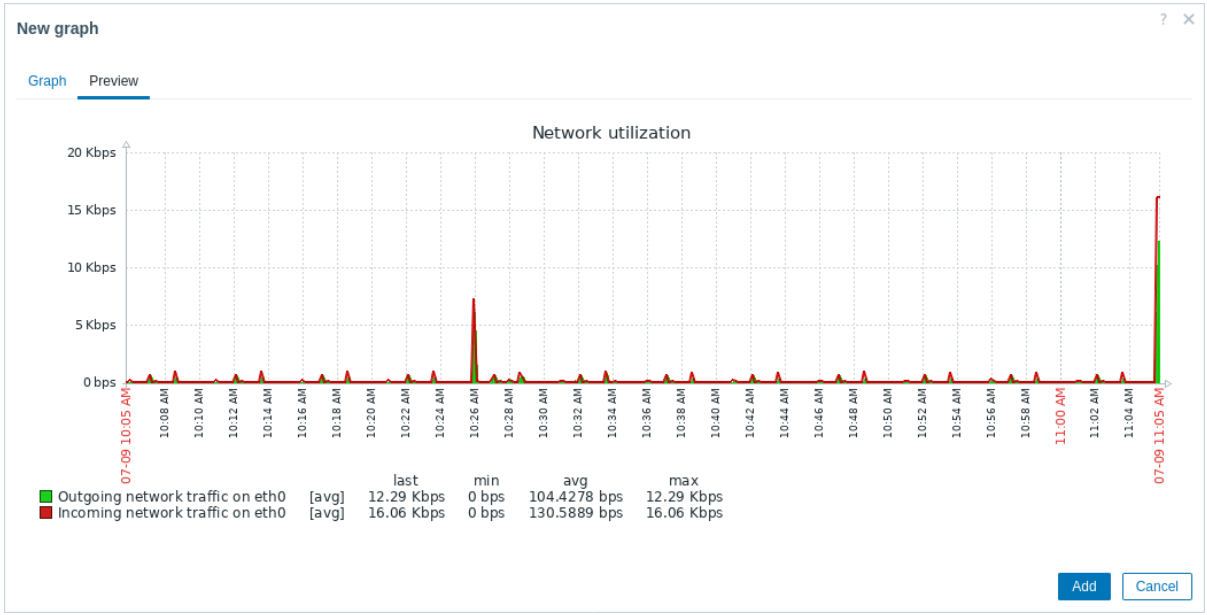
Atributos do gráfico:

Parâmetro	Descrição
Nome	Nome único do gráfico. Expressões macros são suportadas neste campo, mas apenas com as funções `avg`, `last`, `min` e `max`, com o tempo como parâmetro (por exemplo, `{?avg(/host/key,1h)}`). As macros {HOST.HOST<1-9>} são suportadas para uso dentro desta macro, referenciando o primeiro, segundo, terceiro, etc. host no gráfico, por exemplo `{?avg(/{HOST.HOST2}/key,1h)}`. Observe que referenciar o primeiro host com esta macro é redundante, pois o primeiro host pode ser referenciado implicitamente, por exemplo `{?avg(//key,1h)}`.
Largura	Largura do gráfico em pixels (apenas para visualização e gráficos de pizza/explodidos).
Altura	Altura do gráfico em pixels.
Tipo de gráfico	Tipo de gráfico: **Normal** - gráfico normal, valores exibidos como linhas **Empilhado** - gráfico empilhado, áreas preenchidas exibidas **Pizza** - gráfico de pizza **Explodido** - gráfico de pizza ”explodido”, porções exibidas como ”recortadas” da pizza
Mostrar legenda	Marcar esta caixa exibirá a legenda do gráfico.
Mostrar horário de trabalho	Se selecionado, as horas não trabalhadas serão mostradas com um fundo cinza. Este parâmetro não está disponível para gráficos de pizza e pizza explodida.
Mostrar triggers	Se selecionado, triggers simples serão exibidos como linhas com traços pretos sobre a cor de severidade do trigger. Este parâmetro não está disponível para gráficos de pizza e pizza explodida.
Linha percentil (esquerda)	Exibe o percentil para o eixo Y esquerdo. Se, por exemplo, o percentil de 95% for definido, a linha do percentil estará no nível onde 95 por cento dos valores ficam abaixo. Exibido como uma linha verde brilhante. Disponível apenas para gráficos normais.

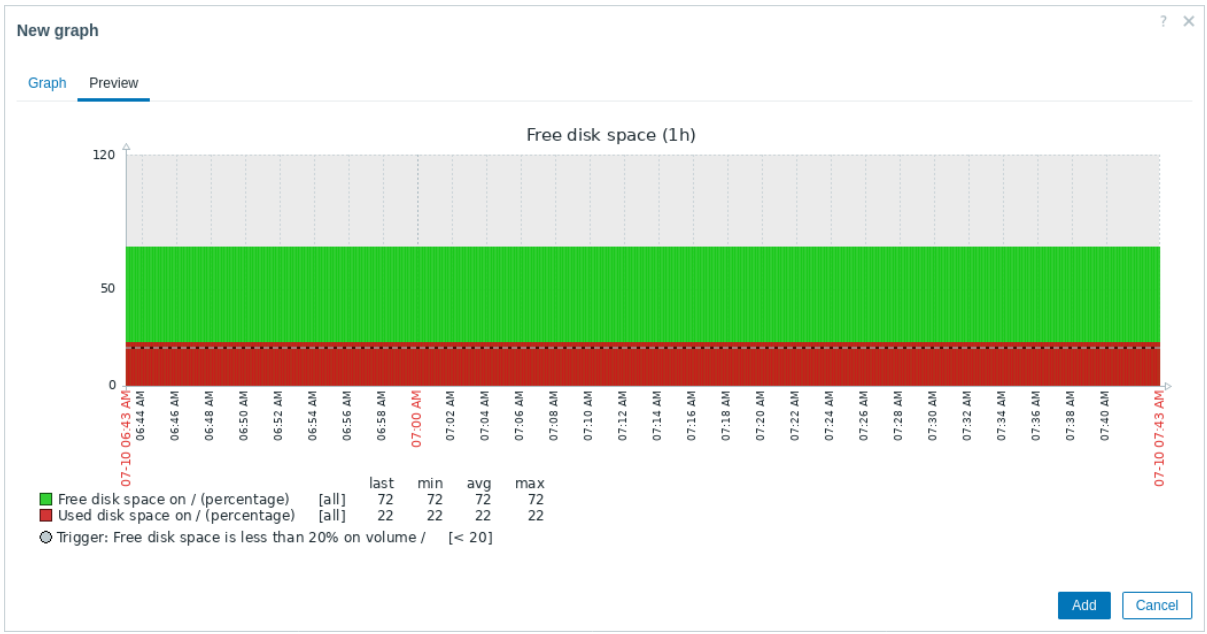
Parâmetro	Descrição
Linha percentil (direita)	Exibe o percentil para o eixo Y direito. Se, por exemplo, o percentil de 95% for definido, a linha do percentil estará no nível onde 95 por cento dos valores ficam abaixo. Exibido como uma linha vermelha brilhante. Disponível apenas para gráficos normais.
Valor MÍN do eixo Y	Valor mínimo do eixo Y: **Calculado** - o valor mínimo do eixo Y será calculado automaticamente. **Fixo** - valor mínimo fixo para o eixo Y. **Item** - o último valor do item selecionado será o valor mínimo. Este parâmetro não está disponível para gráficos de pizza e pizza explodida.
Valor MÁX do eixo Y	Valor máximo do eixo Y: **Calculado** - o valor máximo do eixo Y será calculado automaticamente. **Fixo** - valor máximo fixo para o eixo Y. **Item** - o último valor do item selecionado será o valor máximo Este parâmetro não está disponível para gráficos de pizza e pizza explodida.
Visualização 3D	Habilita o estilo 3D. Apenas para gráficos de pizza e pizza explodida.
Itens	Itens, cujos dados serão exibidos neste gráfico. Clique em *Adicionar* para selecionar os itens. Você também pode selecionar várias opções de exibição (função, estilo de desenho, exibição do eixo esquerdo/direito, cor).
Ordem de classificação (0→100)	Ordem de desenho. 0 será processado primeiro. Pode ser usado para desenhar linhas ou regiões atrás (ou na frente) de outra. Você pode arrastar e soltar os itens usando o ícone no início de uma linha para definir a ordem de classificação ou qual item é exibido na frente do outro.
Nome	O nome do item selecionado é exibido como um link. Clicar no link abre a lista de outros itens disponíveis.
Tipo	Tipo (disponível apenas para gráficos de pizza e pizza explodida): **Simples** - o valor do item é representado proporcionalmente na pizza **Soma do gráfico** - o valor do item representa toda a pizza Observe que a coloração do item ”soma do gráfico” só será visível na medida em que não for ocupada por itens ”proporcionais”.
Função	Selecione quais valores serão exibidos quando mais de um valor existir por pixel vertical do gráfico para um item: **all** - exibe todos os valores possíveis (mínimo, máximo, média) no gráfico. Observe que, para períodos mais curtos, esta configuração não tem efeito; apenas para períodos mais longos, quando a congestão de dados em um pixel vertical do gráfico aumenta, ’all’ começa a exibir valores mínimos, máximos e médios. Esta função está disponível apenas para o tipo de gráfico *Normal*. Veja também: Gerando gráficos a partir de histórico/tendências. **avg** - exibe os valores médios **last** - exibe os valores mais recentes. Esta função está disponível apenas se *Pizza/Pizza explodida* for selecionado como tipo de gráfico. **max** - exibe os valores máximos **min** - exibe os valores mínimos
Estilo de desenho	Selecione o estilo de desenho (disponível apenas para gráficos normais; para gráficos empilhados, a região preenchida é sempre usada) a ser aplicado aos dados do item - *Linha*, *Linha em negrito*, *Região preenchida*, *Ponto*, *Linha tracejada*, *Linha gradiente*.
Lado do eixo Y	Selecione o lado do eixo Y para mostrar os dados do item - *Esquerda*, *Direita*.
Cor	Selecione a cor a ser aplicada aos dados do item.

Pré-visualização do gráfico

Na aba *Pré-visualização*, uma pré-visualização do gráfico é exibida para que você possa ver imediatamente o que está criando.



Observe que a pré-visualização não mostrará nenhum dado para items de template.



Neste exemplo, preste atenção à linha tracejada em negrito exibindo o nível do trigger e as informações do trigger exibidas na legenda.

Note:
Não mais do que 3 linhas de trigger podem ser exibidas. Se houver mais triggers, os triggers com menor severidade terão prioridade para exibição.

 Se a altura do gráfico for definida como menor que 120 pixels, nenhum trigger será exibido na legenda.

3 Gráficos ad hoc

Visão geral

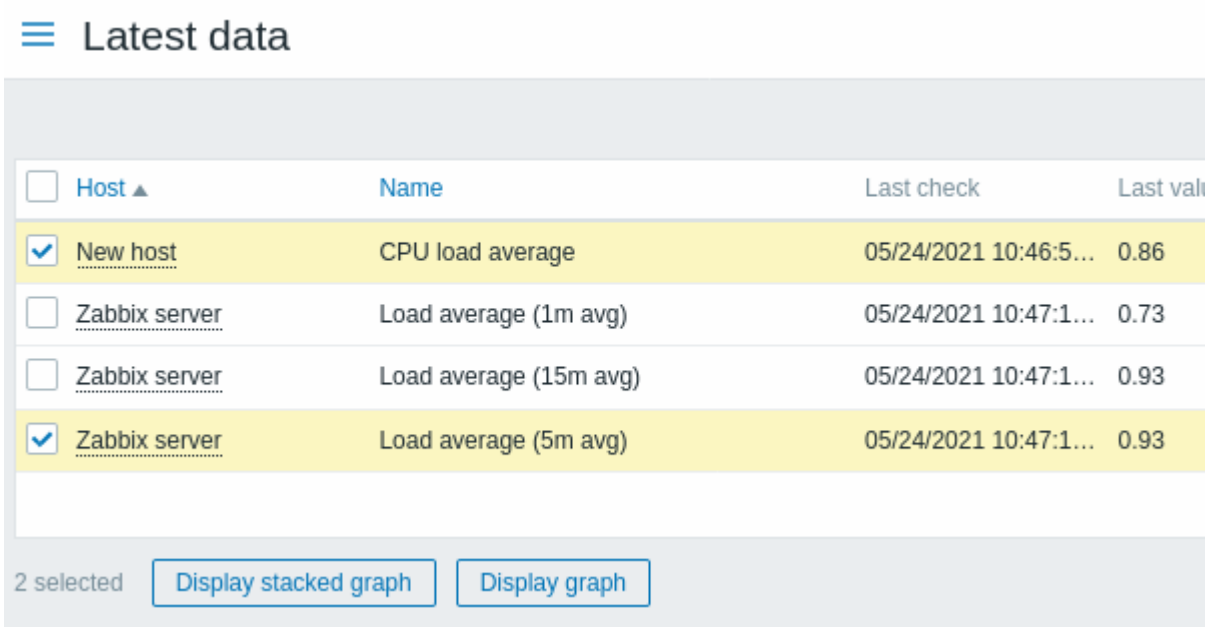
Embora um gráfico simples seja ótimo para acessar os dados de um item e gráficos personalizados ofereçam opções de personalização, nenhum dos dois permite criar rapidamente um gráfico de comparação para vários itens com pouco esforço e sem manutenção.

Para resolver esse problema, é possível criar gráficos ad-hoc para vários itens de forma muito rápida.

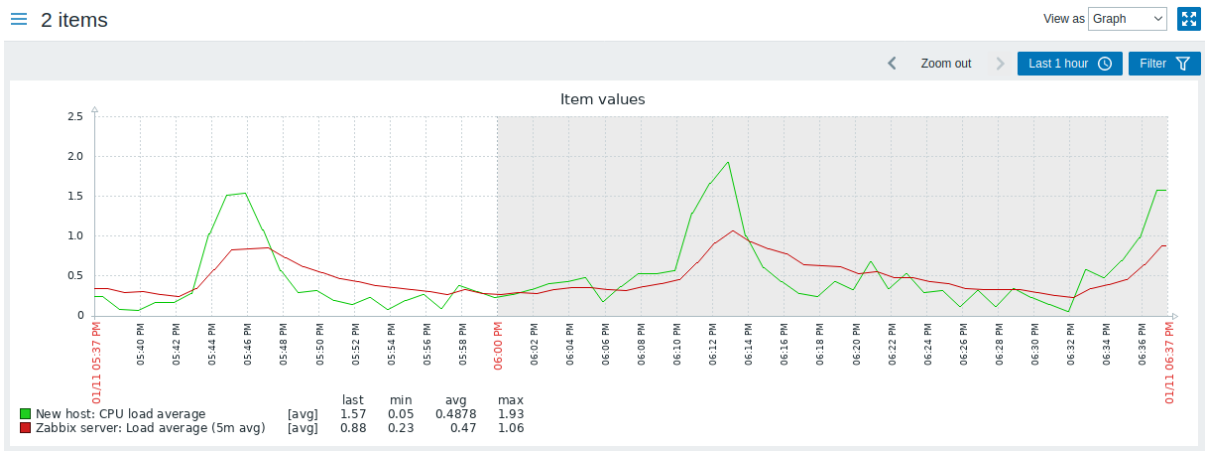
Configuração

Para criar um gráfico ad-hoc, faça o seguinte:

- Vá para *Monitoramento → Últimos dados*
- Use o filtro para exibir os items que deseja
- Marque as caixas dos items que deseja exibir no gráfico
- Clique nos botões *Exibir gráfico empilhado* ou *Exibir gráfico*

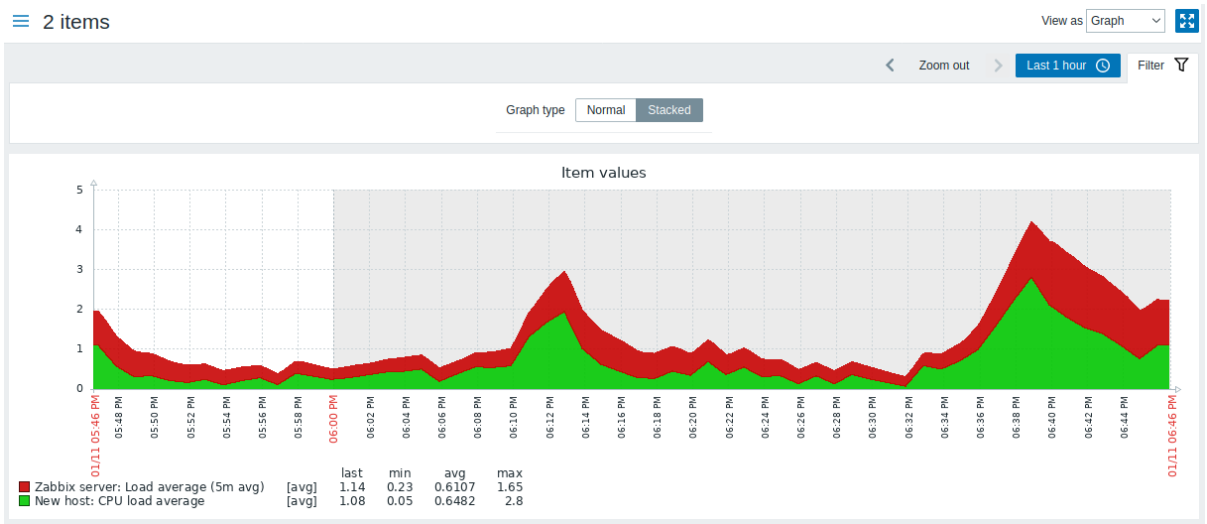


Seu gráfico é criado instantaneamente:



Observe que, para evitar a exibição de muitas linhas no gráfico, apenas o valor médio de cada item é exibido (linhas de valor mínimo/máximo não são exibidas). Triggers e informações de trigger não são exibidas no gráfico.

Na janela do gráfico criado, você tem disponível o seletor de *Período de tempo* e a possibilidade de alternar do gráfico de linhas "normal" para o empilhado (e vice-versa).



4 Agregação em gráficos

Visão geral

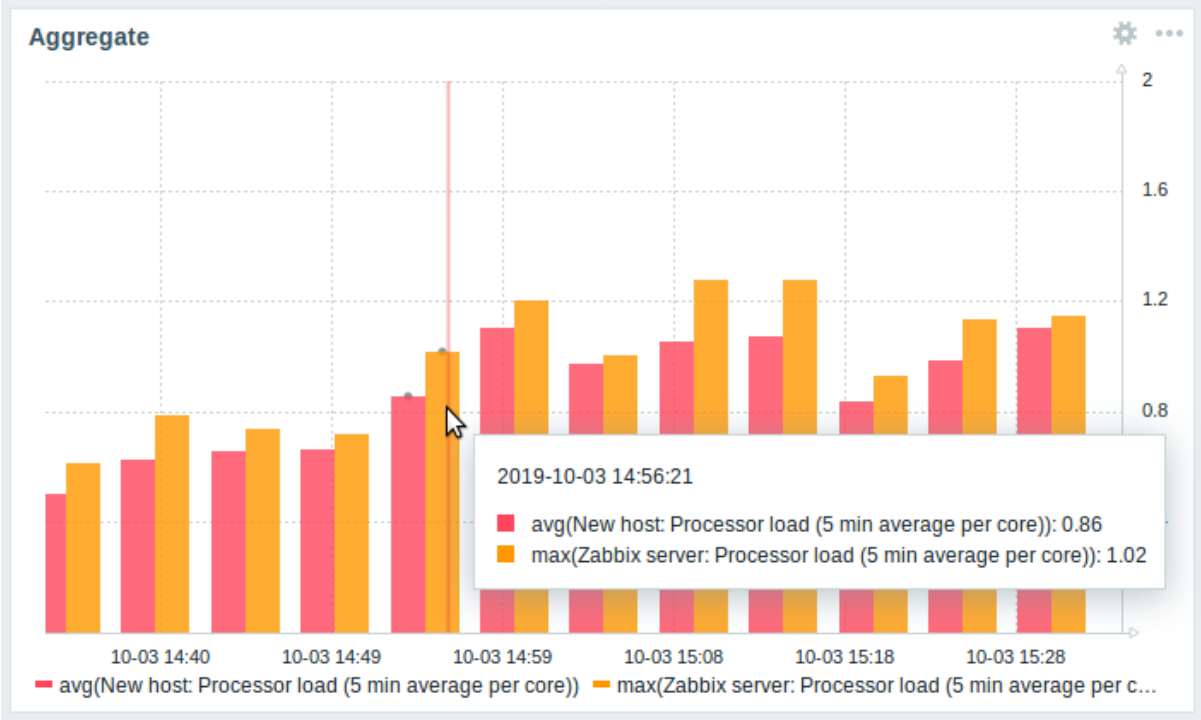
Funções de agregação, disponíveis nos widgets de gráfico e gráfico de pizza do dashboard, permitem exibir um valor agregado para o intervalo escolhido (5 minutos, uma hora, um dia), em vez de todos os valores.

Esta seção fornece mais detalhes sobre as opções de agregação no widget de gráfico.

As opções de agregação são as seguintes:

- min
- max
- avg
- count
- sum
- first (primeiro valor exibido)
- last (último valor exibido)

O uso mais interessante da agregação de dados é a possibilidade de criar belas comparações lado a lado de dados para algum período:



Ao passar o mouse sobre um ponto no tempo no gráfico, a data e a hora são exibidas, além dos items e seus valores agregados. Os items são exibidos entre parênteses, prefixados pela função de agregação utilizada. Se o widget de gráfico tiver um *Rótulo do conjunto de dados* configurado, o rótulo é exibido entre parênteses, prefixado pela função de agregação utilizada. Observe que esta é a data e hora do ponto no gráfico, não dos valores reais.

Configuração

As opções de agregação estão disponíveis nas configurações do conjunto de dados ao configurar um widget de gráfico.

Você pode escolher a função de agregação e o intervalo de tempo. Como o conjunto de dados pode conter vários items, há também outra opção que permite mostrar os dados agregados para cada item separadamente ou para todos os items do conjunto de dados como um único valor agregado.

Casos de uso

Contagem média de solicitações para o servidor Nginx

Veja a contagem média de solicitações por segundo por dia para o servidor Nginx:

- adicione o item de contagem de solicitações por segundo ao conjunto de dados
- selecione a função agregada `avg` e especifique o intervalo `1d`
- é exibido um gráfico de barras, onde cada barra representa a média número de solicitações por segundo por dia

Espaço mínimo semanal em disco entre clusters

Visualize o menor espaço em disco entre os clusters ao longo de uma semana.

- adicione ao conjunto de dados: hosts `cluster*`, chave "Free disk space on /dados"
- selecione a função agregada `min` e especifique o intervalo `1w`
- é exibido um gráfico de barras, onde cada barra representa o disco mínimo espaço por semana para cada volume /data do cluster

2 Mapas de rede

Visão geral

Se você tem uma rede para cuidar, pode querer ter uma visão geral da sua infraestrutura em algum lugar. Para esse fim, você pode criar mapas no Zabbix - de redes e de qualquer coisa que desejar.

Todos os usuários podem criar mapas de rede. Os mapas podem ser públicos (disponíveis para todos os usuários) ou privados (disponíveis para usuários selecionados).

Prossiga para configurar um mapa de rede.

1 Configurando um mapa de rede

Visão geral

Configurar um mapa no Zabbix requer que você primeiro crie um mapa definindo seus parâmetros gerais e, em seguida, comece a preencher o mapa real com elementos e seus links.

Você pode preencher o mapa com elementos que podem ser um host, um grupo de hosts, um trigger, uma imagem ou outro mapa.

Ícones são usados para representar elementos do mapa. Você pode definir as informações que serão exibidas com os ícones e definir que problemas recentes sejam exibidos de uma maneira especial. Você pode vincular os ícones e definir informações a serem exibidas nos links.

Você pode adicionar URLs personalizadas para serem acessadas ao clicar nos ícones. Assim, você pode vincular um ícone de host às propriedades do host ou um ícone de mapa a outro mapa.

A contagem de problemas nos mapas é exibida apenas para problemas de causa.

Os mapas são gerenciados em *Monitoramento* > *Mapas*, onde podem ser configurados, gerenciados e visualizados. Na visualização de monitoramento, você pode clicar nos ícones e aproveitar os links para alguns scripts e URLs.

Os mapas de rede são baseados em gráficos vetoriais (SVG).

Mapas públicos e privados

Todos os usuários no Zabbix (incluindo usuários não administradores) podem criar mapas de rede. Os mapas têm um proprietário - o usuário que os criou. Os mapas podem ser tornados públicos ou privados.

- *Mapas públicos* são visíveis para todos os usuários, embora, para vê-los, o usuário deva ter permissão de leitura para pelo menos um elemento do mapa. Mapas públicos podem ser editados caso um usuário/grupo de usuários tenha permissões de leitura e escrita para este mapa e pelo menos permissões de leitura para todos os elementos do mapa correspondente, incluindo triggers nos links.
- *Mapas privados* são visíveis apenas para seu proprietário e para os usuários/grupos de usuários com os quais o mapa é compartilhado pelo proprietário. Usuários regulares (não Super administradores) só podem compartilhar com os grupos aos quais pertencem e usuários que pertencem a esses grupos. Usuários com nível de administrador podem ver mapas privados independentemente de serem o proprietário ou pertencerem à lista de usuários compartilhados. Mapas privados podem ser editados pelo proprietário do mapa e caso um usuário/grupo de usuários tenha permissões de leitura e escrita para este mapa e pelo menos permissões de leitura para todos os elementos do mapa correspondente, incluindo triggers nos links.

Elementos do mapa para os quais o usuário não tem permissão de leitura são exibidos com um ícone acinzentado e todas as informações textuais sobre o elemento são ocultadas. No entanto, o rótulo do trigger é visível mesmo que o usuário não tenha permissão para o trigger.

Para adicionar um elemento ao mapa, o usuário também deve ter pelo menos permissão de leitura para o elemento.

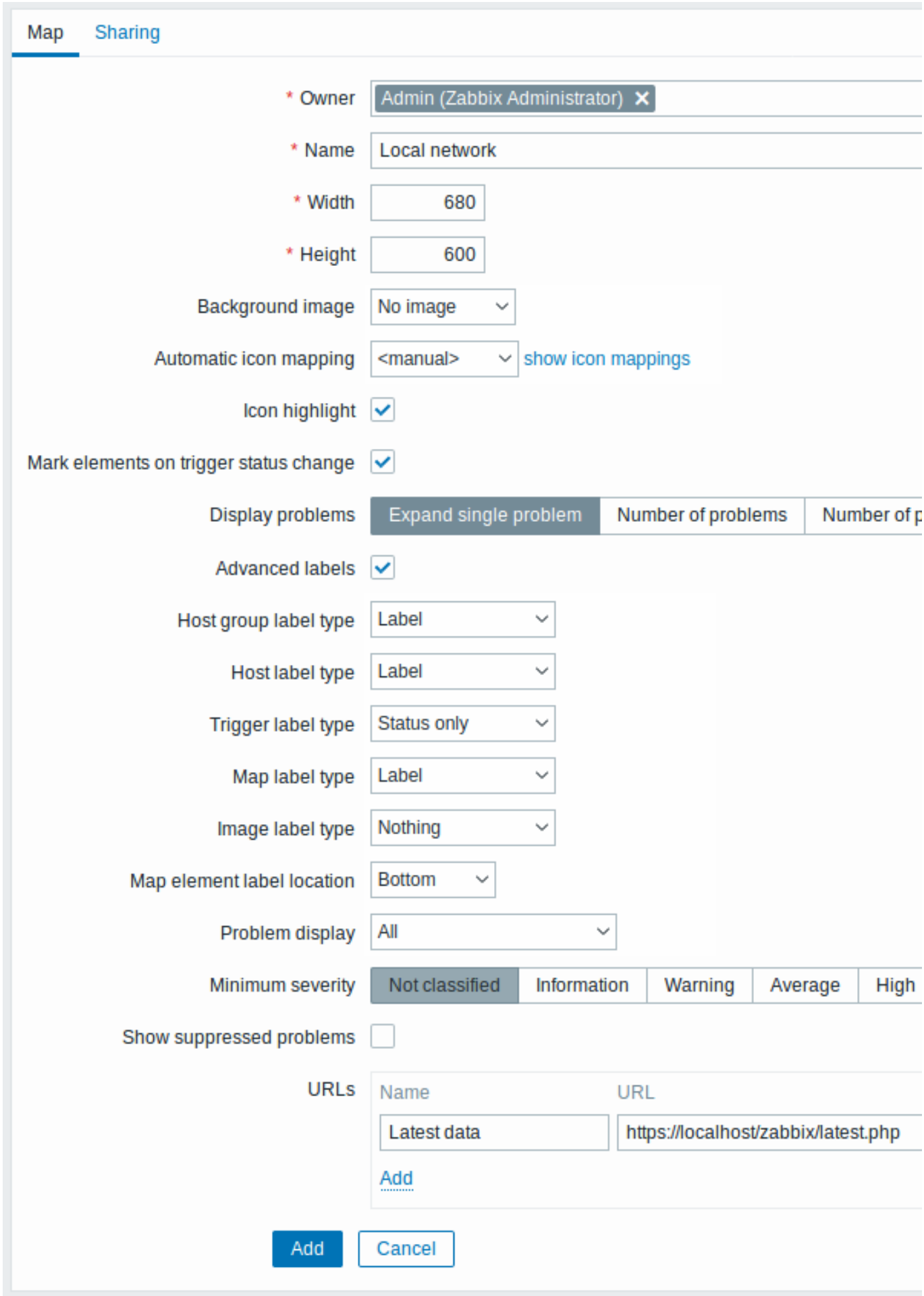
Criando um mapa

Para criar um mapa, faça o seguinte:

- Vá para *Monitoramento* > *Mapas*
- Vá para a visualização com todos os mapas
- Clique em *Criar mapa*

Você também pode usar o botão *Clonar* no formulário de configuração de um mapa existente para criar um novo mapa. Este mapa terá todas as propriedades do mapa existente, incluindo atributos gerais de layout, bem como os elementos do mapa existente.

A aba **Mapa** contém os atributos gerais do mapa:



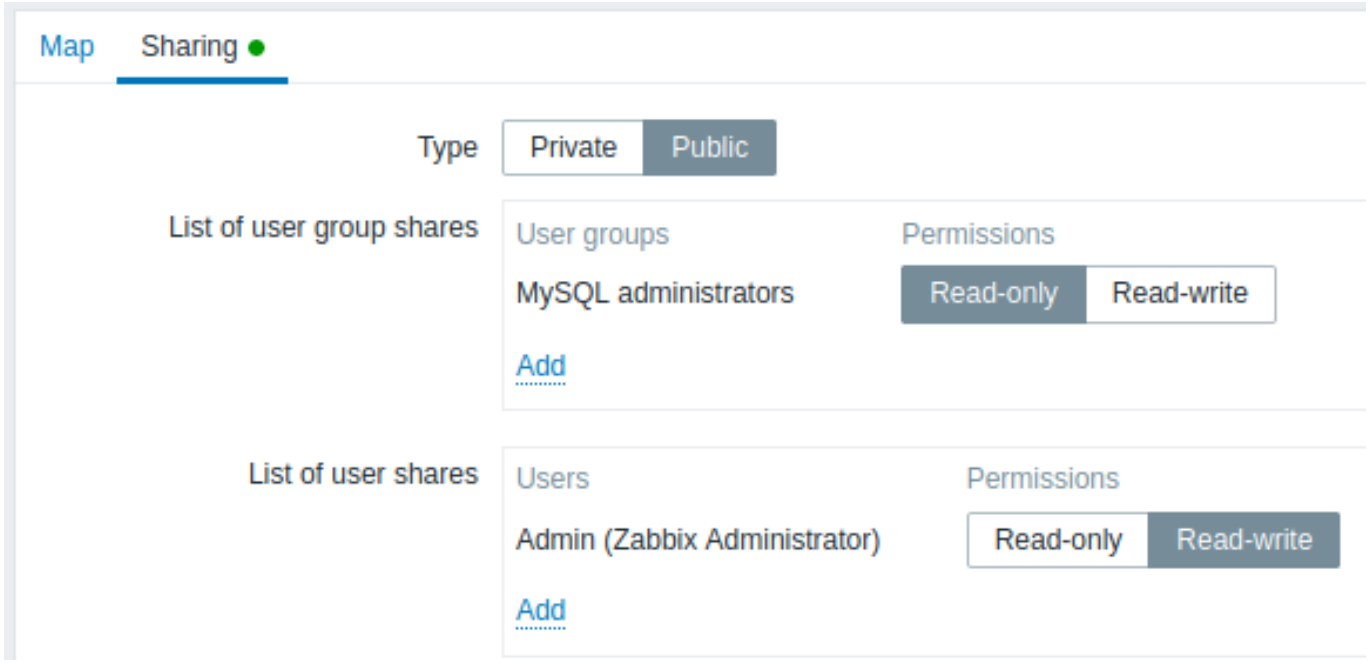
Todos os campos obrigatórios estão marcados com um asterisco vermelho.

Atributos gerais do mapa:

Parâmetro	Descrição
Proprietário	Nome do proprietário do mapa.
Nome	Nome único do mapa.
Largura	Largura do mapa em pixels.
Altura	Altura do mapa em pixels.
Imagem de fundo	Usar imagem de fundo: **Sem imagem** - sem imagem de fundo (fundo branco) **Imagem** - imagem selecionada para ser usada como imagem de fundo. Nenhum redimensionamento é realizado. Você pode usar um mapa geográfico ou qualquer outra imagem para aprimorar seu mapa.
Mapeamento automático de ícones	Você pode definir o uso de um mapeamento automático de ícones, configurado em *Administração → Geral → Mapeamento de ícones*. O mapeamento de ícones permite mapear certos ícones para determinados campos de inventário do host.
Destaque de ícone	Se você marcar esta caixa, os elementos do mapa serão destacados. Elementos com um trigger ativo receberão um fundo circular, na mesma cor do trigger de maior severidade. Além disso, uma linha verde grossa será exibida ao redor do círculo se todos os problemas forem reconhecidos. Elementos com status ”desabilitado” ou ”em manutenção” receberão um fundo quadrado, cinza e laranja, respectivamente. Veja também: Visualizando mapas
Marcar elementos na mudança de status do trigger	Uma mudança recente no status do trigger (problema recente ou resolução) será destacada com marcadores (triângulos vermelhos apontando para dentro) nos três lados do ícone do elemento que estiverem livres do rótulo. Os marcadores são exibidos por 30 minutos.
Exibir problemas	Selecione como os problemas são exibidos com um elemento do mapa: **Expandir problema único** - se houver apenas um problema, o nome do problema é exibido. Caso contrário, o número total de problemas é exibido. **Número de problemas** - o número total de problemas é exibido **Número de problemas e expandir o mais crítico** - o nome do problema mais crítico e o número total de problemas são exibidos. ’Mais crítico’ é determinado com base na severidade do problema e, se igual, no ID do evento do problema (ID maior ou problema mais recente exibido primeiro). Para um *elemento de mapa de trigger* é baseado na severidade do problema e, se igual, na posição do trigger na lista de triggers. No caso de vários problemas do mesmo trigger, o mais recente será exibido.
Rótulos avançados	Se você marcar esta caixa, poderá definir tipos de rótulos separados para tipos de elementos separados.
Tipo de rótulo do elemento do mapa	Tipo de rótulo usado para elementos do mapa: **Rótulo** - rótulo do elemento do mapa **Endereço IP** - endereço IP **Nome do elemento** - nome do elemento (por exemplo, nome do host) **Apenas status** - apenas status (OK ou PROBLEMA) **Nada** - nenhum rótulo é exibido
Localização do rótulo do elemento do mapa	Localização do rótulo em relação ao elemento do mapa: **Inferior** - abaixo do elemento do mapa **Esquerda** - à esquerda **Direita** - à direita **Superior** - acima do elemento do mapa
Exibição de problemas	Exibir a contagem de problemas como: **Todos** - a contagem total de problemas será exibida **Separados** - a contagem de problemas não reconhecidos será exibida separadamente como um número da contagem total de problemas **Apenas não reconhecidos** - apenas a contagem de problemas não reconhecidos será exibida
Severidade mínima do trigger	Problemas abaixo do nível mínimo de severidade selecionado não serão exibidos no mapa. Por exemplo, com *Aviso* selecionado, alterações com triggers de nível *Informação* e *Não classificado* não serão refletidas no mapa.
Mostrar problemas suprimidos	Marque a caixa para exibir problemas que, de outra forma, seriam suprimidos (não exibidos) devido à manutenção do host.
URLs	URLs (até 2048 caracteres) para cada tipo de elemento podem ser definidas. Um rótulo para a URL também pode ser definido. Eles serão exibidos como links quando um usuário clicar no elemento no modo de visualização do mapa. Macros podem ser usadas em nomes e valores de URLs de mapas. Para uma lista completa, consulte macros suportadas e pesquise por ’nomes e valores de URLs de mapas’.

Compartilhamento

A guia **Compartilhamento** contém o tipo de mapa, bem como as opções de compartilhamento (grupos de usuários, usuários) para mapas privados:



Parâmetro	Descrição
Tipo	Selecione o tipo de mapa: **Privado** - o mapa é visível apenas para grupos de usuários e usuários selecionados **Público** - o mapa é visível para todos
Lista de compartilhamento de grupos de usuários	Selecione os grupos de usuários que terão acesso ao mapa. Você pode permitir acesso somente leitura ou leitura e gravação.
Lista de compartilhamento de usuários	Selecione os usuários que terão acesso ao mapa. Você pode permitir acesso somente leitura ou leitura e gravação.

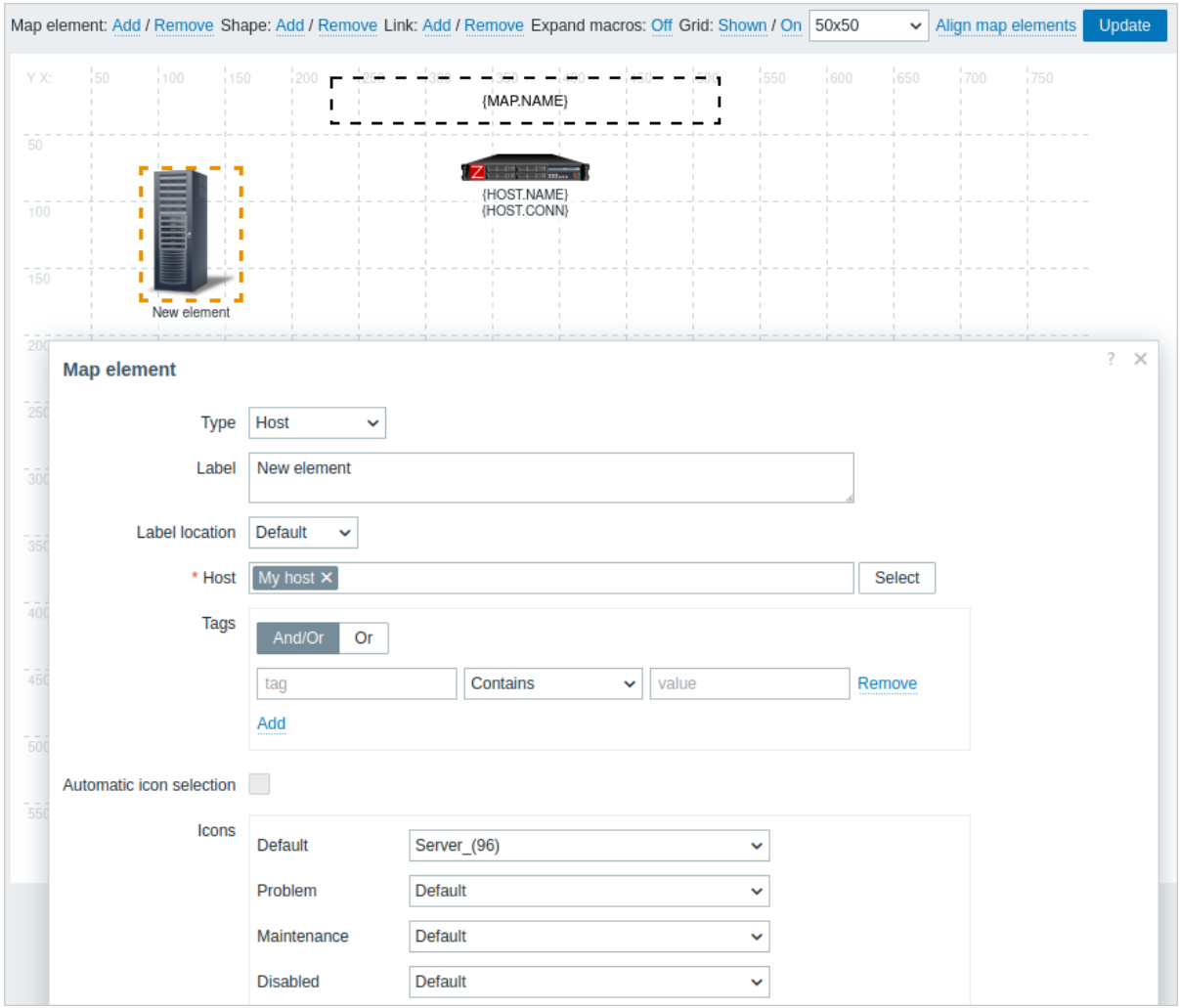
Ao clicar em *Adicionar* para salvar este mapa, você criou um mapa vazio com um nome, dimensões e algumas preferências. Agora você precisa adicionar alguns elementos. Para isso, clique em *Editar* na lista de mapas para abrir a área editável.

Adicionando elementos

Para adicionar um elemento, clique em *Adicionar* ao lado de `Elemento do mapa`. O novo elemento aparecerá no canto superior esquerdo do mapa. Arraste e solte-o onde desejar.

Observe que, com a opção Grade ”Ligada”, os elementos sempre se alinharão à grade (você pode escolher vários tamanhos de grade no menu suspenso, além de ocultar/exibir a grade). Se quiser posicionar elementos em qualquer lugar sem alinhamento, altere a opção para ”Desligada”. (Você pode alinhar elementos aleatórios à grade posteriormente, clicando em *Alinhar elementos do mapa*.)

Agora que você já tem alguns elementos no lugar, talvez queira começar a diferenciá-los dando nomes, etc. Ao clicar no elemento, um formulário é exibido e você pode definir o tipo de elemento, dar um nome, escolher um ícone diferente, etc.



Atributos do elemento do mapa:

Parâmetro	Descrição
Tipo	Tipo do elemento: **Host** - ícone representando o status de todos os triggers do host selecionado **Mapa** - ícone representando o status de todos os elementos de um mapa **Trigger** - ícone representando o status de um ou mais triggers **Grupo de hosts** - ícone representando o status de todos os triggers de todos os hosts pertencentes ao grupo selecionado **Imagem** - um ícone, não vinculado a nenhum recurso
Rótulo	Rótulo do ícone, qualquer string. Macros e strings multilinha podem ser usadas. Expressões de macros são suportadas neste campo, mas apenas com as funções `avg`, `last`, `min` e `max`, com o tempo como parâmetro (por exemplo, `{?avg(/host/key,1h)}`). Para uma lista completa de macros suportadas, consulte macros suportadas e pesquise por 'rótulos de elementos do mapa'.
Localização do rótulo	Localização do rótulo em relação ao ícone: **Padrão** - localização padrão do rótulo do mapa **Inferior** - abaixo do ícone **Esquerda** - à esquerda **Direita** - à direita **Superior** - acima do ícone
Host	Informe o host se o tipo de elemento for 'Host'. Este campo é autocompletável, então ao começar a digitar o nome de um host será exibida uma lista suspensa de hosts correspondentes. Role para baixo para selecionar. Clique em 'x' para remover o selecionado.
Mapa	Selecione o mapa se o tipo de elemento for 'Mapa'. Este campo é autocompletável, então ao começar a digitar o nome de um mapa será exibida uma lista suspensa de mapas correspondentes. Role para baixo para selecionar. Clique em 'x' para remover o selecionado.

Parâmetro	Descrição
Triggers	Se o tipo de elemento for 'Trigger', selecione um ou mais triggers no campo *Novos triggers* abaixo e clique em *Adicionar*. A ordem dos triggers selecionados pode ser alterada, mas apenas dentro da mesma severidade de triggers. A seleção de múltiplos triggers também afeta a resolução de macros {HOST.*} tanto nos modos de edição quanto de visualização. No modo de edição, as primeiras macros {HOST.*} exibidas serão resolvidas dependendo do primeiro trigger na lista (com base na severidade do trigger). O modo de visualização depende do parâmetro Exibir problemas nos atributos gerais do mapa: - Se o modo *Expandir único problema* for escolhido, as primeiras macros {HOST.*} exibidas serão resolvidas dependendo do último trigger de problema detectado (não importando a severidade) ou do primeiro trigger na lista (caso nenhum problema seja detectado). - Se o modo *Número de problemas e expandir o mais crítico* for escolhido, as primeiras macros {HOST.*} exibidas serão resolvidas dependendo da severidade do trigger.
Grupo de hosts	Informe o grupo de hosts se o tipo de elemento for 'Grupo de hosts'. Este campo é autocompletável, então ao começar a digitar o nome de um grupo será exibida uma lista suspensa de grupos correspondentes. Role para baixo para selecionar. Clique em 'x' para remover o selecionado.
Tags	Especifique tags para limitar o número de problemas exibidos no widget. É possível incluir ou excluir tags e valores de tags específicos. Várias condições podem ser definidas. A correspondência do nome da tag diferencia maiúsculas de minúsculas. Há vários operadores disponíveis para cada condição: **Existe** - inclui os nomes de tags especificados **Igual** - inclui os nomes e valores de tags especificados (diferencia maiúsculas de minúsculas) **Contém** - inclui os nomes de tags especificados onde os valores das tags contenham a string inserida (correspondência de substring, não diferencia maiúsculas de minúsculas) **Não existe** - exclui os nomes de tags especificados **Diferente** - exclui os nomes e valores de tags especificados (diferencia maiúsculas de minúsculas) **Não contém** - exclui os nomes de tags especificados onde os valores das tags contenham a string inserida (correspondência de substring, não diferencia maiúsculas de minúsculas) Existem dois tipos de cálculo para as condições: **E/Ou** - todas as condições devem ser atendidas, condições com o mesmo nome de tag serão agrupadas pela condição Ou **Ou** - basta que uma condição seja atendida Este campo está disponível para os tipos de elemento host e grupo de hosts.
Seleção automática de ícone	Neste caso, um mapeamento de ícone será usado para determinar qual ícone exibir.
Ícones	Você pode escolher exibir ícones diferentes para o elemento nestes casos: padrão, problema, manutenção, desabilitado.
Coordenada X	Coordenada X do elemento do mapa.
Coordenada Y	Coordenada Y do elemento do mapa.
URLs	URLs específicas do elemento (até 2048 caracteres) podem ser definidas para o elemento. Um rótulo para a URL também pode ser definido. Elas serão exibidas como links quando um usuário clicar no elemento no modo de visualização do mapa. Se o elemento tiver suas próprias URLs e houver URLs em nível de mapa para seu tipo definidas, elas serão combinadas no mesmo menu. Macros podem ser usadas em nomes e valores de elementos do mapa. Para uma lista completa, consulte macros suportadas e pesquise por 'nomes e valores de URL do mapa'.

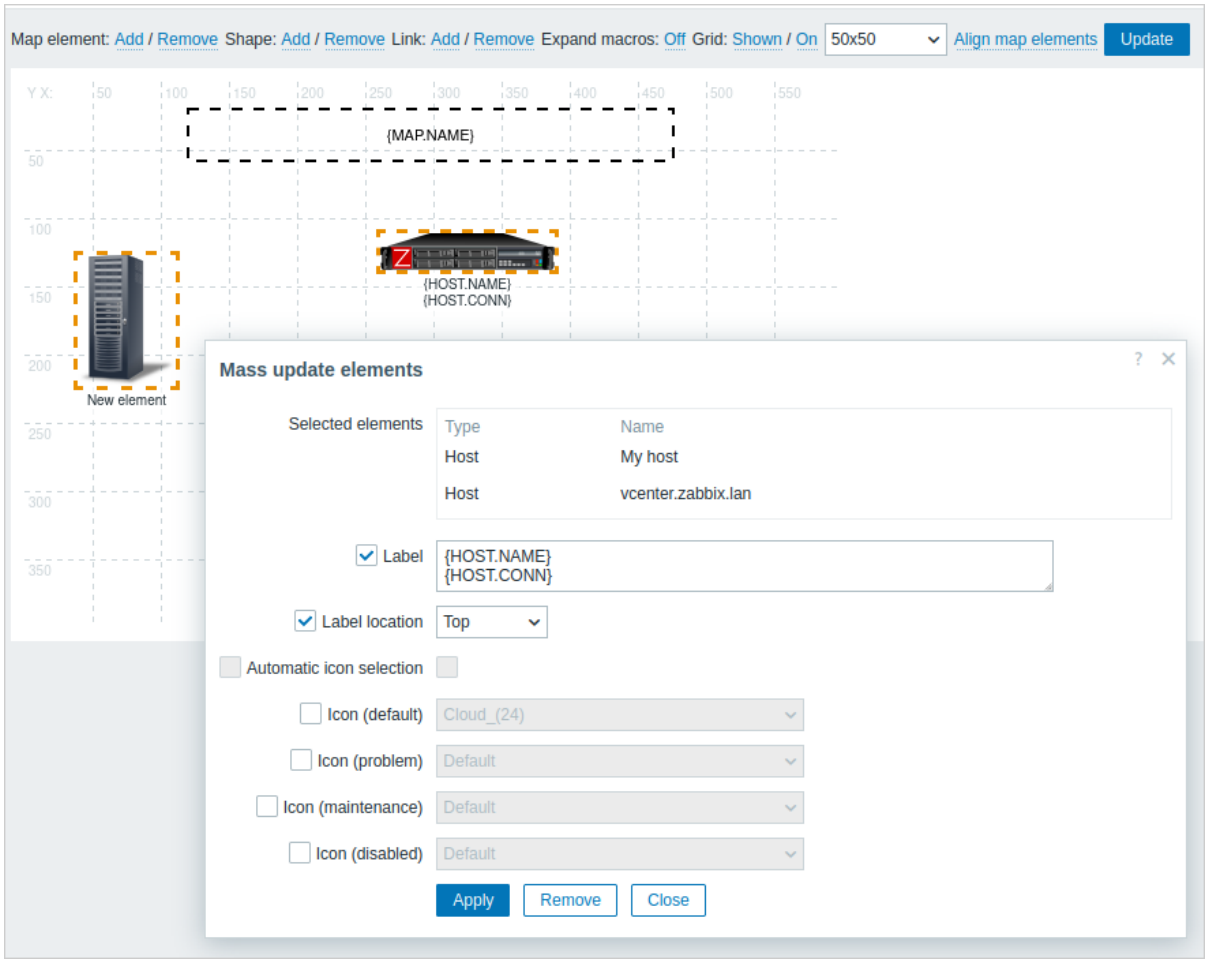
Attention:
Os elementos adicionados não são salvos automaticamente. Se você navegar para fora da página, todas as alterações podem ser perdidas.
Portanto, é uma boa ideia clicar no botão **Atualizar** no canto superior direito. Uma vez clicado, as alterações são salvas independentemente do que você escolher no pop-up seguinte.
As opções de grade selecionadas também são salvas com cada mapa.

Selecionando elementos

Para selecionar elementos, selecione um e, em seguida, mantenha pressionado *Ctrl* para selecionar o outras.

Você também pode selecionar vários elementos arrastando um retângulo na área editável e selecionando todos os elementos nela.

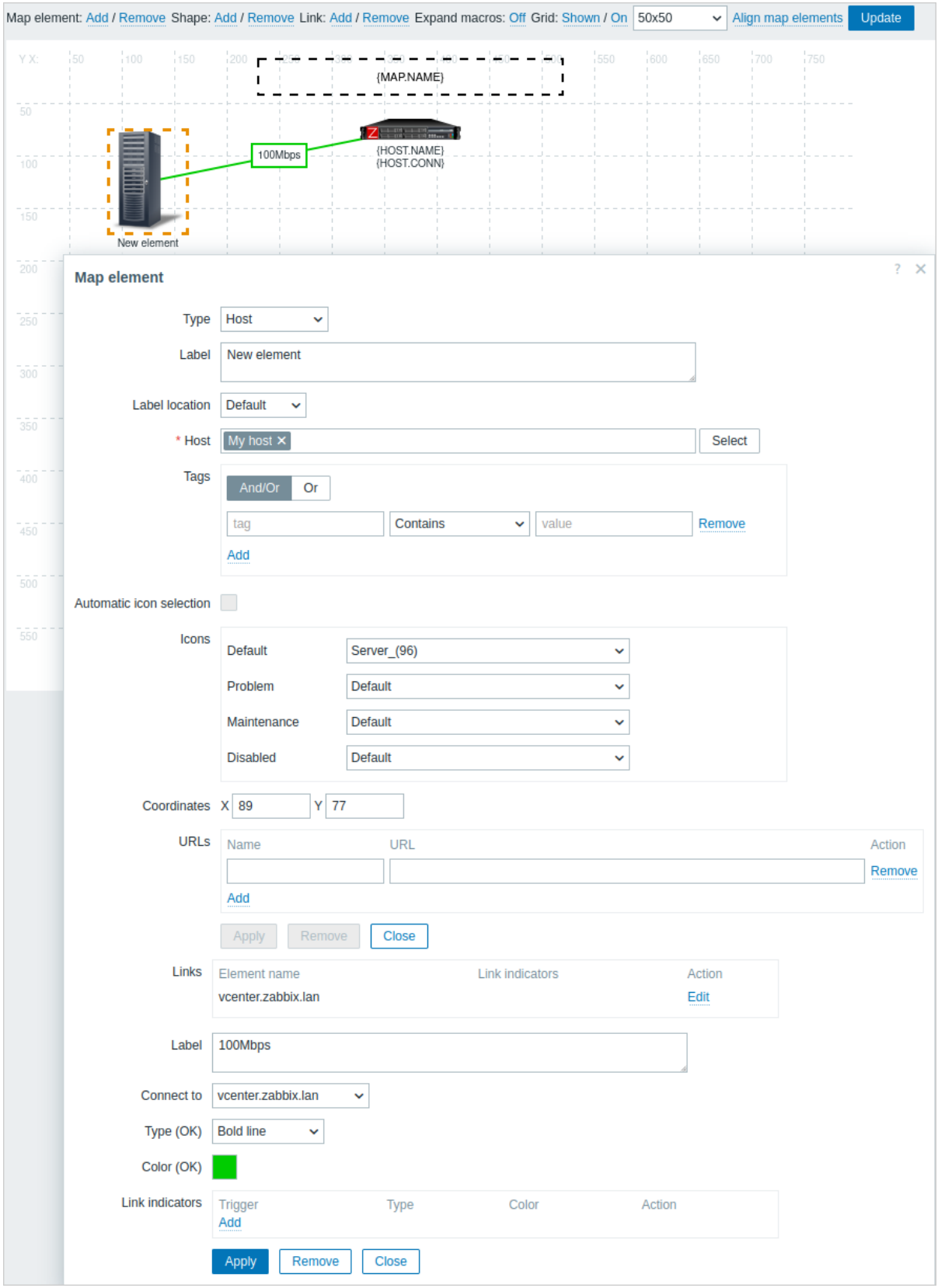
Depois de selecionar mais de um elemento, a forma de propriedade do elemento muda para o modo de atualização em massa para que você possa alterar os atributos de elementos de uma só vez. Para fazer isso, marque o atributo usando a caixa de seleção e insira um novo valor para ele. Você pode usar macros aqui (por exemplo, {HOST.NAME} para o rótulo do elemento).



Ligando elementos

Depois de colocar alguns elementos no mapa, é hora de começar a ligá-los. Para ligar dois elementos, você deve primeiro selecioná-los. Com os elementos selecionados, clique em *Adicionar* ao lado de Link.

Com um link criado, o formulário de elemento único agora contém uma seção adicional *Links*. Clique em *Editar* para editar os atributos do link.



Atributos do link:

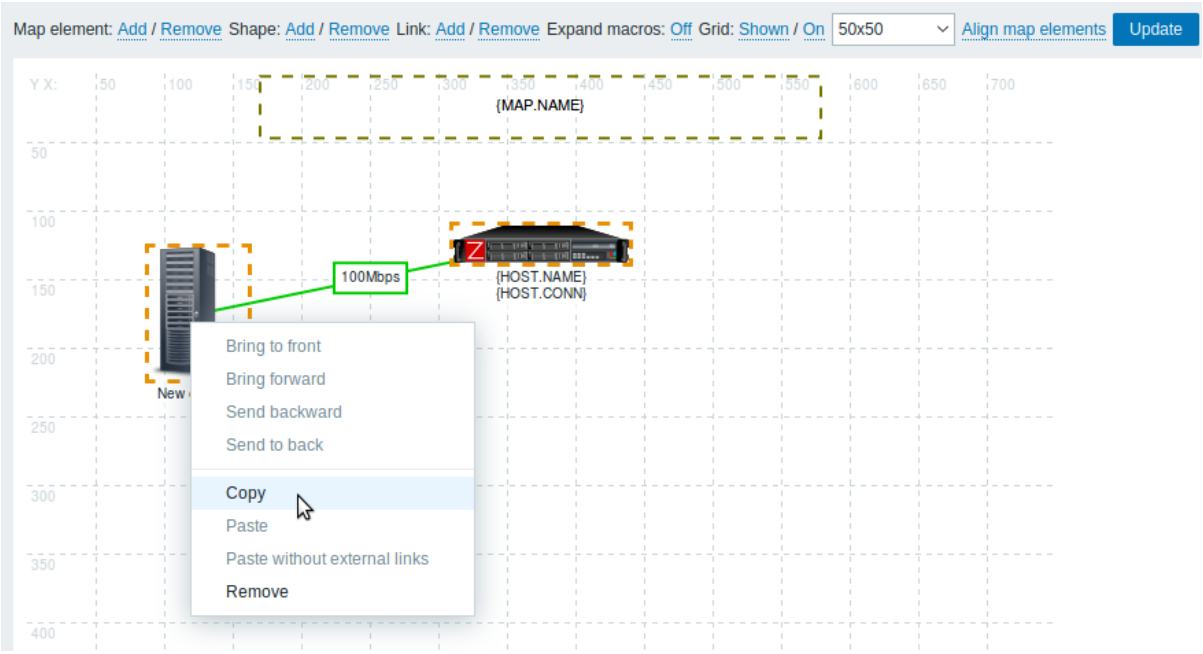
Parâmetro	Descrição
Rótulo	Rótulo que será exibido sobre o link. Expressões macros são suportadas neste campo, mas apenas com as funções `avg`, `last`, `min` e `max`, com o tempo como parâmetro (por exemplo, `{?avg(/host/key,1h)}`).
Conectar a	O elemento ao qual o link se conecta.

Parâmetro	Descrição
Tipo (OK)	Estilo padrão do link: **Linha** - linha simples **Linha em negrito** - linha em negrito **Pontos** - pontos **Linha tracejada** - linha tracejada
Cor (OK)	Cor padrão do link.
Indicadores de link	Lista de triggers vinculados ao link. Caso um trigger tenha status PROBLEM, seu estilo será aplicado ao link.

Movendo e copiando e colando elementos

Vários elementos selecionados podem ser **movidos** para outro lugar no mapa clicando em um dos elementos selecionados, mantendo pressionado o mouse botão e movendo o cursor para o local desejado.

Um ou mais elementos podem ser **copiados** selecionando os elementos e, em seguida, clicando em um elemento selecionado com o botão direito do mouse e selecionando *Copiar* do menu.



Para colar os elementos, clique em uma área do mapa com o botão direito do mouse e selecione *Colar* no menu. O *Colar sem links externos* opção irá colar os elementos retendo apenas os links que estão entre os elementos selecionados.

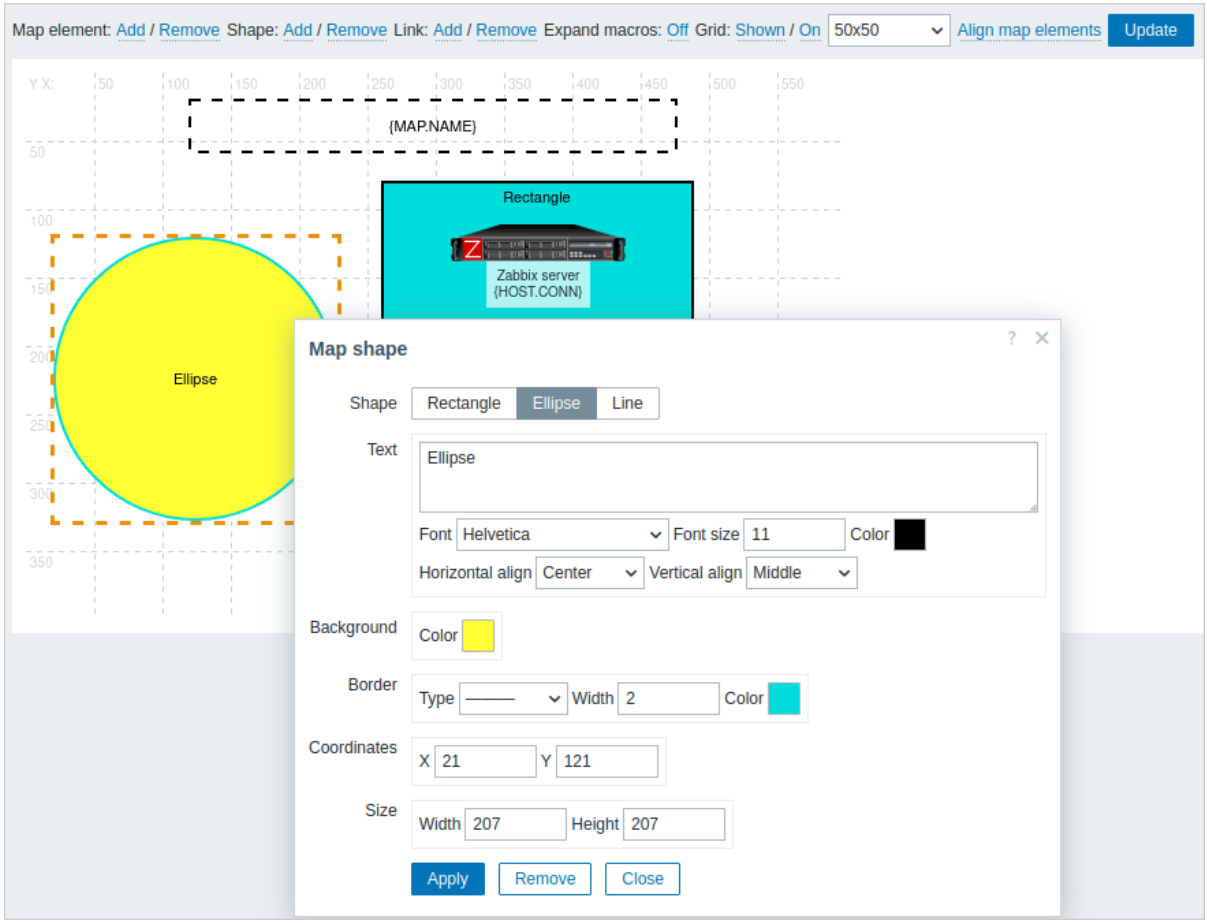
Copiar e colar funciona na mesma janela do navegador. Atalhos do teclado não são suportados.

Adicionando formas

Além dos elementos do mapa, também é possível adicionar algumas formas. As formas não são elementos do mapa; são apenas uma representação visual. Por exemplo, uma forma de retângulo pode ser usada como fundo para agrupar alguns hosts. É possível adicionar formas de retângulo e elipse.

Para adicionar uma forma, clique em *Adicionar* ao lado de Forma. A nova forma aparecerá no canto superior esquerdo do mapa. Arraste e solte onde quiser.

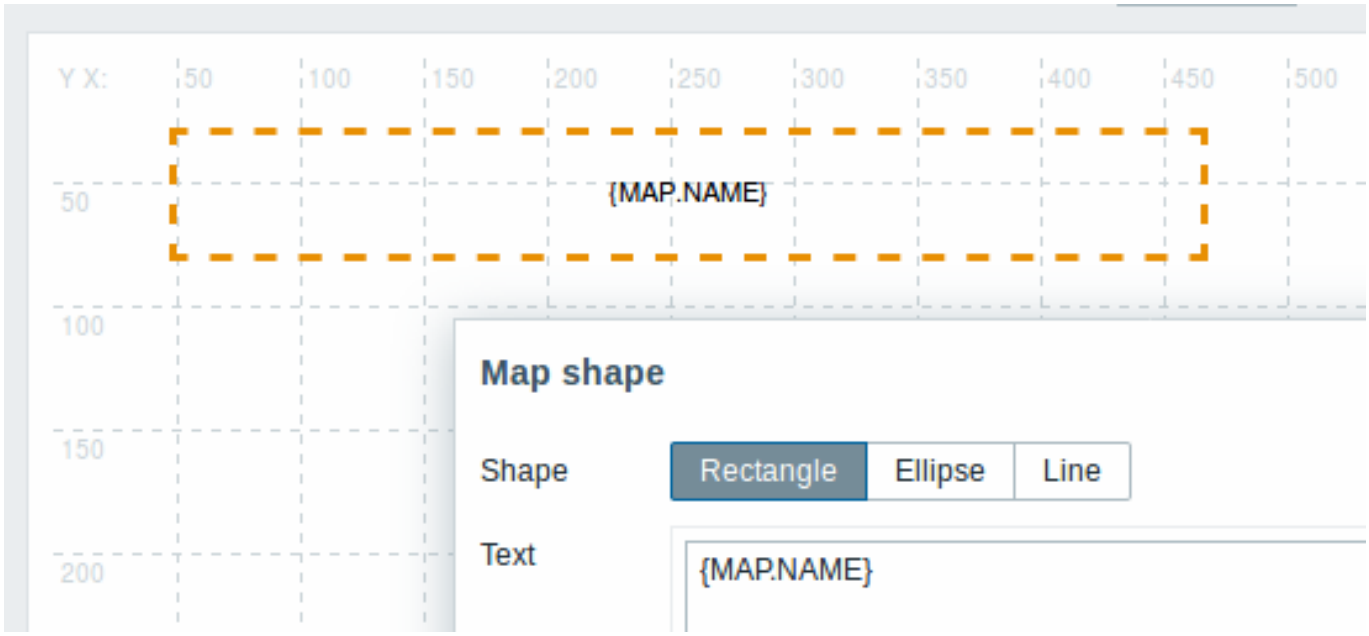
Uma nova forma é adicionada com cores padrão. Ao clicar na forma, um formulário é exibido e você pode personalizar a aparência da forma, adicionar texto, etc.



Para selecionar formas, selecione uma e depois mantenha pressionada a tecla *Ctrl* para selecionar as outras. Com várias formas selecionadas, propriedades comuns podem ser atualizadas em massa, da mesma forma que com os elementos.

É possível adicionar texto nas formas. Expressões macros são suportadas no texto, mas apenas com as funções `avg`, `last`, `min` e `max`, com tempo como parâmetro (por exemplo, `{?avg(/host/key,1h)}`).

Para exibir apenas texto, a forma pode ser tornada invisível removendo a borda da forma (selecione 'Nenhuma' no campo *Borda*). Por exemplo, observe como a macro {MAP.NAME}, visível na captura de tela acima, é na verdade uma forma de retângulo com texto, que pode ser vista ao clicar na macro:



{MAP.NAME} é resolvida para o nome do mapa configurado ao visualizar o mapa.

Se hiperlinks forem usados no texto, eles se tornam clicáveis ao visualizar o mapa.

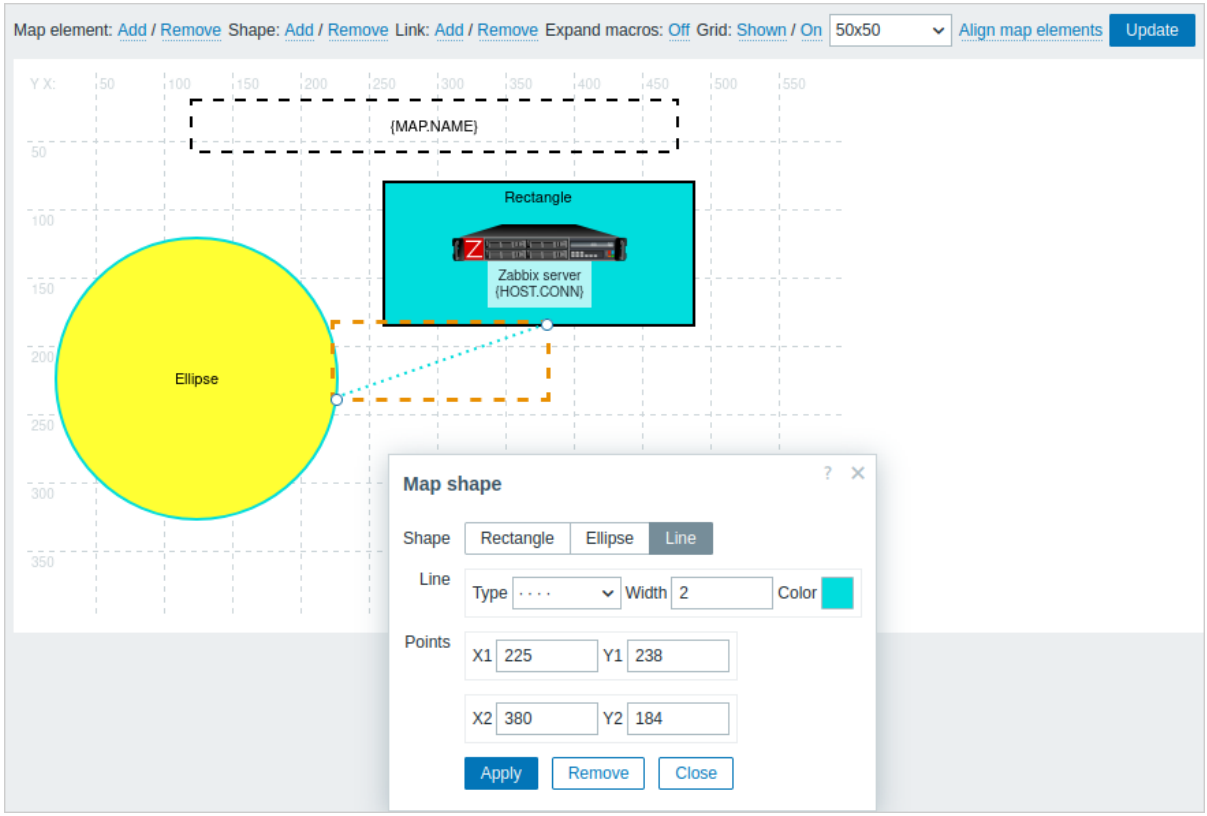
A quebra de linha para texto está sempre "ativada" dentro das formas. No entanto, dentro de uma elipse, as linhas são quebradas como se a elipse fosse um retângulo. A quebra de palavras não é implementada, portanto, palavras longas (palavras que não

cabem na forma) não são quebradas, mas são mascaradas (na página de edição do mapa) ou cortadas (em outras páginas com mapas).

Adicionando linhas

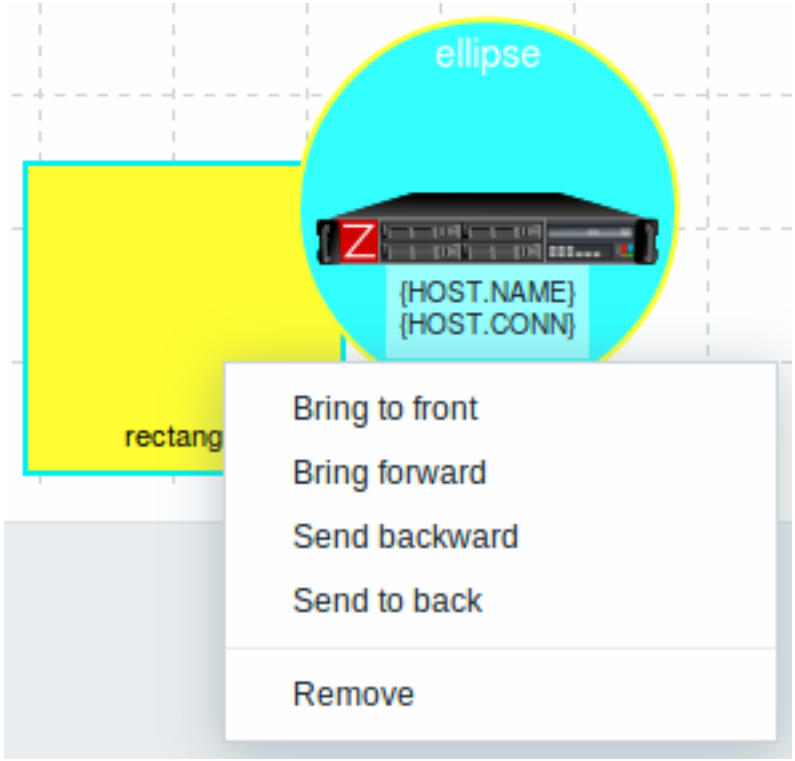
Além das formas, também é possível adicionar algumas linhas. As linhas podem ser usado para vincular elementos ou formas em um mapa.

Para adicionar uma linha, clique em *Adicionar* ao lado de Forma. Uma nova forma aparecerá em no canto superior esquerdo do mapa. Selecione-o e clique em *Linha* no formulário de edição para alterar a forma em uma linha. Em seguida, ajuste a linha propriedades, como tipo de linha, largura, cor, etc.



Ordenando formas e linhas

Para trazer uma forma para frente da outra (ou vice-versa), clique na forma com o botão direito do mouse para abrir o menu de formas do mapa.

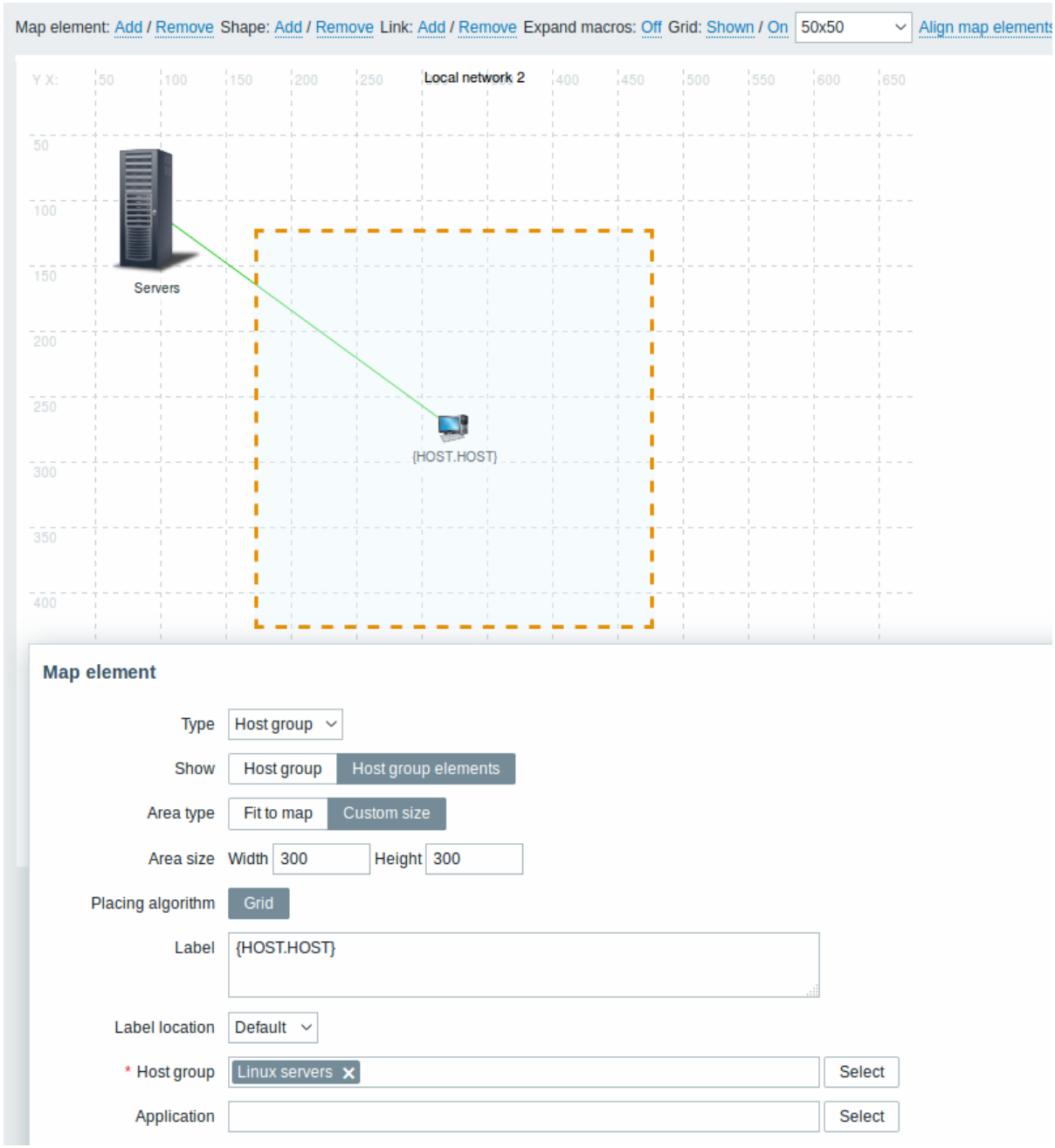


2 Elementos do grupo de hosts

Visão geral

Esta seção explica como adicionar um elemento do tipo *Grupo de hosts* ao configurar um mapa de rede.

Configuração



Todos os campos obrigatórios estão marcados com um asterisco vermelho.

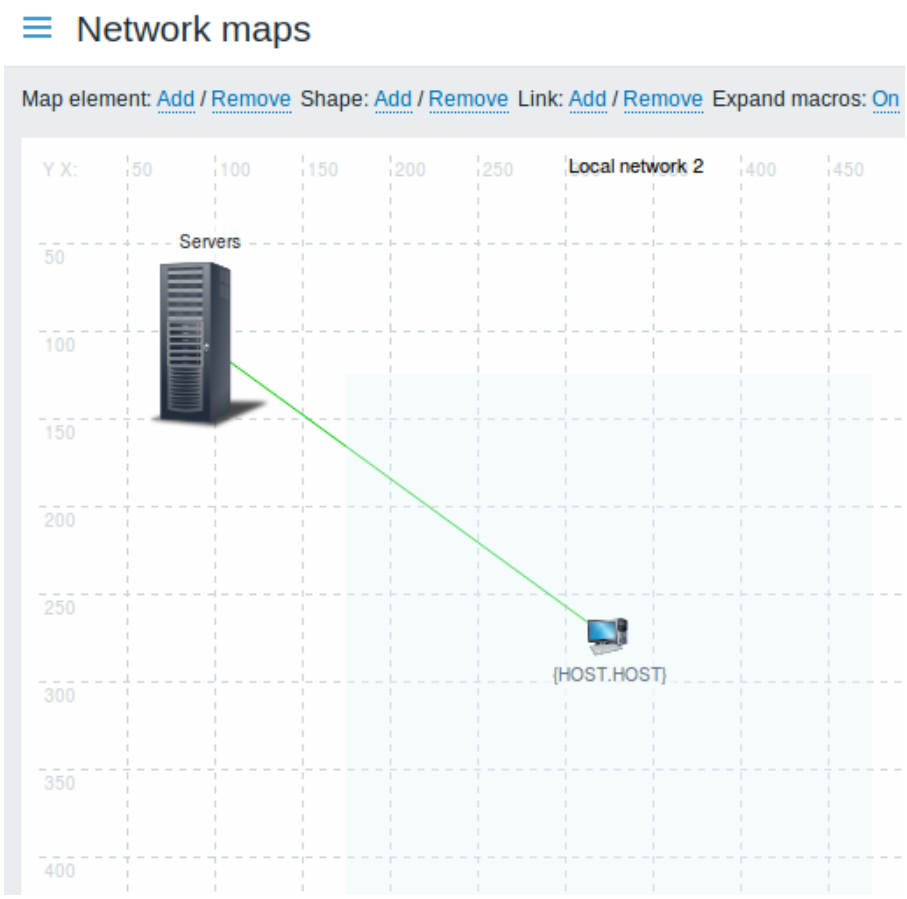
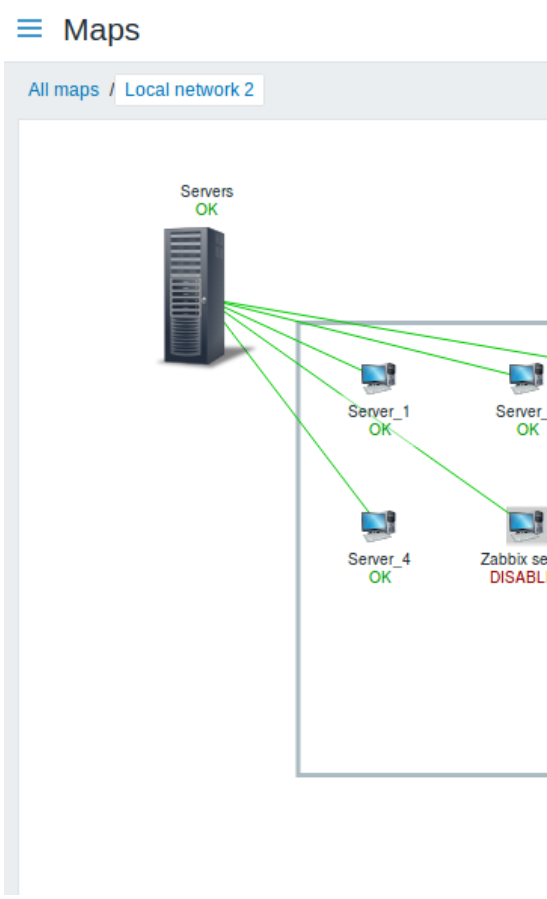
Esta tabela consiste em parâmetros típicos para o tipo de elemento *Grupo de hosts*:

Parâmetro	Descrição
Tipo	Selecione o tipo do elemento: **Grupo de hosts** - Ícone representando o status de todos os triggers de todos os hosts pertencentes ao grupo selecionado.
Mostrar	Opções de exibição: **Grupo de hosts** - Selecionando esta opção resultará em um único ícone exibindo informações correspondentes sobre o determinado grupo de hosts. **Elementos do grupo de hosts** - Selecionando esta opção resultará em vários ícones exibindo informações correspondentes sobre cada elemento individual (host) do determinado grupo de hosts.
Tipo de área	Esta configuração está disponível se *Mostrar* estiver definido como *Elementos do grupo de hosts*: **Ajustar ao mapa** - Todos os elementos do grupo de hosts são igualmente posicionados dentro do mapa. **Tamanho personalizado** - Uma configuração manual da área do mapa para todos os elementos do grupo de hosts a serem exibidos.

Parâmetro	Descrição
Tamanho da área	Esta configuração está disponível se *Tipo de área* estiver definido como *Tamanho personalizado*: **Largura** - Valor em pixels para especificar a largura da área do mapa. **Altura** - Valor em pixels para especificar a altura da área do mapa.
Algoritmo de posicionamento	**Grade** - Única opção disponível para exibir todos os elementos do grupo de hosts.
Rótulo	Rótulo do ícone, qualquer string. Macros e strings multilinha podem ser usadas em rótulos. Se o tipo do elemento do mapa for *Grupo de hosts*, especificar certas macros tem impacto na visualização do mapa, exibindo informações correspondentes sobre cada host individual. Por exemplo, se a macro {HOST.IP} for usada, a visualização de edição do mapa exibirá apenas a macro {HOST.IP} em si, enquanto a visualização do mapa incluirá e exibirá o endereço IP exclusivo de cada host.

Visualizando elementos de grupos de hosts

Esta opção está disponível se a opção *Elementos do grupo de hosts* for escolhida. Ao selecionar *Elementos do grupo de hosts* como a opção *mostrar*, você verá inicialmente apenas um ícone para o grupo de hosts. No entanto, ao salvar o mapa e depois ir para a visualização do mapa, você verá que o mapa inclui todos os elementos (hosts) do determinado grupo de hosts:

Visualização de edição do mapa	Visualização do mapa
	

Observe como a macro {HOST.HOST} é usada. Na edição do mapa, o nome da macro não é resolvido, enquanto na visualização do mapa todos os nomes únicos dos hosts são exibidos.

3 Indicadores de links

Visão geral

Você pode atribuir alguns gatilhos a um link entre elementos em um mapa de rede. Quando esses gatilhos entram em um estado de problema, o link pode refletir isso.

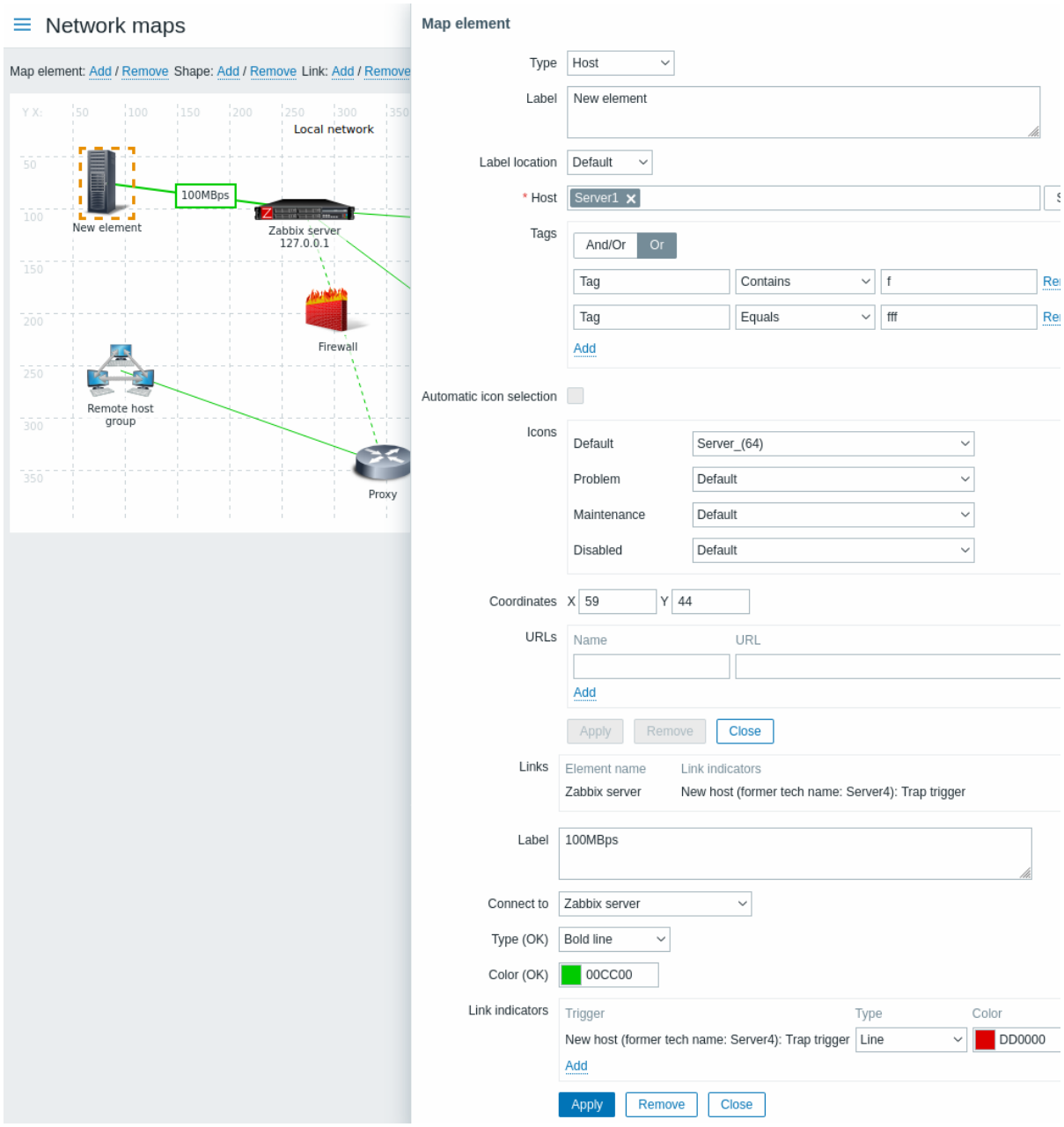
Ao configurar um link, você define o tipo e a cor do link padrão. Quando você atribui gatilhos a um link, você pode atribuir diferentes tipos de link e cores com esses gatilhos.

Se algum desses gatilhos entrar em um estado de problema, seu estilo de link e a cor será exibida no link. Então, talvez seu link padrão fosse uma linha verde. Agora, com o gatilho no estado do problema, seu link pode tornar-se vermelho em negrito (se você definiu assim).

Configuração

Para atribuir acionadores como indicadores de link, faça o seguinte:

- selecione um elemento do mapa
- clique em *Editar* na seção *Links* para obter o link apropriado
- clique em *Adicionar* no bloco *Indicadores de link* e selecione um ou mais gatilhos



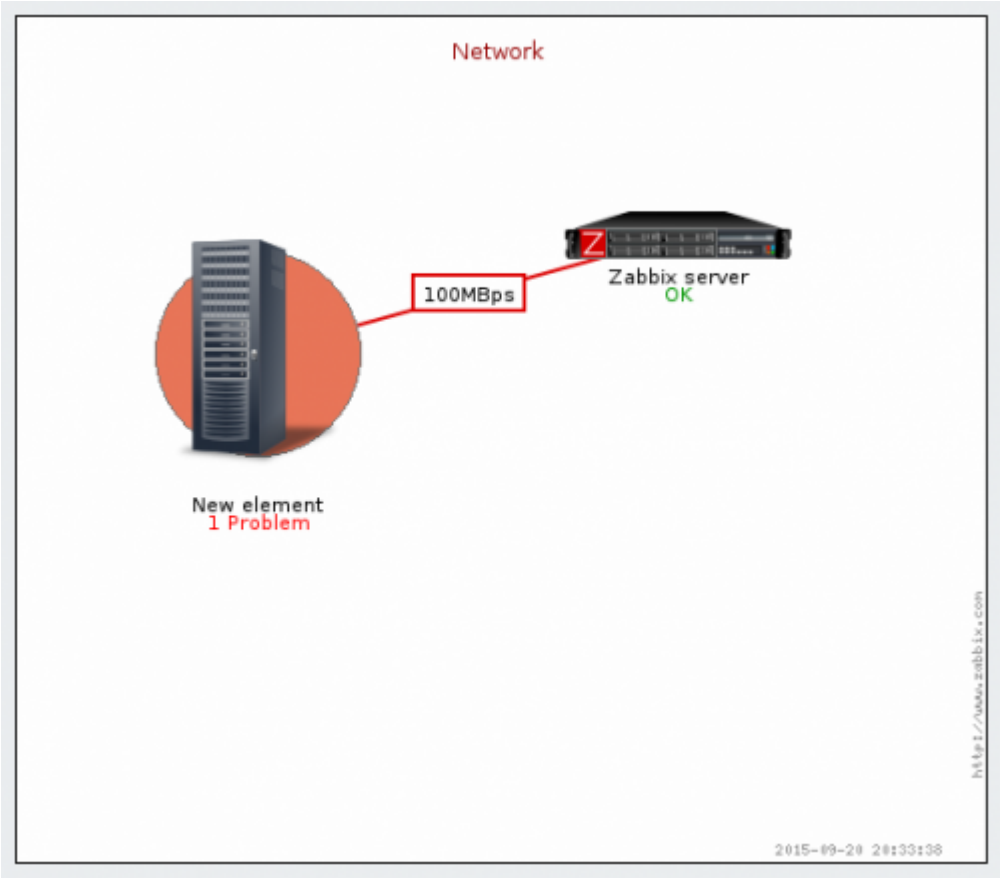
Todos os campos de entrada obrigatórios estão marcados com um asterisco vermelho.

Os gatilhos adicionados podem ser vistos na lista *Indicadores de link*.

Você pode definir o tipo e a cor do link para cada acionador diretamente do Lista. Quando terminar, clique em *Aplicar*, feche o formulário e clique em *Atualizar* para salvar as alterações do mapa.

Exibição

Em *Monitoramento* → *Mapas*, a respectiva cor será exibida no link se o trigger entrar em estado de problema.

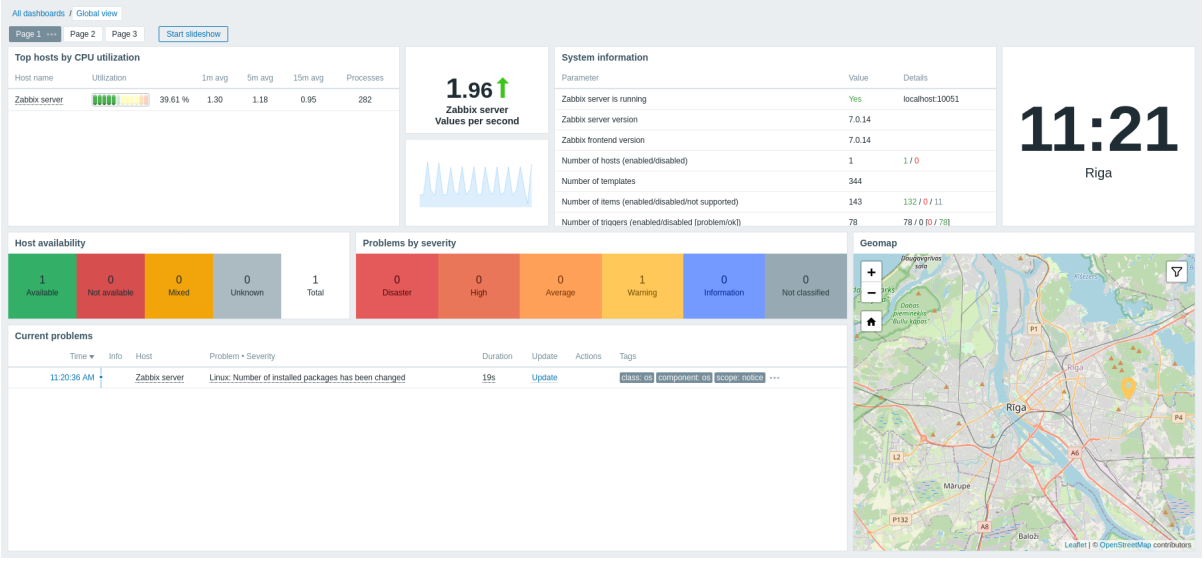


Note:
Se vários triggers entrarem em estado de problema, o problema com a maior severidade determinará o estilo e a cor do link. Se vários triggers com a mesma severidade forem atribuídos ao mesmo link do mapa, aquele com o menor ID terá precedência. Observe também que:

1. As configurações *Severidade mínima do trigger* e *Exibir problema suprimido* da configuração do mapa afetam quais problemas são considerados.
2. No caso de triggers com múltiplos problemas (geração múltipla de problemas), cada problema pode ter uma severidade diferente da severidade do trigger (alterada manualmente), pode ter tags diferentes (devido a macros) e pode ser suprimido.

3 Dashboards

Dashboards - tanto dashboards globais quanto dashboards de host - fornecem uma poderosa plataforma de visualização com widgets e ferramentas como gráficos modernos, mapas, slideshows e muito mais.



8 Templates e grupos de templates

Visão geral

O uso de templates é uma excelente maneira de reduzir o trabalho e simplificar a configuração do Zabbix. Um template é um conjunto de entidades que podem ser convenientemente aplicadas a vários hosts.

As entidades podem ser:

- items
- triggers
- gráficos
- dashboards
- regras de descoberta de baixo nível
- cenários web

Como muitos hosts na vida real são idênticos ou bastante semelhantes, é natural que o conjunto de entidades (items, triggers, gráficos,...) que você criou para um host possa ser útil para muitos outros. Claro, você poderia copiá-los para cada novo host, mas isso seria muito trabalho manual. Em vez disso, com templates você pode copiá-los para um template e então aplicar o template a quantos hosts forem necessários.

Quando um template é vinculado a um host, todas as entidades (items, triggers, gráficos,...) do template são adicionadas ao host. Os templates são atribuídos a cada host individualmente (e não a um grupo de hosts).

Os templates são frequentemente usados para agrupar entidades para serviços ou aplicativos específicos (como Apache, MySQL, PostgreSQL, Postfix...) e, em seguida, aplicados aos hosts que executam esses serviços.

Outro benefício do uso de templates é quando algo precisa ser alterado para todos os hosts. Alterar algo no nível do template uma vez propagará a alteração para todos os hosts vinculados.

Os templates são organizados em grupos de templates.

Continue para criar e configurar um template.

9 Modelos prontos para uso

Visão geral

O Zabbix fornece um conjunto crescente de templates pré-configurados para simplificar e agilizar a configuração dos alvos de monitoramento.

Todos os templates prontos para uso estão disponíveis em *Coleta de dados* → *Templates*.

Ao atualizar o Zabbix, os templates existentes não são atualizados automaticamente para evitar a sobrescrição de modificações personalizadas. Para atualizar os templates existentes ou adicionar novos, consulte Atualização de template.

Use a barra lateral para acessar informações sobre tipos específicos de templates e requisitos de operação.

Veja também:

- Vinculando um template
- Problemas conhecidos para templates

Atualização de template

Para atualizar templates existentes ou adicionar novos após uma atualização do Zabbix:

1. Acesse o repositório Git do Zabbix.
2. No seletor de branch (sob o rótulo *Source*), escolha o branch que corresponde à sua versão do Zabbix (por exemplo, *release/7.0* para o Zabbix 7.0).
3. Abra o template que deseja adicionar ou atualizar (por exemplo, MySQL by Zabbix agent).
4. Copie o conteúdo do arquivo do template (por exemplo, `template_db_mysql_agent.yaml`) e salve em um arquivo local.
5. No frontend do Zabbix, acesse *Coleta de dados* > *Templates* e importe o arquivo local.

1 Operação do template do Zabbix agent

Etapas para garantir a operação correta dos templates que coletam métricas com o Zabbix agent:

1. Certifique-se de que o Zabbix agent está instalado no host. Para checagens ativas, também certifique-se de que o endereço do Zabbix server/proxy está adicionado ao parâmetro 'ServerActive' no arquivo de configuração do agent.
2. Vincule o template ao host de destino (se o template não estiver disponível na sua instalação do Zabbix, talvez seja necessário importar o template primeiro).
3. Se necessário, ajuste os valores das macros do template.
4. Configure a instância monitorada para permitir o compartilhamento de dados com o Zabbix.

Uma descrição detalhada do template, incluindo a lista completa de macros, items e triggers, está disponível no arquivo README do template (acessível clicando no nome do template).

Os seguintes templates estão disponíveis:

- Apache por Zabbix agent
- Apache por Zabbix agent active
- HAProxy por Zabbix agent
- IIS por Zabbix agent
- IIS por Zabbix agent active
- Microsoft Exchange Server 2016 por Zabbix agent
- Microsoft Exchange Server 2016 por Zabbix agent active
- MySQL por Zabbix agent
- MySQL por Zabbix agent active
- Nginx por Zabbix agent
- Nginx por Zabbix agent active
- PHP-FPM por Zabbix agent
- PHP-FPM por Zabbix agent active
- PostgreSQL por Zabbix agent
- PostgreSQL por Zabbix agent active
- RabbitMQ cluster por Zabbix agent

2 Operação de template do Zabbix agent 2

Etapas para garantir a operação correta dos templates que coletam métricas com o Zabbix agent 2:

1. Certifique-se de que o agent 2 está instalado no host e que a versão instalada contém o plugin necessário. Em alguns casos, pode ser necessário atualizar o agent 2 primeiro.
2. Vincule o template ao host de destino (se o template não estiver disponível em sua instalação do Zabbix, talvez seja necessário importar o template primeiro).
3. Se necessário, ajuste os valores das macros do template. Observe que macros de usuário podem ser usadas para substituir parâmetros de configuração.
4. Configure a instância monitorada para permitir o compartilhamento de dados com o Zabbix.

> **Attention:**
> Os templates do Zabbix agent 2 funcionam em conjunto com os plugins. Embora a configuração básica possa ser feita simplesmente ajustando macros de usuário, a personalização mais profunda pode ser alcançada configurando o plugin em si. Por exemplo, se um plugin suportar sessões nomeadas, é possível monitorar várias entidades do mesmo tipo (por exemplo, MySQL1 e MySQL2) especificando uma sessão nomeada com seu próprio URI, nome de usuário e senha para cada entidade no arquivo de configuração.

Uma descrição detalhada de um template, incluindo a lista completa de macros, items e triggers, está disponível no arquivo README do template (acessível clicando no nome do template).

Os seguintes templates estão disponíveis:

- Ceph por Zabbix agent 2
- Docker
- Memcached
- MongoDB cluster por Zabbix agent 2
- MongoDB node por Zabbix agent 2
- MySQL por Zabbix agent 2
- MySQL por Zabbix agent 2 ativo
- Oracle por Zabbix agent 2

- PostgreSQL por Zabbix agent 2
- PostgreSQL por Zabbix agent 2 ativo
- Redis
- SMART por Zabbix agent 2
- SMART por Zabbix agent 2 ativo
- Systemd por Zabbix agent 2
- Certificado de website por Zabbix agent 2

3 Operação de template HTTP

Etapas para garantir a operação correta de templates que coletam métricas com o HTTP agent:

1. Crie um host no Zabbix que corresponda ao alvo de monitoramento. Especifique o endereço IP/nome DNS do alvo de monitoramento como a interface principal do host. Isso é necessário para que a macro {HOST.CONN} seja resolvida corretamente nos items do template.
2. Vincule o template ao host criado na etapa 1 (se o template não estiver disponível em sua instalação do Zabbix, talvez seja necessário importar o template primeiro).
3. Se necessário, ajuste os valores das macros do template.
4. Configure a instância que está sendo monitorada para permitir o compartilhamento de dados com o Zabbix.

Uma descrição detalhada de um template, incluindo a lista completa de macros, items e triggers, está disponível no arquivo README do template (acessível clicando no nome do template).

Os seguintes templates estão disponíveis:

- Acronis Cyber Protect Cloud por HTTP
- Apache por HTTP
- Asterisk por HTTP
- AWS por HTTP
- AWS Cost Explorer por HTTP
- AWS EC2 por HTTP
- AWS ECS Cluster por HTTP
- AWS ECS Serverless Cluster por HTTP
- AWS ELB Application Load Balancer por HTTP
- AWS ELB Network Load Balancer por HTTP
- AWS Lambda por HTTP
- AWS RDS instance por HTTP
- AWS S3 bucket por HTTP
- Azure por HTTP
- Cisco Meraki organization por HTTP
- Cisco SD-WAN por HTTP
- Cisco Secure Firewall Threat Defense por HTTP
- ClickHouse por HTTP
- Cloudflare por HTTP
- CockroachDB por HTTP
- Control-M enterprise manager por HTTP
- Control-M server por HTTP
- DELL PowerEdge R720 por HTTP
- DELL PowerEdge R740 por HTTP
- DELL PowerEdge R820 por HTTP
- DELL PowerEdge R840 por HTTP
- Elasticsearch Cluster por HTTP
- Envoy Proxy por HTTP
- Etcd por HTTP
- FortiGate por HTTP
- GitHub repository por HTTP
- GitLab por HTTP
- Google Cloud Platform (GCP) por HTTP
- Hadoop por HTTP
- HAProxy por HTTP
- HashiCorp Consul Cluster por HTTP
- HashiCorp Consul Node por HTTP
- HashiCorp Nomad por HTTP
- HashiCorp Vault por HTTP

- Câmera Hikvision por HTTP
- HPE iLO por HTTP
- HPE MSA 2040 Storage por HTTP
- HPE MSA 2060 Storage por HTTP
- HPE Primera por HTTP
- HPE Synergy por HTTP
- InfluxDB por HTTP
- Jenkins por HTTP
- Kubernetes API server por HTTP
- Kubernetes cluster state por HTTP
- Kubernetes Controller manager por HTTP
- Kubernetes kubelet por HTTP
- Kubernetes nodes por HTTP
- Kubernetes Scheduler por HTTP
- MantisBT por HTTP
- Microsoft 365 reports por HTTP
- Microsoft SharePoint por HTTP
- NetApp AFF A700 por HTTP
- Nextcloud por HTTP
- NGINX por HTTP
- NGINX Plus por HTTP
- Nutanix Prism Element por HTTP
- OpenStack por HTTP
- OpenWeatherMap por HTTP
- Oracle Cloud por HTTP
- Palo Alto PA-440 por HTTP
- PHP-FPM por HTTP
- Proxmox VE por HTTP
- Pure Storage FlashArray v1 e v2 por HTTP
- RabbitMQ cluster por HTTP
- Ribbon SBC Edge por HTTP
- Ribbon SBC SWe CE por HTTP
- Ribbon SBC SWe Core por HTTP
- TiDB por HTTP
- TiDB PD por HTTP
- TiDB TiKV por HTTP
- Travis CI por HTTP
- Veeam Backup Enterprise Manager por HTTP
- Veeam Backup and Replication por HTTP
- VeloCloud SD-WAN por HTTP
- VeloCloud SD-WAN Edge por HTTP
- YugabyteDB por HTTP
- ZooKeeper por HTTP

4 Operação de template IPMI

Templates IPMI não requerem nenhuma configuração específica. Para começar a monitorar, vincule o template ao host de destino (se o template não estiver disponível em sua instalação do Zabbix, talvez seja necessário importar o template primeiro).

Uma descrição detalhada de um template, incluindo a lista completa de macros, items e triggers, está disponível no arquivo README do template (acessível clicando no nome do template).

Template disponível:

- Chassis por IPMI

5 Operação do template JMX

Etapas para garantir a operação correta de templates que coletam métricas por JMX:

1. Certifique-se de que o Java gateway do Zabbix está instalado e configurado corretamente.
2. Vincule o template ao host de destino. O host deve ter a interface JMX configurada.
 Se o template não estiver disponível em sua instalação do Zabbix, talvez seja necessário importar o template primeiro.
3. Se necessário, ajuste os valores das macros do template.

4. Configure a instância monitorada para permitir o compartilhamento de dados com o Zabbix.

Uma descrição detalhada do template, incluindo a lista completa de macros, items e triggers, está disponível no arquivo README do template (acessível clicando no nome do template).

Os seguintes templates estão disponíveis:

- Apache ActiveMQ por JMX
- Apache Cassandra por JMX
- Apache Kafka por JMX
- Apache Tomcat por JMX
- GridGain por JMX
- Ignite por JMX
- Jira Data Center por JMX
- WildFly Domain por JMX
- WildFly Server por JMX

6 Operação do template ODBC

Etapas para garantir a operação correta de templates que coletam métricas via monitoramento ODBC:

1. Certifique-se de que o driver ODBC necessário está instalado no Zabbix server ou proxy.
2. Vincule o template ao host de destino (se o template não estiver disponível na sua instalação do Zabbix, talvez seja necessário importar o template primeiro).
3. Se necessário, ajuste os valores dos macros do template.
4. Configure a instância monitorada para permitir o compartilhamento de dados com o Zabbix.

Uma descrição detalhada do template, incluindo a lista completa de macros, items e triggers, está disponível no arquivo README do template (acessível clicando no nome do template).

Os seguintes templates estão disponíveis:

- MSSQL por ODBC
- MySQL por ODBC
- Oracle por ODBC
- PostgreSQL por ODBC

7 Templates padronizados para dispositivos de rede

Visão geral

Para fornecer monitoramento para dispositivos de rede, como switches e roteadores, criamos dois chamados modelos: para o próprio dispositivo de rede (basicamente seu chassi) e para a interface de rede.

Modelos para muitas famílias de dispositivos de rede são fornecidos. Todos os modelos cobrem (quando possível obter esses itens do dispositivo):

- Monitoramento de falhas do chassi (fontes de alimentação, ventoinhas e temperatura, status geral)
- Monitoramento de desempenho do chassi (itens de CPU e memória)
- Coleta de inventário do chassi (números de série, nome do modelo, versão do firmware)
- Monitoramento de interface de rede com IF-MIB e EtherLike-MIB (status da interface, carga de tráfego da interface, status de duplex para Ethernet)

Se o template não estiver disponível em sua instalação do Zabbix, talvez seja necessário importar o template primeiro.

Se você estiver importando os novos templates prontos para uso, talvez também queira atualizar a expressão regular global `@Network interfaces for discovery` para:

```
Result is FALSE: ^Software Loopback Interface
Result is FALSE: ^(In)?[lL]oop[bB]ack[0-9._]*$
Result is FALSE: ^NULL[0-9.]*$
Result is FALSE: ^[lL]o[0-9.]*$
Result is FALSE: ^[sS]ystem$
Result is FALSE: ^Nu[0-9.]*$
```

para filtrar loopbacks e interfaces nulas na maioria dos sistemas.

Dispositivos

Lista de famílias de dispositivos para as quais os templates estão disponíveis:

Nome do template	Fabricante	Família de dispositivos	Modelos conhecidos	SO	MIBs utilizadas	**Tags**
Alcatel Timetra TiMOS SNMP	Alcatel	Alcatel Timetra	ALCATEL SR 7750	TiMOS	TIMETRA-SYSTEM-MIB,TIMETRA-CHASSIS-MIB	Certified
Aruba CX 8300s by SNMP	HPE Aruba Networking	Aruba CX Switch Series	Aruba JL636A 8325, Aruba JL717A 8360	Aruba AOS-CX	ARUBAWIRED-FAN-MIB, ARUBAWIRED-POWERSUPPLY-MIB, ARUBAWIRED-SYSTEMINFO-MIB, ARUBAWIRED-TEMPSENSOR-MIB, OSPF-MIB	Certified
Brocade FC SNMP	Brocade	Brocade FC switches	Brocade 300 SAN Switch-	-	SW-MIB,ENTITY-MIB	Performance Fault
Brocade_Foundry Stackable SNMP	Brocade	Brocade ICX	Brocade ICX6610, Brocade ICX7250-48, Brocade ICX7450-48F		FOUNDRY-SN-AGENT-MIB, FOUNDRY-SN-STACKING-MIB	Certified
Brocade_Foundry Nonstackable SNMP	Brocade, Foundry	Brocade MLX, Foundry	Brocade MLXe, Foundry FLS648, Foundry FWSX424		FOUNDRY-SN-AGENT-MIB	Performance Fault
Check Point Next Generation Firewall by SNMP	Check Point	Next Generation Firewall	-	Gaia	HOST-RESOURCES-MIB, CHECKPOINT-MIB, UCD-SNMP-MIB, SNMPv2-MIB, IF-MIB	Certified
Ciena 3906 by SNMP	Ciena	Ciena CPE	Ciena 3906	SAOS	WWP-LEOS-BLADE-MIB, WWP-LEOS-CHASSIS-MIB, WWP-LEOS-SYSTEM-CONFIG-MIB	Certified
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	Certified
Cisco IOS SNMP	Cisco	Cisco IOS ver > 12.2 3.5	Cisco C2950	IOS	CISCO-PROCESS-MIB,CISCO-MEMORY-POOL-MIB,CISCO-ENVMON-MIB	Certified
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	Certified

Nome do template	Fabricante	Família de dispositivos	Modelos conhecidos	SO	MIBs utilizadas	**Tags**
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	Certified
D-Link DES_DGS Switch SNMP	D-Link	DES/DGX switches	D-Link DES-xxxx/DGS-xxxx,DLINK DGS-3420-26SC	-	DLINK-AGENT-MIB,EQUIPMENT-MIB,ENTITY-MIB	Certified
D-Link DES 7200 SNMP	D-Link	DES-7xxx	D-Link DES 7206	-	ENTITY-MIB,MY-SYSTEM-MIB,MY-PROCESS-MIB,MY-MEMORY-MIB	Performance Fault Interfaces
Dell Force S-Series SNMP	Dell	Dell Force S-Series	S4810		F10-S-SERIES-CHASSIS-MIB	Certified
Extreme Exos SNMP	Extreme	Extreme EXOS	X670V-48x	EXOS	EXTREME-SYSTEM-MIB,EXTREME-SOFTWARE-MONITOR-MIB	Certified
FortiGate by SNMP	Fortinet	FortiGate (NGFW)	-	FortiOS	HOST-RESOURCES-MIB FORTINET-FORTIGATE-MIB FORTINET-CORE-MIB SNMPv2-MIB IF-MIB ENTITY-MIB	Performance Inventory
HP Comware HH3C SNMP	HP	HP (H3C) Comware	HP A5500-24G-4SFP HI Switch		HH3C-ENTITY-EXT-MIB,ENTITY-MIB	Certified
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	Certified
Huawei OceanStor 5300 V5 by SNMP	Huawei	Huawei OceanStor Dorado	Huawei OceanStor Dorado with V5 software		HUAWEI-STORAGE-HARDWARE-MIB, HUAWEI-STORAGE-SPACE-MIB, ISM-PERFORMANCE-MIB, HOST-RESOURCES-MIB, SNMPv2-MIB	Certified

Nome do template	Fabricante	Família de dispositivos	Modelos conhecidos	SO	MIBs utilizadas	**Tags**
Huawei OceanStor Dorado by SNMP	Huawei	Huawei OceanStor, Huawei OceanStor Dorado	Huawei OceanStor Dorado 3000/5000/6000/8000/18000/5300V5/5500/5600/6800/18500/18800 with V6/V700 software		HUAWEI-STORAGE-HARDWARE-MIB, HUAWEI-STORAGE-SPACE-MIB, ISM-PERFORMANCE-MIB, HOST-RESOURCES-MIB, SNMPv2-MIB	Certified
Huawei VRP by SNMP	Huawei	Huawei VRP	S2352P-EI		ENTITY-MIB,HUAWEI-ENTITY-EXTENT-MIB	Certified
Intel_Qlogic Infiniband SNMP	Intel/QLogic	Intel/QLogic Infiniband devices	Infiniband 12300		ICS-CHASSIS-MIB	Fault Inventory
Juniper SNMP	Juniper	MX,SRX,EX models	Juniper MX240, Juniper EX4200-24F	JunOS	JUNIPER-MIB	Certified
Juniper MX SNMP	Juniper	MX models	Juniper MX204 Edge Router	JunOS 24.2R1-S1.10	OSPF-MIB, JUNIPER-DOM-MIB, JUNIPER-MIB, BGP4-V2-MIB-JUNIPER, OSPFV3-MIB-JUNIPER	Certified
Mellanox SNMP	Mellanox	Mellanox Infiniband devices	SX1036	MLNX-OS	HOST-RESOURCES-MIB,ENTITY-MIB,ENTITY-SENSOR-MIB,MELLANOX-MIB	Certified

Nome do template	Fabricante	Família de dispositivos	Modelos conhecidos	SO	MIBs utilizadas	**Tags**
MikroTik CCR<device model> SNMP	MikroTik	MikroTik Cloud Core Routers (CCR series)	Templates dedicados separados estão disponíveis para 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	Certified

Nome do template	Fabricante	Família de dispositivos	Modelos conhecidos	SO	MIBs utilizadas	**Tags**
MikroTik CRS<device model> SNMP	MikroTik	MikroTik Cloud Router Switches (CRS series)	Templates dedicados separados estão disponíveis para 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	Certified
MikroTik CSS<device model> SNMP	MikroTik	MikroTik Cloud Smart Switches (CSS series)	Templates dedicados separados estão disponíveis para MikroTik CSS326-24G-2S+RM, MikroTik CSS610-8G-2S+IN	RouterOS	MIKROTIK-MIB,HOST-RESOURCES-MIB	Certified
MikroTik FiberBox SNMP	MikroTik	MikroTik FiberBox	MikroTik FiberBox	RouterOS	MIKROTIK-MIB,HOST-RESOURCES-MIB	Certified
MikroTik hEX <device model> SNMP	MikroTik	MikroTik hEX	Templates dedicados separados estão disponíveis para MikroTik hEX, MikroTik hEX lite, MikroTik hEX PoE, MikroTik hEX PoE lite, MikroTik hEX S	RouterOS	MIKROTIK-MIB,HOST-RESOURCES-MIB	Certified

Nome do template	Fabricante	Família de dispositivos	Modelos conhecidos	SO	MIBs utilizadas	**Tags**
MikroTik netPower <device model> SNMP	MikroTik	MikroTik netPower	Templates dedicados separados estão disponíveis para MikroTik netPower 15FR, MikroTik netPower 16P SNMP, MikroTik netPower Lite 7R	RouterOS/SwitchOS, SwitchOS Lite	MIKROTIK-MIB,HOST-RESOURCES-MIB	Certified
MikroTik PowerBox <device model> SNMP	MikroTik	MikroTik PowerBox	Templates dedicados separados estão disponíveis para MikroTik PowerBox, MikroTik PowerBox Pro	RouterOS	MIKROTIK-MIB,HOST-RESOURCES-MIB	Certified
MikroTik RB<device model> SNMP	MikroTik	Roteadores MikroTik RB series	Templates dedicados separados estão disponíveis para 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	Certified
MikroTik SNMP	MikroTik	Dispositivos MikroTik RouterOS	MikroTik CCR1016-12G, MikroTik RB2011UAS-2HnD, MikroTik 912UAG-5HPnD, MikroTik 941-2nD, MikroTik 951G-2HnD, MikroTik 1100AHx2	RouterOS	MIKROTIK-MIB,HOST-RESOURCES-MIB	Certified

Nome do template	Fabricante	Família de dispositivos	Modelos conhecidos	SO	MIBs utilizadas	**Tags**
Netgear Fastpath SNMP	Netgear	Netgear Fastpath	M5300-28G		FASTPATH-SWITCHING-MIB,FASTPATH-BOXSERVICES-PRIVATE-MIB	Fault Inventory
QTech QSW SNMP	QTech	Dispositivos Qtech	Qtech QSW-2800-28T	-	QTECH-MIB,ENTITY-MIB	Performance Inventory
Stormshield SNS by SNMP	Stormshield	Firewalls Stormshield Network Security (SNS)	SN3100	Stormshield SNS	HOST-RESOURCES-MIB, UCD-SNMP-MIB, STORMSHIELD-ASQ-STATS-MIB, STORMSHIELD-AUTOUPDATE-MIB, STORMSHIELD-HA-MIB, STORMSHIELD-PROPERTY-MIB, STORMSHIELD-HEALTH-MONITOR-MIB, STORMSHIELD-IF-MIB, STORMSHIELD-SYSTEM-MONITOR-MIB, STORMSHIELD-IPSEC-STATS-MIB	Certified
TP-LINK SNMP	TP-LINK	TP-LINK	T2600G-28TS v2.0		TPLINK-SYSMONITOR-MIB,TPLINK-SYSINFO-MIB	Performance Inventory
Ubiquiti AirOS SNMP	Ubiquiti	Dispositivos wireless Ubiquiti AirOS	NanoBridge,NanoStation,Unifi	AirOS	FROGFOOT-RESOURCES-MIB,IEEE802dot11-MIB	Performance
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	Performance Inventory

Design do template

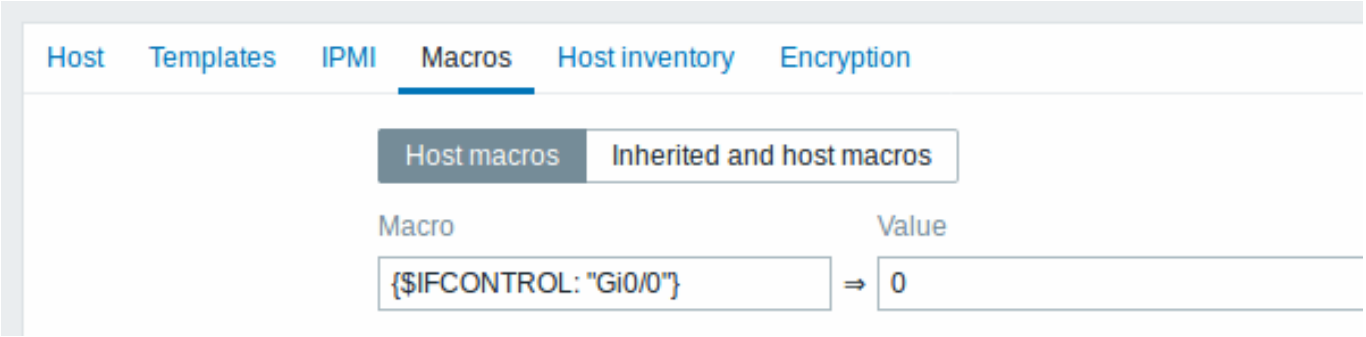
Os templates foram projetados com o seguinte em mente:

- Macros de usuário são usadas tanto quanto possível para que os triggers possam ser ajustados pelo usuário;
- Descoberta de baixo nível é usada tanto quanto possível para minimizar o número de items não suportados;
- Todos os templates dependem do Template ICMP Ping, então todos os dispositivos também são verificados por ICMP;
- Os items não usam nenhum MIB - OIDs SNMP são usados nos items e nas descobertas de baixo nível. Portanto, não é necessário carregar nenhum MIB no Zabbix para que os templates funcionem;
- Interfaces de rede loopback são filtradas durante a descoberta, assim como interfaces com ifAdminStatus = down(2)
- Contadores de 64 bits são usados de IF-MIB::ifXTable sempre que possível. Se não for suportado, contadores padrão de 32 bits são usados em vez disso.

Todas as interfaces de rede descobertas possuem um trigger que monitora seu status operacional (link), por exemplo:

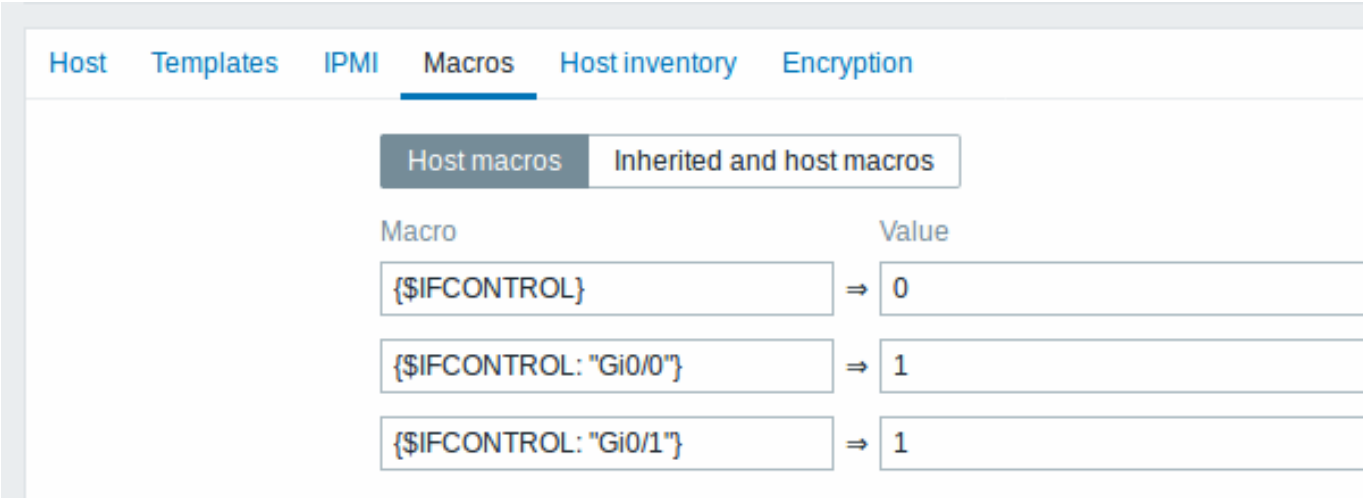
```
{$IFCONTROL:"{#IFNAME}"}=1 and last(/Alcatel Timetra TiMOS SNMP/net.if.status[ifOperStatus.{#SNMPINDEX}])
```

- Se você não quiser monitorar esta condição para uma interface específica, crie uma macro de usuário com contexto com o valor 0. Por exemplo:



onde Gi0/0 é {#IFNAME}. Dessa forma, o trigger não é mais usado para esta interface específica.

- Você também pode alterar o comportamento padrão para que todos os triggers não sejam disparados e ativar este trigger apenas para um número limitado de interfaces, como uplinks:



Tag

- Desempenho - os MIBs da família de dispositivos fornecem uma maneira de monitorar a CPU e itens de memória;
- Falha - os MIBs da família de dispositivos fornecem uma maneira de monitorar pelo menos um sensor de temperatura;
- Inventário – os MIBs da família de dispositivos fornecem uma maneira de coletar pelo menos o número de série do dispositivo e nome do modelo;
- Certificado – todas as três categorias principais acima são cobertas.

8 Operação de template VMware

Visão geral

O Zabbix inclui uma variedade de templates prontos para uso para monitoramento de máquinas virtuais, projetados para os hipervisores VMware vCenter e ESXi. Os templates disponíveis estão contidos em dois conjuntos de templates separados:

- VMware - usa dados de UUID para as macros correspondentes
- VMware FQDN - usa dados de FQDN para as macros correspondentes

Os templates projetados para monitorar hipervisores VMware vCenter ou ESXi contêm regras de descoberta de baixo nível pré-configuradas juntamente com várias verificações integradas para monitorar instalações virtuais.

Note:
Para o funcionamento correto do template *VMware FQDN*, cada máquina virtual monitorada deve ter um nome de SO exclusivo que siga as regras de FQDN. Além disso, o VMware Tools/Open Virtual Machine tools deve ser instalado em cada máquina. Se esses pré-requisitos forem atendidos, recomenda-se o uso do template *VMware FQDN*. O template *VMware FQDN* está disponível desde o Zabbix 5.2 com a introdução da capacidade de criar hosts com interfaces personalizadas.

 Um template clássico *VMware* também está disponível e pode ser usado se os requisitos de FQDN não forem atendidos. No entanto, o template *VMware* possui um problema conhecido. Hosts para máquinas virtuais descobertas são criados com nomes que são salvos no vCenter (por exemplo, ”VM1”, ”VM2”, etc.). Se o agent Zabbix estiver instalado nesses hosts e a autoregistration ativa do agent Zabbix estiver habilitada, o processo de autoregistration lerá os nomes dos hosts conforme foram registrados durante o lançamento (por exemplo, ”vm1.example.com”, ”vm2.example.com”, etc.). Isso pode levar à criação de novos hosts para máquinas virtuais existentes (já que nenhuma correspondência de nome foi encontrada), resultando em hosts duplicados com nomes diferentes.

Observe:

- O template *VMware Hypervisor* pode ser vinculado manualmente a um host, bem como usado na descoberta;
- O template *VMware Guest* não deve ser vinculado manualmente a um host e só pode ser usado na descoberta.

Configuração de macro de host

Para usar verificações simples do VMware, o host deve ter as seguintes macros de usuário definidas:

- {$VMWARE.URL} - URL do SDK do serviço VMware (vCenter ou hipervisor ESXi) (https://servername/sdk)
- {$VMWARE.USERNAME} - Nome de usuário do serviço VMware
- {$VMWARE.PASSWORD} - Senha do usuário {$VMWARE.USERNAME} do serviço VMware

10 Notificações em eventos

Visão geral

Supondo que tenhamos configurado alguns items e triggers e agora estamos recebendo alguns eventos ocorrendo como resultado das triggers mudando de estado, é hora de considerar algumas ações.

Para começar, não gostaríamos de ficar olhando para a lista de triggers ou eventos o tempo todo. Seria muito melhor receber uma notificação se algo significativo (como um problema) aconteceu. Além disso, quando problemas ocorrem, gostaríamos de garantir que todas as pessoas envolvidas sejam informadas.

É por isso que o envio de notificações é uma das principais ações oferecidas pelo Zabbix. Quem e quando deve ser notificado em determinado evento pode ser definido.

Para poder enviar e receber notificações do Zabbix, você deve:

- definir algum meio de comunicação
- configurar uma ação que envie uma mensagem para um dos meios definidos

As ações consistem em *condições* e *operações*. Basicamente, quando as condições são atendidas, as operações são executadas. As duas principais operações são o envio de uma mensagem (notificação) e a execução de um comando remoto.

Para eventos criados por descoberta e autorregistro, algumas operações adicionais estão disponíveis. Isso inclui adicionar ou remover um host, vincular um template, etc.

1 Tipos de mídia

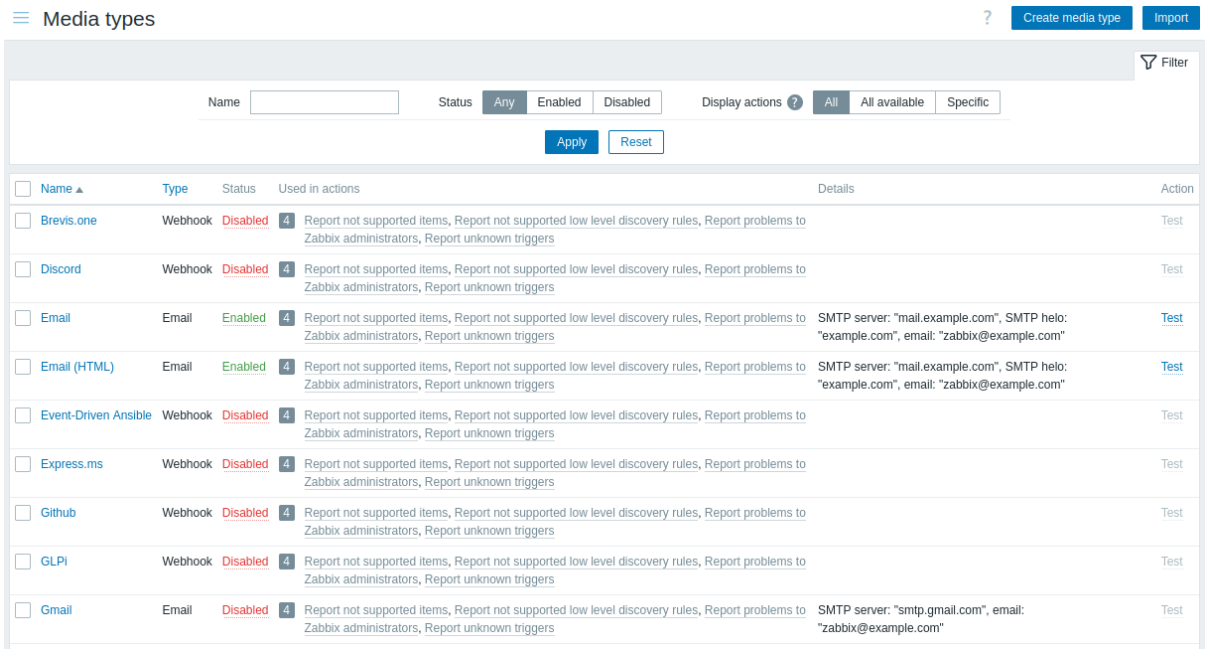
Visão geral

Os tipos de mídia são os canais de entrega usados para enviar notificações e alertas do Zabbix.

Os tipos de mídia suportam os seguintes métodos de entrega:

- E-mail
- SMS
- Script personalizado
- Webhook

Os tipos de mídia são mantidos em *Alertas > Tipos de mídia*. Alguns tipos de mídia vêm pré-definidos no conjunto de dados padrão. Você só precisa ajustar seus parâmetros para que funcionem.



Para ver como os tipos de mídia se encaixam no processo de alerta, vejamos os três requisitos para entregar notificações do Zabbix aos usuários finais:

1. Uma operação de ação deve ser definida para enviar notificações
2. Um tipo de mídia funcional deve ser definido (como *E-mail* que envia alertas usando SMTP)
3. Os detalhes de entrega no nível do usuário (como endereços de e-mail, números de telefone, etc.) devem ser definidos em mídia do usuário

Teste do tipo de mídia

Para testar se um tipo de mídia configurado funciona, clique em *Testar* na lista de tipos de mídia.

A solicitação de teste será enviada ao Zabbix server. O Zabbix server tentará enviar um alerta usando o tipo de mídia especificado e retornará o resultado para o frontend. O frontend aguardará o server retornar os resultados. O teste do tipo de mídia tem um tempo limite padrão de 65 segundos (configurável em *Administração* > *Geral* > *Timeouts*).

Veja também detalhes de teste para:

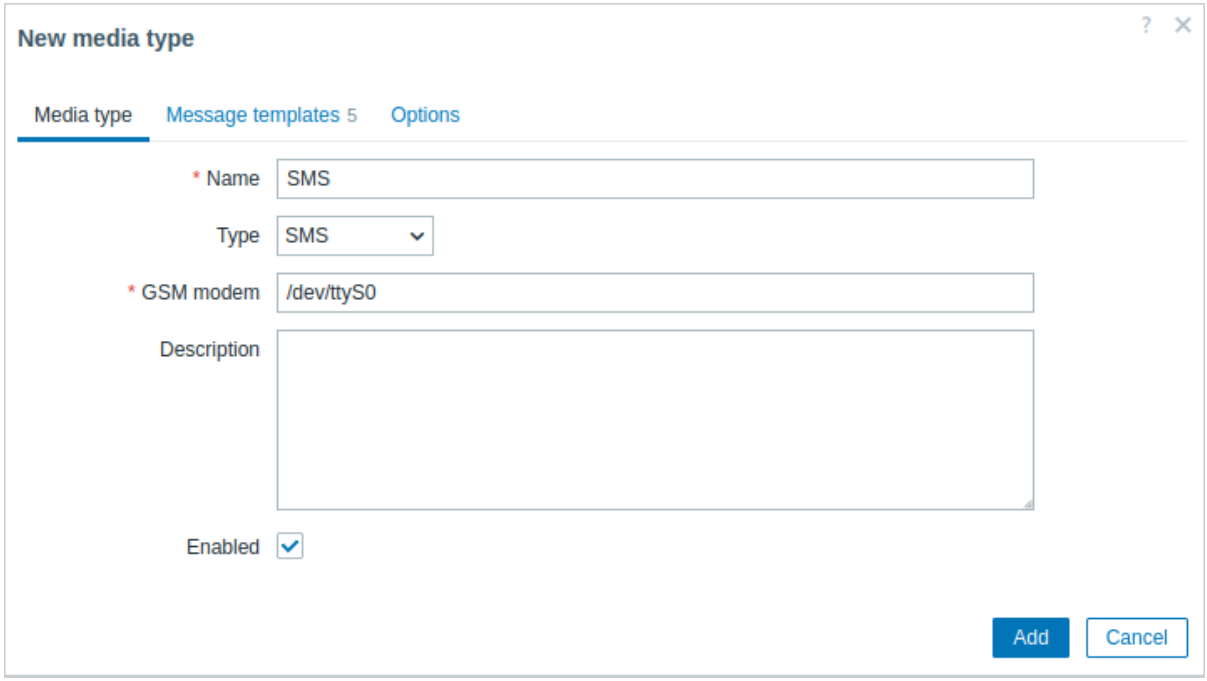
- E-mail
- Webhook
- Script

Configuração

Para criar um tipo de mídia no frontend do Zabbix:

- Vá para: *Alertas* > *Tipos de mídia*
- Clique em *Criar tipo de mídia*
- Insira os parâmetros do tipo de mídia no formulário

Alguns parâmetros são **comuns** para todos os métodos de entrega.

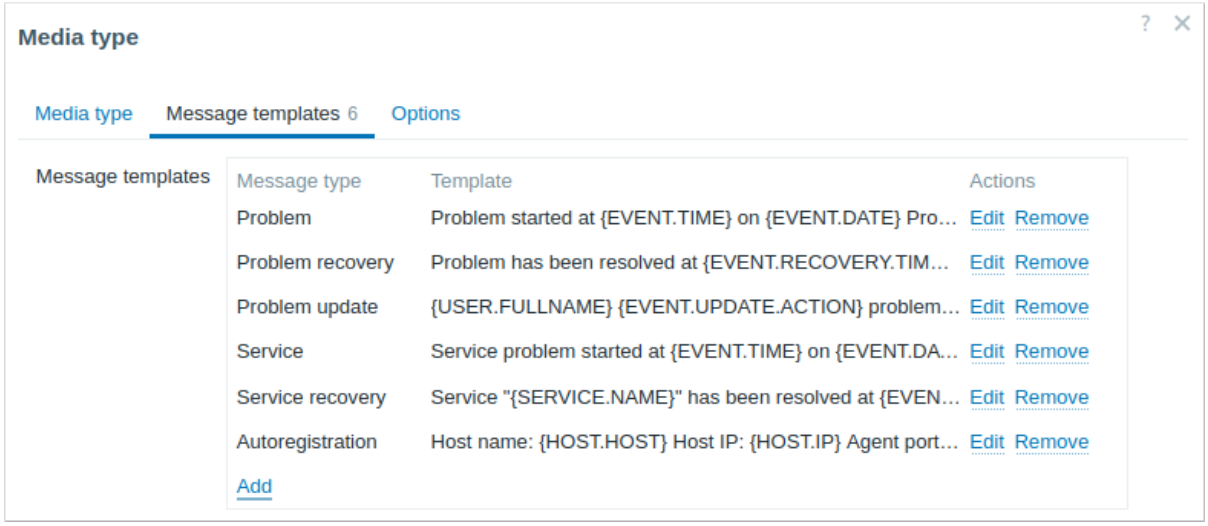


Parâmetro	Descrição
Nome	Nome do tipo de mídia.
Tipo	Selecione o método de entrega para o tipo de mídia.
Descrição	Insira uma descrição para o tipo de mídia.
Ativado	Marque a caixa de seleção para ativar o tipo de mídia.

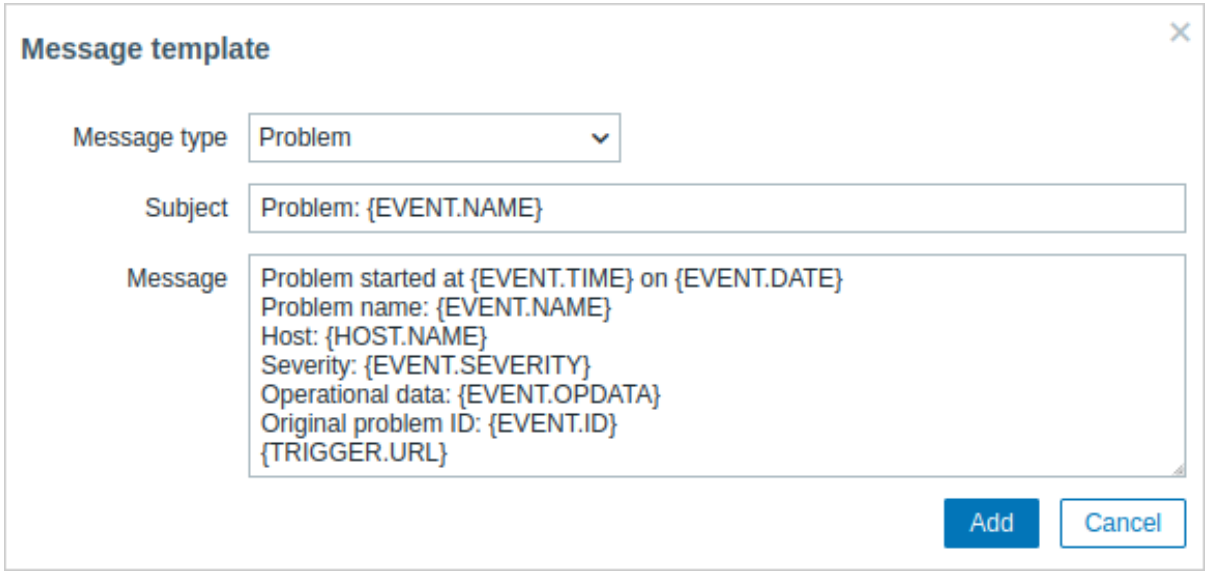
Para parâmetros específicos do método, consulte as páginas de e-mail, SMS, alertscript personalizado ou webhook.

Modelos de mensagem

A aba **Modelos de mensagem** contém mensagens padrão por tipo de evento (problema, recuperação de problema, descoberta, etc).



Clique em **Adicionar** para definir uma mensagem padrão (ou **Editar** para atualizar uma mensagem existente):



Parâmetro	Descrição
Tipo de mensagem	Tipo de evento para o qual a mensagem padrão deve ser usada. Apenas uma mensagem padrão pode ser definida para cada tipo de evento.
Assunto	Assunto da mensagem padrão. O assunto pode conter macros. É limitado a 255 caracteres. O assunto não está disponível para o tipo de mídia SMS.
Mensagem	A mensagem padrão. É limitada a uma certa quantidade de caracteres dependendo do tipo de banco de dados (veja Enviando mensagens para mais informações). A mensagem pode conter macros suportadas. Em mensagens de problema e atualização de problema, macros de expressão são suportadas (por exemplo, `{?avg(/host/key,1h)}`).

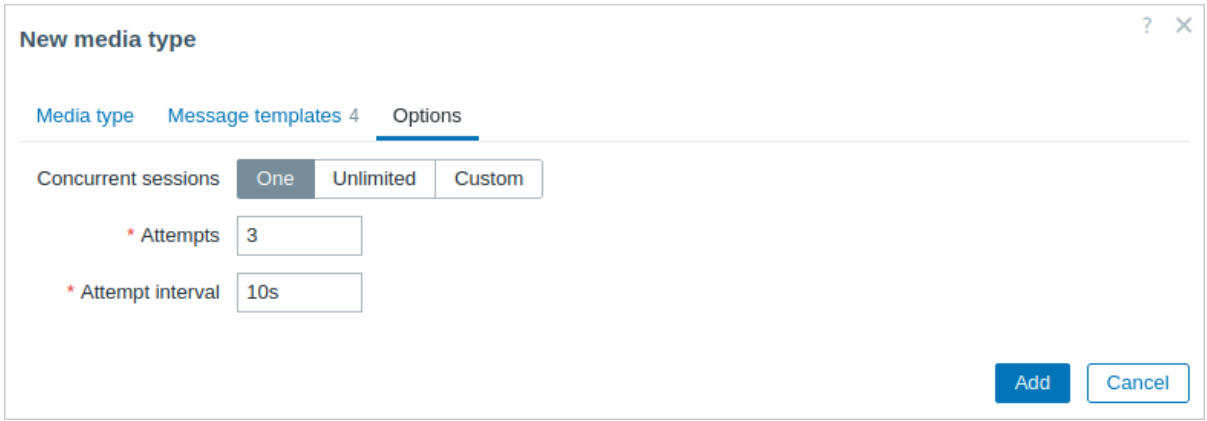
Observe que as mensagens padrão são substituídas por mensagens personalizadas, se definidas nas operações de ação.

Warning:
Definir modelos de mensagem é obrigatório para todos os métodos de entrega, incluindo webhooks ou scripts de alerta personalizados que não usam mensagens padrão para notificações. Por exemplo, a ação "Enviar mensagem para o webhook Pushover" falhará ao enviar notificações de problema, se a mensagem de problema para o webhook Pushover não estiver definida.

Opções

A aba **Opções** contém as configurações de processamento de alertas. O mesmo conjunto de opções é configurável para cada tipo de mídia.

Todos os tipos de mídia são processados em paralelo. Enquanto o número máximo de sessões simultâneas é configurável por tipo de mídia, o número total de processos de alerta no server só pode ser limitado pelo parâmetro StartAlerters parâmetro. Alertas gerados por um trigger são processados sequencialmente. Portanto, várias notificações podem ser processadas simultaneamente apenas se forem geradas por vários triggers.



Parâmetro	Descrição
Sessões simultâneas	Selecione o número de sessões de alerta paralelas para o tipo de mídia: **Uma** - uma sessão **Ilimitado** - número ilimitado de sessões **Personalizado** - selecione um número personalizado de sessões Valores ilimitados/altos significam mais sessões paralelas e maior capacidade para envio de notificações. Valores ilimitados/altos devem ser usados em ambientes grandes onde muitas notificações podem precisar ser enviadas simultaneamente. Se mais notificações precisarem ser enviadas do que o número de sessões simultâneas, as notificações restantes serão enfileiradas; elas não serão perdidas.
Tentativas	Número de tentativas para tentar enviar uma notificação. Até 100 tentativas podem ser especificadas; o valor padrão é '3'. Se '1' for especificado, o Zabbix enviará a notificação apenas uma vez e não tentará novamente se o envio falhar.
Intervalo de tentativas	Frequência de tentativas de reenvio de uma notificação caso o envio falhe, em segundos (0-3600). Se '0' for especificado, o Zabbix tentará novamente imediatamente. Sufixos de tempo são suportados, por exemplo, 5s, 3m, 1h.

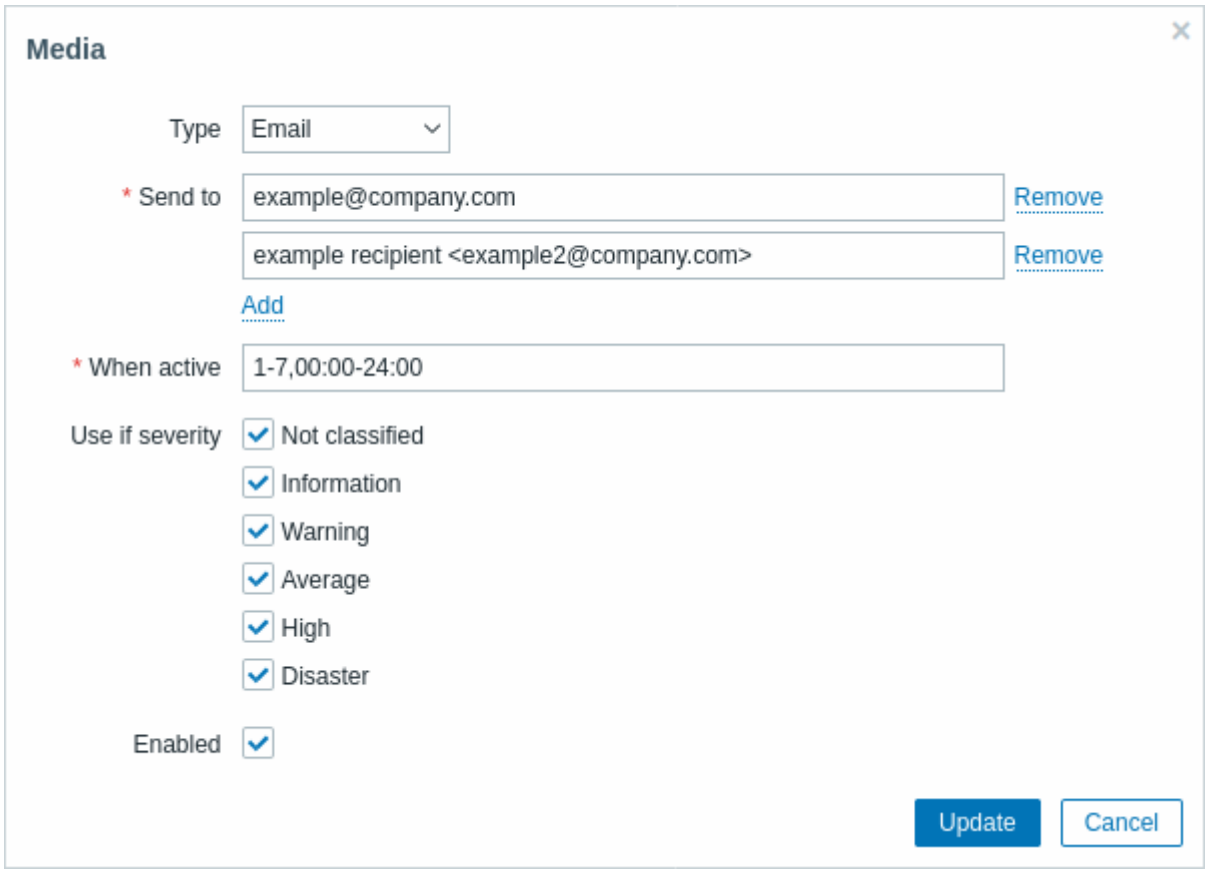
Mídia do usuário

Enquanto os tipos de mídia definem **como** uma notificação será enviada, as mídias do usuário definem **para onde** a notificação deve ser enviada.

A mídia do usuário (por exemplo, endereço de e-mail, ID do usuário do webhook, etc.) deve ser definida no perfil do usuário, independentemente do método de entrega. Uma ação que envia mensagens para o usuário *Admin* usando o webhook *X* falhará na entrega se os detalhes de entrega do webhook *X* não estiverem definidos no perfil do usuário Admin.

Para definir a mídia do usuário:

- Vá em *Usuários* > *Usuários* e abra o formulário de propriedades do usuário (ou vá em *Configurações do usuário* > *Notificações* do seu próprio perfil de usuário)
- Clique em *Adicionar* na aba Mídia

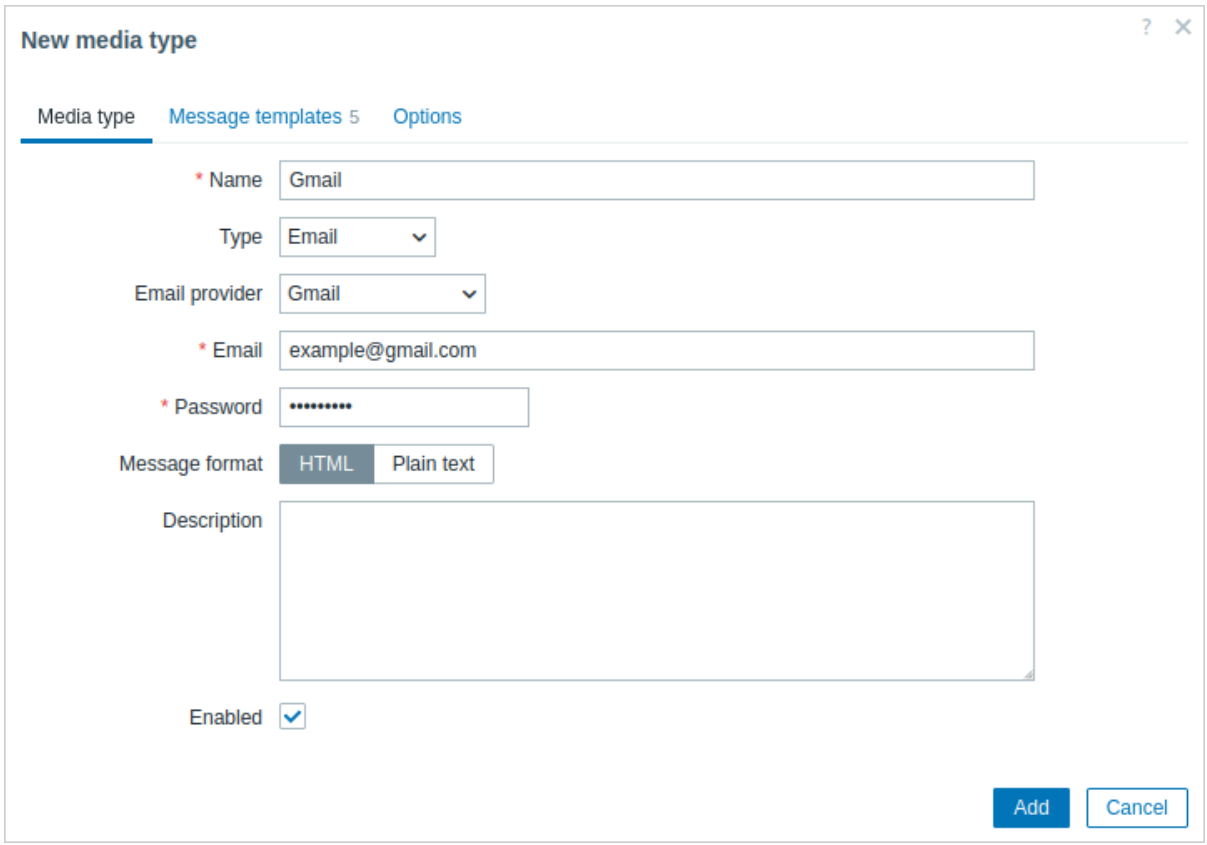


Parâmetro	Descrição
Tipo	A lista suspensa contém os nomes dos tipos de mídia habilitados. Observe que, ao editar uma mídia de um tipo de mídia desabilitado, o tipo será exibido em vermelho.
Enviar para	Insira as informações de contato para onde as mensagens devem ser enviadas. Para o tipo de mídia e-mail, vários endereços podem ser adicionados clicando no botão Add abaixo do campo de endereço. Neste caso, as notificações serão enviadas para todos os endereços listados. Para exemplos de endereços, consulte a descrição do parâmetro *E-mail* para o tipo de mídia e-mail.
Quando ativo	Você pode limitar o horário em que as mensagens são enviadas, por exemplo, definir apenas os dias úteis (1-5,09:00-18:00). Observe que esse limite é baseado no fuso horário do usuário. Se o fuso horário do usuário for alterado e for diferente do fuso horário do sistema, esse limite pode precisar ser ajustado para não perder mensagens importantes. Consulte a página Especificação do período de tempo para descrição do formato. Macros de usuário são suportadas.
Usar se gravidade	Marque as caixas de seleção das gravidades de trigger para as quais você deseja receber notificações. *Nota* que a gravidade padrão ('Não classificado') **deve ser** marcada se você quiser receber notificações para eventos que não sejam de trigger. Após salvar, as gravidades de trigger selecionadas serão exibidas nas cores correspondentes de gravidade, enquanto as não selecionadas ficarão esmaecidas.
Status	Status da mídia do usuário. **Habilitado** - está em uso. **Desabilitado** - não está sendo usado.

Tipos de mídia automatizados do Gmail/Office365

Usuários do Gmail ou Office365 podem se beneficiar de uma configuração mais fácil do tipo de mídia. O campo *Provedor de e-mail* na configuração do tipo de mídia de e-mail permite selecionar opções pré-configuradas para Gmail e Office 365.

Ao selecionar as opções relacionadas ao Gmail/Office365, é necessário apenas fornecer o endereço de e-mail/senha do remetente para criar um tipo de mídia funcional.



Assim que o endereço de e-mail/senha for fornecido, o Zabbix será capaz de preencher automaticamente todas as configurações necessárias para os tipos de mídia do Gmail/Office365 com os valores reais/recomendados, ou seja, *Servidor SMTP*, *Porta do*

servidor SMTP, *SMTP helo* e *Segurança da conexão*. Devido a essa automação, esses campos nem são exibidos, no entanto, é possível ver os detalhes do servidor SMTP e do e-mail na lista de tipos de mídia (veja a coluna *Detalhes*).

Observe também que:

- A senha não é necessária para as opções de relay.
- Para o relay do Office365, o nome de domínio do endereço de e-mail fornecido será usado para preencher dinamicamente o servidor SMTP (ou seja, substituir "example.com" em `example-com.mail.protection.outlook.com` pelo valor real).

1 Email

Visão geral

Para configurar o e-mail como canal de entrega de mensagens, é necessário configurar o e-mail como tipo de mídia e atribuir endereços específicos aos usuários.

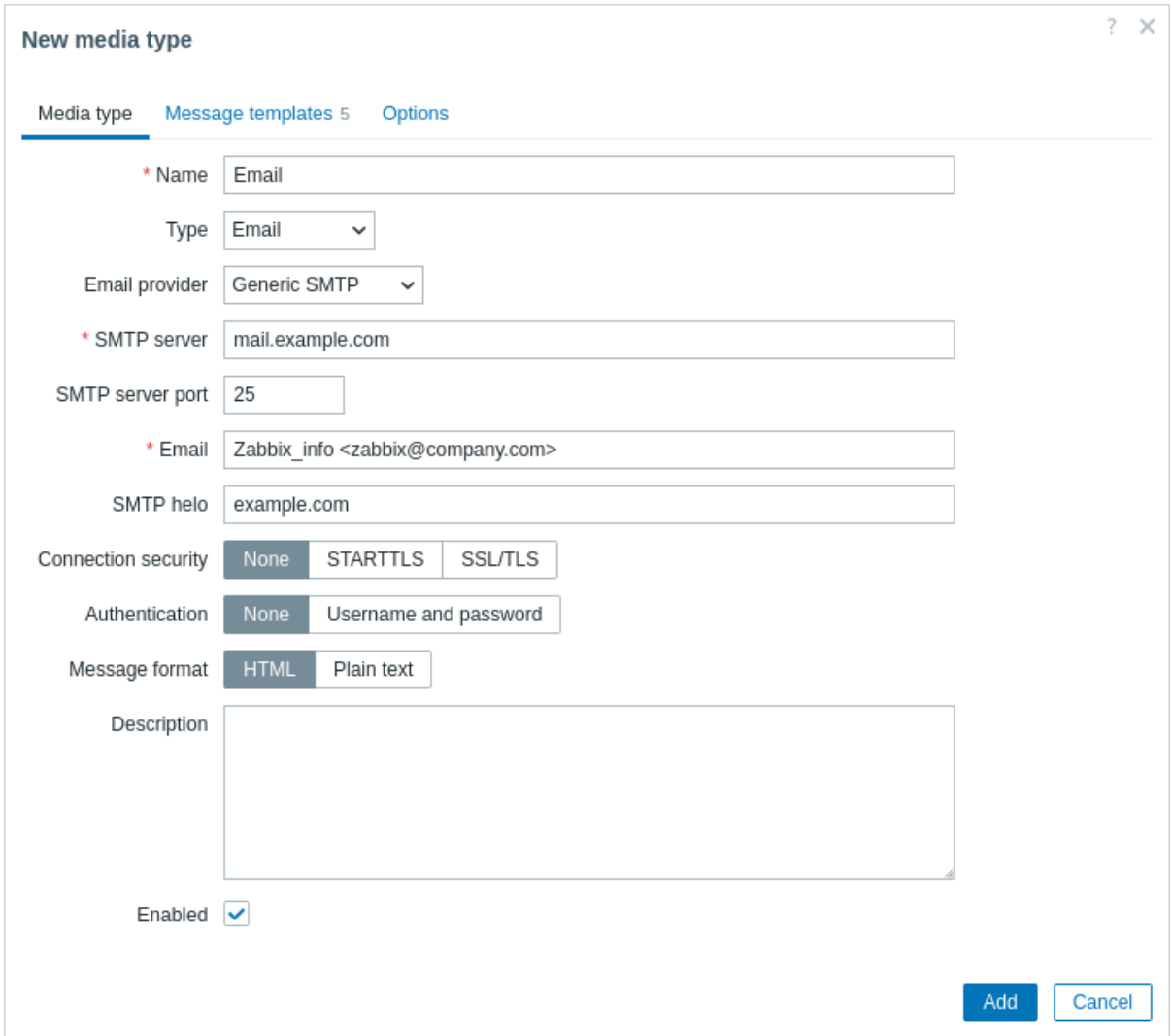
> **Note:**
> Múltiplas notificações para um único evento serão agrupadas no mesmo tópico de e-mail.

Configuração

Para configurar e-mail como tipo de mídia:

1. Vá para *Alertas* > *Tipos de mídia*.
2. Clique em *Criar tipo de mídia* (ou clique em *E-mail* na lista de tipos de mídia predefinidos).

A aba **Tipo de mídia** contém atributos gerais do tipo de mídia:



Todos os campos obrigatórios estão marcados com um asterisco vermelho.

Warning:
A senha é exportada em texto claro ao exportar a configuração do tipo de mídia de e-mail.

Os seguintes parâmetros são específicos para o tipo de mídia de e-mail:

Parâmetro	Descrição
Provedor de e-mail	Selecione o provedor de e-mail: *SMTP genérico*, *Gmail*, *Gmail relay*, *Office365* ou *Office365 relay*. Se você selecionar as opções relacionadas ao Gmail/Office365, será necessário apenas fornecer o endereço de e-mail do remetente e a senha; opções como *Servidor SMTP*, *Porta do servidor SMTP*, *SMTP helo* e *Segurança da conexão* serão preenchidas automaticamente pelo Zabbix. Veja também: Tipos de mídia automatizados do Gmail/Office365.
Servidor SMTP	Defina um servidor SMTP para lidar com mensagens de saída. Este campo está disponível se *SMTP genérico* for selecionado como provedor de e-mail.
Porta do servidor SMTP	Defina a porta do servidor SMTP para lidar com mensagens de saída. Este campo está disponível se *SMTP genérico* for selecionado como provedor de e-mail.
E-mail	O endereço inserido aqui será usado como o endereço **De** para as mensagens enviadas. Adicionar um nome de exibição do remetente (como "Zabbix_info" em *Zabbix_info <zabbix@company.com>* na captura de tela acima) com o endereço de e-mail real é suportado. Existem algumas restrições para nomes de exibição em e-mails do Zabbix em comparação com o que é permitido pelo RFC 5322, conforme ilustrado pelos exemplos: Exemplos válidos: *zabbix@company.com* (apenas endereço de e-mail, não é necessário usar colchetes angulares) *Zabbix_info <zabbix@company.com>* (nome de exibição e endereço de e-mail entre colchetes angulares) *∑Ω-monitoring <zabbix@company.com>* (caracteres UTF-8 no nome de exibição) Exemplos inválidos: *Zabbix HQ zabbix@company.com* (nome de exibição presente, mas sem colchetes angulares ao redor do endereço de e-mail) *"Zabbix\@\<H(comment)Q\>" <zabbix@company.com>* (embora válido pelo RFC 5322, pares entre aspas e comentários não são suportados em e-mails do Zabbix)
SMTP helo	Defina um valor SMTP helo correto, normalmente um nome de domínio. Se estiver vazio, o nome de domínio do e-mail será enviado (ou seja, o que vem após @ no campo *E-mail*). Se não for possível obter o nome de domínio, um aviso de nível de depuração será registrado e o nome do host do servidor será enviado como domínio para o comando HELO. Este campo está disponível se *SMTP genérico* for selecionado como provedor de e-mail.
Segurança da conexão	Selecione o nível de segurança da conexão: **Nenhum** - não usar a opção CURLOPT_USE_SSL **STARTTLS** - usar a opção CURLOPT_USE_SSL com valor CURLUSESSL_ALL **SSL/TLS** - o uso de CURLOPT_USE_SSL é opcional
Verificar par SSL	Marque a caixa de seleção para verificar o certificado SSL do servidor SMTP. O valor da diretiva de configuração do servidor "SSLCALocation" deve ser colocado em CURLOPT_CAPATH para validação do certificado. Isso define a opção cURL CURLOPT_SSL_VERIFYPEER.
Verificar host SSL	Marque a caixa de seleção para verificar se o campo *Common Name* ou o campo *Subject Alternate Name* do certificado do servidor SMTP corresponde. Isso define a opção cURL CURLOPT_SSL_VERIFYHOST.
Autenticação	Selecione o nível de autenticação: **Nenhum** - nenhuma opção cURL é definida **Nome de usuário e senha** - implica "AUTH=*" deixando a escolha do mecanismo de autenticação para o cURL
Nome de usuário	Nome de usuário a ser usado na autenticação. Isso define o valor de CURLOPT_USERNAME. Macros de usuário suportadas.
Senha	Senha a ser usada na autenticação. Isso define o valor de CURLOPT_PASSWORD. Macros de usuário suportadas. O valor inserido aqui é exportado como está ao exportar a configuração do tipo de mídia. Para evitar expor credenciais confidenciais em arquivos exportados, use uma macro de usuário (por exemplo, `{$EMAIL_SMTP_PASSWORD}`) em vez de uma senha literal — a referência da macro será exportada enquanto o valor secreto deve ser (re)configurado no sistema de destino.

Parâmetro	Descrição
Formato da mensagem	Selecione o formato da mensagem: **HTML** - enviar como HTML **Texto simples** - enviar como texto simples

Attention:
Para habilitar a autenticação SMTP, o servidor Zabbix deve ser compilado com a opção de compilação `--with-libcurl`, que requer o libcurl (veja os requisitos do servidor para detalhes da versão).

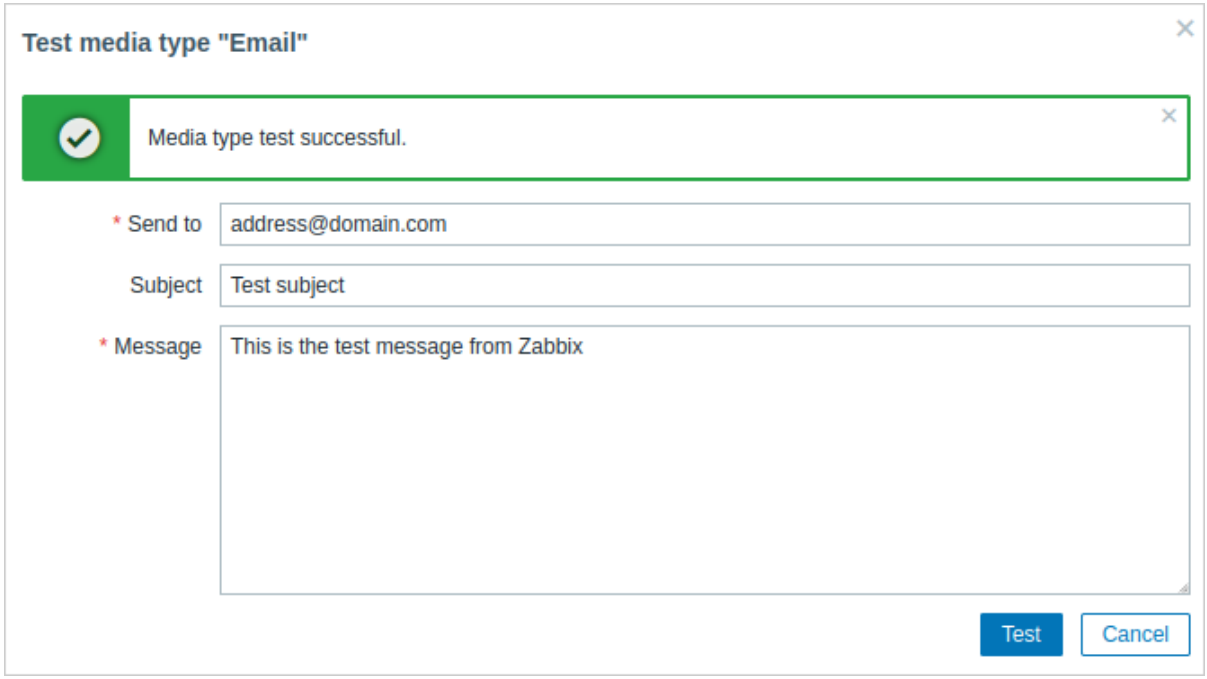
Veja também parâmetros comuns do tipo de mídia para detalhes sobre como configurar mensagens padrão e opções de processamento de alertas.

Testando

Para testar se um tipo de mídia de e-mail configurado funciona corretamente:

1. Localize o e-mail relevante na lista de tipos de mídia.
2. Clique em *Testar* na última coluna da lista (uma janela de teste será aberta).
3. Digite um endereço de destinatário em *Enviar para*, o corpo da mensagem e, opcionalmente, o assunto.
4. Clique em *Testar* para enviar uma mensagem de teste.

Uma mensagem de sucesso ou falha no teste será exibida na mesma janela:



Mídia do usuário

Depois que o tipo de mídia de e-mail for configurado, vá para a seção *Usuários* > *Usuários* e edite o perfil do usuário para atribuir a mídia de e-mail ao usuário. As etapas para configurar a mídia do usuário, comuns a todos os tipos de mídia, estão descritas na página Tipos de mídia.

2 SMS

Visão geral

O Zabbix suporta o envio de mensagens SMS usando um modem GSM serial conectado à porta serial do servidor Zabbix.

Certifique-se de que:

- A velocidade do dispositivo serial (normalmente /dev/ttyS0 no Linux) corresponde à do modem GSM. O Zabbix não define a velocidade do link serial. Ele usa as configurações padrão.
- O usuário 'zabbix' tem acesso de leitura/gravação ao dispositivo serial. Execute o comando ls –l /dev/ttyS0 para ver as permissões atuais do dispositivo serial.

- O modem GSM tem o PIN inserido e o preserva após o reset de energia. Alternativamente, você pode desabilitar o PIN no cartão SIM. O PIN pode ser inserido emitindo o comando AT+CPIN="NNNN" (NNNN é o seu número PIN, as aspas devem estar presentes) em um software de terminal, como o minicom do Unix ou o HyperTerminal do Windows.

O Zabbix foi testado com estes modems GSM:

- Siemens MC35
- Teltonika ModemCOM/G10

Para configurar o SMS como canal de entrega de mensagens, você também precisa configurar o SMS como tipo de mídia e inserir os respectivos números de telefone para os usuários.

Configuração

Para configurar SMS como o tipo de mídia:

- Vá para *Alertas* → *Tipos de mídia*
- Clique em *Criar tipo de mídia* (ou clique em *SMS* na lista de tipos de mídia pré-definidos).

Os seguintes parâmetros são específicos para o tipo de mídia SMS:

Parâmetro	Descrição
GSM modem	Defina o nome do dispositivo serial do modem GSM. O caminho inserido aqui será validado em relação ao parâmetro do servidor SMSDevices (se especificado).

Veja parâmetros comuns do tipo de mídia para detalhes sobre como configurar mensagens padrão e opções de processamento de alertas. Observe que o processamento paralelo do envio de notificações SMS não é possível.

Mídia do usuário

Depois que o tipo de mídia SMS estiver configurado, vá para a seção *Usuários* → *Usuários* e edite o perfil do usuário para atribuir a mídia SMS ao usuário. As etapas para configurar a mídia do usuário, comuns a todos os tipos de mídia, estão descritas na página Tipos de mídia.

3 Scripts de alerta personalizados

Visão geral

Se você não estiver satisfeito com os tipos de mídia existentes para envio de alertas, existe uma maneira alternativa de fazer isso. Você pode criar um script que irá lidar com a notificação do seu jeito.

Scripts de alerta personalizados são executados no Zabbix server. Esses scripts devem estar localizados no diretório especificado no parâmetro AlertScriptsPath do arquivo de configuração do server.

Aqui está um exemplo de script de alerta personalizado:

```
#####!/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:
O Zabbix verifica o código de saída dos comandos e scripts executados. Qualquer código de saída diferente de **0** é considerado um erro de execução de comando. Nesses casos, o Zabbix tentará repetir a execução que falhou.

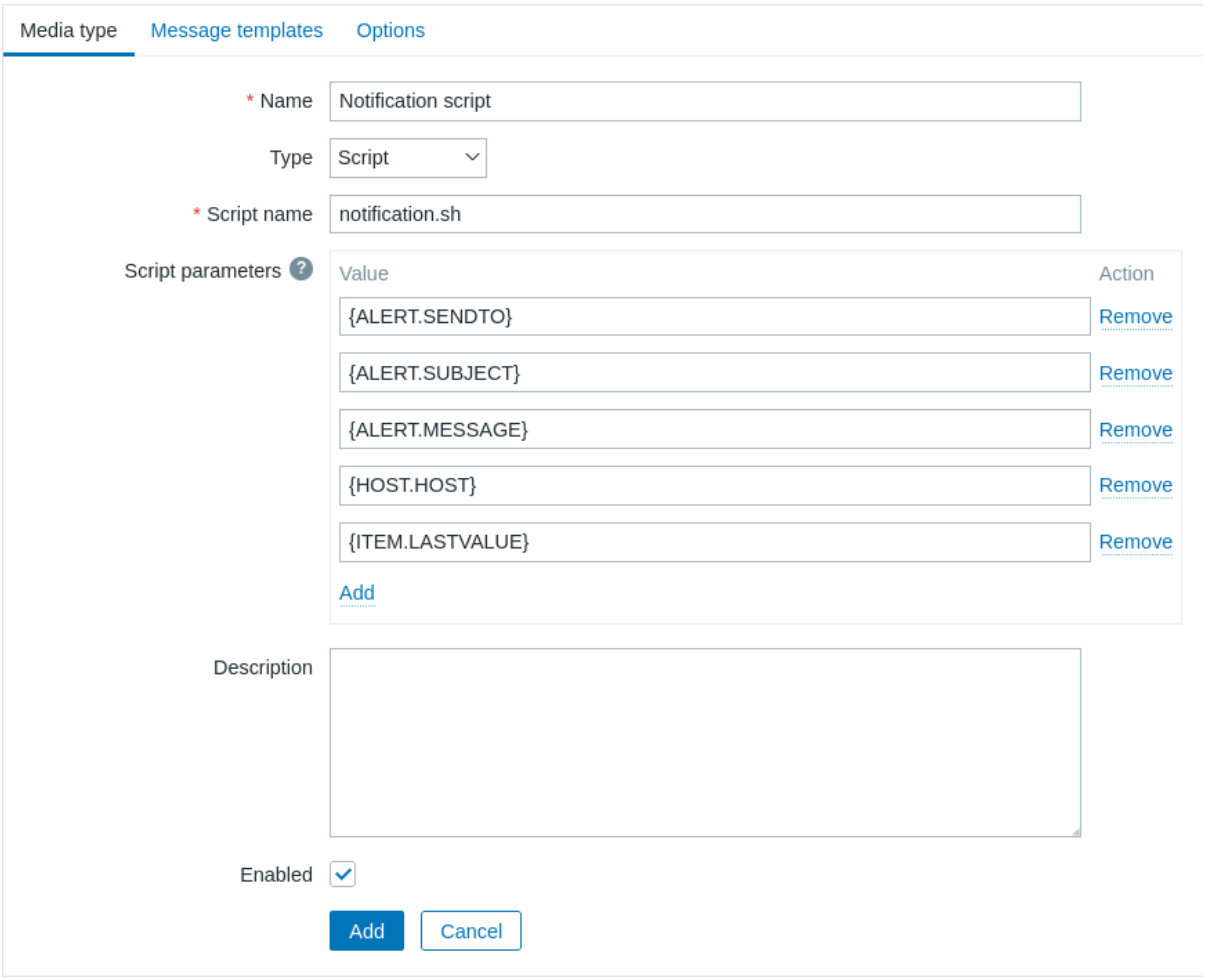
Variáveis de ambiente não são preservadas ou criadas para o script, portanto, devem ser tratadas explicitamente.

Configuração

Para configurar scripts de alerta personalizados como um tipo de mídia:

1. Vá para *Alertas* → *Tipos de mídia*.
2. Clique em *Criar tipo de mídia*.

A aba **Tipo de mídia** contém os atributos gerais do tipo de mídia:



Todos os campos obrigatórios estão marcados com um asterisco vermelho.

Os seguintes parâmetros são específicos para o tipo de mídia de script:

Parâmetro	Descrição
Nome do script	Insira o nome do arquivo de script (por exemplo, notification.sh) que está localizado no diretório especificado no parâmetro de configuração do server AlertScriptsPath.
Parâmetros do script	Adicione parâmetros de script opcionais que serão passados para o script como argumentos de linha de comando na ordem em que forem definidos. Os parâmetros do script suportam as macros {ALERT.SENDTO}, {ALERT.SUBJECT}, {ALERT.MESSAGE} e todas as macros que são suportadas em notificações, bem como macros de usuário.

Veja parâmetros comuns de tipo de mídia para detalhes sobre como configurar mensagens padrão e opções de processamento de alertas.

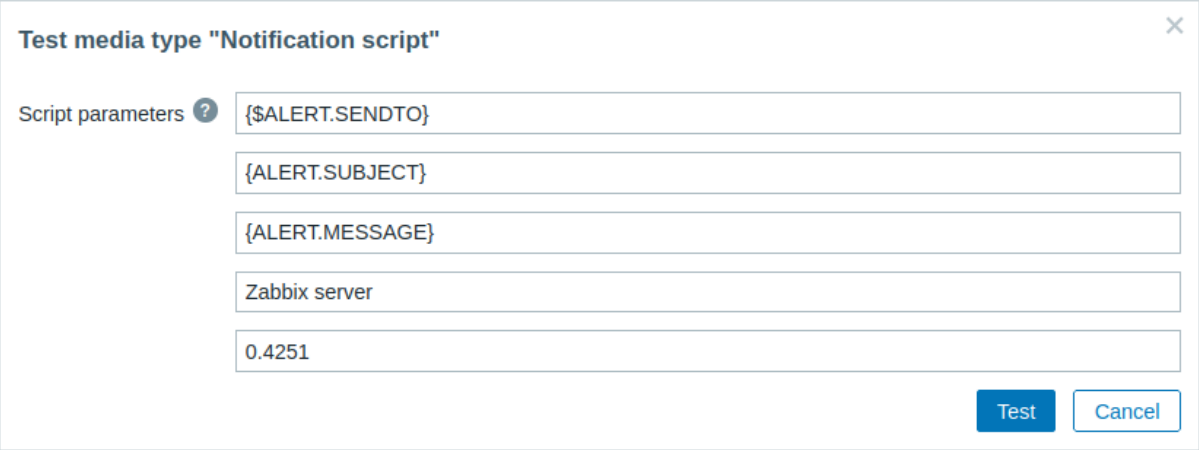
Warning:
Mesmo que um script de alerta não utilize mensagens padrão, os modelos de mensagem para os tipos de operação usados por este tipo de mídia ainda devem ser definidos. Caso contrário, uma notificação não será enviada.

Attention:
Se mais de um tipo de mídia de script for configurado, esses scripts podem ser processados em paralelo pelos processos de alerta. O número total de processos de alerta é limitado pelo parâmetro `StartAlerters` no arquivo de configuração do server.

Testando

Para testar um tipo de mídia de script configurado:

1. Localize o script relevante na lista de tipos de mídia.
2. Clique em *Testar* na última coluna da lista; um formulário de teste será aberto em uma janela pop-up. O formulário de teste conterá o mesmo número de parâmetros que estão configurados para o tipo de mídia de script.
3. Edite os valores dos parâmetros do script, se necessário. A edição afeta apenas o procedimento de teste; os valores reais não serão alterados.
4. Clique em *Testar*.



Note:
Ao testar um tipo de mídia de script configurado, as macros {ALERT.SENDTO}, {ALERT.SUBJECT}, {ALERT.MESSAGE} e macros de usuário serão resolvidas para seus valores, mas macros relacionadas a eventos (por exemplo, {HOST.HOST}, {ITEM.LASTVALUE}, etc.) não serão resolvidas, pois durante o teste não há evento relacionado para obter os detalhes. Observe que macros dentro das macros {ALERT.SUBJECT} e {ALERT.MESSAGE} também não serão resolvidas. Por exemplo, se o valor de {ALERT.SUBJECT} for composto por "Problem: {EVENT.NAME}", então a macro {EVENT.NAME} não será resolvida.

Mídia do usuário

Depois que o tipo de mídia for configurado, vá para a seção *Usuários* → *Usuários* e edite um perfil de usuário atribuindo esse tipo de mídia ao usuário. As etapas para configurar a mídia do usuário, comuns a todos os tipos de mídia, estão descritas na página Tipos de mídia.

Observe que, ao definir a mídia do usuário, o campo *Enviar para* não pode ficar vazio. Se este campo não for usado no script de alerta, insira qualquer combinação de caracteres suportados para contornar os requisitos de validação.

4 Webhook

Visão geral

O tipo de mídia webhook é útil para fazer chamadas HTTP usando código JavaScript personalizado para integração direta com software externo, como sistemas de helpdesk, chats ou mensageiros. Você pode optar por importar uma integração fornecida pelo Zabbix ou criar uma integração personalizada do zero.

Integrações

As seguintes integrações estão disponíveis, permitindo o uso de tipos de mídia webhook predefinidos para enviar notificações do Zabbix para:

- brevis.one
- Discord
- Event-Driven Ansible

- Express.ms messenger
- GitHub
- GLPI
- IBM Maximo Service Request
- iLert
- iTop
- Jira
- Jira Service Management
- ManageEngine ServiceDesk
- Mantis Bug Tracker
- Mattermost
- MS Teams
- MS Teams Workflows
- LINE
- Opsgenie
- OTRS CE
- PagerDuty
- Pushover
- Redmine
- Rocket.Chat
- ServiceNow
- SIGNL4
- Slack
- SolarWinds
- SysAid
- Telegram
- TOPdesk
- VictorOps
- Zammad
- Zendesk

Note:
Além dos serviços listados aqui, o Zabbix pode ser integrado ao **Spiceworks** (nenhum webhook é necessário). Para converter notificações do Zabbix em tickets do Spiceworks, crie um tipo de mídia de e-mail e insira o endereço de e-mail do helpdesk do Spiceworks (por exemplo, help@zabbix.on.spiceworks.com) nas configurações de perfil de um usuário Zabbix designado.

Configuração

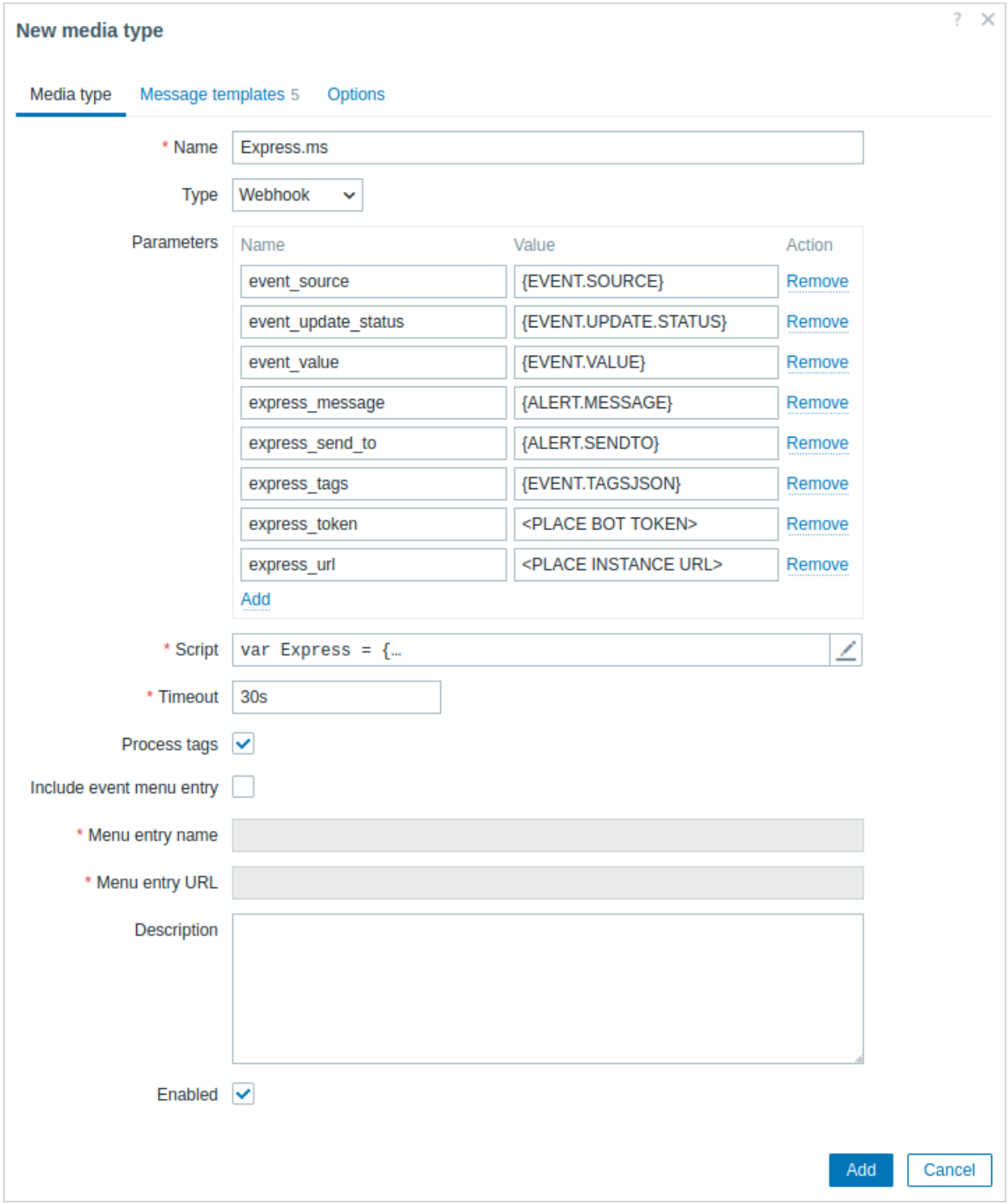
Para começar a usar uma integração de webhook:

1. Localize o arquivo .yaml necessário no diretório `templates/media` da versão do Zabbix baixada ou faça o download a partir do repositório git do Zabbix.
2. Importe o arquivo para sua instalação do Zabbix. O webhook aparecerá na lista de tipos de mídia.
3. Configure o webhook de acordo com as instruções no arquivo *Readme.md* (você pode clicar no nome do webhook acima para acessar rapidamente o *Readme.md*).

Para criar um webhook personalizado do zero:

1. Vá para *Alertas* > *Tipos de mídia*.
2. Clique em *Criar tipo de mídia*.

A aba **Tipo de mídia** contém vários atributos específicos para este tipo de mídia:



Todos os campos obrigatórios estão marcados com um asterisco vermelho.

Os seguintes parâmetros são específicos para o tipo de mídia webhook:

Parâmetro	Descrição
Parâmetros	Especifique as variáveis do webhook como pares de atributo e valor. Para webhooks pré-configurados, a lista de parâmetros varia dependendo do serviço. Verifique o arquivo *Readme.md* do webhook para a descrição dos parâmetros. Para novos webhooks, várias variáveis comuns são incluídas por padrão (URL:<vazio>, HTTPProxy:<vazio>, To:{ALERT.SENDTO}, Subject:{ALERT.SUBJECT}, Message:{ALERT.MESSAGE}), fique à vontade para mantê-las ou removê-las. Os parâmetros do webhook suportam macros de usuário, todas as macros que são suportadas em notificações de problemas e, adicionalmente, as macros {ALERT.SENDTO}, {ALERT.SUBJECT} e {ALERT.MESSAGE}. Se você especificar um proxy HTTP, o campo suporta a mesma funcionalidade que no campo de configuração do item HTTP proxy. A string do proxy pode ser prefixada com `[scheme]://` para especificar qual tipo de proxy é usado (por exemplo, https, socks4, socks5; veja a documentação).
Script	Insira o código JavaScript no editor modal que abre ao clicar no campo de parâmetro ou no ícone de lápis ao lado dele. Este código executará a operação do webhook. O script é um código de função que aceita pares de parâmetro - valor. Os valores devem ser convertidos em objetos JSON usando o método JSON.parse(), por exemplo: `var params = JSON.parse(value);`. O código tem acesso a todos os parâmetros, pode executar requisições HTTP GET, POST, PUT e DELETE, suportar métodos adicionais como CONNECT, PATCH, HEAD, OPTIONS e TRACE, e tem controle sobre os cabeçalhos HTTP e o corpo da requisição. O script deve conter um operador return, caso contrário não será válido. Ele pode retornar o status OK junto com uma lista opcional de tags e valores de tags (veja a opção *Processar tags*) ou uma string de erro. Observe que o script é executado apenas após um alerta ser criado. Se o script estiver configurado para retornar e processar tags, essas tags não serão resolvidas nas macros {EVENT.TAGS} e {EVENT.RECOVERY.TAGS} na mensagem inicial do problema e nas mensagens de recuperação porque o script ainda não teve tempo de ser executado. *Nota*: Usar variáveis locais (por exemplo, `var local = 1`) em vez de globais (por exemplo, `global = 1`) é recomendado para garantir que cada script opere em seus próprios dados e evitar colisões entre chamadas simultâneas (veja problemas conhecidos). Veja também: Diretrizes de desenvolvimento de webhook, Exemplos de scripts de webhook, Objetos JavaScript adicionais.
Timeout	Tempo limite de execução do JavaScript (1-60s, padrão 30s). Sufixos de tempo são suportados, por exemplo, 30s, 1m.
Processar tags	Marque a caixa de seleção para processar os valores das propriedades JSON retornadas como tags. Essas tags são adicionadas a quaisquer tags de problema existentes. Observe que ao usar tags de webhook, o webhook deve retornar um objeto JSON contendo pelo menos um objeto tags vazio: `var result = {tags: {}};` Exemplos de tags que podem ser retornadas: *jira-id:prod-1234*, *responsible:John Smith*, *processed:<sem valor>*
Incluir entrada no menu de eventos	Marque a caixa de seleção para incluir uma entrada no menu de eventos vinculando a um ticket externo criado. Uma entrada será incluída para cada webhook que estiver habilitado e tiver esta caixa marcada. Observe que se os parâmetros *Nome da entrada do menu* e *URL da entrada do menu* contiverem quaisquer macros {EVENT.TAGS.<tag name>}, uma entrada será incluída apenas se essas macros puderem ser resolvidas (ou seja, o evento tiver essas tags definidas). Se marcado, o webhook não deve ser usado para enviar notificações para diferentes usuários (considere criar um usuário dedicado em vez disso) e não deve ser usado em várias ações de alerta para um único evento de problema.
Nome da entrada do menu	Especifique o nome da entrada do menu. A macro {EVENT.TAGS.<tag name>} é suportada. Este campo só é obrigatório se *Incluir entrada no menu de eventos* estiver marcado.

Parâmetro	Descrição
URL da entrada do menu	Especifique a URL subjacente da entrada do menu. A macro {EVENT.TAGS.<tag name>} é suportada. Este campo só é obrigatório se *Incluir entrada no menu de eventos* estiver marcado.

Veja parâmetros comuns de tipo de mídia para detalhes sobre como configurar mensagens padrão e opções de processamento de alertas.

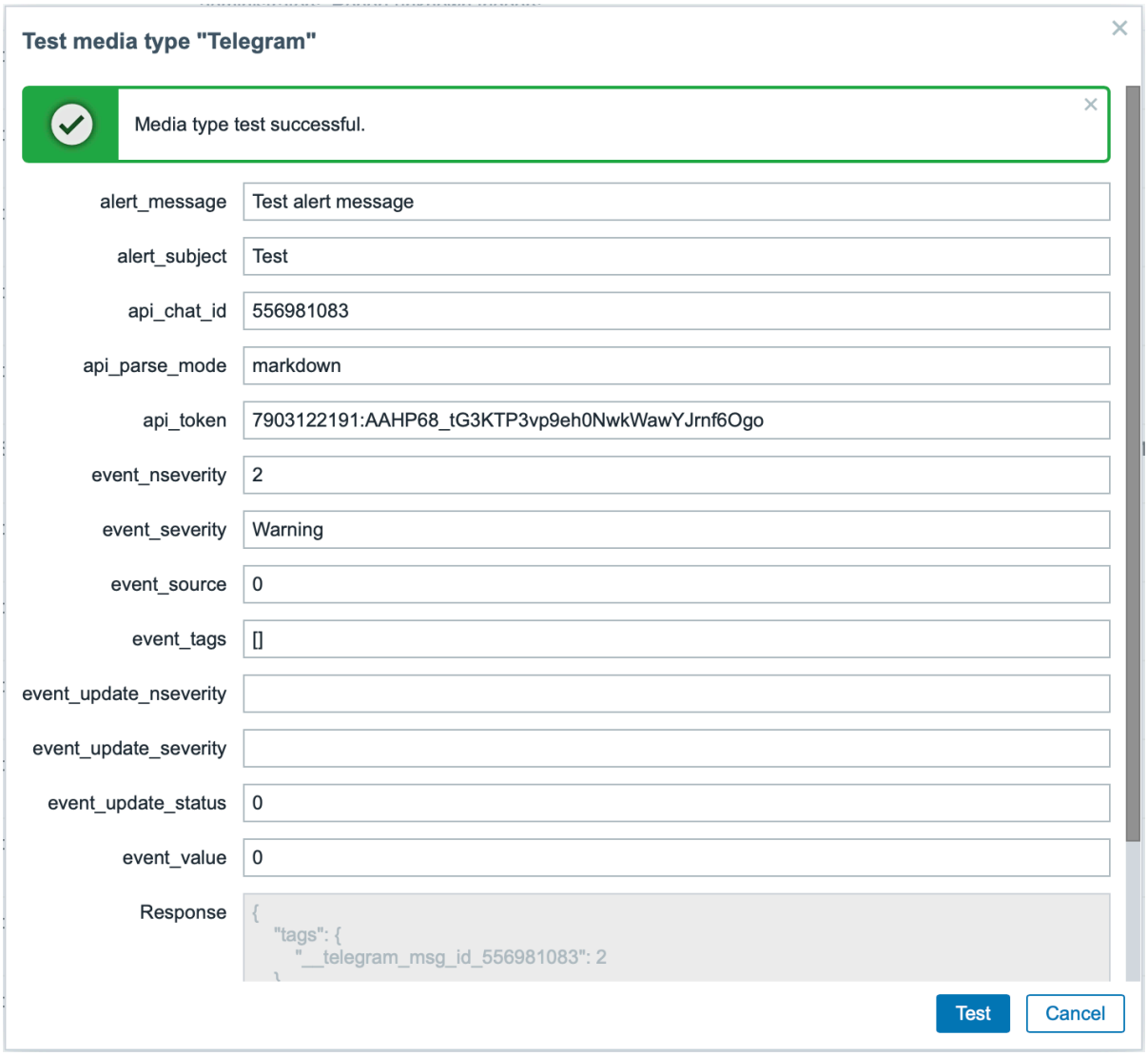
Warning:
Mesmo que um webhook não use mensagens padrão, os modelos de mensagem para os tipos de operação usados por este webhook ainda devem ser definidos.

Testando

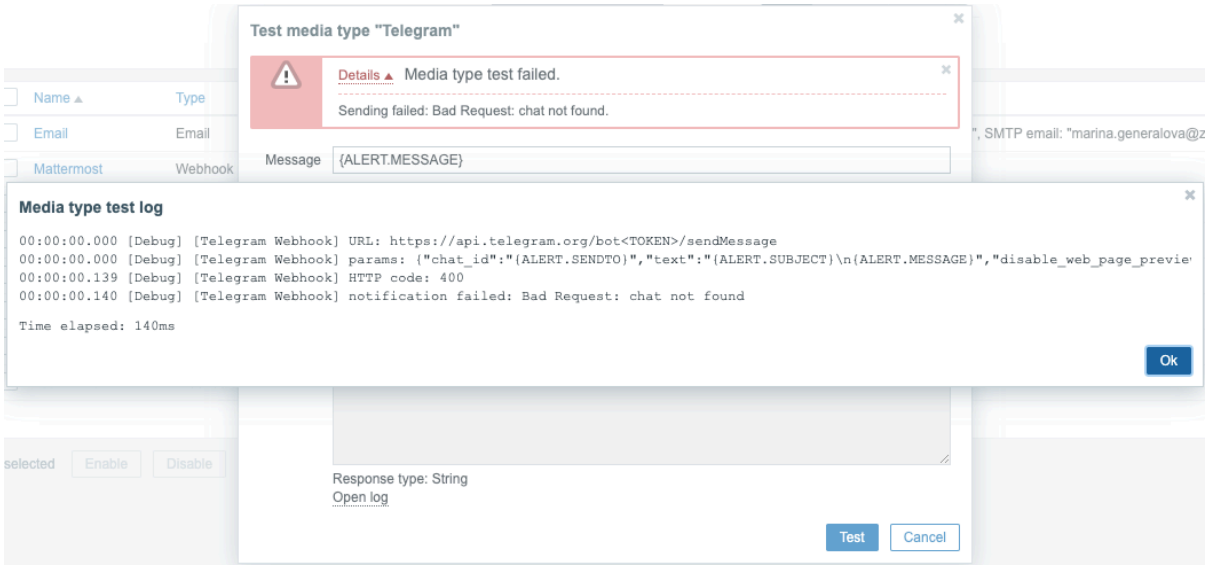
Para testar um tipo de mídia webhook configurado:

1. Localize o webhook relevante na lista de tipos de mídia.
2. Clique em *Testar* na última coluna da lista (uma janela de teste será aberta).
3. Edite os valores dos parâmetros do webhook conforme necessário. Substitua as macros por valores de exemplo; caso contrário, as macros não serão resolvidas e o teste falhará.
4. Clique em *Testar*.

Substituir ou excluir valores na janela de teste afeta apenas o procedimento de teste, os valores reais dos atributos do webhook permanecerão inalterados.



Para visualizar as entradas de log do teste do tipo de mídia sem sair da janela de teste, clique em *Abrir log* (uma nova janela pop-up será aberta).



Se o teste do webhook for bem-sucedido:

- A mensagem *"Teste do tipo de mídia bem-sucedido."* é exibida.
- A resposta do servidor aparece no campo cinza *Resposta*.
- O tipo de resposta (JSON ou String) é especificado abaixo do campo *Resposta*.

Se o teste do webhook falhar:

- A mensagem *"Media type test failed."* é exibida, seguida por detalhes adicionais da falha.

Mídia do usuário

Depois que o tipo de mídia for configurado, vá para a seção *Usuários* > *Usuários* e atribua a mídia webhook a um usuário existente ou crie um novo usuário para representar o webhook. As etapas para configurar a mídia do usuário para um usuário existente, sendo comuns para todos os tipos de mídia, estão descritas na página Tipos de mídia.

Se um webhook usar tags para armazenar o ID do ticket\mensagem, evite atribuir o mesmo webhook como mídia para diferentes usuários, pois isso pode causar erros no webhook (aplica-se à maioria dos webhooks que utilizam a opção *Incluir entrada de menu de evento*). Nesse caso, a melhor prática é criar um usuário dedicado para representar o webhook:

1. Após configurar o tipo de mídia webhook, vá para a seção *Usuários* > *Usuários* e crie um usuário Zabbix dedicado para representar o webhook - por exemplo, com o nome de usuário *Slack* para o webhook do Slack. Todas as configurações, exceto a mídia, podem ser deixadas como padrão, pois esse usuário não fará login no Zabbix.
2. No perfil do usuário, vá até a guia *Mídia* e adicione um webhook com as informações de contato necessárias. Se o webhook não usar um campo *Enviar para*, insira qualquer combinação de caracteres suportados para contornar os requisitos de validação.
3. Conceda a esse usuário pelo menos permissões de leitura permissões para todos os hosts para os quais ele deve enviar os alertas.

Ao configurar a ação de alerta, adicione esse usuário no campo *Enviar para usuários* nos Detalhes da operação - isso informará ao Zabbix para usar o webhook para notificações dessa ação.

Configurando ações de alerta

As ações determinam quais notificações devem ser enviadas via webhook. Os passos para configurar ações envolvendo webhooks são os mesmos que para todos os outros tipos de mídia, com estas exceções:

- Se um webhook usa tags de webhook para armazenar o ID do ticket\mensagem e lidar com operações de atualização\resolução, evite usar o mesmo webhook em várias ações de alerta para um único evento de problema. Se {EVENT.TAGS.<tag name>} existir e for atualizada no webhook, seu valor resultante será indefinido. Para evitar isso, use um novo nome de tag no webhook para armazenar valores atualizados. Isso se aplica aos webhooks Jira, Jira Service Desk, Mattermost, Opsgenie, OTRS, Redmine, ServiceNow, Slack, Zammad e Zendesk fornecidos pelo Zabbix e à maioria dos webhooks que utilizam a opção *Incluir entrada de menu de evento*. Observe, no entanto, que um único webhook pode ser usado em várias operações ou etapas de escalonamento da mesma ação, bem como em diferentes ações que não serão acionadas pelo mesmo evento de problema devido a diferentes condições.
- Ao usar um webhook em ações para eventos internos, certifique-se de marcar a caixa de seleção *Mensagem personalizada* e definir uma mensagem personalizada na configuração da operação da ação. Caso contrário, uma notificação não será enviada.

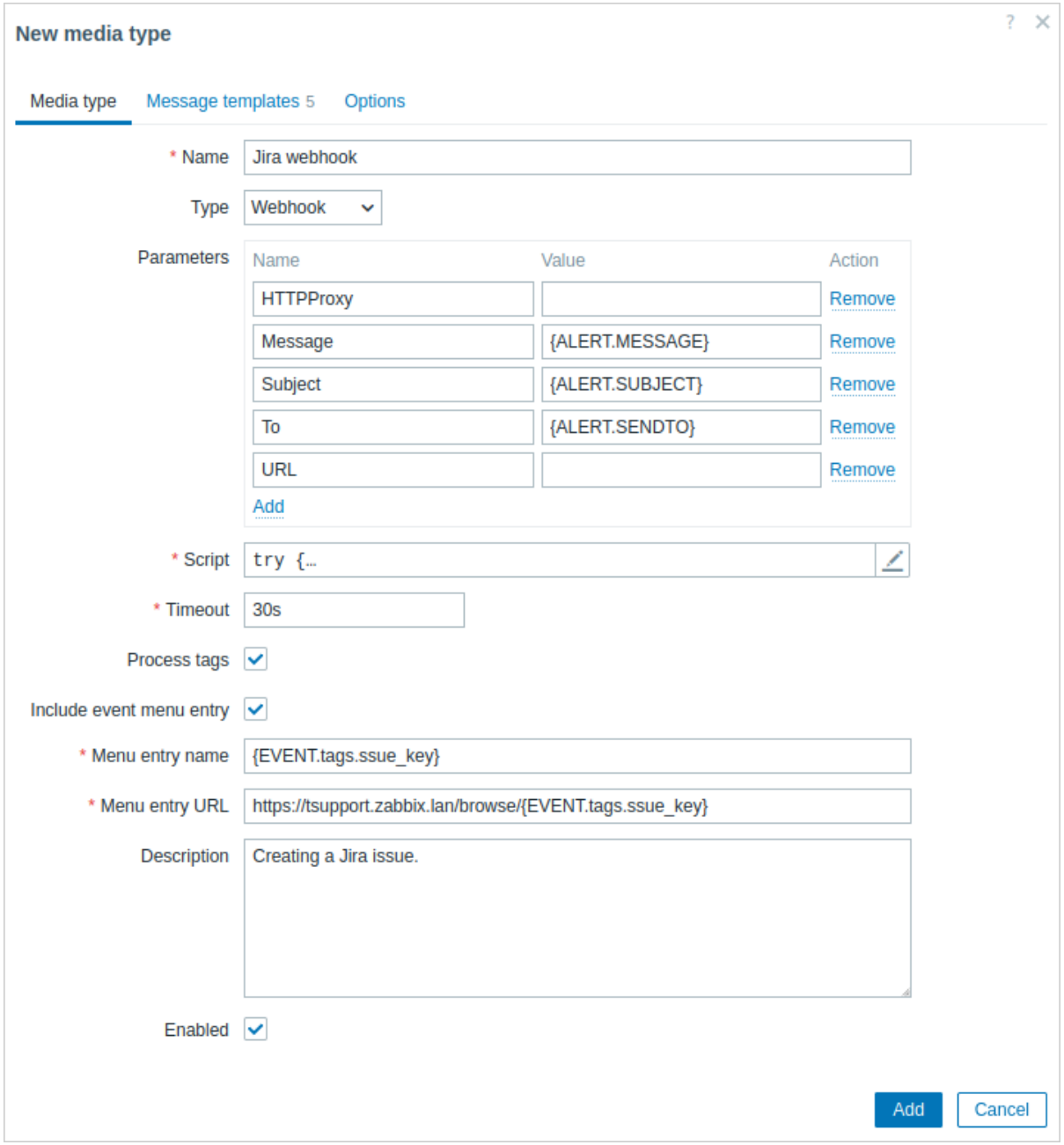
1 Exemplos de scripts de webhook

Visão geral

Embora o Zabbix ofereça um grande número de integrações de webhook disponíveis por padrão, você pode querer criar seus próprios webhooks. Esta seção fornece exemplos de scripts de webhook personalizados (usados no parâmetro *Script*). Consulte a seção webhook para a descrição de outros parâmetros de webhook.

Attention:
Não use atribuições não declaradas no pré-processamento do JavaScript. Use `var` para declarar variáveis locais.

Jira webhook (personalizado)



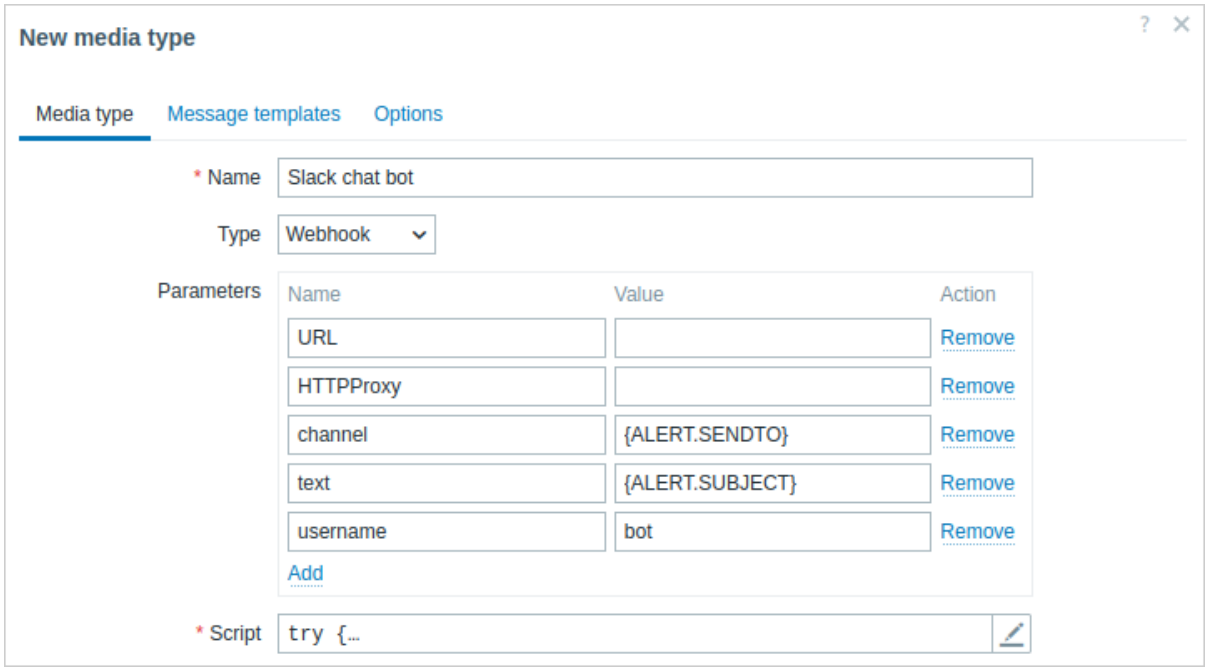
Este script criará um ticket no JIRA e retornará algumas informações sobre o ticket criado.

```
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 (personalizado)

Este webhook encaminhará notificações do Zabbix para um canal do Slack.



```
try {
    var params = JSON.parse(value),
        req = new HttpRequest(),
        response;

    if (params.HTTPProxy) {
        req.setProxy(params.HTTPProxy);
    }

    req.addHeader('Content-Type: application/x-www-form-urlencoded');

    Zabbix.log(4, '[ Slack webhook ] Webhook request with value=' + value);

    response = req.post(params.hook_url, 'payload=' + encodeURIComponent(value));
    Zabbix.log(4, '[ Slack webhook ] Responded with code: ' + req.getStatus() + '. Response: ' + response)

    try {
        response = JSON.parse(response);
    }
    catch (error) {
        if (req.getStatus() < 200 || req.getStatus() >= 300) {
            throw 'Request failed with status code ' + req.getStatus();
        }
        else {
            throw 'Request success, but response parsing failed.';
        }
    }

    if (req.getStatus() !== 200 || !response.ok || response.ok === 'false') {
        throw response.error;
    }

    return 'OK';
}
catch (error) {
    Zabbix.log(3, '[ Slack webhook ] Sending failed. Error: ' + error);

    throw 'Failed with error: ' + error;
}
```

2 Ações

Visão geral

Se você deseja que algumas operações ocorram como resultado de eventos (por exemplo, notificações enviadas), é necessário configurar ações.

As ações podem ser definidas em resposta a eventos de todos os tipos suportados:

- Ações de trigger - para eventos quando o status do trigger muda de *OK* para *PROBLEM* e vice-versa
- Ações de serviço - para eventos quando o status do serviço muda de *OK* para *PROBLEM* e vice-versa
- Ações de descoberta - para eventos quando ocorre a descoberta de rede
- Ações de autorregistro - para eventos quando novos agents ativos se autorregistram (ou quando a metadata do host muda para os já registrados)
- Ações internas - para eventos quando os items se tornam não suportados ou triggers entram em estado desconhecido

As principais diferenças das ações de serviço são:

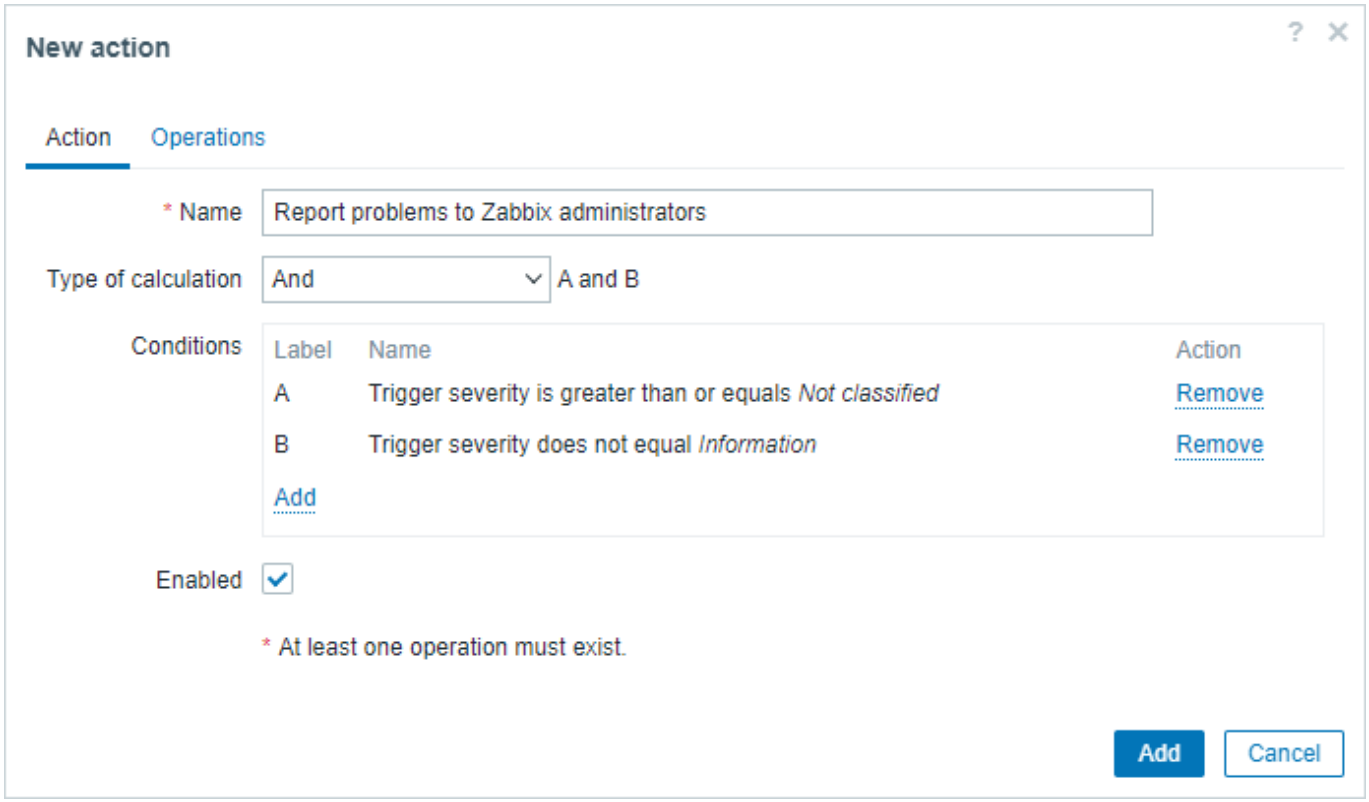
- O acesso do usuário às ações de serviço depende dos direitos de acesso aos serviços concedidos pela função do usuário
- As ações de serviço suportam um conjunto diferente de condições

Configurando uma ação

Para configurar uma ação, faça o seguinte:

- Vá para *Alertas* > *Ações* e selecione o tipo de ação necessário no submenu (você pode alternar para outro tipo depois, usando o menu suspenso do título).
- Clique em *Criar ação*.
- Nomeie a ação.
- Escolha as condições sob as quais as operações serão executadas.
- Escolha as operações a serem realizadas.

Atributos gerais da ação:



Todos os campos obrigatórios são marcados com um asterisco vermelho.

Parâmetro	Descrição
Nome	Nome único da ação.
Tipo de cálculo	Selecione a opção de avaliação para as condições da ação (com mais de uma condição): **E** - todas as condições devem ser atendidas. **Ou** - basta que uma condição seja atendida. **E/Ou** - combinação dos dois: E com tipos de condição diferentes e Ou com o mesmo tipo de condição. **Expressão personalizada** - uma fórmula de cálculo definida pelo usuário para avaliar as condições da ação.
Condições	Lista de condições da ação. Clique em *Adicionar* para adicionar uma nova condição. Se nenhuma condição for configurada, a ação será executada para cada evento que corresponda ao tipo de ação que está sendo configurado.
Habilitada	Marque a caixa de seleção para habilitar a ação. Caso contrário, ela será desabilitada.

1 Condições

Visão geral

É possível definir que uma ação seja executada somente se o evento corresponder a um conjunto definido de condições. As condições são definidas ao configurar a ação.

A correspondência de condições diferencia maiúsculas de minúsculas.

Ações de trigger

As seguintes condições podem ser usadas em ações baseadas em trigger:

Tipo de condição	Operadores suportados	Descrição
Grupo de hosts	igual não igual	Especifique grupos de hosts ou grupos de hosts a serem excluídos. **igual** - o evento pertence a este grupo de hosts. **não igual** - o evento não pertence a este grupo de hosts. Especificar um grupo de hosts pai seleciona implicitamente todos os grupos de hosts aninhados. Para especificar apenas o grupo pai, todos os grupos aninhados devem ser definidos adicionalmente com o operador **não igual**.
Template	igual não igual	Especifique templates ou templates a serem excluídos. **igual** - o evento pertence a uma trigger herdada deste template. **não igual** - o evento não pertence a uma trigger herdada deste template.
Host	igual não igual	Especifique hosts ou hosts a serem excluídos. **igual** - o evento pertence a este host. **não igual** - o evento não pertence a este host.
Nome da tag	igual não igual contém não contém	Especifique a tag do evento ou a tag do evento a ser excluída. **igual** - o evento possui esta tag. **não igual** - o evento não possui esta tag. **contém** - o evento possui uma tag contendo esta string. **não contém** - o evento não possui uma tag contendo esta string.
Valor da tag	igual não igual contém não contém	Especifique a combinação de tag e valor do evento ou a combinação de tag e valor a ser excluída. **igual** - o evento possui esta tag e valor. **não igual** - o evento não possui esta tag e valor. **contém** - o evento possui uma tag e valor contendo estas strings. **não contém** - o evento não possui uma tag e valor contendo estas strings.
Trigger	igual não igual	Especifique triggers ou triggers a serem excluídas. **igual** - o evento é gerado por esta trigger. **não igual** - o evento é gerado por qualquer outra trigger, exceto esta.
Nome do evento	contém não contém	Especifique uma string no nome do evento gerado pela trigger ou uma string a ser excluída. Por padrão, o nome do evento corresponde ao nome da trigger, a menos que um nome de evento personalizado seja especificado na configuração da trigger. **contém** - o nome do evento contém esta string. **não contém** - esta string é excluída do nome do evento. Nota: O valor inserido será comparado ao nome do evento com todas as macros expandidas.
Severidade da trigger	igual não igual é maior ou igual é menor ou igual	Especifique a severidade da trigger. **igual** - igual à severidade da trigger. **não igual** - diferente da severidade da trigger. **é maior ou igual** - maior ou igual à severidade da trigger. **é menor ou igual** - menor ou igual à severidade da trigger.
Período de tempo	em não em	Especifique um período de tempo ou um período de tempo a ser excluído. **em** - o horário do evento está dentro do período de tempo. **não em** - o horário do evento não está dentro do período de tempo. Veja a página de especificação do período de tempo para descrição do formato. Macros de usuário são suportadas.
O problema está suprimido	não sim	Especifique se o problema está suprimido (não exibido) devido à manutenção do host. **não** - o problema não está suprimido. **sim** - o problema está suprimido.

Ações de serviço

As seguintes condições podem ser usadas em ações de serviço:

Tipo de condição	Operadores suportados	Descrição
Serviço	igual não é igual	Especifique um serviço ou um serviço a ser excluído. **igual** - o evento pertence a este serviço. **não é igual** - o evento não pertence a este serviço. Especificar um serviço pai seleciona implicitamente todos os serviços filhos. Para especificar apenas o serviço pai, todos os serviços aninhados devem ser definidos adicionalmente com o operador **não é igual**.
Nome do serviço	contém não contém	Especifique uma string no nome do serviço ou uma string a ser excluída. **contém** - o evento é gerado por um serviço que contém esta string no nome. **não contém** - esta string não pode ser encontrada no nome do serviço.
Nome da tag do serviço	igual não é igual contém não contém	Especifique uma tag de evento ou uma tag de evento a ser excluída. As tags de evento de serviço podem ser definidas na seção de configuração do serviço *Tags*. **igual** - o evento possui esta tag. **não é igual** - o evento não possui esta tag. **contém** - o evento possui uma tag contendo esta string. **não contém** - o evento não possui uma tag contendo esta string.
Valor da tag do serviço	igual não é igual contém não contém	Especifique uma combinação de tag e valor de evento ou uma combinação de tag e valor a ser excluída. As tags de evento de serviço podem ser definidas na seção de configuração do serviço *Tags*. **igual** - o evento possui esta tag e valor. **não é igual** - o evento não possui esta tag e valor. **contém** - o evento possui uma tag e valor contendo essas strings. **não contém** - o evento não possui uma tag e valor contendo essas strings.

Attention:
Certifique-se de definir modelos de mensagem para ações de serviço no menu *Alertas* → *Tipos de mídia*. Caso contrário, as notificações não serão enviadas.

Ações de descoberta

As seguintes condições podem ser usadas em eventos baseados em descoberta:

Tipo de condição	Operadores suportados	Descrição
Host IP	equals does not equal	Especifique um intervalo de endereços IP ou um intervalo a ser excluído para um host descoberto. **equals** - o IP do host está no intervalo. **does not equal** - o IP do host não está no intervalo. Pode ter os seguintes formatos: IP único: 192.168.1.33 Intervalo de endereços IP: 192.168.1-10.1-254 Máscara de IP: 192.168.4.0/24 Lista: 192.168.1.1-254, 192.168.2.1-100, 192.168.2.200, 192.168.4.0/24 Espaços no formato de lista são suportados.
Service type	equals does not equal	Especifique um tipo de serviço de um serviço descoberto ou um tipo de serviço a ser excluído. **equals** - corresponde ao serviço descoberto. **does not equal** - não corresponde ao serviço descoberto. Tipos de serviço disponíveis: SSH, LDAP, SMTP, FTP, HTTP, HTTPS, POP, NNTP, IMAP, TCP, Zabbix agent, SNMPv1 agent, SNMPv2 agent, SNMPv3 agent, ICMP ping, telnet.

Tipo de condição	Operadores suportados	Descrição
Service port	equals does not equal	Especifique um intervalo de portas TCP de um serviço descoberto ou um intervalo a ser excluído. **equals** - a porta do serviço está no intervalo. **does not equal** - a porta do serviço não está no intervalo.
Discovery rule	equals does not equal	Especifique uma regra de descoberta ou uma regra de descoberta a ser excluída. **equals** - usando esta regra de descoberta. **does not equal** - usando qualquer outra regra de descoberta, exceto esta.
Discovery check	equals does not equal	Especifique uma verificação de descoberta ou uma verificação de descoberta a ser excluída. **equals** - usando esta verificação de descoberta. **does not equal** - usando qualquer outra verificação de descoberta, exceto esta.
Discovery object	equals	Especifique o objeto descoberto. **equals** - igual ao objeto descoberto (um dispositivo ou um serviço).
Discovery status	equals	**Up** - corresponde aos eventos 'Host Up' e 'Service Up'. **Down** - corresponde aos eventos 'Host Down' e 'Service Down'. **Discovered** - corresponde aos eventos 'Host Discovered' e 'Service Discovered'. **Lost** - corresponde aos eventos 'Host Lost' e 'Service Lost'.
Uptime/Downtime	is greater than or equals is less than or equals	Tempo de atividade para eventos 'Host Up' e 'Service Up'. Tempo de inatividade para eventos 'Host Down' e 'Service Down'. **is greater than or equals** - é maior ou igual a. O parâmetro é dado em segundos. **is less than or equals** - é menor ou igual a. O parâmetro é dado em segundos.
Received value	equals does not equal is greater than or equals is less than or equals contains does not contain	Especifique o valor recebido de uma verificação de agent (Zabbix, SNMP) em uma regra de descoberta. Comparação de string. Se várias verificações de agent Zabbix ou SNMP forem configuradas para uma regra, os valores recebidos para cada uma delas serão verificados (cada verificação gera um novo evento que é comparado com todas as condições). **equals** - igual ao valor. **does not equal** - diferente do valor. **is greater than or equals** - maior ou igual ao valor. **is less than or equals** - menor ou igual ao valor. **contains** - contém a substring. O parâmetro é dado como uma string. **does not contain** - não contém a substring. O parâmetro é dado como uma string.
Proxy	equals does not equal	Especifique um proxy ou um proxy a ser excluído. **equals** - usando este proxy. **does not equal** - usando qualquer outro proxy, exceto este.

Note:
As verificações de serviço em uma regra de descoberta, que resultam em eventos de descoberta, não ocorrem simultaneamente. Portanto, se **vários** valores forem configurados para as condições `Service type`, `Service port` ou `Received value` na ação, eles serão comparados a um evento de descoberta por vez, mas **não** a vários eventos simultaneamente. Como resultado, ações com vários valores para os mesmos tipos de verificação podem não ser executadas corretamente.

Ações de autorregistro

As seguintes condições podem ser usadas em ações baseadas no autorregistro ativo do agent:

Tipo de condição	Operadores suportados	Descrição
Host metadata	contains does not contain matches does not match	Especifique a metadata do host ou a metadata do host a ser excluída. **contains** - a metadata do host contém a string. **does not contain** - a metadata do host não contém a string. A metadata do host pode ser especificada em um arquivo de configuração do agent. **matches** - a metadata do host corresponde à expressão regular. **does not match** - a metadata do host não corresponde à expressão regular.
Host name	contains does not contain matches does not match	Especifique um nome de host ou um nome de host a ser excluído. **contains** - o nome do host contém a string. **does not contain** - o nome do host não contém a string. **matches** - o nome do host corresponde à expressão regular. **does not match** - o nome do host não corresponde à expressão regular.
Proxy	equals does not equal	Especifique um proxy ou um proxy a ser excluído. **equals** - usando este proxy. **does not equal** - usando qualquer outro proxy, exceto este.

Ações de eventos internos

As seguintes condições podem ser definidas para ações baseadas em eventos internos:

Tipo de condição	Operadores suportados	Descrição
Tipo de evento	igual	**Item em estado "não suportado"** - corresponde a eventos em que um item passa de um estado 'normal' para 'não suportado'. **Regra de descoberta de baixo nível em estado "não suportado"** - corresponde a eventos em que uma regra de descoberta de baixo nível passa de um estado 'normal' para 'não suportado'. **Trigger em estado "desconhecido"** - corresponde a eventos em que um trigger passa de um estado 'normal' para 'desconhecido'.
Grupo de hosts	igual não é igual	Especifique grupos de hosts ou grupos de hosts a serem excluídos. **igual** - o evento pertence a este grupo de hosts. **não é igual** - o evento não pertence a este grupo de hosts.
Nome da tag	igual não é igual contém não contém	Especifique a tag do evento ou a tag do evento a ser excluída. **igual** - o evento possui esta tag. **não é igual** - o evento não possui esta tag. **contém** - o evento possui uma tag que contém esta string. **não contém** - o evento não possui uma tag que contenha esta string.
Valor da tag	igual não é igual contém não contém	Especifique a combinação de tag e valor do evento ou a combinação de tag e valor a ser excluída. **igual** - o evento possui esta tag e valor. **não é igual** - o evento não possui esta tag e valor. **contém** - o evento possui uma tag e valor que contenham estas strings. **não contém** - o evento não possui uma tag e valor que contenham estas strings.
Template	igual não é igual	Especifique templates ou templates a serem excluídos. **igual** - o evento pertence a um item/trigger/regra de descoberta de baixo nível herdado deste template. **não é igual** - o evento não pertence a um item/trigger/regra de descoberta de baixo nível herdado deste template.
Host	igual não é igual	Especifique hosts ou hosts a serem excluídos. **igual** - o evento pertence a este host. **não é igual** - o evento não pertence a este host.

Tipo de cálculo

As seguintes opções de cálculo de condições estão disponíveis:

- **E** - todas as condições devem ser atendidas

Observe que o uso do cálculo "E" não é permitido entre vários triggers quando eles são selecionados como uma condição `Trigger=`. As ações só podem ser executadas com base no evento de um trigger.

- **Ou** - basta que uma condição seja atendida
- **E/Ou** - combinação dos dois: E com tipos de condição diferentes e OU com o mesmo tipo de condição, por exemplo:

Grupo de hosts igual a Oracle servers
 Grupo de hosts igual a MySQL servers
 Nome do evento contém 'Database is down'
 Nome do evento contém 'Database is unavailable'

é avaliado como

(Grupo de hosts igual a Oracle servers **ou** Grupo de hosts igual a MySQL servers**) e (**Nome do evento contém 'Database is down' **ou** Nome do evento contém 'Database is unavailable'**)**

- **Expressão personalizada** - uma fórmula de cálculo definida pelo usuário para avaliar as condições da ação. Deve incluir todas as condições (representadas por letras maiúsculas A, B, C, ...) e pode incluir espaços, tabulações, parênteses ( ), **and** (sensível a maiúsculas e minúsculas), **or** (sensível a maiúsculas e minúsculas), **not** (sensível a maiúsculas e minúsculas).

Enquanto o exemplo anterior com `E/Ou` seria representado como (A ou B) e (C ou D), em uma expressão personalizada você pode ter várias outras formas de cálculo:

(A e B) e (C ou D)
 (A e B) ou (C e D)
 ((A ou B) e C) ou D
 (not (A ou B) e C) ou not D
 etc.

Excluindo objetos usados em ações

Ao tentar excluir objetos que são usados em uma condição/operação de ação, o usuário receberá uma mensagem de erro correspondente.

As ações não são desabilitadas se as condições ou operações da ação:

- tiverem objetos excluídos;
- tiverem objetos que não são mais descobertos pela descoberta de baixo nível e foram removidos (após o período de *Excluir recursos perdidos*).

As condições/operações de ação não são removidas se tiverem objetos ausentes. Em vez disso, os objetos excluídos são exibidos como *Grupo de hosts excluído*, *Host excluído*, *Trigger excluído*, etc. Um ícone de aviso com a mensagem "Esta ação possui condições ou operações que fazem referência a objeto(s) excluído(s)." é exibido na coluna "Info" da lista de ações.

Observe que **antes do Zabbix 7.0.10**, se um determinado objeto (host, template, trigger, etc.) usado em uma condição/operação de ação for excluído, a condição/operação é removida e a ação é desabilitada.

2 Operações

Visão geral

Você pode definir as seguintes operações para todos os eventos:

- Enviar uma mensagem
- Executar um comando remoto

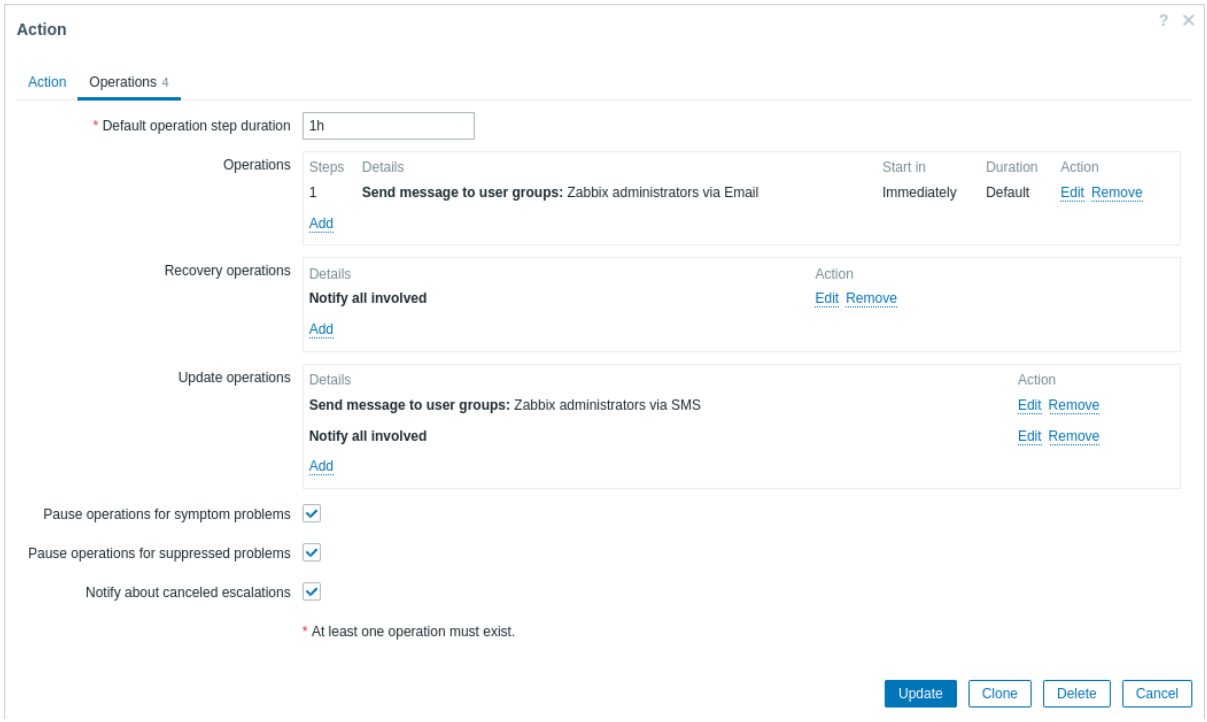
> **Attention:**
> O Zabbix server não cria alertas se o acesso ao host for explicitamente "negado" para o usuário definido como destinatário da operação da ação ou se o usuário não tiver direitos definidos para o host.

Para eventos de descoberta e autorregistro, operações adicionais estão disponíveis:

- Adicionar host
- Remover host
- Habilitar host
- Desabilitar host
- Adicionar ao grupo de hosts
- Remover do grupo de hosts
- Adicionar tags ao host
- Remover tags do host
- Vincular template
- Desvincular template
- Definir modo de inventário do host

Configurando uma operação

Para configurar uma operação, vá até a aba *Operações* na configuração da ação.



Atributos gerais da operação:

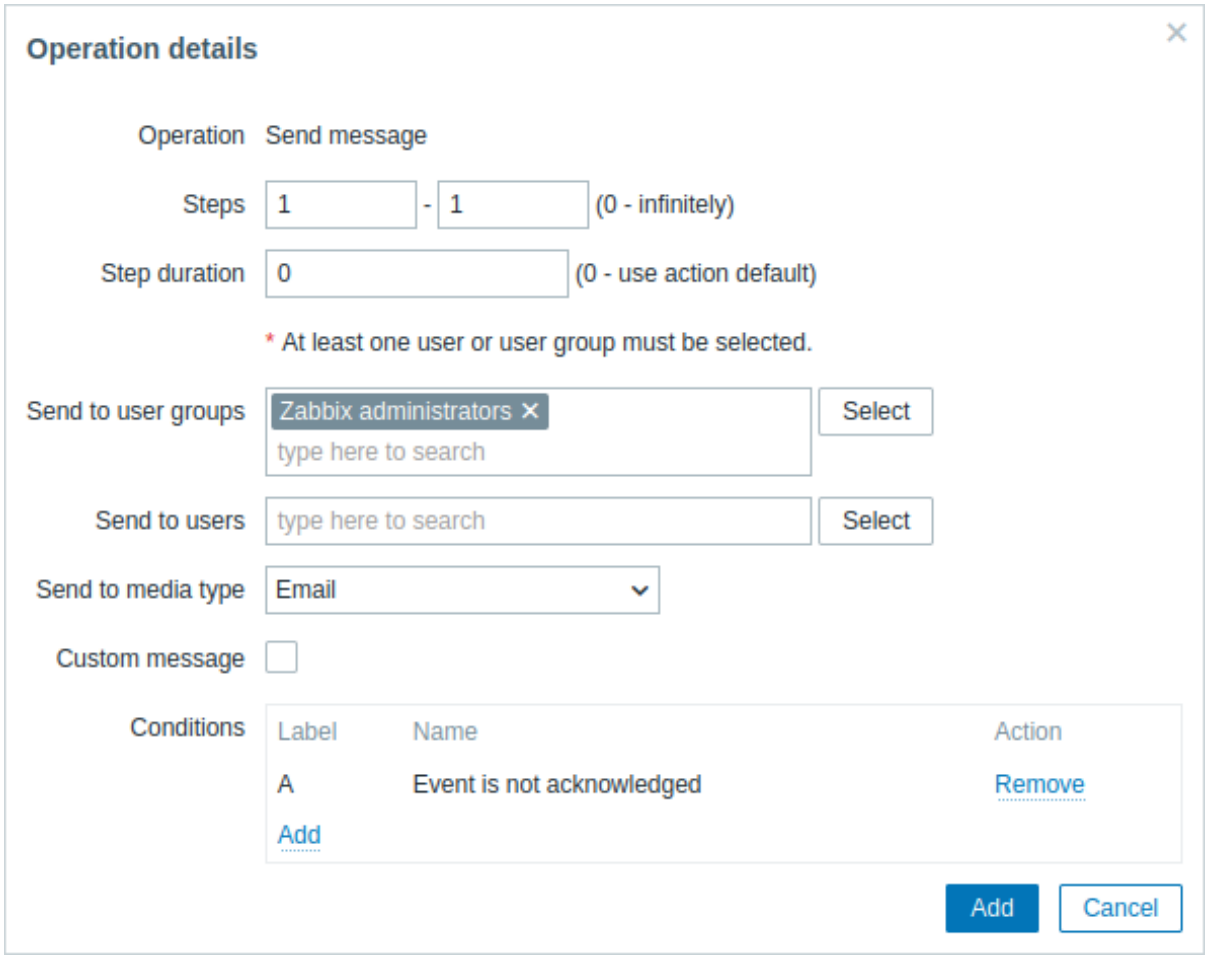
Parâmetro	Descrição
Duração padrão do passo da operação	Duração de um passo da operação por padrão (60 segundos a 1 semana). Por exemplo, uma duração de passo de uma hora significa que, se uma operação for realizada, uma hora se passará antes do próximo passo. Sufixos de tempo são suportados, por exemplo, 60s, 1m, 2h, 1d. Macros de usuário são suportadas.
Operações	As operações da ação (se houver) são exibidas, com os seguintes detalhes: **Passos** - passo(s) de escalonamento ao qual a operação está atribuída. **Detalhes** - tipo de operação e seu destinatário/alvo. A lista de operações também exibe o tipo de mídia (e-mail, SMS ou script) usado, bem como o nome e sobrenome (entre parênteses após o nome de usuário) de um destinatário da notificação. **Iniciar em** - quanto tempo após um evento a operação é realizada. **Duração (seg)** - a duração do passo é exibida. *Padrão* é exibido se o passo usar a duração padrão, e um tempo é exibido se uma duração personalizada for usada. **Ação** - links para editar e remover uma operação são exibidos.
Operações de recuperação	As operações da ação (se houver) são exibidas, com os seguintes detalhes: **Detalhes** - tipo de operação e seu destinatário/alvo. A lista de operações também exibe o tipo de mídia (e-mail, SMS ou script) usado, bem como o nome e sobrenome (entre parênteses após o nome de usuário) de um destinatário da notificação. **Ação** - links para editar e remover uma operação são exibidos.
Operações de atualização	As operações da ação (se houver) são exibidas, com os seguintes detalhes: **Detalhes** - tipo de operação e seu destinatário/alvo. A lista de operações também exibe o tipo de mídia (e-mail, SMS ou script) usado, bem como o nome e sobrenome (entre parênteses após o nome de usuário) de um destinatário da notificação. **Ação** - links para editar e remover uma operação são exibidos.
Pausar operações para problemas sintomáticos	Marque esta caixa de seleção para pausar as operações (após a primeira operação) para problemas sintomáticos. Observe que esta configuração afeta apenas escalonamentos de problemas; operações de recuperação e atualização não serão afetadas. Esta opção está disponível apenas para *Ações de trigger*.

Parâmetro	Descrição
Pausar operações para problemas suprimidos	Marque esta caixa de seleção para atrasar o início das operações durante o período de manutenção. Quando as operações forem iniciadas, após a manutenção, todas as operações serão realizadas, incluindo aquelas para os eventos ocorridos durante a manutenção. Observe que esta configuração afeta apenas escalonamentos de problemas; operações de recuperação e atualização não serão afetadas. Se você desmarcar esta caixa de seleção, as operações serão executadas sem atraso, mesmo durante um período de manutenção. Esta opção não está disponível para *Ações de serviço*.
Notificar sobre escalonamentos cancelados	Desmarque esta caixa de seleção para desabilitar notificações sobre escalonamentos cancelados (quando host, item, trigger ou ação for desabilitado).

Todos os campos obrigatórios são marcados com um asterisco vermelho.

Para configurar os detalhes de uma nova operação, clique em Add no bloco *Operações*. Para editar uma operação existente, clique em Edit ao lado da operação. Uma janela pop-up será aberta onde você poderá editar os detalhes do passo da operação.

Detalhes da operação



Parâmetro	Descrição
Operação	Selecione a operação: **Enviar mensagem** - enviar mensagem para o usuário. **<nome do comando remoto>** - executar um comando remoto. Os comandos estão disponíveis para execução se previamente definidos em scripts globais com *Operação de ação* selecionada como seu escopo. Mais operações estão disponíveis para eventos baseados em descoberta e auto-registro (veja acima).
Etapas	Selecione a(s) etapa(s) para atribuir a operação em uma programação de escalação: **De** - executar a partir desta etapa. **Até** - executar até esta etapa (0=infinito, a execução não será limitada).

Parâmetro	Descrição
Duração da etapa	Duração personalizada para estas etapas (0=usar a duração padrão da etapa). Sufixos de tempo são suportados, por exemplo, 60s, 1m, 2h, 1d. Macros de usuário são suportadas. Várias operações podem ser atribuídas à mesma etapa. Se essas operações tiverem diferentes durações de etapa definidas, a menor será considerada e aplicada à etapa.
Tipo de operação: enviar mensagem	
Enviar para grupos de usuários	Selecione os grupos de usuários para enviar a mensagem. O grupo de usuários deve ter pelo menos permissão de ”leitura” permissões ao host para ser notificado.
Enviar para usuários	Selecione os usuários para enviar a mensagem. O usuário deve ter pelo menos permissão de ”leitura” permissões ao host para ser notificado.
Enviar para tipo de mídia	Enviar mensagem para todos os tipos de mídia disponíveis (configurados e habilitados) ou apenas para um específico. Desde o Zabbix 7.0.7, este parâmetro foi renomeado de *Enviar apenas para* para *Enviar para tipo de mídia*.
Mensagem personalizada	Se selecionado, a mensagem personalizada pode ser configurada. Para notificações sobre eventos internos via webhooks, a mensagem personalizada é obrigatória.
Assunto	Assunto da mensagem personalizada. O assunto pode conter macros. É limitado a 255 caracteres.
Mensagem	A mensagem personalizada. A mensagem pode conter macros. É limitada a uma certa quantidade de caracteres dependendo do tipo de banco de dados (veja Enviando mensagem para mais informações).
Tipo de operação: comando remoto	

Parâmetro	Descrição
Lista de destinos	Selecione os destinos para executar o comando: **Host atual** - o comando é executado no host do trigger que causou o evento de problema. Esta opção não funcionará se houver vários hosts no trigger. **Host** - selecione o(s) host(s) para executar o comando. **Grupo de hosts** - selecione o(s) grupo(s) de hosts para executar o comando. Especificar um grupo de hosts pai seleciona implicitamente todos os grupos de hosts aninhados. Assim, o comando remoto também será executado em hosts de grupos aninhados. Um comando em um host é executado apenas uma vez, mesmo que o host corresponda mais de uma vez (por exemplo, de vários grupos de hosts; individualmente e de um grupo de hosts). A lista de destinos é irrelevante se um script personalizado for executado no Zabbix server. Selecionar mais destinos neste caso apenas resulta na execução do script no server mais vezes. Observe que, para scripts globais, a seleção de destino também depende da configuração *Grupo de hosts* no script global. A opção *Lista de destinos* não está disponível para *Ações de serviço* porque, neste caso, comandos remotos são sempre executados no Zabbix server.
Condições	Condição para realizar a operação: **Evento não reconhecido** - apenas quando o evento não for reconhecido. **Evento reconhecido** - apenas quando o evento for reconhecido. A opção *Condições* está disponível apenas para *Ações de trigger*.

Quando terminar, clique em *Adicionar* para adicionar a operação à lista de *Operações*.

1 Enviando mensagem

Visão geral

Enviar uma mensagem é uma das melhores maneiras de notificar as pessoas sobre um problema. Por isso é uma das principais ações oferecidas pelo Zabbix.

Configuração

Para poder enviar e receber notificações do Zabbix, você deve:

- definir a mídia para a qual enviar uma mensagem

Se a operação ocorrer fora do período de tempo Quando ativo definido para a mídia selecionada na configuração do usuário, a mensagem não será enviada.

A gravidade padrão do trigger (’Não classificado’) **deve ser** marcada na configuração da mídia do usuário se você deseja receber notificações para eventos que não sejam de trigger, como descoberta, autoregistro ativo do agent ou eventos internos.

- configurar uma operação de ação que envie uma mensagem para uma das mídias definidas

Attention:
O Zabbix envia notificações apenas para os usuários que tenham pelo menos permissão de ’leitura’ para o host que gerou o evento. Pelo menos um host de uma expressão de trigger deve ser acessível.

Você pode configurar cenários personalizados para envio de mensagens usando escalonamentos.

Para receber e ler e-mails do Zabbix com sucesso, os servidores/clients de e-mail devem suportar o formato padrão ’SMTP/MIME email’, pois o Zabbix envia dados em UTF-8 (se o assunto contiver apenas caracteres ASCII, ele não será codificado em UTF-8). O assunto e o corpo da mensagem são codificados em base64 para seguir o padrão do formato ’SMTP/MIME email’.

O limite de mensagem após a expansão de todas as macros é o mesmo que o limite de mensagem para Comandos remotos.

Acompanhamento de mensagens

Você pode visualizar o status das mensagens enviadas em *Monitoramento* → *Problemas*.

Na coluna *Ações* você pode ver informações resumidas sobre as ações realizadas. Números verdes representam mensagens enviadas, números vermelhos - mensagens com falha. *Em andamento* indica que uma ação foi iniciada. *Falha* informa que nenhuma ação foi executada com sucesso.

Se você clicar no horário do evento para visualizar os detalhes do evento, poderá ver detalhes das mensagens enviadas (ou não enviadas) devido ao evento no bloco *Ações*.

Em *Relatórios* → *Log de ações* você verá detalhes de todas as ações realizadas para aqueles eventos que possuem uma ação configurada.

2 Comandos remotos

Visão geral

Com comandos remotos você pode definir que um determinado comando pré-definido é executado automaticamente no host monitorado sob alguma condição.

Assim, os comandos remotos são um mecanismo poderoso para inteligência proativa monitoramento.

Nos usos mais óbvios do recurso, você pode tentar:

- Reinicie automaticamente algum aplicativo (servidor web, middleware, CRM) se não responder
- Use o comando IPMI 'reboot' para reinicializar algum servidor remoto se isso acontecer não atende pedidos
- Liberar automaticamente o espaço em disco (removendo arquivos antigos, limpando /tmp) se ficar sem espaço em disco
- Migrar uma VM de uma caixa física para outra dependendo da CPU carga
- Adicione novos nós a um ambiente de nuvem em caso de CPU insuficiente (disco, memória, o que for) recursos

A configuração de uma ação para comandos remotos é semelhante à de envio uma mensagem, a única diferença é que o Zabbix irá executar um comando em vez de enviar uma mensagem.

Comandos remotos podem ser executados pelo servidor Zabbix, proxy ou agente. Controlo remoto comandos no agente Zabbix podem ser executados diretamente pelo servidor Zabbix ou através do proxy Zabbix. Tanto no Zabbix Agent quanto no Zabbix proxy remote comandos são desabilitados por padrão. Eles podem ser habilitados por:

- adicionar um parâmetro `AllowKey=system.run[*]` na configuração do agente;
- definindo o parâmetro EnableRemoteCommands para '1' no proxy configuração.

Os comandos remotos executados pelo servidor Zabbix são executados conforme descrito em Execução do comando incluindo saída verificação de código.

Os comandos remotos são executados mesmo se o host de destino estiver em manutenção.

Limite de comando remoto

O limite de comando remoto após resolver todas as macros depende do tipo de banco de dados e do conjunto de caracteres (caracteres não ASCII requerem mais de um byte para serem armazenados):

Banco de dados	Limite em caracteres	Limite em bytes
MySQL	65535	65535
Oracle Database	2048	4000
PostgreSQL	65535	sem limite
SQLite (apenas Zabbix proxy)	65535	sem limite

A saída da execução do comando remoto (valor de retorno) é limitada a 16MB (incluindo espaços em branco à direita que são truncados). O limite do comando remoto IPMI é baseado na biblioteca IPMI instalada. Observe que os limites do banco de dados se aplicam a todos os comandos remotos.

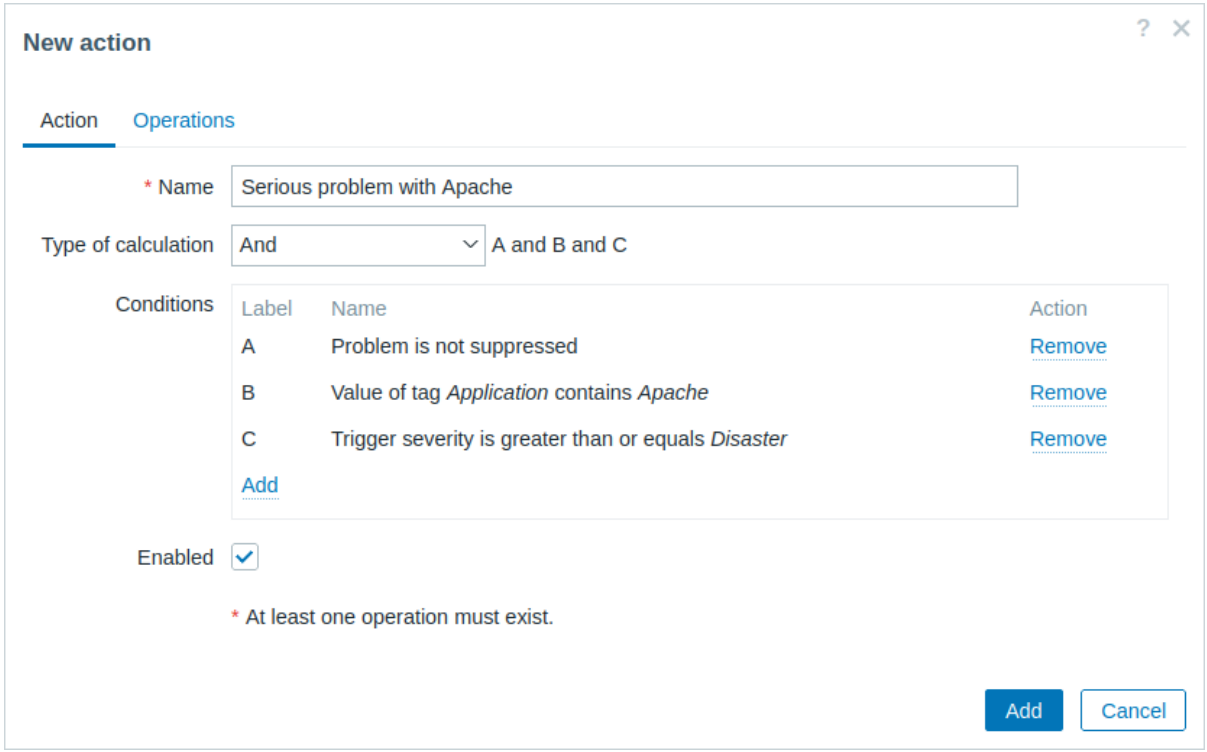
Configuração

Aqueles comandos remotos que são executados no agent Zabbix (scripts personalizados) devem ser primeiro habilitados na configuração do agent.

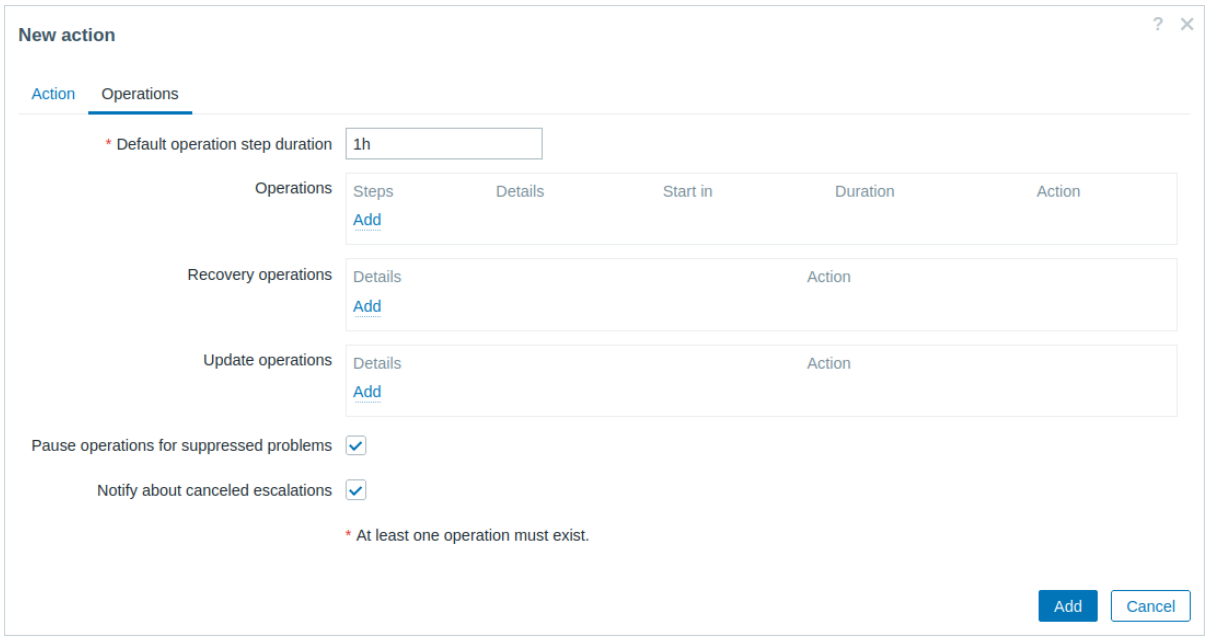
Certifique-se de que o parâmetro AllowKey=system.run[<command>,*] seja adicionado para cada comando permitido na configuração do agent para permitir comandos específicos com o modo nowait. Reinicie o daemon do agent se alterar este parâmetro.

Então, ao configurar uma nova ação em *Alertas* → *Ações* → *Ações de trigger*:

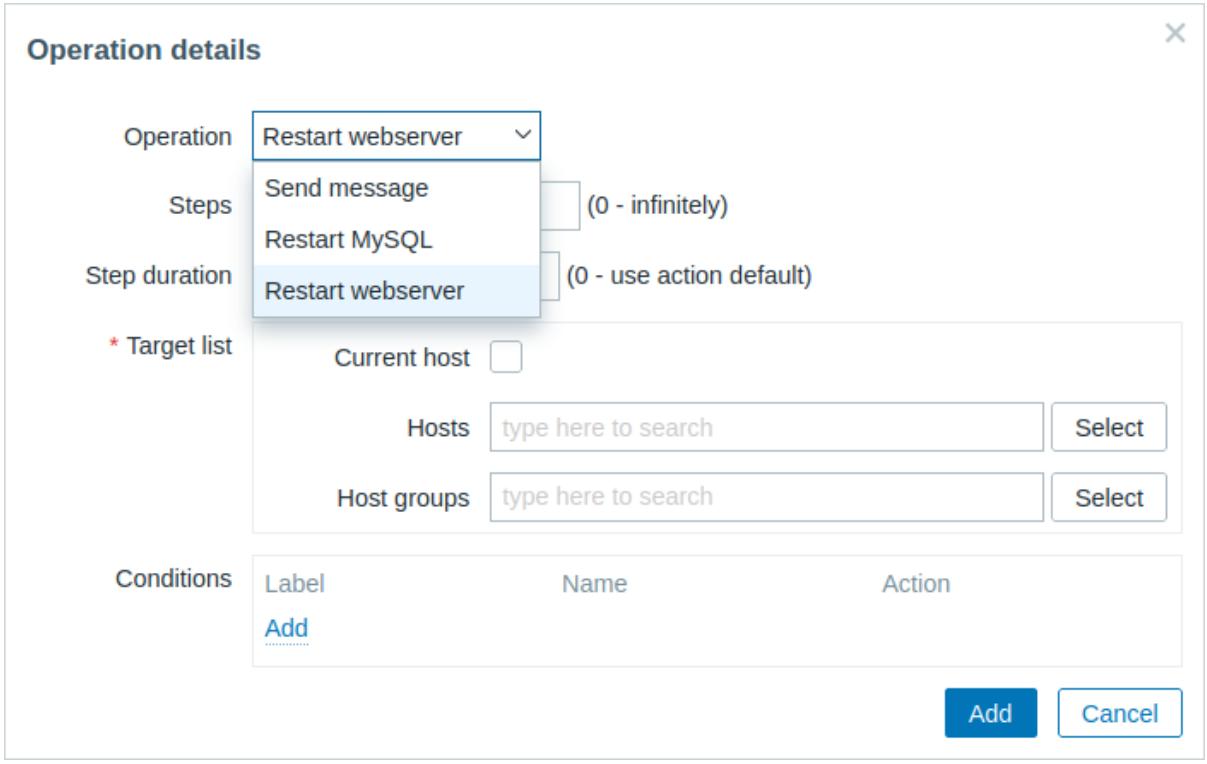
1. Defina as condições apropriadas, por exemplo, defina que a ação será ativada para qualquer problema de desastre com uma das aplicações Apache.



2. Na aba *Operações*, clique em *Adicionar* no bloco *Operações*, *Operações de recuperação* ou *Operações de atualização*.



3. Selecione um dos scripts predefinidos na lista suspensa *Operação* e defina a *Lista de destinos* para o script.



Scripts predefinidos

Os scripts disponíveis para operações de ação (webhook, script, SSH, Telnet, IPMI) são definidos em scripts globais.

Por exemplo:

```
sudo /etc/init.d/apache restart
```

Neste caso, o Zabbix tentará reiniciar um processo do Apache. Com este comando, certifique-se de que o comando seja executado no agent do Zabbix (clique no botão *Zabbix agent* em *Executar em*).

Attention:
Observe o uso de **sudo** - o usuário do Zabbix não tem permissões para reiniciar serviços do sistema por padrão. Veja abaixo dicas de como configurar o **sudo**.

Note:
A partir do agent Zabbix 7.0, comandos remotos também podem ser executados em um agent que está operando no modo ativo. O agent Zabbix - seja ativo ou passivo - deve ser executado no host remoto e executa os comandos em segundo plano.

Comandos remotos no agent Zabbix são executados sem timeout pela chave system.run[,nowait] e não são verificados quanto aos resultados da execução. No server Zabbix e no proxy Zabbix, comandos remotos são executados com timeout conforme definido no parâmetro TrapperTimeout do arquivo zabbix_server.conf ou zabbix_proxy.conf e são verificados quanto aos resultados da execução. Para informações adicionais, consulte *Timeout do script*.

Permissões de acesso

Certifique-se de que o usuário 'zabbix' tenha permissões de execução para os comandos configurados. Pode ser interessante usar o **sudo** para dar acesso a comandos privilegiados. Para configurar o acesso, execute como root:

```
visudo
```

Exemplos de linhas que podem ser usadas no arquivo *sudoers*:

```
# permite que o usuário 'zabbix' execute todos os comandos sem senha.
zabbix ALL=NOPASSWD: ALL

# permite que o usuário 'zabbix' reinicie o apache sem senha.
zabbix ALL=NOPASSWD: /etc/init.d/apache restart
```

Note:
Em alguns sistemas, o arquivo *sudoers* impedirá que usuários não locais executem comandos. Para alterar isso, comente a opção **requiretty** em */etc/sudoers*.

Comandos remotos com múltiplas interfaces

Se o sistema de destino tiver várias interfaces do tipo selecionado (agent Zabbix ou IPMI), os comandos remotos serão executados na interface padrão.

É possível executar comandos remotos via SSH e Telnet usando uma interface diferente da do agent Zabbix. A interface disponível para uso é selecionada na seguinte ordem:

- Interface padrão do agent Zabbix
- Interface padrão SNMP
- Interface padrão JMX
- Interface padrão IPMI

Comandos remotos IPMI

Para comandos remotos IPMI, a seguinte sintaxe deve ser usada:

```
<comando> [<valor>]
```

Onde

- <command> - um dos comandos IPMI sem espaços
- <valor> - 'on', 'off' ou qualquer inteiro sem sinal. <valor> é um parâmetro opcional.

Exemplos

Exemplos de scripts globais que podem ser usados como comandos remotos em operações de ação.

Exemplo 1

Reinício do Windows em determinada condição.

Para reiniciar automaticamente o Windows ao detectar um problema pelo Zabbix, defina o seguinte script:

Parâmetro do script	Valor
Escopo	'Operação de ação'
Tipo	'Script'
Comando	c:\windows\system32\shutdown.exe -r -f

Exemplo 2

Reinicie o host usando o controle IPMI.

Parâmetro do script	Valor
Escopo	'Operação de ação'
Tipo	'IPMI'
Comando	reset

Exemplo 3

Desligue o host usando o controle IPMI.

Parâmetro do script	Valor
Escopo	'Operação de ação'
Tipo	'IPMI'
Comando	desligar

3 Operações adicionais

Visão geral

Nesta seção você pode encontrar alguns detalhes de adicionais operações para eventos de descoberta/autoregistro.

Adicionando host

Hosts são adicionados durante o processo de descoberta, assim que um host é descoberto, em vez de no final do processo de descoberta.

> **Note:**
> Como a descoberta de rede pode levar algum tempo devido a muitos hosts/serviços indisponíveis, é aconselhável ter paciência e usar intervalos de IP razoáveis.

Ao adicionar um host, seu nome é decidido pela função padrão **gethostbyname**. Se o host puder ser resolvido, o nome resolvido será usado. Caso contrário, o endereço IP será usado. Além disso, se o endereço IPv6 precisar ser usado como nome do host, todos os ":" (dois-pontos) serão substituídos por "_" (sublinhados), pois dois-pontos não são permitidos em nomes de host.

> **Attention:**
> Se estiver realizando a descoberta por um proxy, atualmente a pesquisa do nome do host ainda ocorre no Zabbix server.

> **Attention:**
> Se já existir um host na configuração do Zabbix com o mesmo nome de um recém-descoberto, o Zabbix adicionará **_N** ao nome do host, onde **N** é um número crescente, começando por 2.

4 Usando macros em mensagens

Visão geral

Nos assuntos e textos das mensagens, você pode usar macros para um relatório de problemas mais eficiente.

Além de várias macros internas, macros de usuário e macros de expressão também são suportadas. Uma lista completa de macros suportadas pelo Zabbix está disponível.

Exemplos

Exemplos aqui ilustram como você pode usar macros em mensagens.

Exemplo 1

Assunto da mensagem:

```
Problem: {TRIGGER.NAME}
```

Quando você receber a mensagem, o assunto da mensagem será substituído por algo como:

```
Problem: Processor load is too high on Zabbix server
```

Exemplo 2

Mensagem:

```
Processor load is: {?last(/zabbix.zabbix.com/system.cpu.load[,avg1])}
```

Quando você receber a mensagem, a mensagem será substituída por algo como:

```
Processor load is: 1.45
```

Exemplo 3

Mensagem:

```
Último valor: {?last(/{HOST.HOST}/{ITEM.KEY})}
MÁX nos últimos 15 minutos: {?max(/{HOST.HOST}/{ITEM.KEY},15m)}
MÍN nos últimos 15 minutos: {?min(/{HOST.HOST}/{ITEM.KEY},15m)}
```

Quando você receber a mensagem, ela será substituída por algo como:

```
Último valor: 1.45
MÁX nos últimos 15 minutos: 2.33
MÍN nos últimos 15 minutos: 1.01
```

Exemplo 4

Mensagem:

```
http://<server_ip_or_name>/zabbix/tr_events.php?triggerid={TRIGGER.ID}&eventid={EVENT.ID}
```

Quando você receber a mensagem, ela conterá um link para a página de *Detalhes do Evento*, que fornece informações sobre o evento, seu trigger e uma lista dos últimos eventos gerados pelo mesmo trigger.

Exemplo 5

Informando sobre valores de vários hosts em uma expressão de trigger.

Mensagem:

```
Nome do problema: {TRIGGER.NAME}
Expressão do trigger: {TRIGGER.EXPRESSION}

1. Valor do item em {HOST.NAME1}: {ITEM.VALUE1} ({ITEM.NAME1})
2. Valor do item em {HOST.NAME2}: {ITEM.VALUE2} ({ITEM.NAME2})
```

Quando você receber a mensagem, a mensagem será substituída por algo como:

```
Nome do problema: A carga do processador está muito alta em um host local
Expressão do trigger: last(/Myhost/system.cpu.load[percpu,avg1])>5 or last(/Myotherhost/system.cpu.load[pe

1. Valor do item em Myhost: 0.83 (Carga do processador (média de 1 min por núcleo))
2. Valor do item em Myotherhost: 5.125 (Carga do processador (média de 1 min por núcleo))
```

Exemplo 6

Recebendo detalhes tanto do evento de problema quanto do evento de recuperação em uma mensagem de recuperação:

Mensagem:

```
Problema:

ID do evento: {EVENT.ID}
Valor do evento: {EVENT.VALUE}
Status do evento: {EVENT.STATUS}
Hora do evento: {EVENT.TIME}
Data do evento: {EVENT.DATE}
Idade do evento: {EVENT.AGE}
Reconhecimento do evento: {EVENT.ACK.STATUS}
Histórico de atualização do evento: {EVENT.UPDATE.HISTORY}

Recuperação:

ID do evento: {EVENT.RECOVERY.ID}
Valor do evento: {EVENT.RECOVERY.VALUE}
Status do evento: {EVENT.RECOVERY.STATUS}
Hora do evento: {EVENT.RECOVERY.TIME}
Data do evento: {EVENT.RECOVERY.DATE}
Dados operacionais: {EVENT.OPDATA}
```

Quando você receber a mensagem, as macros serão substituídas por algo como:

```
Problema:

ID do evento: 21874
Valor do evento: 1
Status do evento: PROBLEM
Hora do evento: 13:04:30
Data do evento: 2018.01.02
Idade do evento: 5m 0s
Reconhecimento do evento: Yes
Histórico de atualização do evento: 2018.01.02 13:05:51 "John Smith (Admin)"
Ações: acknowledged.

Recuperação:
```

```
ID do evento: 21896
Valor do evento: 0
Status do evento: OK
Hora do evento: 13:10:07
Data do evento: 2018.01.02
Dados operacionais: Current value is 0.83
```

3 Operações de recuperação

Visão geral

As operações de recuperação permitem que você seja notificado quando os problemas forem resolvidos.

Tanto mensagens quanto comandos remotos são suportados em operações de recuperação. Embora várias operações possam ser adicionadas, a escalonamento não é suportada - todas as operações são atribuídas a uma única etapa e, portanto, serão executadas simultaneamente.

Casos de uso

Alguns casos de uso para operações de recuperação são os seguintes:

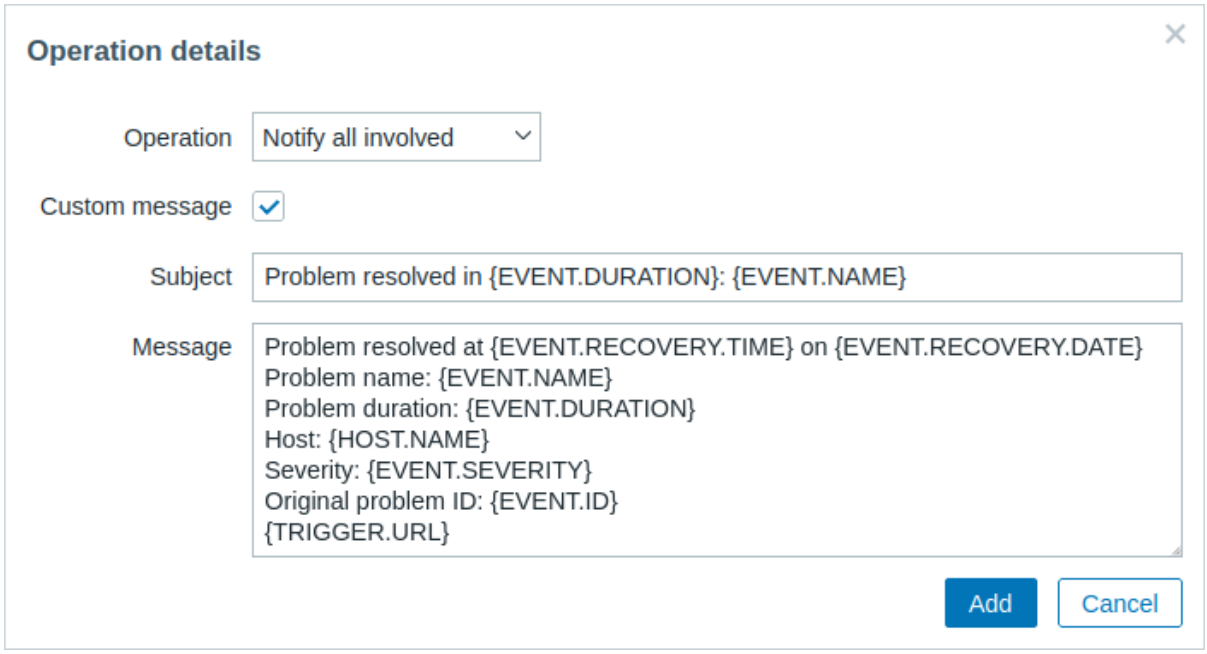
1. Notificar na recuperação todos os usuários que foram notificados sobre o problema:
   - Selecione *Notificar todos os envolvidos* como tipo de operação.
2. Ter várias operações na recuperação: enviar uma notificação e executar um comando remoto:
   - Adicione tipos de operação para enviar uma mensagem e executar um comando.
3. Abrir um ticket em um sistema externo de helpdesk/ticket e fechá-lo quando o problema for resolvido:
   - Crie um script externo que se comunique com o sistema de helpdesk.
   - Crie uma ação com uma operação que execute esse script e, assim, abra um ticket.
   - Tenha uma operação de recuperação que execute esse script com outros parâmetros e feche o ticket.
   - Use a macro {EVENT.ID} para referenciar o problema original.

Configurando uma operação de recuperação

Para configurar uma operação de recuperação, siga estas etapas:

1. Vá para a guia *Operações* no formulário de configuração de ação.
2. Para configurar uma nova operação de recuperação, clique em *Adicionar* na seção *Operações de recuperação*. Para editar uma operação existente, clique em *Editar* ao lado da operação.
3. Configure os detalhes da operação.

Detalhes da operação de recuperação



Três tipos de operação estão disponíveis para eventos de recuperação:

- **Enviar mensagem** - envia mensagem de recuperação para o usuário especificado;

- **Notificar todos os envolvidos** - envia mensagem de recuperação para todos os usuários que foram notificados no evento de problema;
- **<nome do comando remoto>** - executa um comando remoto. Os comandos estão disponíveis para execução se previamente definidos em scripts globais com *Operação de ação* selecionada como seu escopo.

Os parâmetros para cada tipo de operação são descritos abaixo. Todos os campos obrigatórios estão marcados com um asterisco vermelho. Quando terminar, clique em *Adicionar* para adicionar a operação à lista de *Operações de recuperação*.

Note:
Observe que, se o mesmo destinatário for definido em vários tipos de operação sem *Mensagem personalizada* especificada, notificações duplicadas não são enviadas.

Tipo de operação: enviar mensagem

Parâmetro	Descrição
Enviar para grupos de usuários	Selecione os grupos de usuários para os quais enviar a mensagem de recuperação. O grupo de usuários deve ter pelo menos permissão de "leitura" permissões para o host para ser notificado.
Enviar para usuários	Selecione os usuários para os quais enviar a mensagem de recuperação. O usuário deve ter pelo menos permissão de "leitura" permissões para o host para ser notificado.
Enviar para tipo de mídia	Enviar mensagem de recuperação padrão para todos os tipos de mídia disponíveis (configurados e habilitados) ou apenas para um específico. Desde o Zabbix 7.0.7, este parâmetro foi renomeado de *Enviar apenas para* para *Enviar para tipo de mídia*.
Mensagem personalizada	Se selecionado, uma mensagem personalizada pode ser definida.
Assunto	Assunto da mensagem personalizada. O assunto pode conter macros.
Mensagem	A mensagem personalizada. A mensagem pode conter macros.

Tipo de operação: comando remoto

Parâmetro	Descrição
Lista de destinos	Selecione os destinos para executar o comando: **Host atual** - o comando é executado no host do trigger que causou o evento de problema. Esta opção não funcionará se houver vários hosts no trigger. **Host** - selecione o(s) host(s) para executar o comando. **Grupo de hosts** - selecione o(s) grupo(s) de hosts para executar o comando. Especificar um grupo de hosts pai seleciona implicitamente todos os grupos de hosts aninhados. Assim, o comando remoto também será executado em hosts de grupos aninhados. Um comando em um host é executado apenas uma vez, mesmo que o host corresponda mais de uma vez (por exemplo, de vários grupos de hosts; individualmente e de um grupo de hosts). A lista de destinos é irrelevante se o comando for executado no Zabbix server. Selecionar mais destinos neste caso apenas resulta na execução do comando no server mais vezes. Observe que, para scripts globais, a seleção de destino também depende da configuração *Grupo de hosts* na configuração do script global.

Tipo de operação: notificar todos os envolvidos

Parâmetro		Descrição
Mensagem personalizada		Se selecionado, uma mensagem personalizada pode ser definida.
	Assunto	Assunto da mensagem personalizada. O assunto pode conter macros.
	Mensagem	A mensagem personalizada. A mensagem pode conter macros.

4 Operações de atualização

Visão geral

As operações de atualização estão disponíveis em ações com as seguintes fontes de evento:

- *Triggers* - quando problemas são atualizados por outros usuários, ou seja, comentados, reconhecidos, a severidade foi alterada, fechados (manualmente);
- *Serviços* - quando a severidade de um serviço foi alterada, mas o serviço ainda não foi recuperado.

Observe que os usuários não recebem notificações sobre suas próprias atualizações.

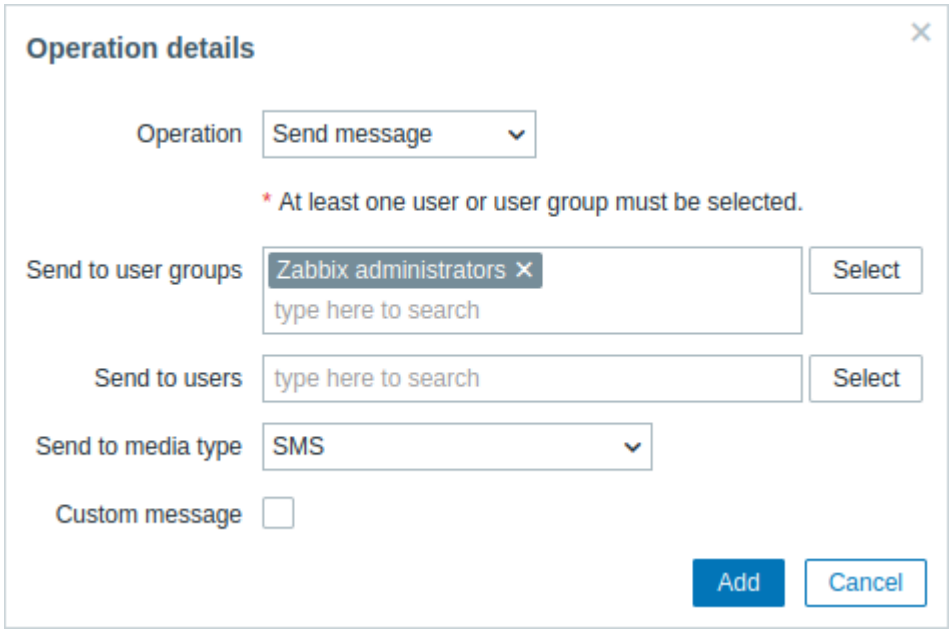
Tanto mensagens quanto comandos remotos são suportados em operações de atualização. Embora várias operações possam ser adicionadas, a escalonamento não é suportada - todas as operações são atribuídas a uma única etapa e, portanto, serão executadas simultaneamente.

Configurando uma operação de atualização

Para configurar uma operação de atualização, siga estas etapas:

1. Vá para a guia *Operações* no formulário de configuração de ação.
2. Para configurar uma nova operação de atualização, clique em *Adicionar* na seção *Operações de atualização*. Para editar uma operação existente, clique em *Editar* ao lado da operação.
3. Configure os detalhes da operação.

Detalhes da operação de atualização



As operações de atualização oferecem o mesmo conjunto de parâmetros das operações de recuperação.

5 Escalações

Visão geral

Com os escalonamentos, você pode criar cenários personalizados para envio notificações ou execução de comandos remotos.

Em termos práticos significa que:

- Os usuários podem ser informados sobre novos problemas imediatamente
- As notificações podem ser repetidas até que o problema seja resolvido
- O envio de uma notificação pode ser atrasado
- As notificações podem ser escaladas para outro grupo de usuários ”superior”
- Comandos remotos podem ser executados imediatamente ou quando um problema não é resolvido por um longo período

As ações são escalonadas com base na **etapa de escalonamento**. Cada passo tem um duração no tempo.

Você pode definir a duração padrão e uma duração personalizada de um passo individual. A duração mínima de uma etapa de escalonamento é de 60 segundos.

Você pode iniciar ações, como enviar notificações ou executar comandos, a partir de qualquer etapa. O primeiro passo é para ações imediatas. Se você quiser para atrasar uma ação, você pode atribuí-la a uma etapa posterior. Para cada passo, várias ações podem ser definidas.

O número de etapas de escalonamento não é limitado.

Os escalonamentos são definidos ao configurar um operação. As escalações são suportado apenas para operações problemáticas, não para recuperação.

Aspectos diversos do comportamento de escalação

Vamos considerar o que acontece em diferentes circunstâncias se uma ação contém várias etapas de escalonamento.

Situação	Comportamento
O host em questão entra em manutenção após o envio da notificação inicial do problema	Dependendo da configuração *Pausar operações para problemas suprimidos* na ação configuração, todos as etapas de escalonamento restantes são executadas com um atraso causado pelo período de manutenção ou sem atraso. Um período de manutenção não cancela as operações.
O período de tempo definido na condição de ação ***Período de tempo*** *termina após o envio da notificação inicial*	Todas as etapas de escalonamento restantes são executadas. A condição *Período de tempo* não pode interromper as operações; tem efeito em relação a quando as ações são iniciadas/não iniciadas, não as operações.
Um problema começa durante a manutenção e continua (não é resolvido) após o término da manutenção	Dependendo da configuração *Pausar operações para problemas suprimidos* na ação configuração, todas as etapas de escalonamento são executadas a partir do momento em que a manutenção termina ou imediatamente.
Um problema começa durante uma manutenção sem dados e continua (não é resolvido) após o término da manutenção	Ele deve aguardar o disparo do gatilho, antes que todas as etapas de escalonamento sejam executadas.
Diferentes escalações seguem em sucessão e sobreposição	A execução de cada nova escalação substitui a escalação anterior, mas para pelo menos uma etapa de escalação que é sempre executada na escalação anterior. Esse comportamento é relevante em ações em eventos que são criados com TODA avaliação de problema do gatilho.
Durante um escalonamento em andamento (como uma mensagem sendo enviada), com base em qualquer tipo de evento: - a ação é desativada Com base no evento acionador: - o acionador é desabilitado - o host ou item está desabilitado Com base no evento interno sobre gatilhos: - o gatilho está desabilitado Com base no evento interno sobre itens/regras de descoberta de baixo nível: - o item está desabilitado >- o host está desabilitado	A mensagem em andamento é enviada e mais uma mensagem no escalonamento é enviada. A mensagem de acompanhamento terá o texto de cancelamento no início do corpo da mensagem (*NOTA: Escalação cancelada*) nomeando o motivo (por exemplo, *NOTA: Escalação cancelada: ação '<Nome da ação>' desabilitada*). Desta forma o destinatário é informado que o escalonamento foi cancelado e não serão executadas mais etapas. Esta mensagem é enviada para todos que receberam as notificações anteriormente. O motivo do cancelamento também é registrado no arquivo de log do servidor (começando em Debug Level 3=Aviso). Observe que a mensagem *Escalação cancelada* também é enviado se as operações forem concluídas, mas as operações de recuperação estiverem configuradas e ainda não forem executadas.

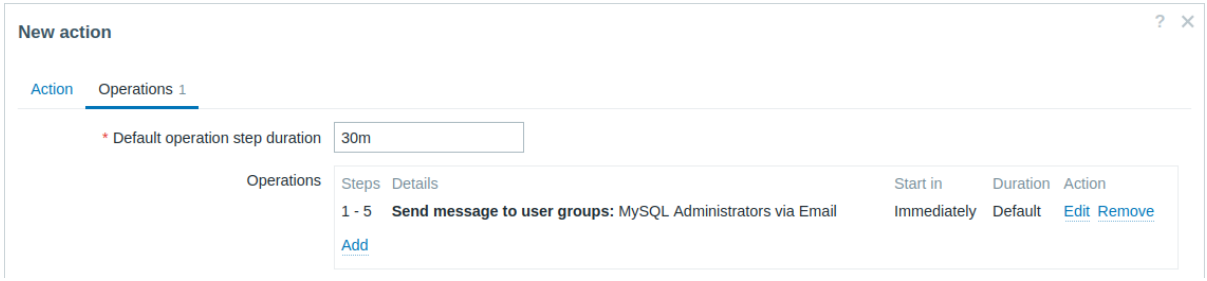
Situação	Comportamento
Durante uma escalação em andamento (como uma mensagem sendo enviada), a ação é excluída	Não são enviadas mais mensagens. As informações são registradas no arquivo de log do servidor (começando em Debug Level 3=Aviso), por exemplo: `escalação cancelada: ação id:334 excluído`

Exemplos de escalonamento

Exemplo 1

Enviando uma notificação repetida a cada 30 minutos (5 vezes no total) para um grupo "Administradores MySQL". Para configurar:

- Na aba *Operações*, defina a *Duração padrão do passo da operação* como "30m" (30 minutos).
- Defina os *Passos* da escalonamento de "1" a "5".
- Selecione o grupo "Administradores MySQL" como destinatário da mensagem.



As notificações serão enviadas às 0:00, 0:30, 1:00, 1:30, 2:00 horas após o início do problema (a menos, é claro, que o problema seja resolvido antes).

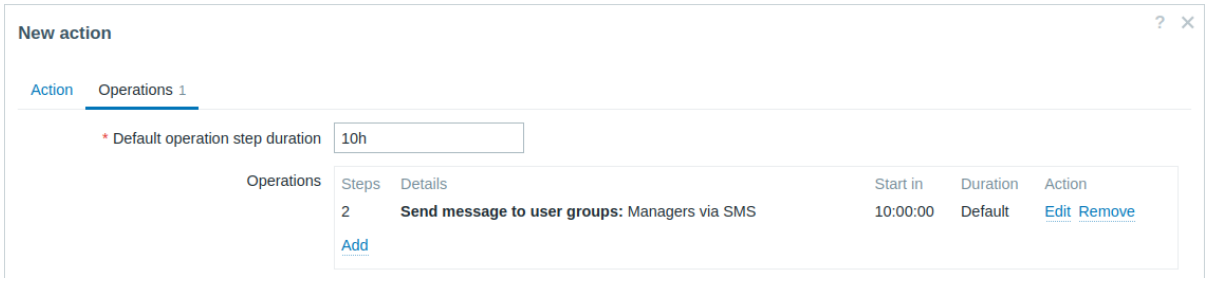
Se o problema for resolvido e uma mensagem de recuperação estiver configurada, ela será enviada para aqueles que receberam pelo menos uma mensagem de problema dentro deste cenário de escalonamento.

Note:
Se o trigger que gerou um escalonamento ativo for desabilitado, o Zabbix envia uma mensagem informativa sobre isso para todos que já receberam notificações.

Exemplo 2

Enviando uma notificação atrasada sobre um problema de longa duração. Para configurar:

- Na guia *Operações*, defina a *Duração padrão da etapa de operação* como "10h" (10 horas).
- Defina as *Etapas* da escalonamento de "2" a "2".



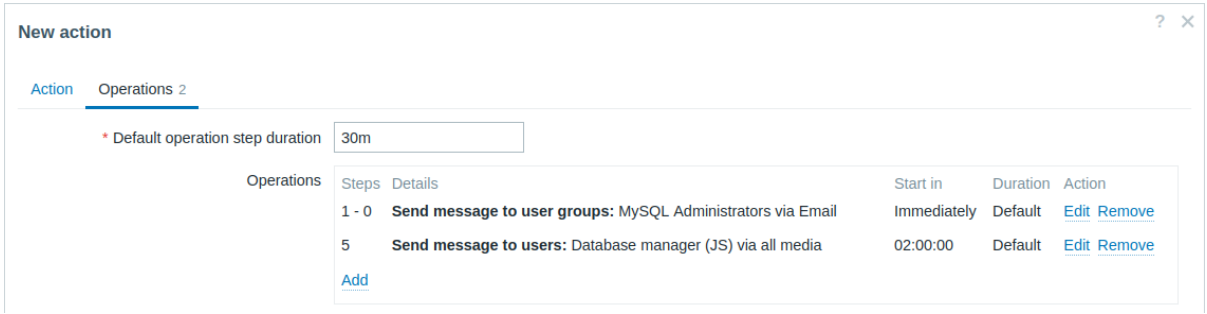
Uma notificação só será enviada na Etapa 2 do cenário de escalonamento, ou 10 horas após o início do problema.

Você pode personalizar o texto da mensagem para algo como "O problema tem mais de 10 horas".

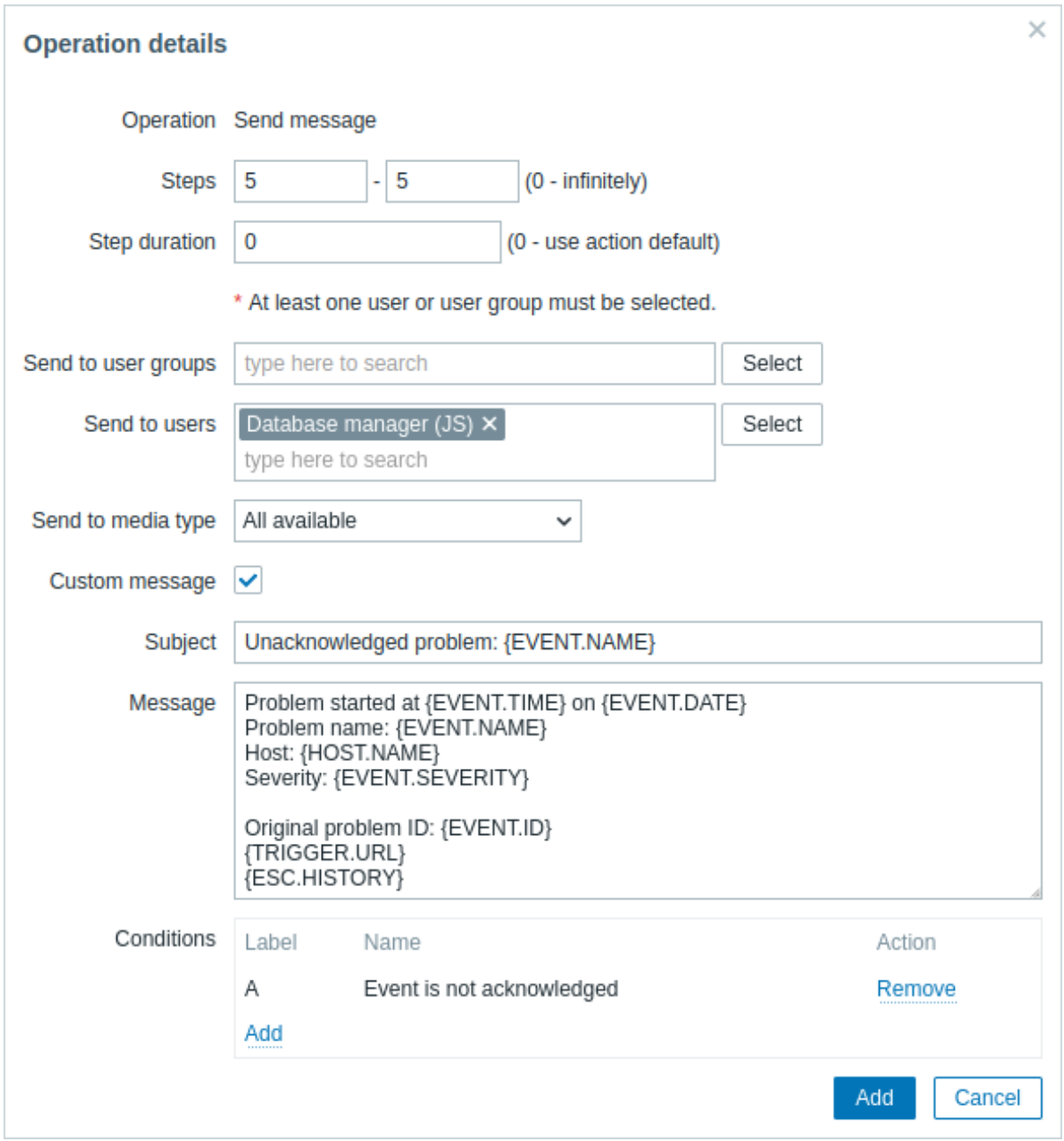
Exemplo 3

Escalando o problema para o Chefe.

No primeiro exemplo acima, configuramos o envio periódico de mensagens para os administradores do MySQL. Neste caso, os administradores receberão quatro mensagens antes que o problema seja escalado para o Gerente de Banco de Dados. Observe que o gerente receberá uma mensagem apenas se o problema ainda não tiver sido reconhecido, presumivelmente ninguém está trabalhando nele.



Detalhes da Operação 2:



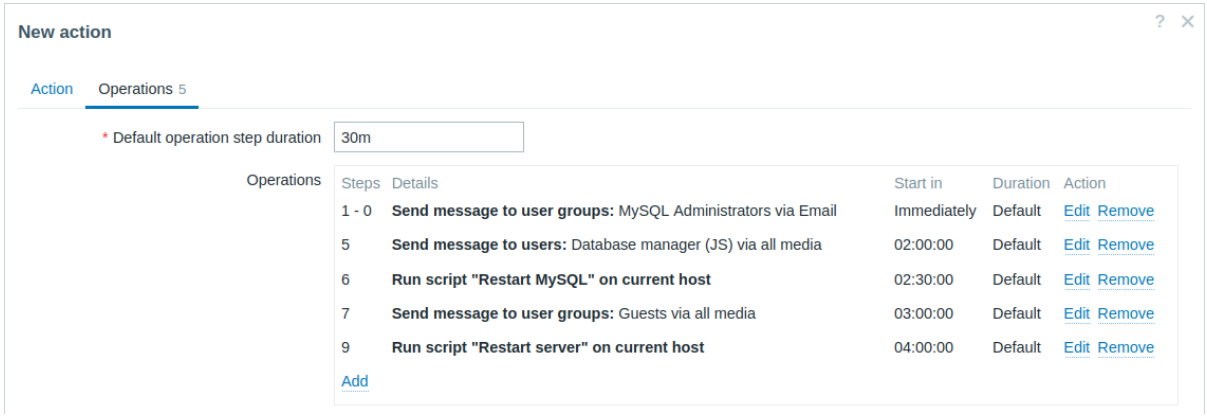
Observe o uso da macro {ESC.HISTORY} na mensagem personalizada. A macro conterá informações sobre todas as etapas executadas anteriormente nesta escalação, como notificações enviadas e comandos executados.

Exemplo 4

Um cenário mais complexo. Após várias mensagens para administradores do MySQL e escalação para o gerente, o Zabbix tentará reiniciar o MySQL base de dados. Isso acontecerá se o problema persistir por 2:30 horas e não foi reconhecido.

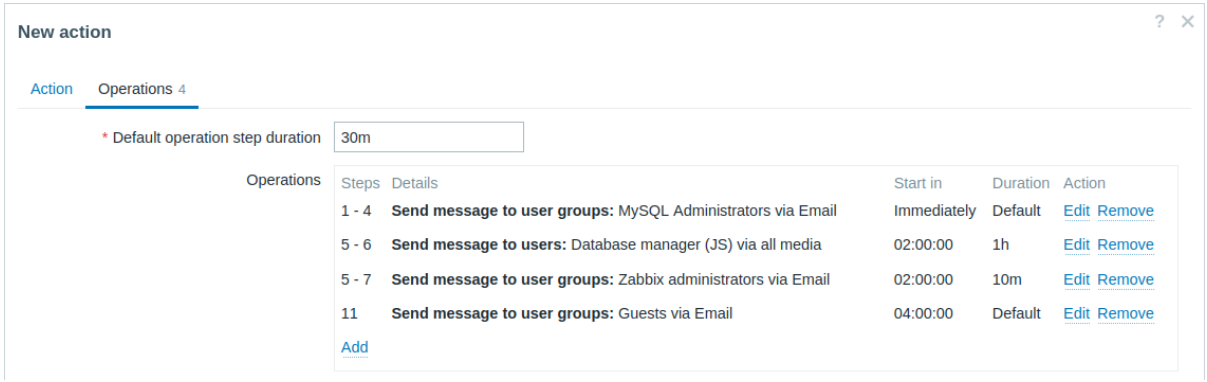
Se o problema persistir, após mais 30 minutos o Zabbix enviará um mensagem para todos os usuários convidados.

Se isso não ajudar, depois de mais uma hora o Zabbix irá reiniciar o servidor com o banco de dados MySQL (segundo comando remoto) usando comandos IPMI.



Exemplo 5

Uma escalada com várias operações que possuem intervalos de etapas sobrepostos e intervalos personalizados. A duração padrão da etapa da operação é de 30 minutos.



As notificações serão enviadas da seguinte forma:

- Para os administradores do MySQL às 0:00, 0:30, 1:00 e 1:30 após o início do problema.
- Para o gerente do banco de dados às 2:00 e 2:10 (a duração da etapa personalizada mais curta de 10 minutos definida na operação subsequente substitui a duração da etapa mais longa de 1 hora configurada aqui, conforme descrito em Detalhes da operação para *Duração da etapa* quando as etapas se sobrepõem).
- Para os administradores do Zabbix às 2:00, 2:10 e 2:20 após o início do problema (a duração da etapa personalizada de 10 minutos é aplicada).
- Para usuários convidados às 4:00 após o início do problema (a duração padrão da etapa de 30 minutos entra em vigor entre as etapas 8 e 11).

3 Recebendo notificação sobre itens não suportados

Visão geral

É possível receber notificações sobre items não suportados no Zabbix.

Isso faz parte do conceito de eventos internos no Zabbix, permitindo que os usuários sejam notificados nessas ocasiões. Eventos internos refletem uma mudança de estado:

- quando items passam de 'normal' para 'não suportado' (e voltam);
- quando triggers passam de 'normal' para 'desconhecido' (e voltam);
- quando regras de descoberta de baixo nível passam de 'normal' para 'não suportado' (e voltam).

Esta seção apresenta um passo a passo para **receber notificações** quando um item se torna não suportado.

Configuração

No geral, o processo de configuração da notificação deve parecer familiar para aqueles que configuraram alertas no Zabbix antes.

Passo 1

Configure algum meio, como e-mail, SMS ou script para usar nas notificações. Consulte as seções correspondentes do manual para realizar esta tarefa.

Attention:
Para notificar sobre eventos internos, a severidade padrão (’Não classificado’) é usada, portanto, deixe-a marcada ao configurar o meio de usuário se você quiser receber notificações para eventos internos.

Etapa 2

Vá para *Alertas* → *Ações* → *Ações internas*.

Clique em *Criar ação* no canto superior direito da página para abrir um formulário de configuração de ação.

Passo 3

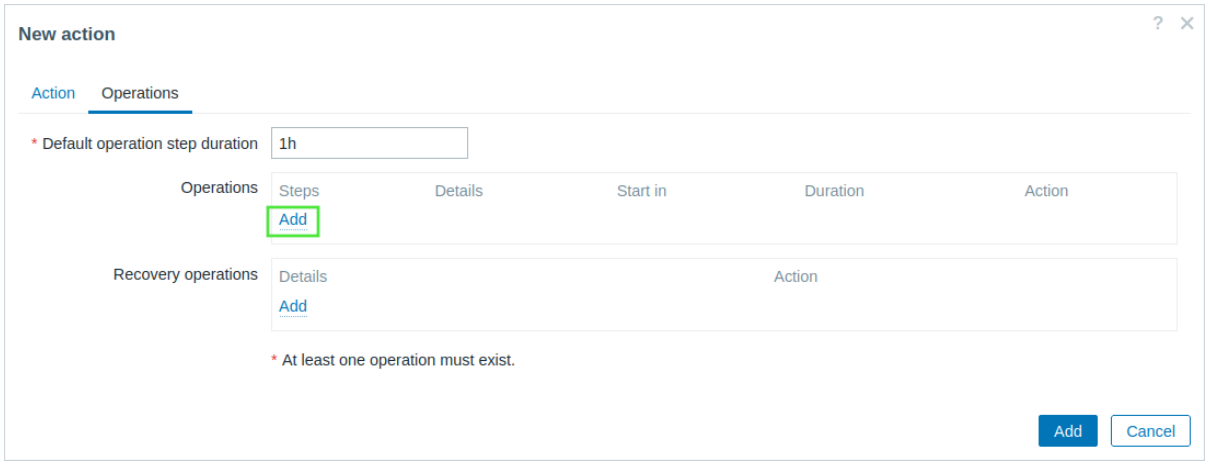
Na aba *Ação*, insira um nome para a ação. Em seguida, clique em *Adicionar* no bloco *Condições* para adicionar uma nova condição.

Na janela pop-up *Nova condição*, selecione ”Tipo de evento” como o tipo de condição e, em seguida, selecione ”Item em estado ’não suportado’” como o tipo de evento.

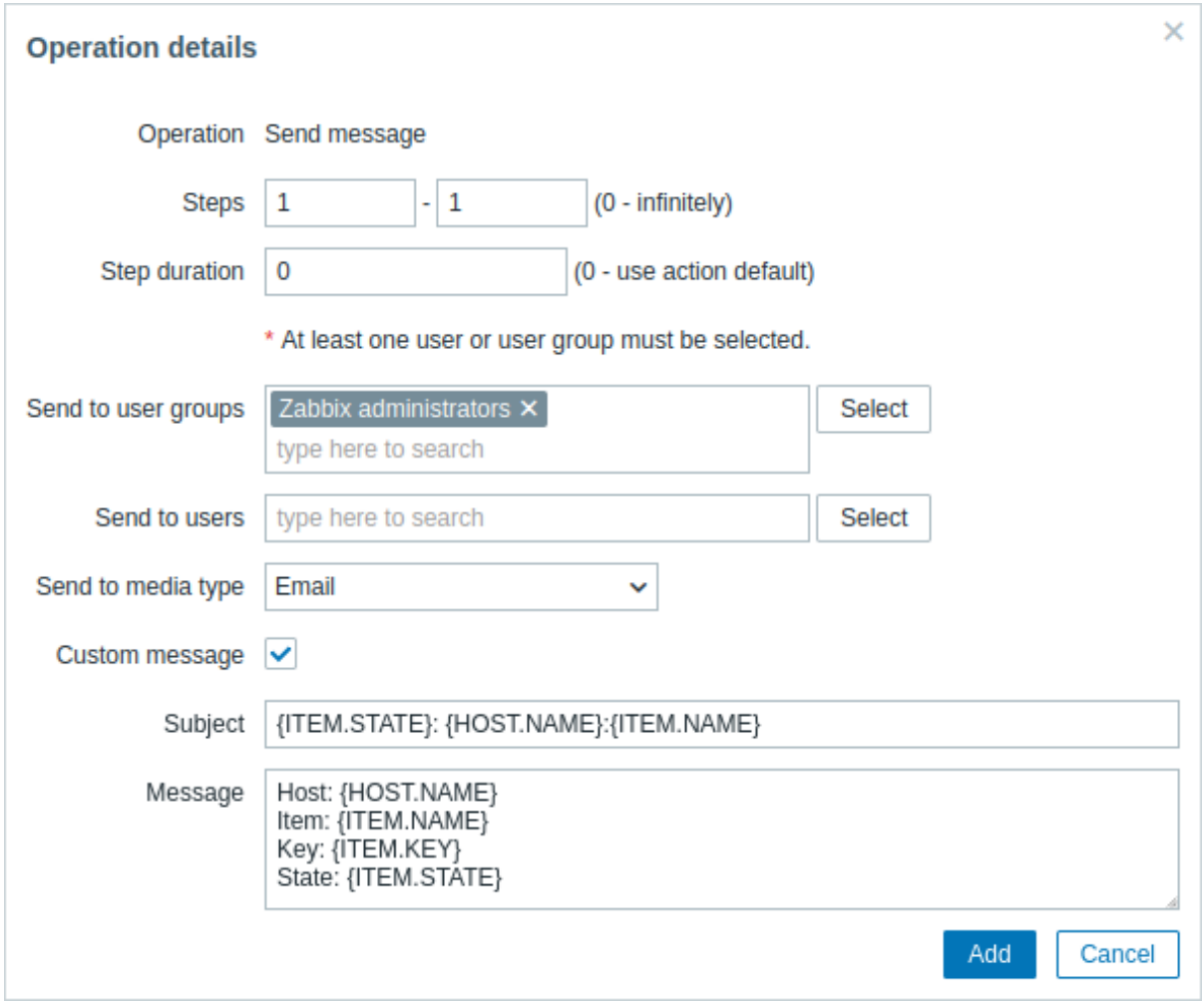
Não se esqueça de clicar em *Adicionar* para realmente listar a condição no bloco *Condições*.

Passo 4

Na aba *Operações*, clique em *Adicionar* no bloco *Operações* para adicionar uma nova operação.



Selecione alguns destinatários da mensagem (grupos de usuários/usuários) e o tipo de mídia (ou ”Todos”) a ser usado para a entrega. Marque a caixa de seleção *Mensagem personalizada* se desejar inserir o assunto/conteúdo personalizado da mensagem de problema.

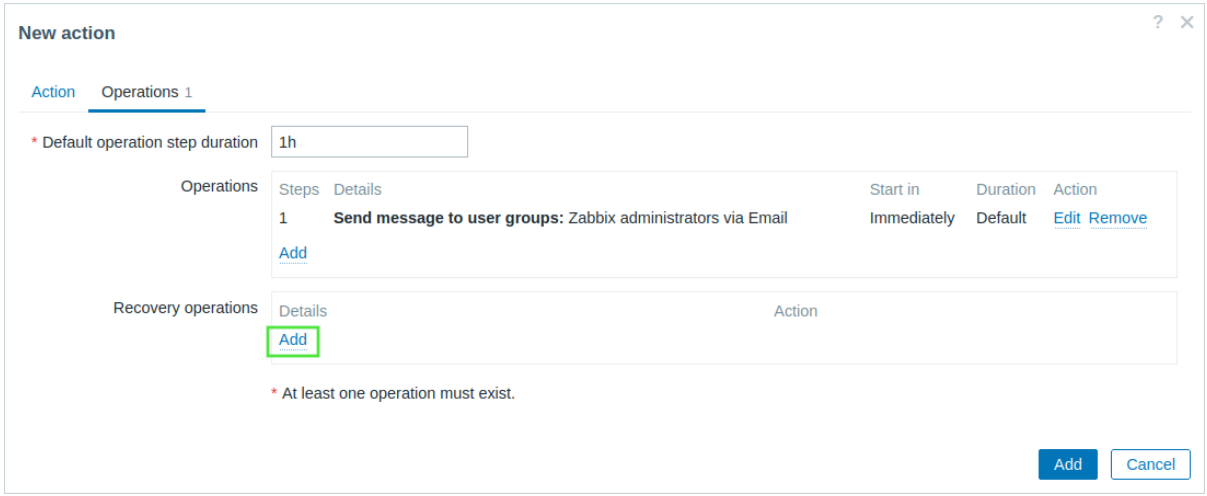


Clique em *Adicionar* para realmente listar a operação no bloco *Operações*.

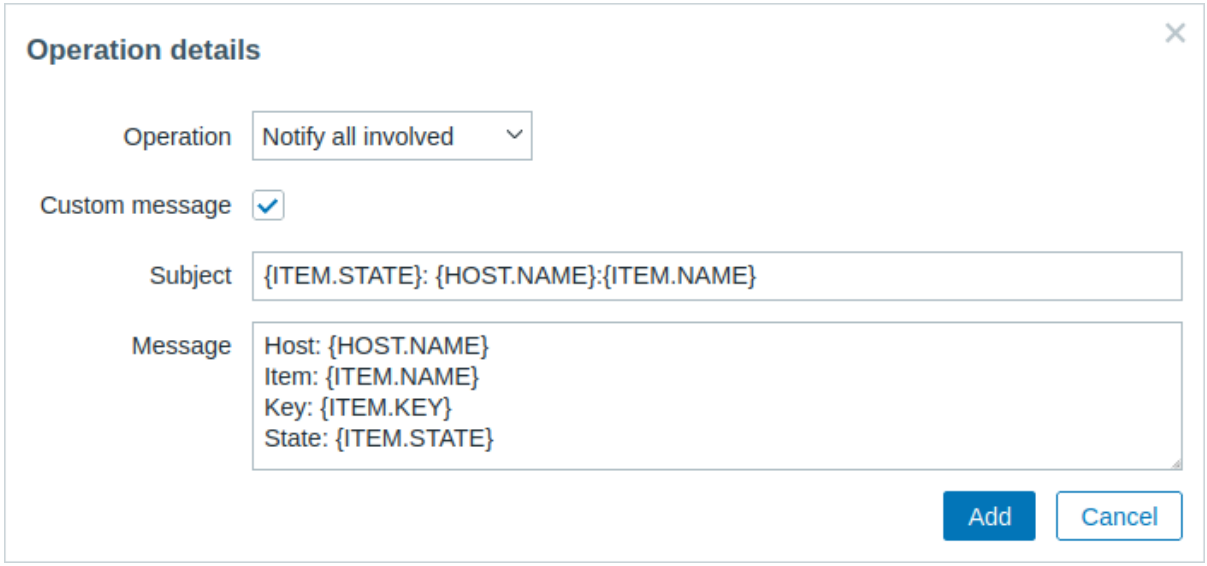
Se você deseja receber mais de uma notificação, defina a duração do passo da operação (intervalo entre as mensagens enviadas) e adicione outro passo.

Passo 5

O bloco *Operações de recuperação* permite configurar uma notificação de recuperação quando um item retorna ao estado normal. Clique em *Adicionar* no bloco *Operações de recuperação* para adicionar uma nova operação de recuperação.



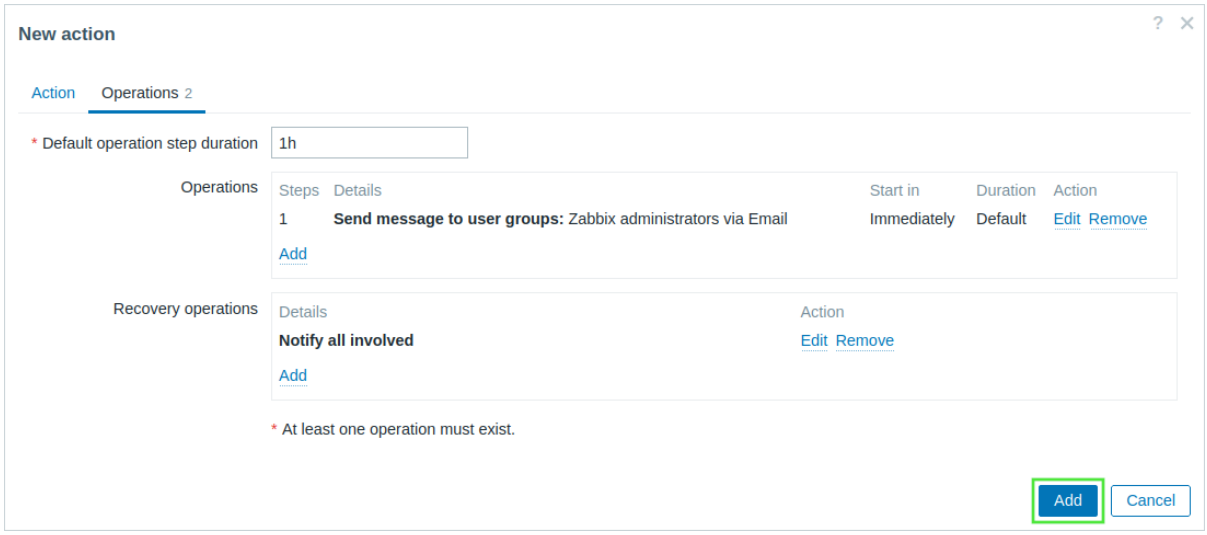
Selecione o tipo de operação ”Notificar todos os envolvidos”. Selecione a caixa de seleção *Mensagem personalizada* se desejar inserir o assunto/conteúdo personalizado da mensagem de recuperação.



Clique em *Adicionar* na janela pop-up *Detalhes da operação* para realmente listar a operação no bloco *Operações de recuperação*.

Passo 6

Quando terminar, clique no botão *Adicionar* na parte inferior do formulário.



E é isso, você terminou! Agora você pode esperar receber sua primeira notificação do Zabbix se algum item se tornar não suportado.

11 Macros

Visão geral

O Zabbix suporta vários macros internos que podem ser usados em várias situações. Esses macros são variáveis, identificadas por uma sintaxe específica:

`{MACRO}`

Os macros são resolvidos para um valor específico dependendo do contexto.

O uso eficaz de macros permite economizar tempo e tornar a configuração do Zabbix mais transparente.

Em um dos usos típicos, um macro pode ser usado em um template. Assim, um trigger em um template pode ser nomeado como "Processor load is too high on {HOST.NAME}". Quando o template é aplicado ao host, como o Zabbix server, o nome será resolvido para "Processor load is too high on Zabbix server" quando o trigger for exibido na seção Monitoramento.

Macros podem ser usados nos parâmetros da chave do item. Um macro pode ser usado apenas para uma parte do parâmetro, por exemplo, `item.key[server_{HOST.HOST}_local]`. Não é necessário colocar o parâmetro entre aspas duplas, pois o Zabbix cuidará de quaisquer símbolos especiais ambíguos, se presentes no macro resolvido.

Existem outros tipos de macros no Zabbix.

O Zabbix suporta os seguintes macros:

- {MACRO} - macro interno (veja a lista completa)
- {<macro>.<func>(<params>)} - funções de macro
- {$MACRO} - macro definido pelo usuário, opcionalmente com contexto
- {#MACRO} - macro para descoberta de baixo nível
- {?EXPRESSION} - macro de expressão

1 Funções de macro

Visão geral

As funções de macro oferecem a capacidade de personalizar os valores das macros (por exemplo, encurtar ou extrair substrings específicas), tornando-os mais fáceis de trabalhar.

A sintaxe de uma função de macro é:

```
{macro.func(params)}
```

onde

- **macro** - a macro a ser personalizada;
- **func** - a função a ser aplicada (veja funções suportadas);
- **params** - uma lista de parâmetros da função separados por vírgula, que devem ser **entre aspas duplas** se:
  - começar com um espaço ou aspas duplas;
  - conter parênteses de fechamento ou uma vírgula.

Por exemplo:

```
{{TIME}.fmttime(format,time_shift)}
{{ITEM.VALUE}.regsub(pattern, output)}
{{$USERMACRO}.regsub(pattern, output)}
{{#LLDMACRO}.regsub(pattern, output)}
```

As funções de macro são suportadas para:

- Macros internas
- Macros de usuário
- Macros de descoberta de baixo nível
- Macros de expressão

As funções de macro podem ser usadas em todos os locais que suportam as macros listadas. Isso se aplica, a menos que seja declarado explicitamente que apenas uma macro é esperada (por exemplo, ao configurar macros de host ou filtros de regras de descoberta de baixo nível).

Veja também: problemas conhecidos.

É suportada apenas uma função por macro; múltiplas funções de macro em cadeia não são suportadas.

> **Note:**
> Consulte exemplos de escape para casos em que funções de macro são usadas dentro de outros contextos (função, chave de item, outra macro, etc).

Funções suportadas

As funções são listadas sem informações adicionais. Clique na função para ver os detalhes completos.

Função	Descrição
btoa	Codificação do valor da macro no formato Base64.
fmtnum	Formatação de número para controlar o número de dígitos impressos após o ponto decimal.
fmttime	Formatação de tempo.
htmldecode	Decodificação do valor da macro da codificação HTML.
htmlencode	Codificação do valor da macro na codificação HTML.
iregsub	Extração de substring por correspondência de expressão regular (case-insensitive).
lowercase	Transformação dos caracteres do valor da macro em minúsculas.
regrepl	Substituição de caractere/substring no valor da macro.

Função	Descrição
regsub	Extração de substring por correspondência de expressão regular (case-sensitive).
tr	Transliteração dos caracteres do valor da macro.
uppercase	Transformação dos caracteres do valor da macro em maiúsculas.
urldecode	Decodificação do valor da macro da codificação de URL.
urlencode	Codificação do valor da macro na codificação de URL.

Detalhes da função

Os parâmetros opcionais da função são indicados por < >.

btoa

Codificando um valor de macro em codificação Base64. A codificação Base64 é um método para representar dados binários como texto, útil para armazenar e transmitir com segurança conteúdo binário por meio de protocolos baseados em texto.

Exemplo:

```
{{ITEM.VALUE}.btoa()} - codificará em Base64 um valor como "zabbix" em "emFiYml4"
```

Esta função é suportada desde o Zabbix 7.0.4.

fmtnum(digits)

Formatação de número para controlar o número de dígitos impressos após o ponto decimal.

Parâmetros:

- **digits** - o número de dígitos após o ponto decimal. Intervalo válido: 0-20. Zeros à direita não serão produzidos.

Exemplos:

```
{{ITEM.VALUE}.fmtnum(2)} - retornará "24.35" a partir de um valor recebido de "24.3483523"
{{ITEM.VALUE}.fmtnum(0)} - retornará "24" a partir de um valor recebido de "24.3483523"
```

fmttime(format,<time_shift>)

Formatação de hora.
 Observe que esta função pode ser usada com macros que resultam em um valor em um dos seguintes formatos de hora:

- `hh:mm:ss`
- `yyyy-mm-ddThh:mm:ss[tz]` (padrão ISO8601)
- timestamp UNIX

Parâmetros:

- **format** - string de formato obrigatória, compatível com a formatação da função `strftime`;
- **time_shift** (opcional) - o deslocamento de tempo aplicado ao horário antes da formatação; deve começar com `-<N><time_unit>` ou `+<N><time_unit>`, onde:
  - `N` - o número de unidades de tempo a adicionar ou subtrair;
  - `time_unit` - h (hora), d (dia), w (semana), M (mês) ou y (ano).

Comentários:

- O parâmetro `time_shift` suporta operações de tempo em múltiplos passos e pode incluir `/<time_unit>` para deslocar para o início da unidade de tempo (`/d` - meia-noite, `/w` - 1º dia da semana (segunda-feira), `/M` - 1º dia do mês, etc.). Exemplos: `-1w` - exatamente 7 dias atrás; `-1w/w` - segunda-feira da semana anterior; `-1w/w+1d` - terça-feira da semana anterior.
- As operações de tempo são calculadas da esquerda para a direita sem prioridades. Por exemplo, `-1M/d+1h/w` será interpretado como `((-1M/d)+1h)/w`.

Exemplos:

```
{{TIME}.fmttime(%B)} - retornará "October" de um valor recebido de "1633098961"
{{TIME}.fmttime(%d %B,-1M/M)} - retornará "1 September" de um valor recebido de "1633098961"
```

htmldecode

Decodificando um valor de macro de codificação HTML. Esta função é suportada desde o Zabbix 7.0.4.

Os seguintes caracteres são suportados:

Valor	Valor decodificado
&	&
<	<
>	>
"	"
'	'
'	'

Exemplo:

```
{{ITEM.VALUE}.htmldecode()} - irá decodificar um valor como "&lt;" em "<"
```

htmlencode

Codifica o valor de uma macro em codificação HTML. Esta função é suportada a partir do Zabbix 7.0.4.

Os seguintes caracteres são suportados:

Valor	Valor codificado
&	&
<	<
>	>
"	"
'	'

Exemplo:

```
{{ITEM.VALUE}.htmlencode()} - irá codificar em HTML um caractere como "<" em "&lt;"
```

iregsub(pattern,output)

Extração de substring por correspondência de expressão regular (case-insensitive).

Parâmetros:

- **pattern** - a expressão regular a ser correspondida;
- **output** - as opções de saída. Os placeholders **\1 - \9** são suportados para capturar grupos. **\0** retorna o texto correspondente.

Comentários:

- Se não houver correspondência para a expressão regular, a função retorna uma string vazia.
- Se o padrão da função for uma expressão regular incorreta, a macro será avaliada como 'UNKNOWN' (exceto para macros de descoberta de baixo nível, caso em que a função será ignorada e a macro permanecerá não resolvida).
- Referências a grupos de captura inexistentes na string de substituição são substituídas por uma string vazia.

Exemplo:

```
{{ITEM.VALUE}.iregsub("fail|error|fault|problem","ERROR")} - será resolvido para "ERROR" se as substrings
```

lowercase

Transformação de todos os caracteres do valor da macro em minúsculas. Funciona com conjuntos de caracteres de byte único (como ASCII) e não suporta UTF-8. Esta função é suportada desde o Zabbix 7.0.4.

Exemplo:

```
{{ITEM.VALUE}.lowercase()} - irá transformar um valor como "Zabbix SERVER" em "zabbix server" (minúsculo)
```

regrepl(pattern,replacement,<pattern2>,<replacement2>,...)

Substituição de caractere/substring no valor da macro.
 Observe que esta função só é suportada com a biblioteca **libpcre2**. Se o Zabbix server/proxy foi compilado com `libpcre`, esta função retornará UNKNOWN. Esta função é suportada a partir do Zabbix 7.0.4.

Parâmetros:

- **pattern** - a expressão regular a ser correspondida;
- **replacement** - a string de substituição. Os placeholders **\1 - \9** são suportados em strings de substituição para capturar grupos.

Comentários:

- Os padrões e substituições são processados sequencialmente, com cada par subsequente sendo aplicado de acordo com o resultado da substituição anterior;
- Referências a grupos de captura inexistentes na string de substituição são substituídas por uma string vazia.

Exemplos:

```
{{ITEM.VALUE}.regrepl("oldParam", "newParam")} - irá substituir "oldParam" por "newParam"
{{ITEM.VALUE}.regrepl("([^a-z])","\\\1")} - todos os caracteres que não são letras serão escapados com uma
{$THRESHOLD:"{{#FSNAME}.regrepl(\"\\$\",\"\")}"} - irá remover uma barra invertida no final (por exemplo,
{{ITEM.VALUE}.regrepl("_v1\.0", "_v2.0", "\(final\)", "")} - irá substituir várias partes no valor do item
```

regsub(pattern,output)

Extração de substring por correspondência de expressão regular (diferencia maiúsculas de minúsculas).

Parâmetros:

- **pattern** - a expressão regular a ser correspondida;
- **output** - as opções de saída. Os placeholders **\1 - \9** são suportados para capturar grupos. **\0** retorna o texto correspondente.

Comentários:

- Se não houver correspondência para a expressão regular, a função retorna uma string vazia.
- Se o padrão da função for uma expressão regular incorreta, a macro será avaliada como 'UNKNOWN' (exceto para macros de descoberta de baixo nível, caso em que a função será ignorada e a macro permanecerá não resolvida).
- Referências a grupos de captura inexistentes na string de substituição são substituídas por uma string vazia.

Exemplos:

```
{{ITEM.VALUE}.regsub("^([0-9]+)", Problem ID: \1)} - será resolvido como "Problem ID: 123" se um valor com
{{ITEM.VALUE}.regsub("fail|error|fault|problem","ERROR")} - será resolvido como "ERROR" se substrings "fai
```

Veja mais exemplos.

tr(caracteres,substituição)

Transliteração de caracteres do valor da macro. Esta função é suportada desde o Zabbix 7.0.4.

- **caracteres** - o conjunto de caracteres a serem substituídos;
- **substituição** - o conjunto de caracteres de substituição correspondentes posicionalmente.

Exemplos:

```
{{ITEM.VALUE}.tr(abc, xyz)} - substituirá todas as ocorrências de "a" por "x", "b" por "y", "c" por "z"
{{ITEM.VALUE}.tr(abc, xyzq)} - substituirá todas as ocorrências de "a" por "x", "b" por "y", "c" por "z" (
{{ITEM.VALUE}.tr(abcde, xyz)} - substituirá todas as ocorrências de "a" por "x", "b" por "y", "c" por "z",
{{ITEM.VALUE}.tr("\\\'", "\/\"")} - substituirá todas as ocorrências de barra invertida por barra, aspas s
{{ITEM.VALUE}.tr(A-Z,a-z)} - converterá todas as letras para minúsculas
{{ITEM.VALUE}.tr(0-9a-z,*)} - substituirá todos os números e letras minúsculas por "*"
{{ITEM.VALUE}.tr(0-9,ab)} - substituirá todas as ocorrências de 0 por "a" e substituirá todas as ocorrênci
{{ITEM.VALUE}.tr(0-9abcA-L,*)} - substituirá todos os números, caracteres "abc" e o intervalo A-L por "*"
{{ITEM.VALUE}.tr("\n","*")} - substituirá as ocorrências de fim de linha por *
{{ITEM.VALUE}.tr("e", "\n")} - substituirá todos os "e" por fim de linha
```

Para incluir caracteres literais:

```
barra invertida - deve ser escapada como \\
aspas simples - deve ser escapada como \'
aspas duplas - deve ser escapada como \"
```

Sequências de escape suportadas com barra invertida:

```
\\\\ => \\ - barra invertida dupla para barra invertida simples
\\a  => \a - alerta
\\b  => \b - backspace
\\f  => \f - form feed
\\n  => \n - nova linha
\\r  => \r - retorno
\\t  => \t - tabulação horizontal
\\v  => \v - tabulação vertical
```

uppercase

Transformação de todos os caracteres do valor da macro em maiúsculas. Funciona com conjuntos de caracteres de byte único (como ASCII) e não suporta UTF-8. Esta função é suportada desde o Zabbix 7.0.4.

Exemplo:

```
{{ITEM.VALUE}.uppercase()} - transformará um valor como "Zabbix Server" em "ZABBIX SERVER" (maiúsculas)
```

urldecode

Decodificando um valor de macro de uma codificação de URL. Esta função é suportada desde o Zabbix 7.0.4.

Exemplo:

```
{{ITEM.VALUE}.urldecode()} - irá decodificar um valor como "%2F" em "/"
```

urlencode

Codifica um valor de macro em codificação de URL. Esta função é suportada desde o Zabbix 7.0.4.

Exemplo:

```
{{ITEM.VALUE}.urlencode()} - irá codificar um caractere como "/" em "%2F"
```

Exemplos adicionais

A tabela abaixo mostra mais exemplos de uso de funções de macro.

{#IFALIAS} é uma macro LLD e só é definida em contextos de descoberta de baixo nível (regras de descoberta, protótipos e os items/triggers criados a partir deles). Usá-la fora do LLD deixará o token não expandido.

Função de macro	Valor recebido	Saída
{{ITEM.VALUE}.regsub(^[0-9]+, Problem)}	123Log line	Problem
{{ITEM.VALUE}.regsub("^([0-9]+)", "Problem")}	123 Log line	Problem
{{ITEM.VALUE}.regsub(".*", "Problem ID: \1")}	Log line	Problem ID:
{{ITEM.VALUE}.regsub("^(\w+).*?([0-9]+)", " Problem ID: \1_\2 ")}	MySQL crashed errno 123	Problem ID: MySQL_123
{{ITEM.VALUE}.regsub("([1-9]+", "Problem ID: \1")}	123 Log line	*UNKNOWN* (expressão regular inválida)
{{#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)} (expressão regular inválida)
{$MACRO:"{{#IFALIAS}.regsub(\"(.*)_([0-9]+\", \1)}"}	customername_1	{$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)}"} (expressão regular inválida)
"{$MACRO:"\{{#IFALIAS}.regsub("(.*)_([0-9]+", \1)}\"}"	customername_1	"{$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)}\"}" (expressão regular inválida)

Visualizando valores completos de items

Valores longos das macros {ITEM.VALUE} e {ITEM.LASTVALUE} resolvidas para items de texto/log são truncados para 20 caracteres em alguns locais do frontend. Para ver os valores completos dessas macros, você pode usar funções de macro, por exemplo:

```
{{ITEM.VALUE}.regsub("(.*)", \1)}
{{ITEM.LASTVALUE}.regsub("(.*)", \1)}
```

Veja também: detalhes das macros {ITEM.VALUE} e {ITEM.LASTVALUE} macro details.

2 Macros de usuário

Visão geral

Macros de usuário são suportadas no Zabbix para maior flexibilidade, além das macros suportadas por padrão.

Macros de usuário podem ser definidas em nível global, de template e de host. Essas macros têm uma sintaxe especial:

```
{$MACRO}
```

O Zabbix resolve as macros de acordo com a seguinte precedência:

1. macros em nível de host (verificadas primeiro)
2. macros definidas para templates de primeiro nível do host (ou seja, templates vinculados diretamente ao host), ordenadas pelo ID do template
3. macros definidas para templates de segundo nível do host, ordenadas pelo ID do template
4. macros definidas para templates de terceiro nível do host, ordenadas pelo ID do template, etc.
5. macros globais (verificadas por último)

Em outras palavras, se uma macro não existir para um host, o Zabbix tentará encontrá-la nos templates do host em profundidade crescente. Se ainda não for encontrada, uma macro global será usada, se existir.

Warning:
Se uma macro com o **mesmo nome** existir em vários templates vinculados do mesmo nível, a macro do template com o menor ID será usada. Portanto, ter macros com o mesmo nome em vários templates é um risco de configuração.

Se o Zabbix não conseguir encontrar uma macro, a macro não será resolvida.

Attention:
As macros (incluindo macros de usuário) permanecem não resolvidas na seção de Configuração (por exemplo, na lista de triggers) por design, para tornar a configuração complexa mais transparente.

Macros de usuário podem ser usadas em:

- nome do item
- parâmetro da chave do item
- intervalos de atualização do item e intervalos flexíveis
- nome e descrição do trigger
- parâmetros e constantes de expressão do trigger (veja exemplos)
- muitos outros locais - veja a lista completa

Casos de uso comuns de macros globais e de host

- use uma macro global em vários locais; depois altere o valor da macro e aplique as alterações de configuração a todos os locais com um clique
- aproveite os templates com atributos específicos do host: senhas, números de porta, nomes de arquivos, expressões regulares, etc.

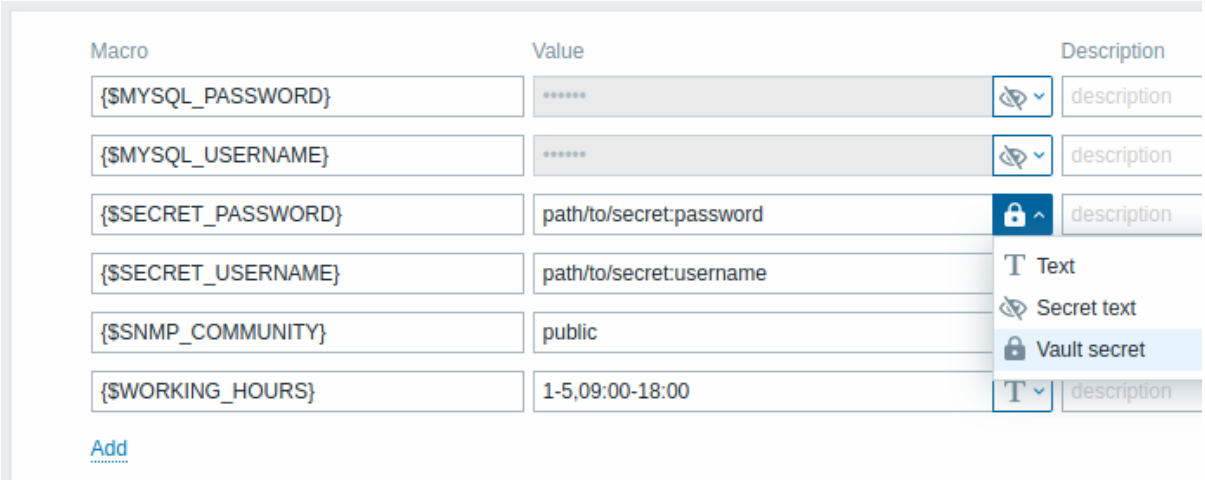
Note:
É recomendável usar macros de host em vez de macros globais, pois adicionar, atualizar ou excluir macros globais força a atualização incremental da configuração para todos os hosts. Para mais informações, consulte Verificações passivas e ativas do agent.

Configuração

Para definir macros de usuário, vá até o local correspondente no frontend:

- para macros globais, acesse *Administração* → *Macros*
- para macros em nível de host e template, abra as propriedades do host ou template e procure pela aba *Macros*

Uma macro de usuário possui os seguintes atributos:



Parâmetro	Descrição
Macro	Nome da macro. O nome deve estar entre chaves e começar com um cifrão. Exemplo: {$FRONTEND_URL}. Os seguintes caracteres são permitidos nos nomes das macros: **A-Z** (apenas maiúsculas), **0-9**, _, **.**
Valor	Valor da macro. Três tipos de valor são suportados: **Texto** (padrão) - valor em texto simples **Texto secreto** - o valor é mascarado com asteriscos **Segredo do Vault** - o valor contém um caminho/consulta para um segredo do vault. Para alterar o tipo de valor, clique no botão ao final do campo de valor. O comprimento máximo do valor de uma macro de usuário é de 2048 caracteres.
Descrição	Campo de texto usado para fornecer mais informações sobre esta macro.

Ao configurar macros de usuário, **observe** os seguintes comportamentos específicos de contexto:

- se macros de usuário forem usadas em items ou triggers de template, considere adicionar essas macros também ao template (mesmo que estejam definidas globalmente); dessa forma, macros do tipo *Texto* funcionarão conforme o esperado após exportar o template para XML e importar em outro sistema (valores de macros secretas não são exportados)
- se macros de usuário forem usadas em expressões de trigger, essas macros serão resolvidas apenas se fizerem referência a um parâmetro ou constante; elas NÃO serão resolvidas se fizerem referência a um host, chave de item, função, operador ou outra expressão de trigger (macros secretas não podem ser usadas em expressões de trigger)
- se macros de usuário forem usadas em um host que possui uma regra de descoberta de baixo nível com protótipos de host, os hosts descobertos herdarão todas as macros de usuário definidas nesse host

Exemplos

Exemplo 1

Uso de macro de nível de host na chave do item ”Status do daemon SSH”:

```
net.tcp.service[ssh,,{$SSH_PORT}]
```

Este item pode ser atribuído a vários hosts, desde que o valor de **{$SSH_PORT}** esteja definido nesses hosts.

Exemplo 2

Uso de macro de nível de host no trigger ”CPU load is too high”:

```
last(/ca_001/system.cpu.load[,avg1])>{$MAX_CPULOAD}
```

Esse trigger seria criado no template, não editado em hosts individuais.

> **Note:**
> Se você quiser usar a quantidade de valores como parâmetro da função (por exemplo, **max(/host/key,#3)**), inclua o símbolo de cerquilha na definição da macro assim: SOME_PERIOD => #3

Exemplo 3

Uso de duas macros no trigger ”CPU load is too high”:

```
min(/ca_001/system.cpu.load[,avg1],{$CPULOAD_PERIOD})>{$MAX_CPULOAD}
```

Observe que uma macro pode ser usada como parâmetro da função do trigger, neste exemplo a função **min()**.

Exemplo 4

Sincronize a condição de indisponibilidade do agent com o intervalo de atualização do item:

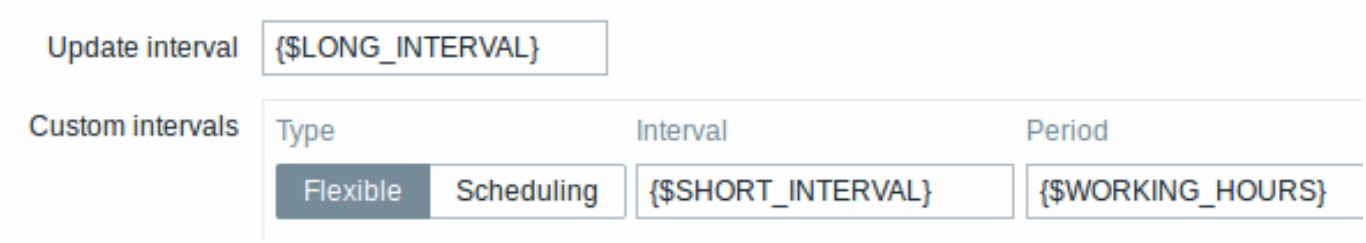
- defina a macro {$INTERVAL} e use-a no intervalo de atualização do item;
- use {$INTERVAL} como parâmetro do trigger de indisponibilidade do agent:

```
nodata(/ca_001/agent.ping,{$INTERVAL})=1
```

Exemplo 5

Centralize a configuração do horário de trabalho:

- crie uma macro global {$WORKING_HOURS} igual a 1-5,09:00-18:00;
- use-a no campo *Horário de trabalho* em *Administração → Geral → GUI*;
- use-a no campo *Quando ativo* em *Usuários → Usuários*, aba *Mídias* de um usuário;
- use-a para configurar uma coleta de item mais frequente durante o horário de trabalho:

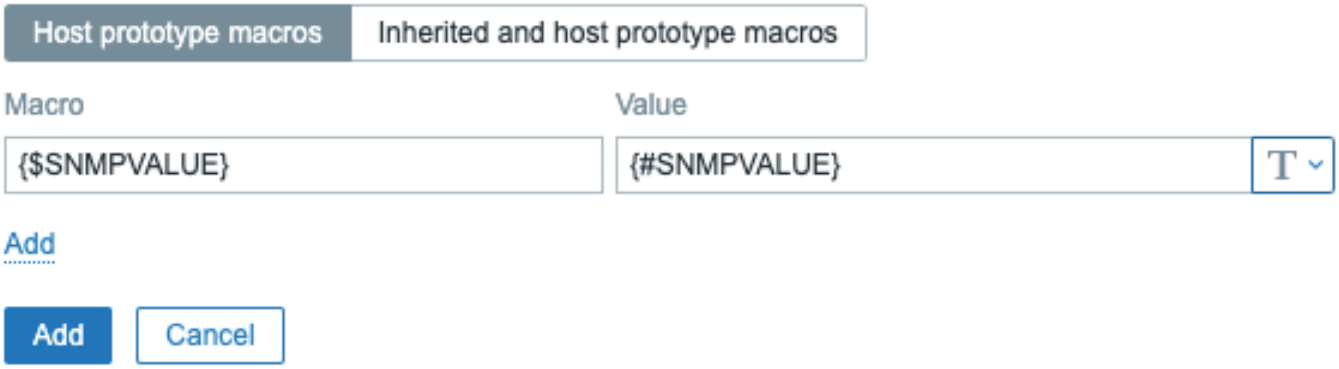


- use-a na condição de ação *Período de tempo*;
- ajuste o horário de trabalho em *Administração → Macros*, se necessário.

Exemplo 6

Use a macro de protótipo de host para configurar items para hosts descobertos:

- em um protótipo de host, defina a macro de usuário {$SNMPVALUE} com a macro {#SNMPVALUE} descoberta de baixo nível como valor:



- atribua o template *Generic SNMPv2* ao protótipo de host;
- use {$SNMPVALUE} no campo *SNMP OID* dos items do template *Generic SNMPv2*.

Contexto de macro de usuário

Veja macros de usuário com contexto.

3 Macros de usuário com contexto

Visão geral

Um contexto opcional pode ser usado em user macros, permitindo substituir o valor padrão com um específico do contexto.

O contexto é anexado ao nome da macro; a sintaxe depende se o contexto é um valor de texto estático:

```
{$MACRO:"texto estático"}
```

ou uma expressão regular:

```
{$MACRO:regex:"expressão regular"}
```

Observe que uma macro com contexto de expressão regular só pode ser definida em configuração de macro do usuário. Se o prefixo `regex:` for usado em outro lugar como contexto de macro do usuário, como em uma expressão de gatilho, será tratado como contexto estático.

A citação de contexto é opcional (veja também importante notas).

Exemplos de contexto de macro:

Exemplo	Descrição
{$LOW_SPACE_LIMIT}	Macro do usuário sem contexto.
{$LOW_SPACE_LIMIT:/tmp}	Macro do usuário com contexto (string estática).
{$LOW_SPACE_LIMIT:regex:"^/tmp$"}	Macro de usuário com contexto (expressão regular). O mesmo que {$LOW_SPACE_LIMIT:/tmp}.
{$LOW_SPACE_LIMIT:regex:"^/var/log/.*$"}	Macro de usuário com contexto (expressão regular). Corresponde a todas as strings prefixadas com /var/log/.

Casos de uso

As macros de usuário com contexto podem ser definidas para permitir limites de threshold mais flexíveis em expressões de trigger (com base nos valores obtidos pela descoberta de baixo nível). Por exemplo, você pode definir as seguintes macros:

- {$LOW_SPACE_LIMIT} = 10
- {$LOW_SPACE_LIMIT:/home} = 20
- {$LOW_SPACE_LIMIT:regex:”^/[a-z]+$”} = 30

Então, uma macro de descoberta de baixo nível pode ser usada como contexto de macro em um protótipo de trigger para descoberta de sistemas de arquivos montados:

```
last(/host/vfs.fs.size[{#FSNAME},pfree])<{$LOW_SPACE_LIMIT:"{#FSNAME}"}
```

Após a descoberta, diferentes limites de espaço livre serão aplicados nos triggers dependendo dos pontos de montagem ou tipos de sistema de arquivos descobertos. Eventos de problema serão gerados se:

- a pasta /home tiver menos de 20% de espaço livre em disco
- pastas que correspondam ao padrão regexp (como /etc, /tmp ou /var) tiverem menos de 30% de espaço livre em disco
- pastas que não correspondam ao padrão regexp e não sejam /home tiverem menos de 10% de espaço livre em disco

Anotações importantes

- Se existir mais de uma macro de usuário com contexto, o Zabbix tentará combinar as macros de contexto simples primeiro e depois as macros de contexto com expressões regulares em uma ordem indefinida.

Não crie macros de contexto diferentes que correspondam ao mesma string para evitar comportamento indefinido.

- Se uma macro com seu contexto não for encontrada no host, modelos vinculados ou globalmente, a macro sem contexto é pesquisada.
- Somente macros de descoberta de baixo nível são suportadas no contexto. Algum outras macros são ignoradas e tratadas como texto simples.

Tecnicamente, o contexto macro é especificado usando regras semelhantes a item key, exceto que o contexto da macro é não analisado como vários parâmetros se houver um caractere `,`:

- O contexto da macro deve ser citado com " se o contexto contiver um } caractere ou começa com um caractere ". Aspas entre aspas contexto deve ser escapado com o caractere \.
- O caractere \ em si não tem escape, o que significa que é impossível ter um contexto citado terminando com o caractere \ - a macro {$MACRO:”a:\b\c\”} é inválido.
- Os espaços iniciais no contexto são ignorados, os espaços finais são não:
  - Por exemplo, {$MACRO:A} é igual a {$MACRO: A}, mas não {$MACRO:A }.
- Todos os espaços antes das cotações iniciais e após as cotações finais são ignorado, mas todos os espaços entre aspas não são:
  - Macros {$MACRO:”A”}, {$MACRO: ”A”}, {$MACRO:”A” } e {$MACRO: ”A” } são iguais, mas as macros {$MACRO:”A”} e {$MACRO:” A ”} não são.

As macros a seguir são todas equivalentes, pois possuem a mesma contexto: {$MACRO:A}, {$MACRO: A} e {$MACRO:”A”}. Isso está em contraste com chaves de item, onde ’key[a]’, ’key[a]’ e ’key[”a”]’ são as mesmo semanticamente, mas diferente para fins de exclusividade.

4 Macros de usuário secretas

Visão geral

O Zabbix fornece duas opções para proteger informações confidenciais nos valores das macros de usuário:

- Texto secreto
- Segredo do Vault

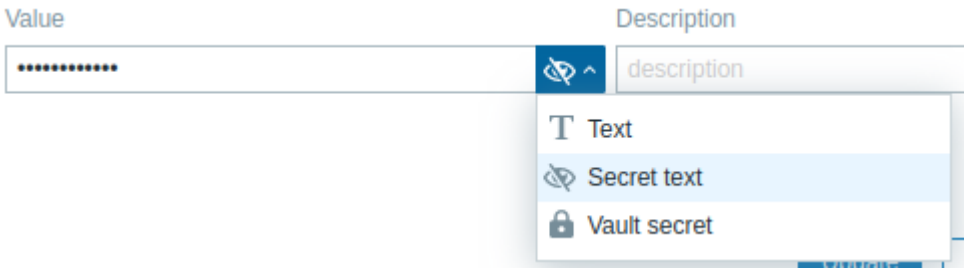
Note:
Embora o valor de uma macro secreta fique oculto, ele pode ser revelado por meio do uso em items. Por exemplo, em um script externo, uma instrução `echo` referenciando uma macro secreta pode ser usada para revelar o valor da macro no frontend, porque o Zabbix server tem acesso ao valor real da macro. Veja os locais onde os valores das macros secretas são desmascarados.

Macros secretas não podem ser usadas em expressões de trigger.

Texto secreto

Com as macros de texto secreto, o valor da macro é mascarado com asteriscos.

Para tornar o valor de uma macro secreto, clique no botão no final do campo *Valor* e selecione a opção *Texto secreto*:



Depois que a configuração for salva, não será mais possível visualizar o valor.

Para alterar o valor da macro, passe o mouse sobre o campo *Valor* e clique no botão *Definir novo valor* (aparece ao passar o mouse):

Ao clicar no botão *Definir novo valor* (ou alterar o tipo de valor da macro), o valor atual será apagado. Você pode restaurar o valor original clicando na seta no final do campo *Valor* (disponível apenas antes de salvar a nova configuração). Observe que restaurar o valor original não irá expô-lo.

Note:
URLs que contenham uma macro secreta não funcionarão, pois a macro nelas será resolvida como "******".

Segredo do Vault

Com as macros de segredo do Vault, o valor da macro é armazenado em um software externo de gerenciamento de segredos (vault).

Para configurar uma macro de segredo do Vault, clique no botão ao final do campo *Valor* e selecione a opção *Segredo do Vault*:

O valor da macro deve apontar para um segredo do vault. O formato de entrada depende do provedor do vault. Para exemplos de configuração específicos do provedor, consulte:

- HashiCorp
- CyberArk

Os valores das macros secretas do Vault são recuperados do vault pelo Zabbix server a cada atualização dos dados de configuração e, em seguida, armazenados no cache de configuração.

O Zabbix proxy recebe os valores das macros secretas do Vault do Zabbix server em cada sincronização de configuração e os armazena em seu próprio cache de configuração. O proxy nunca recupera os valores das macros diretamente do vault. Isso significa que um Zabbix proxy não pode iniciar a coleta de dados após uma reinicialização até que receba a atualização de configuração do Zabbix server.

Para atualizar manualmente os valores secretos do vault, use a opção `secrets_reload` do controle em tempo de execução.

A criptografia deve estar habilitada entre o Zabbix server e o proxy; caso contrário, uma mensagem de aviso do server será registrada.

Warning:
Se um valor de macro não puder ser recuperado com sucesso, o item correspondente que utiliza o valor ficará como não suportado.

Locais não mascarados

Esta lista fornece os locais dos parâmetros onde os valores das macros secretas não são mascarados.

Note:
A partir do Zabbix 7.0.13, os valores das macros secretas permanecerão mascarados nos locais abaixo se referenciados indiretamente. Por exemplo, {ITEM.KEY}, {ITEM.KEY<1-9>}, {LLDRULE.KEY} macros internas usadas em tipos de mídia (parâmetros de Script ou Webhook) serão resolvidas para chaves de item contendo macros secretas mascaradas, como `net.tcp.port[******,******]` em vez de `net.tcp.port[192.0.2.0,80]`.

Contexto	Parâmetro
Itens, protótipos de item, regras LLD	
Item	*Parâmetros da chave do item*
Protótipo de item	*Parâmetros da chave do protótipo de item*
Regra de descoberta de baixo nível	*Parâmetros da chave do item de descoberta*
Agente SNMP	*Comunidade SNMP*
	Nome do contexto (SNMPv3)
	Nome de segurança (SNMPv3)
	Frase secreta de autenticação (SNMPv3)
	Frase secreta de privacidade (SNMPv3)
Agente HTTP	*URL*
	Campos de consulta
	Corpo da requisição
	Cabeçalhos
	Nome de usuário
	Senha
	Senha da chave SSL
Script	*Parâmetros*
	Script
Navegador	*Parâmetros*
	Script
Monitor de banco de dados	*Consulta SQL*
Agente TELNET	*Script*
	Nome de usuário
	Senha
Agente SSH	*Script*
	Nome de usuário
	Senha
Verificação simples	*Nome de usuário*
	Senha
Agente JMX	*Nome de usuário*
	Senha

Contexto	Parâmetro
Pré-processamento de valor de item	
Etapa de pré-processamento JavaScript	*Script*
Cenários web	
Cenário web	*Valor da variável* *Valor do cabeçalho* *URL* *Valor do campo de consulta* *Valor do campo post* *Post bruto*
Autenticação do cenário web	*Usuário* *Senha* *Senha da chave SSL*
Conectores	
Conector	*URL* *Nome de usuário* *Senha* *Token* *Proxy HTTP* *Arquivo de certificado SSL* *Arquivo de chave SSL* *Senha da chave SSL*
Descoberta de rede	
SNMP	*Comunidade SNMP* *Nome do contexto* (SNMPv3) *Nome de segurança* (SNMPv3) *Frase secreta de autenticação* (SNMPv3) *Frase secreta de privacidade* (SNMPv3)
Scripts globais	
Webhook	*Script JavaScript* *Valor do parâmetro do script JavaScript*
Telnet	*Nome de usuário* *Senha*
SSH	*Nome de usuário* *Senha*
Script	*Script*
Tipos de mídia	
Script	*Parâmetros do script*
Webhook	*Parâmetros*
Gerenciamento IPMI	
Host	*Nome de usuário* *Senha*

5 Macros de descoberta de baixo nível

Visão geral

Existe um tipo de macro usada dentro da função de descoberta de baixo nível (LLD):

```
{#MACRO}
```

É uma macro que é usada em uma regra LLD e retorna valores reais do nome do sistema de arquivos, interface de rede, OID SNMP, etc.

Essas macros podem ser usadas para criar protótipos de entidades (protótipos de item, trigger, gráfico, host e grupo de hosts). Durante a descoberta de sistemas de arquivos, interfaces de rede, máquinas virtuais, etc., essas macros são substituídas por valores reais e são a base para a criação de entidades reais.

Algumas macros de descoberta de baixo nível já vêm ”pré-embaladas” com a função LLD no Zabbix - {#FSNAME}, {#FSTYPE}, {#IFNAME}, {#SNMPINDEX}, {#SNMPVALUE}. No entanto, não é obrigatório aderir a esses nomes ao criar uma regra de descoberta de baixo nível personalizada. Nesse caso, você pode usar qualquer outro nome de macro LLD e referenciar esse nome.

Tipos de dados suportados

Ao definir regras de descoberta personalizadas, os valores das propriedades retornados em objetos JSON para macros LLD devem ser um dos seguintes tipos primitivos:

- string
- number
- boolean

Arrays, objetos e valores nulos não são suportados. Qualquer macro LLD que se refira a esse valor permanecerá não expandida e aparecerá literalmente (por exemplo, `'{#MY_MACRO}'`) durante o pré-processamento e criação do item.

Locais suportados

As macros LLD podem ser usadas:

- no filtro da regra de descoberta de baixo nível
- para protótipos de item em
  - nome
  - parâmetros da chave
  - unidade
  - intervalo de atualização[1]
  - tempo limite[1]
  - período de armazenamento do histórico[1]
  - período de armazenamento das tendências[1]
  - etapas de pré-processamento do valor do item
  - OID SNMP
  - campo do sensor IPMI
  - expressão de item calculado/agregado, em:
    - constantes de expressão e parâmetros de função
    - parâmetros da chave do item
  - condições de filtro de item agregado (nome do grupo de hosts e nome da tag)
  - script SSH e script Telnet
  - consulta SQL de monitoramento de banco de dados
  - campo endpoint do item JMX
  - descrição
  - campo URL do agente HTTP
  - campo de parâmetros de consulta HTTP do agente HTTP
  - campo de corpo da requisição do agente HTTP
  - campo de códigos de status obrigatórios do agente HTTP
  - campo de chave e valor dos cabeçalhos do agente HTTP
  - campo de nome de usuário de autenticação HTTP do agente HTTP
  - campo de senha de autenticação HTTP do agente HTTP
  - campo de proxy HTTP do agente HTTP
  - campo de arquivo de certificado SSL HTTP do agente HTTP
  - campo de arquivo de chave SSL HTTP do agente HTTP
  - campo de senha da chave SSL HTTP do agente HTTP
  - tags
- para protótipos de trigger em
  - nome
  - dados operacionais
  - expressão (apenas em constantes e parâmetros de função)

    - URL
    - descrição
    - tags
- para protótipos de gráfico em
    - nome
- para protótipos de host em
    - nome
    - nome visível
    - campos de interface personalizada: IP, DNS, porta, comunidade SNMP v1/v2, nome de contexto SNMP v3, nome de segurança SNMP v3, senha de autenticação SNMP v3, senha de privacidade SNMP v3
    - nome do protótipo de grupo de hosts
    - valor da tag do host
    - valor da macro do host
    - (veja a lista completa)

Em todos esses locais, exceto no filtro da regra de descoberta de baixo nível, as macros LLD podem ser usadas dentro do contexto de macro de usuário estática.

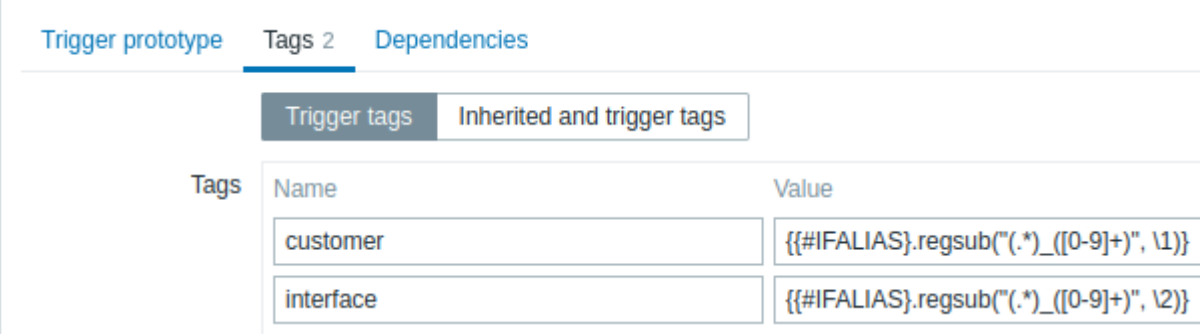
Usando funções de macro

Funções de macro são suportadas com macros de descoberta de baixo nível (exceto na regra de descoberta de baixo nível), permitindo extrair uma determinada parte do valor da macro usando uma expressão regular.

Por exemplo, você pode querer extrair o nome do cliente e o número da interface da seguinte macro LLD para fins de marcação de eventos:

```
{#IFALIAS}=customername_1
```

Para isso, a função de macro `regsub` pode ser usada com a macro no campo de valor da tag de evento de um protótipo de trigger:



Observe que vírgulas não são permitidas em parâmetros de chave de item não citados, portanto, o parâmetro contendo uma função de macro deve ser citado. O caractere barra invertida (\) deve ser usado para escapar aspas duplas dentro do parâmetro. Exemplo:

```
net.if.in["{{#IFALIAS}.regsub(\"(.*)_([0-9]+)\", \1)}",bytes]
```

Para mais informações sobre a sintaxe da função de macro, veja: Funções de macro

Funções de macro são suportadas em macros de descoberta de baixo nível desde o Zabbix 4.0.

Prototipagem de item sem parâmetros de chave

Quando os parâmetros de chave do item não são usados, coloque a macro LLD dentro dos colchetes do parâmetro item key `[...]` para que seja tratada como um parâmetro e substituída durante a descoberta:

```
v_[{#MACRO}]
```

Notas de rodapé

[1] Nos campos marcados com [1], uma única macro deve preencher todo o campo. Múltiplas macros em um campo ou macros misturadas com texto não são suportadas.

6 Macros de expressão

Visão geral

As macros de expressão permitem que você realize cálculos em campos.

Seu valor é calculado primeiro resolvendo quaisquer macros internas e, em seguida, avaliando a expressão resultante.

Sintaxe:

```
{?EXPRESSION}
```

EXPRESSION usa a mesma sintaxe e suporta as mesmas funções que as expressões de trigger.

Exemplo:

```
{?trendavg(/host/item1,1M:now/M)/trendavg(/host/item1,1M:now/M-1y)*100}
```

Para locais que suportam macros de expressão, procure por "{?EXPRESSION}" na tabela de macros suportadas.

Notas sobre o uso:

- Use as macros {HOST.HOST<1-9>} e {ITEM.KEY<1-9>} para referenciar hosts e items—por exemplo, {?max(/{HOST.HOST}/{ITEM.
- Em templates, use as macros {HOST.HOST<1-9>} ou omita o host completamente para o primeiro host—por exemplo, {?avg(//item1,1h)}—em vez dos nomes dos templates, pois os nomes dos templates não são substituídos pelos nomes dos hosts durante o vinculamento de templates.

Veja também Exemplos de expressão de trigger para um exemplo de uso de uma macro de expressão em um nome de evento.

12 Usuários e grupos de usuários

Visão geral

Todos os usuários do Zabbix acessam o aplicativo Zabbix através do a parte dianteira. Cada usuário recebe um nome de login exclusivo e uma senha.

Todas as senhas dos usuários são criptografadas e armazenadas no banco de dados Zabbix. Os usuários não podem usar seu ID de usuário e senha para fazer login diretamente no servidor UNIX, a menos que também tenham sido configurados de acordo com o UNIX. A comunicação entre o servidor web e o navegador do usuário pode ser protegidos por SSL.

Com um flexível permissão de usuário esquema você pode restringir e diferenciar os direitos de:

- acessar funções administrativas de frontend do Zabbix
- executar certas ações no frontend
- acessar hosts monitorados em grupos de hosts
- use métodos de API específicos

1 Configurando um usuário

Visão geral

A instalação inicial do Zabbix possui dois usuários predefinidos:

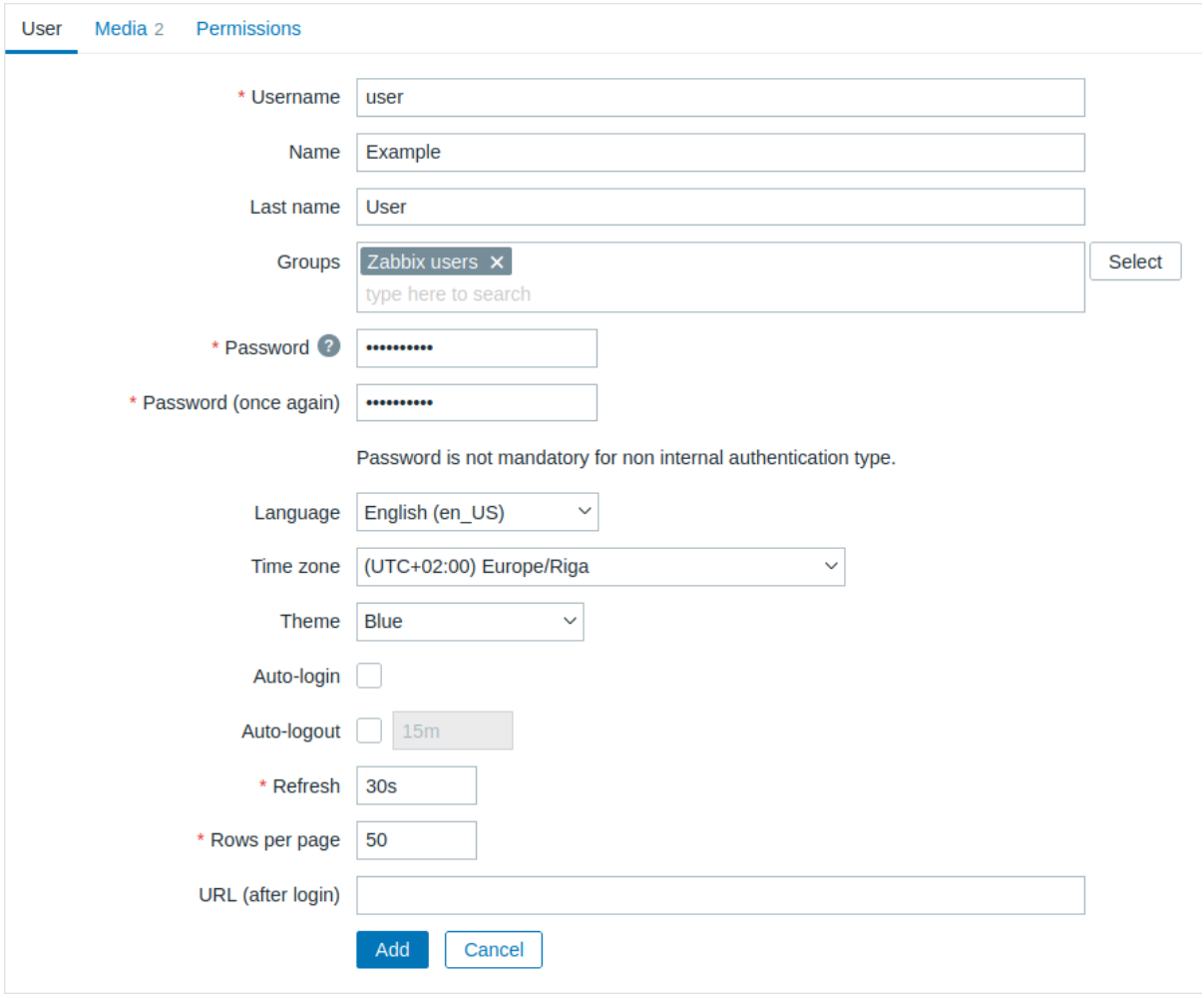
- *Admin* - um superusuário do Zabbix com permissões totais.
- *guest* - um usuário especial do Zabbix. O usuário 'guest' está desabilitado por padrão. Se você adicioná-lo ao grupo de usuários Guests, poderá fazer login com esse usuário e acessar as páginas de monitoramento no Zabbix. Observe que, por padrão, o 'guest' não possui permissões sobre objetos do Zabbix.

Para configurar um usuário:

- Vá para *Usuários* → *Usuários*.
- Clique em *Criar usuário* (ou em um nome de usuário para editar um usuário existente).
- Edite os atributos do usuário no formulário.

Atributos gerais

A aba *Usuário* contém os atributos gerais do usuário:



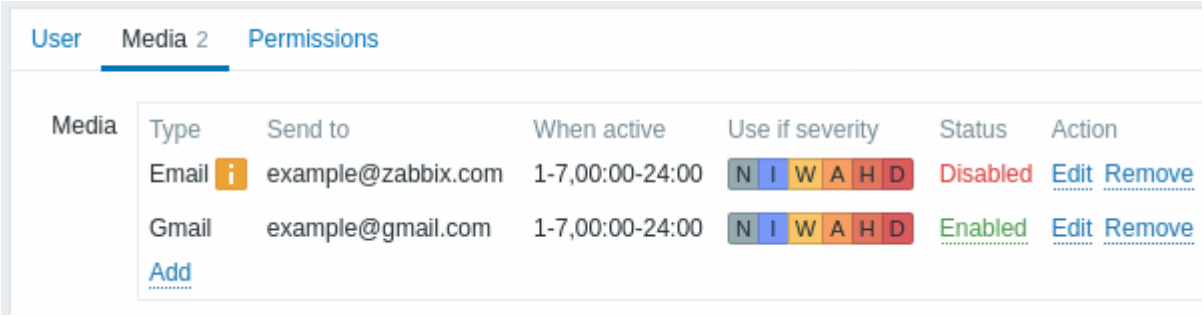
Todos os campos obrigatórios são marcados com um asterisco vermelho.

Parâmetro	Descrição
Nome de usuário	Nome de usuário único, usado como nome de login.
Nome	Primeiro nome do usuário (opcional). Se não estiver vazio, será visível nas informações de reconhecimento e nas informações do destinatário da notificação.
Sobrenome	Sobrenome do usuário (opcional). Se não estiver vazio, será visível nas informações de reconhecimento e nas informações do destinatário da notificação.
Grupos	Selecione os grupos de usuários aos quais o usuário pertence. Este campo é auto-completável, então ao começar a digitar o nome de um grupo de usuários será exibida uma lista suspensa com os grupos correspondentes. Role para baixo para selecionar. Alternativamente, clique em *Selecionar* para adicionar grupos. Clique em 'x' para remover o selecionado. A adesão a grupos de usuários determina a quais grupos de hosts e hosts o usuário terá acesso.
Senha	Dois campos para digitar a senha do usuário, ou um botão *Alterar senha* se o usuário já existir. Ao clicar no botão *Alterar senha*, são exibidos dois campos para digitar uma nova senha. Para o usuário com a *Função de superadministrador* alterando sua própria senha, ao clicar no botão *Alterar senha* é exibido um campo adicional para digitar a senha atual (antiga). Em uma alteração de senha bem-sucedida, o usuário para o qual a senha foi alterada será desconectado de todas as sessões ativas. Observe que a senha só pode ser alterada para usuários que utilizam a autenticação interna do Zabbix.
Idioma	Idioma do frontend do Zabbix. A extensão php gettext é necessária para que as traduções funcionem.
Fuso horário	Selecione o fuso horário para substituir o fuso horário global no nível do usuário ou selecione **Padrão do sistema** para usar as configurações globais de fuso horário.

Parâmetro	Descrição
Tema	Define como o frontend será exibido: **Padrão do sistema** - usa as configurações padrão do sistema **Azul** - tema azul padrão **Escuro** - tema escuro alternativo **Alto contraste claro** - tema claro com alto contraste **Alto contraste escuro** - tema escuro com alto contraste
Login automático	Marque esta caixa para que o Zabbix lembre do usuário e faça o login automaticamente por 30 dias. Ao fazer login com *Lembrar por 30 dias*: - O logout automático é redefinido (a sessão persiste por 30 dias). - O login automático é ativado para reautenticação contínua. Ao fazer login sem *Lembrar por 30 dias*: - O login automático é ativado para reautenticação contínua. - O logout automático permanece regido pela configuração padrão de tempo limite. Cookies do navegador são usados para isso.
Logout automático	Com esta caixa marcada, o usuário será desconectado automaticamente após a quantidade definida de segundos (mínimo de 90 segundos, máximo de 1 dia). Observe que esta configuração é substituída se *Lembrar por 30 dias* estiver ativado, pois a sessão é estendida por todo o período. Sufixos de tempo são suportados, por exemplo, 90s, 5m, 2h, 1d. Observe que esta opção não funcionará: * Se a opção de configuração global "Mostrar aviso se o servidor Zabbix estiver inativo" estiver ativada e o frontend do Zabbix for mantido aberto. * Quando as páginas do menu Monitoramento realizam atualizações de informações em segundo plano. * Se fizer login com a opção *Lembrar por 30 dias* marcada.
Atualização	Defina a taxa de atualização usada para gráficos, dados em texto simples, etc. Pode ser definido como 0 para desativar. Sufixos de tempo são suportados, por exemplo, 90s, 5m, 1h.
Linhas por página	Você pode determinar quantas linhas por página serão exibidas nas listas.
URL (após login)	Você pode fazer com que o Zabbix transfira o usuário para uma URL específica após o login bem-sucedido, por exemplo, para a página *Problemas*.

Mídia do usuário

A aba *Mídia* contém uma lista de todas as mídias definidas para o usuário. As mídias são usadas para o envio de notificações.



Clique em *Adicionar* para atribuir uma mídia ao usuário.

Se o tipo de mídia foi desabilitado:

- Um ícone de informação amarelo é exibido após o nome.
- *Desabilitado* é exibido na coluna Status.

Veja a seção Tipos de mídia para detalhes sobre como configurar a mídia do usuário.

Permissões

A aba *Permissões* contém informações sobre os seguintes elementos:

- Função do usuário (obrigatória para qualquer usuário recém-criado) que só pode ser alterada por um usuário do tipo *Super admin*.

Attention:
Usuários não podem ser criados sem uma função de usuário (exceto com a User API do Zabbix). Usuários criados anteriormente que não possuem uma função ainda podem ser editados sem atribuir uma função a eles. No entanto, uma vez que uma função é atribuída, ela só pode ser alterada, não removida.

 Observe que usuários sem função podem fazer login no Zabbix apenas usando autenticação LDAP ou SAML, desde que suas informações LDAP/SAML correspondam aos mapeamentos de grupos de usuários configurados no Zabbix.

- Tipo de usuário (*User*, *Admin*, *Super admin*) que é definido na configuração da função do usuário.
- Grupos de hosts e templates aos quais o usuário tem acesso.
  - Usuários do tipo *User* e *Admin*, por padrão, não têm acesso a nenhum grupo, template ou host. Para conceder esse acesso, os usuários devem ser incluídos em grupos de usuários configurados com permissões para as entidades relevantes.
- Direitos de acesso a seções e elementos do frontend do Zabbix, módulos e métodos da API.
  - Elementos com acesso permitido são exibidos em verde, enquanto aqueles com acesso negado - em cinza claro.
- Direitos para executar ações específicas.
  - Ações que o usuário tem permissão para executar são exibidas em verde, enquanto aquelas que são negadas - em cinza claro.

Consulte a página Permissões para mais detalhes.

2 Permissões

Visão geral

As permissões no Zabbix dependem do tipo de usuário, das funções de usuário personalizadas e do acesso aos hosts, que é especificado com base no grupo de usuários.

Tipos de usuário

As permissões no Zabbix dependem, principalmente, do tipo de usuário:

- *Usuário* - tem direitos de acesso limitados às seções do menu (veja abaixo) e nenhum acesso a quaisquer recursos por padrão. Quaisquer permissões para grupos de host ou template devem ser explicitamente atribuídas;
- *Admin* - tem direitos de acesso incompletos às seções do menu (veja abaixo). O usuário não tem acesso a nenhum grupo de host por padrão. Quaisquer permissões para grupos de host ou template devem ser explicitamente atribuídas;
- *Super admin* - tem acesso a todas as seções do menu. O usuário tem acesso de leitura e gravação a todos os grupos de host e template. As permissões não podem ser revogadas negando o acesso a grupos específicos.

Acesso ao menu

Note:
O acesso restrito a alguns elementos da interface do usuário apenas impede a abertura dessa página — não remove a capacidade de acessar os dados subjacentes em outras partes da interface.

A tabela a seguir ilustra o acesso às seções do menu do Zabbix por tipo de usuário:

Seção do menu		Usuário	Admin	Super admin
Dashboards		+	+	+
Monitoramento		+	+	+
	Problemas	+	+	+
	Hosts	+	+	+
	Últimos dados	+	+	+
	Mapas	+	+	+
	Descoberta		+	+
Serviços		+	+	+
	Serviços	+	+	+
	SLA		+	+
	Relatório de SLA	+	+	+
Inventário		+	+	+
	Visão geral	+	+	+
	Hosts	+	+	+
Relatórios		+	+	+
	Informações do sistema			+
	Relatórios agendados		+	+

Seção do menu		Usuário	Admin	Super admin
	Relatório de disponibilidade	+	+	+
	Top 100 triggers	+	+	+
	Log de auditoria			+
	Log de ações			+
	Notificações		+	+
Coleta de dados			+	+
	Grupos de template		+	+
	Grupos de host		+	+
	Templates		+	+
	Hosts		+	+
	Manutenção		+	+
	Correlação de eventos			+
	Descoberta		+	+
Alertas			+	+
	Ações de trigger		+	+
	Ações de serviço		+	+
	Ações de descoberta		+	+
	Ações de autoregistro		+	+
	Ações internas		+	+
	Tipos de mídia			+
	Scripts			+
Usuários				+
	Grupos de usuários			+
	Funções de usuário			+
	Usuários			+
	Tokens de API			+
	Autenticação			+
Administração				+
	Geral			+
	Log de auditoria			+
	Housekeeping			+
	Grupos de proxy			+
	Proxies			+
	Macros			+
	Fila			+

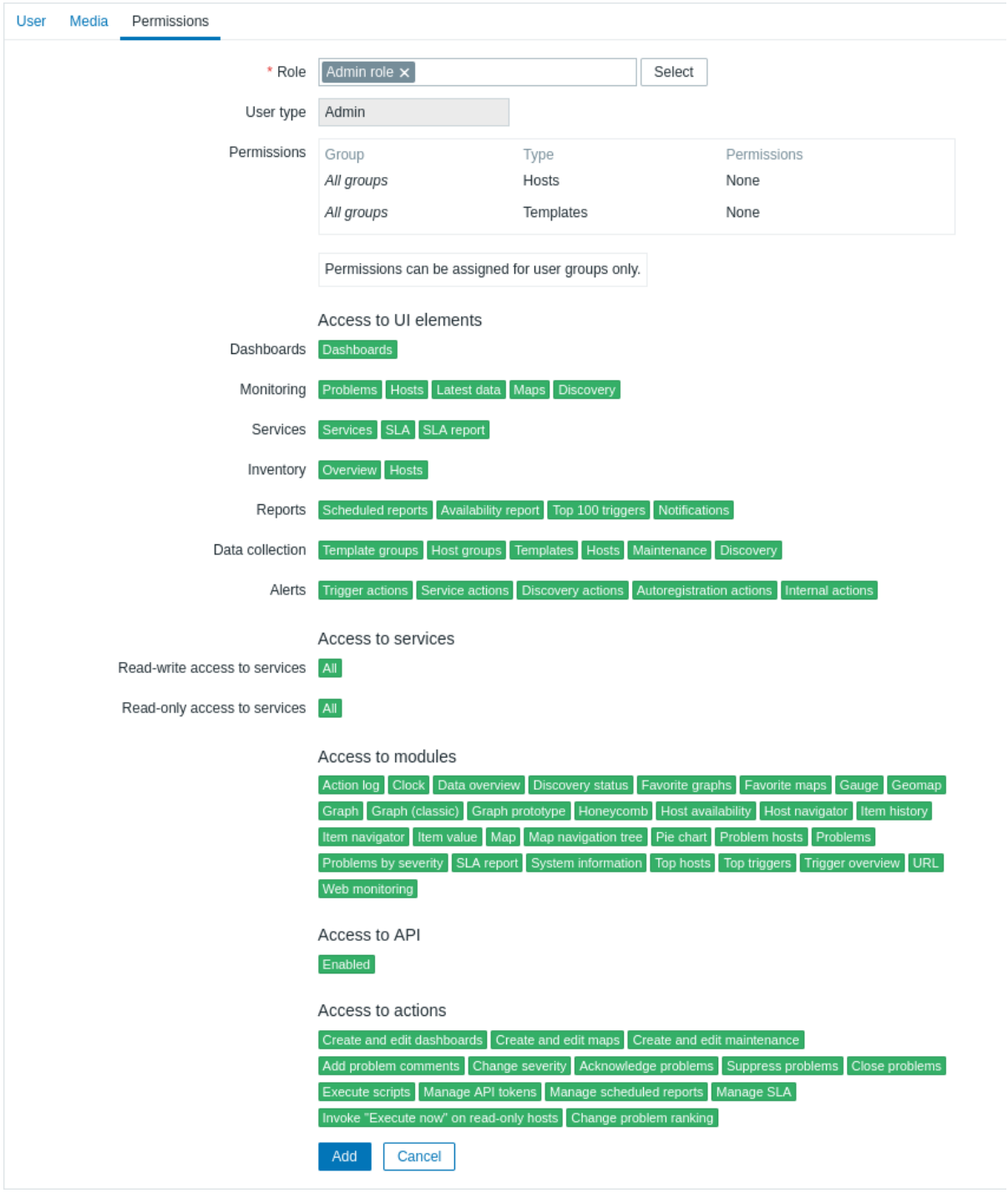
Funções de usuário

As funções de usuário permitem fazer ajustes personalizados nas permissões definidas pelo tipo de usuário. Embora nenhuma permissão possa ser adicionada (o que excederia as do tipo de usuário), algumas permissões podem ser revogadas.

Além disso, uma função de usuário determina o acesso não apenas às seções do menu, mas também a serviços, módulos, métodos de API e várias ações no frontend.

As Funções de usuário são configuradas na seção *Usuários* → *Funções de usuário* por usuários Super admin.

As funções de usuário são atribuídas aos usuários no formulário de configuração do usuário, na guia *Permissões*, por usuários Super admin.



Acesso a hosts

O acesso a qualquer dado de host e template no Zabbix é concedido a grupos de usuários apenas no nível do grupo de host/template.

Isso significa que um usuário individual não pode receber acesso diretamente a um host (ou grupo de hosts). Ele só pode receber acesso a um host fazendo parte de um grupo de usuários que tem acesso ao grupo de hosts que contém o host.

Da mesma forma, um usuário só pode receber acesso a um template fazendo parte de um grupo de usuários que tem acesso ao grupo de templates que contém o template.

3 Grupos de usuários

Visão geral

Os grupos de usuários permitem agrupar usuários tanto para fins organizacionais quanto para atribuição de permissões aos dados. As permissões para visualizar e configurar dados de grupos de hosts e grupos de templates são atribuídas a grupos de usuários, não a usuários individuais.

Muitas vezes, pode fazer sentido separar quais informações estão disponíveis para um grupo de usuários e quais para outro. Isso pode ser feito agrupando usuários e, em seguida, atribuindo permissões variadas a grupos de hosts e de templates.

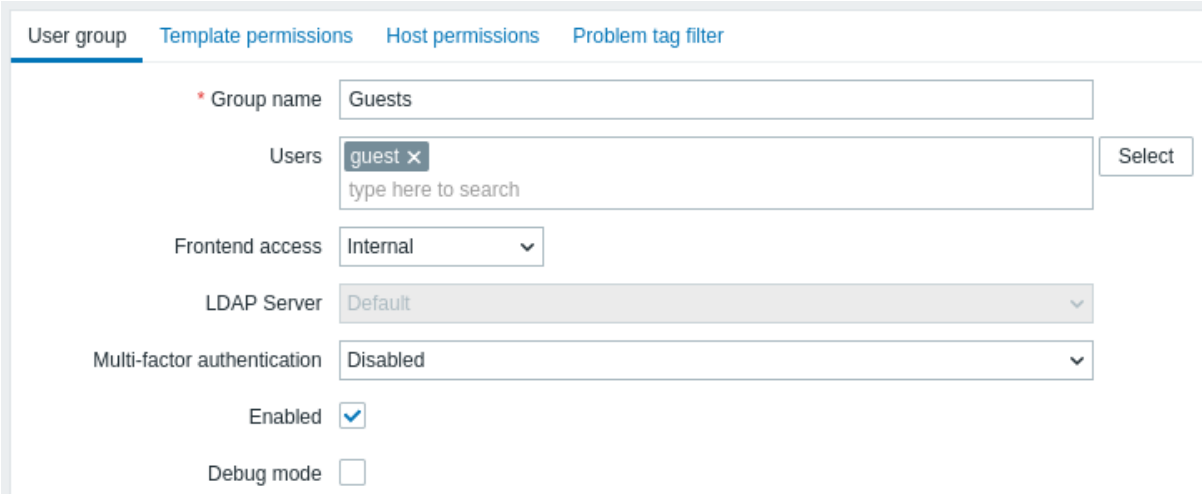
Um usuário pode pertencer a qualquer número de grupos.

Configuração

Para configurar um grupo de usuários:

- Vá para *Usuários* → *Grupos de usuários*
- Clique em *Criar grupo de usuários* (ou no nome do grupo para editar um grupo existente)
- Edite os atributos do grupo no formulário

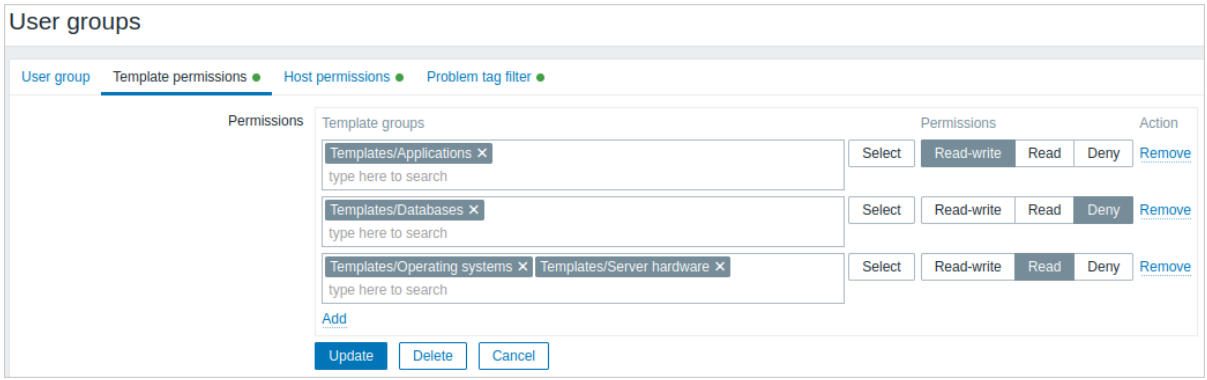
A aba **Grupo de usuários** contém os atributos gerais do grupo:



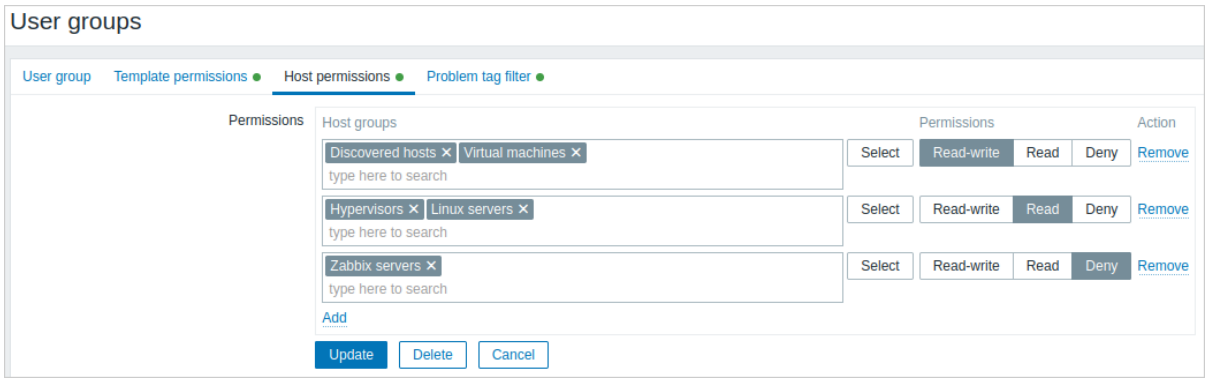
Todos os campos obrigatórios estão marcados com um asterisco vermelho.

Parâmetro	Descrição
Nome do grupo	Nome único do grupo.
Usuários	Para adicionar usuários ao grupo, comece a digitar o nome de um usuário existente. Quando o menu suspenso com nomes de usuários correspondentes aparecer, role para baixo para selecionar. Alternativamente, você pode clicar no botão *Selecionar* para selecionar usuários em um pop-up.
Acesso ao frontend	Como os usuários do grupo são autenticados. **Padrão do sistema** - usa o método de autenticação padrão (definido globalmente) **Interno** - usa a autenticação interna do Zabbix (mesmo que a autenticação LDAP seja usada globalmente). Ignorado se a autenticação HTTP for o padrão global. **LDAP** - usa autenticação LDAP (mesmo que a autenticação interna seja usada globalmente). Ignorado se a autenticação HTTP for o padrão global. **Desabilitado** - o acesso ao frontend do Zabbix é proibido para este grupo
Servidor LDAP	Selecione qual servidor LDAP usar para autenticar o usuário. Este campo é habilitado apenas se *Acesso ao frontend* estiver definido como LDAP ou Padrão do sistema.
Autenticação multifator	Selecione qual método de autenticação multifator usar para autenticar o usuário: **Padrão** - usa o método definido como padrão na configuração MFA; esta opção é selecionada por padrão para novos grupos de usuários se o MFA estiver habilitado; **<Nome do método>** - usa o método selecionado (por exemplo, ”Zabbix TOTP”); **Desabilitado** - o MFA está desabilitado para este grupo; esta opção é selecionada por padrão para novos grupos de usuários se o MFA estiver desabilitado. Observe que, se um usuário pertencer a vários grupos de usuários com MFA habilitado (ou pelo menos um grupo tiver o MFA habilitado), as seguintes regras de autenticação se aplicam: se algum grupo usar o método MFA ”Padrão”, ele autenticará o usuário; caso contrário, o primeiro método (ordenado alfabeticamente) será usado para autenticação.
Habilitado	Status do grupo de usuários e membros do grupo. *Marcado* - grupo de usuários e usuários estão habilitados *Desmarcado* - grupo de usuários e usuários estão desabilitados
Modo de depuração	Marque esta caixa para ativar o modo de depuração para os usuários.

A aba **Permissões de template** permite especificar o acesso do grupo de usuários aos dados do grupo de template (e, portanto, do template):



A aba **Permissões de host** permite especificar o acesso do grupo de usuários aos dados do grupo de host (e, portanto, do host):



Clique em Add para escolher os grupos de template/host (seja um grupo pai ou um grupo aninhado) e atribuir permissões a eles. Comece a digitar o nome do grupo (um menu suspenso de grupos correspondentes aparecerá) ou clique em *Selecionar* para abrir uma janela pop-up listando todos os grupos.

Em seguida, use os botões de opção para atribuir permissões aos grupos escolhidos. As permissões possíveis são as seguintes:

- **Leitura-escrita** - acesso de leitura e escrita a um grupo;
- **Leitura** - acesso somente leitura a um grupo;
- **Negar** - acesso a um grupo negado.

Se o mesmo grupo de template/host for adicionado em várias linhas com permissões diferentes definidas, a permissão mais restritiva será aplicada.

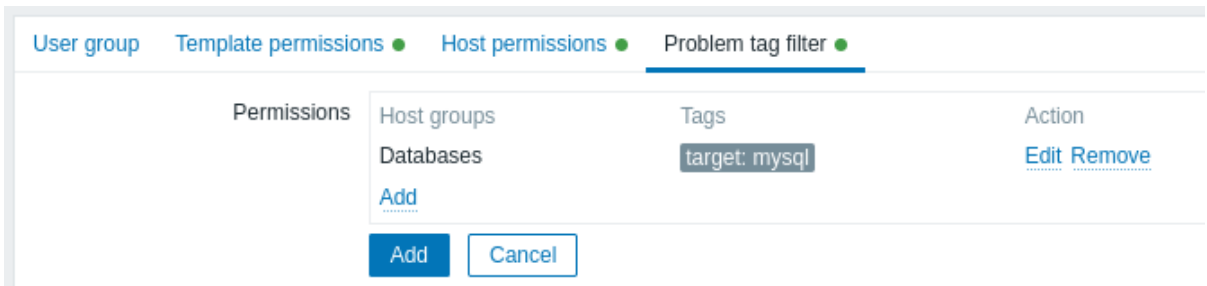
Observe que um usuário *Super admin* pode forçar grupos aninhados a terem o mesmo nível de permissões que o grupo pai; isso pode ser feito no formulário de configuração do grupo de host/template.

As abas **Permissões de template** e **Permissões de host** suportam o mesmo conjunto de parâmetros.

As permissões atuais para os grupos são exibidas no bloco *Permissões*, e podem ser modificadas ou removidas.

Note:
Se um grupo de usuários tiver permissões de **Leitura-escrita** para um host e **Negar** ou nenhuma permissão para um template vinculado a esse host, os usuários desse grupo não poderão editar os items do template no host, e o nome do template será exibido como *Template inacessível*.

A aba **Filtro de tags de problema** permite definir permissões baseadas em tags para que grupos de usuários vejam problemas filtrados por nome e valor da tag:

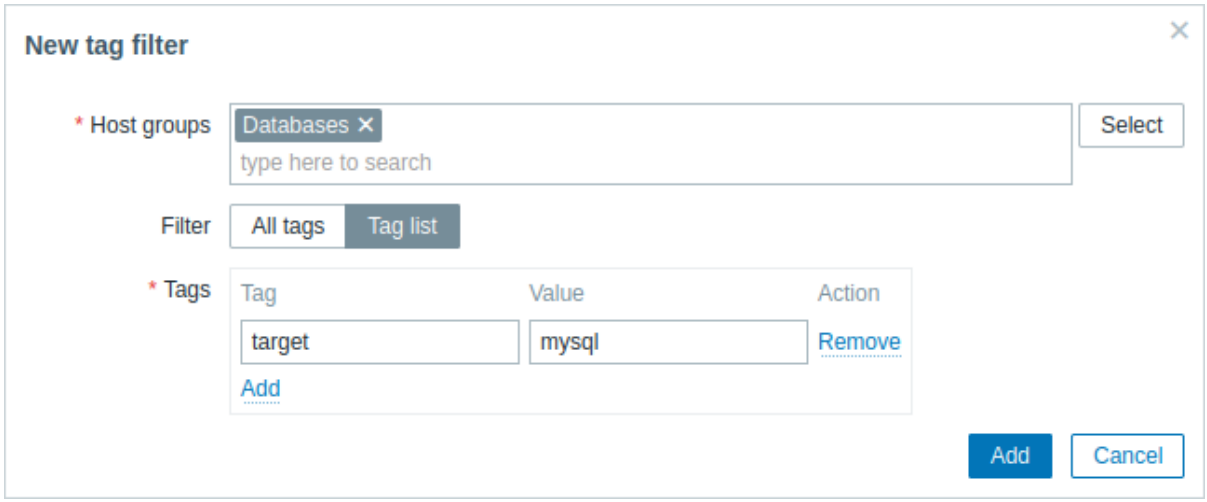


Clique em Add para escolher os grupos de host. Para selecionar um grupo de host para aplicar um filtro de tag, clique em *Selecionar* para obter a lista completa de grupos de host existentes ou comece a digitar o nome de um grupo de host para obter um menu suspenso de grupos correspondentes. Apenas grupos de host serão exibidos, pois o filtro de tags de problema não pode ser aplicado a grupos de template.

Em seguida, é possível alternar de *Todas as tags* para *Lista de tags* para definir tags específicas e seus valores para filtragem. Nomes de tags sem valores podem ser adicionados, mas valores sem nomes não. Apenas as três primeiras tags (com valores, se houver) são exibidas no bloco *Permissões*; se houver mais, elas podem ser vistas clicando ou passando o mouse sobre o ícone •••.

O filtro de tags permite separar o acesso ao grupo de host da possibilidade de ver problemas.

Por exemplo, se um administrador de banco de dados precisa ver apenas problemas do banco de dados "MySQL", é necessário criar primeiro um grupo de usuários para administradores de banco de dados e, em seguida, especificar o nome da tag "target" e o valor "mysql".



Se o nome da tag "target" for especificado e o campo de valor for deixado em branco, o grupo de usuários verá todos os problemas com o nome da tag "target" para o grupo de host selecionado. Se *Todas as tags* for selecionado, o grupo de usuários verá todos os problemas para o grupo de host especificado.

Certifique-se de que o nome e o valor da tag estejam corretamente especificados, caso contrário, o grupo de usuários não verá nenhum problema.

Vamos analisar um exemplo em que um usuário é membro de vários grupos de usuários selecionados. A filtragem, neste caso, usará a condição OR para as tags.

Grupo de usuários A			**Grupo de usuários B**			**Resultado visível para um usuário (membro) de ambos os grupos**
Filtro de tag						
Grupo de host	*Nome da tag*	*Valor da tag*	*Grupo de host*	*Nome da tag*	*Valor da tag*	
Databases	target	mysql	Databases	target	oracle	problemas target:mysql ou target:oracle visíveis
Databases	definido como: *Todas as tags*		Databases	target	oracle	Todos os problemas visíveis
Não configurado no **Filtro de tags de problema**			Databases	target	oracle	problemas target:oracle visíveis

Attention:
Adicionar um filtro (por exemplo, todas as tags em um determinado grupo de host ”Databases”) resulta em não ser possível ver os problemas de outros grupos de host.

Acesso de vários grupos de usuários

Um usuário pode pertencer a qualquer número de grupos de usuários. Esses grupos podem ter permissões de acesso diferentes a hosts ou templates.

Portanto, é importante saber a quais entidades um usuário sem privilégios poderá acessar como resultado. Por exemplo, vamos considerar como o acesso ao host **X** (no Grupo de hosts 1) será afetado em várias situações para um usuário que está nos grupos de usuários A e B.

- Se o Grupo A tiver apenas acesso *Leitura* ao Grupo de hosts 1, mas o Grupo B tiver acesso *Leitura e gravação* ao Grupo de hosts 1, o usuário terá acesso **Leitura e gravação** a ’X’.

Attention:
Permissões de “Leitura e gravação” têm precedência sobre permissões de “Leitura”.

- No mesmo cenário acima, se ’X’ estiver simultaneamente também no Grupo de hosts 2 que está **negado** ao Grupo A ou B, o acesso a ’X’ ficará **indisponível**, apesar do acesso *Leitura e gravação* ao Grupo de hosts 1.
- Se o Grupo A não tiver permissões definidas e o Grupo B tiver acesso *Leitura e gravação* ao Grupo de hosts 1, o usuário terá acesso **Leitura e gravação** a ’X’.
- Se o Grupo A tiver acesso *Negado* ao Grupo de hosts 1 e o Grupo B tiver acesso *Leitura e gravação* ao Grupo de hosts 1, o acesso do usuário a ’X’ será **negado**.

Outros detalhes

- Um usuário de nível Admin com acesso *Leitura-escrita* a um host não poderá vincular/desvincular templates, se ele não tiver acesso ao grupo de templates ao qual eles pertencem. Com acesso *Leitura* ao grupo de templates, ele poderá vincular/desvincular templates ao host, no entanto, não verá nenhum template na lista de templates e não poderá operar com templates em outros lugares.
- Um usuário de nível Admin com acesso *Leitura* a um host não verá o host na lista de hosts da seção de configuração; no entanto, os triggers do host estarão acessíveis na configuração de serviços de TI.
- Qualquer usuário que não seja Super Admin (incluindo ’guest’) pode ver mapas de rede, desde que o mapa esteja vazio ou tenha apenas imagens. Quando hosts, grupos de hosts ou triggers são adicionados ao mapa, as permissões são respeitadas.
- O Zabbix server não enviará notificações para usuários definidos como destinatários de operações de ação se o acesso ao host em questão for explicitamente ”negado”.

13 Armazenamento de segredos

Visão geral O Zabbix pode ser configurado para recuperar informações confidenciais de um cofre seguro. Os seguintes serviços de gerenciamento de segredos são suportados: HashiCorp Vault KV Secrets Engine - Versão 2, CyberArk Vault CV12.

Os segredos podem ser usados para recuperar:

- valores de macros de usuário
- credenciais de acesso ao banco de dados

O Zabbix fornece acesso somente leitura aos segredos em um cofre, assumindo que os segredos são gerenciados por outra pessoa.

Para obter informações sobre a configuração de um provedor de cofre específico, consulte:

- Configuração do HashiCorp
- Configuração do CyberArk

Cache de valores secretos Os valores de macros secretas do Vault são recuperados pelo Zabbix server a cada atualização dos dados de configuração e, em seguida, armazenados no cache de configuração. O Zabbix proxy recebe os valores das macros secretas do vault do Zabbix server em cada sincronização de configuração e os armazena em seu próprio cache de configuração.

Attention:
A criptografia deve estar habilitada entre o Zabbix server e o proxy; caso contrário, uma mensagem de aviso do server será registrada.

Para acionar manualmente a atualização dos valores secretos em cache a partir de um vault, use a opção de linha de comando 'secrets_reload' option.

Para as credenciais do banco de dados do frontend do Zabbix, o cache está desabilitado por padrão, mas pode ser habilitado definindo a opção `$DB['VAULT_CACHE'] = true` em zabbix.conf.php. As credenciais serão armazenadas em um cache local usando o diretório de arquivos temporários do sistema de arquivos. O servidor web deve permitir gravação em uma pasta temporária privada (por exemplo, para o Apache, a opção de configuração `PrivateTmp=True` deve ser definida). Para controlar com que frequência o cache de dados é atualizado/inválido, use a constante ZBX_DATA_CACHE_TTL constant.

Configuração do TLS Para configurar o TLS para comunicação entre os componentes do Zabbix e o vault, adicione um certificado assinado por uma autoridade certificadora (CA) ao repositório padrão de CA do sistema. Para usar outro local, especifique o diretório no parâmetro de configuração SSLCALocation do Zabbix server/proxy, coloque o arquivo de certificado dentro desse diretório e execute o comando CLI:

```
c_rehash .
```

1 Configuração do CyberArk

Esta seção explica como configurar o Zabbix para recuperar segredos do CyberArk Vault CV12.

O vault deve ser instalado e configurado conforme descrito na documentação oficial do CyberArk.

Para saber mais sobre a configuração do TLS no Zabbix, consulte *Armazenamento de segredos*.

Credenciais do banco de dados

O acesso a um segredo com credenciais do banco de dados é configurado separadamente para cada componente do Zabbix.

Server e proxies

Para obter credenciais do banco de dados do vault para o Zabbix server ou proxy, especifique os seguintes parâmetros de configuração no arquivo de configuração:

- `Vault` - qual provedor de vault deve ser usado;
- `VaultURL` - URL HTTP[S] do servidor de vault;
- `VaultDBPath` - consulta ao segredo do vault contendo as credenciais do banco de dados que serão recuperadas pelas chaves "Content" e "UserName" (esta opção só pode ser usada se DBUser e DBPassword não estiverem especificados);
- `VaultTLSCertFile`, `VaultTLSKeyFile` - nomes dos arquivos de certificado e chave SSL; configurar essas opções não é obrigatório, mas altamente recomendado;
- `VaultPrefix` - prefixo personalizado para o caminho ou consulta do vault, dependendo do vault; se não especificado, o padrão mais adequado será usado.

> **Attention:**
> O Zabbix server também usa os parâmetros de configuração `Vault`, `VaultURL`, `VaultTLSCertFile`, `VaultTLSKeyFile` e `VaultPrefix` para autenticação no vault ao processar macros de segredo do vault.

O Zabbix server e o Zabbix proxy leem os parâmetros de configuração relacionados ao vault dos arquivos *zabbix_server.conf* e *zabbix_proxy.conf* na inicialização.

Exemplo

1. No *zabbix_server.conf*, especifique os seguintes parâmetros:

```
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. O Zabbix enviará a seguinte solicitação de API para o cofre:

```
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. A resposta do cofre conterá as chaves "Content" e "UserName":

```
{
    "Content": <password>,
    "UserName": <username>,
    "Address": <address>,
    "Database": <Database>,
    "PasswordChangeInProcess":<PasswordChangeInProcess>
}
```

4. Como resultado, o Zabbix usará as seguintes credenciais para autenticação no banco de dados:

- Nome de usuário: <username>
- Senha: <password>

Frontend

Para obter credenciais do banco de dados do vault para o frontend do Zabbix, especifique os seguintes parâmetros durante a instalação do frontend.

1. Na etapa *Configurar conexão com o BD*, defina o parâmetro *Armazenar credenciais em* como "CyberArk Vault".
2. Em seguida, preencha os parâmetros adicionais:

Parâmetro	Obrigatório	Valor padrão	Descrição
Endpoint da API do Vault	sim	https://localhost:1858	Especifique a URL para conectar ao vault no formato `scheme://host:port`
Prefixo do Vault	não	/AIMWebService/api/Accounts	Forneça um prefixo personalizado para o caminho ou consulta do vault. Se não for especificado, o padrão será usado.
String de consulta secreta do Vault	sim		Uma consulta que especifica de onde as credenciais do banco de dados devem ser recuperadas. Exemplo: `AppID=foo&Query=Safe=bar;Object=buzz`
Certificados do Vault	não		Após marcar a caixa de seleção, parâmetros adicionais aparecerão permitindo configurar a autenticação do cliente. Embora este parâmetro seja opcional, é altamente recomendável habilitá-lo para comunicação com o CyberArk Vault.
Arquivo de certificado SSL	não	conf/certs/cyberark-cert.pem	Caminho para o arquivo de certificado SSL. O arquivo deve estar no formato PEM. Se o arquivo de certificado também contiver a chave privada, deixe o parâmetro de arquivo de chave SSL vazio.
Arquivo de chave SSL	não	conf/certs/cyberark-key.pem	Nome do arquivo de chave privada SSL usado para autenticação do cliente. O arquivo deve estar no formato PEM.

Valores de macro de usuário

Para usar o CyberArk Vault para armazenar valores de macro de usuário *Vault secret*, certifique-se de que:

- O Zabbix server está configurado para trabalhar com o CyberArk Vault;
- O parâmetro *Vault provider* em *Administração → Geral → Outros* está definido como "CyberArk Vault".

Note:
Apenas o Zabbix server requer acesso aos valores de macro *Vault secret* do vault. Outros componentes do Zabbix (proxy, frontend) não precisam desse acesso.

O valor da macro deve conter uma consulta (como `query:key`).

Veja *Macros de segredo do Vault* para informações detalhadas sobre o processamento do valor da macro pelo Zabbix.

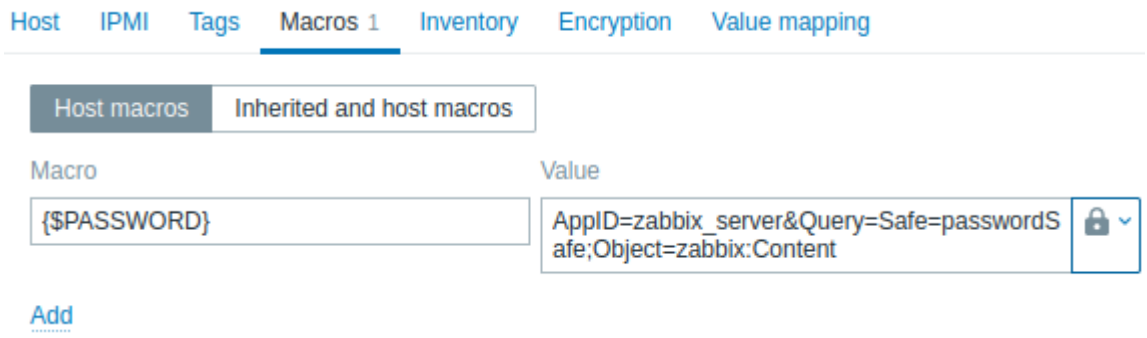
Sintaxe da consulta

O símbolo de dois pontos (":") é reservado para separar a consulta da chave.

Se uma consulta contiver uma barra ou dois pontos, esses símbolos devem ser codificados em URL ("/" é codificado como "%2F", ":" é codificado como "%3A").

Exemplo

1. No Zabbix, adicione uma macro de usuário {$PASSWORD} do tipo *Vault secret* e com o valor AppID=zabbix_server&Query=Safe=pa



2. O Zabbix enviará a seguinte solicitação de API para o vault:

```
curl \
--header "Content-Type: application/json" \
--cert cert.pem \
--key key.pem \
https://127.0.0.1:1858/AIMWebService/api/Accounts?AppID=zabbix_server&Query=Safe=passwordSafe;Object=zabbi
```

3. A resposta do vault conterá a chave "Content":

```
{
    "Content": <password>,
    "UserName": <username>,
    "Address": <address>,
    "Database" :<Database>,
    "PasswordChangeInProcess":<PasswordChangeInProcess>
}
```

4. Como resultado, o Zabbix resolverá a macro {$PASSWORD} para o valor - <password>

Atualizar a configuração existente

Para atualizar uma configuração existente para recuperar segredos de um CyberArk Vault:

1. Atualize os parâmetros do arquivo de configuração do Zabbix server ou proxy conforme descrito na seção *Credenciais do banco de dados*.

2. Atualize as configurações de conexão com o banco de dados reconfigurando o frontend do Zabbix e especificando os parâmetros necessários conforme descrito na seção *Frontend*. Para reconfigurar o frontend do Zabbix, abra a URL de configuração do frontend no navegador:

- para Apache: http://<server_ip_ou_nome>/zabbix/setup.php
- para Nginx: http://<server_ip_ou_nome>/setup.php

Como alternativa, esses parâmetros podem ser definidos no arquivo de configuração do frontend (*zabbix.conf.php*):

```
$DB['VAULT']                    = 'CyberArk';
$DB['VAULT_URL']                = 'https://127.0.0.1:1858';
$DB['VAULT_DB_PATH']            = 'AppID=foo&Query=Safe=bar;Object=buzz';
$DB['VAULT_TOKEN']              = '';
$DB['VAULT_CERT_FILE']          = 'conf/certs/cyberark-cert.pem';
$DB['VAULT_KEY_FILE']           = 'conf/certs/cyberark-key.pem';
$DB['VAULT_PREFIX']             = '';
```

3. Configure macros de usuário conforme descrito na seção *Valores de macro de usuário*, se necessário.

Para atualizar uma configuração existente para recuperar segredos de um HashiCorp Vault, consulte *Configuração do HashiCorp*.

2 Configuração do HashiCorp

Visão geral

Esta seção explica como configurar o Zabbix para recuperar segredos do HashiCorp Vault KV Secrets Engine - Versão 2.

O vault deve ser implantado e configurado conforme descrito na documentação oficial do HashiCorp.

Para saber mais sobre como configurar o TLS no Zabbix, consulte *Armazenamento de segredos*.

Recuperando credenciais do banco de dados

Para recuperar um segredo com credenciais do banco de dados com sucesso, é necessário configurar ambos:

- Zabbix server/proxy
- Zabbix frontend

Server/proxy

Para configurar o Zabbix server ou proxy, especifique os seguintes parâmetros de configuração no arquivo de configuração:

- `Vault` - qual provedor de vault deve ser usado;
- `VaultToken` - token de autenticação do vault (veja o arquivo de configuração do Zabbix server/proxy para detalhes);
- `VaultURL` - URL HTTP[S] do servidor vault;
- `VaultDBPath` - caminho para o segredo do vault contendo as credenciais do banco de dados (esta opção só pode ser usada se DBUser e DBPassword não estiverem especificados); o Zabbix server ou proxy irá recuperar as credenciais pelas chaves ”password” e ”username”;
- `VaultPrefix` - prefixo personalizado para o caminho ou consulta do vault, dependendo do vault; se não especificado, o padrão mais adequado será usado.

Attention:
O Zabbix server também usa os parâmetros de configuração `Vault`, `VaultToken`, `VaultURL` e `VaultPrefix` para autenticação no vault ao processar macros de segredo do vault.

O Zabbix server e o Zabbix proxy leem os parâmetros de configuração relacionados ao vault dos arquivos *zabbix_server.conf* e *zabbix_proxy.conf* na inicialização. Além disso, o Zabbix server e o Zabbix proxy irão ler a variável de ambiente `VAULT_TOKEN` uma vez durante a inicialização e irão removê-la para que não fique disponível através de scripts derivados; é um erro se ambos os parâmetros `VaultToken` e `VAULT_TOKEN` contiverem um valor.

Note:
O Zabbix server ou proxy suporta renovações automáticas de tokens de serviço e tokens de serviço periódicos. Se o Zabbix detectar que um token é renovável, ele continuará renovando-o automaticamente até que o TTL máximo do token seja atingido ou, no caso de tokens de serviço periódicos, indefinidamente.

Exemplo

1. Em *zabbix_server.conf*, especifique os seguintes parâmetros:

```
Vault=HashiCorp
VaultToken=hvs.CAESIIG_PILmULFYOsEyWHxkZ2mF2a8VPKNLE8eHqd4autYGGh4KHGh2cy5aeTY0NFNSaUp3ZnpWbDF1RUNjUkNTZEg
VaultURL=https://127.0.0.1:8200
VaultDBPath=database
VaultPrefix=/v1/secret/data/zabbix/
```

2. Execute os seguintes comandos CLI para criar o segredo necessário no vault:

```
#### Habilite o ponto de montagem "secret/" se ainda não estiver habilitado; observe que "kv-v2" deve ser
vault secrets enable -path=secret/ kv-v2

#### Adicione novos segredos com as chaves username e password sob o ponto de montagem "secret/" e caminho
vault kv put -mount=secret zabbix/database username=zabbix password=<password>

#### Teste se o segredo foi adicionado com sucesso.
vault kv get secret/zabbix/database

#### Finalmente, teste com Curl; observe que "data" precisa ser adicionado manualmente após o ponto de mon
curl --header "X-Vault-Token: <VaultToken>" https://127.0.0.1:8200/v1/secret/data/zabbix/database
```

3. Como resultado, o Zabbix server irá recuperar as seguintes credenciais para autenticação no banco de dados:

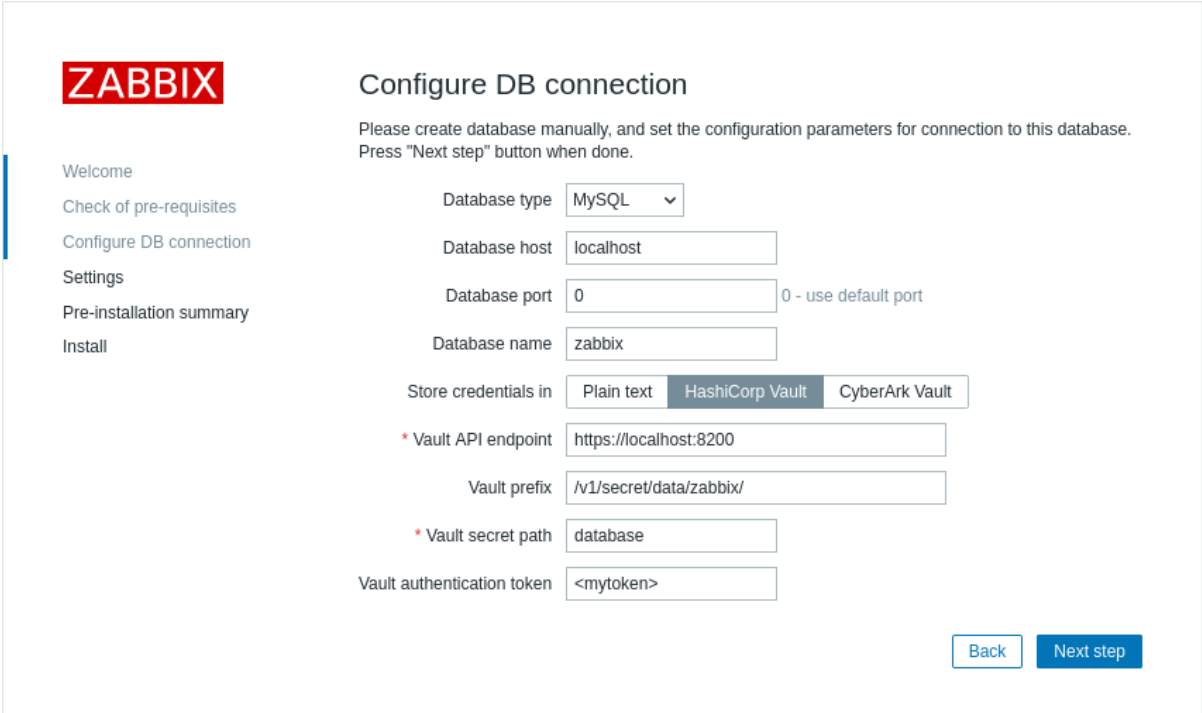
- Usuário: zabbix
- Senha: <password>

Frontend

O frontend do Zabbix pode ser configurado para recuperar as credenciais do banco de dados do vault durante a instalação do frontend ou atualizando o arquivo de configuração do frontend (`zabbix.conf.php`).

Attention:
Se as credenciais do vault foram alteradas desde a instalação anterior do frontend, execute novamente a instalação do frontend ou atualize o `zabbix.conf.php`. Veja também: Atualizando a configuração existente.

Durante a **instalação do frontend**, os parâmetros de configuração devem ser especificados na etapa *Configurar conexão com o BD*:



- Defina o parâmetro *Armazenar credenciais em* como ”HashiCorp Vault”.
- Especifique os parâmetros de conexão:

Parâmetro	Obrigatório	Valor padrão	Descrição
Endpoint da API do Vault	sim	https://localhost:8200	Especifique a URL para conectar ao vault no formato `scheme://host:port`
Prefixo do Vault	não	/v1/secret/data/	Forneça um prefixo personalizado para o caminho ou consulta do vault. Se não for especificado, o padrão será usado. Exemplo: `/v1/secret/data/zabbix/`
Caminho do segredo do Vault	não		Um caminho para o segredo de onde as credenciais para o banco de dados devem ser recuperadas pelas chaves ”password” e ”username”. Exemplo: `database`
Token de autenticação do Vault	não		Forneça um token de autenticação para acesso somente leitura ao caminho do segredo. Consulte a documentação da HashiCorp para obter informações sobre a criação de tokens e políticas do vault.

Recuperando valores de macros de usuário

Para usar o HashiCorp Vault para armazenar valores de macros de usuário *Vault secret*, certifique-se de que:

- O Zabbix server está configurado para trabalhar com o HashiCorp Vault;
- O parâmetro *Vault provider* em *Administração → Geral → Outros* está definido como "HashiCorp Vault" (padrão).

Note:
Apenas o Zabbix server requer acesso aos valores das macros *Vault secret* do vault. Outros componentes do Zabbix (proxy, frontend) não precisam desse acesso.

O valor da macro deve conter um caminho de referência (como `path:key`, por exemplo, `macros:password`). O token de autenticação especificado durante a configuração do Zabbix server (pelo parâmetro `VaultToken`) deve fornecer acesso somente leitura a esse caminho.

Veja *Macros Vault secret* para informações detalhadas sobre o processamento do valor da macro pelo Zabbix.

Sintaxe do caminho

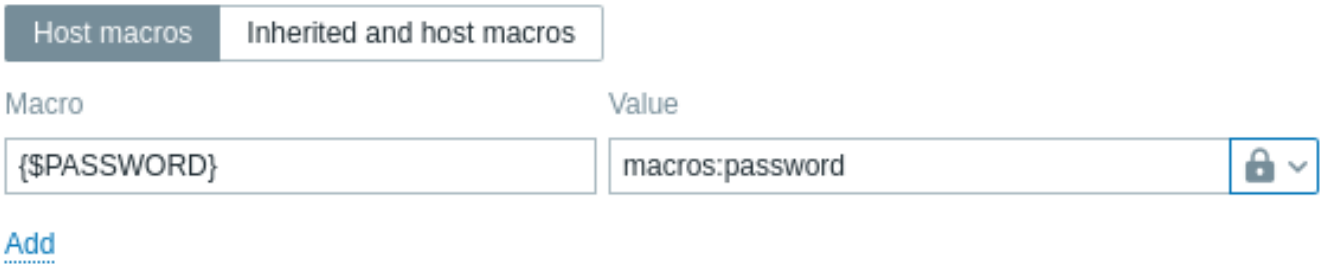
Os símbolos barra ("/") e dois-pontos (":") são reservados.

Uma barra só pode ser usada para separar um ponto de montagem de um caminho (por exemplo, *secret/zabbix*, onde o ponto de montagem é "secret" e o caminho é "zabbix"). No caso das macros do Vault, dois-pontos só podem ser usados para separar um caminho/consulta de uma chave.

É possível codificar em URL os símbolos de barra e dois-pontos se houver necessidade de criar um ponto de montagem com o nome separado por uma barra (por exemplo, *foo/bar/zabbix*, onde o ponto de montagem é "foo/bar" e o caminho é "zabbix", pode ser codificado como "foo%2Fbar/zabbix") e se um nome de ponto de montagem ou caminho precisar conter dois-pontos.

Exemplo

1. No Zabbix, adicione uma macro de usuário {$PASSWORD} do tipo "Vault secret" e com o valor `macros:password`



2. Execute os seguintes comandos CLI para criar o segredo necessário no vault:

```
#### Habilite o ponto de montagem "secret/" se ainda não estiver habilitado; observe que "kv-v2" deve ser
vault secrets enable -path=secret/ kv-v2

#### Adicione um novo segredo com a chave "password" sob o ponto de montagem "secret/" e caminho "zabbix/m
vault kv put -mount=secret zabbix/macros password=<password>

#### Teste se o segredo foi adicionado com sucesso.
vault kv get secret/zabbix/macros

#### Finalmente, teste com Curl; observe que "data" precisa ser adicionado manualmente após o ponto de mon
curl --header "X-Vault-Token: <VaultToken>" https://127.0.0.1:8200/v1/secret/data/zabbix/macros
```

3. Como resultado, o Zabbix irá resolver a macro {$PASSWORD} para o valor: <password>

Atualizando a configuração existente

Para atualizar uma configuração existente para recuperar segredos de um HashiCorp Vault:

1. Atualize os parâmetros do arquivo de configuração do Zabbix server ou proxy conforme descrito na seção *Credenciais do banco de dados*.
2. Atualize as configurações de conexão com o banco de dados reconfigurando o frontend do Zabbix e especificando os parâmetros necessários conforme descrito na seção *Frontend*. Para reconfigurar o frontend do Zabbix, abra a URL de configuração do frontend no navegador:

- para Apache: http://<server_ip_or_name>/zabbix/setup.php
- para Nginx: http://<server_ip_or_name>/setup.php

Como alternativa, esses parâmetros podem ser definidos no arquivo de configuração do frontend (*zabbix.conf.php*):

```
$DB['VAULT']                    = 'HashiCorp';
$DB['VAULT_URL']                = 'https://localhost:8200';
$DB['VAULT_DB_PATH']            = 'database';
$DB['VAULT_TOKEN']              = '<mytoken>';
$DB['VAULT_CERT_FILE']          = '';
$DB['VAULT_KEY_FILE']           = '';
$DB['VAULT_PREFIX']             = '/v1/secret/data/zabbix/';
```

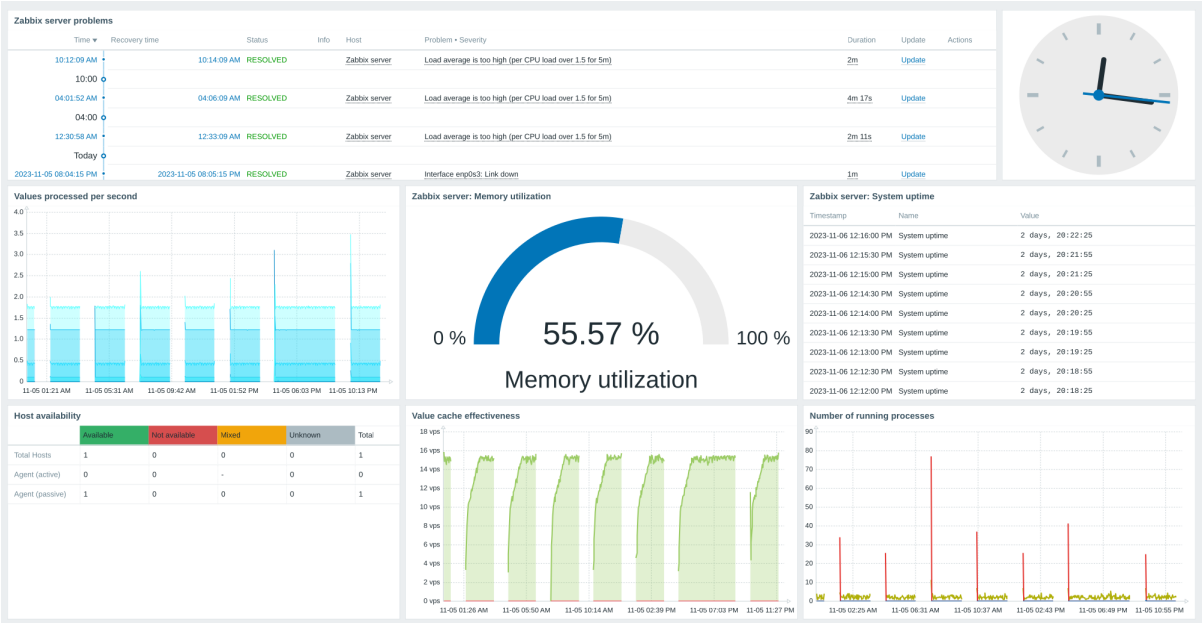
3. Configure macros de usuário conforme descrito na seção *Valores de macro de usuário*, se necessário.

Para atualizar uma configuração existente para recuperar segredos de um CyberArk Vault, consulte *Configuração do CyberArk*.

14 Relatórios agendados

Visão geral

Com o recurso *Relatórios agendados*, você pode configurar o envio de uma versão em PDF de um determinado dashboard para destinatários especificados em intervalos recorrentes.



Pré-requisitos:

- O serviço web do Zabbix deve estar instalado e configurado corretamente para permitir a geração de relatórios agendados - consulte Configurando relatórios agendados para instruções.
- Um usuário deve ter uma função de usuário do tipo *Admin* ou *Super admin* com as seguintes permissões:
  - *Relatórios agendados* no bloco *Acesso aos elementos da interface* (para visualizar as configurações do relatório)
  - *Gerenciar relatórios agendados* no bloco *Acesso às ações* (para criar/editar relatórios)

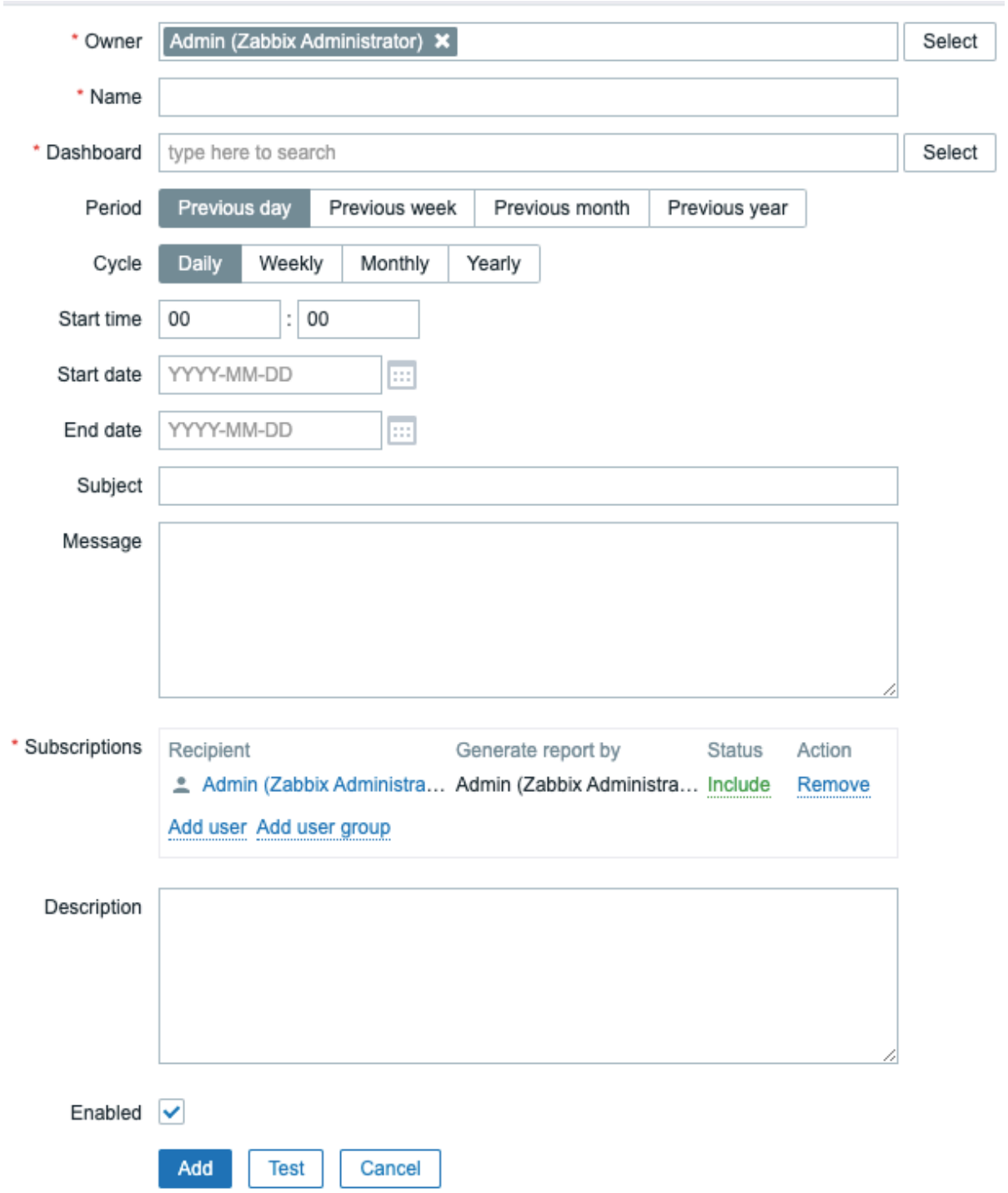
Para criar um relatório agendado no frontend do Zabbix, faça o seguinte:

- Vá para: *Relatórios* > *Relatórios agendados*.
- Clique em *Criar relatório* no canto superior direito da tela.
- Insira os parâmetros do relatório no formulário.

Você também pode criar um relatório abrindo um já existente, clicando no botão *Clonar* e, em seguida, salvando-o com um nome diferente.

Configuração

A aba *Relatórios agendados* contém os atributos gerais do relatório.



Todos os campos obrigatórios estão marcados com um asterisco vermelho.

Parâmetro	Descrição
Proprietário	Usuário que cria o relatório. Usuários com nível *Super admin* podem alterar o proprietário. Para usuários com nível *Admin*, este campo é somente leitura.
Nome	Nome do relatório; deve ser único.
Dashboard	Dashboard no qual o relatório é baseado; apenas um dashboard pode ser selecionado por vez. Para selecionar um dashboard, comece a digitar o nome - uma lista de dashboards correspondentes aparecerá; role para baixo para selecionar. Alternativamente, você pode clicar em *Selecionar* ao lado do campo e selecionar um dashboard da lista exibida.
Período	Período de tempo para o qual o relatório será preparado. Selecione o dia, semana, mês ou ano anterior.
Ciclo	Frequência de geração do relatório. Os relatórios podem ser enviados diariamente, semanalmente, mensalmente ou anualmente. O modo ”Semanal” permite selecionar os dias da semana em que o relatório será enviado.
Hora de início	Hora do dia no formato hh:mm em que o relatório será preparado. Observe que será usado o fuso horário do Zabbix server.

Parâmetro	Descrição
Repetir em	Dias da semana em que o relatório será enviado. Este campo está disponível apenas se o *Ciclo* estiver definido como ”Semanal”.
Data de início	Data em que a geração regular do relatório deve ser iniciada.
Data de término	Data em que a geração regular do relatório deve ser interrompida.
Assunto	Assunto do e-mail do relatório. Suporta a macro {TIME}.
Mensagem	Corpo do e-mail do relatório. Suporta a macro {TIME}.
Assinaturas	Lista de destinatários do relatório. Por padrão, inclui apenas o proprietário do relatório. Qualquer usuário do Zabbix com mídia de e-mail configurada pode ser especificado como destinatário do relatório. Clique em *Adicionar usuário* ou *Adicionar grupo de usuários* para adicionar mais destinatários. Clique no nome de usuário para editar as configurações: *Gerar relatório por* - se os dados do relatório devem ser gerados com base nas permissões do dashboard do usuário atual ou do destinatário. *Status* - selecione ”Incluir” para enviar o relatório ao usuário ou ”Excluir” para impedir o envio do relatório a esse usuário. Pelo menos um usuário deve ter o status ”Incluir”. O status ”Excluir” pode ser usado para excluir usuários específicos de um grupo de usuários que está incluído. Observe que usuários com permissões insuficientes (ou seja, usuários com uma função baseada no tipo de usuário *Admin* que não são membros do mesmo grupo de usuários que o destinatário ou proprietário do relatório) verão ”Usuário inacessível” ou ”Grupo de usuários inacessível” em vez dos nomes reais nos campos *Destinatário* e *Gerar relatório por*; os campos *Status* e *Ação* serão exibidos como somente leitura.
Habilitado	Status do relatório. Desmarcar esta caixa desabilitará o relatório.
Descrição	Uma descrição opcional do relatório. Esta descrição é para uso interno e não será enviada aos destinatários do relatório.

Botões do formulário

Os botões na parte inferior do formulário permitem realizar várias operações.

Add	Adicionar um relatório. Este botão está disponível apenas para novos relatórios.
Update	Atualizar as propriedades de um relatório.
Clone	Criar outro relatório com base nas propriedades do relatório atual.
Test	Testar se a configuração do relatório está correta enviando um relatório para o usuário atual.
Delete	Excluir o relatório.
Cancel	Cancelar a edição das propriedades do relatório.

Testando

Para testar um relatório, clique no botão *Testar* na parte inferior do formulário de configuração do relatório.

> **Note:**
> O botão *Testar* não está disponível se o formulário de configuração do relatório foi aberto a partir do menu de ações do dashboard.

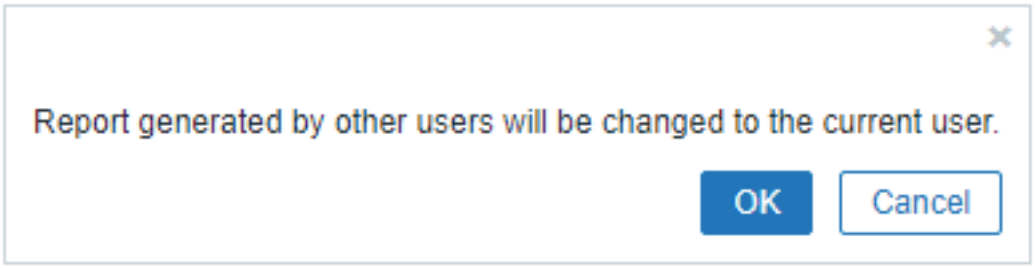
Se a configuração estiver correta, o relatório de teste é enviado imediatamente para o usuário atual. Para relatórios de teste, as configurações de assinantes e de usuário em *Gerar relatório por* são ignoradas.

Se a configuração estiver incorreta, uma mensagem de erro é exibida descrevendo a possível causa.

Atualizando um relatório

Para atualizar um relatório existente, clique no nome do relatório, faça as alterações de configuração necessárias e clique no botão *Atualizar*.

Se um relatório existente for atualizado por outro usuário e esse usuário alterar o Dashboard, ao clicar no botão *Atualizar*, será exibida uma mensagem de aviso "O relatório gerado por outros usuários será alterado para o usuário atual".



Clicar em *OK* nesta etapa levará às seguintes alterações:

- As configurações de *Gerar relatório por* serão atualizadas para exibir o usuário que editou o relatório por último (a menos que *Gerar relatório por* esteja definido para o destinatário).
- Usuários que foram exibidos como "Usuário inacessível" ou "Grupo de usuários inacessível" serão excluídos da lista de assinantes do relatório.

Clicar em *Cancelar* fechará o formulário de configuração e cancelará a atualização do relatório.

Clonando um relatório

Para clonar rapidamente um relatório existente, clique no botão *Clonar* na parte inferior de um formulário de configuração de relatório existente. Ao clonar um relatório criado por outro usuário, o usuário atual se torna o proprietário do novo relatório.

As configurações do relatório serão copiadas para o novo formulário de configuração do relatório, respeitando as permissões do usuário:

- Se o usuário que clona um relatório não tiver permissões para um dashboard, o campo *Dashboard* será limpo.
- Se o usuário que clona um relatório não tiver permissões para alguns usuários ou grupos de usuários na lista de *Assinaturas*, os destinatários inacessíveis não serão clonados.
- As configurações de *Gerar relatório por* serão atualizadas para exibir o usuário atual (a menos que *Gerar relatório por* esteja definido para o destinatário).

Altere as configurações necessárias e o nome do relatório e clique em *Adicionar*.

15 Exportação de dados

Visão geral

O Zabbix suporta exportação de dados em tempo real de duas maneiras:

- exportação para arquivos
- streaming para sistemas externos

As seguintes entidades podem ser exportadas:

- eventos de trigger
- valores de item
- tendências (apenas exportação para arquivos)

1 Exportar para arquivos

Visão geral

É possível configurar a exportação em tempo real de eventos de trigger, valores de item e tendências em um formato JSON delimitado por nova linha.

A exportação é feita em arquivos, onde cada linha do arquivo de exportação é um objeto JSON. Mapeamentos de valores não são aplicados.

Em caso de erros (os dados não podem ser gravados no arquivo de exportação ou o arquivo de exportação não pode ser renomeado ou um novo não pode ser criado após renomeá-lo), o item de dados é descartado e nunca gravado no arquivo de exportação. Ele é gravado apenas no banco de dados do Zabbix. A gravação de dados no arquivo de exportação é retomada quando o problema de gravação é resolvido.

Desde o Zabbix 7.0.22, o arquivo de exportação é criado com permissões de leitura e gravação apenas para o proprietário do arquivo. Além disso, o arquivo é legível pelo grupo do proprietário. Todas as outras permissões são negadas.

Para detalhes precisos sobre quais informações são exportadas, consulte a página do protocolo de exportação.

Observe que o host/item pode não ter metadados (grupos de host, nome do host, nome do item) se o host/item foi removido após o recebimento dos dados, mas antes do server exportar os dados.

Configuração

A exportação em tempo real de eventos de trigger, valores de item e tendências é configurada especificando um diretório para os arquivos de exportação - veja o parâmetro ExportDir na configuração do server.

Outros dois parâmetros estão disponíveis:

- `ExportFileSize` pode ser usado para definir o tamanho máximo permitido de um arquivo de exportação individual. Quando um processo precisa gravar em um arquivo, ele verifica primeiro o tamanho do arquivo. Se exceder o limite de tamanho configurado, o arquivo é renomeado adicionando .old ao seu nome e um novo arquivo com o nome original é criado.

Attention:
Um arquivo será criado para cada processo que irá gravar dados (ou seja, aproximadamente 4-30 arquivos). Como o tamanho padrão por arquivo de exportação é 1G, manter arquivos de exportação grandes pode esgotar rapidamente o espaço em disco.

- `ExportType` permite especificar quais tipos de entidades (eventos, histórico, tendências) serão exportados.

2 Transmitindo para sistemas externos

Visão geral

É possível transmitir valores de item e eventos do Zabbix para sistemas externos via HTTP (veja detalhes do protocolo).

O filtro de tags pode ser usado para transmitir subconjuntos de valores de item ou eventos.

Dois processos do Zabbix server são responsáveis pelo streaming de dados: `connector manager` e `connector worker`. Um item interno do Zabbix `zabbix[connector_queue]` permite monitorar a contagem de valores enfileirados na fila do conector.

Configuração

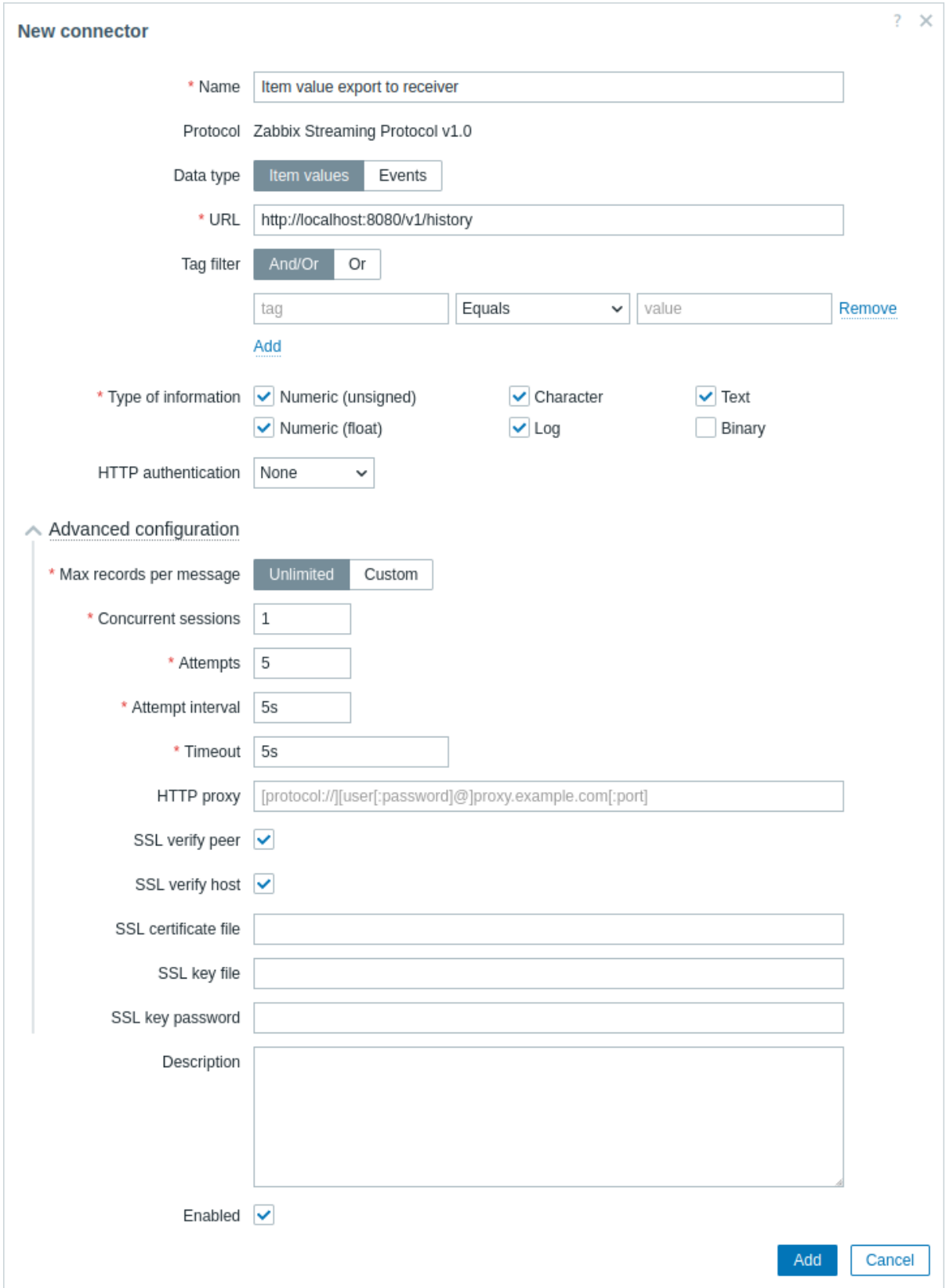
As seguintes etapas são necessárias para configurar o streaming de dados para um sistema externo:

1. Tenha um sistema remoto configurado para receber dados do Zabbix. Para este propósito, as seguintes ferramentas estão disponíveis:

- Um exemplo de receiver simples que registra as informações recebidas nos arquivos `events.ndjson` e `history.ndjson`.
- Conector Kafka para o Zabbix server - um servidor leve escrito em Go, projetado para encaminhar valores de item e eventos de um Zabbix server para um broker Kafka.

2. Defina o número necessário de workers de conector no Zabbix ajustando o parâmetro `StartConnectors` em `zabbix_server.conf`. O número de workers de conector deve corresponder (ou exceder, se sessões concorrentes forem mais de 1) à contagem de conectores configurada no frontend do Zabbix. Em seguida, reinicie o Zabbix server.

3. Configure um novo conector no frontend do Zabbix (*Administração* > *Geral* > *Conectores*) e recarregue o cache do server com o comando `zabbix_server -R config_cache_reload`.



Os campos obrigatórios são marcados com um asterisco.

Parâmetro	Descrição
Nome	Insira o nome do conector.
Tipo de dado	Selecione o tipo de dado para streaming: **Valores de item** - faz streaming dos valores de item do Zabbix para sistemas externos; **Eventos** - faz streaming de eventos do Zabbix para sistemas externos.
URL	Insira a URL do receiver. Macros de usuário são suportadas.

Parâmetro	Descrição
Filtro de tag	Exporte apenas valores de item ou eventos que correspondam ao filtro de tag. Se não definido, exporta tudo. É possível incluir e também excluir tags e valores de tag específicos. Várias condições podem ser definidas. A correspondência do nome da tag é sempre sensível a maiúsculas e minúsculas. Há vários operadores disponíveis para cada condição: **Existe** - inclui os nomes de tag especificados; **Igual** - inclui os nomes e valores de tag especificados (sensível a maiúsculas e minúsculas); **Contém** - inclui os nomes de tag especificados onde os valores de tag contêm a string inserida (correspondência de substring, insensível a maiúsculas e minúsculas); **Não existe** - exclui os nomes de tag especificados; **Não é igual** - exclui os nomes e valores de tag especificados (sensível a maiúsculas e minúsculas); **Não contém** - exclui os nomes de tag especificados onde os valores de tag contêm a string inserida (correspondência de substring, insensível a maiúsculas e minúsculas). Existem dois tipos de cálculo para as condições: **E/Ou** - todas as condições devem ser atendidas, condições com o mesmo nome de tag serão agrupadas pela condição Ou; **Ou** - basta que uma condição seja atendida.
Tipo de informação	Selecione o tipo de informação (numérico (sem sinal), numérico (float), caractere, etc.), pelo qual filtrar os valores de item que o conector deve transmitir. Este campo está disponível se o *Tipo de dado* estiver definido como ”Valores de item”.
Autenticação HTTP	Selecione a opção de autenticação: **Nenhuma** - nenhuma autenticação usada; **Básica** - autenticação básica é usada; **NTLM** - autenticação NTLM (Windows NT LAN Manager) é usada; **Kerberos** - autenticação Kerberos é usada (veja também: Configurando Kerberos com Zabbix); **Digest** - autenticação Digest é usada; **Bearer** - autenticação Bearer é usada.
Nome de usuário	Insira o nome do usuário (até 255 caracteres). Macros de usuário são suportadas. Este campo está disponível se *Autenticação HTTP* estiver definido como ”Básica”, ”NTLM”, ”Kerberos” ou ”Digest”.
Senha	Insira a senha do usuário (até 255 caracteres). Macros de usuário são suportadas. Este campo está disponível se *Autenticação HTTP* estiver definido como ”Básica”, ”NTLM”, ”Kerberos” ou ”Digest”.
Token Bearer	Insira o token Bearer. Macros de usuário são suportadas. Este campo está disponível e é obrigatório se *Autenticação HTTP* estiver definido como ”Bearer”.
Configuração avançada	Clique no rótulo *Configuração avançada* para exibir as opções de configuração avançada (veja abaixo).
Máx. registros por mensagem	Especifique o número máximo de valores ou eventos que podem ser transmitidos em uma mensagem.
Sessões concorrentes	Selecione o número de processos de envio a serem executados para este conector. Até 100 sessões podem ser especificadas; o valor padrão é ”1”.
Tentativas	Número de tentativas para transmitir dados. Até 5 tentativas podem ser especificadas; o valor padrão é ”1”.
Intervalo de tentativas	Especifique quanto tempo o conector deve aguardar após uma tentativa malsucedida de transmitir dados. Até 10s podem ser especificados; o valor padrão é ”5s”. Este campo está disponível se *Tentativas* estiver definido como ”2” ou mais. Tentativas malsucedidas são aquelas em que o estabelecimento de uma conexão falhou ou em que o código de resposta HTTP não é 200, 201, 202, 203, 204. As novas tentativas são **acionadas** em caso de erros de comunicação ou quando o código de resposta HTTP não é 200, 201, 202, 203, 204, 400, 401, 403, 404, 405, 415, 422. Redirecionamentos são seguidos, então 302 -> 200 é uma resposta positiva; enquanto 302 -> 503 acionará uma nova tentativa.
Timeout	Especifique o timeout da mensagem (1-60 segundos, padrão - 5 segundos). Sufixos de tempo são suportados, por exemplo, 30s, 1m. Macros de usuário são suportadas.

Parâmetro	Descrição
Proxy HTTP	Você pode especificar um proxy HTTP para usar no seguinte formato: `[protocol://][username[:password]@]proxy.example.com[:port]` Macros de usuário são suportadas. O prefixo opcional `protocol://` pode ser usado para especificar protocolos de proxy alternativos (o suporte ao prefixo de protocolo foi adicionado no cURL 7.21.7). Sem protocolo especificado, o proxy será tratado como um proxy HTTP. Por padrão, a porta 1080 será usada. Se *Proxy HTTP* for especificado, o proxy substituirá as variáveis de ambiente relacionadas ao proxy, como `http_proxy`, `HTTPS_PROXY`. Se não for especificado, o proxy não substituirá as variáveis de ambiente relacionadas ao proxy. O valor inserido é passado como está, sem verificação de sanidade. Você também pode inserir um endereço de proxy SOCKS. Se você especificar o protocolo errado, a conexão falhará e o item se tornará não suportado. Observe que apenas autenticação simples é suportada com proxy HTTP.
Verificar SSL peer	Marque a caixa de seleção para verificar o certificado SSL do servidor web. O certificado do servidor será automaticamente obtido do local da autoridade certificadora (CA) do sistema. Você pode substituir o local dos arquivos CA usando o parâmetro de configuração do Zabbix server ou proxy `SSLCALocation`.
Verificar host SSL	Marque a caixa de seleção para verificar se o campo *Common Name* ou o campo *Subject Alternate Name* do certificado do servidor web corresponde. Isso define a opção cURL `CURLOPT_SSL_VERIFYHOST`.
Arquivo de certificado SSL	Nome do arquivo de certificado SSL usado para autenticação do cliente. O arquivo de certificado deve estar no formato PEM[1]. Macros de usuário são suportadas. Se o arquivo de certificado também contiver a chave privada, deixe o campo *Arquivo de chave SSL* vazio. Se a chave estiver criptografada, especifique a senha no campo *Senha da chave SSL*. O diretório que contém este arquivo é especificado pelo parâmetro de configuração do Zabbix server ou proxy `SSLCertLocation`.
Arquivo de chave SSL	Nome do arquivo de chave privada SSL usado para autenticação do cliente. O arquivo de chave privada deve estar no formato PEM[1]. Macros de usuário são suportadas. O diretório que contém este arquivo é especificado pelo parâmetro de configuração do Zabbix server ou proxy `SSLKeyLocation`.
Senha da chave SSL	Senha do arquivo de chave privada SSL. Macros de usuário são suportadas.
Descrição	Insira a descrição do conector.
Habilitado	Marque a caixa de seleção para habilitar o conector.

Quando o conector Kafka é configurado com uma lista separada por vírgulas de endereços de bootstrap broker (por exemplo, `Kafka.URL=kafka1.example.com:9093,kafka2.example.com:9093`), o cliente Kafka se conecta ao(s) broker(s) que respondem primeiro e usa seus metadados de cluster. Se a lista contiver endereços de diferentes clusters Kafka, apenas o cluster que responder mais rápido será usado e outros endereços serão registrados como indisponíveis; como resultado, avisos de inicialização como o seguinte podem aparecer mesmo que o conector esteja conectado:

```
kafka cluster connected, but broker(s) "kafka1.example.com:9093, kafka2.example.com:9093" unavailable; wil
```

Em alguns ambientes (redes privadas, redes de contêineres ou configurações DNS/hosts não padrão), nomes de host ou IPs podem ser resolvidos para endereços de loopback (por exemplo, `127.0.0.1/localhost`) ou serem normalizados pelo cliente, o que pode tornar esses avisos enganosos. Para reduzir a confusão, certifique-se de que todos os endereços `Kafka.URL` pertençam ao mesmo cluster Kafka, verifique a resolução DNS a partir do host do conector e os `advertised.listeners` dos brokers, e prefira endereços que resolvam para o endereço anunciado do broker.

Protocolo

A comunicação entre o server e o receptor é feita via HTTP usando REST API, NDJSON, "Content-Type: application/x-ndjson".

Para mais detalhes, consulte Protocolo de exportação de JSON delimitado por nova linha.

Solicitação do servidor

Exemplo de valores de item em streaming:

```
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
```

Exemplo de eventos em streaming:

```
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}
```

Resposta do receptor

A resposta consiste no código de status da resposta HTTP e no payload JSON. O código de status da resposta HTTP deve ser ”200”, ”201”, ”202”, ”203” ou ”204” para solicitações que foram tratadas com sucesso, outros para solicitações com falha.

Exemplo de sucesso:

```
localhost:8080/v1/history": HTTP/1.1 200 OK
Date: Wed, 11 Jan 2023 16:40:30 GMT
Content-Length: 0
```

Exemplo com erros:

```
localhost:8080/v1/history": HTTP/1.1 422 Unprocessable Entity
Content-Type: application/json
Date: Wed, 11 Jan 2023 17:07:36 GMT
Content-Length: 55

{"error":"invalid character '{' after top-level value"}
```

3 Gateway SNMP

Visão geral

O Zabbix SNMP gateway é uma extensão AgentX para o snmpd que suporta polling e trapping SNMP.

Com o Zabbix SNMP gateway é possível usar o protocolo SNMP para recuperar:

- dados de trigger;
- dados de trigger de problema;
- status do grupo de hosts (contagem de triggers por status de trigger por grupo)

Os dados são recuperados pelo OID, que é uma combinação de uma base comum e um sufixo específico. A **base** comum é definida no arquivo de configuração do SNMP gateway, por exemplo:

- `BaseOID=1.3.6.1.4.1.3043.7.55` - para qualquer dado de trigger;
- `ProblemBaseOID=1.3.6.1.4.1.3047.7.55` - para dados de trigger de problema;
- `BaseOID=1.3.6.1.4.1.3046.7.55` - para status do grupo de hosts.

O **sufixo** do OID é definido na configuração dos triggers do host como uma **tag** (por exemplo, `OIDSuffix:3`) no frontend.

Neste caso, todas as informações para o trigger estarão disponíveis em `OID=1.3.6.1.4.1.3043.7.55.X.3`. ”X” aqui será o número dos campos de dados do trigger (ou seja, 1 - sufixo, 2 - ID, 3 - expressão, 4 - descrição, etc.).

Para uma descrição mais detalhada e um exemplo de arquivo de configuração, consulte o arquivo readme do SNMP gateway.

Instalação e configuração

Consulte o arquivo readme do SNMP gateway para obter instruções sobre:

- instalar e configurar o snmpd;
- habilitar o suporte ao AgentX;
- configurar o Zabbix SNMP gateway;
- configurar traps SNMP para alterações de estado de trigger.

Recuperando dados

Com tudo configurado corretamente, você pode usar os comandos `snmpwalk` e `snmpget` para recuperar dados:

```
[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
...
```

Opções de filtragem

Você pode limitar as informações de trigger de problema na configuração do SNMP gateway:

- por severidade (por padrão `ProblemMinSeverity=-1`)
- ocultando problemas reconhecidos (por padrão `ProblemHideAck=false`)

Você pode limitar a contagem de problemas por grupo de host na configuração do SNMP gateway:

- por triggers em estado desconhecido (por padrão `CountUnknown=false`)
- por triggers com problemas reconhecidos/não reconhecidos/todos (por padrão `CountAcknowledgeStatus=all`)

8. Monitoramento de serviço

Visão geral O monitoramento de serviços é um monitoramento em nível de negócios que pode ser usado para obter uma visão geral de toda a árvore de serviços da infraestrutura de TI, identificar pontos fracos da infraestrutura, calcular o SLA de vários serviços de TI e verificar outras informações em um nível mais alto. O monitoramento de serviços foca na disponibilidade geral de um serviço em vez de detalhes de baixo nível, como falta de espaço em disco, alta carga do processador, etc. O monitoramento de serviços também fornece funcionalidade para encontrar a causa raiz de um problema se um serviço não estiver funcionando como esperado.

O monitoramento de serviços permite criar uma representação hierárquica dos dados monitorados.

Uma estrutura de serviço muito simples pode ser assim:

```
Serviço
|
|-Estações de trabalho
| |
| |-Estação de trabalho1
| |
| |-Estação de trabalho2
|
|-Servidores
```

Cada nó da estrutura possui o atributo status. O status é calculado e propagado para os níveis superiores de acordo com o algoritmo selecionado. O status dos nós individuais é afetado pelo status dos problemas mapeados. O mapeamento de problemas é realizado com tagging.

O Zabbix pode enviar notificações ou executar automaticamente um script no servidor Zabbix caso seja detectada uma alteração no status do serviço. É possível definir regras flexíveis para que um serviço pai entre em 'Estado de Problema' com base nos status dos serviços filhos. Os dados de problemas dos serviços podem então ser usados para calcular o SLA e enviar relatórios de SLA com base em um conjunto flexível de condições.

O monitoramento de serviços é configurado no menu Serviços, que consiste nas seguintes seções:

- Serviços

A seção Serviços permite construir uma hierarquia da sua infraestrutura monitorada adicionando serviços pai e, em seguida, serviços filhos aos serviços pai.

Além de configurar a árvore de serviços, esta seção fornece uma visão geral de toda a infraestrutura e permite identificar rapidamente os problemas que levaram a uma alteração no status do serviço.

- SLA

Nesta seção você pode definir acordos de nível de serviço e definir objetivos de nível de serviço para serviços específicos.

- Relatório de SLA

Nesta seção você pode visualizar relatórios de SLA.

Ações de serviço

Você também pode configurar ações de serviço.

As ações de serviço são opcionais e permitem:

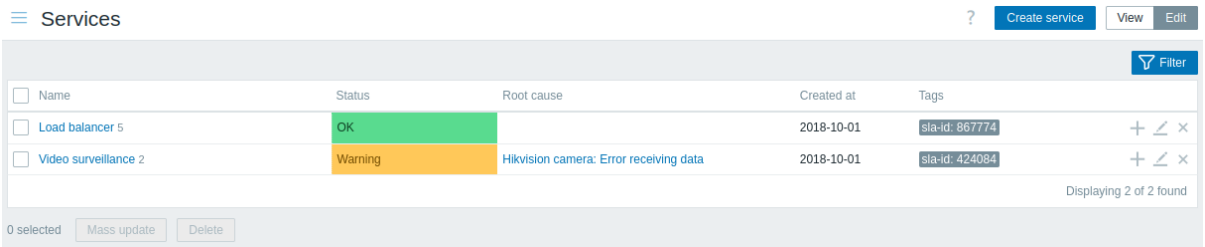
- enviar uma notificação de que um serviço está fora do ar
- executar um comando remoto em um servidor Zabbix após uma alteração no status do serviço
- enviar uma notificação de recuperação quando um serviço estiver ativo novamente.

Veja também:

- Exemplo de configuração de monitoramento de SLA
- Notas sobre atualização de serviços de versões do Zabbix abaixo da 6.0

1 Árvore de serviços

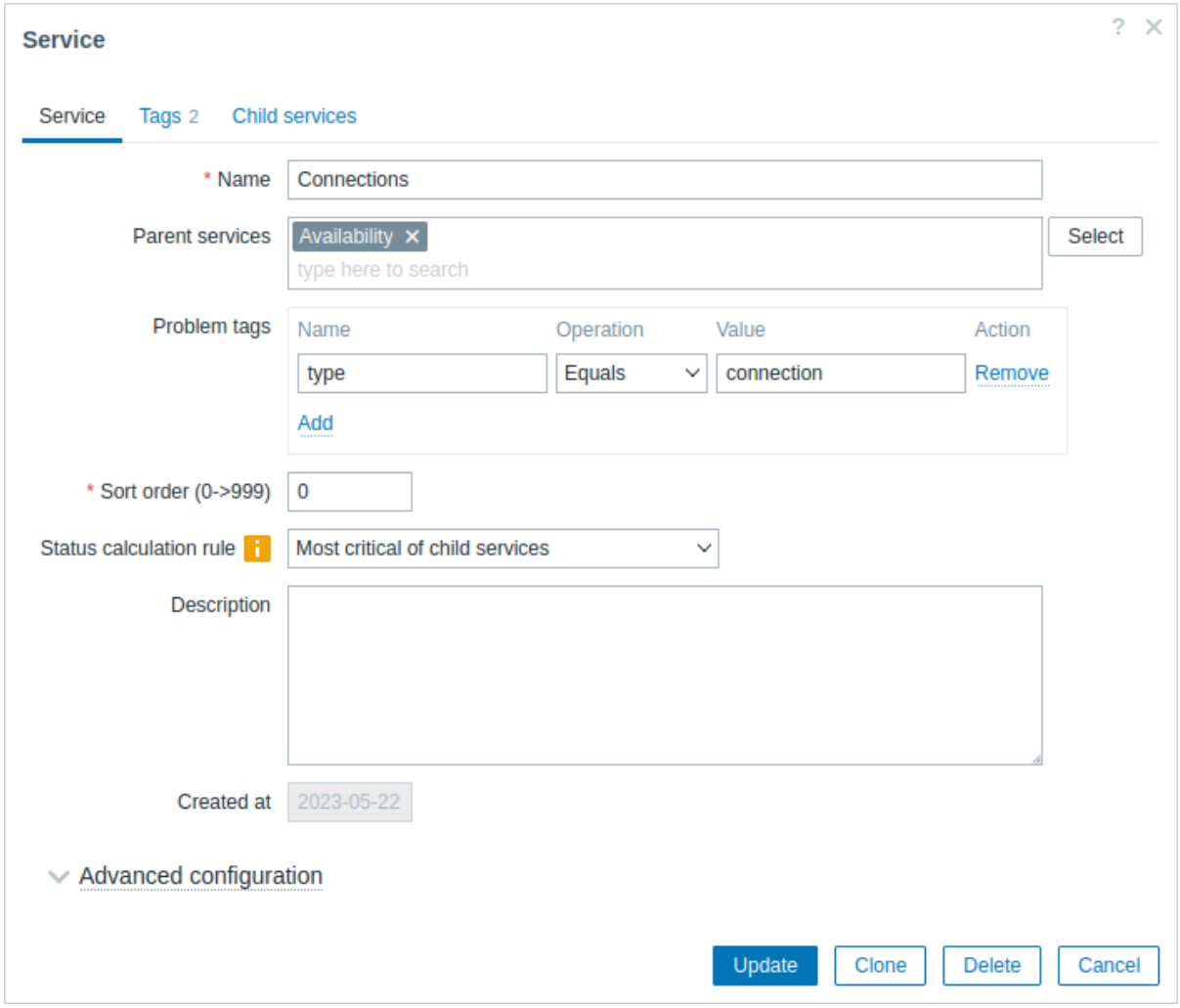
A árvore de serviços é configurada na seção de menu *Serviços -> Serviços*. No canto superior direito, alterne de Visualizar para o modo de Edição.



Para configurar um novo serviço, clique no botão *Criar serviço* no canto superior direito.

Para adicionar rapidamente um serviço filho, você pode, alternativamente, pressionar o ícone de mais ao lado do serviço pai. Isso abrirá o mesmo formulário de configuração de serviço, mas o parâmetro *Serviços pai* será preenchido automaticamente.

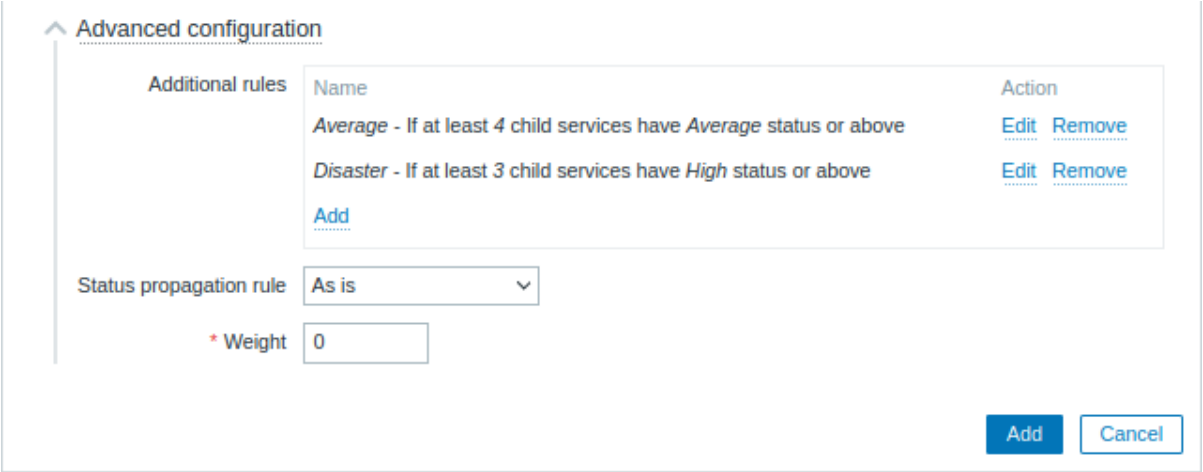
Configuração do serviço Na aba **Serviço**, especifique os parâmetros necessários do serviço:



Todos os campos obrigatórios estão marcados com um asterisco vermelho.

Parâmetro	Descrição
Nome	Nome do serviço.
Serviços pai	Serviços pai aos quais o serviço pertence. Deixe este campo vazio se você estiver adicionando o serviço de nível mais alto. Um serviço pode ter vários serviços pai. Neste caso, ele será exibido na árvore de serviços sob cada um dos serviços pai.
Tags de problema	Especifique tags para mapear os dados do problema ao serviço. Várias condições podem ser definidas. A correspondência do nome da tag diferencia maiúsculas de minúsculas. Existem dois operadores disponíveis para cada condição: **Igual** - inclui os nomes e valores de tags especificados (diferencia maiúsculas de minúsculas) **Contém** - inclui os nomes de tags especificados onde os valores das tags contêm a string inserida (correspondência de substring, diferencia maiúsculas de minúsculas)
Ordem de classificação	Ordem de classificação para exibição, o menor vem primeiro.
Regra de cálculo de status	Regra para calcular o status do serviço: **Mais crítico se todos os filhos tiverem problemas** - o problema mais crítico nos serviços filhos é usado para colorir o status do serviço, se todos os filhos tiverem problemas **Mais crítico dos serviços filhos** - o problema mais crítico nos serviços filhos é usado para colorir o status do serviço **Definir status como OK** - não calcula o status do serviço Regras adicionais de cálculo de status podem ser configuradas nas opções de configuração avançada.
Descrição	Descrição do serviço.
Criado em	A hora em que o serviço foi criado; exibido ao editar um serviço existente.
Configuração avançada	Clique no rótulo *Configuração avançada* para exibir as opções de configuração avançada.

Configuração avançada



Parâmetro	Descrição
Regras adicionais	Clique em *Adicionar* para configurar regras adicionais de cálculo de status.
Definir status como	Defina o status do serviço como *OK* (padrão), *Não classificado*, *Informação*, *Aviso*, *Média*, *Alta* ou *Desastre* em caso de correspondência de condição.
Condição	Selecione a condição para serviços filhos diretos: **se pelo menos (N) serviços filhos tiverem status (Status) ou superior** **se pelo menos (N%) dos serviços filhos tiverem status (Status) ou superior** **se menos de (N) serviços filhos tiverem status (Status) ou inferior** **se menos de (N%) dos serviços filhos tiverem status (Status) ou inferior** **se o peso dos serviços filhos com status (Status) ou superior for pelo menos (W)** **se o peso dos serviços filhos com status (Status) ou superior for pelo menos (N%)** **se o peso dos serviços filhos com status (Status) ou inferior for menor que (W)** **se o peso dos serviços filhos com status (Status) ou inferior for menor que (N%)** Se várias condições forem especificadas e a situação corresponder a mais de uma condição, a maior gravidade será definida.
N (W)	Defina o valor de N ou W (1-100000), ou N% (1-100) na condição.
Status	Selecione o valor de *Status* na condição: *OK* (padrão), *Não classificado*, *Informação*, *Aviso*, *Média*, *Alta* ou *Desastre*.
Regra de propagação de status	Regra para propagar o status do serviço para o serviço pai: **Como está** - o status é propagado sem alteração **Aumentar em** - você pode aumentar o status propagado em 1 a 5 níveis de gravidade **Diminuir em** - você pode diminuir o status propagado em 1 a 5 níveis de gravidade **Ignorar este serviço** - o status não é propagado para o serviço pai **Status fixo** - o status é propagado estaticamente, ou seja, sempre o mesmo
Peso	Peso do serviço (inteiro no intervalo de 0 (padrão) a 1000000).

Note:
Regras adicionais de cálculo de status só podem ser usadas para aumentar o nível de gravidade em relação ao nível calculado de acordo com o parâmetro principal *Regra de cálculo de status*. Se, de acordo com as regras adicionais, o status deve ser Aviso, mas de acordo com a *Regra de cálculo de status* o status é Desastre - o serviço terá status Desastre.

A aba **Tags** contém tags de nível de serviço. As tags de nível de serviço são usadas para identificar um serviço. As tags desse tipo não são usadas para mapear problemas para o serviço (para isso, use as *Tags de problema* da primeira aba).

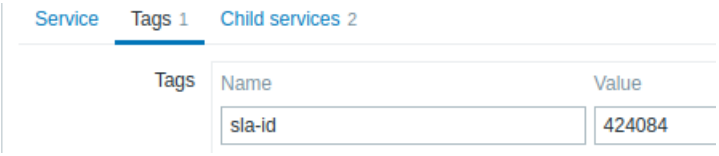
A aba **Serviços filhos** permite especificar serviços dependentes. Clique em *Adicionar* para adicionar um serviço da lista de serviços existentes. Se você quiser adicionar um novo serviço filho, salve este serviço primeiro, depois clique no ícone de mais ao lado do serviço que você acabou de criar.

Tags Existem dois tipos diferentes de tags em serviços:

- Tags de serviço
- Tags de problema

Tags de serviço

As tags de serviço são usadas para associar serviços a ações de serviço e SLAs. Essas tags são especificadas na aba *Tags* da configuração do serviço. Para o mapeamento de SLAs, é usada a lógica *OU*: um serviço será mapeado para um SLA se tiver pelo menos uma tag correspondente. Nas ações de serviço, as regras de mapeamento são configuráveis e podem usar lógica *E*, *OU* ou *E/OU*.

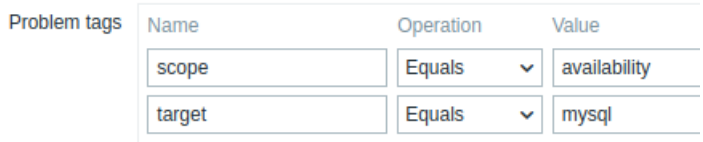


Tags de problema

As tags de problema são usadas para correlacionar problemas e serviços. Essas tags são especificadas na aba principal de configuração do serviço.

Apenas serviços filhos do nível mais baixo da hierarquia podem ter tags de problema definidas e serem correlacionados diretamente a problemas. Se as tags de problema corresponderem, o status do serviço mudará para o mesmo status do problema. No caso de vários problemas, um serviço terá o status do mais severo. O status de um serviço pai é então calculado com base nos status dos serviços filhos de acordo com as regras de cálculo de status.

Se várias tags forem especificadas, a lógica *E* é usada: um problema deve ter todas as tags especificadas na configuração do serviço para ser mapeado para o serviço.



Note:
Um problema no Zabbix herda tags de toda a cadeia de templates, hosts, items, cenários web e triggers. Qualquer uma dessas tags pode ser usada para correlacionar problemas a serviços.

Exemplo:

O problema *Web camera 3 is down* possui as tags `type:video-surveillance`, `floor:1` e `name:webcam-3` e status *Warning*

O serviço **Web camera 3** possui apenas a tag de problema especificada: `name:webcam-3`

O status do serviço mudará de *OK* para *Warning* quando este problema for detectado.

Se o serviço **Web camera 3** tivesse as tags de problema `name:webcam-3` e `floor:2`, seu status não seria alterado quando o problema fosse detectado, porque as condições seriam atendidas apenas parcialmente.

Note:
Os botões descritos abaixo são visíveis apenas quando a seção *Serviços* está no modo de Edição.

Modificando serviços existentes

Para editar um serviço existente, pressione o ícone de lápis ao lado do serviço.

Para clonar um serviço existente, pressione o ícone de lápis para abrir sua configuração e, em seguida, pressione o botão Clonar. Quando um serviço é clonado, seus links de pai são preservados, enquanto os links de filho não.

Para excluir um serviço, pressione o ícone x ao lado dele. Ao excluir um serviço pai, seus serviços filhos não serão excluídos e passarão para um nível acima na árvore de serviços (os filhos de 1º nível terão o mesmo nível do serviço pai excluído).

Dois botões abaixo da lista de serviços oferecem algumas opções de edição em massa:

- *Atualização em massa* - atualiza em massa as propriedades do serviço
- *Excluir* - exclui os serviços

Para usar essas opções, marque as caixas de seleção antes dos respectivos serviços e clique no botão desejado.

2 SLA

Visão geral Depois que os serviços são criados, você pode começar a monitorar se seu desempenho está de acordo com o acordo de nível de serviço (SLA).

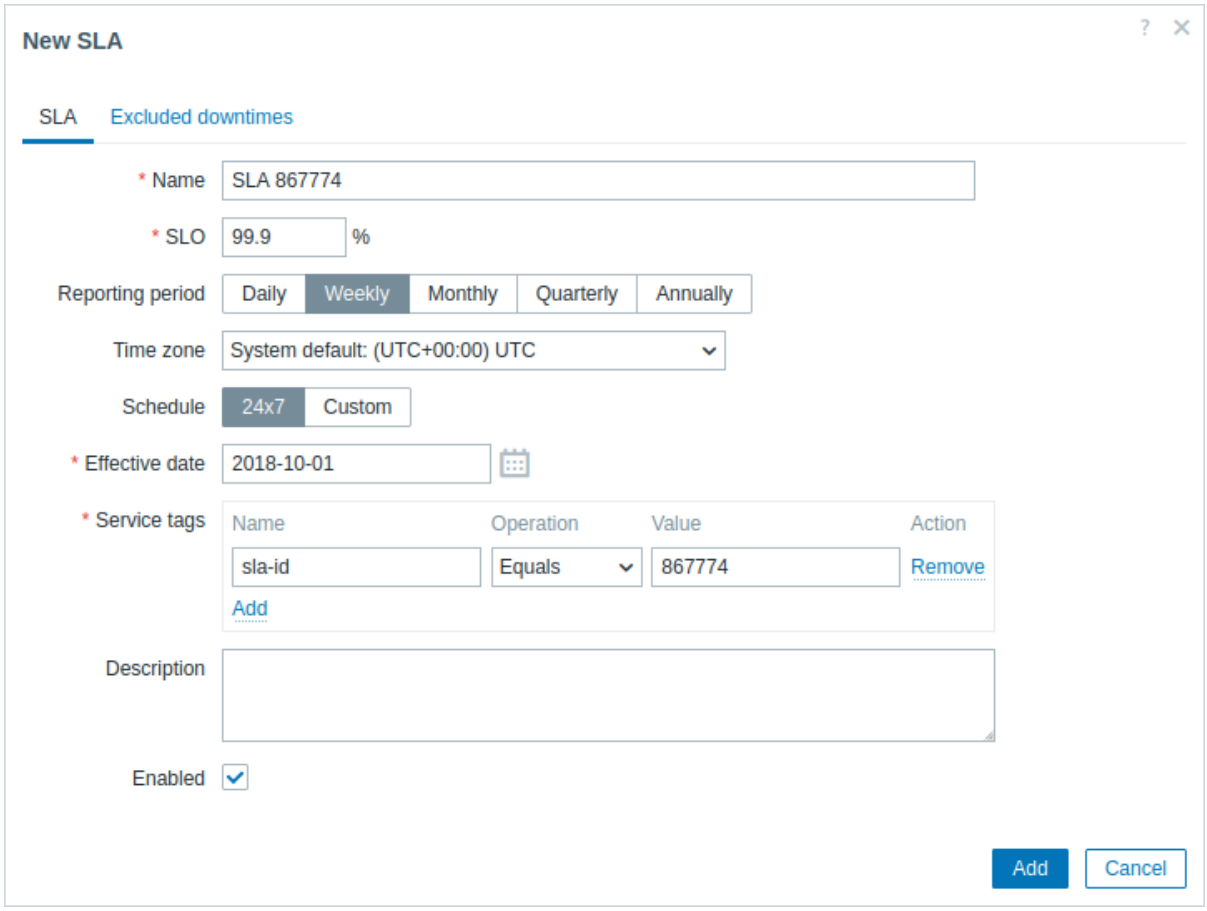
A seção de menu *Serviços->SLA* permite configurar SLAs para vários serviços. Um SLA no Zabbix define o objetivo de nível de serviço (SLO), a programação de tempo de atividade esperada e os períodos de inatividade planejados.

SLAs e serviços são correspondidos por tags de serviço. O mesmo SLA pode ser aplicado a vários serviços - o desempenho será medido separadamente para cada serviço correspondente. Um único serviço pode ter vários SLAs atribuídos - os dados de cada um dos SLAs serão exibidos separadamente.

Nos relatórios de SLA, o Zabbix fornece dados do Indicador de Nível de Serviço (SLI), que mede a disponibilidade real do serviço. Se um serviço atende às metas do SLA é determinado comparando o SLO (disponibilidade esperada em %) com o SLI (disponibilidade real em %).

Configuração Para criar um novo SLA, clique no botão *Criar SLA*.

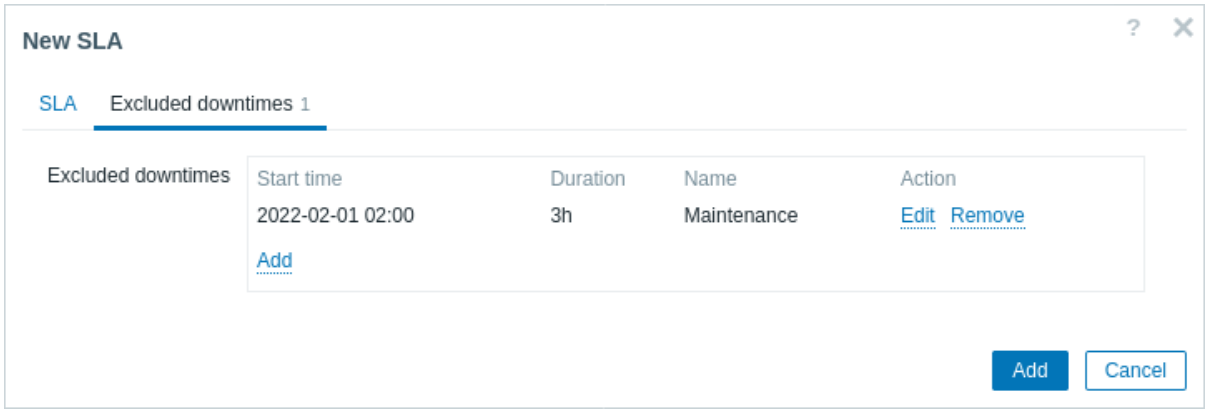
A aba **SLA** permite especificar os parâmetros gerais do SLA.



Parâmetro	Descrição
Nome	Insira o nome do SLA.
SLO	Insira o objetivo do nível de serviço (SLO) como porcentagem.
Período de relatório	Selecione o período de relatório do SLI usado para calcular os dados do SLI no Relatório de SLA: **Diário** - cada dia, das 00:00:00 às 23:59:59. **Semanal** - cada semana, do domingo 00:00:00 ao sábado 23:59:59. **Mensal** - cada mês, do primeiro dia 00:00:00 ao último dia 23:59:59. **Trimestral** - cada trimestre do calendário: Q1 (jan–mar), Q2 (abr–jun), Q3 (jul–set), Q4 (out–dez). **Anual** - cada ano do calendário, de 1º de janeiro 00:00:00 a 31 de dezembro 23:59:59.
Fuso horário	Selecione o fuso horário do SLA.
Agenda	Selecione a agenda do SLA - 24x7 ou personalizada.
Data efetiva	Selecione a data de início do cálculo do SLA.

Parâmetro	Descrição
Tags de serviço	Adicione tags de serviço para identificar os serviços aos quais este SLA deve ser aplicado. **Nome** - nome da tag de serviço, deve ser uma correspondência exata, sensível a maiúsculas e minúsculas. **Operação** - selecione *Igual* se o valor da tag deve corresponder exatamente (sensível a maiúsculas e minúsculas) ou *Contém* se parte do valor da tag deve corresponder (não sensível a maiúsculas e minúsculas). **Valor** - valor da tag de serviço a ser pesquisado de acordo com a operação selecionada. O SLA é aplicado a um serviço, se pelo menos uma tag de serviço corresponder.
Descrição	Adicione uma descrição para o SLA.
Habilitado	Marque a caixa de seleção para habilitar o cálculo do SLA.

A aba **Períodos de indisponibilidade excluídos** permite especificar períodos de indisponibilidade que são excluídos do cálculo do SLA.



Clique em *Adicionar* para configurar períodos de indisponibilidade excluídos, depois insira o nome do período, a data de início e a duração.

Relatórios de SLA Como um serviço se comporta em relação a um SLA é visível no relatório de SLA. Os relatórios de SLA podem ser visualizados:

- na seção *SLA* clicando no hiperlink do relatório de SLA;
- na seção *Serviços* clicando no nome do SLA na guia de informações;
- no widget do Dashboard *Relatório de SLA*.

Uma vez que um SLA esteja configurado, a guia *Info* na seção de serviços também exibirá algumas informações sobre o desempenho do serviço.

3 Exemplo de configuração

Visão geral Esta seção descreve uma configuração simples para monitorar o cluster de alta disponibilidade do Zabbix como um serviço.

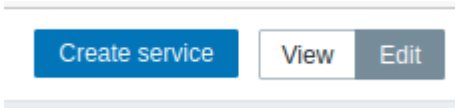
Pré-requisitos Antes de configurar o monitoramento de serviços, você precisa ter os hosts configurados:

- *HA node 1* com pelo menos um trigger e uma tag (de preferência definida no nível do trigger) `component:ha-node-1`
- *HA node 2* com pelo menos um trigger e uma tag (de preferência definida no nível do trigger) `component:ha-node-2`

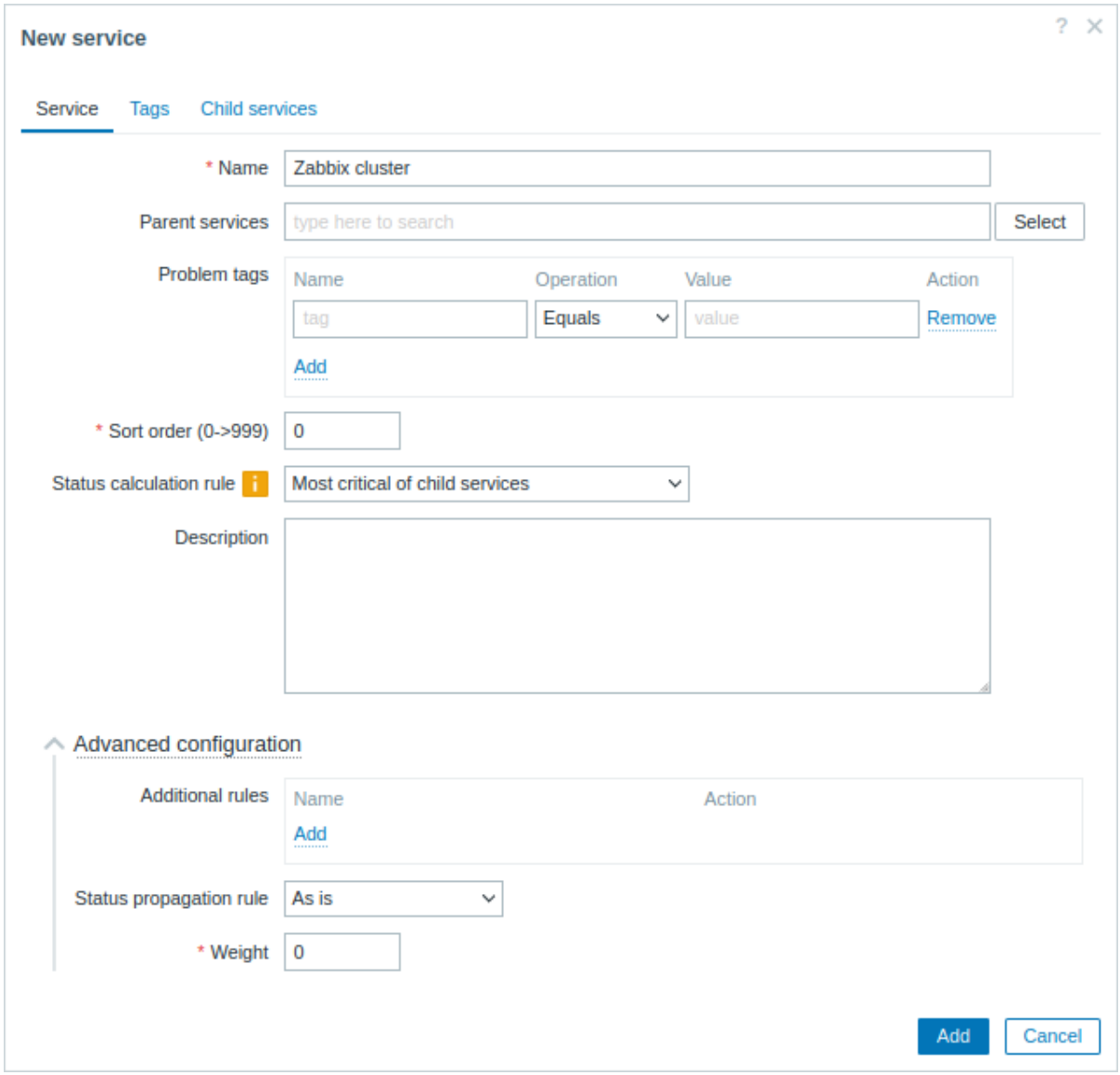
Árvore de serviços O próximo passo é construir a árvore de serviços. Neste exemplo, a infraestrutura é muito básica e consiste em três serviços: *Zabbix cluster* (pai) e dois serviços filhos *Zabbix server node 1* e *Zabbix server node 2*.

```
Zabbix cluster
|
|- Zabbix server node 1
|- Zabbix server node 2
```

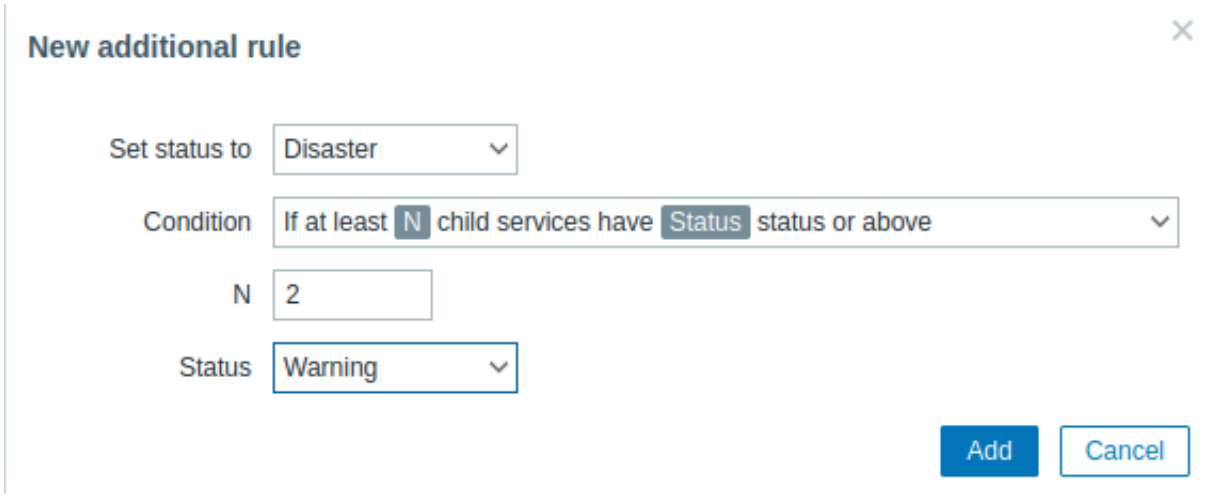
Na página Serviços, ative o modo *Editar* e pressione Criar serviço:



Na janela de configuração do serviço, insira o nome *Zabbix cluster* e clique no rótulo *Configuração avançada* para exibir as opções de configuração avançada.



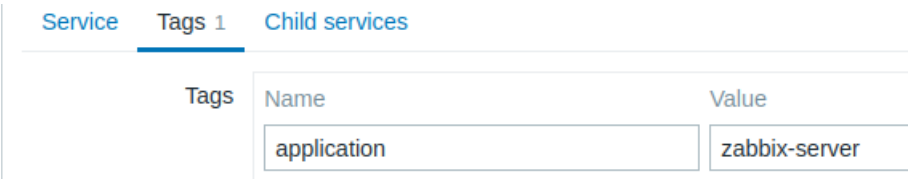
Configure a regra adicional:



O Zabbix cluster terá dois serviços filhos - um para cada nó HA. Se ambos os nós HA tiverem problemas de pelo menos status *Aviso*, o status do serviço pai deve ser definido como *Desastre*. Para isso, a regra adicional deve ser configurada como:

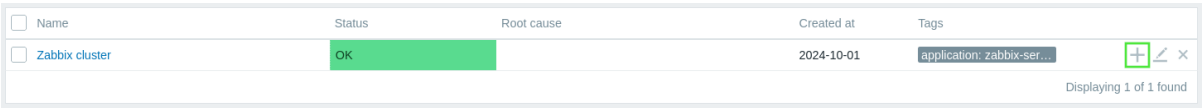
- Definir status como: Desastre
- Condição: Se pelo menos `N` serviços filhos tiverem status `Status` ou superior
- N: 2
- Status: Aviso

Altere para a guia *Tags* e adicione a tag `application:zabbix-server`. Esta tag será usada posteriormente para ações de serviço e relatórios de SLA.



Salve o novo serviço.

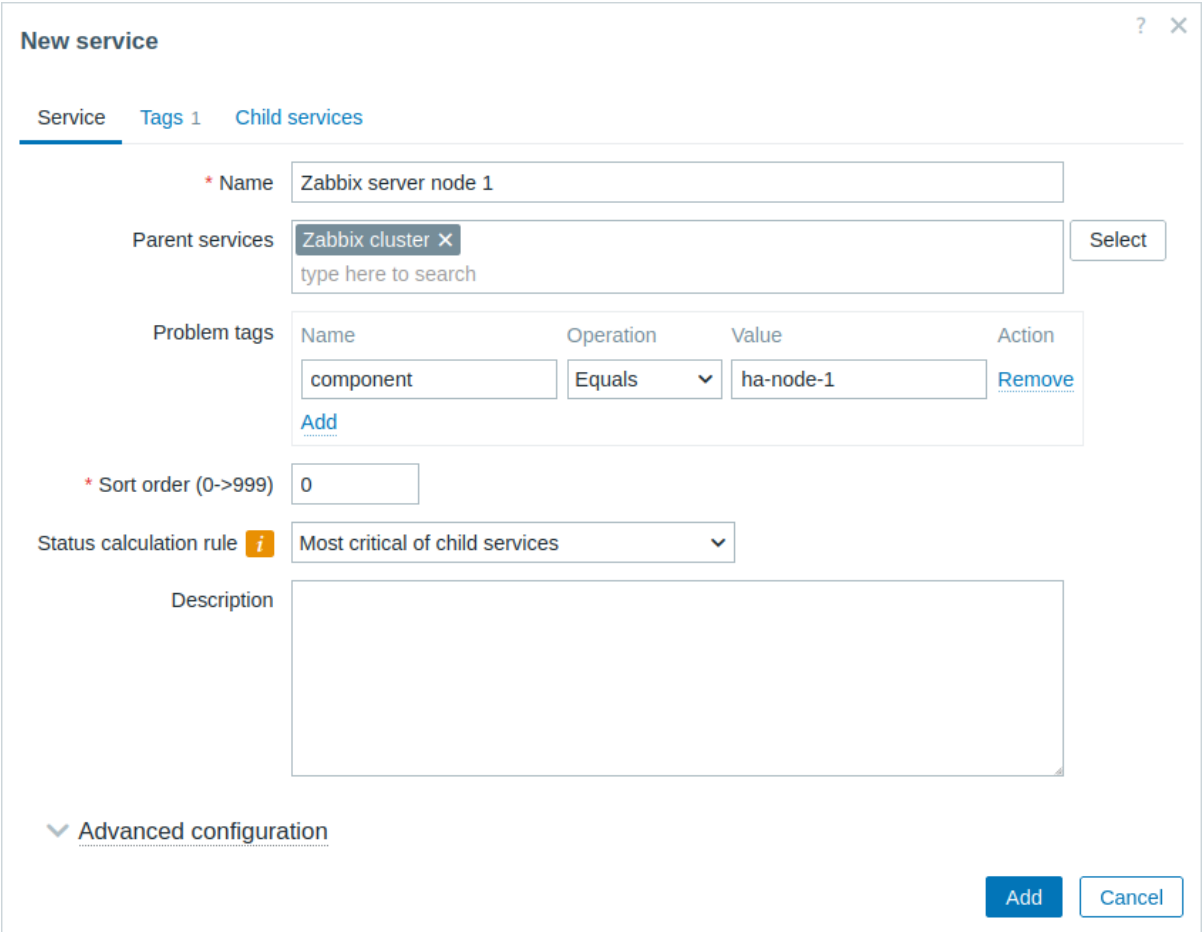
Para adicionar um serviço filho, pressione o ícone de mais ao lado do serviço do cluster Zabbix (o ícone é visível apenas no modo *Editar*).



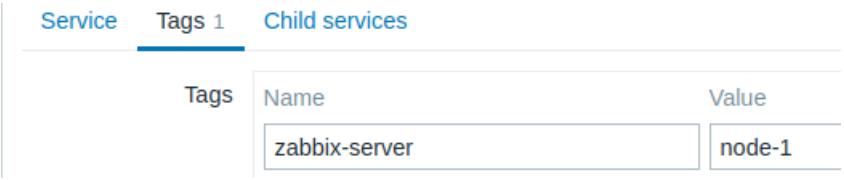
Na janela de configuração do serviço, insira o nome *Zabbix server node 1*. Observe que o parâmetro Serviços pai já está preenchido com *Zabbix cluster*.

A disponibilidade deste serviço é afetada por problemas no host *HA node 1*, marcados com a tag de problema `component:ha-node-1`. No parâmetro Tags de problema, insira:

- Nome: component
- Operação: Igual
- Valor: ha-node-1



Altere para a guia *Tags* e adicione uma tag de serviço: `zabbix-server:node-1`. Esta tag será usada posteriormente para ações de serviço e relatórios de SLA.



Salve o novo serviço.

Crie outro serviço filho do cluster Zabbix com o nome ”Zabbix server node 2”.

Defina as tags de problema como:

- Nome: component
- Operação: Igual
- Valor: ha-node-2

Altere para a guia *Tags* e adicione uma tag de serviço: `zabbix-server:node-2`.

Salve o novo serviço.

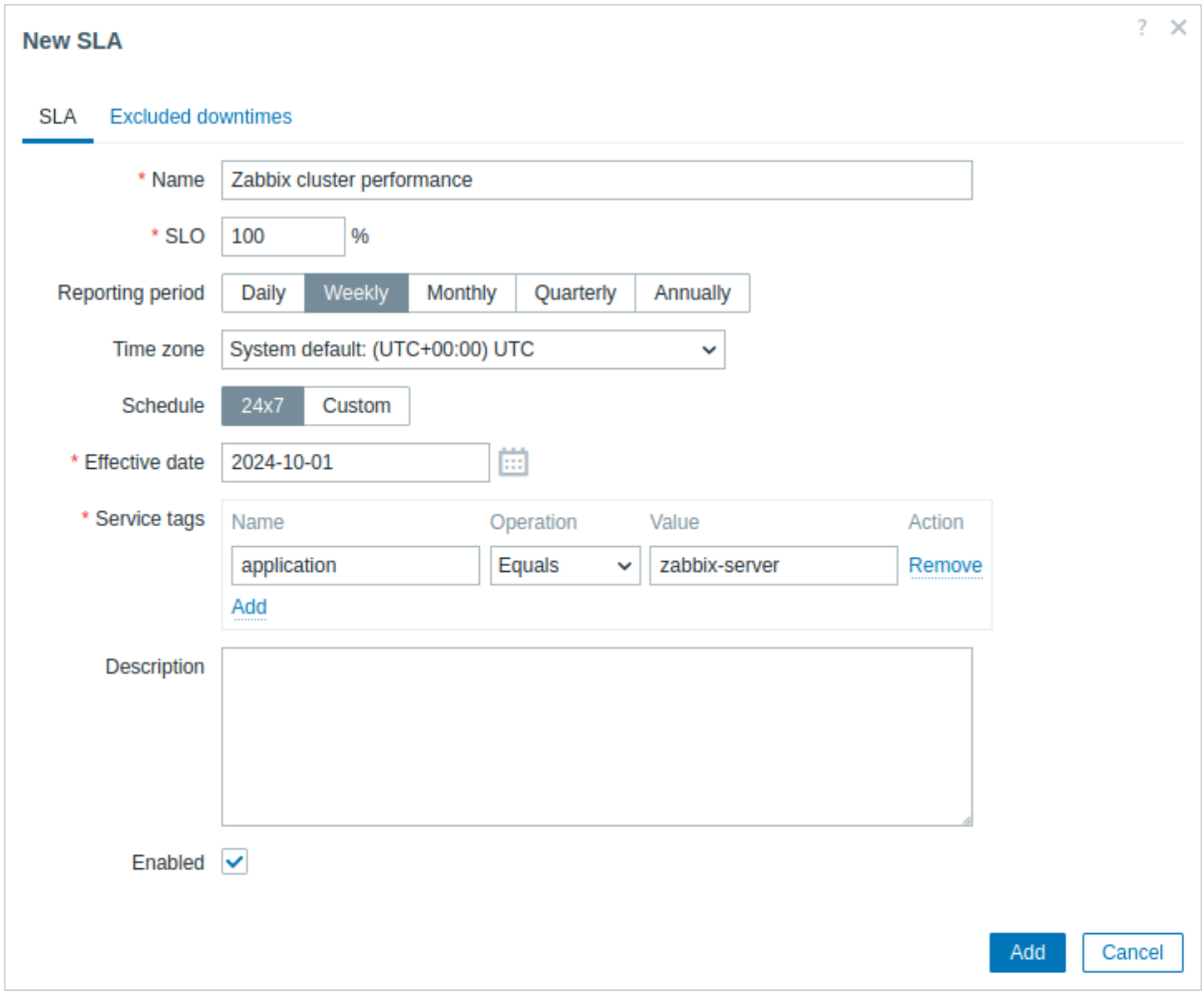
SLA Neste exemplo, o desempenho esperado do cluster Zabbix é de 100%, excluindo o período de manutenção semestral de uma hora.

Primeiro, você precisa adicionar um novo acordo de nível de serviço (SLA).

Vá para a seção de menu *Serviços->SLA* e pressione Criar SLA. Digite o nome *Desempenho do cluster Zabbix* e defina o SLO para 100%.

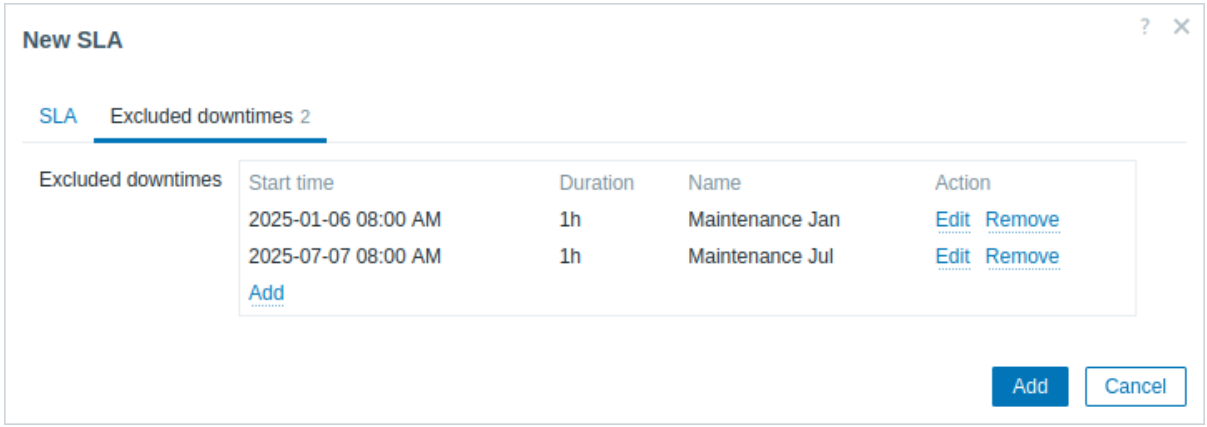
O serviço Zabbix cluster possui uma tag de serviço `application:zabbix-server`. Para usar este SLA para medir o desempenho do cluster Zabbix, no parâmetro *Tags de serviço*, especifique:

- Nome: application
- Operação: Igual
- Valor: zabbix-server



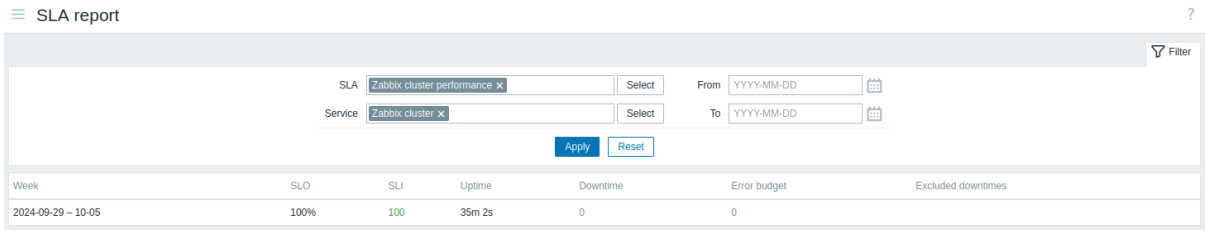
Em um ambiente real, você também pode atualizar o período de relatório desejado, o fuso horário e a data de início ou alterar a programação de 24/7 para personalizada. Para este exemplo, as configurações padrão são suficientes.

Altere para a guia *Períodos de inatividade excluídos* e adicione períodos de inatividade para períodos de manutenção agendada para excluir esses períodos do cálculo do SLA. Na seção Períodos de inatividade excluídos, pressione o link Adicionar, insira o nome do período de inatividade, o horário de início planejado e a duração.

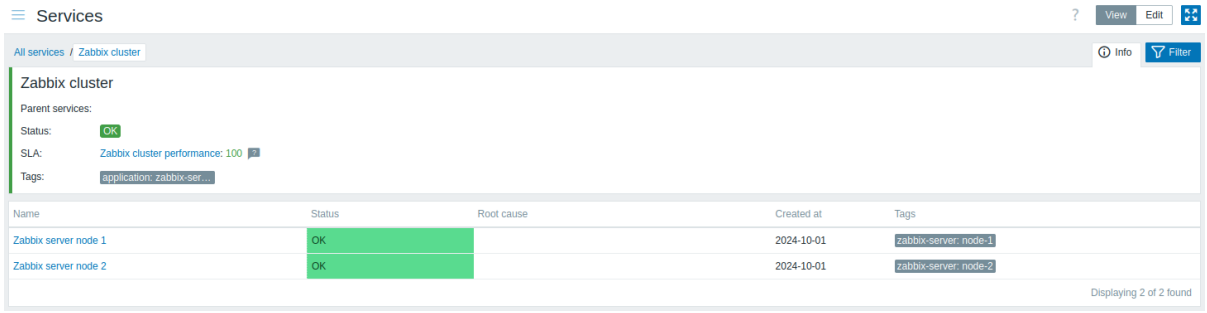


Pressione Adicionar para salvar o novo SLA.

Altere para a seção de relatórios de SLA para visualizar o relatório de SLA para o cluster Zabbix.



As informações do SLA também podem ser verificadas na seção *Serviços*.



9. Monitoramento web

Visão geral Com o Zabbix você pode verificar vários aspectos de disponibilidade de sites.

> **Attention:**
> Para realizar o monitoramento web, o Zabbix server deve ser inicialmente configurado com suporte a cURL (libcurl).

Para ativar o monitoramento web, você precisa definir cenários web. Um cenário web consiste em uma ou várias requisições HTTP ou "etapas". As etapas são executadas periodicamente pelo Zabbix server em uma ordem pré-definida. Se um host for monitorado por um proxy, as etapas são executadas pelo proxy.

Os cenários web são vinculados a hosts/templates da mesma forma que items, triggers, etc. Isso significa que cenários web também podem ser criados em nível de template e, em seguida, aplicados a vários hosts de uma só vez.

As seguintes informações são coletadas em qualquer cenário web:

- velocidade média de download por segundo para todas as etapas do cenário inteiro
- número da etapa que falhou

- última mensagem de erro

As seguintes informações são coletadas em qualquer etapa de cenário web:

- velocidade de download por segundo
- tempo de resposta
- código de resposta

Para mais detalhes, veja itens de monitoramento web.

Os dados coletados a partir da execução de cenários web são mantidos no banco de dados. Os dados são usados automaticamente para gráficos, triggers e notificações.

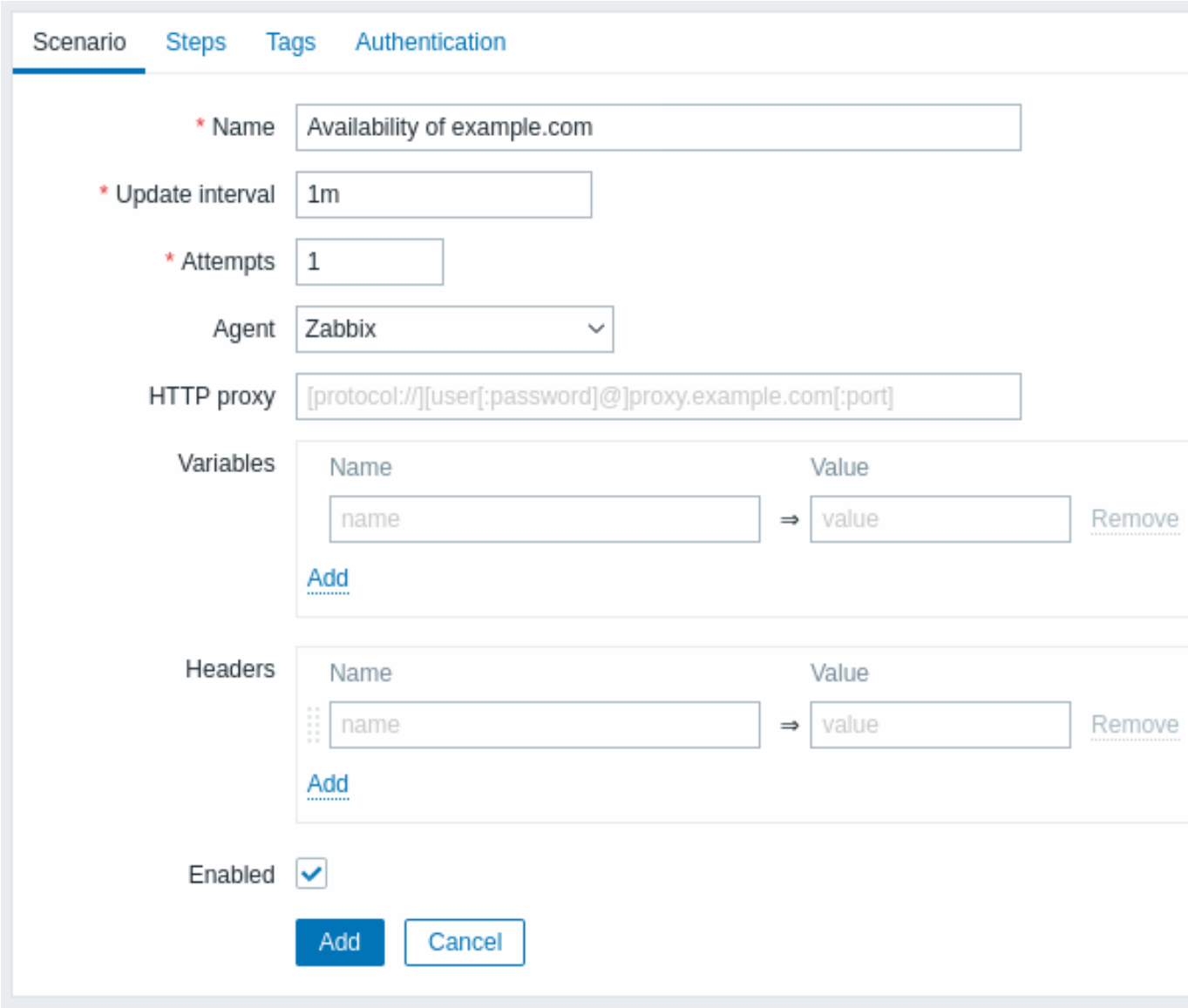
O Zabbix também pode verificar se uma página HTML recuperada contém uma string pré-definida. Ele pode executar um login simulado e seguir um caminho de cliques simulados do mouse na página.

O monitoramento web do Zabbix suporta HTTP e HTTPS. Ao executar um cenário web, o Zabbix seguirá redirecionamentos opcionalmente (veja a opção *Seguir redirecionamentos* abaixo). O número máximo de redirecionamentos é definido em 10 (usando a opção cURL CURLOPT_MAXREDIRS). Todos os cookies são preservados durante a execução de um único cenário.

Configurando um cenário web Para configurar um cenário web:

- Vá para: *Coleta de dados* → *Hosts* (ou *Templates*)
- Clique em *Web* na linha do host/template
- Clique em *Criar cenário web* à direita (ou no nome do cenário para editar um cenário existente)
- Insira os parâmetros do cenário no formulário

A aba **Cenário** permite configurar os parâmetros gerais de um cenário web.



Todos os campos obrigatórios estão marcados com um asterisco vermelho.

Parâmetros do cenário:

Parâmetro	Descrição
Nome	Nome único do cenário. Macros de usuário são suportadas. *Observação*: se macros de usuário forem usadas, essas macros permanecerão não resolvidas nos nomes dos itens de monitoramento web.
Intervalo de atualização	Com que frequência o cenário será executado. Sufixos de tempo são suportados, por exemplo, 30s, 1m, 2h, 1d. Macros de usuário são suportadas. *Observação*: se uma macro de usuário for usada e seu valor for alterado (por exemplo, 5m → 30s), a próxima verificação será executada de acordo com o valor anterior (mais distante no futuro com os valores de exemplo). Novos cenários web serão verificados em até 60 segundos após sua criação.
Tentativas	O número de tentativas para executar as etapas do cenário web. Em caso de problemas de rede (timeout, sem conectividade, etc), o Zabbix pode repetir a execução de uma etapa várias vezes. O valor definido afetará igualmente cada etapa do cenário. Até 10 tentativas podem ser especificadas, o valor padrão é 1. *Observação*: o Zabbix não repetirá uma etapa devido a um código de resposta errado ou à incompatibilidade de uma string obrigatória.
Agent	Selecione um agent cliente. O Zabbix irá simular ser o navegador selecionado. Isso é útil quando um site retorna conteúdo diferente para navegadores diferentes. Macros de usuário podem ser usadas neste campo.
Proxy HTTP	Você pode especificar um proxy HTTP para usar, usando o formato `[protocol://][username[:password]@]proxy.example.com[:port]`. Isso define a opção cURL CURLOPT_PROXY. O prefixo opcional `protocol://` pode ser usado para especificar protocolos de proxy alternativos (o suporte ao prefixo de protocolo foi adicionado no cURL 7.21.7). Sem protocolo especificado, o proxy será tratado como um proxy HTTP. Por padrão, a porta 1080 será usada. Se especificado, o proxy sobrescreverá as variáveis de ambiente relacionadas ao proxy, como http_proxy, HTTPS_PROXY. Se não especificado, o proxy não sobrescreverá as variáveis de ambiente relacionadas ao proxy. O valor inserido é passado "como está", sem verificação de sanidade. Você também pode inserir um endereço de proxy SOCKS. Se você especificar o protocolo errado, a conexão falhará e o item se tornará não suportado. *Observação*: apenas autenticação simples é suportada com proxy HTTP. Macros de usuário podem ser usadas neste campo.

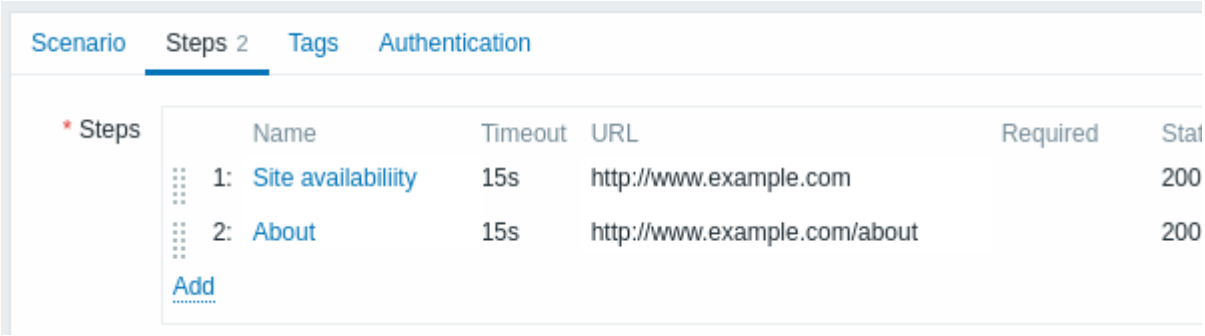
Parâmetro	Descrição
Variáveis	Variáveis que podem ser usadas nas etapas do cenário (URL, variáveis post). Elas têm o seguinte formato: **{macro1}**=valor1 **{macro2}**=valor2 **{macro3}**=regex:<expressão regular> **{macro4}**=jsonpath:<jsonpath> **{macro5}**=xmlxpath:<xmlxpath> **{macro6}**={{macro}.function()} (veja funções de macro). *Observação*: o uso de JSONPath, XML XPath e funções de macro em variáveis é suportado desde o Zabbix 7.0.4. Por exemplo: {username}=Alexei {password}=kj3h5kJ34bd {hostid}=regex:hostid is ([0-9]+) {url}=jsonpath:$.host_url {status}=xmlxpath://host/response/status {newvar}={{myvar}.btoa()} As macros podem então ser referenciadas nas etapas como {username}, {password}, {hostid}, etc. O Zabbix irá substituí-las automaticamente pelos valores reais. Observe que variáveis com `regex:` precisam de uma etapa para obter o valor da expressão regular, para que o valor extraído só possa ser aplicado à etapa seguinte. Se a parte do valor começar com `regex:`, então a parte após ela é tratada como uma expressão regular que pesquisa a página web e, se encontrada, armazena a correspondência na variável. Pelo menos um subgrupo deve estar presente para que o valor correspondente possa ser extraído. Macros de usuário e {HOST.*} macros são suportadas. As variáveis são automaticamente codificadas em URL quando usadas em campos de consulta ou dados de formulário para variáveis post, mas devem ser codificadas manualmente em URL quando usadas em post bruto ou diretamente na URL.
Cabeçalhos	Cabeçalhos HTTP são usados ao realizar uma requisição. Cabeçalhos padrão e personalizados podem ser usados. Cabeçalhos serão atribuídos usando as configurações padrão dependendo do tipo de Agent selecionado em uma lista suspensa no nível do cenário, e serão aplicados a todas as etapas, a menos que sejam definidos de forma personalizada em uma etapa. **Deve-se observar que definir o cabeçalho em uma etapa descarta automaticamente todos os cabeçalhos definidos anteriormente, exceto o cabeçalho padrão que é atribuído selecionando o 'User-Agent' em uma lista suspensa no nível do cenário.** No entanto, mesmo o cabeçalho padrão 'User-Agent' pode ser sobrescrito especificando-o em uma etapa. Para desabilitar o cabeçalho no nível do cenário, o cabeçalho deve ser nomeado e atribuído sem valor no nível da etapa. Cabeçalhos devem ser listados usando a mesma sintaxe que apareceriam no protocolo HTTP, opcionalmente usando alguns recursos adicionais suportados pela opção cURL CURLOPT_HTTPHEADER. Por exemplo: Accept-Charset=utf-8 Accept-Language=en-US Content-Type=application/xml; charset=utf-8 Macros de usuário e {HOST.*} macros são suportadas.
Ativado	O cenário está ativo se esta caixa estiver marcada, caso contrário, está desativado.

Observe que ao editar um cenário existente, dois botões extras estão disponíveis no formulário:

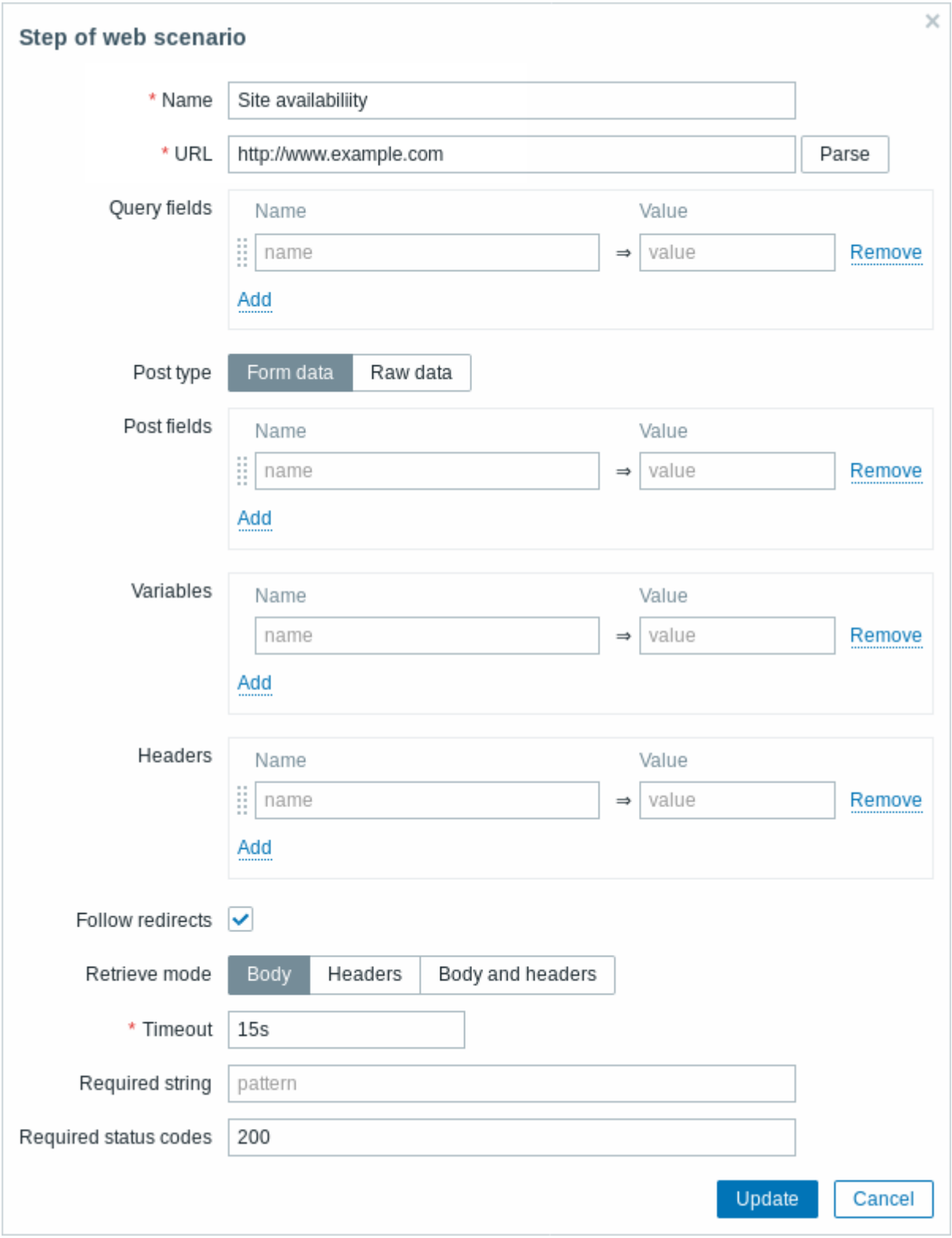
Clone	Cria outro cenário com base nas propriedades do cenário existente.
Clear history and trends	Exclui o histórico e os dados de tendência do cenário. Isso fará com que o server execute o cenário imediatamente após a exclusão dos dados.

Note:
Se o campo *Proxy HTTP* for deixado em branco, outra maneira de usar um proxy HTTP é definir variáveis de ambiente relacionadas ao proxy.
Para verificações HTTP - defina a variável de ambiente **http_proxy** para o usuário do server Zabbix. Por exemplo, `http_proxy=http://proxy_ip:proxy_port`.
Para verificações HTTPS - defina a variável de ambiente **HTTPS_PROXY**. Por exemplo, `HTTPS_PROXY=http://proxy_ip:proxy_port`. Mais detalhes estão disponíveis executando o comando shell: *# man curl*.

A aba **Etapas** permite configurar as etapas do cenário web. Para adicionar uma etapa ao cenário web, clique em *Adicionar* no bloco *Etapas*.



Note:
Macros de usuário secretas não devem ser usadas em URLs, pois serão resolvidas como "******".



Configurando etapas

Parâmetros da etapa:

Parâmetro	Descrição
Nome	Nome único da etapa. Macros de usuário são suportadas. *Nota* que se macros de usuário forem usadas, essas macros permanecerão não resolvidas nos nomes dos itens de monitoramento web.

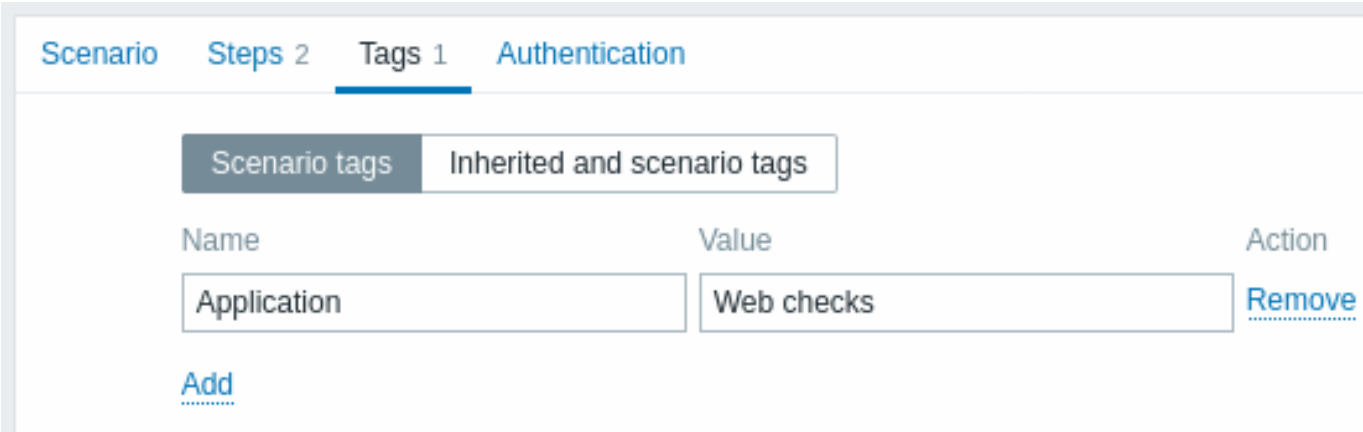
Parâmetro	Descrição
URL	URL para conectar e recuperar dados. Por exemplo: https://www.exemplo.com.br http://www.exemplo.com.br/download Nomes de domínio podem ser especificados em caracteres Unicode. Eles são automaticamente convertidos para ASCII em punycode ao executar a etapa do cenário web. O botão *Analisar* pode ser usado para separar campos de consulta opcionais (como ?name=Admin&password=mypassword) da URL, movendo os atributos e valores para *Campos de consulta* para codificação automática da URL. Variáveis podem ser usadas na URL, usando a sintaxe {macro}. Variáveis podem ser codificadas manualmente na URL usando a sintaxe {{macro}.urlencode()}. Macros de usuário e {HOST.*} macros são suportadas. Limitado a 2048 caracteres.
Campos de consulta	Variáveis HTTP GET para a URL. Especificadas como pares de atributo e valor. Os valores são codificados automaticamente na URL. Valores de variáveis de cenário, macros de usuário ou macros {HOST.*} são resolvidos e então codificados automaticamente na URL. Usar a sintaxe {{macro}.urlencode()} irá codificá-los duas vezes. Macros de usuário e {HOST.*} macros são suportadas.
Post	Variáveis HTTP POST. No modo **Dados do formulário**, especificadas como pares de atributo e valor. Os valores são codificados automaticamente na URL. Valores de variáveis de cenário, macros de usuário ou macros {HOST.*} são resolvidos e então codificados automaticamente na URL. No modo **Dados brutos**, atributos/valores são exibidos em uma única linha e concatenados com o símbolo **&**. Valores brutos podem ser codificados/decodificados manualmente na URL usando a sintaxe {{macro}.urlencode()} ou {{macro}.urldecode()}. Por exemplo: id=2345&userid={user} Se {user} for definida como uma variável do cenário web, ela será substituída por seu valor quando a etapa for executada. Se você deseja codificar a variável na URL, substitua {user} por {{user}.urlencode()}. Macros de usuário e {HOST.*} macros são suportadas.
Variáveis	Variáveis no nível da etapa que podem ser usadas para funções GET e POST. Especificadas como pares de atributo e valor. Variáveis no nível da etapa substituem variáveis no nível do cenário ou variáveis da etapa anterior. No entanto, o valor de uma variável no nível da etapa só afeta a etapa seguinte (e não a etapa atual). Elas têm o seguinte formato: **{macro}**=valor **{macro}**=regex:<expressão regular> Para mais informações, veja a descrição da variável no nível do cenário. As variáveis são automaticamente codificadas na URL quando usadas em campos de consulta ou dados de formulário para variáveis post, mas devem ser codificadas manualmente na URL quando usadas em post bruto ou diretamente na URL.
Cabeçalhos	Cabeçalhos HTTP personalizados que serão enviados ao executar uma requisição. Especificados como pares de atributo e valor. Um cabeçalho definido no nível da etapa será usado para aquela etapa específica. **Deve-se observar que definir o cabeçalho no nível da etapa descarta automaticamente todos os cabeçalhos definidos anteriormente, exceto o cabeçalho padrão que é atribuído ao selecionar o 'User-Agent' em uma lista suspensa no nível do cenário.** No entanto, até mesmo o cabeçalho padrão 'User-Agent' pode ser substituído ao especificá-lo no nível da etapa. Por exemplo, atribuir o nome a um cabeçalho, mas não definir valor, irá desativar o cabeçalho padrão no nível do cenário. Macros de usuário e macros {HOST.*} são suportadas. Isso define a opção cURL CURLOPT_HTTPHEADER.
Seguir redirecionamentos	Marque a caixa de seleção para seguir redirecionamentos HTTP. Isso define a opção cURL CURLOPT_FOLLOWLOCATION.
Modo de recuperação	Selecione o modo de recuperação: **Corpo** - recupera apenas o corpo da resposta HTTP **Cabeçalhos** - recupera apenas os cabeçalhos da resposta HTTP **Corpo e cabeçalhos** - recupera corpo e cabeçalhos da resposta HTTP

Parâmetro	Descrição
Timeout	O Zabbix não gastará mais do que o tempo definido no processamento da URL (de um segundo até o máximo de 1 hora). Na verdade, este parâmetro define o tempo máximo para fazer a conexão com a URL e o tempo máximo para executar uma requisição HTTP. Portanto, o Zabbix não gastará mais do que **2 x Timeout** segundos na etapa. Sufixos de tempo são suportados, por exemplo, 30s, 1m, 1h. Macros de usuário são suportadas.
String obrigatória	Padrão de expressão regular obrigatório. A menos que o conteúdo recuperado (HTML) corresponda ao padrão obrigatório, a etapa falhará. Se vazio, nenhuma verificação de string obrigatória é realizada. Por exemplo: Página inicial do Zabbix Bem-vindo.*admin *Nota*: Referenciar expressões regulares criadas no frontend do Zabbix não é suportado neste campo. Macros de usuário e {HOST.*} macros são suportadas.
Códigos de status obrigatórios	Lista de códigos de status HTTP esperados. Se o Zabbix receber um código que não está na lista, a etapa falhará. Se vazio, nenhuma verificação de códigos de status é realizada. Por exemplo: 200,201,210-299 Macros de usuário são suportadas.

Note:
Quaisquer alterações nas etapas do cenário web só serão salvas quando todo o cenário for salvo.

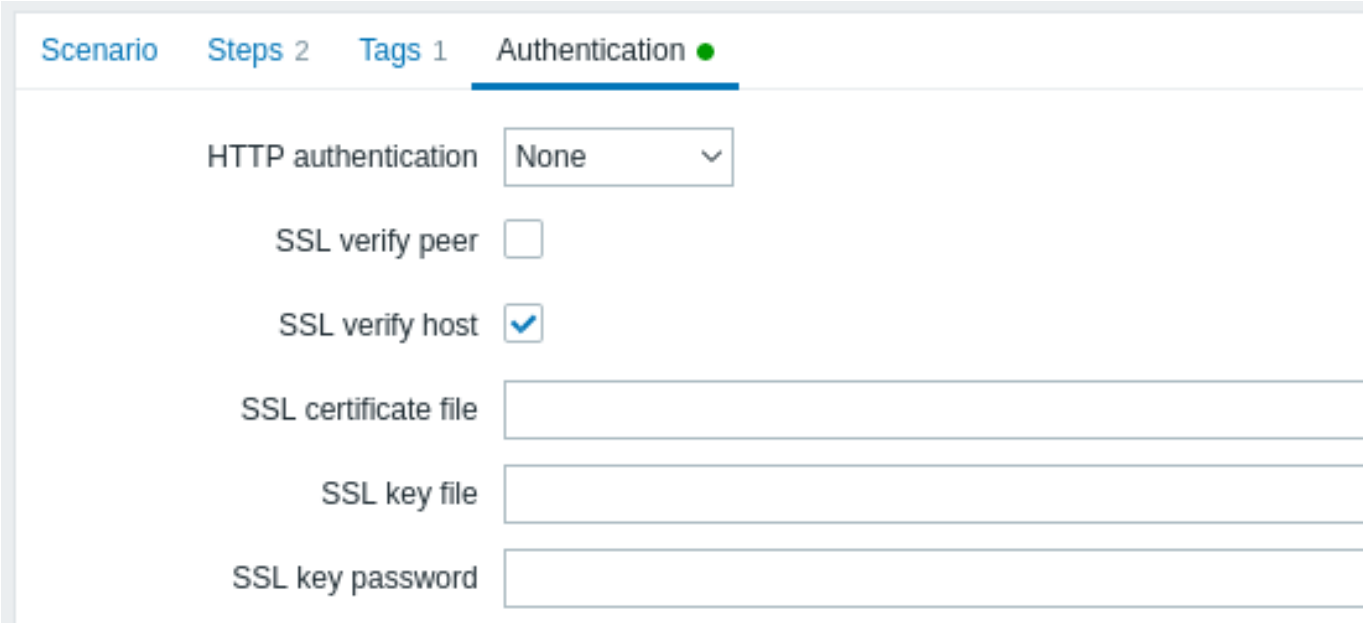
Veja também um exemplo real de como as etapas de monitoramento web podem ser configuradas.

Configurando etiquetas A aba de **Etiquetas** permite definir etiquetas no nível de cenário.



A etiquetagem permite filtrar cenários web e itens de monitoramento web.

Configurando autenticação A aba **Autenticação** permite configurar as opções de autenticação do cenário. Um ponto verde ao lado do nome da aba indica que algum tipo de autenticação HTTP está habilitado.



Parâmetros de autenticação:

Parâmetro	Descrição
Autenticação HTTP	Selecione a opção de autenticação: **Nenhuma** - nenhuma autenticação é usada; **Básica** - autenticação básica é usada; **NTLM** - autenticação NTLM (Windows NT LAN Manager) é usada; **Kerberos** - autenticação Kerberos é usada (veja também: Configurando Kerberos com Zabbix); **Digest** - autenticação Digest é usada.
Usuário	Informe o nome do usuário (até 255 caracteres). Este campo está disponível se *Autenticação HTTP* estiver definida como Básica, NTLM, Kerberos ou Digest. Macros de usuário são suportadas.
Senha	Informe a senha do usuário (até 255 caracteres). Este campo está disponível se *Autenticação HTTP* estiver definida como Básica, NTLM, Kerberos ou Digest. Macros de usuário são suportadas.
Verificar SSL peer	Marque a caixa de seleção para verificar o certificado SSL do servidor web. O certificado do servidor será automaticamente obtido do local da autoridade certificadora (CA) do sistema. Você pode sobrescrever o local dos arquivos CA usando o parâmetro de configuração do Zabbix server ou proxy SSLCALocation. Isso define a opção cURL CURLOPT_SSL_VERIFYPEER.
Verificar host SSL	Marque a caixa de seleção para verificar se o campo *Common Name* ou o campo *Subject Alternate Name* do certificado do servidor web corresponde. Isso define a opção cURL CURLOPT_SSL_VERIFYHOST.
Arquivo de certificado SSL	Nome do arquivo de certificado SSL usado para autenticação do cliente. O arquivo de certificado deve estar no formato PEM[1]. Se o arquivo de certificado também contiver a chave privada, deixe o campo *Arquivo de chave SSL* em branco. Se a chave estiver criptografada, especifique a senha no campo *Senha da chave SSL*. O diretório que contém este arquivo é especificado pelo parâmetro de configuração do Zabbix server ou proxy SSLCertLocation. Macros `HOST.*` e macros de usuário podem ser usadas neste campo. Isso define a opção cURL CURLOPT_SSLCERT.
Arquivo de chave SSL	Nome do arquivo de chave privada SSL usado para autenticação do cliente. O arquivo de chave privada deve estar no formato PEM[1]. O diretório que contém este arquivo é especificado pelo parâmetro de configuração do Zabbix server ou proxy SSLKeyLocation. Macros `HOST.*` e macros de usuário podem ser usadas neste campo. Isso define a opção cURL CURLOPT_SSLKEY.
Senha da chave SSL	Senha do arquivo de chave privada SSL. Macros de usuário podem ser usadas neste campo. Isso define a opção cURL CURLOPT_KEYPASSWD.

Attention:
[1] O Zabbix suporta apenas arquivos de certificado e chave privada no formato PEM. Caso você tenha seu certificado e chave privada em um arquivo no formato PKCS #12 (normalmente com extensão *.p12 ou *.pfx), você pode gerar o arquivo PEM a partir dele usando os seguintes comandos:

```
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:
O Zabbix server detecta alterações nos certificados sem necessidade de reinicialização.

Note:
Se você tiver o certificado do cliente e a chave privada em um único arquivo, basta especificá-lo no campo "Arquivo de certificado SSL" e deixar o campo "Arquivo de chave SSL" em branco. O certificado e a chave ainda devem estar no formato PEM. Combinar o certificado e a chave é fácil:

```
cat client.crt client.key > client.pem
```

Exibição Para visualizar os cenários web configurados para um host, vá para *Monitoramento* → *Hosts*, localize o host na lista e clique no hiperlink *Web* na última coluna. Clique no nome do cenário para obter informações detalhadas.

Uma visão geral dos cenários web também pode ser exibida em *Dashboards* pelo widget de monitoramento web.

Os resultados recentes da execução do cenário web estão disponíveis na seção *Monitoramento → Últimos dados*.

Monitoramento estendido Às vezes, é necessário registrar o conteúdo da página HTML recebida. Isso é especialmente útil se alguma etapa do cenário web falhar. O nível de depuração 5 (trace) serve para esse propósito. Esse nível pode ser definido nos arquivos de configuração do server e do proxy ou usando uma opção de controle em tempo de execução (`-R log_level_increase="http poller,N"`, onde N é o número do processo). Os exemplos a seguir demonstram como o monitoramento estendido pode ser iniciado, desde que o nível de depuração 4 já esteja definido:

```
# Aumentar o nível de log de todos os http pollers:
zabbix_server -R log_level_increase="http poller"

# Aumentar o nível de log do segundo http poller:
zabbix_server -R log_level_increase="http poller,2"
```

Se o monitoramento web estendido não for necessário, ele pode ser interrompido usando a opção `-R log_level_decrease`.

1 Itens de monitoramento web

Visão geral

Alguns novos itens são automaticamente adicionados ao monitoramento quando cenários web são criados.

Todos os itens herdam etiquetas do cenário web.

Itens de cenário

Assim que um cenário é criado, o Zabbix adiciona automaticamente os seguintes itens para monitoramento.

Item	Descrição
Velocidade de download do cenário <Scenario>	Este item coletará informações sobre a velocidade de download (bytes por segundo) de todo o cenário, ou seja, a média de todos os passos. Chave do item: web.test.in[Scenario,,bps] Tipo: *Numérico(flutuante)*
Passo com falha do cenário <Scenario>	Este item exibirá o número do passo que falhou no cenário. Se todos os passos forem executados com sucesso, 0 será retornado. Chave do item: web.test.fail[Scenario] Tipo: *Numérico(não assinado)*
Última mensagem de erro do cenário <Scenario>	Este item retorna o texto da última mensagem de erro do cenário. Um novo valor é armazenado apenas se o cenário tiver um passo com falha. Se todos os passos estiverem ok, nenhum novo valor será coletado. Chave do item: web.test.error[Scenario] Tipo: *Caracter*

O nome real do cenário será usado em vez de ”Scenario”.

Note:
Se o nome do cenário contiver macros de usuário, essas macros permanecerão não resolvidas nos nomes dos itens de monitoramento web.

 Se o nome do cenário começar com aspas duplas ou contiver uma vírgula ou colchete, ele será devidamente citado nas chaves dos itens. Nos outros casos, nenhuma citação adicional será realizada.

Note:
Os itens de monitoramento web são adicionados com 30 dias de histórico e 90 dias de retenção de tendências.

Esses itens podem ser usados para criar triggers e definir condições de notificação.

Exemplo 1

Para criar um gatilho ”Cenário web falhou”, você pode definir uma expressão de gatilho:

```
last(/host/web.test.fail[Cenário])<>0
```

Certifique-se de substituir 'Cenário' com o nome real de seu cenário.

Exemplo 2

Para criar um gatilho "Cenário web falhou" com uma descrição de problema útil no nome do gatilho, você pode definir um gatilho com nome:

```
Cenário web "Cenário" falhou: {ITEM.VALUE}
```

e expressão de gatilho:

```
length(last(/host/web.test.error[Cenário]))>0 and last(/host/web.test.fail[Cenário])>0
```

Certifique-se de substituir 'Cenário' com o nome real de seu cenário.

Exemplo 3

Para criar um gatilho "Aplicação web está lenta", você pode definir uma expressão de gatilho:

```
last(/host/web.test.in[Cenário,,bps])<10000
```

Certifique-se de substituir 'Cenário' com o nome real de seu cenário.

Itens de etapa de cenário

Assim que uma etapa é criada, o Zabbix automaticamente adiciona os seguintes itens para monitoramento.

Item	Descrição
Velocidade de download para a etapa <Etapa> do cenário <Cenário>	Este item coletará informações sobre a velocidade de download (bytes por segundo) da etapa. Chave de item: web.test.in[Cenário,Step,bps] Tipo: *Numérico(float)*
Tempo de resposta para a etapa <Etapa> do cenário <Cenário>	Este item coletará informações sobre o tempo de resposta da etapa em segundos. O tempo de resposta é contado do início da requisição até todas as informações terem sido transferidas. Chave de item: web.test.time[Cenário,Step,resp] Tipo: *Numérico(float)*
Código de resposta para a etapa <Etapa> do cenário <Cenário>	Este item coletará códigos de resposta da etapa. Chave de item: web.test.rspcode[Cenário,Step] Tipo: *Numérico(unsigned)*

O cenário atual e nomes de etapa serão usados em vez de "Cenário" e "Etapa" respectivamente.

Note:
Os itens de monitoramento web são adicionados com um período de retenção de 30 dias para histórico e 90 dias para tendência.

Note:
Se o nome do cenário iniciar com uma aspa dupla ou contém vírgula ou colchete, ele será apropriadamente quotado na chaves de item. Em outros casos ele não será quotado.

Estes itens podem ser usados para criar gatilhos e definir condições de notificação. Por exemplo, para criar um gatilho "Autenticação Zabbix GUI está muito lenta", você pode definir uma expressão de gatilho:

```
last(/zabbix/web.test.time[ZABBIX GUI,Login,resp])>3
```

2 Cenário real

Visão geral

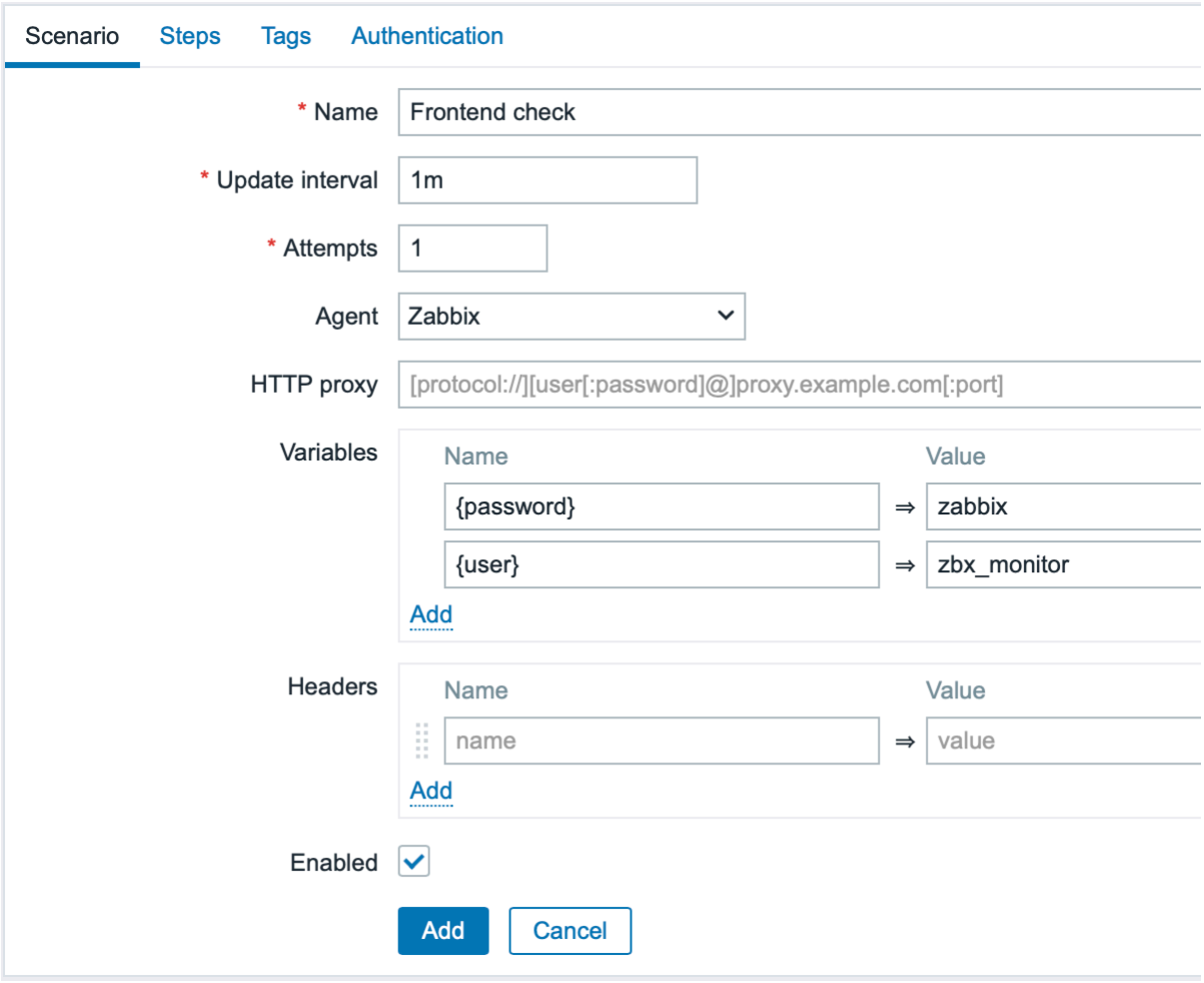
Esta seção apresenta um exemplo passo a passo da vida real de como o monitoramento web pode ser usado.

O monitoramento web do Zabbix será usado para monitorar o frontend do Zabbix. O objetivo é determinar se ele está disponível, fornece o conteúdo correto e quão rapidamente ele funciona. Para isso, são necessárias várias etapas, incluindo verificar a disponibilidade da primeira página, fazer login com um nome de usuário e senha, verificar o sucesso do login, fazer logout e confirmar o logout.

Cenário

Adicionar um novo cenário web

Vá para *Coleta de dados* → *Hosts*, escolha um host e clique em *Web* na linha desse host. Em seguida, clique em *Criar cenário web*.



No formulário do novo cenário, preencha os seguintes campos:

- **Nome** - Verificação do frontend
- **Intervalo de atualização** - 1m
- **Tentativas** - 1
- **Agent** - Zabbix

Na seção *Variáveis*, adicione duas variáveis: *{password}* e *{user}*. Insira as credenciais do seu usuário Zabbix existente como valores.

Por motivos de segurança, recomenda-se criar um usuário separado com permissões mínimas para usar para fins de monitoramento.

Opcionalmente, alterne para a guia *Tags* e adicione tags ao cenário web.

Note:
Uma vez totalmente configurado, este cenário web adicionará automaticamente um item Zabbix trapper ao host. Você pode usar as tags do cenário web para identificar rapidamente itens e triggers relacionados ou pesquisar nos dados coletados. Por exemplo, tags adequadas para este tutorial são `component: web-scenario` e/ou `target: frontend`.

Configurar etapas do cenário web

Alterne para a guia *Etapas* e defina as etapas para o cenário. Clique no botão *Adicionar* para adicionar uma etapa individual.

Campos comuns

Para cada etapa descrita abaixo, preencha os seguintes campos além dos campos específicos da etapa:

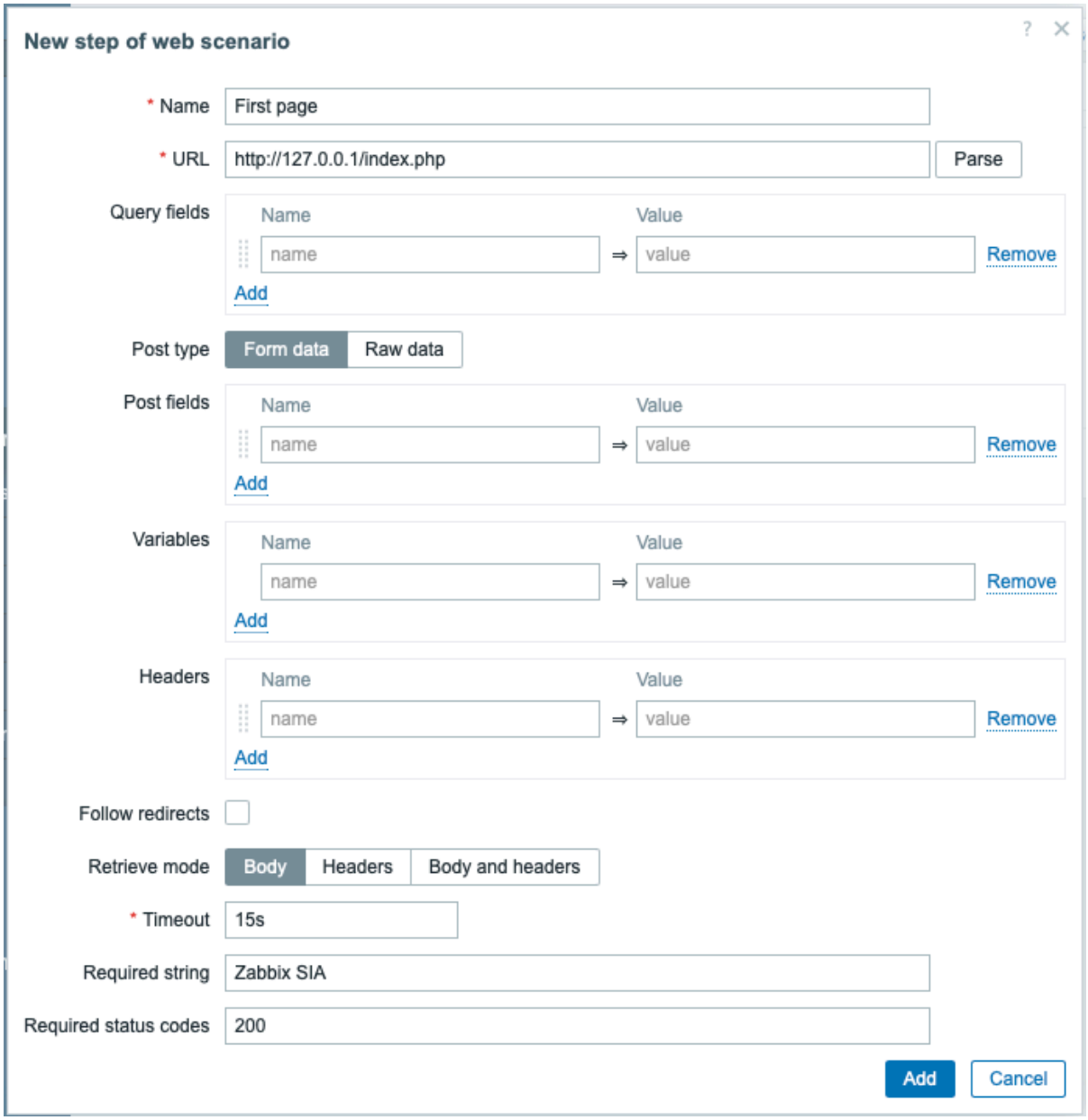
- **URL** - a URL do frontend do Zabbix
- **Timeout** - 15s
- **Required status codes** - 200

Etapa 1 do cenário web

Verifique se a primeira página responde corretamente, retorna o código de resposta HTTP 200 e contém o texto ”Zabbix SIA”.

- No campo **Nome**, insira *Primeira página*.
- No campo **String obrigatória**, insira *Zabbix SIA*.
- Preencha os campos comuns.

Quando terminar de configurar a etapa, pressione o botão *Adicionar*.



Etapa 2 do cenário web

Faça login no frontend do Zabbix usando as macros (variáveis) definidas no nível do cenário - *{user}* e *{password}*.

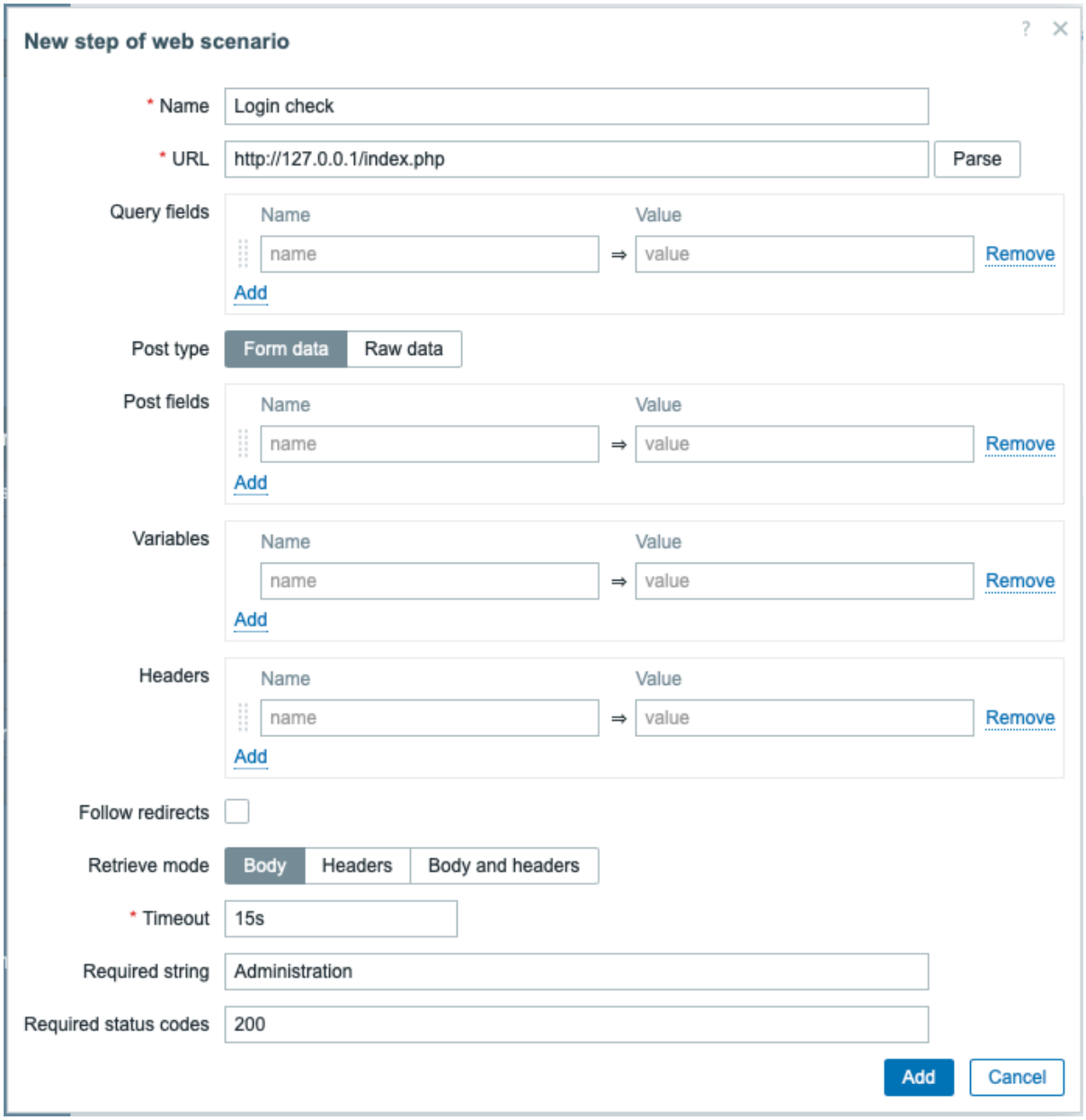
- No campo **Nome**, insira *Login*.
- Na seção **Campos de envio**, adicione três campos de envio:
  - *name* com valor *{user}*
  - *password* com valor *{password}*
  - *enter* com valor *Sign in*
- Na seção **Variáveis**, adicione uma nova variável *{csrf_token}* com valor *regex:([0-9a-z]{64})*. Esta variável irá capturar o valor do token CSRF atribuído para reutilizar na etapa 4.
- Preencha os campos comuns.

Attention:
Observe que o frontend do Zabbix usa redirecionamento JavaScript ao fazer login, portanto, o login deve ocorrer primeiro e os recursos de usuário autenticado só podem ser verificados nas etapas seguintes. Além disso, a etapa de login deve usar a URL completa para o arquivo **index.php**.

Etapa 3 do cenário web

Após o login, verifique o sucesso procurando por uma string visível apenas quando estiver logado - por exemplo, *Administração*.

- No campo **Nome**, insira *Verificação de login*.
- No campo **String obrigatória**, insira *Administração*.
- Preencha os campos comuns.

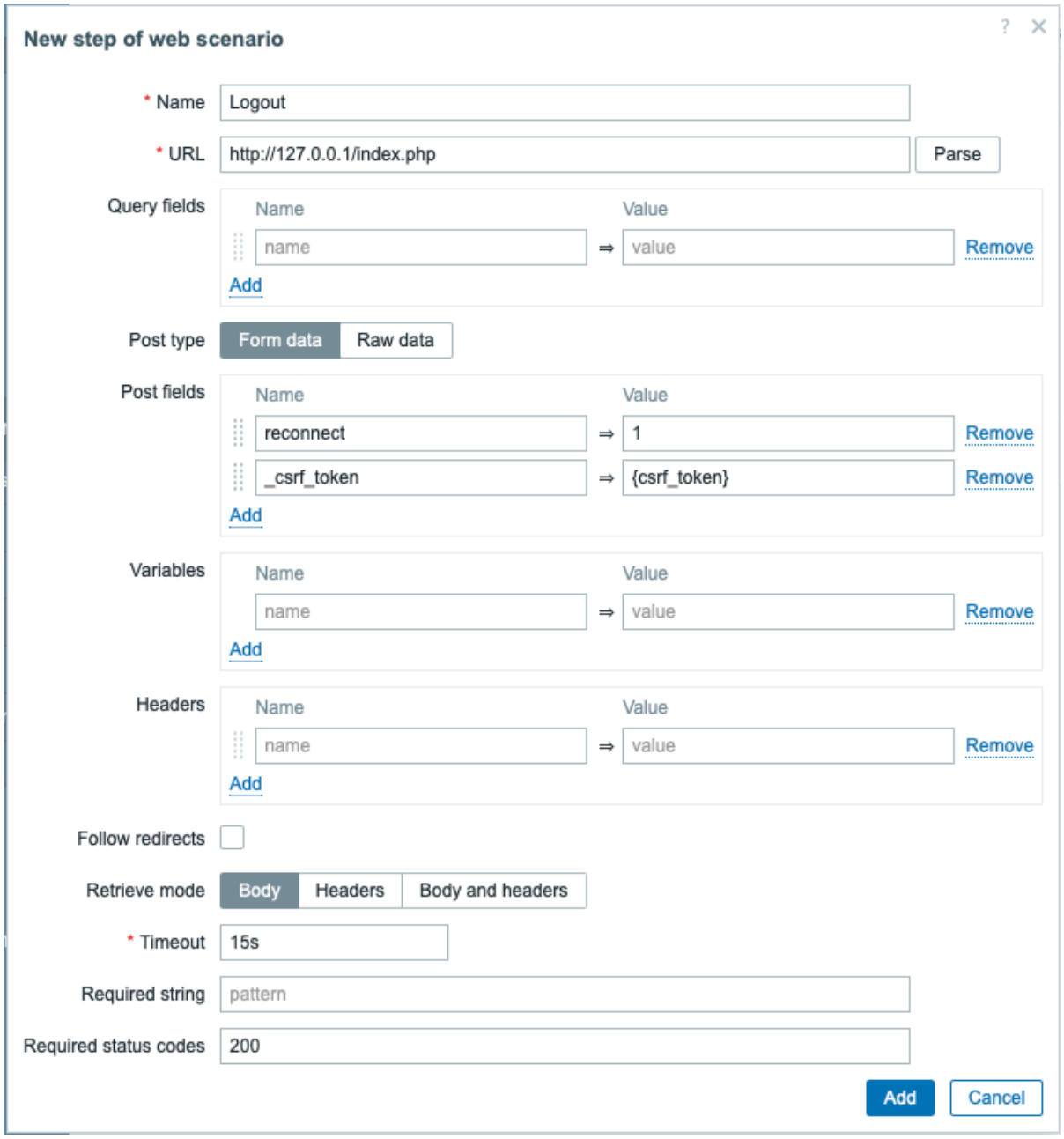


Etapa 4 do cenário web

Depois que a acessibilidade do frontend e o login forem verificados, adicione uma etapa de logout - caso contrário, o banco de dados do Zabbix ficará sobrecarregado com muitos registros de sessões abertas.

- No campo **Nome**, insira *Logout*.
- Na seção **Campos de postagem**, adicione dois campos de postagem:
  - *reconnect* com valor *1*
  - *_csrf_token* com valor *{csrf_token}*.
- Preencha os campos comuns.

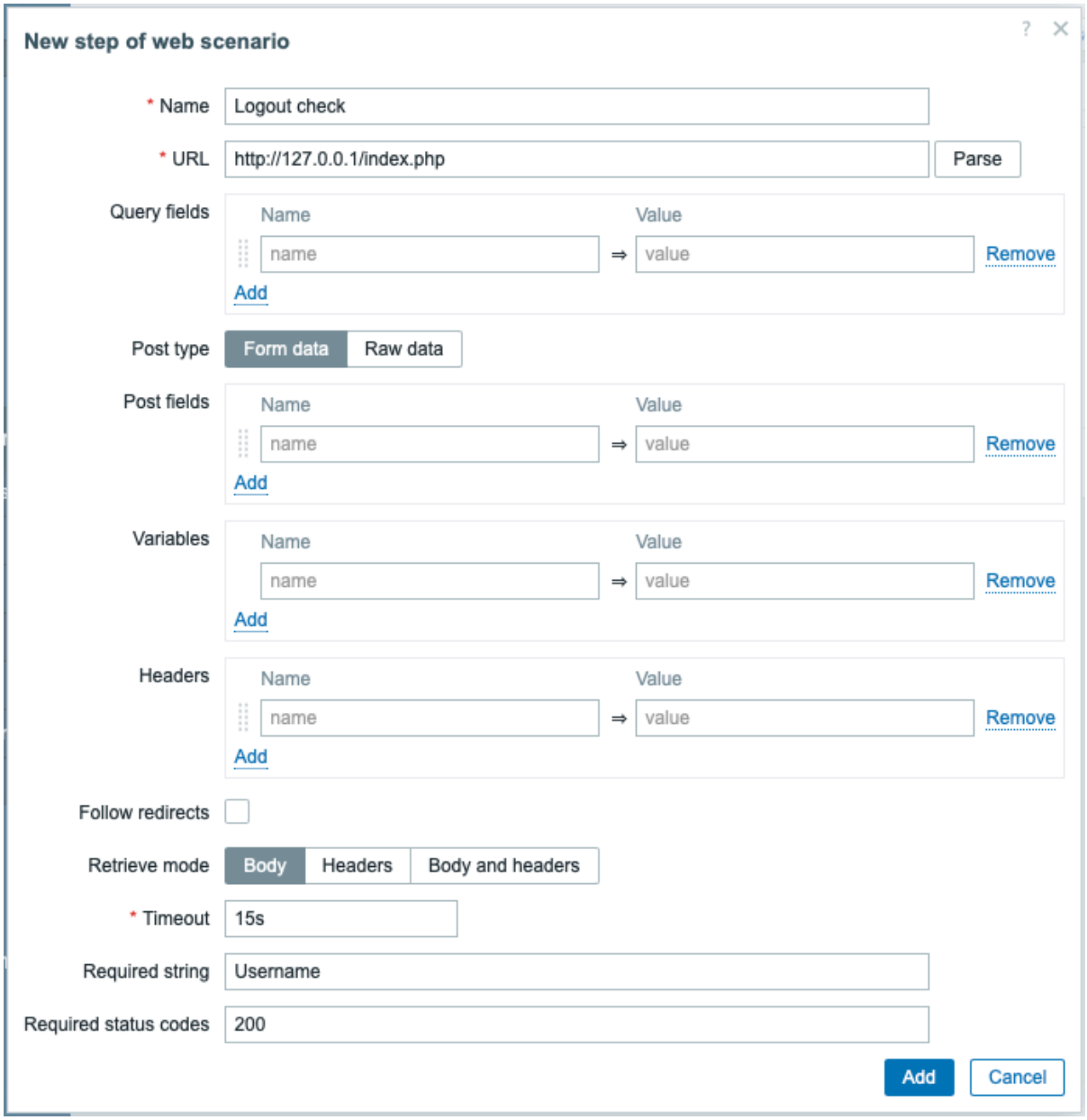
Esta etapa usa a variável {csrf_token} obtida na etapa 2



Etapa 5 do cenário web

Para confirmar o logout, verifique a string **Username**.

- No campo **Nome**, insira *Logout check*.
- No campo **String obrigatória**, insira *Username*.
- Preencha os campos comuns.



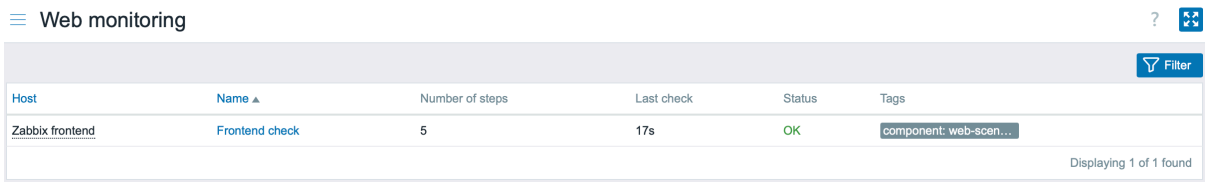
Configuração completa dos passos

Uma configuração completa dos passos do cenário web deve ser semelhante a esta:

Verifique os resultados

Salve o cenário de monitoramento web finalizado.

O cenário será adicionado ao host. Para visualizar as informações do cenário web, vá para *Monitoramento* → *Hosts*, localize o host na lista e clique no hiperlink Web na última coluna.



Clique no nome do cenário para ver estatísticas mais detalhadas:

10. Monitoramento de máquina virtual

Visão geral O Zabbix pode usar regras de descoberta de baixo nível para descobrir automaticamente hipervisores VMware e máquinas virtuais, e criar hosts para monitorá-los com base em prototipos de host predefinidos.

O Zabbix também inclui templates prontos para uso para monitorar o VMware vCenter ou hipervisores ESXi.

A versão mínima exigida do VMware vCenter ou vSphere é 5.1.

Coleta de dados O monitoramento de máquinas virtuais consiste em duas etapas:

1. Os processos *vmware collector* do Zabbix coletam dados da máquina virtual - os processos obtêm as informações necessárias dos serviços web do VMware pelo protocolo SOAP, pré-processam e armazenam na memória compartilhada do Zabbix server.
2. Os processos *poller* do Zabbix recuperam os dados usando as chaves de item de monitoramento VMware de verificação simples do Zabbix.

O Zabbix divide os dados coletados em dados de configuração do VMware e dados de contadores de desempenho do VMware. Ambos os tipos de dados são coletados independentemente pelos processos *vmware collector*.

As seguintes estatísticas estão disponíveis com base nas informações dos contadores de desempenho do VMware:

- Datastore
- Dispositivo de disco
- CPU
- Energia
- Interface de rede
- Itens de contador de desempenho personalizados

Para a lista completa de itens que obtêm dados dos contadores de desempenho do VMware, consulte *chaves de item de monitoramento VMware*.

Observe que a frequência de recuperação de eventos do VMware depende do intervalo de polling do vmware.eventlog, mas não pode ser inferior a 5 segundos.

Configuração Se o Zabbix server for compilado a partir dos códigos-fonte, ele deve ser compilado com as opções de configuração `--with-libcurl --with-libxml2` para habilitar o monitoramento de máquinas virtuais. Os pacotes do Zabbix já são compilados com essas opções habilitadas.

Os seguintes parâmetros do arquivo de configuração do Zabbix server podem ser modificados para o monitoramento de máquinas virtuais:

- `StartVMwareCollectors`

Note:
Recomenda-se habilitar mais coletores do que o número de serviços VMware monitorados; caso contrário, a obtenção de estatísticas de contadores de desempenho do VMware pode ser atrasada pela obtenção de dados de configuração do VMware (o que leva um tempo em instalações grandes).

 Geralmente, o valor de `StartVMwareCollectors` não deve ser inferior a 2 e não deve exceder o dobro da quantidade de serviços VMware monitorados: Quantidade de serviços < `StartVMwareCollectors` < (Quantidade de serviços * 2). Por exemplo, ao monitorar um serviço VMware, defina `StartVMwareCollectors` como 2; ao monitorar três serviços, defina `StartVMwareCollectors` como 5.

 Observe que o número necessário de coletores também depende do escopo do ambiente VMware, e dos parâmetros de configuração `VMwareFrequency` e `VMwarePerfFrequency`.

- `VMwareCacheSize`
- `VMwareFrequency`
- `VMwarePerfFrequency`
- `VMwareTimeout`

Attention:
Para suportar métricas de capacidade de datastore, certifique-se de que o valor da chave VMware `vpxd.stats.maxQueryMetrics` esteja definido para pelo menos 64. Para mais informações, consulte o artigo da *Base de Conhecimento VMware*.

Descoberta

O Zabbix pode usar regras de descoberta de baixo nível (por exemplo, *vmware.hv.discovery[{$VMWARE.URL}]*) para descobrir automaticamente hipervisores VMware e máquinas virtuais. Além disso, o Zabbix pode usar protótipos de host para gerar automaticamente hosts reais para as entidades descobertas. Para mais informações, consulte *Protótipos de host*.

Exemplos de configuração

Para um exemplo básico de como configurar o Zabbix para monitorar o VMware usando o template *VMware FQDN*, consulte *Monitorar VMware com Zabbix*.

Para um exemplo mais detalhado de como criar um host, uma regra de descoberta de baixo nível e um protótipo de host para monitorar o VMware, consulte *Exemplo de configuração*.

Log estendido Os dados coletados pelos processos *vmware collector* podem ser registrados para depuração detalhada usando o nível de depuração 5. O nível de depuração pode ser configurado nos arquivos de configuração do server e do proxy ou usando a opção de controle em tempo de execução `-R log_level_increase="vmware collector,N"`, onde "N" é o número do processo.

Por exemplo, para aumentar o nível de depuração de 4 para 5 para todos os processos *vmware collector*, execute o seguinte comando:

```
zabbix_server -R log_level_increase="vmware collector"
```

Para aumentar o nível de depuração de 4 para 5 para o segundo processo *vmware collector*, execute o seguinte comando:

```
zabbix_server -R log_level_increase="vmware collector,2"
```

Quando o log estendido dos dados do VMware collector não for mais necessário, recomenda-se diminuir o nível de depuração para o padrão (3) executando o comando `-R log_level_decrease`.

Solução de problemas

- Em caso de métricas indisponíveis, verifique se elas não estão indisponíveis ou desativadas por padrão nas versões recentes do VMware vSphere, ou se não há algum limite imposto nas consultas ao banco de dados de métricas de desempenho. Para mais informações, consulte ZBX-12094.
- Se `config.vpxd.stats.maxQueryMetrics` for inválido ou exceder o número máximo de caracteres permitidos, adicione um parâmetro `config.vpxd.stats.maxQueryMetrics` nas configurações do vCenter Server. O valor deste parâmetro deve ser o mesmo que o valor de `maxQuerysize` no arquivo *web.xml* do VMware. Para mais informações, consulte o artigo da *Base de Conhecimento VMware*.
- Se você suspeitar que sua instalação do Zabbix está usando muita memória, consulte Perfilando uso excessivo de memória com tcmalloc.

1 Chaves de item de monitoramento da VMware

Visão geral Esta página fornece detalhes sobre as verificações simples que podem ser usadas para monitorar ambientes VMware. As métricas são agrupadas pelo destino do monitoramento.

Chaves de item suportadas As chaves de item são listadas sem parâmetros e informações adicionais. Clique na chave do item para ver os detalhes completos.

Chave do item	Descrição	Grupo de item
vmware.eventlog	O log de eventos do VMware.	Serviço geral
vmware.fullname	O nome completo do serviço VMware.	
vmware.version	A versão do serviço VMware.	
vmware.cl.perfcounter	As métricas do contador de desempenho do cluster VMware.	Cluster
vmware.cluster.alarms.get	Os dados de alarmes do cluster VMware.	
vmware.cluster.discovery	A descoberta de clusters VMware.	
vmware.cluster.property	A propriedade do cluster VMware.	
vmware.cluster.status	O status do cluster VMware.	
vmware.cluster.tags.get	O array de tags do cluster VMware.	
vmware.datastore.alarms.get	Os dados de alarmes do datastore VMware.	Datastore
vmware.datastore.discovery	A descoberta de datastores VMware.	
vmware.datastore.hv.list	A lista de hipervisores do datastore.	
vmware.datastore.perfcounter	O valor do contador de desempenho do datastore VMware.	
vmware.datastore.property	A propriedade do datastore VMware.	
vmware.datastore.read	A quantidade de tempo para uma operação de leitura do datastore.	
vmware.datastore.size	O espaço do datastore VMware em bytes ou em porcentagem do total.	
vmware.datastore.tags.get	O array de tags do datastore VMware.	
vmware.datastore.write	A quantidade de tempo para uma operação de gravação no datastore.	
vmware.dc.alarms.get	Os dados de alarmes do datacenter VMware.	Datacenter
vmware.dc.discovery	A descoberta de datacenters VMware.	
vmware.dc.tags.get	O array de tags do datacenter VMware.	
vmware.dvswitch.discovery	A descoberta de VMware vSphere Distributed Switches.	vSphere Distributed Switch
vmware.dvswitch.fetchportsdata	Os dados das portas do VMware vSphere Distributed Switch.	
vmware.hv.alarms.get	Os dados de alarmes do hipervisor VMware.	Hipervisor
vmware.hv.cluster.name	O nome do cluster do hipervisor VMware.	
vmware.hv.connectionstate	O estado da conexão do hipervisor VMware.	
vmware.hv.cpu.usage	O uso do processador do hipervisor VMware (Hz).	
vmware.hv.cpu.usage.perf	O uso do processador do hipervisor VMware como porcentagem durante o intervalo.	

Chave do item	Descrição	Grupo de item
vmware.hv.cpu.utilization	O uso do processador do hipervisor VMware como porcentagem durante o intervalo, depende do gerenciamento de energia ou HT.	
vmware.hv.datacenter.name	O nome do datacenter do hipervisor VMware.	
vmware.hv.datastore.discovery	A descoberta de datastores do hipervisor VMware.	
vmware.hv.datastore.list	A lista de datastores do hipervisor VMware.	
vmware.hv.datastore.multipath	O número de caminhos de datastore disponíveis.	
vmware.hv.datastore.read	A quantidade média de tempo para uma operação de leitura do datastore.	
vmware.hv.datastore.size	O espaço do datastore VMware em bytes ou em porcentagem do total.	
vmware.hv.datastore.write	A quantidade média de tempo para uma operação de gravação no datastore.	
vmware.hv.discovery	A descoberta de hipervisores VMware.	
vmware.hv.diskinfo.get	Os dados de disco do hipervisor VMware.	
vmware.hv.fullname	O nome completo do produto, incluindo informações de versão.	
vmware.hv.hw.cpu.freq	A frequência do processador do hipervisor VMware.	
vmware.hv.hw.cpu.model	O modelo do processador do hipervisor VMware.	
vmware.hv.hw.cpu.num	O número de núcleos de processador no hipervisor VMware.	
vmware.hv.hw.cpu.threads	O número de threads de processador no hipervisor VMware.	
vmware.hv.hw.memory	O tamanho total da memória do hipervisor VMware.	
vmware.hv.hw.model	O modelo do hipervisor VMware.	
vmware.hv.hw.sensors.get	O valor dos sensores de hardware do hipervisor VMware.	
vmware.hv.hw.serialnumber	O número de série do hipervisor VMware.	
vmware.hv.hw.uuid	O UUID da BIOS do hipervisor VMware.	
vmware.hv.hw.vendor	O nome do fornecedor do hipervisor VMware.	
vmware.hv.maintenance	O status de manutenção do hipervisor VMware.	
vmware.hv.memory.size.ballooned	O tamanho da memória ballooned do hipervisor VMware.	
vmware.hv.memory.used	O tamanho da memória usada do hipervisor VMware.	
vmware.hv.net.if.discovery	A descoberta de interfaces de rede do hipervisor VMware.	
vmware.hv.network.in	As estatísticas de entrada de rede do hipervisor VMware.	
vmware.hv.network.linkspeed	A velocidade da interface de rede do hipervisor VMware.	
vmware.hv.network.out	As estatísticas de saída de rede do hipervisor VMware.	
vmware.hv.perfcounter	O valor do contador de desempenho do hipervisor VMware.	
vmware.hv.property	A propriedade do hipervisor VMware.	
vmware.hv.power	O uso de energia do hipervisor VMware.	
vmware.hv.sensor.health.state	O sensor de resumo do estado de integridade do hipervisor VMware.	
vmware.hv.sensors.get	Os sensores de estado do fornecedor de hardware do hipervisor VMware.	
vmware.hv.status	O status do hipervisor VMware.	
vmware.hv.tags.get	O array de tags do hipervisor VMware.	
vmware.hv.uptime	O tempo de atividade do hipervisor VMware.	
vmware.hv.version	A versão do hipervisor VMware.	
vmware.hv.vm.num	O número de máquinas virtuais no hipervisor VMware.	
vmware.rp.cpu.usage	O uso da CPU em hertz durante o intervalo no VMware Resource Pool.	Resource pool
vmware.rp.memory	As métricas de memória do resource pool VMware.	
vmware.alarms.get	Os dados de alarmes do virtual center VMware.	Virtual center
vmware.vm.alarms.get	Os dados de alarmes da máquina virtual VMware.	Máquina virtual
vmware.vm.attribute	O valor do atributo personalizado da máquina virtual VMware.	
vmware.vm.cluster.name	O nome da máquina virtual VMware.	
vmware.vm.consolidationneeded	O disco da máquina virtual VMware requer consolidação.	
vmware.vm.cpu.latency	A porcentagem de tempo em que a máquina virtual não pode ser executada porque está competindo pelo acesso à(s) CPU(s) física(s).	
vmware.vm.cpu.num	O número de processadores na máquina virtual VMware.	
vmware.vm.cpu.readiness	A porcentagem de tempo em que a máquina virtual estava pronta, mas não pôde ser agendada para rodar na CPU física.	
vmware.vm.cpu.ready	O tempo em que a máquina virtual estava pronta, mas não pôde ser agendada para rodar na CPU física.	
vmware.vm.cpu.swapwait	A porcentagem do tempo da CPU gasto esperando pelo swap-in.	
vmware.vm.cpu.usage	O uso do processador da máquina virtual VMware (Hz).	
vmware.vm.cpu.usage.perf	O uso do processador da máquina virtual VMware como porcentagem durante o intervalo.	
vmware.vm.datacenter.name	O nome do datacenter da máquina virtual VMware.	

Chave do item	Descrição	Grupo de item
vmware.vm.discovery	A descoberta de máquinas virtuais VMware.	
vmware.vm.guest.memory.swapped	A quantidade de memória física do guest que foi trocada para o espaço de swap.	
vmware.vm.guest.osuptime	O tempo total decorrido desde a última inicialização do sistema operacional.	
vmware.vm.hv.name	O nome do hipervisor da máquina virtual VMware.	
vmware.vm.memory.size	O tamanho total da memória da máquina virtual VMware.	
vmware.vm.memory.size.ballooned	O tamanho da memória ballooned da máquina virtual VMware.	
vmware.vm.memory.size.compressed	O tamanho da memória comprimida da máquina virtual VMware.	
vmware.vm.memory.size.consumed	A quantidade de memória física do host consumida para backup das páginas de memória física do guest.	
vmware.vm.memory.size.private	O tamanho da memória privada da máquina virtual VMware.	
vmware.vm.memory.size.shared	O tamanho da memória compartilhada da máquina virtual VMware.	
vmware.vm.memory.size.swapped	O tamanho da memória trocada da máquina virtual VMware.	
vmware.vm.memory.size.usage.guest	O uso de memória do guest da máquina virtual VMware.	
vmware.vm.memory.size.usage.host	O uso de memória do host da máquina virtual VMware.	
vmware.vm.memory.usage	A porcentagem de memória física do host que foi consumida.	
vmware.vm.net.if.discovery	A descoberta de interfaces de rede da máquina virtual VMware.	
vmware.vm.net.if.in	As estatísticas de entrada da interface de rede da máquina virtual VMware.	
vmware.vm.net.if.out	As estatísticas de saída da interface de rede da máquina virtual VMware.	
vmware.vm.net.if.usage	A utilização da rede da máquina virtual VMware durante o intervalo.	
vmware.vm.perfcounter	O valor do contador de desempenho da máquina virtual VMware.	
vmware.vm.powerstate	O estado de energia da máquina virtual VMware.	
vmware.vm.property	A propriedade da máquina virtual VMware.	
vmware.vm.snapshot.get	O estado do snapshot da máquina virtual VMware.	
vmware.vm.state	O estado da máquina virtual VMware.	
vmware.vm.storage.committed	O espaço de armazenamento comprometido da máquina virtual VMware.	
vmware.vm.storage.readoio	O número médio de solicitações de leitura pendentes para o disco virtual durante o intervalo de coleta.	
vmware.vm.storage.totalreadlatency	O tempo médio que uma leitura do disco virtual leva.	
vmware.vm.storage.totalwritelatency	O tempo médio que uma gravação no disco virtual leva.	
vmware.vm.storage.uncommitted	O espaço de armazenamento não comprometido da máquina virtual VMware.	
vmware.vm.storage.unshared	O espaço de armazenamento não compartilhado da máquina virtual VMware.	
vmware.vm.storage.writeoio	O número médio de solicitações de gravação pendentes para o disco virtual durante o intervalo de coleta.	
vmware.vm.tags.get	O array de tags da máquina virtual VMware.	
vmware.vm.tools	O estado ou versão das ferramentas do guest da máquina virtual VMware.	
vmware.vm.uptime	O tempo de atividade da máquina virtual VMware.	
vmware.vm.vfs.dev.discovery	A descoberta de dispositivos de disco da máquina virtual VMware.	
vmware.vm.vfs.dev.read	As estatísticas de leitura do dispositivo de disco da máquina virtual VMware.	
vmware.vm.vfs.dev.write	As estatísticas de gravação do dispositivo de disco da máquina virtual VMware.	
vmware.vm.vfs.fs.discovery	A descoberta de sistemas de arquivos da máquina virtual VMware.	
vmware.vm.vfs.fs.size	As estatísticas do sistema de arquivos da máquina virtual VMware.	

Detalhes da chave do item Parâmetros sem colchetes angulares são obrigatórios. Parâmetros marcados com colchetes angulares **< >** são opcionais.

vmware.eventlog[url,<mode>,<severity>]

 O log de eventos do VMware.
 Valor de retorno: *Log*.

Parâmetros:

- **url** - a URL do serviço VMware;
- **mode** - *all* (padrão) ou *skip* - ignora o processamento de dados antigos;
- **severity** - filtra por severidade: *error*, *warning*, *info* ou *user*. Este parâmetro deve ser colocado entre aspas se mais de uma severidade for especificada em uma lista separada por vírgulas (por exemplo, `"error,warning,info,user"`). Desabilitado por padrão.

Comentários:

- Deve haver apenas uma chave de item `vmware.eventlog` por URL;
- Veja também exemplo de filtragem de registros de log de eventos do VMware;
- Este item retorna informações do usuário desde o Zabbix 7.0.1.

vmware.fullnameurl

 O nome completo do serviço VMware.
 Valor de retorno: *String*.

Parâmetros:

- **url** - a URL do serviço VMware.

vmware.versionurl

 A versão do serviço VMware.
 Valor de retorno: *String*.

Parâmetros:

- **url** - a URL do serviço VMware.

vmware.cl.perfcounter[url,id,path,<instance>]

 As métricas do contador de desempenho do cluster VMware.
 Valor de retorno: *Integer*.

Parâmetros:

- **url** - a URL do serviço VMware;
- **id** - o ID do cluster VMware. `id` pode ser recebido de `vmware.cluster.discovery[]` como `{#CLUSTER.ID}`.
- **path** - o caminho do contador de desempenho[1];
- **instance** - a instância do contador de desempenho.

vmware.cluster.alarms.get[url,id]

 Os dados de alarmes do cluster VMware.
 Valor de retorno: objeto JSON.

Parâmetros:

- **url** - o URL do serviço VMware;
- **id** - o ID do cluster VMware.

vmware.cluster.discoveryurl

 A descoberta de clusters VMware.
 Valor de retorno: objeto JSON.

Parâmetros:

- **url** - a URL do serviço VMware.

vmware.cluster.property[url,id,prop]

 A propriedade do cluster VMware.
 Valor de retorno: *String*.

Parâmetros:

- **url** - a URL do serviço VMware;
- **id** - o ID do cluster VMware;
- **prop** - o caminho da propriedade.

vmware.cluster.status[url,name]

 O status do cluster VMware.
 Valor de retorno: *0* - cinza; *1* - verde; *2* - amarelo; *3* - vermelho.

Parâmetros:

- **url** - a URL do serviço VMware;
- **name** - o nome do cluster VMware.

vmware.cluster.tags.get[url,id]

 O array de tags do cluster VMware.
 Valor de retorno: objeto JSON.

Parâmetros:

- **url** - a URL do serviço VMware;
- **id** - o ID do cluster VMware.

Este item funciona com o vSphere 6.5 e mais recente.

vmware.datastore.alarms.get[url,uuid]

 Os dados de alarmes do datastore VMware.
 Valor de retorno: objeto JSON.

Parâmetros:

- **url** - a URL do serviço VMware;

- **uuid** - o identificador global exclusivo do datastore VMware.

vmware.datastore.discovery[url,<filter_uuid>]

 A descoberta de datastores VMware.
 Valor de retorno: objeto JSON.

Parâmetros:

- **url** - a URL do serviço VMware;
- **filter_uuid** - um único identificador exclusivo de HV ou VM (padrão: vazio). Se especificado, apenas os datastores vinculados serão descobertos.

vmware.datastore.hv.list[url,datastore]

 A lista de hypervisors do datastore.
 Valor de retorno: *String*.

Parâmetros:

- **url** - a URL do serviço VMware;
- **datastore** - o UUID ou nome do datastore.

Exemplo de saída:

```
esx7-01-host.zabbix.sandbox
esx7-02-host.zabbix.sandbox
```

vmware.datastore.perfcounter[url,uuid,path,<instance>]

 O valor do contador de desempenho do datastore VMware.
 Valor de retorno: *Integer* [2].

Parâmetros:

- **url** - a URL do serviço VMware;
- **uuid** - o identificador global exclusivo do datastore VMware;
- **path** - o caminho do contador de desempenho[1];
- **instance** - a instância do contador de desempenho. Use uma instância vazia para valores agregados (padrão). `instance` pode ser obtido de `vmware.datastore.discovery[]` como parte do array {#DATASTORE.EXTENT}.

vmware.datastore.property[url,uuid,prop]

 A propriedade do datastore VMware.
 Valor de retorno: *String*.

Parâmetros:

- **url** - a URL do serviço VMware;
- **uuid** - o identificador global exclusivo do datastore VMware;
- **prop** - o caminho da propriedade.

vmware.datastore.read[url,datastore,<mode>]

 A quantidade de tempo para uma operação de leitura do datastore (milissegundos).
 Valor de retorno: *Integer* [2].

Parâmetros:

- **url** - a URL do serviço VMware;
- **datastore** - o UUID ou nome do datastore;
- **mode** - *latency* (valor médio, padrão) ou *maxlatency* (valor máximo).

vmware.datastore.size[url,datastore,<mode>]

 O espaço do datastore VMware em bytes ou em porcentagem do total.
 Valor de retorno: *Integer* - para bytes; *Float* - para porcentagem.

Parâmetros:

- **url** - a URL do serviço VMware;
- **datastore** - o UUID ou nome do datastore;
- **mode** - valores possíveis: *total* (padrão), *free*, *pfree* (porcentagem livre), *uncommitted*.

vmware.datastore.tags.get[url,uuid]

 O array de tags do datastore do VMware.
 Valor de retorno: objeto JSON.

Parâmetros:

- **url** - a URL do serviço VMware;
- **uuid** - o identificador global exclusivo do datastore VMware.

Este item funciona com o vSphere 6.5 e mais recentes.

vmware.datastore.write[url,datastore,<mode>]

 A quantidade de tempo para uma operação de gravação no datastore (milissegundos).
 Valor de retorno: *Integer* [2].

Parâmetros:

- **url** - a URL do serviço VMware;
- **datastore** - o UUID ou nome do datastore;
- **mode** - *latency* (valor médio, padrão) ou *maxlatency* (valor máximo).

vmware.dc.alarms.get[url,id]

 Os dados de alarmes do datacenter VMware.
 Valor de retorno: objeto JSON.

Parâmetros:

- **url** - a URL do serviço VMware;
- **id** - o ID do datacenter VMware.

vmware.dc.discoveryurl

 A descoberta de datacenters VMware.
 Valor de retorno: objeto JSON.

Parâmetros:

- **url** - a URL do serviço VMware.

vmware.dc.tags.get[url,id]

 O array de tags do datacenter VMware.
 Valor de retorno: objeto JSON.

Parâmetros:

- **url** - a URL do serviço VMware;
- **id** - o ID do datacenter VMware.

Este item funciona com o vSphere 6.5 e mais recente.

vmware.dvswitch.discoveryurl

 A descoberta de VMware vSphere Distributed Switches.
 Valor de retorno: objeto JSON.

Parâmetros:

- **url** - a URL do serviço VMware.

vmware.dvswitch.fetchports.get[url,uuid,<filter>,<mode>]

 Os dados das portas do VMware vSphere Distributed Switch.
 Valor de retorno: objeto JSON.

Parâmetros:

- **url** - a URL do serviço VMware;
- **uuid** - o identificador global exclusivo do VMware DVSwitch.
- **filter** - uma única string com critérios separados por vírgula para selecionar portas;
- **mode** - *state* (todo o XML sem nós XML ”config”, padrão) ou *full*.

O parâmetro **filter** suporta os critérios disponíveis no objeto de dados VMware DistributedVirtualSwitchPortCriteria.

Exemplo:

```
vmware.dvswitch.fetchports.get[{$VMWARE.URL},{$VMWARE.DVS.UUID},"connected:true,active:true,uplinkPort:fal
```

vmware.hv.alarms.get[url,uuid]

 Os dados dos alarmes do hipervisor VMware.
 Valor de retorno: objeto JSON.

Parâmetros:

- **url** - o URL do serviço VMware;
- **uuid** - o identificador global exclusivo do hipervisor VMware.

vmware.hv.cluster.name[url,uuid]

 O nome do cluster do hipervisor VMware.
 Valor de retorno: *String*.

Parâmetros:

- **url** - a URL do serviço VMware;

- **uuid** - o identificador exclusivo global do hipervisor VMware.

vmware.hv.connectionstate[url,uuid]

 O estado de conexão do hipervisor VMware.
 Valor de retorno: *String*: *connected*, *disconnected* ou *notResponding*.

Parâmetros:

- **url** - a URL do serviço VMware;
- **uuid** - o identificador global exclusivo do hipervisor VMware.

vmware.hv.cpu.usage[url,uuid]

 O uso do processador do hypervisor VMware (Hz).
 Valor de retorno: *Inteiro*.

Parâmetros:

- **url** - a URL do serviço VMware;
- **uuid** - o identificador global exclusivo do hypervisor VMware.

vmware.hv.cpu.usage.perf[url,uuid]

 O uso do processador do hypervisor VMware como uma porcentagem durante o intervalo.
 Valor de retorno: *Float*.

Parâmetros:

- **url** - a URL do serviço VMware;
- **uuid** - o identificador global exclusivo do hypervisor VMware.

vmware.hv.cpu.utilization[url,uuid]

 O uso do processador do hypervisor VMware como uma porcentagem durante o intervalo, depende do gerenciamento de energia ou HT.
 Valor de retorno: *Float*.

Parâmetros:

- **url** - a URL do serviço VMware;
- **uuid** - o identificador exclusivo global do hypervisor VMware.

vmware.hv.datacenter.name[url,uuid]

 O nome do datacenter do hypervisor VMware.
 Valor de retorno: *String*.

Parâmetros:

- **url** - a URL do serviço VMware;
- **uuid** - o identificador exclusivo global do hypervisor VMware.

vmware.hv.datastore.discovery[url,uuid]

 A descoberta de datastores do hipervisor VMware.
 Valor de retorno: objeto JSON.

Parâmetros:

- **url** - a URL do serviço VMware;
- **uuid** - o identificador exclusivo global do hipervisor VMware.

vmware.hv.datastore.list[url,uuid]

 A lista de datastores do hypervisor VMware.
 Valor de retorno: *String*.

Parâmetros:

- **url** - o URL do serviço VMware;
- **uuid** - o identificador global exclusivo do hypervisor VMware.

Exemplo de saída:

```
SSD-RAID1-VAULT1
SSD-RAID1-VAULT2
SSD-RAID10
```

vmware.hv.datastore.multipath[url,uuid,<datastore>,<partitionid>]

 O número de caminhos disponíveis para o datastore.
 Valor de retorno: *Integer*.

Parâmetros:

- **url** - a URL do serviço VMware;
- **uuid** - o identificador global exclusivo do hypervisor VMware;

- **datastore** - o UUID ou nome do datastore;
- **partitionid** - o ID interno do dispositivo físico de `vmware.hv.datastore.discovery`.

vmware.hv.datastore.read[url,uuid,datastore,<mode>]

 A quantidade média de tempo para uma operação de leitura do datastore (milissegundos).
 Valor de retorno: *Integer* [2].

Parâmetros:

- **url** - a URL do serviço VMware;
- **uuid** - o identificador global exclusivo do hypervisor VMware;
- **datastore** - o UUID ou nome do datastore;
- **mode** - *latency* (padrão).

vmware.hv.datastore.size[url,uuid,datastore,<mode>]

 O espaço do datastore VMware em bytes ou em porcentagem do total.
 Valor de retorno: *Integer* - para bytes; *Float* - para porcentagem.

Parâmetros:

- **url** - a URL do serviço VMware;
- **uuid** - o identificador global exclusivo do hipervisor VMware;
- **datastore** - o UUID ou nome do datastore;
- **mode** - valores possíveis: *total* (padrão), *free*, *pfree* (porcentagem livre), *uncommitted*.

vmware.hv.datastore.write[url,uuid,datastore,<mode>]

 O tempo médio de uma operação de gravação no datastore (milissegundos).
 Valor de retorno: *Integer* [2].

Parâmetros:

- **url** - a URL do serviço VMware;
- **uuid** - o identificador global exclusivo do hypervisor VMware;
- **datastore** - o UUID ou nome do datastore;
- **mode** - *latency* (padrão).

vmware.hv.discoveryurl

 A descoberta de hipervisores VMware.
 Valor de retorno: objeto JSON.

Parâmetros:

- **url** - a URL do serviço VMware.

vmware.hv.diskinfo.get[url,uuid]

 Os dados do disco do hypervisor VMware.
 Valor de retorno: objeto JSON.

Parâmetros:

- **url** - o URL do serviço VMware;
- **uuid** - o identificador global exclusivo do hypervisor VMware.

vmware.hv.fullname[url,uuid]

 O nome completo do produto, incluindo informações de versão.
 Valor de retorno: *String*.

Parâmetros:

- **url** - a URL do serviço VMware;
- **uuid** - o identificador global exclusivo do hypervisor VMware.

vmware.hv.hw.cpu.freq[url,uuid]

 A frequência do processador do hipervisor VMware (Hz).
 Valor de retorno: *Inteiro*.

Parâmetros:

- **url** - a URL do serviço VMware;
- **uuid** - o identificador exclusivo global do hipervisor VMware.

vmware.hv.hw.cpu.model[url,uuid]

 O modelo do processador do hypervisor VMware.
 Valor de retorno: *String*.

Parâmetros:

- **url** - a URL do serviço VMware;

- **uuid** - o identificador global exclusivo do hypervisor VMware.

vmware.hv.hw.cpu.num[url,uuid]

 O número de núcleos de processador no hypervisor VMware.
 Valor de retorno: *Integer*.

Parâmetros:

- **url** - a URL do serviço VMware;
- **uuid** - o identificador global exclusivo do hypervisor VMware.

vmware.hv.hw.cpu.threads[url,uuid]

 O número de threads do processador no hypervisor VMware.
 Valor de retorno: *Integer*.

Parâmetros:

- **url** - o URL do serviço VMware;
- **uuid** - o identificador global exclusivo do hypervisor VMware.

vmware.hv.hw.memory[url,uuid]

 O tamanho total da memória do hypervisor VMware (bytes).
 Valor de retorno: *Integer*.

Parâmetros:

- **url** - a URL do serviço VMware;
- **uuid** - o identificador global exclusivo do hypervisor VMware.

vmware.hv.hw.model[url,uuid]

 O modelo do hypervisor VMware.
 Valor de retorno: *String*.

Parâmetros:

- **url** - a URL do serviço VMware;
- **uuid** - o identificador global exclusivo do hypervisor VMware.

vmware.hv.hw.sensors.get[url,uuid]

 O valor dos sensores de hardware do hypervisor VMware.
 Valor de retorno: objeto JSON.

Parâmetros:

- **url** - a URL do serviço VMware;
- **uuid** - o identificador global exclusivo do hypervisor VMware.

vmware.hv.hw.serialnumber[url,uuid]

 O número de série do hypervisor VMware.
 Valor de retorno: *String*.

Parâmetros:

- **url** - a URL do serviço VMware;
- **uuid** - o identificador global exclusivo do hypervisor VMware.

Este item funciona com a API vSphere 6.7 e mais recente.

vmware.hv.hw.uuid[url,uuid]

 O UUID da BIOS do hypervisor VMware.
 Valor de retorno: *String*.

Parâmetros:

- **url** - a URL do serviço VMware;
- **uuid** - o identificador global exclusivo do hypervisor VMware.

Este item funciona com a API vSphere 6.7 e mais recente.

vmware.hv.hw.vendor[url,uuid]

 O nome do fornecedor do hipervisor VMware.
 Valor de retorno: *String*.

Parâmetros:

- **url** - a URL do serviço VMware;
- **uuid** - o identificador global exclusivo do hipervisor VMware.

Este item funciona com a API vSphere 6.7 e mais recentes.

vmware.hv.maintenance[url,uuid]

 O status de manutenção do hypervisor VMware.
 Valor de retorno: *0* - não está em manutenção; *1* - está em manutenção.

Parâmetros:

- **url** - a URL do serviço VMware;
- **uuid** - o identificador global exclusivo do hypervisor VMware.

vmware.hv.memory.size.ballooned[url,uuid]

 Tamanho da memória inflada do hypervisor VMware (bytes).
 Valor de retorno: *Integer*.

Parâmetros:

- **url** - a URL do serviço VMware;
- **uuid** - o identificador global exclusivo do hypervisor VMware.

vmware.hv.memory.used[url,uuid]

 O tamanho da memória usada pelo hypervisor VMware (bytes).
 Valor de retorno: *Integer*.

Parâmetros:

- **url** - o URL do serviço VMware;
- **uuid** - o identificador global exclusivo do hypervisor VMware.

vmware.hv.net.if.discovery[url,uuid]

 A descoberta das interfaces de rede do hipervisor VMware.
 Valor de retorno: *objeto JSON*.

Parâmetros:

- **url** - o URL do serviço VMware;
- **uuid** - o identificador global exclusivo do hipervisor VMware.

vmware.hv.network.in[url,uuid,<mode>]

 As estatísticas de entrada da rede do hypervisor VMware (bytes por segundo).
 Valor de retorno: *Integer* [2].

Parâmetros:

- **url** - a URL do serviço VMware;
- **uuid** - o identificador global exclusivo do hypervisor VMware;
- **mode** - *bps* (padrão), *packets*, *dropped*, *errors*, *broadcast*.

vmware.hv.network.linkspeed[url,uuid,ifname]

 A velocidade da interface de rede do hypervisor VMware.
 Valor de retorno: *Inteiro*. Retorna *0* se a interface de rede estiver inativa, caso contrário, retorna o valor da velocidade da interface.

Parâmetros:

- **url** - o URL do serviço VMware;
- **uuid** - o identificador global exclusivo do hypervisor VMware;
- **ifname** - o nome da interface.

vmware.hv.network.out[url,uuid,<mode>]

 As estatísticas de saída de rede do hypervisor VMware (bytes por segundo).
 Valor de retorno: *Integer* [2].

Parâmetros:

- **url** - a URL do serviço VMware;
- **uuid** - o identificador global exclusivo do hypervisor VMware;
- **mode** - *bps* (padrão), *packets*, *dropped*, *errors*, *broadcast*.

vmware.hv.perfcounter[url,uuid,path,<instance>]

 O valor do contador de desempenho do hypervisor VMware.
 Valor de retorno: *Integer* [2].

Parâmetros:

- **url** - a URL do serviço VMware;
- **uuid** - o identificador global exclusivo do hypervisor VMware;
- **path** - o caminho do contador de desempenho [1];
- **instance** - a instância do contador de desempenho. Use uma instância vazia para valores agregados (padrão).

vmware.hv.property[url,uuid,prop]

 A propriedade do hipervisor VMware.
 Valor de retorno: *String*.

Parâmetros:

- **url** - o URL do serviço VMware;
- **uuid** - o identificador global exclusivo do hipervisor VMware;
- **prop** - o caminho da propriedade.

vmware.hv.power[url,uuid,<max>]

 O uso de energia do hypervisor VMware (W).
 Valor de retorno: *Inteiro*.

Parâmetros:

- **url** - a URL do serviço VMware;
- **uuid** - o identificador exclusivo global do hypervisor VMware;
- **max** - o uso máximo de energia permitido.

vmware.hv.sensor.health.state[url,uuid]

 O sensor de estado de integridade do hypervisor VMware.
 Valor de retorno: *Integer*: *0* - cinza; *1* - verde; *2* - amarelo; *3* - vermelho.

Parâmetros:

- **url** - a URL do serviço VMware;
- **uuid** - o identificador global exclusivo do hypervisor VMware.

Observe que o item pode não funcionar no VMware vSphere 6.5 e mais recente, porque a VMware descontinuou o sensor *VMware Rollup Health State*.

vmware.hv.sensors.get[url,uuid]

 Os sensores de estado do fornecedor de HW do hypervisor VMware.
 Valor de retorno: objeto JSON.

Parâmetros:

- **url** - a URL do serviço VMware;
- **uuid** - o identificador global exclusivo do hypervisor VMware.

vmware.hv.status[url,uuid]

 O status do hypervisor VMware.
 Valor de retorno: *Integer*: *0* - cinza; *1* - verde; *2* - amarelo; *3* - vermelho.

Parâmetros:

- **url** - o URL do serviço VMware;
- **uuid** - o identificador global exclusivo do hypervisor VMware.

Este item usa a propriedade de status geral do sistema host.

vmware.hv.tags.get[url,uuid]

 O array de tags do hypervisor VMware.
 Valor de retorno: objeto JSON.

Parâmetros:

- **url** - a URL do serviço VMware;
- **uuid** - o identificador global exclusivo do hypervisor VMware.

Este item funciona com o vSphere 6.5 e mais recente.

vmware.hv.uptime[url,uuid]

 O tempo de atividade do hypervisor VMware (segundos).
 Valor de retorno: *Inteiro*.

Parâmetros:

- **url** - a URL do serviço VMware;
- **uuid** - o identificador global exclusivo do hypervisor VMware.

Este item usa a propriedade de status geral do sistema host.

vmware.hv.version[url,uuid]

 A versão do hypervisor VMware.
 Valor de retorno: *String*.

Parâmetros:

- **url** - a URL do serviço VMware;
- **uuid** - o identificador global exclusivo do hypervisor VMware.

vmware.hv.vm.num[url,uuid]

 O número de máquinas virtuais no hypervisor VMware.
 Valor de retorno: *Inteiro*.

Parâmetros:

- **url** - o URL do serviço VMware;
- **uuid** - o identificador global exclusivo do hypervisor VMware.

vmware.rp.cpu.usage[url,rpid]

 O uso da CPU em hertz durante o intervalo no VMware Resource Pool.
 Valor de retorno: *Integer*.

Parâmetros:

- **url** - a URL do serviço VMware;
- **rpid** - o ID do resource pool do VMware.

vmware.rp.memory[url,rpid,<mode>]

 As métricas de memória do pool de recursos VMware.
 Valor de retorno: *Integer*.

Parâmetros:

- **url** - a URL do serviço VMware;
- **rpid** - o ID do pool de recursos VMware;
- **mode** - valores possíveis:
consumed (padrão) - a quantidade de memória física do host consumida para backup das páginas de memória física do guest
 ballooned - a quantidade de memória física do guest recuperada da máquina virtual pelo driver balloon no guest
overhead - a memória física do host consumida pelas estruturas de dados do ESXi para executar as máquinas virtuais

vmware.alarms.geturl

 Os dados de alarmes do VMware virtual center.
 Valor de retorno: objeto JSON.

Parâmetros:

- **url** - a URL do serviço VMware.

vmware.vm.alarms.get[url,uuid]

 Os dados de alarmes da máquina virtual VMware.
 Valor de retorno: objeto JSON.

Parâmetros:

- **url** - o URL do serviço VMware;
- **uuid** - o identificador global exclusivo da máquina virtual VMware.

vmware.vm.attribute[url,uuid,name]

 O valor do atributo personalizado da máquina virtual VMware.
 Valor de retorno: *String*.

Parâmetros:

- **url** - a URL do serviço VMware;
- **uuid** - o identificador exclusivo global da máquina virtual VMware;
- **name** - o nome do atributo personalizado.

vmware.vm.cluster.name[url,uuid]

 O nome da máquina virtual VMware.
 Valor de retorno: *String*.

Parâmetros:

- **url** - a URL do serviço VMware;
- **uuid** - o identificador global exclusivo da máquina virtual VMware.

vmware.vm.consolidationneeded[url,uuid]

 O disco da máquina virtual VMware requer consolidação.
 Valor de retorno: *String*: *true* - a consolidação é necessária; *false* - a consolidação não é necessária.

Parâmetros:

- **url** - o URL do serviço VMware;
- **uuid** - o identificador global exclusivo da máquina virtual VMware.

vmware.vm.cpu.latency[url,uuid]

 A porcentagem de tempo que a máquina virtual não pode ser executada porque está competindo pelo acesso à(s) CPU(s) física(s).
 Valor de retorno: *Float*.

Parâmetros:

- **url** - o URL do serviço VMware;
- **uuid** - o identificador global exclusivo da máquina virtual VMware.

vmware.vm.cpu.num[url,uuid]

 O número de processadores na máquina virtual VMware.
 Valor de retorno: *Integer*.

Parâmetros:

- **url** - a URL do serviço VMware;
- **uuid** - o identificador global exclusivo da máquina virtual VMware.

vmware.vm.cpu.readiness[url,uuid,<instance>]

 A porcentagem do tempo em que a máquina virtual estava pronta, mas não conseguiu ser agendada para execução na CPU física.
 Valor de retorno: *Float*.

Parâmetros:

- **url** - o URL do serviço VMware;
- **uuid** - o identificador global exclusivo da máquina virtual VMware;
- **instance** - a instância da CPU.

vmware.vm.cpu.ready[url,uuid]

 O tempo (em milissegundos) que a máquina virtual estava pronta, mas não conseguiu ser agendada para rodar na CPU física. O tempo de CPU ready depende do número de máquinas virtuais no host e de suas cargas de CPU (%).
 Valor de retorno: *Integer* [2].

Parâmetros:

- **url** - a URL do serviço VMware;
- **uuid** - o identificador global exclusivo da máquina virtual VMware.

vmware.vm.cpu.swapwait[url,uuid,<instance>]

 A porcentagem do tempo da CPU gasto esperando pelo swap-in.
 Valor de retorno: *Float*.

Parâmetros:

- **url** - o URL do serviço VMware;
- **uuid** - o identificador global exclusivo da máquina virtual VMware;
- **instance** - a instância da CPU.

vmware.vm.cpu.usage[url,uuid]

 O uso do processador da máquina virtual VMware (Hz).
 Valor de retorno: *Integer*.

Parâmetros:

- **url** - o URL do serviço VMware;
- **uuid** - o identificador global exclusivo da máquina virtual VMware.

vmware.vm.cpu.usage.perf[url,uuid]

 O uso do processador da máquina virtual VMware como uma porcentagem durante o intervalo.
 Valor de retorno: *Float*.

Parâmetros:

- **url** - o URL do serviço VMware;
- **uuid** - o identificador global exclusivo da máquina virtual VMware.

vmware.vm.datacenter.name[url,uuid]

 O nome do datacenter da máquina virtual VMware.
 Valor de retorno: *String*.

Parâmetros:

- **url** - a URL do serviço VMware;
- **uuid** - o identificador global exclusivo da máquina virtual VMware.

vmware.vm.discoveryurl

 A descoberta de máquinas virtuais VMware.
 Valor de retorno: objeto JSON.

Parâmetros:

- **url** - a URL do serviço VMware.

vmware.vm.guest.memory.size.swapped[url,uuid]

 A quantidade de memória física do guest que foi trocada para o espaço de swap (KB).
 Valor de retorno: *Integer*.

Parâmetros:

- **url** - a URL do serviço VMware;
- **uuid** - o identificador global exclusivo da máquina virtual VMware.

vmware.vm.guest.osuptime[url,uuid]

 O tempo total decorrido desde a última inicialização do sistema operacional (em segundos).
 Valor de retorno: *Inteiro*.

Parâmetros:

- **url** - a URL do serviço VMware;
- **uuid** - o identificador global exclusivo da máquina virtual VMware.

vmware.vm.hv.name[url,uuid]

 O nome do hypervisor da máquina virtual VMware.
 Valor de retorno: *String*.

Parâmetros:

- **url** - a URL do serviço VMware;
- **uuid** - o identificador global exclusivo da máquina virtual VMware.

vmware.vm.memory.size[url,uuid]

 O tamanho total da memória da máquina virtual VMware (bytes).
 Valor de retorno: *Integer*.

Parâmetros:

- **url** - o URL do serviço VMware;
- **uuid** - o identificador global exclusivo da máquina virtual VMware.

vmware.vm.memory.size.ballooned[url,uuid]

 O tamanho da memória inflada da máquina virtual VMware (bytes).
 Valor de retorno: *Inteiro*.

Parâmetros:

- **url** - a URL do serviço VMware;
- **uuid** - o identificador global exclusivo da máquina virtual VMware.

vmware.vm.memory.size.compressed[url,uuid]

 O tamanho da memória compactada da máquina virtual VMware (bytes).
 Valor de retorno: *Integer*.

Parâmetros:

- **url** - a URL do serviço VMware;
- **uuid** - o identificador exclusivo global da máquina virtual VMware.

vmware.vm.memory.size.consumed[url,uuid]

 A quantidade de memória física do host consumida para backup das páginas de memória física do guest (KB).
 Valor de retorno: *Integer*.

Parâmetros:

- **url** - a URL do serviço VMware;
- **uuid** - o identificador global exclusivo da máquina virtual VMware.

vmware.vm.memory.size.private[url,uuid]

 O tamanho da memória privada da máquina virtual VMware (bytes).
 Valor de retorno: *Integer*.

Parâmetros:

- **url** - a URL do serviço VMware;
- **uuid** - o identificador global exclusivo da máquina virtual VMware.

vmware.vm.memory.size.shared[url,uuid]

 O tamanho da memória compartilhada da máquina virtual VMware (bytes).
 Valor de retorno: *Integer*.

Parâmetros:

- **url** - o URL do serviço VMware;
- **uuid** - o identificador exclusivo global da máquina virtual VMware.

vmware.vm.memory.size.swapped[url,uuid]

 O tamanho da memória trocada da máquina virtual VMware (bytes).
 Valor de retorno: *Integer*.

Parâmetros:

- **url** - o URL do serviço VMware;
- **uuid** - o identificador global exclusivo da máquina virtual VMware.

vmware.vm.memory.size.usage.guest[url,uuid]

 O uso de memória do guest da máquina virtual VMware (bytes).
 Valor de retorno: *Integer*.

Parâmetros:

- **url** - a URL do serviço VMware;
- **uuid** - o identificador global exclusivo da máquina virtual VMware.

vmware.vm.memory.size.usage.host[url,uuid]

 O uso de memória do host da máquina virtual VMware (bytes).
 Valor de retorno: *Integer*.

Parâmetros:

- **url** - a URL do serviço VMware;
- **uuid** - o identificador global exclusivo da máquina virtual VMware.

vmware.vm.memory.usage[url,uuid]

 A porcentagem da memória física do host que foi consumida.
 Valor de retorno: *Float*.

Parâmetros:

- **url** - o URL do serviço VMware;
- **uuid** - o identificador global exclusivo da máquina virtual VMware.

vmware.vm.net.if.discovery[url,uuid]

 A descoberta de interfaces de rede de máquinas virtuais VMware.
 Valor de retorno: objeto JSON.

Parâmetros:

- **url** - a URL do serviço VMware;
- **uuid** - o identificador global exclusivo da máquina virtual VMware.

vmware.vm.net.if.in[url,uuid,instance,<mode>]

 As estatísticas de entrada da interface de rede da máquina virtual VMware (bytes/pacotes por segundo).
 Valor de retorno: *Integer* [2].

Parâmetros:

- **url** - o URL do serviço VMware;
- **uuid** - o identificador global exclusivo da máquina virtual VMware;
- **instance** - a instância da interface de rede;
- **mode** - *bps* (padrão) ou *pps* - bytes ou pacotes por segundo.

vmware.vm.net.if.out[url,uuid,instance,<mode>]

 As estatísticas de saída da interface de rede da máquina virtual VMware (bytes/pacotes por segundo).
 Valor de retorno: *Integer* [2].

Parâmetros:

- **url** - a URL do serviço VMware;
- **uuid** - o identificador global exclusivo da máquina virtual VMware;
- **instance** - a instância da interface de rede;
- **mode** - *bps* (padrão) ou *pps* - bytes ou pacotes por segundo.

vmware.vm.net.if.usage[url,uuid,<instance>]

 A utilização da rede da máquina virtual VMware (taxas de transmissão e recepção combinadas) durante o intervalo (KBps).
 Valor de retorno: *Integer*.

Parâmetros:

- **url** - a URL do serviço VMware;
- **uuid** - o identificador global exclusivo da máquina virtual VMware;
- **instance** - a instância da interface de rede.

vmware.vm.perfcounter[url,uuid,path,<instance>]

 O valor do contador de desempenho da máquina virtual VMware.
 Valor de retorno: *Integer* [2].

Parâmetros:

- **url** - a URL do serviço VMware;
- **uuid** - o identificador global exclusivo da máquina virtual VMware;
- **path** - o caminho do contador de desempenho [1];
- **instance** - a instância do contador de desempenho. Use uma instância vazia para valores agregados (padrão).

vmware.vm.powerstate[url,uuid]

 O estado de energia da máquina virtual VMware.
 Valor de retorno: *0* - poweredOff; *1* - poweredOn; *2* - suspended.

Parâmetros:

- **url** - o URL do serviço VMware;
- **uuid** - o identificador global exclusivo da máquina virtual VMware.

vmware.vm.property[url,uuid,prop]

 A propriedade da máquina virtual VMware.
 Valor de retorno: *String*.

Parâmetros:

- **url** - a URL do serviço VMware;
- **uuid** - o identificador global exclusivo da máquina virtual VMware;
- **prop** - o caminho da propriedade, que é o nome de uma propriedade do objeto VM conforme definido no VMware SDK.

Exemplos:

```
vmware.vm.property[{$VMWARE.URL},{$VMWARE.VM.UUID},overallStatus]
vmware.vm.property[{$VMWARE.URL},{$VMWARE.VM.UUID},runtime.powerState]
```

vmware.vm.snapshot.get[url,uuid]

 O estado do snapshot da máquina virtual VMware.
 Valor de retorno: objeto JSON.

Parâmetros:

- **url** - a URL do serviço VMware;
- **uuid** - o identificador global exclusivo da máquina virtual VMware.

vmware.vm.state[url,uuid]

 O estado da máquina virtual VMware.
 Valor de retorno: *String*: *notRunning*, *resetting*, *running*, *shuttingDown*, *standby* ou *unknown*.

Parâmetros:

- **url** - a URL do serviço VMware;
- **uuid** - o identificador global exclusivo da máquina virtual VMware.

vmware.vm.storage.committed[url,uuid]

 O espaço de armazenamento comprometido da máquina virtual VMware (bytes).
 Valor de retorno: *Integer*.

Parâmetros:

- **url** - a URL do serviço VMware;
- **uuid** - o identificador global exclusivo da máquina virtual VMware.

vmware.vm.storage.readoio[url,uuid,instance]

 O número médio de solicitações de leitura pendentes para o disco virtual durante o intervalo de coleta.
 Valor de retorno: *Integer*.

Parâmetros:

- **url** - a URL do serviço VMware;
- **uuid** - o identificador global exclusivo da máquina virtual VMware;
- **instance** - a instância do dispositivo de disco.

vmware.vm.storage.totalreadlatency[url,uuid,instance]

 O tempo médio que uma leitura do disco virtual leva (milissegundos).
 Valor de retorno: *Integer*.

Parâmetros:

- **url** - a URL do serviço VMware;
- **uuid** - o identificador global exclusivo da máquina virtual VMware;
- **instance** - a instância do dispositivo de disco.

vmware.vm.storage.totalwritelatency[url,uuid,instance]

 O tempo médio que uma gravação no disco virtual leva (milissegundos).
 Valor de retorno: *Integer*.

Parâmetros:

- **url** - a URL do serviço VMware;
- **uuid** - o identificador global exclusivo da máquina virtual VMware;
- **instance** - a instância do dispositivo de disco.

vmware.vm.storage.uncommitted[url,uuid]

 O espaço de armazenamento não comprometido da máquina virtual VMware (bytes).
 Valor de retorno: *Integer*.

Parâmetros:

- **url** - a URL do serviço VMware;
- **uuid** - o identificador exclusivo global da máquina virtual VMware.

vmware.vm.storage.unshared[url,uuid]

 O espaço de armazenamento não compartilhado da máquina virtual VMware (bytes).
 Valor de retorno: *Integer*.

Parâmetros:

- **url** - o URL do serviço VMware;
- **uuid** - o identificador global exclusivo da máquina virtual VMware.

vmware.vm.storage.writeoio[url,uuid,instance]

 O número médio de solicitações de gravação pendentes no disco virtual durante o intervalo de coleta.
 Valor de retorno: *Integer*.

Parâmetros:

- **url** - a URL do serviço VMware;
- **uuid** - o identificador global exclusivo da máquina virtual VMware;
- **instance** - a instância do dispositivo de disco.

vmware.vm.tags.get[url,uuid]

 O array de tags da máquina virtual VMware.
 Valor de retorno: objeto JSON.

Parâmetros:

- **url** - a URL do serviço VMware;
- **uuid** - o identificador global exclusivo da máquina virtual VMware.

Este item funciona com o vSphere 6.5 e mais recente.

vmware.vm.tools[url,uuid,<mode>]

 O estado das ferramentas do guest da máquina virtual VMware.
 Valor de retorno: *String*.
 No modo *status*: *guestToolsExecutingScripts* - VMware Tools está iniciando; *guestToolsNotRunning* - VMware Tools não está em execução; *guestToolsRunning* - VMware Tools está em execução.
 No modo *version*: versão.

Parâmetros:

- **url** - a URL do serviço VMware;
- **uuid** - o identificador global exclusivo da máquina virtual VMware;
- **mode** - *version*, *status* (padrão).

vmware.vm.uptime[url,uuid]

 O tempo de atividade da máquina virtual VMware (segundos).
 Valor de retorno: *Inteiro*.

Parâmetros:

- **url** - o URL do serviço VMware;
- **uuid** - o identificador global exclusivo da máquina virtual VMware.

vmware.vm.vfs.dev.discovery[url,uuid]

 A descoberta de dispositivos de disco de máquina virtual VMware.
 Valor de retorno: objeto JSON.

Parâmetros:

- **url** - a URL do serviço VMware;
- **uuid** - o identificador global exclusivo da máquina virtual VMware.

vmware.vm.vfs.dev.read[url,uuid,instance,<mode>]

 As estatísticas de leitura do dispositivo de disco da máquina virtual VMware (bytes/operações por segundo).
 Valor de retorno: *Integer* [2].

Parâmetros:

- **url** - a URL do serviço VMware;
- **uuid** - o identificador global exclusivo da máquina virtual VMware;
- **instance** - a instância do dispositivo de disco;
- **mode** - *bps* (padrão) ou *ops* - bytes ou operações por segundo.

vmware.vm.vfs.dev.write[url,uuid,instance,<mode>]

 As estatísticas de gravação do dispositivo de disco da máquina virtual VMware (bytes/operações por segundo).
 Valor de retorno: *Integer* [2].

Parâmetros:

- **url** - a URL do serviço VMware;
- **uuid** - o identificador global exclusivo da máquina virtual VMware;
- **instance** - a instância do dispositivo de disco;
- **mode** - *bps* (padrão) ou *ops* - bytes ou operações por segundo.

vmware.vm.vfs.fs.discovery[url,uuid]

 A descoberta dos sistemas de arquivos da máquina virtual VMware.
 Valor de retorno: objeto JSON.

Parâmetros:

- **url** - a URL do serviço VMware;
- **uuid** - o identificador exclusivo global da máquina virtual VMware.

O VMware Tools deve estar instalado na máquina virtual convidada para que este item funcione.

vmware.vm.vfs.fs.size[url,uuid,fsname,<mode>]

 As estatísticas do sistema de arquivos da máquina virtual VMware (bytes/porcentagens).
 Valor de retorno: *Integer* - para bytes; *Float* - para porcentagem.

Parâmetros:

- **url** - a URL do serviço VMware;
- **uuid** - o identificador global exclusivo da máquina virtual VMware;
- **fsname** - o nome do sistema de arquivos;
- **mode** - *total*, *free*, *used*, *pfree* ou *pused*.

O VMware Tools deve estar instalado na máquina virtual convidada para que este item funcione.

Notas de rodapé

[1] Veja Criando nomes personalizados de contadores de desempenho para VMware.

[2] O valor desses items é obtido dos contadores de desempenho do VMware e o parâmetro VMwarePerfFrequency parâmetro é usado para atualizar seus dados no cache do Zabbix VMware:

- 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

Mais informações

Consulte Virtual machine monitoring para obter informações detalhadas informações detalhadas sobre como configurar o Zabbix para monitorar ambientes VMware.

2 Campos de chave de descoberta de máquina virtual

A tabela a seguir lista os campos retornados pelas chaves de descoberta relacionadas à máquina virtual.

Chave do item		
Descrição	**Campo**	**Conteúdo recuperado**
vmware.cluster.discovery		
Executa a descoberta de cluster.	{#CLUSTER.ID}	Identificador do cluster.
	{#CLUSTER.NAME}	Nome do cluster.

	”resource_pool”	Um array contendo dados do pool de recursos, incluindo ID do grupo de recursos, array de tags, caminho do pool de recursos, número de máquinas virtuais.
		
		Estrutura do array:
		`[{`
		`"rpid":"id do grupo de recursos",`
		`"tags":[{}],`
		`"rpath":"caminho do grupo de recursos",`
		`"vm_count":0`
		`}]`
		
		Para a estrutura do array `"tags"`, consulte o campo ”tags”.

	”tags”	Um array contendo tags com nome, descrição e categoria da tag.
		
		Estrutura do array:
		`[{`
		`"tag":"nome da tag",`
		`"tag_description":"descrição da tag",`
		`"category":"categoria da tag"`
		`}]`

vmware.datastore.discovery		
Executa a descoberta de datastore.	{#DATASTORE}	Nome do datastore.

	{#DATASTORE.EXTENT}	Um array contendo o ID da partição de extensão do datastore e o nome da instância.
		
		Estrutura do array:
		`[{`
		`"partitionid":1,`
		`"instance":"nome"`
		`}]`

	{#DATASTORE.TYPE}	Tipo de datastore. Exemplos de valores: `VMFS`, `NFS`, `vsan`, etc.
	{#DATASTORE.UUID}	Identificador do datastore.

Chave do item		

	”tags”	Um array contendo tags com nome, descrição e categoria da tag.
		
		Estrutura do array:
		`[{`
		`"tag":"nome da tag",`
		`"tag_description":"descrição da tag",`
		`"category":"categoria da tag"`
		`}]`

vmware.dc.discovery		
Executa a descoberta de datacenter.	{#DATACENTER}	Nome do datacenter.
	{#DATACENTERID}	Identificador do datacenter.

	”tags”	Um array contendo tags com nome, descrição e categoria da tag.
		
		Estrutura do array:
		`[{`
		`"tag":"nome da tag",`
		`"tag_description":"descrição da tag",`
		`"category":"categoria da tag"`
		`}]`

vmware.dvswitch.discovery		
Executa a descoberta de switches distribuídos do vSphere.	{#DVS.NAME}	Nome do switch.
	{#DVS.UUID}	Identificador do switch.
vmware.hv.discovery		
Executa a descoberta de hipervisores.	{#HV.UUID}	Identificador exclusivo do hipervisor.
	{#HV.ID}	Identificador do hipervisor (nome do objeto gerenciado HostSystem).
	{#HV.NAME}	Nome do hipervisor.
	{#HV.NETNAME}	Nome do host de rede do hipervisor.
	{#HV.IP}	Endereço IP do hipervisor, pode estar vazio. No caso de uma configuração HA com várias interfaces de rede, a seguinte prioridade de seleção para a interface é observada: - preferir o IP que compartilha a sub-rede IP com o IP do vCenter; - preferir o IP da sub-rede IP com gateway padrão; - preferir o IP da interface com o menor ID.
	{#CLUSTER.NAME}	Nome do cluster, pode estar vazio.
	{#DATACENTER.NAME}	Nome do datacenter.
	{#PARENT.NAME}	Nome do contêiner que armazena o hipervisor.
	{#PARENT.TYPE}	Tipo de contêiner no qual o hipervisor está armazenado. Os valores podem ser `Datacenter`, `Folder`, `ClusterComputeResource`, `VMware`, onde ”VMware” significa tipo de contêiner desconhecido.

	”resource_pool”	Um array contendo dados do pool de recursos, incluindo ID do grupo de recursos, array de tags, caminho do pool de recursos, número de máquinas virtuais.
		
		Estrutura do array:
		`[{`
		`"rpid":"id do grupo de recursos",`
		`"tags":[{}],`
		`"rpath":"caminho do grupo de recursos",`
		`"vm_count":0`
		`}]`
		
		Para a estrutura do array `"tags"`, consulte o campo ”tags”.

Chave do item		

	”tags”	Um array contendo tags com nome, descrição e categoria da tag.
		
		Estrutura do array:
		`[{`
		`"tag":"nome da tag",`
		`"tag_description":"descrição da tag",`
		`"category":"categoria da tag"`
		`}]`

vmware.hv.datastore.discovery		
Executa a descoberta de datastore de hipervisor. Observe que vários hipervisores podem usar o mesmo datastore.	{#DATASTORE}	Nome do datastore.
	{#DATASTORE.TYPE}	Tipo de datastore. Exemplos de valores: `VMFS`, `NFS`, `vsan`, etc.
	{#DATASTORE.UUID}	Identificador do datastore.
	{#MULTIPATH.COUNT}	Número registrado de caminhos do datastore.
	{#MULTIPATH.PARTITION.COUNT}	Número de partições de disco disponíveis.

	”datastore_extent”	Um array contendo o nome da instância de extensão do datastore e o ID da partição.
		
		Estrutura do array:
		`[{`
		`"partitionid":1,`
		`"instance":"nome"`
		`}]`

	”tags”	Um array contendo tags com nome, descrição e categoria da tag.
		
		Estrutura do array:
		`[{`
		`"tag":"nome da tag",`
		`"tag_description":"descrição da tag",`
		`"category":"categoria da tag"`
		`}]`

vmware.hv.net.if.discovery		
Executa a descoberta de interfaces de rede do hipervisor.	{#IFNAME}	Nome da interface.
	{#IFDRIVER}	Driver da interface.
	{#IFDUPLEX}	Configurações de duplex da interface.
	{#IFSPEED}	Velocidade da interface.
	{#IFMAC}	Endereço MAC da interface.
vmware.vm.discovery		
Executa a descoberta de máquina virtual.	{#VM.UUID}	Identificador exclusivo da máquina virtual.
	{#VM.ID}	Identificador da máquina virtual (nome do objeto gerenciado VirtualMachine).
	{#VM.NAME}	Nome da máquina virtual.
	{#HV.NAME}	Nome do hipervisor.
	{#HV.UUID}	Identificador exclusivo do hipervisor.
	{#HV.ID}	Identificador do hipervisor (nome do objeto gerenciado HostSystem).
	{#CLUSTER.NAME}	Nome do cluster, pode estar vazio.
	{#DATACENTER.NAME}	Nome do datacenter.
	{#DATASTORE.NAME}	Nome do datastore.
	{#DATASTORE.UUID}	Identificador do datastore.
	{#VM.IP}	Endereço IP da máquina virtual, pode estar vazio.
	{#VM.DNS}	Nome DNS da máquina virtual, pode estar vazio.
	{#VM.GUESTFAMILY}	Família do SO da máquina virtual guest, pode estar vazio.

Chave do item		
	{#VM.GUESTFULLNAME}	Nome completo do SO da máquina virtual guest, pode estar vazio.
	{#VM.FOLDER}	A cadeia de pastas pai da máquina virtual, pode ser usada como valor para grupos aninhados; os nomes das pastas são combinados com "/". Pode estar vazio.
	{#VM.TOOLS.STATUS}	Estado das ferramentas da máquina virtual VMware.
	{#VM.POWERSTATE}	Estado de energia da máquina virtual VMware (`poweredOff`, `poweredOn` ou `suspended`).
	{#VM.RPOOL.ID}	Identificador do pool de recursos.
	{#VM.RPOOL.PATH}	Caminho completo do pool de recursos, excluindo o nome "root" "Resources". Os nomes das pastas são combinados com "/".
	{#VM.SNAPSHOT.COUNT}	Número de snapshots da VM.

	"tags"	Um array contendo tags com nome, descrição e categoria da tag.
		
		Estrutura do array:
		`[{`
		`"tag":"nome da tag",`
		`"tag_description":"descrição da tag",`
		`"category":"categoria da tag"`
		`}]`

	"vm_customattribute"	Um array de atributos personalizados da máquina virtual (se definidos).
		
		Estrutura do array:
		`[{`
		`"name":"nome do campo personalizado",`
		`"value":"valor do campo personalizado"`
		`}]`

	"net_if"	Um array de interfaces de rede da máquina virtual.
		
		Estrutura do array:
		`[{`
		`"ifname": "nome da interface",`
		`"ifdesc": "descrição da interface",`
		`"ifmac": "00:00:00:00:00:00",`
		`"ifconnected": true,`
		`"iftype": "tipo da interface",`
		`"ifbackingdevice": "dispositivo de suporte da interface",`
		`"ifdvswitch_uuid": "uuid do switch da interface",`
		`"ifdvswitch_portgroup": "grupo de portas do switch da interface",`
		`"ifdvswitch_port": "porta do switch da interface",`
		`"ifip": ["endereços IP da interface"]`
		`}]`
		
		Para a descrição dos dados retornados, consulte a chave de item "vmware.vm.net.if.discovery".

vmware.vm.net.if.discovery		
Executa a descoberta de interfaces de rede da máquina virtual.	{#IFNAME}	Nome da interface de rede.
	{#IFDESC}	Descrição da interface.
	{#IFMAC}	Endereço MAC da interface.
	{#IFCONNECTED}	Status de conexão da interface (`false` - desconectada; `true` - conectada).
	{#IFTYPE}	Tipo da interface.
	{#IFBACKINGDEVICE}	Nome do dispositivo de suporte.
	{#IFDVSWITCH.UUID}	Identificador exclusivo do vSphere Distributed Switch.
	{#IFDVSWITCH.PORTGROUP}	Grupo de portas distribuídas.
	{#IFDVSWITCH.PORT}	Porta do vSphere Distributed Switch.
	"ifip"	Um array de endereços da interface.
vmware.vm.vfs.dev.discovery		

Chave do item		
Executa a descoberta de dispositivos de disco da máquina virtual.	{#DISKNAME}	Nome do dispositivo de disco.
vmware.vm.vfs.fs.discovery		
Executa a descoberta do sistema de arquivos da máquina virtual.	{#FSNAME}	Nome do sistema de arquivos.

3 exemplos de JSON para items VMware

Visão geral Esta seção fornece informações adicionais sobre objetos JSON retornados por vários items do VMware.

vmware.*.alarms.get Os items **vmware.alarms.get[]**, **vmware.cluster.alarms.get[]**, **vmware.datastore.alarms.get[]**, **vmware.dc.alarms.get[]**, **vmware.hv.alarms.get[]**, **vmware.vm.alarms.get[]** retornam objetos JSON com a seguinte estrutura (os valores são fornecidos como exemplo):

```
{
    "alarms": [
        {
            "name": "Host connection and power state",
            "system_name": "alarm.HostConnectionStateAlarm",
            "description": "Default alarm to monitor host connection and power state",
            "enabled": true,
            "key": "alarm-1.host-2013",
            "time": "2022-06-27T05:27:38.759976Z",
            "overall_status": "red",
            "acknowledged": false
        },
        {
            "name": "Host memory usage",
            "system_name": "alarm.HostMemoryUsageAlarm",
            "description": "Default alarm to monitor host memory usage",
            "enabled": true,
            "key": "alarm-4.host-1004",
            "time": "2022-05-16T13:32:42.47863Z",
            "overall_status": "yellow",
            "acknowledged": false
        },
        {
            // other alarms
        }
    ]
}
```

vmware.*.tags.get Os items **vmware.cluster.tags.get[]**, **vmware.datastore.tags.get[]**, **vmware.dc.tags.get[]**, **vmware.hv.tags.get[]**, **vmware.vm.tags.get[]** retornam objetos JSON com a seguinte estrutura (os valores são fornecidos como exemplo):

```
{
    "tags": [
        {
            "name": "Windows",
        "description": "tag for cat OS type",
        "category": "OS type"
    },
    {
            "name": "SQL Server",
```

```
        "description": "tag for cat application name",
        "category": "application name"
        },
    {
            // outras tags
        }
    ]
}
```

vmware.hv.diskinfo.get O item **vmware.hv.diskinfo.get[]** retorna objetos JSON com a seguinte estrutura (os valores são fornecidos como exemplo):

```
[
  {
    "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": {}
  },
  {
    // outras instâncias
  }
]
```

vmware.dvswitch.fetchports.get O item **vmware.dvswitch.fetchports.get[]** retorna objetos JSON com a seguinte estrutura (os valores são fornecidos como exemplo):

```
{
    "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"
            },
            {
                //other keys
            }
        ]
    }
}
```

vmware.hv.hw.sensors.get O item **vmware.hv.hw.sensors.get[]** retorna objetos JSON com a seguinte estrutura (os valores são fornecidos como exemplo):

```
{
    "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"
                 }
            },
            {
                    // other hv hw sensors
                }
        ]
        }
}
```

vmware.hv.sensors.get O item **vmware.hv.sensors.get[]** retorna objetos JSON com a seguinte estrutura (os valores são fornecidos como exemplo):

```
{
    "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"
        },
            {
                    // other hv sensors
                }
        ]
        }
}
```

vmware.vm.snapshot.get Se existirem snapshots, o item **vmware.snapshot.get[]** retorna um objeto JSON com a seguinte estrutura (os valores são fornecidos como exemplo):

```
{
  "snapshot": [
```

```
    {
      "name": "VM Snapshot 4%2f1%2f2022, 9:16:39 AM",
      "description": "Descr 1",
      "createtime": "2022-04-01T06:16:51.761Z",
      "size": 5755795171,
      "uniquesize": 5755795171
    },
    {
      "name": "VM Snapshot 4%2f1%2f2022, 9:18:21 AM",
      "description": "Descr 2",
      "createtime": "2022-04-01T06:18:29.164999Z",
      "size": 118650595,
      "uniquesize": 118650595
    },
    {
      "name": "VM Snapshot 4%2f1%2f2022, 9:37:29 AM",
      "description": "Descr 3",
      "createtime": "2022-04-01T06:37:53.534999Z",
      "size": 62935016,
      "uniquesize": 62935016
    }
  ],
  "count": 3,
  "latestdate": "2022-04-01T06:37:53.534999Z",
  "latestage": 22729203,
  "oldestdate": "2022-04-01T06:16:51.761Z",
  "oldestage": 22730465,
  "size": 5937380782,
  "uniquesize": 5937380782
}
```


Se não existir snapshot, o item **vmware.snapshot.get[]** retorna um objeto JSON com valores vazios:


```
{
  "snapshot": [],
  "count": 0,
  "latestdate": null,
  "latestage": 0,
  "oldestdate": null,
  "oldestage": 0,
  "size": 0,
  "uniquesize": 0
}
```


4 Exemplo de configuração de monitoramento do VMware

Visão geral

O exemplo a seguir descreve como configurar o Zabbix para monitorar máquinas virtuais VMware. Isso envolve:

- criar um host que representa seu ambiente VMware;
- criar uma regra de descoberta de baixo nível que descobre máquinas virtuais em seu ambiente VMware;
- criar um protótipo de host, com base no qual o Zabbix irá gerar hosts reais para as máquinas virtuais descobertas pela regra de descoberta de baixo nível.

Pré-requisitos

> **Note:**
> Este exemplo não cobre a configuração do VMware. Pressupõe-se que o VMware já esteja configurado.

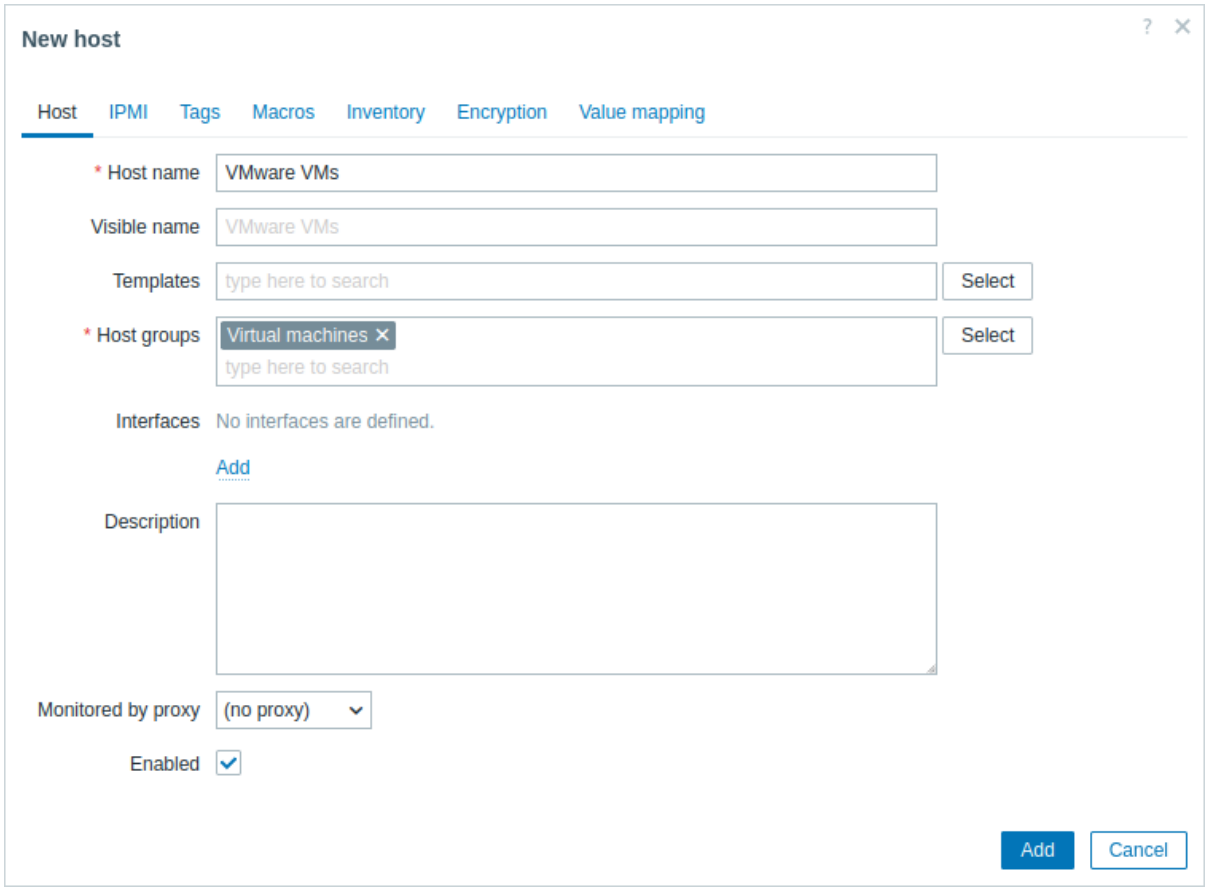
Antes de prosseguir, defina o parâmetro `StartVMwareCollectors` no arquivo de configuração do Zabbix server para 2 ou mais (o valor padrão é 0).

Crie um host

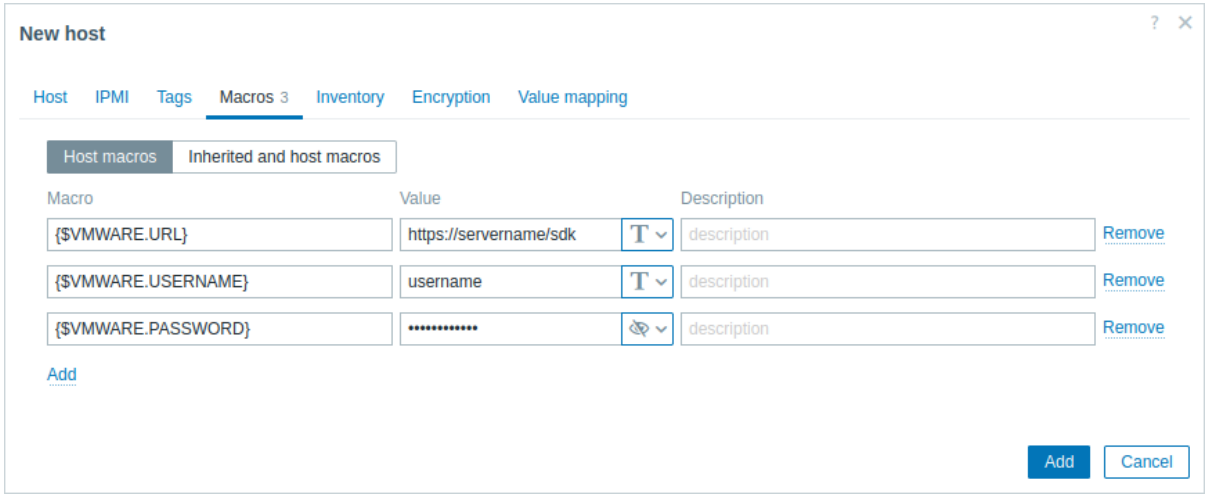
1. Vá para *Coleta de dados* → *Hosts*.

2. Crie um host:

- No campo *Nome do host*, insira um nome para o host (por exemplo, ”VMware VMs”).
- No campo *Grupos de hosts*, digite ou selecione um grupo de hosts (por exemplo, ”Virtual machines”).



- Na aba *Macros*, defina as seguintes macros de host:
  - {$VMWARE.URL} - URL do SDK do serviço VMware (hypervisor ESXi) (https://servername/sdk)
  - {$VMWARE.USERNAME} - Nome de usuário do serviço VMware
  - {$VMWARE.PASSWORD} - Senha do usuário {$VMWARE.USERNAME} do serviço VMware



3. Clique no botão *Adicionar* para criar o host. Este host representará seu ambiente VMware.

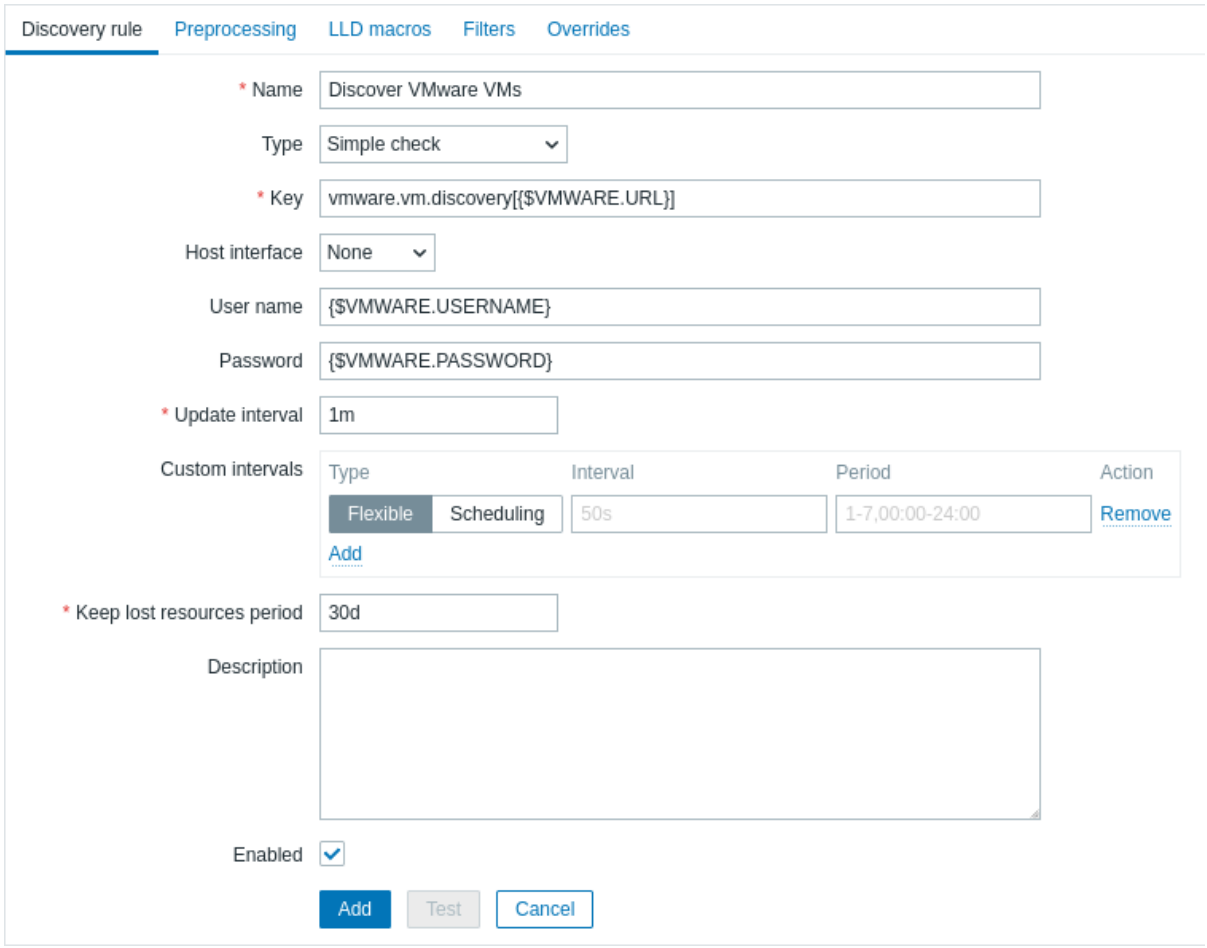
Crie uma regra de descoberta de baixo nível

1. Clique em *Descoberta* para o host criado para acessar a lista de regras de descoberta de baixo nível para esse host.

2. Crie uma regra de descoberta de baixo nível:

- No campo *Nome*, insira um nome para a regra de descoberta de baixo nível (por exemplo, ”Descobrir VMs VMware”).

- No campo *Tipo*, selecione "Verificação simples".
- No campo *Chave*, insira a chave de item interna para descobrir máquinas virtuais VMware: `vmware.vm.discovery[{$VMWARE.URL}]`
- Nos campos *Nome de usuário* e *Senha*, insira as macros correspondentes previamente configuradas no host.



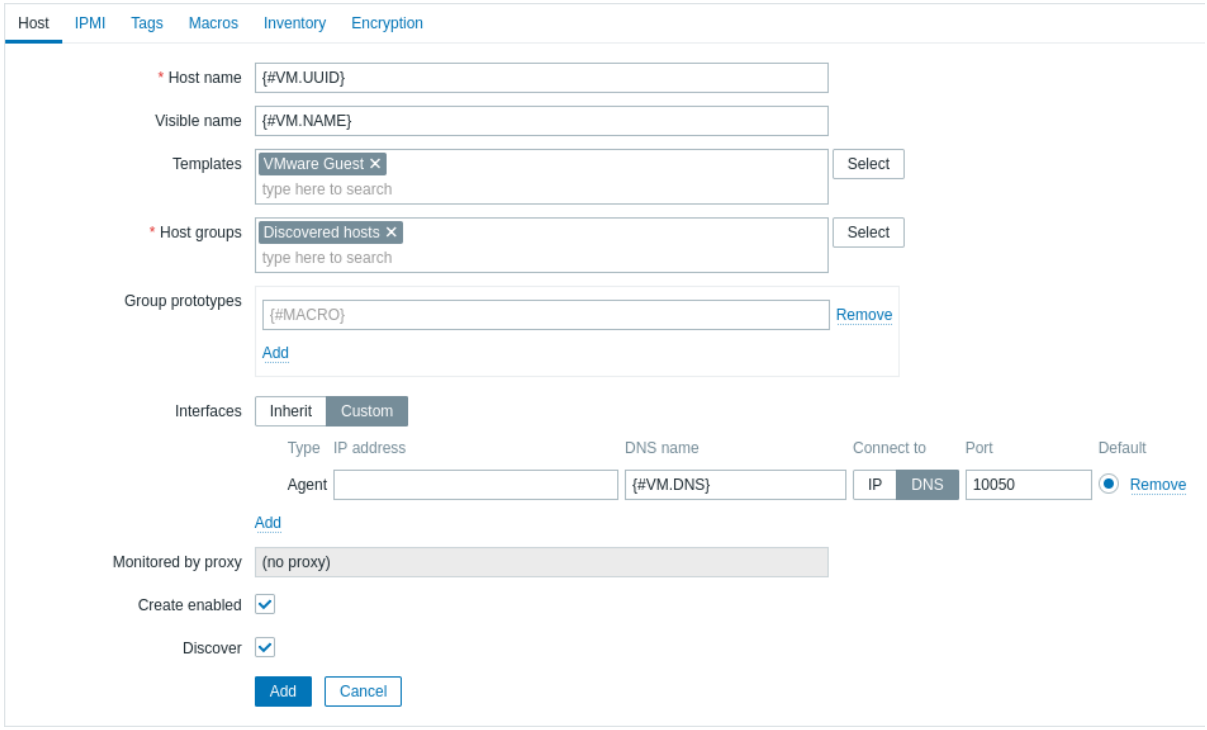
3. Clique no botão *Adicionar* para criar a regra de descoberta de baixo nível. Esta regra de descoberta irá descobrir máquinas virtuais em seu ambiente VMware.

Criar um protótipo de host

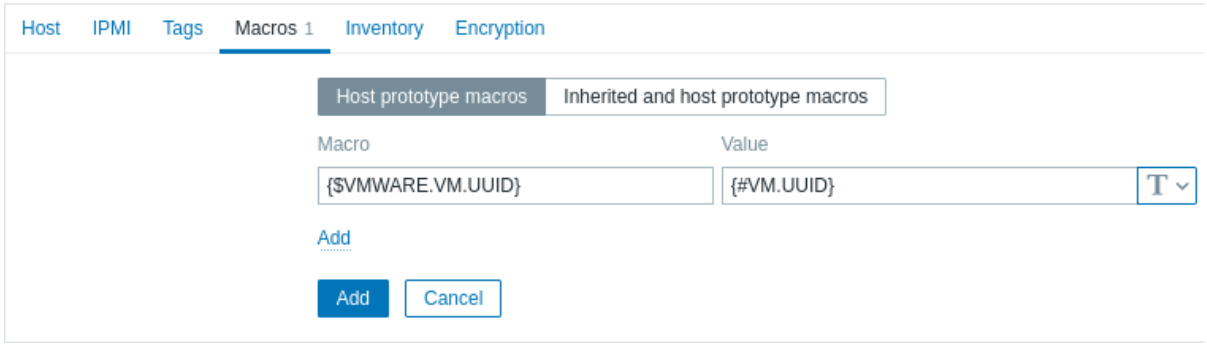
1. Na lista de regras de descoberta de baixo nível, clique em *Protótipos de host* para a regra de descoberta de baixo nível criada anteriormente.

2. Crie um protótipo de host. Como os protótipos de host são modelos para criar hosts por meio de regras de descoberta de baixo nível, a maioria dos campos conterá macros de descoberta de baixo nível. Isso garante que os hosts sejam criados com propriedades baseadas no conteúdo recuperado pela regra de descoberta de baixo nível criada anteriormente.

- No campo *Nome do host*, insira a macro {#VM.UUID}.
- No campo *Nome visível*, insira a macro {#VM.NAME}.
- No campo *Templates*, insira ou selecione o template "VMware Guest". Esse template contém itens VMware e regras de descoberta para monitorar o estado de energia de uma máquina virtual, uso de CPU, uso de memória, dispositivos de rede, etc.
- No campo *Grupos de hosts*, insira ou selecione um grupo de hosts (por exemplo, "Hosts descobertos").
- No campo *Interfaces*, adicione uma interface de host personalizada. Em seguida, insira a macro {#VM.DNS} no campo *Nome DNS* ou insira a macro {#VM.IP} no campo *Endereço IP*. Alternativamente, se as máquinas virtuais do seu ambiente VMware tiverem várias interfaces, prossiga para a seção *Configuração avançada de interface de host*. Configurar uma interface de host personalizada é necessário para o funcionamento correto do template *VMware Guest*.



- Na aba *Macros*, defina a macro {$VMWARE.VM.UUID} com o valor {#VM.UUID}. Isso é necessário para o funcionamento correto do template *VMware Guest*, que utiliza essa macro como uma macro de usuário em nível de host nos parâmetros do item (por exemplo, *vmware.vm.net.if.discovery[{$VMWARE.URL}, {$VMWARE.VM.UUID}]*).

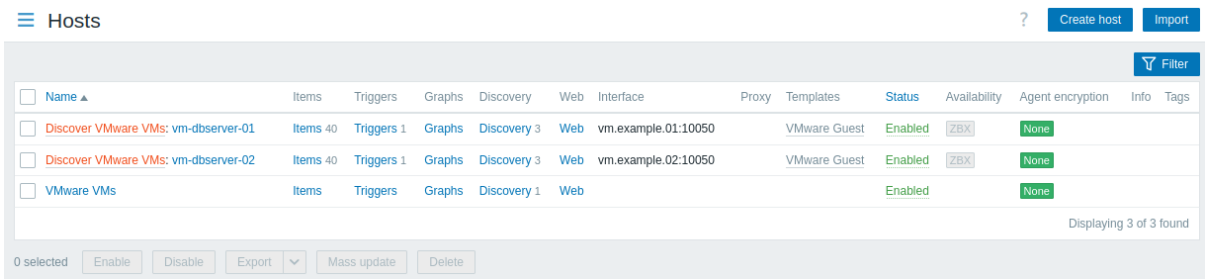


3. Clique no botão *Adicionar* para criar o protótipo de host. Esse protótipo de host será usado para criar hosts para máquinas virtuais descobertas pela regra de descoberta de baixo nível criada anteriormente.

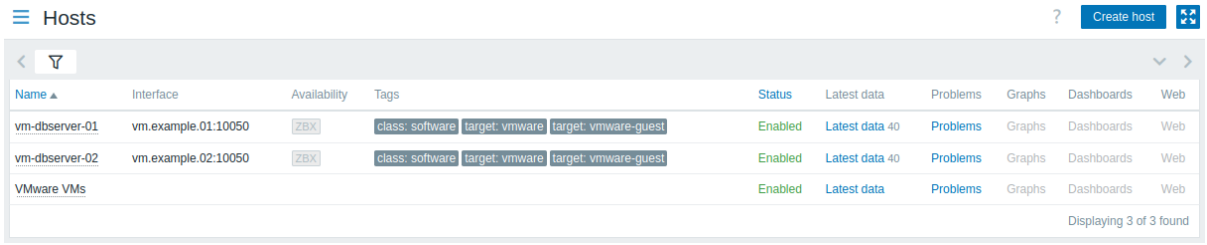
Visualizar hosts e métricas

Após a criação do protótipo de host, a regra de descoberta de baixo nível criará hosts para as máquinas virtuais VMware descobertas, e o Zabbix começará a monitorá-las. Observe que a descoberta e a criação de hosts também podem ser executadas manualmente, se necessário.

Para visualizar os hosts criados, navegue até a seção de menu *Coleta de dados → Hosts*.



Para visualizar as métricas coletadas, navegue até a seção de menu *Monitoramento → Hosts* e clique em *Últimos dados* para um dos hosts.



Configuração avançada de interface de host

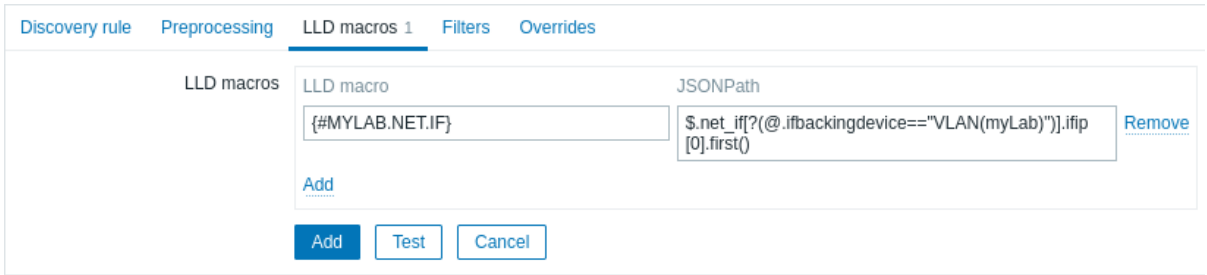
A chave de item `vmware.vm.discovery[{$VMWARE.URL}]`, configurada na seção *Criar uma regra de descoberta de baixo nível*, retorna dados das interfaces de rede no campo "net_if":

```
"net_if": [
      {
        "ifname": "5000",
        "ifdesc": "Network adapter 1",
        "ifmac": "00:11:22:33:44:55",
        "ifconnected": true,
        "iftype": "VirtualVmxnet3",
        "ifbackingdevice": "VLAN(myLab)",
        "ifdvswitch_uuid": "",
        "ifdvswitch_portgroup": "",
        "ifdvswitch_port": "",
        "ifip": [
          "127.0.0.1",
          "::1"
        ]
      },
      {
        "ifname": "5001",
        "ifdesc": "Network adapter 2",
        "ifmac": "00:11:22:33:44:55",
        "ifconnected": false,
        "iftype": "VirtualVmxnet3",
        "ifbackingdevice": "VLAN(myLab2)",
        "ifdvswitch_uuid": "",
        "ifdvswitch_portgroup": "",
        "ifdvswitch_port": "",
        "ifip": []
      }
    ]
```

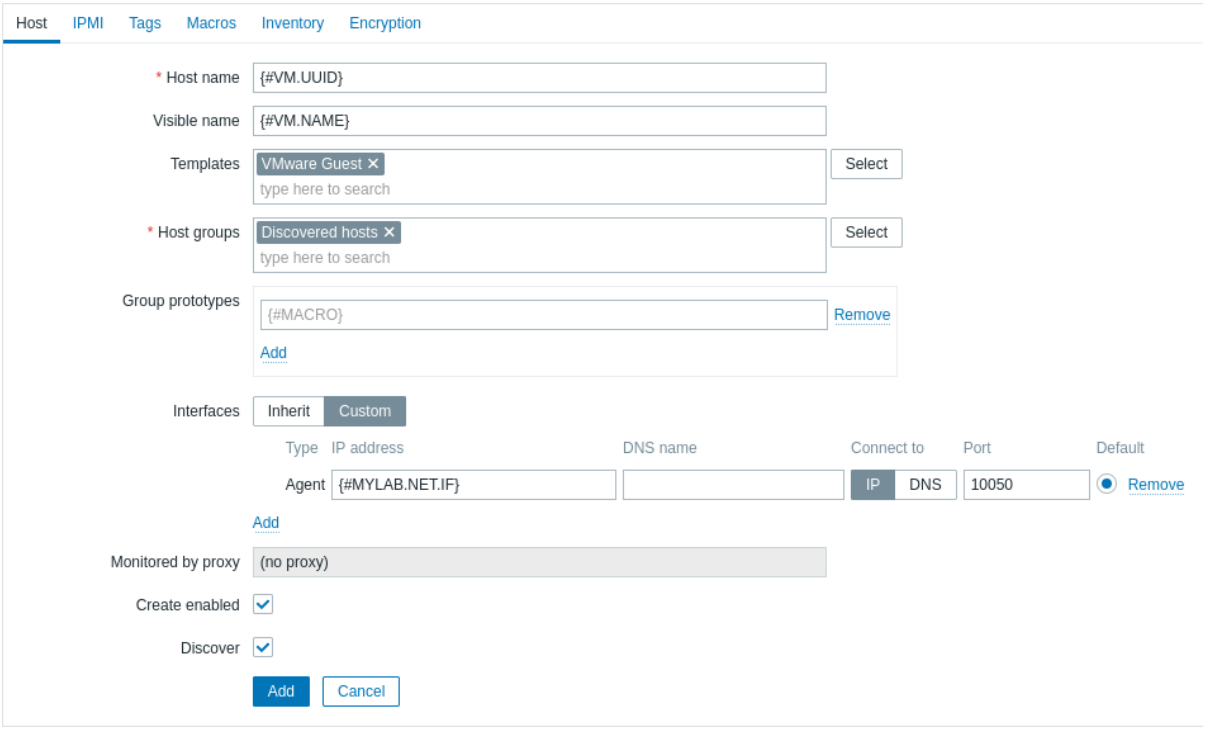
Esses dados podem ser usados para configurar uma interface de host personalizada.

1. Ao criar uma regra de descoberta de baixo nível, configure adicionalmente uma macro de descoberta de baixo nível. Na aba *Macros LLD*, crie uma macro LLD personalizada com um valor JSONPath. Por exemplo:

- `{#MYLAB.NET.IF}` - `$.net_if[?(@.ifbackingdevice=="VLAN(myLab)")].ifip[0].first()`



2. Ao criar um protótipo de host, adicione uma interface de host personalizada e insira a macro LLD no campo *Nome DNS* ou *Endereço IP*.



11. Manutenção

Visão geral Você pode definir períodos de manutenção para hosts e grupos de hosts no Zabbix.

Além disso, é possível definir manutenção apenas para um único trigger (ou subconjunto de triggers) especificando tags de trigger. Neste caso, a manutenção será ativada apenas para esses triggers; todos os outros triggers do host ou grupo de hosts não estarão em manutenção.

Existem dois tipos de manutenção - com coleta de dados e sem coleta de dados.

Durante uma manutenção ”com coleta de dados”, os triggers são processados normalmente e eventos são criados quando necessário. No entanto, as escalonamentos de problemas são pausados para hosts/triggers em manutenção, se a opção *Pausar operações para problemas suprimidos* estiver marcada na configuração da ação. Neste caso, as etapas de escalonamento que podem incluir o envio de notificações ou comandos remotos serão ignoradas enquanto durar o período de manutenção. Observe que as operações de recuperação e atualização de problemas não são suprimidas durante a manutenção, apenas os escalonamentos.

Por exemplo, se as etapas de escalonamento estiverem agendadas para 0, 30 e 60 minutos após o início de um problema, e houver uma manutenção de meia hora durando de 10 minutos a 40 minutos após o surgimento de um problema real, as etapas dois e três serão executadas meia hora depois, ou em 60 minutos e 90 minutos (desde que o problema ainda exista). Da mesma forma, se um problema surgir durante a manutenção, o escalonamento começará após a manutenção.

Para receber notificações de problemas durante a manutenção normalmente (sem atraso), você deve desmarcar a opção *Pausar operações para problemas suprimidos* na configuração da ação.

> **Note:**
> Se pelo menos um host (usado na expressão do trigger) não estiver em modo de manutenção, o Zabbix enviará uma notificação de problema.

O servidor Zabbix deve estar em execução durante a manutenção. As manutenções são recalculadas a cada minuto ou assim que o cache de configuração é recarregado se houver alterações no período de manutenção.

Os processos de temporizador verificam se o status do host deve ser alterado para/de manutenção a 0 segundos de cada minuto. Além disso, a cada segundo o processo de temporizador verifica se alguma manutenção deve ser iniciada/parada com base em alterações nos períodos de manutenção após a atualização da configuração. Assim, a velocidade de início/parada dos períodos de manutenção depende do intervalo de atualização da configuração (10 segundos por padrão). Observe que as alterações no período de manutenção não incluem as configurações *Ativo de/Ativo até*. Além disso, se um host/grupo de hosts for adicionado a um período de manutenção ativo existente, as alterações só serão ativadas pelo processo de temporizador no início do próximo minuto.

Observe que quando um host entra em manutenção, os processos de temporizador do servidor Zabbix lerão todos os problemas abertos para verificar se é necessário suprimir esses problemas. Isso pode ter impacto no desempenho se houver muitos problemas abertos. O servidor Zabbix também lerá todos os problemas abertos na inicialização, mesmo que não haja manutenções configuradas no momento.

Observe que o servidor Zabbix (ou proxy) sempre coleta dados independentemente do tipo de manutenção (incluindo manutenção "sem dados"). Os dados são posteriormente ignorados pelo servidor se 'sem coleta de dados' estiver definido.

Quando a manutenção "sem dados" termina, triggers que usam a função nodata() não serão disparados antes da próxima verificação durante o período que estão verificando.

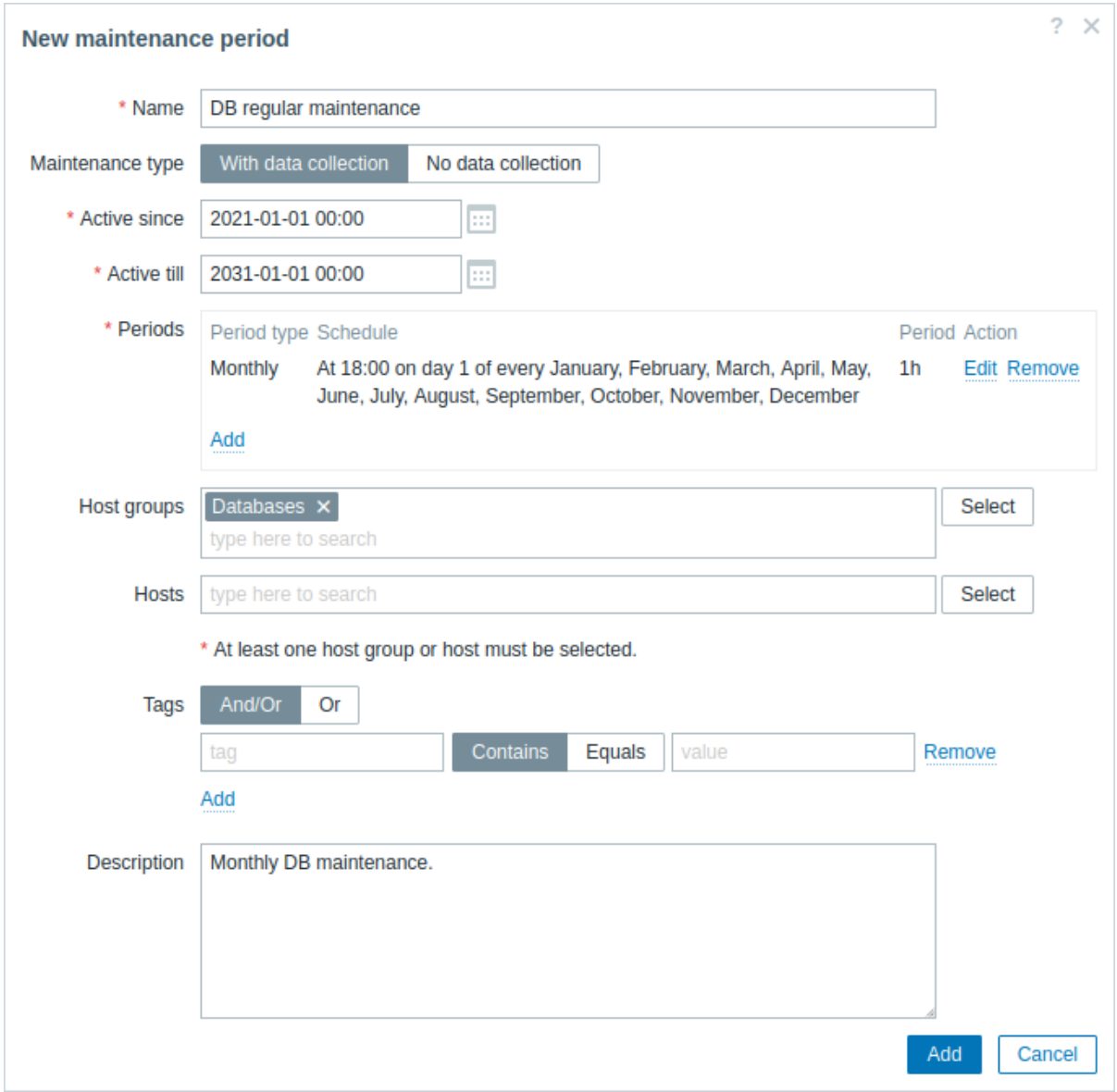
Se um item de log for adicionado enquanto um host está em manutenção e a manutenção terminar, apenas novas entradas do arquivo de log desde o final da manutenção serão coletadas.

Se um valor com carimbo de data/hora for enviado para um host que está em um tipo de manutenção "sem dados" (por exemplo, usando o Zabbix sender), esse valor será descartado, no entanto, é possível enviar um valor com carimbo de data/hora para um período de manutenção expirado e ele será aceito.

Se o período de manutenção, hosts, grupos ou tags forem alterados pelo usuário, as alterações só terão efeito após a sincronização do cache de configuração.

Configuração Para configurar um período de manutenção:

1. Vá para: *Coleta de dados* > *Manutenção*.
2. Clique em *Criar período de manutenção* (ou no nome de um período de manutenção existente).
3. Insira os parâmetros de manutenção no formulário.

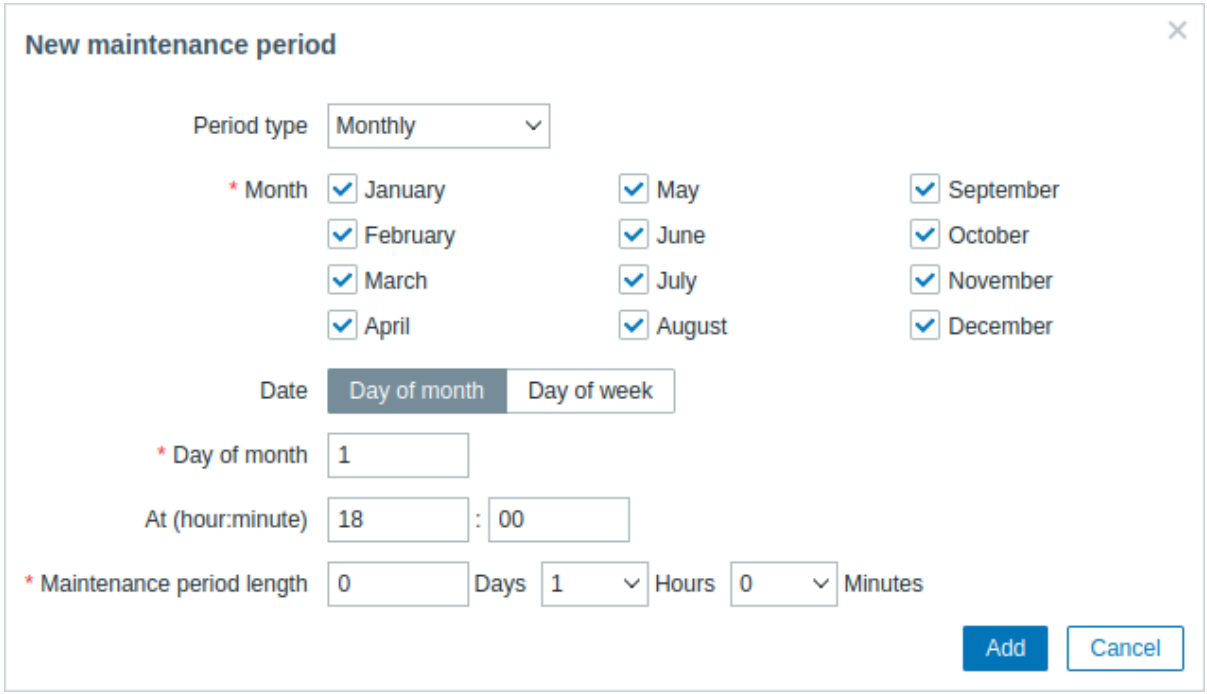


Todos os campos obrigatórios estão marcados com um asterisco vermelho.

Parâmetro	Descrição
Nome	Nome do período de manutenção.
Tipo de manutenção	Dois tipos de manutenção podem ser definidos: **Com coleta de dados** - os dados serão coletados pelo server durante a manutenção, triggers serão processados; **Sem coleta de dados** - os dados não serão coletados pelo server durante a manutenção. Veja Efeito dos períodos de manutenção para saber como cada tipo afeta o relatório de disponibilidade.
Ativo desde	A data e hora em que a execução dos períodos de manutenção se torna ativa. *Nota:* Definir apenas este horário não ativa um período de manutenção; os períodos de manutenção devem ser configurados em *Períodos* (veja abaixo).
Ativo até	A data e hora em que a execução dos períodos de manutenção deixa de estar ativa.
Períodos	Este bloco permite definir os dias e horários exatos em que a manutenção ocorrerá. Clicar em Add abre uma janela pop-up com um formulário flexível de *Período de manutenção* onde você pode definir o agendamento da manutenção. Veja Períodos de manutenção para uma descrição detalhada.
Grupos de hosts	Selecione os grupos de hosts para os quais a manutenção será ativada. A manutenção será ativada para todos os hosts do(s) grupo(s) de hosts especificado(s). Este campo é de preenchimento automático, portanto, ao começar a digitar, será exibida uma lista suspensa de todos os grupos de hosts disponíveis. Especificar um grupo de hosts pai seleciona implicitamente todos os grupos de hosts aninhados. Assim, a manutenção também será ativada nos hosts dos grupos aninhados.
Hosts	Selecione os hosts para os quais a manutenção será ativada. Este campo é de preenchimento automático, portanto, ao começar a digitar, será exibida uma lista suspensa de todos os hosts disponíveis.
Tags	Especifique tags para suprimir problemas com tags correspondentes em hosts em manutenção. Várias condições podem ser definidas. A correspondência do nome da tag diferencia maiúsculas de minúsculas. Existem dois operadores disponíveis para cada condição: **Contém** - inclui os nomes de tags especificados onde os valores das tags contêm a string inserida (correspondência de substring, diferencia maiúsculas de minúsculas); **Igual** - inclui os nomes e valores de tags especificados (diferencia maiúsculas de minúsculas). Existem dois tipos de cálculo para as condições: **E/Ou** - todas as condições devem ser atendidas, condições com o mesmo nome de tag serão agrupadas pela condição Ou; **Ou** - basta que uma condição seja atendida. As tags só podem ser especificadas se o modo *Com coleta de dados* estiver selecionado.
Descrição	Descrição do período de manutenção.

Períodos de manutenção

A janela de período de manutenção serve para agendar um horário para uma manutenção recorrente ou única. O formulário é dinâmico, com os campos disponíveis mudando com base no *Tipo de período* selecionado.



Tipo de período	Descrição
Apenas uma vez	Configurar um período de manutenção único: *Data* - data e hora do período de manutenção; *Duração do período de manutenção* - por quanto tempo a manutenção estará ativa.
Diariamente	Configurar um período de manutenção diário: *A cada dia(s)* - frequência da manutenção (1 - *(padrão)* todos os dias, 2 - a cada dois dias, etc.); *Às (hora:minuto)* - horário do dia em que a manutenção começa; *Duração do período de manutenção* - por quanto tempo a manutenção estará ativa. Quando o parâmetro *A cada dia(s)* for maior que "1", o dia de início será o dia em que o horário *Ativo desde* cair. Exemplos: - se *Ativo desde* estiver definido como "2021-01-01 12:00", *A cada dia(s)* estiver definido como "2" e *Às (hora:minuto)* estiver definido como "23:00", então o primeiro período de manutenção começará em 1º de janeiro às 23:00, enquanto o segundo período de manutenção começará em 3 de janeiro às 23:00; - se *Ativo desde* estiver definido como "2021-01-01 12:00", *A cada dia(s)* estiver definido como "2" e *Às (hora:minuto)* estiver definido como "01:00", então o primeiro período de manutenção começará em 3 de janeiro às 01:00, enquanto o segundo período de manutenção começará em 5 de janeiro às 01:00.
Semanalmente	Configurar um período de manutenção semanal: *A cada semana(s)* - frequência da manutenção (1 - *(padrão)* toda semana, 2 - a cada duas semanas, etc.); *Dia da semana* - em qual dia a manutenção deve ocorrer; *Às (hora:minuto)* - horário do dia em que a manutenção começa; *Duração do período de manutenção* - por quanto tempo a manutenção estará ativa. Quando o parâmetro *A cada semana(s)* for maior que "1", a semana de início será a semana em que o horário *Ativo desde* cair. Para exemplos, veja a descrição do parâmetro *Diariamente* acima.
Mensalmente	Configurar um período de manutenção mensal: *Mês* - selecione todos os meses durante os quais a manutenção regular será realizada; *Data:* ***Dia do mês*** - selecione esta opção se a manutenção deve ocorrer na mesma data de cada mês (por exemplo, todo dia 1º do mês) e, em seguida, selecione o dia necessário no campo *Dia do mês* que aparece; *Data:* ***Dia da semana*** - selecione esta opção se a manutenção deve ocorrer apenas em determinados dias (por exemplo, toda primeira segunda-feira do mês), depois selecione (no menu suspenso) a semana necessária do mês (primeira, segunda, terceira, quarta ou última) e, em seguida, marque as caixas de seleção para o(s) dia(s) de manutenção; *Às (hora:minuto)* - horário do dia em que a manutenção começa; *Duração do período de manutenção* - por quanto tempo a manutenção estará ativa.

Attention:
Ao criar um período de manutenção, o fuso horário do usuário que o cria é utilizado. No entanto, quando períodos de manutenção recorrentes (*Diário*, *Semanal*, *Mensal*) são agendados, o fuso horário do Zabbix server é utilizado. Para garantir um comportamento previsível dos períodos de manutenção recorrentes, é necessário usar um fuso horário comum para todas as partes do Zabbix.

Quando terminar, pressione *Adicionar* para adicionar o período de manutenção ao bloco *Períodos*.

Observe que as mudanças de Horário de Verão (DST) não afetam a duração da manutenção. Por exemplo, suponha que temos uma manutenção de duas horas configurada que normalmente começa às 01:00 e termina às 03:00:

- se após uma hora de manutenção (às 02:00) ocorrer uma mudança de DST e o horário atual mudar de 02:00 para 03:00, a manutenção continuará por mais uma hora (até às 04:00);
- se após duas horas de manutenção (às 03:00) ocorrer uma mudança de DST e o horário atual mudar de 03:00 para 02:00, a manutenção será encerrada, pois já se passaram duas horas;
- se um período de manutenção começar durante a hora que é pulada por uma mudança de DST, então a manutenção não será iniciada.

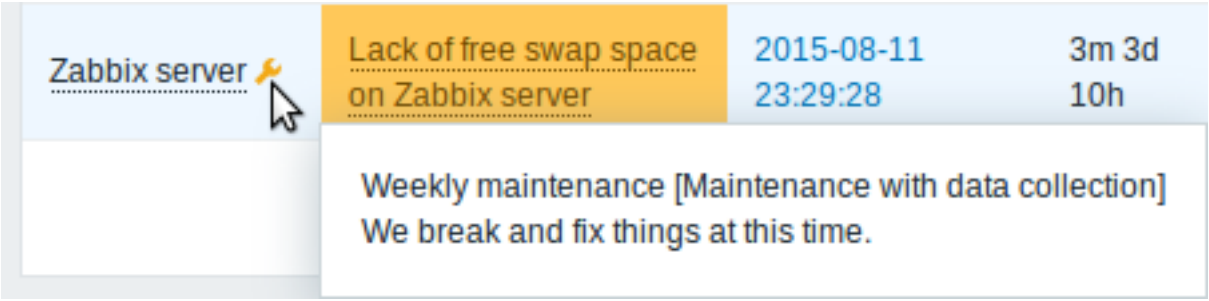
Se um período de manutenção for definido como "1 dia" (o período real da manutenção é de 24 horas, já que o Zabbix calcula dias em horas), começa às 00:00 e termina às 00:00 do dia seguinte:

- a manutenção terminará às 01:00 do dia seguinte se o horário atual avançar uma hora;
- a manutenção terminará às 23:00 do mesmo dia se o horário atual retroceder uma hora.

Exibição Exibindo hosts em manutenção

Um ícone de chave inglesa laranja ao lado do nome do host indica que este host está em manutenção em:

- *Dashboards*
- *Monitoramento* > *Problemas*
- *Inventário* > *Hosts* > *Detalhes do inventário do host*
- *Coleta de dados* > *Hosts* (Veja a coluna 'Status')



Os detalhes da manutenção são exibidos quando o ponteiro do mouse é posicionado sobre o ícone.

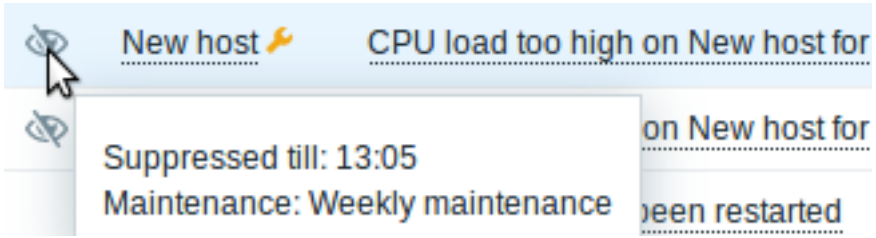
Além disso, hosts em manutenção recebem um fundo laranja em *Monitoramento* > *Mapas*.

Exibindo problemas suprimidos

Normalmente, problemas para hosts em manutenção são suprimidos, ou seja, não são exibidos no frontend. No entanto, também é possível configurar para que problemas suprimidos sejam exibidos, selecionando a opção *Exibir problemas suprimidos* nestes locais:

- *Dashboards* (na configuração dos widgets *Hosts com problemas*, *Problemas*, *Problemas por severidade*, *Visão geral de triggers*)
- *Monitoramento* > *Problemas* (no filtro)
- *Monitoramento* > *Mapas* (na configuração do mapa)
- Notificações globais (na configuração do perfil do usuário)

Quando problemas suprimidos são exibidos, o seguinte ícone é exibido: . Ao passar o mouse sobre o ícone, mais detalhes são exibidos.



Cálculo de filas durante a manutenção As filas exibidas no frontend do Zabbix (Administração > Fila) são calculadas pelo server Zabbix. Elas não incluem items em manutenção sem coleta de dados—a extensão da fila é sempre zero para esses items, mesmo quando seus valores estão atrasados. Items atrasados em manutenção com coleta de dados ainda são contabilizados na fila.

O proxy Zabbix não tem conhecimento dos períodos de manutenção porque não há sincronização da configuração de manutenção entre o server Zabbix e o proxy. Verificações internas calculadas em proxies Zabbix (por exemplo, `zabbix[queue,,]` e `zabbix[stats,,,queue,,]`) relatam items atrasados independentemente do status de manutenção no server Zabbix.

Como resultado, diferentes extensões de fila podem ser relatadas para os mesmos items em manutenção sem coleta de dados pelo frontend do Zabbix e por verificações internas em proxies Zabbix.

12. Expressões regulares

Visão geral Expressões regulares compatíveis com Perl (PCRE, PCRE2) são suportadas no Zabbix.

Há dois modos de usar expressões regulares no Zabbix:

- informando manualmente uma expressão regular
- usando uma expressão regular global criada no Zabbix

Expressões regulares Você pode inserir manualmente uma expressão regular nos locais suportados. Observe que a expressão não pode começar com @ porque esse símbolo é usado no Zabbix para referenciar expressões regulares globais.

> **Warning:**
> É possível ficar sem pilha ao usar expressões regulares. Consulte a página do manual do pcrestack para obter mais informações.

Observe que, na correspondência multilinha, os âncoras ^ e $ correspondem ao início/fim de cada linha, respectivamente, em vez do início/fim de toda a string.

Veja também exemplos de escape correto em vários contextos.

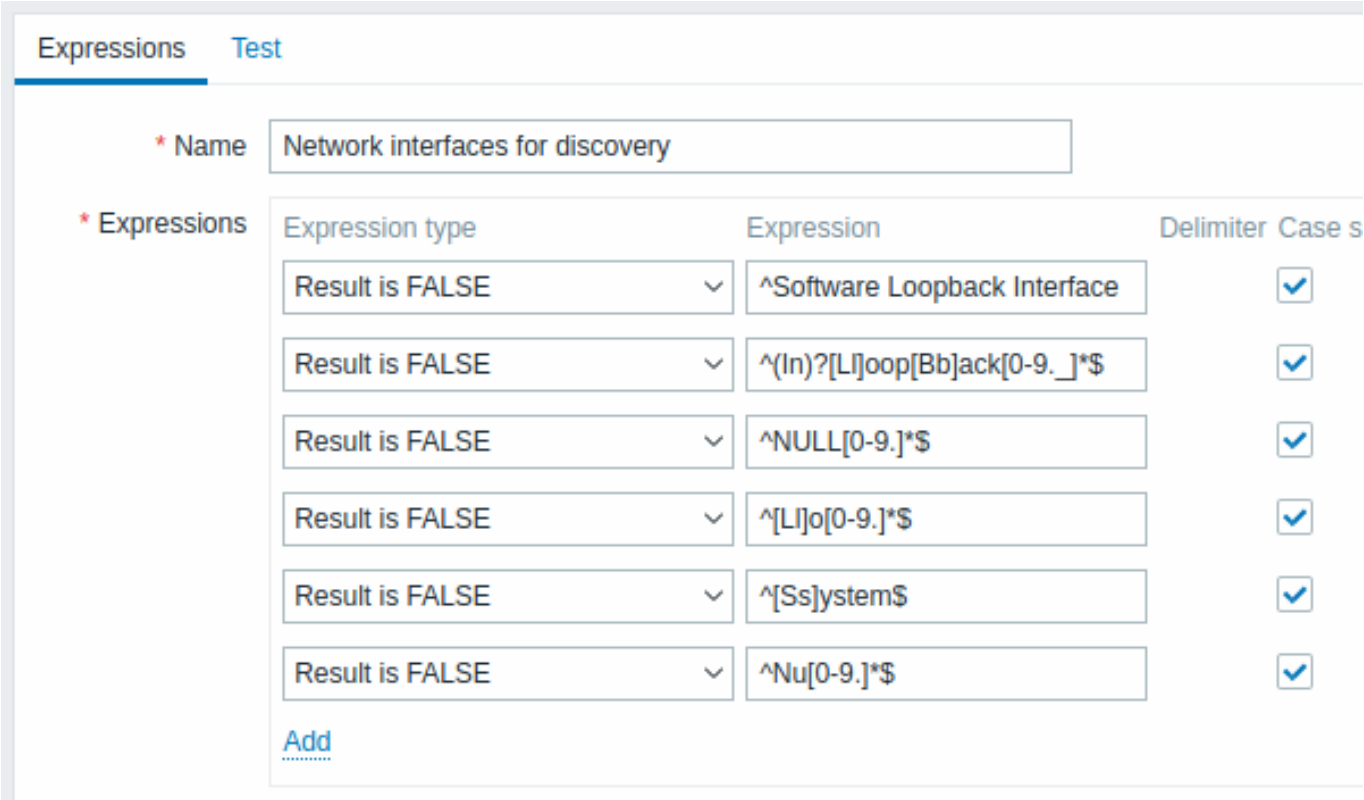
Expressões regulares globais Existe um editor avançado para criar e testar expressões regulares complexas no frontend do Zabbix.

Depois que uma expressão regular for criada dessa forma, ela poderá ser usada em vários locais no frontend, referindo-se ao seu nome, prefixado com @, por exemplo, *@minhaexpressaoregular*.

Para criar uma expressão regular global:

- Vá para: *Administração → Geral*
- Selecione *Expressões regulares* no menu suspenso
- Clique em *Nova expressão regular*

A aba **Expressões** permite definir o nome da expressão regular e adicionar subexpressões.



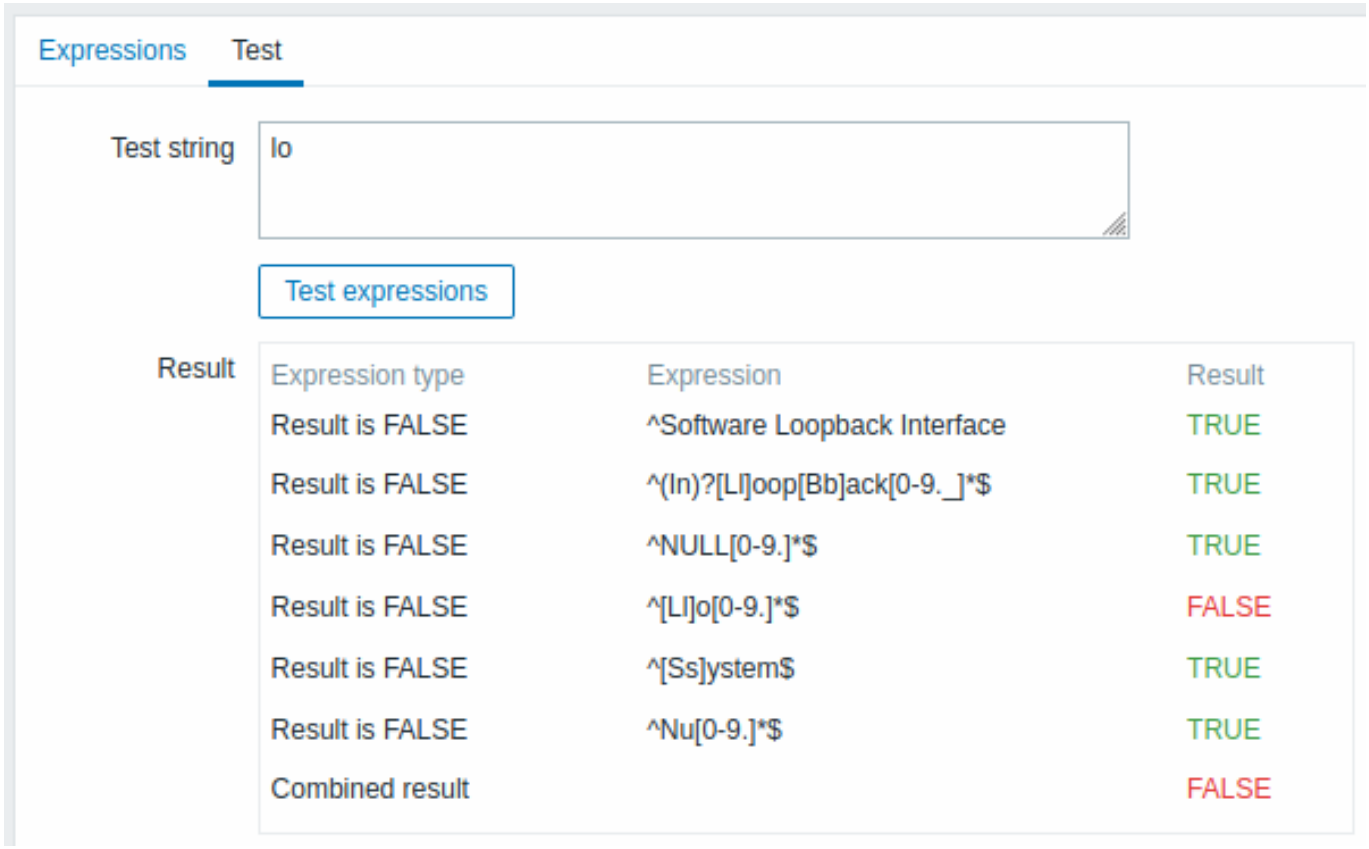
Todos os campos obrigatórios estão marcados com um asterisco vermelho.

Parâmetro	Descrição
Nome	Defina o nome da expressão regular. Qualquer caractere Unicode é permitido.
Expressões	Clique em *Adicionar* no bloco Expressões para adicionar uma nova subexpressão.
Tipo de expressão	Selecione o tipo de expressão: **String de caracteres incluída** - corresponde à substring **Qualquer string de caracteres incluída** - corresponde a qualquer substring de uma lista delimitada. A lista delimitada inclui uma vírgula (,), um ponto (.) ou uma barra (/). **String de caracteres não incluída** - corresponde a qualquer string, exceto a substring **Resultado é VERDADEIRO** - corresponde à expressão regular **Resultado é FALSO** - não corresponde à expressão regular
Expressão	Digite a substring/expressão regular.
Delimitador	Uma vírgula (,), um ponto (.) ou uma barra (/) para separar strings de texto em uma expressão regular. Este parâmetro só está ativo quando o tipo de expressão *"Qualquer string de caracteres incluída"* está selecionado.
Diferenciar maiús-cu-las de minús-cu-las	Uma caixa de seleção para especificar se uma expressão regular diferencia maiúsculas de minúsculas.

Uma barra (/) na expressão é tratada literalmente, e não como um delimitador. Dessa forma, é possível salvar expressões contendo uma barra, sem erros.

Attention:
Um nome de expressão regular personalizada no Zabbix pode conter vírgulas, espaços, etc. Nos casos em que isso possa levar a uma má interpretação ao referenciar (por exemplo, uma vírgula no parâmetro de uma chave de item), toda a referência pode ser colocada entre aspas, assim: *"@Minha expressão regular personalizada para propósito1, propósito2"*.
Os nomes das expressões regulares não devem ser colocados entre aspas em outros locais (por exemplo, nas propriedades da regra LLD).

Na aba **Testar**, a expressão regular e suas subexpressões podem ser testadas fornecendo uma string de teste.



Os resultados mostram o status de cada subexpressão e o status total da expressão personalizada.

O status total da expressão personalizada é definido como *Resultado combinado*. Se várias subexpressões forem definidas, o Zabbix usa o operador lógico AND para calcular o *Resultado combinado*. Isso significa que, se pelo menos um Resultado for Falso, o *Resultado combinado* também terá o status Falso.

Expressões regulares globais padrão O Zabbix vem com várias expressões regulares globais em seu conjunto de dados padrão.

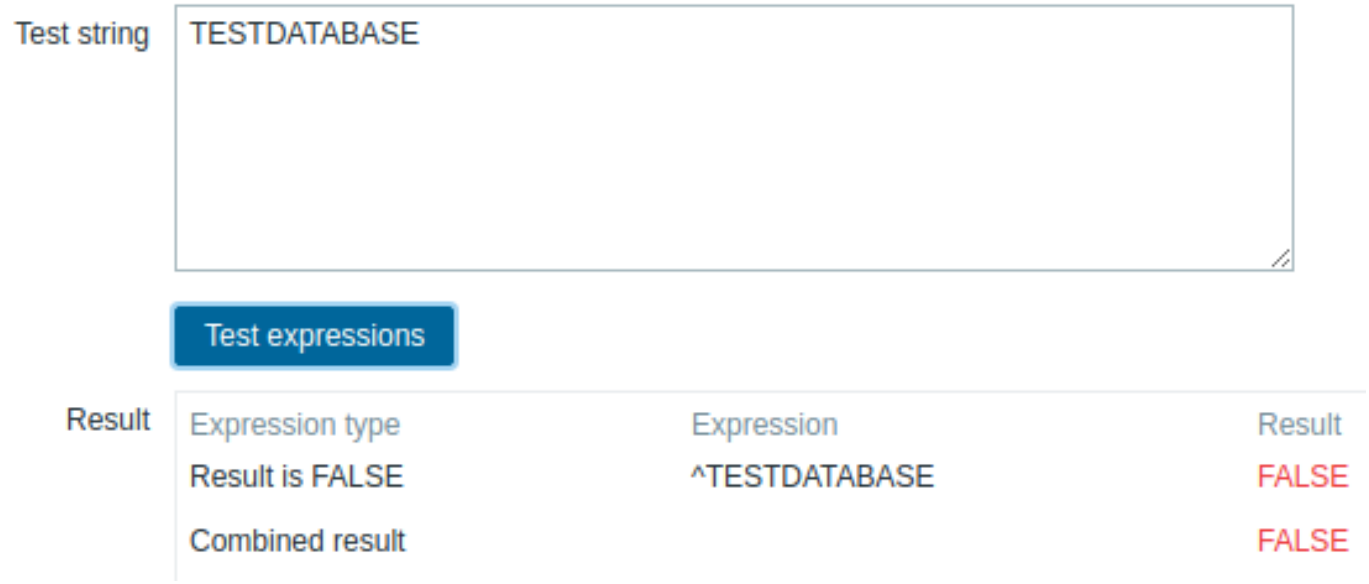
Nome	Expressão	Corresponde
Sistemas de arquivos para descoberta	`^(btrfs|ext2|ext3|ext4|jfs|reiser|xfs|ffs|ufs|jfs|jfs2|vxfs|hfs|refs|apfs|ntfs|fat32|zfs)`	”btrfs” ou ”ext2” ou ”ext3” ou ”ext4” ou ”jfs” ou ”reiser” ou ”xfs” ou ”ffs” ou ”ufs” ou ”jfs” ou ”jfs2” ou ”vxfs” ou ”hfs” ou ”refs” ou ”apfs” ou ”ntfs” ou ”fat32” ou ”zfs”
Interfaces de rede para descoberta	`^Software Loopback Interface`	Strings começando com ”Software Loopback Interface”.
	`^lo$`	”lo”
	`^(In)?[Ll]oop[Bb]ack[0-9._]*$`	Strings que opcionalmente começam com ”In”, depois têm ”L” ou ”l”, depois ”oop”, depois ”B” ou ”b”, depois ”ack”, que podem ser seguidas opcionalmente por qualquer número de dígitos, pontos ou sublinhados.
	`^NULL[0-9.]*$`	Strings começando com ”NULL” opcionalmente seguidas por qualquer número de dígitos ou pontos.
	`^[Ll]o[0-9.]*$`	Strings começando com ”Lo” ou ”lo” e opcionalmente seguidas por qualquer número de dígitos ou pontos.
	`^[Ss]ystem$`	”System” ou ”system”
	`^Nu[0-9.]*$`	Strings começando com ”Nu” opcionalmente seguidas por qualquer número de dígitos ou pontos.

Nome	Expressão	Corresponde
Dispositivos de armazenamento para descoberta SNMP	`^(Physical memory|Virtual memory|Memory buffers|Cached memory|Swap space)$`	”Physical memory” ou ”Virtual memory” ou ”Memory buffers” ou ”Cached memory” ou ”Swap space”
Nomes de serviços do Windows para descoberta	`^(MMCSS|gupdate|SysmonLog|clr_optimization_v4.0.30319_32)$`	”MMCSS” ou ”gupdate” ou ”SysmonLog” ou strings como ”clr_optimization_v2.0.50727_32” e ”clr_optimization_v4.0.30319_32” onde, em vez de pontos, você pode colocar qualquer caractere, exceto nova linha.
Estados de inicialização de serviços do Windows para descoberta	`^(automatic|automatic delayed)$`	”automatic” ou ”automatic delayed”

Exemplos Exemplo 1

Uso da seguinte expressão em descoberta de baixo-nível para descobrir bancos de dados, exceto um banco de dados com um nome específico:

`^TESTDATABASE$`

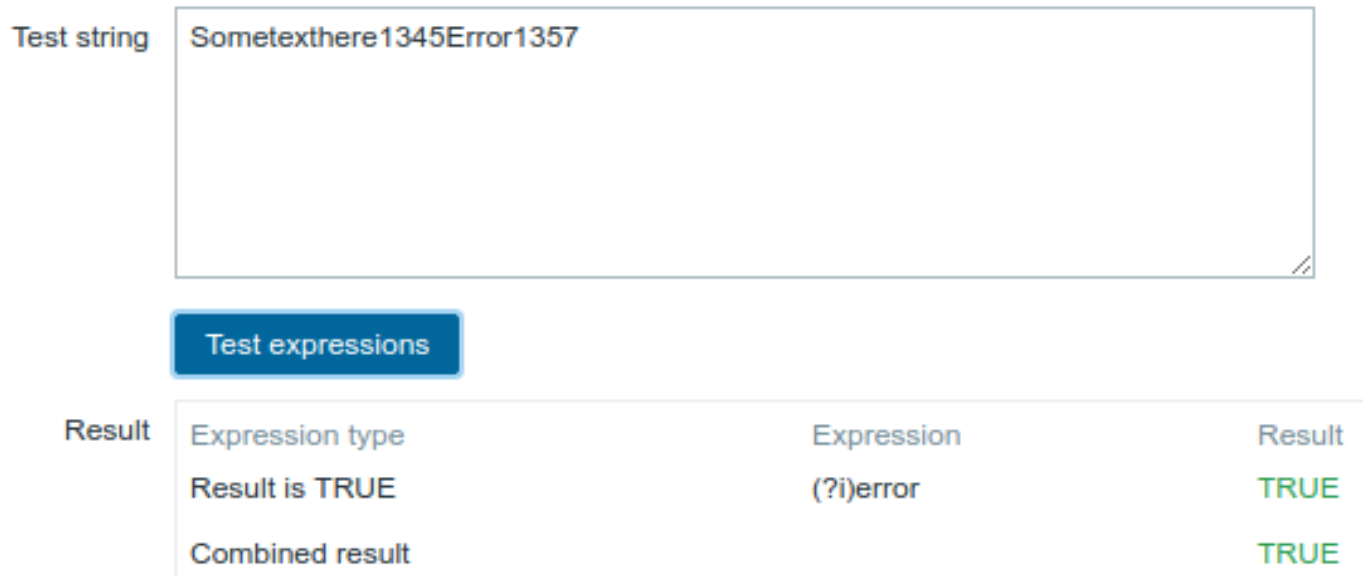


Tipo de expressão escolhido: ”Resultado é FALSO”. Não corresponde ao nome, contendo string ”*TESTDATABASE*”.

Exemplo com um modificador regex em linha

Uso da seguinte expressão regular incluindo um modificador em linha (?i) para corresponder os caracteres ”erro”:

`(?i)erro`



Tipo de expressão escolhido: ”Resultado é VERDADEIRO”. Os caracteres ”erro” são correspondidos.

Outro exemplo com um modificador regex em linha

Uso da seguinte expressão regular incluindo múltiplos modificadores em linha para corresponder aos caracteres após uma linha específica:

```
(?<=testar (?i)tudo(?-i) apos esta linha\n)(?sx).*# nós adicionamos o modificador s para permitir . corres
```

Tipo de expressão escolhido: ”Resultado é VERDADEIRO”. Os caracteres após uma linha específica são correspondidos.

Attention:
O modificador **g** não pode ser especificado em linha. A lista de modificadores disponíveis pode ser encontrada na página de manual pcresyntax. Para mais informações sobre a sintaxe PCRE, por favor consulte a documentação HTML PCRE.

Suporte a expressões regulares por local

Localização	Expressão regular	Expressão regular global	Correspondência multilinha	Comentários
Itens do agent				
eventlog[]	Sim	Sim	Sim	Parâmetros `regexp`, `severity`, `source`, `eventid`
eventlog.count[]				Parâmetros `regexp`, `severity`, `source`, `eventid`
log[]				Parâmetro `regexp`
log.count[]				
logrt[]		Sim/Não		O parâmetro `regexp` suporta ambos, o parâmetro `file_regexp` suporta apenas expressões não globais
logrt.count[]				

Localização	Expressão regular	Expressão regular global	Correspondência multilinha	Comentários
proc.cpu.util[] proc.get[] proc.mem[] proc.num[]		Não	Não	Parâmetro `cmdline`
sensor[]				Parâmetros `device` e `sensor` no Linux 2.4
system.hw.macaddr[]				Parâmetro `interface`
system.sw.packages[]				Parâmetro `regexp`
system.sw.packages.get[]				Parâmetro `regexp`
vfs.dir.count[]				Parâmetros `regex_incl`, `regex_excl`, `regex_excl_dir`
vfs.dir.get[]				Parâmetros `regex_incl`, `regex_excl`, `regex_excl_dir`
vfs.dir.size[]				Parâmetros `regex_incl`, `regex_excl`, `regex_excl_dir`
vfs.file.regexp[] vfs.file.regmatch[] web.page.regexp[]			Sim	Parâmetro `regexp`
Traps SNMP				
snmptrap[]	Sim	Sim	Não	Parâmetro `regexp`
Pré-processamento de valor de item	Sim	Não	Não	Parâmetro `pattern`
Funções para trig-gers/itens cal-cu-la-dos				
count()	Sim	Sim	Sim	Parâmetro `pattern` se o parâmetro `operator` for *regexp* ou *iregexp*
countunique()	Sim	Sim		
find()	Sim	Sim		
logeventid()	Sim	Sim	Não	Parâmetro `pattern`
logsource()				
Descoberta de baixo nível				
Filtros	Sim	Sim	Não	Campo *Expressão regular*
Substituições	Sim	Não		Nas opções *corresponde*, *não corresponde* para condições de *Operação*
Condições de ação	Sim	Não	Não	Nas opções *corresponde*, *não corresponde* para condições de autoregistro *Nome do host* e *Metadados do host*
Scripts	Sim	Sim	Não	Campo *Regra de validação de entrada*
Monitoramento web	Sim	Não	Sim	*Variáveis* com prefixo **regex:** campo *String obrigatória*

Localização	Expressão regular	Expressão regular global	Correspondência multilinha	Comentários
Contexto de macro de usuário Funções de macro	Sim	Não	Não	No contexto da macro com prefixo **regex:**
regsub() iregsub()	Sim	Não	Não	Parâmetro `pattern`
Mapeamento de ícones	Sim	Sim	Não	Campo *Expressão*
Mapeamento de val-ores	Sim	Não	Não	Campo *Valor* se o tipo de mapeamento for `regexp`

13 Reconhecimento do Problema

Visão geral Os eventos problemáticos no Zabbix podem ser reconhecidos pelos usuários. Se um usuário é notificado sobre um evento problemático, ele pode acessar o frontend Zabbix, abrir a janela de atualização do problema desse evento usando uma das formas listadas abaixo e reconhecer o problema.

Ao reconhecer o problema, ele pode inserir comentários, dizendo que está trabalhando nele ou qualquer outra coisa que deseje escrever a respeito.

Dessa forma, se outro usuário do sistema identificar o mesmo problema, ele imediatamente verá se esse problema já foi reconhecimento e comentado.

Desta forma, o workflow para resolver problemas com mais de um usuário do sistema pode ocorrer de forma coordenada.

O Status de reconhecimento também é utilizado ao definir operações de ação action operations. You can define, for example, that a notification is sent to a higher level manager only if an event is not acknowledged for some time. Por exemplo, você pode definir que uma notificação seja enviada para um gerente de nível superior somente se um evento não for reconhecido por algum tempo.

Para reconhecer eventos e fazer comentários, o usuário deve ter ao menos permissões de leitura correspondentes aos triggers. Para alterar a severidade ou fechar o problema, o usuário deve ter permissões de leitura-escrita correspondentes aos triggers.

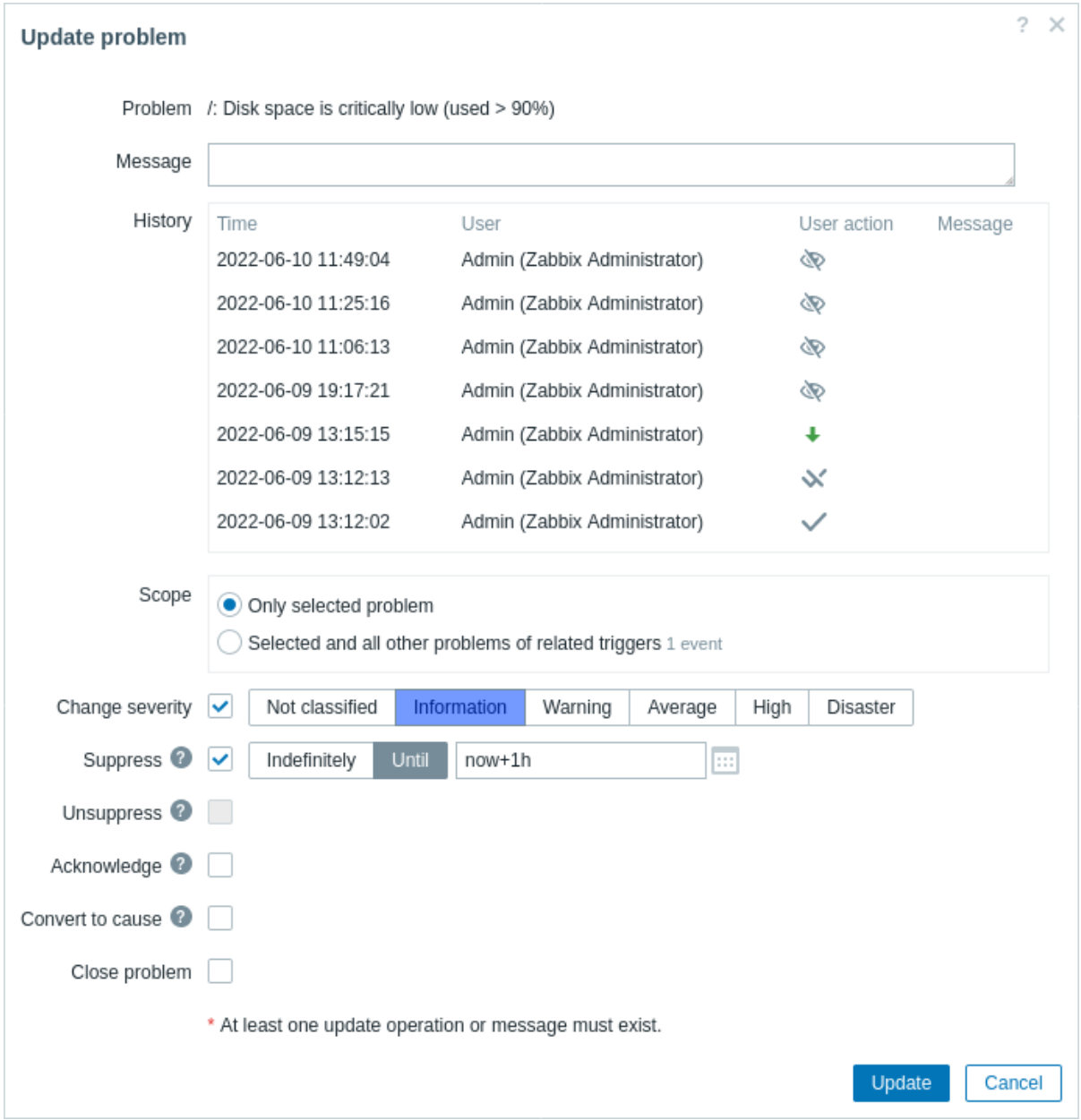
Há ** diversas** formas de acessar a janela de atualização do problema, que permitem reconhecer o problema. There are **several** ways to access the problem update popup window, which allows acknowledging a problem.

- Você pode selecionar problemas em *Monitoramento* → *Problemas* e então clicar em *Atualização em massa* abaixo da lista
- Você pode clicar em *Atualizar* na coluna de *Atualizar* o problema em:
  - *Painel de controles* (*Problemas* e *Problemas por severidade* widgets)
  - *Monitoramento* → *Problemas*
  - *Monitoramento* → *Problemas* → *Detalhes do evento*
- Você pode clicar em um problema não resolvido em:
  - *Painel de controles* (*Visão geral do Trigger* widget)

O menu popup contem uma opção de *Atualização* que levará você até a janela de atualização do problema.

Atualizando problemas O pop-up de atualização de problemas permite:

- comentar sobre o problema
- visualizar comentários e ações até o momento
- alterar a severidade do problema
- suprimir/retirar a supressão do problema
- reconhecer/remover o reconhecimento do problema
- alterar o problema sintoma para problema causa
- fechar o problema manualmente



Todos os campos obrigatórios estão marcados com um asterisco vermelho.

Parâmetro	Descrição
Problema	Se apenas um problema for selecionado, o nome do problema será exibido. Se vários problemas forem selecionados, será exibido *N problemas selecionados*.
Mensagem	Digite um texto para comentar sobre o problema (máximo de 2048 caracteres).
Histórico	As atividades e comentários anteriores sobre o problema são listados, juntamente com o horário e detalhes do usuário. Para o significado dos ícones usados para denotar ações do usuário, consulte a página de detalhes do evento. Observe que o histórico é exibido apenas se um problema for selecionado para atualização.
Escopo	Defina o escopo de ações como alteração de severidade, reconhecimento ou fechamento manual de problemas: **Apenas o problema selecionado** - afetará apenas este evento **Problemas selecionados e todos os outros problemas dos triggers relacionados** - no caso de reconhecimento/fechamento do problema, afetará este evento e todos os outros problemas que ainda não foram reconhecidos/fechados. Se o escopo contiver problemas já reconhecidos ou fechados, esses problemas não serão reconhecidos/fechados novamente. Por outro lado, o número de operações de mensagem e alteração de severidade não é limitado.

Parâmetro	Descrição
Alterar severidade	Marque a caixa de seleção e clique no botão de severidade para atualizar a severidade do problema. A caixa de seleção para alterar a severidade estará disponível se houver permissões de leitura e gravação para pelo menos um dos problemas selecionados. Somente esses problemas que possuem permissão de leitura e gravação serão atualizados ao clicar em *Atualizar*. Se não houver permissões de leitura e gravação para nenhum dos triggers selecionados, a caixa de seleção estará desabilitada.
Suprimir	Marque a caixa de seleção para suprimir o problema: **Indefinidamente** - suprimir indefinidamente **Até** - suprimir até um determinado horário. Ambos os formatos de tempo absoluto e relativo são suportados, por exemplo: `now+1d` - por um dia a partir de agora (padrão) `now/w` - até o final da semana atual `2022-05-28 12:00:00` - até a data/hora absoluta Observe que um período simples (por exemplo, `1d`, `1w`) não é suportado. A disponibilidade desta opção depende das configurações de função de usuário "Suprimir problemas". Veja também: Supressão de problemas
Retirar supressão	Marque a caixa de seleção para retirar a supressão do problema. Esta caixa de seleção está ativa apenas se pelo menos um dos problemas selecionados estiver suprimido. A disponibilidade desta opção depende das configurações de função de usuário "Suprimir problemas".
Reconhecer	Marque a caixa de seleção para reconhecer o problema. Esta caixa de seleção está disponível se houver pelo menos um problema não reconhecido entre os selecionados. Não é possível adicionar outro reconhecimento para um problema já reconhecido (é possível adicionar outro comentário, no entanto).
Remover reconhecimento	Marque a caixa de seleção para remover o reconhecimento do problema. Esta caixa de seleção está disponível se houver pelo menos um problema reconhecido entre os selecionados.
Converter para causa	Marque a caixa de seleção para converter o(s) problema(s) sintoma(s) em problema(s) causa(s).
Fechar problema	Marque a caixa de seleção para fechar manualmente o(s) problema(s) selecionado(s). A caixa de seleção para fechar um problema estará disponível se a opção *Permitir fechamento manual* estiver marcada na configuração do trigger para pelo menos um dos problemas selecionados. Somente esses problemas serão fechados ao clicar em *Atualizar*. Se nenhum problema puder ser fechado manualmente, a caixa de seleção estará desabilitada. Problemas já fechados não serão fechados novamente.

Exibição Com base nas informações de reconhecimento, é possível configurar como o contador de problemas será exibido no painel ou mapas. Para fazer isso, você precisa fazer seleções na opção *Exibição de problema*, disponível tanto na configuração do mapa map configuration quanto no widget *Problemas por gravidade* dashboard widget. É possível exibir o total de problemas, o contador de problemas não reconhecidos separados do total ou apenas o contador de problemas não reconhecidos. Com base nas informações de atualização de problemas (reconhecimento, etc.), é possível configurar operações de atualização - enviar uma mensagem ou executar comandos remotos.

1 Supressão de problemas

COMENTE https://git.zabbix.com/projects/WEB/repos/documentation/browse/en/manual/acknowledgment/suppression.md?at=refs%2Fheads
CONTEXTO 0d6af14b RECURSO Documentação 6.4•manual/acknowledgement/suppression.xliff

Visão geral

A supressão de problemas oferece uma maneira de temporariamente ocultar um problema que pode ser resolvido posteriormente. Isso é útil para limpar a lista de problemas, dando alta prioridade às questões mais urgentes. Por exemplo, às vezes surge um problema no fim de semana que não é urgente o suficiente para ser resolvido imediatamente, então pode ser "adiado" até segunda-feira de manhã.

A supressão de problemas permite ocultar um *único* problema, ao contrário da supressão de problemas por meio da manutenção do host, quando todos os problemas do host em manutenção são ocultados.

Operações para ações de triggers serão pausadas para problemas suprimidos da mesma forma que é feito com a manutenção do host host maintenance.

Configuração

Um problema pode ser suprimido através da janela de atualização de problemas **problem update**, onde a supressão é uma das opções de atualização do problema, juntamente com comentários, alteração de gravidade, reconhecimento, etc.

Um problema também pode ser desativado através da mesma janela de atualização de problemas.

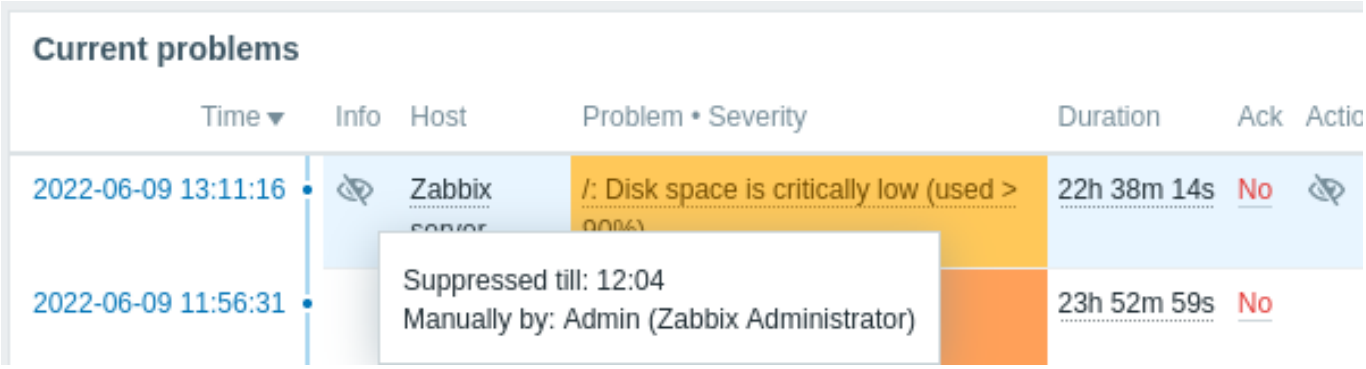
Exibição

Uma vez suprimido, o problema é marcado por um ícone de supressão piscando na coluna de * Informações* , antes de ser ocultado.

O ícone de supressão pisca enquanto a tarefa de supressão está na lista de espera. Assim que o gerenciador de tarefas suprimir o problema, o ícone parará de piscar. Se o ícone de supressão continuar piscando por um longo período, isso pode indicar um problema no servidor, por exemplo, se o servidor estiver inativo e o gerenciador de tarefas não puder concluir a tarefa. A mesma lógica se aplica à dessupressão. No curto período após a submissão da tarefa e o servidor não tiver concluído, o ícone de dessupressão estará piscando.

Um problema suprimido pode ser tanto oculto quanto mostrado, dependendo das configurações do filtro/widget de problema.

Quando mostrado na lista de problemas, um problema suprimido é marcado pelo ícone de supressão e os detalhes da supressão são mostrados ao passar o mouse sobre ele:



Detalhes de supressão também são exibidos em um pop-up ao posicionar o mouse sobre o ícone de supressão na coluna *Ações*.

14. Configuração exportação/importação

Visão geral A funcionalidade de exportação/importação do Zabbix possibilita a troca de várias entidades de configuração entre um sistema Zabbix e outro.

Casos de uso típicos para essa funcionalidade:

- compartilhar templates ou mapas de rede - os usuários do Zabbix podem compartilhar seus parâmetros de configuração
- fazer upload de um template para o Zabbix Community templates. Assim, outros podem baixar o template e importar o arquivo para o Zabbix.
- integrar com ferramentas de terceiros - os formatos universais YAML, XML e JSON tornam possível a integração e a importação/exportação de dados com ferramentas e aplicativos de terceiros

O que pode ser exportado/importado

Os objetos que podem ser exportados/importados são:

- Grupos de hosts (*apenas via Zabbix API*)
- Grupos de templates (*apenas via Zabbix API*)
- Templates
- Hosts
- Mapas de rede
- Tipos de mídia
- Imagens

Formato de exportação

Os dados podem ser exportados usando o frontend da web Zabbix ou Zabbix API. Os formatos de exportação suportados são YAML, XML e JSON.

Detalhes sobre a exportação

- Todos os elementos suportados são exportados em um único arquivo.
- Entidades de host e template (itens, triggers, gráficos, regras de descoberta) que são herdadas de templates vinculados não são exportadas. Quaisquer alterações feitas nessas entidades no nível do host (como alteração do intervalo do item, modificação da expressão regular ou adição de protótipos à regra de descoberta de baixo nível) serão perdidas ao exportar; ao importar, todas as entidades dos templates vinculados são recriadas conforme no template original vinculado.
- Entidades criadas por descoberta de baixo nível e quaisquer entidades dependentes delas não são exportadas. Por exemplo, um trigger criado para um item gerado por uma regra LLD não será exportado.
- Quando o host/template exportado contém entidades que suportam timeouts, os valores de timeout serão exportados se essas entidades tiverem seus próprios timeouts configurados.

Detalhes sobre a importação

- A importação é interrompida no primeiro erro.
- Ao atualizar imagens existentes durante a importação de imagens, o campo "imagetype" é ignorado, ou seja, não é possível alterar o tipo de imagem via importação.
- Ao importar hosts/templates usando a opção "Excluir ausentes", macros de host/template que não estiverem presentes no arquivo de importação serão excluídas do host/template após a importação.
- Tags vazias para items, triggers, gráficos, discoveryRules, itemPrototypes, triggerPrototypes, graphPrototypes são irrelevantes, ou seja, é o mesmo que se estivessem ausentes.
- Se as entidades do host/template importado tiverem seus próprios timeouts configurados, eles serão aplicados; caso contrário, os timeouts do proxy/global serão aplicados.
- A importação suporta YAML, XML e JSON, o arquivo de importação deve ter uma extensão de arquivo correta: .yaml e .yml para YAML, .xml para XML e .json para JSON. Veja informações de compatibilidade sobre as versões de XML suportadas.
- A importação suporta arquivos de configuração apenas na codificação UTF-8 (com ou sem BOM); outras codificações (UTF16LE, UTF16BE, UTF32LE, UTF32BE, etc.) resultarão em erro de conversão de importação.

Formato base YAML O formato de exportação YAML contém os seguintes nós:

- Nó raiz para exportação YAML do Zabbix
- Versão da exportação

```
zabbix_export:
  version: '7.0'
```

Outros nós dependem dos objetos exportados.

Formato XML O formato de exportação XML contém as seguintes tags:

- Cabeçalho padrão para documentos XML
- Tag raiz para exportação XML do Zabbix
- Versão da exportação

```
<?xml version="1.0" encoding="UTF-8"?>
<zabbix_export>
    <version>7.0</version>
</zabbix_export>
```

Outras tags dependem dos objetos exportados.

Formato JSON O formato de exportação JSON contém os seguintes objetos:

- Objeto raiz para exportação JSON do Zabbix
- Versão da exportação

```
{
    "zabbix_export": {
        "version": "7.0"
    }
}
```

Outros objetos dependem dos objetos exportados.

1 Grupos de templates

Visão geral

No frontend, os grupos de templates podem ser exportados apenas com a exportação do template. Quando um template é exportado, todos os grupos aos quais ele pertence são exportados automaticamente junto com ele.

A API permite exportar grupos de templates independentemente dos templates.

Formato de exportação

```
template_groups:
  - uuid: 36bff6c29af64692839d077febfc7079
    name: 'Network devices'
```

Elementos exportados

Elemento	Tipo	Descrição
uuid	string	Identificador exclusivo para este grupo de templates.
name	string	Nome do grupo.

2 Grupos de hosts

Visão geral

No frontend, os grupos de hosts podem ser exportados apenas com a exportação do host. Quando um host é exportado, todos os grupos aos quais ele pertence são exportados automaticamente junto com ele.

A API permite exportar grupos de hosts independentemente dos hosts.

Formato de exportação

```
host_groups:
  - uuid: 6f6799aa69e844b4b3918f779f2abf08
    name: 'Zabbix servers'
```

Elementos exportados

Elemento	Tipo	Descrição
uuid	string	Identificador exclusivo para este grupo de hosts.
name	string	Nome do grupo.

3 Templates

Visão geral

Os templates são exportados com muitos objetos relacionados e relações de objetos.

A exportação de template contém:

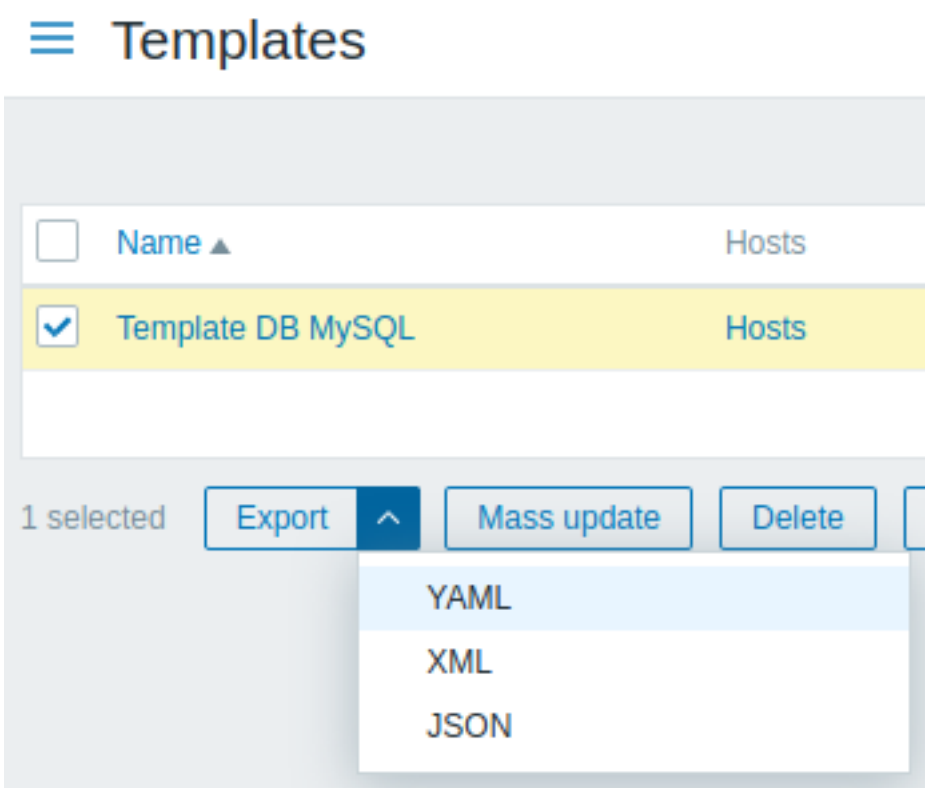
- Grupos de templates vinculados
- Grupos de hosts vinculados (se usados na configuração de prototipagem de host)
- Dados do template
- Vinculação a outros templates
- Vinculação a grupos de templates
- Items vinculados diretamente
- Triggers vinculados diretamente
- Gráficos vinculados diretamente
- Dashboards vinculados diretamente
- Regras de descoberta vinculadas diretamente com todos os protótipos
- Cenários web vinculados diretamente

- Mapas de valores

Exportando

Para exportar templates, faça o seguinte:

1. Vá para *Coleta de dados* → *Templates*.
2. Marque as caixas de seleção dos templates a serem exportados.
3. Clique em *Exportar* abaixo da lista.



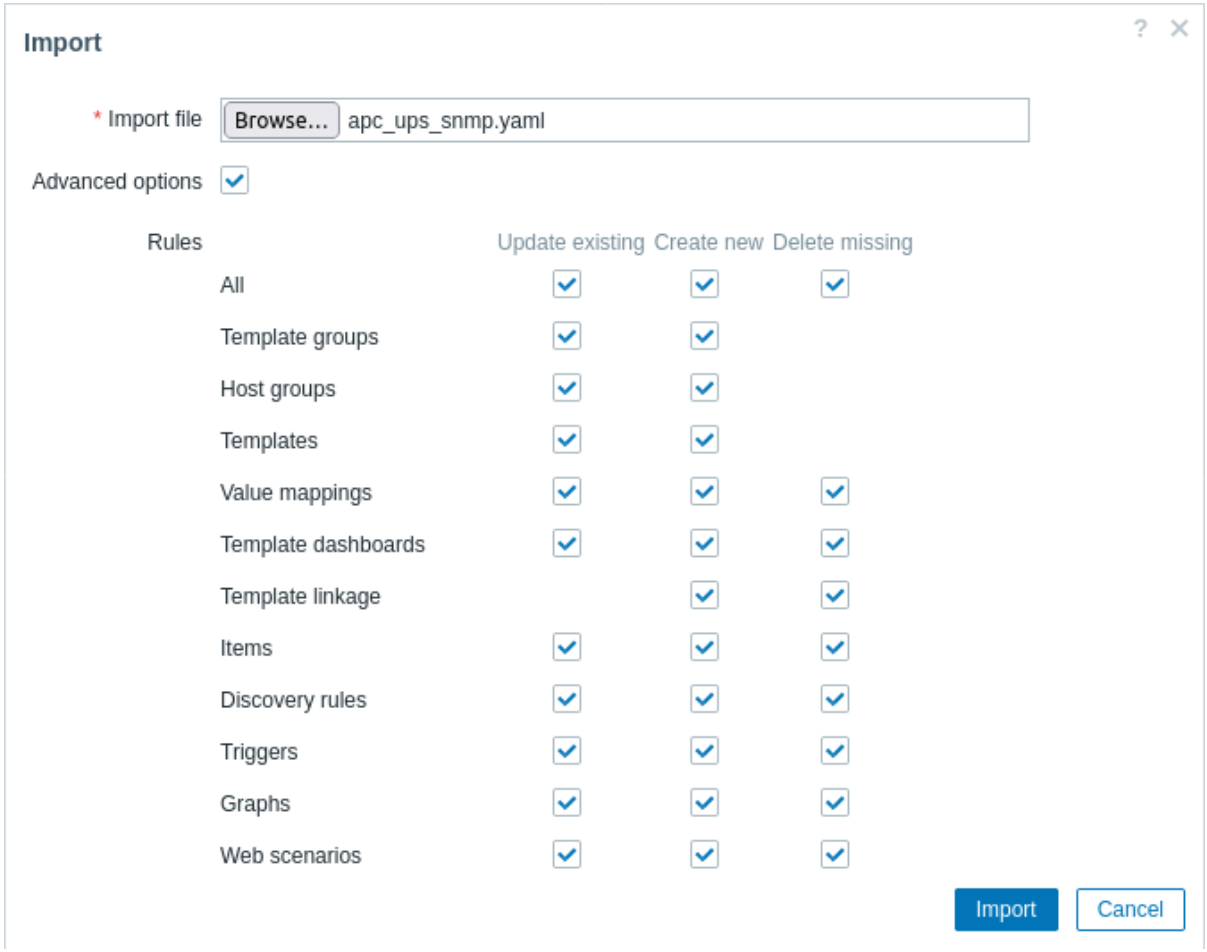
Dependendo do formato selecionado, os templates são exportados para um arquivo local com um nome padrão:

- `zabbix_export_templates.yaml` - na exportação YAML (opção padrão para exportação);
- `zabbix_export_templates.xml` - na exportação XML;
- `zabbix_export_templates.json` - na exportação JSON.

Importando

Para importar templates, faça o seguinte:

1. Vá para *Coleta de dados* → *Templates*.
2. Clique em *Importar* no canto superior direito.
3. Selecione o arquivo de importação.
4. Clique em *Importar* no canto inferior direito do formulário de configuração.



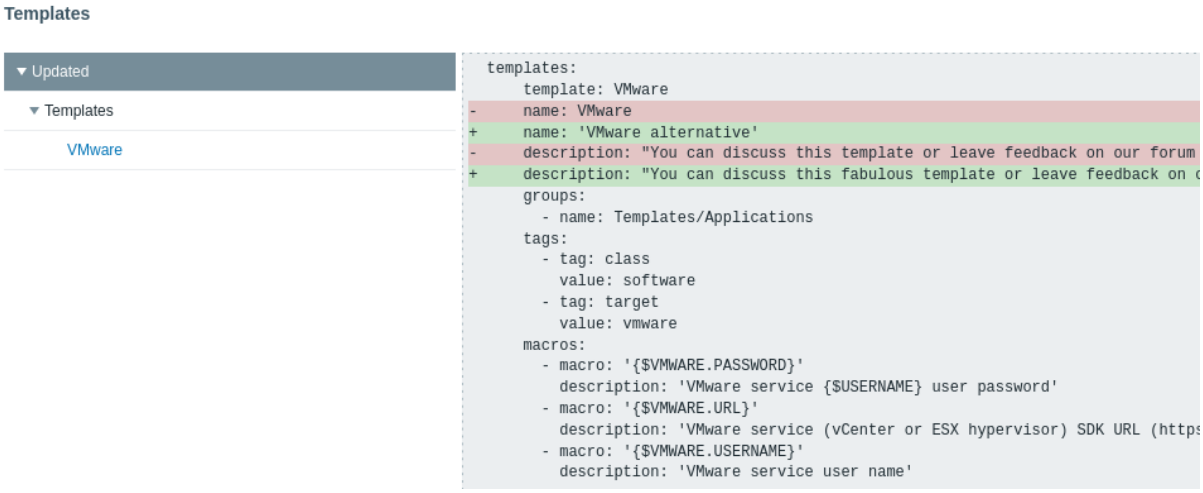
Se você marcar a caixa de seleção *Opções avançadas*, uma lista detalhada de todos os elementos importáveis será exibida - marque ou desmarque cada regra de importação conforme necessário.

Se você clicar na caixa de seleção na linha *Todos*, todos os elementos abaixo dela serão marcados/desmarcados.

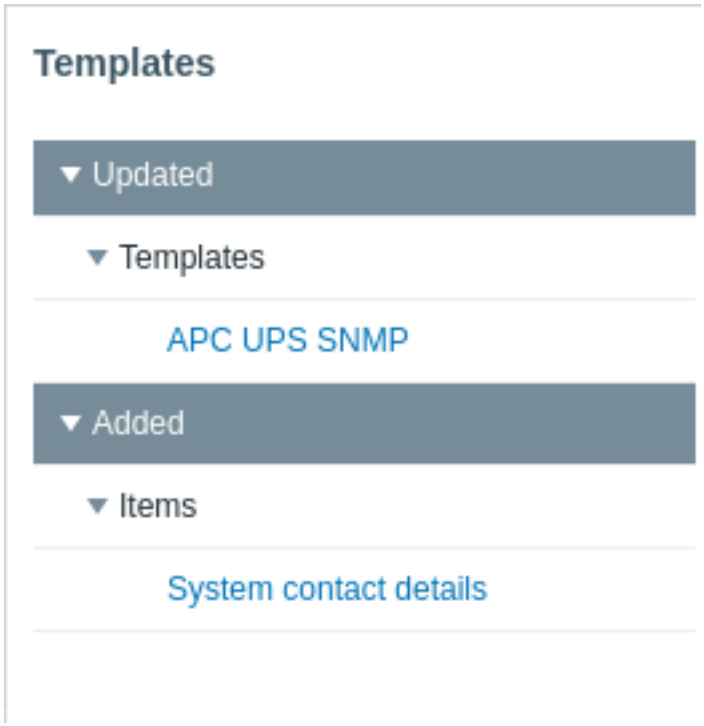
Regras de importação:

Regra	Descrição
Atualizar existentes	Os elementos existentes serão atualizados usando os dados do arquivo de importação. Caso contrário, eles não serão atualizados.
Criar novos	Novos elementos serão criados usando os dados do arquivo de importação. Caso contrário, eles não serão criados.
Excluir ausentes	Os elementos existentes que não estiverem presentes no arquivo de importação serão removidos. Caso contrário, eles não serão removidos. Se *Excluir ausentes* estiver marcado para *Vinculação de template*, a vinculação de template atual que não estiver presente no arquivo de importação será desvinculada. As entidades (items, triggers, gráficos, etc.) herdadas dos templates desvinculados não serão removidas (a menos que a opção *Excluir ausentes* também seja selecionada para cada entidade).

Na próxima tela, você poderá visualizar o conteúdo de um template sendo importado. Se este for um novo template, todos os elementos serão listados em verde. Se estiver atualizando um template existente, os novos elementos do template serão destacados em verde; os elementos removidos do template serão destacados em vermelho; os elementos que não foram alterados serão listados em um fundo cinza.



O menu à esquerda pode ser usado para navegar pela lista de alterações. A seção *Atualizado* destaca todas as alterações feitas nos elementos existentes do template. A seção *Adicionado* lista os novos elementos do template. Os elementos em cada seção são agrupados por tipo de elemento; clique na seta cinza para expandir ou recolher o grupo de elementos.



Revise as alterações do template e clique em *Importar* para realizar a importação do template. Uma mensagem de sucesso ou falha da importação será exibida no frontend.

Formato de exportação

Formato de exportação em YAML:

```
zabbix_export:
  version: '7.0'
  template_groups:
    - uuid: a571c0d144b14fd4a87a9d9b2aa9fcd6
      name: Templates/Applications
  host_groups:
    - uuid: a571c0d144b14fd4a87a9d9b2aa9fcd6
      name: Applications
  templates:
    - uuid: 56079badd056419383cc26e6a4fcc7e0
      template: VMware
      name: VMware
      description: |
        Você pode discutir este template ou deixar seu feedback em nosso fórum https://www.zabbix.com/foru
```

```
  Versão da ferramenta de template utilizada: 0.41
groups:
  - name: Templates/Applications
items:
  - uuid: 5ce209f4d94f460488a74a92a52d92b1
    name: 'VMware: Event log'
    type: SIMPLE
    key: 'vmware.eventlog[{$VMWARE.URL},skip]'
    history: 7d
    trends: '0'
    value_type: LOG
    username: '{$VMWARE.USERNAME}'
    password: '{$VMWARE.PASSWORD}'
    description: 'Coleta do log de eventos do VMware.'
    tags:
      - tag: component
        value: log
  - uuid: ee2edadb8ce943ef81d25dbbba8667a4
    name: 'VMware: Full name'
    type: SIMPLE
    key: 'vmware.fullname[{$VMWARE.URL}]'
    delay: 1h
    history: 7d
    trends: '0'
    value_type: CHAR
    username: '{$VMWARE.USERNAME}'
    password: '{$VMWARE.PASSWORD}'
    description: 'Nome completo do serviço VMware.'
    preprocessing:
      - type: DISCARD_UNCHANGED_HEARTBEAT
        parameters:
          - 1d
    tags:
      - tag: component
        value: system
  - uuid: a0ec9145f2234fbea79a28c57ebdb44d
    name: 'VMware: Version'
    type: SIMPLE
    key: 'vmware.version[{$VMWARE.URL}]'
    delay: 1h
    history: 7d
    trends: '0'
    value_type: CHAR
    username: '{$VMWARE.USERNAME}'
    password: '{$VMWARE.PASSWORD}'
    description: 'Versão do serviço VMware.'
    preprocessing:
      - type: DISCARD_UNCHANGED_HEARTBEAT
        parameters:
          - 1d
    tags:
      - tag: component
        value: system
discovery_rules:
  - uuid: 16ffc933cce74cf28a6edf306aa99782
    name: 'Discover VMware clusters'
    type: SIMPLE
    key: 'vmware.cluster.discovery[{$VMWARE.URL}]'
    delay: 1h
    username: '{$VMWARE.USERNAME}'
    password: '{$VMWARE.PASSWORD}'
    description: 'Descoberta de clusters'
```

```
    item_prototypes:
      - uuid: 46111f91dd564a459dbc1d396e2e6c76
        name: 'VMware: Status of "{#CLUSTER.NAME}" cluster'
        type: SIMPLE
        key: 'vmware.cluster.status[{$VMWARE.URL},{#CLUSTER.NAME}]'
        history: 7d
        username: '{$VMWARE.USERNAME}'
        password: '{$VMWARE.PASSWORD}'
        description: 'Status do cluster VMware.'
        valuemap:
          name: 'VMware status'
        tags:
          - tag: cluster
            value: '{#CLUSTER.NAME}'
          - tag: component
            value: cluster
  - uuid: 8fb6a45cbe074b0cb6df53758e2c6623
    name: 'Discover VMware datastores'
    type: SIMPLE
    key: 'vmware.datastore.discovery[{$VMWARE.URL}]'
    delay: 1h
    username: '{$VMWARE.USERNAME}'
    password: '{$VMWARE.PASSWORD}'
    item_prototypes:
      - uuid: 4b61838ba4c34e709b25081ae5b059b5
        name: 'VMware: Average read latency of the datastore {#DATASTORE}'
        type: SIMPLE
        key: 'vmware.datastore.read[{$VMWARE.URL},{#DATASTORE},latency]'
        history: 7d
        username: '{$VMWARE.USERNAME}'
        password: '{$VMWARE.PASSWORD}'
        description: 'Tempo de leitura de uma operação no datastore (milissegundos).'
        tags:
          - tag: component
            value: datastore
          - tag: datastore
            value: '{#DATASTORE}'
      - uuid: 5355c401dc244bc588ccd18767577c93
        name: 'VMware: Free space on datastore {#DATASTORE} (percentage)'
        type: SIMPLE
        key: 'vmware.datastore.size[{$VMWARE.URL},{#DATASTORE},pfree]'
        delay: 5m
        history: 7d
        value_type: FLOAT
        units: '%'
        username: '{$VMWARE.USERNAME}'
        password: '{$VMWARE.PASSWORD}'
        description: 'Espaço do datastore VMware em porcentagem do total.'
        tags:
          - tag: component
            value: datastore
          - tag: datastore
            value: '{#DATASTORE}'
      - uuid: 84f13c4fde2d4a17baaf0c8c1eb4f2c0
        name: 'VMware: Total size of datastore {#DATASTORE}'
        type: SIMPLE
        key: 'vmware.datastore.size[{$VMWARE.URL},{#DATASTORE}]'
        delay: 5m
        history: 7d
        units: B
        username: '{$VMWARE.USERNAME}'
        password: '{$VMWARE.PASSWORD}'
```

```
        description: 'Espaço do datastore VMware em bytes.'
        tags:
          - tag: component
            value: datastore
          - tag: datastore
            value: '{#DATASTORE}'
      - uuid: 540cd0fbc56c4b8ea19f2ff5839ce00d
        name: 'VMware: Average write latency of the datastore {#DATASTORE}'
        type: SIMPLE
        key: 'vmware.datastore.write[{$VMWARE.URL},{#DATASTORE},latency]'
        history: 7d
        username: '{$VMWARE.USERNAME}'
        password: '{$VMWARE.PASSWORD}'
        description: 'Tempo de escrita de uma operação no datastore (milissegundos).'
        tags:
          - tag: component
            value: datastore
          - tag: datastore
            value: '{#DATASTORE}'
    - uuid: a5bc075e89f248e7b411d8f960897a08
      name: 'Discover VMware hypervisors'
      type: SIMPLE
      key: 'vmware.hv.discovery[{$VMWARE.URL}]'
      delay: 1h
      username: '{$VMWARE.USERNAME}'
      password: '{$VMWARE.PASSWORD}'
      description: 'Descoberta de hipervisores.'
      host_prototypes:
        - uuid: 051a1469d4d045cbbf818fcc843a352e
          host: '{#HV.UUID}'
          name: '{#HV.NAME}'
          group_links:
            - group:
                name: Applications
          group_prototypes:
            - name: '{#CLUSTER.NAME}'
            - name: '{#DATACENTER.NAME}'
          templates:
            - name: 'VMware Hypervisor'
          macros:
            - macro: '{$VMWARE.HV.UUID}'
              value: '{#HV.UUID}'
              description: 'UUID do hipervisor.'
          custom_interfaces: 'YES'
          interfaces:
            - ip: '{#HV.IP}'
    - uuid: 9fd559f4e88c4677a1b874634dd686f5
      name: 'Discover VMware VMs'
      type: SIMPLE
      key: 'vmware.vm.discovery[{$VMWARE.URL}]'
      delay: 1h
      username: '{$VMWARE.USERNAME}'
      password: '{$VMWARE.PASSWORD}'
      description: 'Descoberta de máquinas virtuais convidadas.'
      host_prototypes:
        - uuid: 23b9ae9d6f33414880db1cb107115810
          host: '{#VM.UUID}'
          name: '{#VM.NAME}'
          group_links:
            - group:
                name: Applications
          group_prototypes:
```

```
              - name: '{#CLUSTER.NAME} (vm)'
              - name: '{#DATACENTER.NAME}/{#VM.FOLDER} (vm)'
              - name: '{#HV.NAME}'
            templates:
              - name: 'VMware Guest'
            macros:
              - macro: '{$VMWARE.VM.UUID}'
                value: '{#VM.UUID}'
                description: 'UUID da máquina virtual convidada.'
            custom_interfaces: 'YES'
            interfaces:
              - ip: '{#VM.IP}'
    tags:
      - tag: class
        value: software
      - tag: target
        value: vmware
    macros:
      - macro: '{$VMWARE.PASSWORD}'
        description: 'Senha do usuário {$USERNAME} do serviço VMware'
      - macro: '{$VMWARE.URL}'
        description: 'URL do SDK do serviço VMware (vCenter ou hipervisor ESX) (https://servername/sdk)'
      - macro: '{$VMWARE.USERNAME}'
        description: 'Nome do usuário do serviço VMware'
    valuemaps:
      - uuid: 3c59c22905054d42ac4ee8b72fe5f270
        name: 'VMware status'
        mappings:
          - value: '0'
            newvalue: gray
          - value: '1'
            newvalue: green
          - value: '2'
            newvalue: yellow
          - value: '3'
            newvalue: red
```

Elementos exportados Os elementos exportados são explicados nas tabelas abaixo.

Elemento	Tipo	Descrição
template_groups		(obrigatório) Elemento raiz para grupos de templates.
uuid	string	(obrigatório) Identificador exclusivo para este grupo de templates.
name	string	(obrigatório) Nome do grupo de templates.
host_groups		(obrigatório) Elemento raiz para grupos de hosts que são usados por protótipos de host.
uuid	string	(obrigatório) Identificador exclusivo para este grupo de hosts.
name	string	(obrigatório) Nome do grupo de hosts.
templates		Elemento raiz para templates.

Templates

Elemento	Tipo	Descrição
uuid	string	(obrigatório) Identificador exclusivo para este template.
template	string	(obrigatório) Nome exclusivo do template.
name	string	Nome visível do template.
description	text	Descrição do template.
vendor		Elemento raiz para o fornecedor do template (presente se o template exportado contiver dados do fornecedor).
name	string	(obrigatório) Nome do fornecedor do template.

Elemento	Tipo	Descrição
version	string	(obrigatório) Versão do template. Para templates prontos para uso, a versão é exibida da seguinte forma: versão principal do Zabbix, delimitador (”-”), número de revisão (aumentado a cada nova versão do template e redefinido a cada versão principal do Zabbix). Por exemplo, 6.4-0, 6.4-3, 7.0-0, 7.0-3.
templates		Elemento raiz para templates vinculados.
name	string	(obrigatório) Nome do template.
groups		Elemento raiz para grupos de templates.
name	string	(obrigatório) Nome do grupo de templates.
items		Elemento raiz para itens do template.
discovery_rules		Elemento raiz para regras de descoberta de baixo nível do template.
httptests		Elemento raiz para cenários web do template.
tags		Elemento raiz para tags do template.
tag	string	(obrigatório) Nome da tag.
value	string	Valor da tag.
macros		Elemento raiz para macros de usuário do template.
macro	string	(obrigatório) Nome da macro de usuário.
type	string	Tipo de macro de usuário. Valores possíveis:[1] TEXT (0, padrão), SECRET_TEXT (1), VAULT (2). Veja também: Objeto macro de usuário (`type`).
value	string	Valor da macro de usuário.
description	string	Descrição da macro de usuário.
dashboards		Elemento raiz para dashboards do template.
valuemaps		Elemento raiz para mapas de valor do template.

Itens do template

Elemento	Tipo	Descrição
uuid	string	(obrigatório) Identificador único para este item.
name	string	(obrigatório) Nome do item.
type	string	Tipo do item. Valores possíveis:[1] ZABBIX_PASSIVE (0, padrão), TRAP (2), SIMPLE (3), INTERNAL (5), ZABBIX_ACTIVE (7), EXTERNAL (10), ODBC (11), IPMI (12), SSH (13), TELNET (14), CALCULATED (15), JMX (16), SNMP_TRAP (17), DEPENDENT (18), HTTP_AGENT (19), SNMP_AGENT (20), ITEM_TYPE_SCRIPT (21), ITEM_TYPE_BROWSER (22).
snmp_oid	string	(obrigatório para itens SNMP_AGENT) SNMP object ID.
key	string	(obrigatório) Chave do item.
delay	string	Intervalo de atualização do item. Padrão: 1m. O valor será sempre 0 para itens TRAP.
history	string	Período de tempo (usando sufixo de tempo, macro de usuário ou macro LLD) de quanto tempo os dados de histórico devem ser armazenados. Padrão: 31d.
trends	string	Período de tempo (usando sufixo de tempo, macro de usuário ou macro LLD) de quanto tempo os dados de tendências devem ser armazenados. Padrão: 365d.
status	string	Status do item. Valores possíveis:[1] ENABLED (0, padrão), DISABLED (1).
value_type	string	Tipo de valor recebido. Valores possíveis:[1] FLOAT (0), CHAR (1), LOG (2), UNSIGNED (3, padrão), TEXT (4), BINARY (5).
allowed_hosts	string	Lista de endereços IP separados por vírgula dos hosts permitidos a enviar dados para o item. Suportado para itens TRAP e HTTP_AGENT.
units	string	Unidades do valor recebido (bps, B, etc.).
params	text	Parâmetros adicionais dependendo do tipo do item (script executado para itens SSH e TELNET; consulta SQL para itens ODBC; fórmula para itens CALCULATED; script para itens ITEM_TYPE_SCRIPT e ITEM_TYPE_BROWSER).
ipmi_sensor	string	Sensor IPMI. Suportado para itens IPMI.

Elemento	Tipo	Descrição
authtype	string	Tipo de autenticação. Suportado para itens SSH e HTTP_AGENT. Valores possíveis para itens SSH:[1] PASSWORD (0, padrão), PUBLIC_KEY (1). Valores possíveis para itens HTTP_AGENT:[1] NONE (0, padrão), BASIC (1), NTLM (2), Kerberos (3) ou Digest (4).
username	string	(obrigatório para itens SSH e TELNET) Nome de usuário para autenticação. Suportado para itens SIMPLE, ODBC, JMX e HTTP_AGENT. Quando usado para itens JMX, `password` (veja abaixo) também deve ser especificado ou ambos os elementos devem ser deixados em branco.
password	string	(obrigatório para itens SSH e TELNET) Senha para autenticação. Suportado para itens SIMPLE, ODBC, JMX e HTTP_AGENT. Quando usado para itens JMX, `username` (veja acima) também deve ser especificado ou ambos os elementos devem ser deixados em branco.
publickey	string	(obrigatório para itens SSH) Nome do arquivo da chave pública.
privatekey	string	(obrigatório para itens SSH) Nome do arquivo da chave privada.
description	text	Descrição do item.
inventory_link	string	Campo de inventário do host que é preenchido pelo item. Valores possíveis:[1] NONE (0), ALIAS (4), etc. (veja Inventário do host para campos suportados).
valuemap		Elemento raiz para mapas de valor do item.
name	string	(obrigatório) Nome do mapa de valor a ser usado para o item.
logtimefmt	string	Formato da hora nas entradas de log. Suportado para itens do tipo de valor LOG.
preprocessing		Elemento raiz para pré-processamento de valor do item.
step		Elemento raiz para etapas de pré-processamento de valor do item do template.
jmx_endpoint	string	Endpoint JMX. Suportado para itens JMX.
master_item		(obrigatório para itens DEPENDENT) Elemento raiz para o item mestre do item dependente.
key	string	(obrigatório) Chave do item mestre do item dependente.
timeout	string	Timeout da requisição de coleta de dados do item. Suportado para a lista de tipos de item com suporte a timeout.
url	string	(obrigatório para itens HTTP_AGENT) String da URL.
query_fields		Elemento raiz para parâmetros de consulta. Suportado para itens HTTP_AGENT.
name	string	(obrigatório para itens HTTP_AGENT) Nome do parâmetro de consulta.
value	string	Valor do parâmetro de consulta. Suportado para itens HTTP_AGENT.
parameters		Elemento raiz para parâmetros definidos pelo usuário. Suportado para itens ITEM_TYPE_SCRIPT e ITEM_TYPE_BROWSER.
name	string	(obrigatório para itens ITEM_TYPE_SCRIPT e ITEM_TYPE_BROWSER) Nome do parâmetro definido pelo usuário.
value	string	Valor do parâmetro definido pelo usuário. Suportado para itens ITEM_TYPE_SCRIPT e ITEM_TYPE_BROWSER.
posts	string	Dados do corpo da requisição HTTP(S). Suportado para itens HTTP_AGENT.
status_codes	string	Intervalos de códigos de status HTTP requeridos, separados por vírgula. Suportado para itens HTTP_AGENT.
follow_redirects	string	Seguir redirecionamentos de resposta ao coletar dados. Suportado para itens HTTP_AGENT. Valores possíveis:[1] NO (0), YES (1, padrão).
post_type	string	Tipo do corpo de dados postados. Suportado para itens HTTP_AGENT. Valores possíveis:[1] RAW (0, padrão), JSON (2), XML (3).
http_proxy	string	String de conexão do proxy HTTP(S). Suportado para itens HTTP_AGENT.
headers		Elemento raiz para cabeçalhos de requisição HTTP(S). Suportado para itens HTTP_AGENT.
name	string	(obrigatório para itens HTTP_AGENT) Nome do cabeçalho.
value	string	(obrigatório para itens HTTP_AGENT) Valor do cabeçalho.

Elemento	Tipo	Descrição
retrieve_mode	string	Qual parte da resposta deve ser armazenada. Suportado para itens HTTP_AGENT. Valores possíveis:[1] BODY (0, padrão), HEADERS (1), BOTH (2).
request_method	string	Tipo do método de requisição. Suportado para itens HTTP_AGENT. Valores possíveis:[1] GET (0, padrão), POST (1), PUT (2), HEAD (3).
output_format	string	Como processar a resposta. Suportado para itens HTTP_AGENT. Valores possíveis:[1] RAW (0, padrão), JSON (1).
allow_traps	string	Permitir popular valor similar ao item trapper. Suportado para itens HTTP_AGENT. Valores possíveis:[1] NO (0, padrão), YES (1).
ssl_cert_file	string	Caminho do arquivo da chave pública SSL. Suportado para itens HTTP_AGENT.
ssl_key_file	string	Caminho do arquivo da chave privada SSL. Suportado para itens HTTP_AGENT.
ssl_key_password	string	Senha para o arquivo da chave SSL. Suportado para itens HTTP_AGENT.
verify_peer	string	Se deve validar que o certificado do host é autêntico. Suportado para itens HTTP_AGENT. Valores possíveis:[1] NO (0, padrão), YES (1).
verify_host	string	Se deve validar que o nome do host para a conexão corresponde ao do certificado do host. Suportado para itens HTTP_AGENT. Valores possíveis:[1] NO (0, padrão), YES (1).
tags		Elemento raiz para tags do item.
tag	string	(obrigatório) Nome da tag.
value	string	Valor da tag.
triggers		Elemento raiz para triggers de itens do template.

Note:
Veja também: Objeto item (consulte a propriedade relevante com o nome correspondente).

Etapas de pré-processamento do valor do item do template

Elemento	Tipo	Descrição
type	string	(obrigatório) O tipo da etapa de pré-processamento do valor do item. Valores possíveis:[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, calculado: valor recebido - valor anterior), CHANGE_PER_SECOND (10, calculado: (valor recebido - valor anterior)/(hora atual - hora da última verificação)), 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		(obrigatório) Elemento raiz para os parâmetros da etapa de pré-processamento do valor do item.
parameter	string	Parâmetro individual da etapa de pré-processamento do valor do item.
error_handler	string	Tipo de ação usada em caso de falha na etapa de pré-processamento. Valores possíveis:[1] ORIGINAL_ERROR (0, padrão), DISCARD_VALUE (1), CUSTOM_VALUE (2), CUSTOM_ERROR (3).
error_handler_params	string	Parâmetros do manipulador de erros.

Note:
Veja também: Objeto de pré-processamento do item (consulte a propriedade relevante com o mesmo nome).

Triggers de item de template

Elemento	Tipo	Descrição
uuid	string	(obrigatório) Identificador exclusivo para este trigger.
expression	string	(obrigatório) Expressão do trigger.
recovery_mode	string	Base para gerar eventos OK. Valores possíveis:[1] EXPRESSION (0, padrão), RECOVERY_EXPRESSION (1), NONE (2).
recovery_expression	string	Expressão de recuperação do trigger.
correlation_mode	string	Modo de correlação (sem correlação de evento ou correlação de evento por tag). Valores possíveis:[1] DISABLED (0, padrão), TAG_VALUE (1).
correlation_tag	string	O nome da tag a ser usada para correlação de eventos.
name	string	(obrigatório) Nome do trigger.
event_name	string	Nome do evento.
opdata	string	Dados operacionais.
url_name	string	Rótulo para a URL associada ao trigger.
url	string	URL associada ao trigger.
status	string	Status do trigger. Valores possíveis:[1] ENABLED (0, padrão), DISABLED (1).
priority	string	Severidade do trigger. Valores possíveis:[1] NOT_CLASSIFIED (0, padrão), INFO (1), WARNING (2), AVERAGE (3), HIGH (4), DISASTER (5).
description	text	Descrição do trigger.
type	string	Tipo de geração de evento (evento de problema único ou múltiplos eventos de problema). Valores possíveis:[1] SINGLE (0, padrão), MULTIPLE (1).
manual_close	string	Fechamento manual de eventos de problema. Valores possíveis:[1] NO (0, padrão), YES (1).
dependencies		Elemento raiz para dependências.
name	string	(obrigatório) Nome do trigger de dependência.
expression	string	(obrigatório) Expressão do trigger de dependência.
recovery_expression	string	Expressão de recuperação do trigger de dependência.
tags		Elemento raiz para tags de trigger.
tag	string	(obrigatório) Nome da tag.
value	string	Valor da tag.

Note:
Veja também: Objeto trigger (consulte a propriedade relevante com o nome correspondente).

Regras de descoberta de baixo nível do template

Attention:
A maioria dos elementos da regra de descoberta de baixo nível do template são os mesmos que para itens do template. A tabela abaixo descreve os elementos que diferem dos itens do template.

Elemento	Tipo	Descrição
type	string	Tipo de item. Valores possíveis:[1] ZABBIX_PASSIVE (0, padrão), 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	(obrigatório) A chave da regra de descoberta de baixo nível.
filter		Elemento raiz para filtros de regra de descoberta de baixo nível do template.
lifetime	string	Período de tempo (usando segundos, sufixo de tempo ou macro de usuário), após o qual os recursos não descobertos serão excluídos. Padrão: 7d.
lifetime_type	string	Cenário para excluir recursos LLD perdidos. Valores possíveis: DELETE_NEVER, DELETE_IMMEDIATELY, DELETE_AFTER.
enabled_lifetime	string	Período de tempo (usando segundos, sufixo de tempo ou macro de usuário), após o qual os recursos não descobertos serão desabilitados.

Elemento	Tipo	Descrição
enabled_lifetime_type	string	Cenário para desabilitar recursos LLD perdidos. Valores possíveis: DISABLE_NEVER, DISABLE_IMMEDIATELY, DISABLE_AFTER.
item_prototypes		Elemento raiz para elementos de protótipo de item do template, que são os mesmos que para itens do template.
trigger_prototypes		Elemento raiz para elementos de protótipo de trigger do template, que são os mesmos que para triggers de item do template.
graph_prototypes		Elemento raiz para elementos de protótipo de gráfico do template, que são os mesmos que para gráficos do host.
host_prototypes		Elemento raiz para elementos de protótipo de host do template, que são os mesmos que para hosts.
master_item	string	(obrigatório para regras DEPENDENT) Elemento raiz para o item mestre da regra dependente.
lld_macro_paths		Elemento raiz para caminhos de macro de regra de descoberta de baixo nível.
lld_macro	string	(obrigatório) Nome da macro da regra de descoberta de baixo nível.
path	string	(obrigatório) Seletor para o valor, que será atribuído à macro correspondente.
preprocessing		Elemento raiz para pré-processamento de valor da regra de descoberta de baixo nível.
step		Elemento raiz para elementos de etapa de pré-processamento de valor da regra de descoberta de baixo nível, que são os mesmos que para etapas de pré-processamento de valor de item do template, exceto com menos valores possíveis. Veja também: Objeto de pré-processamento de regra LLD.
overrides		Elemento raiz para regras de sobrescrita de regra de descoberta de baixo nível.
name	string	(obrigatório) Nome único da sobrescrita.
step	string	(obrigatório) Número de ordem único da sobrescrita.
stop	string	Parar o processamento das próximas sobrescritas se corresponder.
filter		Elemento raiz para elementos de filtro de regra de sobrescrita de regra de descoberta de baixo nível do template, que são os mesmos que para filtros de regra de descoberta de baixo nível do template.
operations		Elemento raiz para operações de sobrescrita de regra de descoberta de baixo nível do template.

Note:
Veja também: Objeto de regra LLD (consulte a propriedade relevante com o nome correspondente).

Filtros de regras de descoberta de baixo nível de template

Elemento	Tipo	Descrição
evaltype	string	Substitui o método de avaliação da condição do filtro. Valores possíveis:[1] AND_OR (0, padrão), AND (1), OR (2), FORMULA (3).
formula	string	Fórmula de cálculo personalizada para as condições do filtro.
conditions		Elemento raiz para as condições do filtro.
macro	string	(obrigatório) O nome da macro de descoberta de baixo nível para realizar a verificação.
value	string	Valor para comparar.
operator	string	Operador de condição. Valores possíveis:[1] MATCHES_REGEX (8, padrão), NOT_MATCHES_REGEX (9).
formulaid	string	(obrigatório) ID único arbitrário que é usado para referenciar uma condição da expressão personalizada. Só pode conter letras maiúsculas. O ID deve ser definido pelo usuário ao modificar as condições do filtro, mas será gerado novamente ao solicitá-las posteriormente.

Note:
Veja também: Objeto de filtro de regra LLD (consulte a propriedade relevante com o nome correspondente).

Operações de substituição de regras de descoberta de baixo nível de template

Elemento	Tipo	Descrição
operationobject	string	Objeto ao qual a operação é aplicada. Valores possíveis:[1] ITEM_PROTOTYPE (0), TRIGGER_PROTOTYPE (1), GRAPH_PROTOTYPE (2), HOST_PROTOTYPE (3).
operator	string	Operador da condição de substituição. Valores possíveis:[1] EQUAL (1), NOT_EQUAL (2), LIKE (3), NOT_LIKE (4), REGEXP (5), NOT_REGEXP (6).
value	string	Uma expressão regular ou uma string para o operador de condição de substituição.
status	string	Status do objeto após a operação de substituição.
discover	string	Se o objeto é adicionado como resultado da descoberta.
delay	string	Intervalo de atualização definido para o protótipo de item após a operação de substituição.
history	string	Período de armazenamento do histórico definido para o protótipo de item após a operação de substituição.
trends	string	Período de armazenamento de tendências definido para o protótipo de item após a operação de substituição.
severity	string	Severidade do protótipo de trigger definida após a operação de substituição.
tags		Elemento raiz para as tags definidas para o objeto após a operação de substituição.
tag	string	(obrigatório) Nome da tag.
value	string	Valor da tag.
templates		Elemento raiz para os templates vinculados ao protótipo de host após a operação de substituição.
name	string	(obrigatório) Nome do template.
inventory_mode	string	Modo de inventário do protótipo de host definido após a operação de substituição.

Note:
Veja também: Objeto de operação de substituição de regra LLD (consulte a propriedade relevante com o nome correspondente).

Template de cenários web

Elemento	Tipo	Descrição
uuid	string	(obrigatório) Identificador exclusivo para este cenário web.
name	string	(obrigatório) Nome do cenário web.
delay	string	Frequência (usando segundos, sufixo de tempo ou macro de usuário) de execução do cenário web. Padrão: 1m.
attempts	integer	O número de tentativas para executar as etapas do cenário web. Valores possíveis:[1] 1-10 (padrão: 1).
agent	string	Agent do cliente. O Zabbix irá simular o navegador selecionado. Isto é útil quando um site retorna conteúdo diferente para navegadores diferentes. Padrão: Zabbix.
http_proxy	string	Proxy que será usado pelo cenário web fornecido como: `http://[username[:password]@]proxy.example.com[:port]`
variables		Elemento raiz para variáveis de cenário web que podem ser usadas nas etapas do cenário.
name	string	(obrigatório) Nome da variável.
value	text	(obrigatório) Valor da variável.
headers		Elemento raiz para cabeçalhos HTTP a serem enviados ao executar uma solicitação. Os cabeçalhos devem ser listados usando a mesma sintaxe que apareceriam no protocolo HTTP.
name	string	(obrigatório) Nome do cabeçalho.
value	text	(obrigatório) Valor do cabeçalho.
status	string	Status do cenário web. Valores possíveis:[1] ENABLED (0, padrão), DISABLED (1).
authentication	string	Método de autenticação. Valores possíveis:[1] NONE (0, padrão), BASIC (1), NTLM (2), Kerberos (3) ou Digest (4).
http_user	string	Nome de usuário usado para autenticação BASIC (HTTP), NTLM, Kerberos ou Digest.
http_password	string	Senha usada para autenticação BASIC (HTTP), NTLM, Kerberos ou Digest.

Elemento	Tipo	Descrição
verify_peer	string	Verificar o certificado SSL do servidor web. Valores possíveis:[1] NO (0, padrão), YES (1).
verify_host	string	Verificar se o campo *Common Name* ou o campo *Subject Alternate Name* do certificado do servidor web corresponde. Valores possíveis:[1] NO (0, padrão), YES (1).
ssl_cert_file	string	Nome do arquivo de certificado SSL usado para autenticação do cliente (deve estar no formato PEM).
ssl_key_file	string	Nome do arquivo de chave privada SSL usado para autenticação do cliente (deve estar no formato PEM).
ssl_key_password	string	Senha do arquivo de chave privada SSL.
steps		(obrigatório) Elemento raiz para etapas do cenário web do template.
tags		Elemento raiz para tags de cenário web.
tag	string	(obrigatório) Nome da tag.
value	string	Valor da tag.

Note:
Veja também: Objeto de cenário web (consulte a propriedade relevante com o nome correspondente).

Etapas do cenário web do template

Elemento	Tipo	Descrição
name	string	(obrigatório) Nome da etapa do cenário web.
url	string	(obrigatório) URL para monitoramento.
query_fields		Elemento raiz para parâmetros de consulta (um array de campos HTTP a serem adicionados à URL ao executar uma solicitação).
name	string	(obrigatório) Nome do parâmetro de consulta.
value	string	Valor do parâmetro de consulta.
posts		Elemento raiz para variáveis HTTP POST (uma string (dados brutos do post) ou um array de campos HTTP (dados do campo do formulário)).
name	string	(obrigatório) Nome do campo do post.
value	string	(obrigatório) Valor do campo do post.
variables		Elemento raiz das variáveis (macros) no nível da etapa que devem ser aplicadas após esta etapa. Se o valor da variável tiver o prefixo 'regex:', então seu valor será extraído dos dados retornados por esta etapa de acordo com o padrão de expressão regular que segue o prefixo 'regex:'
name	string	(obrigatório) Nome da variável.
value	text	(obrigatório) Valor da variável.
headers		Elemento raiz para cabeçalhos HTTP a serem enviados ao executar uma solicitação.
name	string	(obrigatório) Nome do cabeçalho.
value	text	(obrigatório) Valor do cabeçalho.
follow_redirects	string	Seguir redirecionamentos HTTP. Valores possíveis:[1] NÃO (0), SIM (1, padrão).
retrieve_mode	string	Modo de obtenção da resposta HTTP. Valores possíveis:[1] BODY (0, padrão), HEADERS (1), BOTH (2).
timeout	string	Timeout (usando segundos, sufixo de tempo ou macro de usuário) da execução da etapa. Padrão: 15s.
required	string	Texto que deve estar presente na resposta (ignorado se vazio).
status_codes	string	Uma lista separada por vírgulas de códigos de status HTTP aceitos (por exemplo, 200-201,210-299; ignorado se vazio).

Note:
Veja também: Objeto de etapa do cenário web (consulte a propriedade relevante com o nome correspondente).

Dashboards de template

Elemento	Tipo	Descrição
uuid	string	(obrigatório) Identificador exclusivo para este dashboard.
name	string	(obrigatório) Nome do dashboard do template.
display_period	integer	Período de exibição das páginas do dashboard.
auto_start	string	Início automático da apresentação de slides. Valores possíveis:[1] NO (0), YES (1, padrão).
pages		Elemento raiz para páginas do dashboard do template.
name	string	Nome da página.
display_period	integer	Período de exibição da página.
sortorder	integer	Ordem de classificação da página.
widgets		Elemento raiz para widgets do dashboard do template.

Note:
Veja também: Objeto dashboard do template (consulte a propriedade relevante com o nome correspondente).

Widgets do dashboard de template

Elemento	Tipo	Descrição
type	string	(obrigatório) Tipo de widget.
name	string	Nome do widget.
x	integer	Posição horizontal a partir do lado esquerdo do dashboard de template. Valores possíveis:[1] 0-71.
y	integer	Posição vertical a partir do topo do dashboard de template. Valores possíveis:[1] 0-63.
width	integer	Largura do widget. Valores possíveis:[1] 1-72.
height	integer	Altura do widget. Valores possíveis:[1] 1-64.
hide_header	string	Ocultar o cabeçalho do widget. Valores possíveis:[1] NO (0, padrão), YES (1).
fields		Elemento raiz para os campos do widget do dashboard de template.
type	string	(obrigatório) Tipo de campo do widget. Valores possíveis:[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	(obrigatório) Nome do campo do widget.
value	mixed	(obrigatório) Valor do campo do widget, dependendo do tipo de campo.

Note:
Veja também: Objeto widget do dashboard de template (consulte a propriedade relevante com o nome correspondente).

Mapas de valores de template

Elemento	Tipo	Descrição
uuid	string	(obrigatório) Identificador exclusivo para este mapa de valores.
name	string	(obrigatório) Nome do mapa de valores.
mapping		Elemento raiz para mapeamentos.
type	string	Tipo de correspondência do mapeamento. Valores possíveis:[1] EQUAL (0, padrão), GREATER_OR_EQUAL (2), LESS_OR_EQUAL (3), IN_RANGE (4), REGEXP (5), DEFAULT (6).
value	string	Valor original.
newvalue	string	(obrigatório) Valor para o qual o valor original é mapeado.

Note:
Veja também: Objeto de mapa de valores (consulte a propriedade relevante com o nome correspondente).

Notas de rodapé [1] Os valores inteiros da API entre parênteses, por exemplo, ENABLED (0), são mencionados apenas para referência. Para obter mais informações, consulte a página do objeto da API vinculada na entrada da tabela ou no final de cada seção.

4 Hosts

Visão geral

Hosts são exportados com muitos objetos relacionados e relações de objetos.

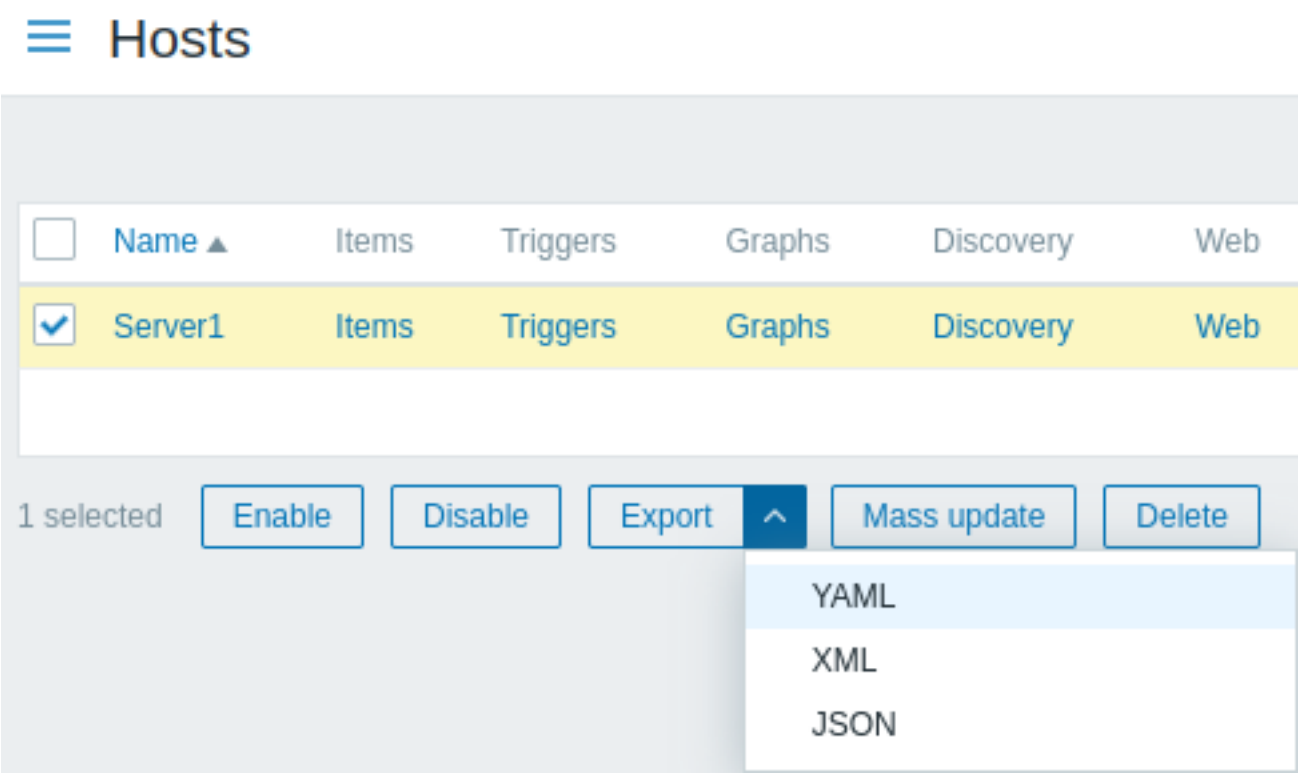
A exportação de host contém:

- Grupos de hosts vinculados
- Dados do host
- Vinculação de template
- Vinculação de grupo de host
- Interfaces do host
- Itens vinculados diretamente
- Triggers vinculados diretamente
- Regras de descoberta vinculadas diretamente com todos os protótipos
- Cenários web vinculados diretamente
- Macros do host
- Dados de inventário do host
- Mapas de valores
- Gráficos vinculados

Exportando

Para exportar hosts, faça o seguinte:

1. Vá para *Coleta de dados* → *Hosts*.
2. Marque as caixas de seleção dos hosts a serem exportados.
3. Clique em *Exportar* abaixo da lista.



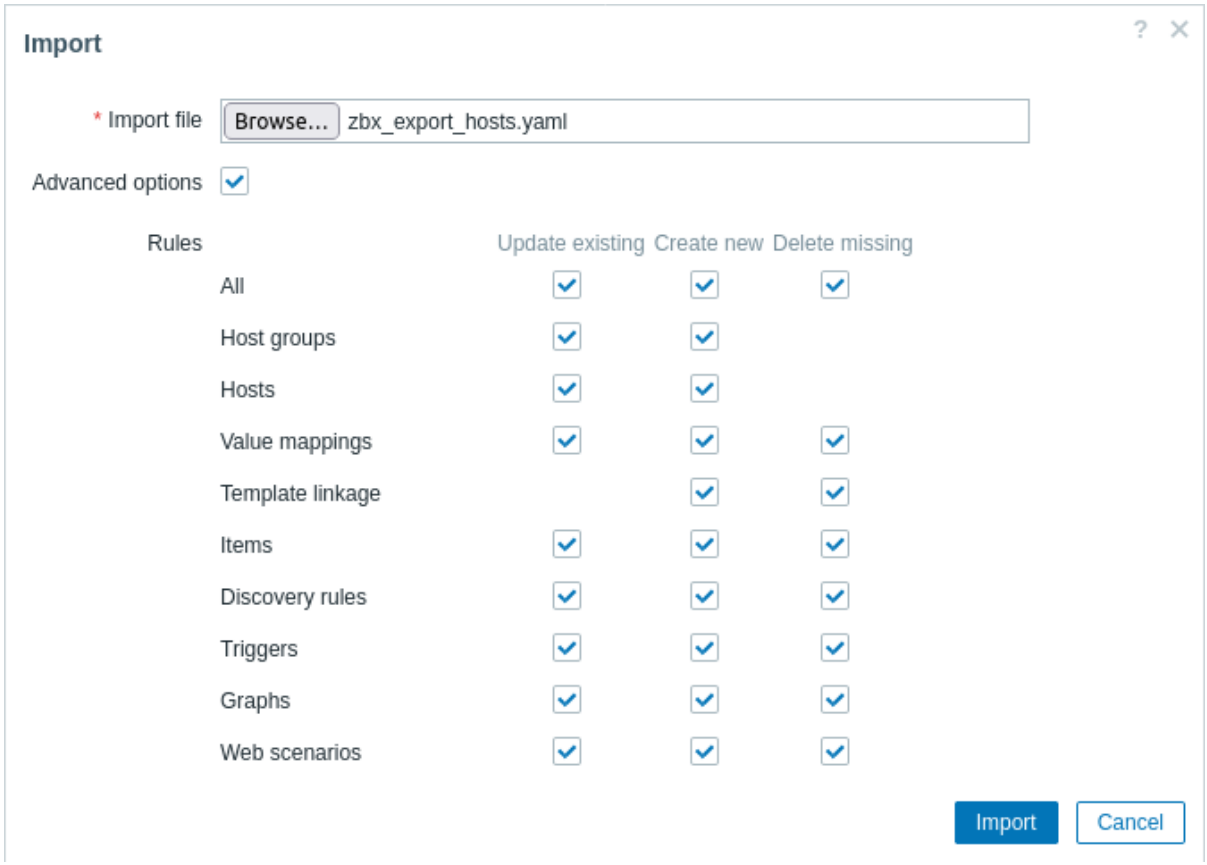
Dependendo do formato selecionado, os hosts são exportados para um arquivo local com um nome padrão:

- `zabbix_export_hosts.yaml` - na exportação YAML (opção padrão para exportação);
- `zabbix_export_hosts.xml` - na exportação XML;
- `zabbix_export_hosts.json` - na exportação JSON.

Importando

Para importar hosts, faça o seguinte:

1. Vá para *Coleta de dados* → *Hosts*.
2. Clique em *Importar* no canto superior direito.
3. Selecione o arquivo de importação.
4. Clique em *Importar* no canto inferior direito do formulário de configuração.



Se você marcar a caixa de seleção *Opções avançadas*, uma lista detalhada de todos os elementos importáveis será exibida - marque ou desmarque cada regra de importação conforme necessário.

Se você clicar na caixa de seleção na linha *Todos*, todos os elementos abaixo dela serão marcados/desmarcados.

Regras de importação:

Regra	Descrição
Atualizar existentes	Os elementos existentes serão atualizados usando os dados do arquivo de importação. Caso contrário, eles não serão atualizados.
Criar novos	Novos elementos serão criados usando os dados do arquivo de importação. Caso contrário, eles não serão criados.
Excluir ausentes	Os elementos existentes que não estiverem presentes no arquivo de importação serão removidos. Caso contrário, eles não serão removidos. Se *Excluir ausentes* estiver marcado para *Vinculação de template*, a vinculação de template atual que não estiver presente no arquivo de importação será desvinculada. Entidades (items, triggers, gráficos, etc.) herdadas dos templates desvinculados não serão removidas (a menos que a opção *Excluir ausentes* também seja selecionada para cada entidade).

Uma mensagem de sucesso ou falha da importação será exibida no frontend.

Formato de exportação

Formato de exportação em YAML:

```
zabbix_export:
  version: '7.0'
  host_groups:
```

```
  - uuid: f2481361f99448eea617b7b1d4765566
    name: 'Discovered hosts'
  - uuid: 6f6799aa69e844b4b3918f779f2abf08
    name: 'Zabbix servers'
hosts:
  - host: 'Zabbix server 1'
    name: 'Main Zabbix server'
    monitored_by: 'SERVER'
    templates:
      - name: 'Linux by Zabbix agent'
      - name: 'Zabbix server health'
    groups:
      - name: 'Discovered hosts'
      - name: 'Zabbix servers'
    interfaces:
      - ip: 192.168.1.1
        interface_ref: if1
    items:
      - name: 'Zabbix trap'
        type: TRAP
        key: trap
        delay: '0'
        history: 1w
        preprocessing:
          - type: MULTIPLIER
            parameters:
              - '8'
        tags:
          - tag: component
            value: 'raw'
        triggers:
          - expression: 'last(/Zabbix server 1/trap)=0'
            name: 'Last value is zero'
            priority: WARNING
            tags:
              - tag: scope
                value: 'availability'
    tags:
      - tag: service
        value: zabbix-server
    macros:
      - macro: '{$HOST.MACRO}'
        value: '123'
      - macro: '{$PASSWORD1}'
        type: SECRET_TEXT
    inventory:
      type: 'Zabbix server'
      name: yyyyyy-HP-Pro-3010-Small-Form-Factor-PC
      os: 'Linux yyyyyy-HP-Pro-3010-Small-Form-Factor-PC 4.4.0-165-generic #193-Ubuntu SMP Tue Sep 17 17
    inventory_mode: AUTOMATIC
graphs:
  - name: 'CPU utilization server'
    show_work_period: 'NO'
    show_triggers: 'NO'
    graph_items:
      - drawtype: FILLED_REGION
        color: FF5555
        item:
          host: 'Zabbix server 1'
          key: 'system.cpu.util[,steal]'
      - sortorder: '1'
        drawtype: FILLED_REGION
```

```
        color: 55FF55
        item:
          host: 'Zabbix server 1'
          key: 'system.cpu.util[,softirq]'
      - sortorder: '2'
        drawtype: FILLED_REGION
        color: '009999'
        item:
          host: 'Zabbix server 1'
          key: 'system.cpu.util[,interrupt]'
      - sortorder: '3'
        drawtype: FILLED_REGION
        color: '990099'
        item:
          host: 'Zabbix server 1'
          key: 'system.cpu.util[,nice]'
      - sortorder: '4'
        drawtype: FILLED_REGION
        color: '999900'
        item:
          host: 'Zabbix server 1'
          key: 'system.cpu.util[,iowait]'
      - sortorder: '5'
        drawtype: FILLED_REGION
        color: '990000'
        item:
          host: 'Zabbix server 1'
          key: 'system.cpu.util[,system]'
      - sortorder: '6'
        drawtype: FILLED_REGION
        color: '000099'
        calc_fnc: MIN
        item:
          host: 'Zabbix server 1'
          key: 'system.cpu.util[,user]'
      - sortorder: '7'
        drawtype: FILLED_REGION
        color: '009900'
        item:
          host: 'Zabbix server 1'
          key: 'system.cpu.util[,idle]'
```

Elementos exportados Os elementos exportados são explicados nas tabelas abaixo.

Elemento		Tipo	Descrição
version		string	(obrigatório) Versão do Zabbix.
host_groups			(obrigatório) Elemento raiz para grupos de hosts.
	uuid	string	(obrigatório) Identificador único para este grupo de hosts.
	name	string	(obrigatório) Nome do grupo de hosts.
hosts			Elemento raiz para hosts.

Hosts

Elemento	Tipo	Descrição
host	string	(obrigatório) Nome exclusivo do host.
name	string	Nome visível do host.
description	text	Descrição do host.
monitored_by	string	O host é monitorado por: SERVER, PROXY ou PROXY_GROUP.
proxy		Elemento raiz para proxy.
name	string	(obrigatório) Nome do proxy (se houver) que monitora o host.

Elemento	Tipo	Descrição
proxy_group		Elemento raiz para grupo de proxy.
name	string	(obrigatório) Nome do grupo de proxy (se houver) que é usado para monitorar o host.
status	string	Status do host. Valores possíveis:[1] ENABLED (0, padrão), DISABLED (1).
ipmi_authtype	string	Tipo de autenticação da sessão IPMI. Valores possíveis:[1] DEFAULT (-1, padrão), NONE (0), MD2 (1), MD5 (2), STRAIGHT (4), OEM (5), RMCP_PLUS (6).
ipmi_privilege	string	Nível de privilégio da sessão IPMI. Valores possíveis:[1] CALLBACK (1), USER (2, padrão), OPERATOR (3), ADMIN (4), OEM (5).
ipmi_username	string	Nome de usuário para verificações IPMI.
ipmi_password	string	Senha para verificações IPMI.
templates		Elemento raiz para templates vinculados.
name	string	(obrigatório) Nome do template.
groups		Elemento raiz para grupos de host aos quais o host pertence.
name	string	(obrigatório) Nome do grupo de host.
interfaces		Elemento raiz para interfaces de host.
items		Elemento raiz para itens do host.
discovery_rules		Elemento raiz para regras de descoberta de baixo nível do host.
http_tests		Elemento raiz para cenários web do host.
tags		Elemento raiz para tags do host.
tag	string	(obrigatório) Nome da tag.
value	string	Valor da tag.
macros		Elemento raiz para macros do host.
macro	string	(obrigatório) Nome da macro de usuário.
type	string	Tipo de macro de usuário. Valores possíveis:[1] TEXT (0, padrão), SECRET_TEXT (1), VAULT (2).
value	string	Valor da macro de usuário.
description	string	Descrição da macro de usuário.
inventory		Elemento raiz para inventário do host.
<inventory_property>	string	Propriedade de inventário. Todas as propriedades têm seu respectivo elemento (`type`, `name`, `os`, etc.; por exemplo, consulte Formato de exportação).
inventory_mode	string	Modo de inventário. Valores possíveis:[1] DISABLED (-1), MANUAL (0, padrão), AUTOMATIC (1).
valuemaps		Elemento raiz para mapas de valor do host.

Note:
Veja também: Objeto host (consulte a propriedade relevante com o nome correspondente).

Interfaces de host

Elemento	Tipo	Descrição
default	string	Se esta é a interface principal do host. Observe que só pode haver uma interface principal de um tipo em um host. Valores possíveis:[1] NÃO (0), SIM (1, padrão).
type	string	Tipo de interface. Valores possíveis:[1] ZABBIX (1, padrão), SNMP (2), IPMI (3), JMX (4).
useip	string	Se deve usar o IP como interface para conectar ao host (caso contrário, o DNS será usado). Valores possíveis:[1] NÃO (0), SIM (1, padrão).
ip	string	(obrigatório para conexões IP) Endereço IP (IPv4 ou IPv6).
dns	string	(obrigatório para conexões DNS) Nome DNS.
port	string	Número da porta.
details		Elemento raiz para detalhes da interface.
version	string	Usar esta versão SNMP. Valores possíveis:[1] SNMPV1 (1), SNMP_V2C (2, padrão), SNMP_V3 (3).
community	string	(obrigatório para itens SNMPv1 e SNMPv2) Comunidade SNMP.
max_repetitions	string	Valor máximo de repetição para solicitações nativas SNMP bulk (GetBulkRequest-PDUs). Suportado para itens SNMPv2 e SNMPv3 (`discovery[]` e `walk[]`). Padrão: 10.

Elemento	Tipo	Descrição
contextname	string	Nome do contexto SNMPv3. Suportado para itens SNMPv3.
securityname	string	Nome de segurança SNMPv3. Suportado para itens SNMPv3.
securitylevel	string	Nível de segurança SNMPv3. Suportado para itens SNMPv3. Valores possíveis:[1] NOAUTHNOPRIV (0, padrão), AUTHNOPRIV (1), AUTHPRIV (2).
authprotocol	string	Protocolo de autenticação SNMPv3. Suportado para itens SNMPv3. Valores possíveis:[1] MD5 (0, padrão), SHA1 (1), SHA224 (2), SHA256 (3), SHA384 (4), SHA512 (5).
authpassphrase	string	Senha de autenticação SNMPv3. Suportado para itens SNMPv3.
privprotocol	string	Protocolo de privacidade SNMPv3. Suportado para itens SNMPv3. Valores possíveis:[1] DES (0, padrão), AES128 (1), AES192 (2), AES256 (3), AES192C (4), AES256C (5).
privpassphrase	string	Senha de privacidade SNMPv3. Suportado para itens SNMPv3.
bulk	string	Usar solicitações bulk para SNMP. Valores possíveis:[1] NÃO (0), SIM (1, padrão).
interface_ref	string	Nome de referência da interface a ser usado nos items (formato: `if<N>`).

Note:
Veja também: Objeto de interface de host (consulte a propriedade relevante com o nome correspondente).

Itens do host

Elemento	Tipo	Descrição
name	string	(obrigatório) Nome do item.
type	string	Tipo do item. Valores possíveis:[1] ZABBIX_PASSIVE (0, padrão), TRAP (2), SIMPLE (3), INTERNAL (5), ZABBIX_ACTIVE (7), EXTERNAL (10), ODBC (11), IPMI (12), SSH (13), TELNET (14), CALCULATED (15), JMX (16), SNMP_TRAP (17), DEPENDENT (18), HTTP_AGENT (19), SNMP_AGENT (20), ITEM_TYPE_SCRIPT (21), ITEM_TYPE_BROWSER (22).
snmp_oid	string	(obrigatório para itens SNMP_AGENT) ID do objeto SNMP.
key	string	(obrigatório) Chave do item.
delay	string	Intervalo de atualização do item. Padrão: 1m. O valor será sempre 0 para itens TRAP.
history	string	Período de tempo (usando sufixo de tempo, macro de usuário ou macro LLD) de quanto tempo os dados do histórico devem ser armazenados. Padrão: 31d.
trends	string	Período de tempo (usando sufixo de tempo, macro de usuário ou macro LLD) de quanto tempo os dados de tendências devem ser armazenados. Padrão: 365d.
status	string	Status do item. Valores possíveis:[1] ENABLED (0, padrão), DISABLED (1).
value_type	string	Tipo de valor recebido. Valores possíveis:[1] FLOAT (0), CHAR (1), LOG (2), UNSIGNED (3, padrão), TEXT (4), BINARY (5).
allowed_hosts	string	Lista de endereços IP separados por vírgula dos hosts permitidos a enviar dados para o item. Suportado para itens TRAP e HTTP_AGENT.
units	string	Unidades do valor recebido (bps, B, etc.).
params	text	Parâmetros adicionais dependendo do tipo do item (script executado para itens SSH e TELNET; consulta SQL para itens ODBC; fórmula para itens CALCULATED; script para itens ITEM_TYPE_SCRIPT e ITEM_TYPE_BROWSER).
ipmi_sensor	string	Sensor IPMI. Suportado para itens IPMI.

Elemento	Tipo	Descrição
authtype	string	Tipo de autenticação. Suportado para itens SSH e HTTP_AGENT. Valores possíveis para itens SSH:[1] PASSWORD (0, padrão), PUBLIC_KEY (1). Valores possíveis para itens HTTP_AGENT:[1] NONE (0, padrão), BASIC (1), NTLM (2), Kerberos (3), ou Digest (4).
username	string	(obrigatório para itens SSH e TELNET) Nome de usuário para autenticação. Suportado para itens SIMPLE, ODBC, JMX e HTTP_AGENT. Quando usado para itens JMX, `password` (veja abaixo) também deve ser especificado ou ambos os elementos devem ser deixados em branco.
password	string	(obrigatório para itens SSH e TELNET) Senha para autenticação. Suportado para itens SIMPLE, ODBC, JMX e HTTP_AGENT. Quando usado para itens JMX, `username` (veja acima) também deve ser especificado ou ambos os elementos devem ser deixados em branco.
publickey	string	(obrigatório para itens SSH) Nome do arquivo da chave pública.
privatekey	string	(obrigatório para itens SSH) Nome do arquivo da chave privada.
description	text	Descrição do item.
inventory_link	string	Campo de inventário do host que é preenchido pelo item. Valores possíveis:[1] NONE (0), ALIAS (4), etc. (veja Inventário do host para campos suportados).
valuemap		Elemento raiz para mapas de valor do item.
name	string	(obrigatório) Nome do mapa de valor a ser usado para o item.
logtimefmt	string	Formato da hora nas entradas de log. Suportado para itens do tipo de valor LOG.
preprocessing		Elemento raiz para pré-processamento de valor do item.
step		Elemento raiz para etapas de pré-processamento de valor do item do host.
interface_ref	string	Referência à interface do host (formato: `if<N>`).
jmx_endpoint	string	Endpoint JMX. Suportado para itens JMX.
master_item		(obrigatório para itens DEPENDENT) Elemento raiz para o item mestre do item dependente.
key	string	(obrigatório) Chave do item mestre do item dependente.
timeout	string	Timeout da solicitação de polling dos dados do item. Suportado para a lista de tipos de item em Timeouts.
url	string	(obrigatório para itens HTTP_AGENT) String de URL.
query_fields		Elemento raiz para parâmetros de consulta. Suportado para itens HTTP_AGENT.
name	string	(obrigatório para itens HTTP_AGENT) Nome do parâmetro de consulta.
value	string	Valor do parâmetro de consulta. Suportado para itens HTTP_AGENT.
parameters		Elemento raiz para parâmetros definidos pelo usuário. Suportado para itens ITEM_TYPE_SCRIPT e ITEM_TYPE_BROWSER.
name	string	(obrigatório para itens ITEM_TYPE_SCRIPT e ITEM_TYPE_BROWSER) Nome do parâmetro definido pelo usuário.
value	string	Valor do parâmetro definido pelo usuário. Suportado para itens ITEM_TYPE_SCRIPT e ITEM_TYPE_BROWSER.
posts	string	Dados do corpo da requisição HTTP(S). Suportado para itens HTTP_AGENT.
status_codes	string	Intervalos de códigos de status HTTP necessários, separados por vírgulas. Suportado para itens HTTP_AGENT.
follow_redirects	string	Seguir redirecionamentos de resposta ao coletar dados. Suportado para itens HTTP_AGENT. Valores possíveis:[1] NO (0), YES (1, padrão).
post_type	string	Tipo de corpo de dados postados. Suportado para itens HTTP_AGENT. Valores possíveis:[1] RAW (0, padrão), JSON (2), XML (3).
http_proxy	string	String de conexão do proxy HTTP(S). Suportado para itens HTTP_AGENT.
headers		Elemento raiz para cabeçalhos de requisição HTTP(S). Suportado para itens HTTP_AGENT.
name	string	(obrigatório para itens HTTP_AGENT) Nome do cabeçalho.
value	string	(obrigatório para itens HTTP_AGENT) Valor do cabeçalho.

Elemento	Tipo	Descrição
retrieve_mode	string	Qual parte da resposta deve ser armazenada. Suportado para itens HTTP_AGENT. Valores possíveis:[1] BODY (0, padrão), HEADERS (1), BOTH (2).
request_method	string	Tipo de método de requisição. Suportado para itens HTTP_AGENT. Valores possíveis:[1] GET (0, padrão), POST (1), PUT (2), HEAD (3).
output_format	string	Como processar a resposta. Suportado para itens HTTP_AGENT. Valores possíveis:[1] RAW (0, padrão), JSON (1).
allow_traps	string	Permitir popular o valor de forma semelhante ao item trapper. Suportado para itens HTTP_AGENT. Valores possíveis:[1] NO (0, padrão), YES (1).
ssl_cert_file	string	Caminho do arquivo da chave pública SSL. Suportado para itens HTTP_AGENT.
ssl_key_file	string	Caminho do arquivo da chave privada SSL. Suportado para itens HTTP_AGENT.
ssl_key_password	string	Senha para o arquivo da chave SSL. Suportado para itens HTTP_AGENT.
verify_peer	string	Se deve validar que o certificado do host é autêntico. Suportado para itens HTTP_AGENT. Valores possíveis:[1] NO (0, padrão), YES (1).
verify_host	string	Se deve validar que o nome do host para a conexão corresponde ao do certificado do host. Suportado para itens HTTP_AGENT. Valores possíveis:[1] NO (0, padrão), YES (1).
tags		Elemento raiz para tags do item.
tag	string	(obrigatório) Nome da tag.
value	string	Valor da tag.
triggers		Elemento raiz para triggers de itens do host.

Note:
Veja também: Objeto item (consulte a propriedade relevante com o nome correspondente).

Etapas de pré-processamento do valor do item do host

Elemento	Tipo	Descrição
type	string	(obrigatório) O tipo de etapa de pré-processamento do valor do item. Valores possíveis:[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, calculado: valor recebido - valor anterior), CHANGE_PER_SECOND (10, calculado: (valor recebido - valor anterior)/(hora atual - hora da última verificação)), 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		(obrigatório) Elemento raiz para os parâmetros da etapa de pré-processamento do valor do item.
parameter	string	Parâmetro individual da etapa de pré-processamento do valor do item.
error_handler	string	Tipo de ação usada em caso de falha na etapa de pré-processamento. Valores possíveis:[1] ORIGINAL_ERROR (0, padrão), DISCARD_VALUE (1), CUSTOM_VALUE (2), CUSTOM_ERROR (3).
error_handler_params	string	Parâmetros do manipulador de erros.

Note:
Veja também: Objeto de pré-processamento do item (consulte a propriedade relevante com o nome correspondente).

Triggers de item de host

Elemento	Tipo	Descrição
uuid	string	(obrigatório) Identificador exclusivo para este trigger.
expression	string	(obrigatório) Expressão do trigger.
recovery_mode	string	Base para geração de eventos OK. Valores possíveis:[1] EXPRESSION (0, padrão), RECOVERY_EXPRESSION (1), NONE (2).
recovery_expression	string	Expressão de recuperação do trigger.
correlation_mode	string	Modo de correlação (sem correlação de eventos ou correlação de eventos por tag). Valores possíveis:[1] DISABLED (0, padrão), TAG_VALUE (1).
correlation_tag	string	O nome da tag a ser usada para correlação de eventos.
name	string	(obrigatório) Nome do trigger.
event_name	string	Nome do evento.
opdata	string	Dados operacionais.
url_name	string	Rótulo para a URL associada ao trigger.
url	string	URL associada ao trigger.
status	string	Status do trigger. Valores possíveis:[1] ENABLED (0, padrão), DISABLED (1).
priority	string	Severidade do trigger. Valores possíveis:[1] NOT_CLASSIFIED (0, padrão), INFO (1), WARNING (2), AVERAGE (3), HIGH (4), DISASTER (5).
description	text	Descrição do trigger.
type	string	Tipo de geração de evento (evento de problema único ou múltiplos eventos de problema). Valores possíveis:[1] SINGLE (0, padrão), MULTIPLE (1).
manual_close	string	Fechamento manual de eventos de problema. Valores possíveis:[1] NO (0, padrão), YES (1).
dependencies		Elemento raiz para dependências.
name	string	(obrigatório) Nome do trigger de dependência.
expression	string	(obrigatório) Expressão do trigger de dependência.
recovery_expression	string	Expressão de recuperação do trigger de dependência.
tags		Elemento raiz para tags do trigger.
tag	string	(obrigatório) Nome da tag.
value	string	Valor da tag.

Note:
Veja também: Objeto trigger (consulte a propriedade relevante com o nome correspondente).

Regras de descoberta de baixo nível de host

Attention:
A maioria dos elementos das regras de descoberta de baixo nível de host são os mesmos que para itens de host. A tabela abaixo descreve os elementos que diferem dos itens de host.

Elemento	Tipo	Descrição
type	string	Tipo de item. Valores possíveis:[1] ZABBIX_PASSIVE (0, padrão), 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	(obrigatório) A chave da regra de descoberta de baixo nível.
filter		Elemento raiz para filtros de regras de descoberta de baixo nível de host.
lifetime	string	Período de tempo (usando segundos, sufixo de tempo ou macro de usuário), após o qual os recursos não descobertos serão excluídos. Padrão: 7d.
lifetime_type	string	Cenário para excluir recursos LLD perdidos. Valores possíveis: DELETE_NEVER, DELETE_IMMEDIATELY, DELETE_AFTER.
enabled_lifetime	string	Período de tempo (usando segundos, sufixo de tempo ou macro de usuário), após o qual os recursos não descobertos serão desabilitados.

Elemento	Tipo	Descrição
enabled_lifetime_type	string	Cenário para desabilitar recursos LLD perdidos. Valores possíveis: DISABLE_NEVER, DISABLE_IMMEDIATELY, DISABLE_AFTER.
item_prototypes		Elemento raiz para elementos de protótipo de item de host, que são os mesmos que para itens de host.
trigger_prototypes		Elemento raiz para elementos de protótipo de trigger de host, que são os mesmos que para triggers de item de host.
graph_prototypes		Elemento raiz para protótipos de gráficos de host, que são os mesmos que para gráficos de host.
host_prototypes		Elemento raiz para protótipos de host, que são os mesmos que para hosts.
master_item	string	(obrigatório para regras DEPENDENT) Elemento raiz para o item mestre da regra dependente.
lld_macro_paths		Elemento raiz para caminhos de macro de regra de descoberta de baixo nível.
lld_macro	string	(obrigatório) Nome da macro da regra de descoberta de baixo nível.
path	string	(obrigatório) Seletor para o valor, que será atribuído à macro correspondente.
preprocessing		Elemento raiz para pré-processamento de valor de regra de descoberta de baixo nível.
step		Elemento raiz para elementos de etapa de pré-processamento de valor de regra de descoberta de baixo nível, que são os mesmos que para etapas de pré-processamento de valor de item de host, exceto com menos valores possíveis. Veja também: Objeto de pré-processamento de regra LLD.
overrides		Elemento raiz para regras de sobrescrita de regra de descoberta de baixo nível.
name	string	(obrigatório) Nome único da sobrescrita.
step	string	(obrigatório) Número de ordem único da sobrescrita.
stop	string	Parar o processamento das próximas sobrescritas se corresponder.
filter		Elemento raiz para elementos de filtro de regra de sobrescrita de regra de descoberta de baixo nível, que são os mesmos que para filtros de regras de descoberta de baixo nível de host.
operations		Elemento raiz para operações de sobrescrita de regra de descoberta de baixo nível de host.

Note:
Veja também: Objeto de regra LLD (consulte a propriedade relevante com o nome correspondente).

Filtros de regras de descoberta de baixo nível de host

Elemento	Tipo	Descrição
evaltype	string	Substitui o método de avaliação da condição do filtro. Valores possíveis:[1] AND_OR (0, padrão), AND (1), OR (2), FORMULA (3).
formula	string	Fórmula de cálculo personalizada para as condições do filtro.
conditions		Elemento raiz para as condições do filtro.
macro	string	(obrigatório) O nome da macro de descoberta de baixo nível para realizar a verificação.
value	string	Valor para comparar.
operator	string	Operador de condição. Valores possíveis:[1] MATCHES_REGEX (8, padrão), NOT_MATCHES_REGEX (9).
formulaid	string	(obrigatório) ID exclusivo arbitrário que é usado para referenciar uma condição da expressão personalizada. Só pode conter letras maiúsculas. O ID deve ser definido pelo usuário ao modificar as condições do filtro, mas será gerado novamente ao solicitá-las posteriormente.

Note:
Veja também: Objeto de filtro de regra de LLD (consulte a propriedade relevante com o nome correspondente).

Operações de substituição de regras de descoberta de baixo nível de host

Elemento	Tipo	Descrição
operationobject	string	Objeto ao qual a operação é aplicada. Valores possíveis:[1] ITEM_PROTOTYPE (0), TRIGGER_PROTOTYPE (1), GRAPH_PROTOTYPE (2), HOST_PROTOTYPE (3).

Elemento	Tipo	Descrição
operator	string	Operador da condição de substituição. Valores possíveis:[1] EQUAL (1), NOT_EQUAL (2), LIKE (3), NOT_LIKE (4), REGEXP (5), NOT_REGEXP (6).
value	string	Uma expressão regular ou uma string para o operador de condição de substituição.
status	string	Status do objeto após a operação de substituição.
discover	string	Se o objeto é adicionado como resultado da descoberta.
delay	string	Intervalo de atualização definido para o protótipo de item após a operação de substituição.
history	string	Período de armazenamento do histórico definido para o protótipo de item após a operação de substituição.
trends	string	Período de armazenamento de tendências definido para o protótipo de item após a operação de substituição.
severity	string	Gravidade do protótipo de trigger definida após a operação de substituição.
tags		Elemento raiz para as tags definidas para o objeto após a operação de substituição.
tag	string	(obrigatório) Nome da tag.
value	string	Valor da tag.
templates		Elemento raiz para os templates vinculados ao protótipo de host após a operação de substituição.
name	string	(obrigatório) Nome do template.
inventory_mode	string	Modo de inventário do protótipo de host definido após a operação de substituição.

Note:
Veja também: Objeto de operação de substituição de regra LLD (consulte a propriedade relevante com o nome correspondente).

Cenários web do host

Elemento	Tipo	Descrição
uuid	string	(obrigatório) Identificador exclusivo para este cenário web.
name	string	(obrigatório) Nome do cenário web.
delay	string	Frequência (usando segundos, sufixo de tempo ou macro de usuário) de execução do cenário web. Padrão: 1m.
attempts	integer	O número de tentativas para executar as etapas do cenário web. Valores possíveis: 1-10 (padrão: 1).
agent	string	Agent do cliente. O Zabbix irá simular ser o navegador selecionado. Isso é útil quando um site retorna conteúdo diferente para navegadores diferentes. Padrão: Zabbix.
http_proxy	string	Proxy que será usado pelo cenário web fornecido como: `http://[username[:password]@]proxy.example.com[:port]`
variables		Elemento raiz para variáveis de cenário web que podem ser usadas nas etapas do cenário.
name	string	(obrigatório) Nome da variável.
value	text	(obrigatório) Valor da variável.
headers		Elemento raiz para cabeçalhos HTTP a serem enviados ao realizar uma requisição. Os cabeçalhos devem ser listados usando a mesma sintaxe que apareceriam no protocolo HTTP.
name	string	(obrigatório) Nome do cabeçalho.
value	text	(obrigatório) Valor do cabeçalho.
status	string	Status do cenário web. Valores possíveis:[1] ENABLED (0, padrão), DISABLED (1).
authentication	string	Método de autenticação. Valores possíveis:[1] NONE (0, padrão), BASIC (1), NTLM (2), Kerberos (3) ou Digest (4).
http_user	string	Nome de usuário usado para autenticação BASIC (HTTP), NTLM, Kerberos ou Digest.
http_password	string	Senha usada para autenticação BASIC (HTTP), NTLM, Kerberos ou Digest.
verify_peer	string	Verifica o certificado SSL do servidor web. Valores possíveis:[1] NO (0, padrão), YES (1).
verify_host	string	Verifica se o campo *Common Name* ou o campo *Subject Alternate Name* do certificado do servidor web corresponde. Valores possíveis:[1] NO (0, padrão), YES (1).

Elemento	Tipo	Descrição
ssl_cert_file	string	Nome do arquivo de certificado SSL usado para autenticação do cliente (deve estar no formato PEM).
ssl_key_file	string	Nome do arquivo de chave privada SSL usado para autenticação do cliente (deve estar no formato PEM).
ssl_key_password	string	Senha do arquivo de chave privada SSL.
steps		(obrigatório) Elemento raiz para etapas do cenário web do host.
tags		Elemento raiz para tags do cenário web.
tag	string	(obrigatório) Nome da tag.
value	string	Valor da tag.

Note:
Veja também: Objeto de cenário web (consulte a propriedade relevante com o nome correspondente).

Etapas do cenário web do host

Elemento	Tipo	Descrição
name	string	(obrigatório) Nome da etapa do cenário web.
url	string	(obrigatório) URL para monitoramento.
query_fields		Elemento raiz para parâmetros de consulta (um array de campos HTTP a serem adicionados à URL ao executar uma solicitação).
name	string	(obrigatório) Nome do parâmetro de consulta.
value	string	Valor do parâmetro de consulta.
posts		Elemento raiz para variáveis HTTP POST (uma string (dados brutos de post) ou um array de campos HTTP (dados de campo de formulário)).
name	string	(obrigatório) Nome do campo de post.
value	string	(obrigatório) Valor do campo de post.
variables		Elemento raiz das variáveis de nível de etapa (macros) que devem ser aplicadas após esta etapa. Se o valor da variável tiver o prefixo 'regex:', então seu valor será extraído dos dados retornados por esta etapa de acordo com o padrão de expressão regular após o prefixo 'regex:'
name	string	(obrigatório) Nome da variável.
value	text	(obrigatório) Valor da variável.
headers		Elemento raiz para cabeçalhos HTTP a serem enviados ao executar uma solicitação.
name	string	(obrigatório) Nome do cabeçalho.
value	text	(obrigatório) Valor do cabeçalho.
follow_redirects	string	Seguir redirecionamentos HTTP. Valores possíveis:[1] NÃO (0), SIM (1, padrão).
retrieve_mode	string	Modo de obtenção da resposta HTTP. Valores possíveis:[1] BODY (0, padrão), HEADERS (1), BOTH (2).
timeout	string	Timeout (usando segundos, sufixo de tempo ou macro de usuário) da execução da etapa. Padrão: 15s.
required	string	Texto que deve estar presente na resposta (ignorado se vazio).
status_codes	string	Uma lista separada por vírgulas de códigos de status HTTP aceitos (por exemplo, 200-201,210-299; ignorado se vazio).

Note:
Veja também: Objeto de etapa de cenário web (consulte a propriedade relevante com o nome correspondente).

Gráficos de host

Elemento	Tipo	Descrição
uuid	string	Identificador exclusivo do gráfico.
name	string	(obrigatório) Nome do gráfico.

Elemento	Tipo	Descrição
width	integer	Largura do gráfico, em pixels. Usado para visualização e para gráficos de pizza/explodidos. Valores possíveis:[1] 20-65535 (padrão: 900).
height	integer	Altura do gráfico, em pixels. Usado para visualização e para gráficos de pizza/explodidos. Valores possíveis:[1] 20-65535 (padrão: 900).
yaxismin	double	Valor mínimo do eixo Y. Suportado para valor mínimo FIXO do eixo Y. Padrão: 0.
yaxismax	double	Valor máximo do eixo Y. Suportado para valor máximo FIXO do eixo X. Padrão: 0.
show_work_period	string	Destacar horas não úteis. Suportado para gráficos NORMAL e EMPILHADO. Valores possíveis:[1] NÃO (0), SIM (1, padrão).
show_triggers	string	Exibir valores simples de trigger como uma linha. Suportado para gráficos NORMAL e EMPILHADO. Valores possíveis:[1] NÃO (0), SIM (1, padrão).
type	string	Tipo de gráfico. Valores possíveis:[1] NORMAL (0, padrão), EMPILHADO (1), PIZZA (2), EXPLODIDO (3).
show_legend	string	Exibir legenda do gráfico. Valores possíveis:[1] NÃO (0), SIM (1, padrão).
show_3d	string	Habilitar estilo 3D. Suportado para gráficos NORMAL e EMPILHADO. Valores possíveis:[1] NÃO (0, padrão), SIM (1).
percent_left	double	Exibir a linha de percentil para o eixo esquerdo. Suportado para gráficos NORMAL. Padrão: 0.
percent_right	double	Exibir a linha de percentil para o eixo direito. Suportado para gráficos NORMAL. Padrão: 0.
ymin_type_1	string	Valor mínimo do eixo Y. Suportado para gráficos NORMAL e EMPILHADO. Valores possíveis:[1] CALCULADO (0, padrão), FIXO (1), ITEM (2).
ymin_item_1		(obrigatório se `ymin_type_1` for definido como `ITEM`) Elemento raiz para detalhes individuais do item.
host	string	(obrigatório) Host do item.
key	string	(obrigatório) Chave do item.
ymax_type_1	string	Valor máximo do eixo Y. Suportado para gráficos NORMAL e EMPILHADO. Valores possíveis:[1] CALCULADO (0, padrão), FIXO (1), ITEM (2).
ymax_item_1		(obrigatório se `ymax_type_1` for definido como `ITEM`) Elemento raiz para detalhes individuais do item.
host	string	(obrigatório) Host do item.
key	string	(obrigatório) Chave do item.
graph_items		(obrigatório) Elemento raiz para itens de gráfico de host.

Note:
Veja também: Objeto gráfico (consulte a propriedade relevante com o nome correspondente).

Itens do gráfico do host

Elemento	Tipo	Descrição
sortorder	integer	Ordem de desenho. O menor valor é desenhado primeiro. Pode ser usado para desenhar linhas ou regiões atrás (ou na frente) de outra.
drawtype	string	Estilo de desenho do item do gráfico. Suportado para gráficos NORMAIS. Valores possíveis:[1] SINGLE_LINE (0, padrão), FILLED_REGION (1), BOLD_LINE (2), DOTTED_LINE (3), DASHED_LINE (4), GRADIENT_LINE (5).

Elemento	Tipo	Descrição
color	string	Cor do elemento (6 símbolos, hex).
yaxisside	string	Lado do gráfico onde a escala Y do item do gráfico será desenhada. Suportado para gráficos NORMAIS e EMPILHADOS.
calc_fnc	string	Dados a serem desenhados se mais de um valor existir para um item. Valores possíveis:[1] MIN (1), AVG (2, padrão), MAX (4), ALL (7; mínimo, média e máximo; suportado para gráficos simples), LAST (9, suportado para gráficos de pizza/explodidos).
type	string	Tipo de item do gráfico. Valores possíveis:[1] SIMPLE (0, padrão), GRAPH_SUM (2; o valor do item representa toda a pizza; suportado para gráficos de pizza/explodidos).
item		(obrigatório) Item individual.
host	string	(obrigatório) Host do item.
key	string	(obrigatório) Chave do item.

Note:
Veja também: Objeto de item de gráfico (consulte a propriedade relevante com o nome correspondente).

Mapas de valores do host

Elemento	Tipo	Descrição
uuid	string	(obrigatório) Identificador exclusivo para este mapa de valores.
name	string	(obrigatório) Nome do mapa de valores.
mapping		Elemento raiz para mapeamentos.
type	string	Tipo de correspondência de mapeamento. Valores possíveis:[1] EQUAL (0, padrão), GREATER_OR_EQUAL (2), LESS_OR_EQUAL (3), IN_RANGE (4), REGEXP (5), DEFAULT (6).
value	string	Valor original.
newvalue	string	(obrigatório) Valor para o qual o valor original é mapeado.

Note:
Veja também: Objeto de mapa de valores (consulte a propriedade relevante com o nome correspondente).

Notas de rodapé [1] Os valores inteiros da API entre colchetes, por exemplo, ENABLED (0), são mencionados apenas para referência. Para obter mais informações, consulte a página do objeto da API vinculada na entrada da tabela ou no final de cada seção.

5 Mapas de rede

Visão geral

A exportação de mapas de rede contém:

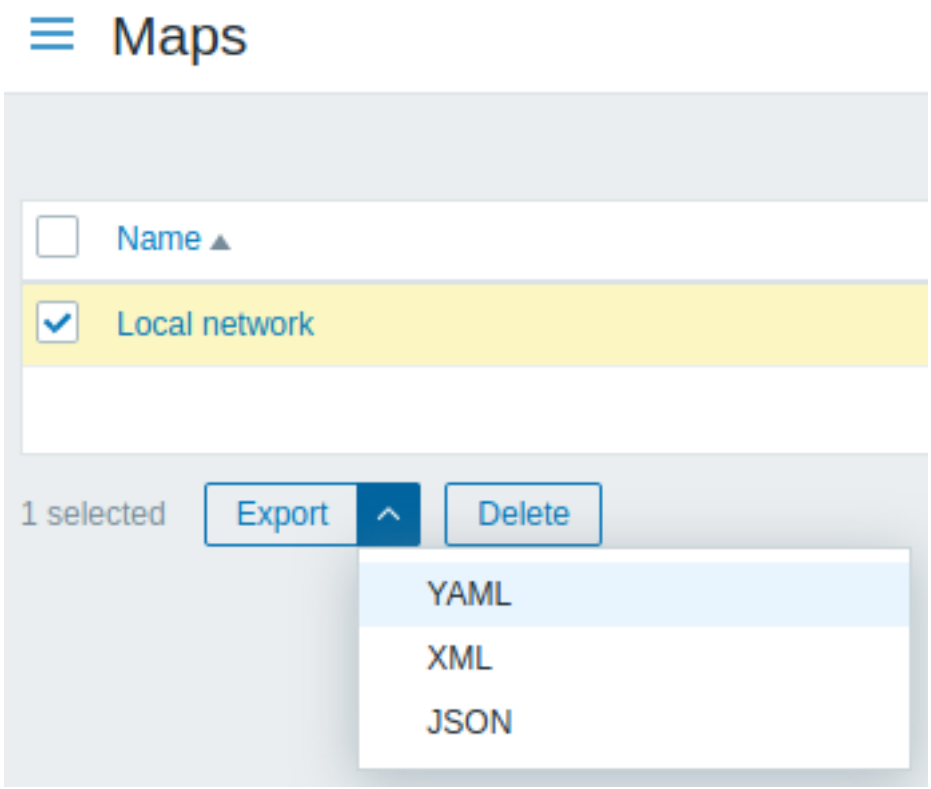
- Todas as imagens relacionadas
- Estrutura do mapa (todas as configurações do mapa, todos os elementos contidos com suas configurações, links do mapa e indicadores de status dos links do mapa)

Warning:
Quaisquer grupos de hosts, hosts, triggers, outros mapas ou outros elementos que possam estar relacionados ao mapa exportado não são exportados. Assim, se pelo menos um dos elementos aos quais o mapa se refere estiver ausente, a importação falhará.

Exportando

Para exportar mapas de rede, faça o seguinte:

1. Vá para *Monitoramento* → *Mapas*.
2. Marque as caixas de seleção dos mapas de rede a serem exportados.
3. Clique em *Exportar* abaixo da lista.



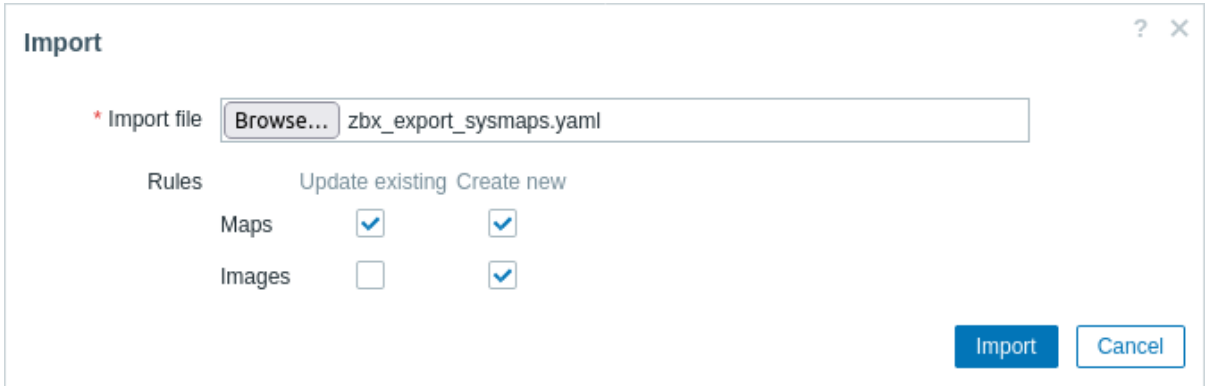
Dependendo do formato selecionado, os mapas são exportados para um arquivo local com um nome padrão:

- `zabbix_export_maps.yaml` - na exportação YAML (opção padrão para exportação);
- `zabbix_export_maps.xml` - na exportação XML;
- `zabbix_export_maps.json` - na exportação JSON.

Importando

Para importar mapas de rede, faça o seguinte:

1. Vá para *Monitoramento* → *Mapas*.
2. Clique em *Importar* no canto superior direito.
3. Selecione o arquivo de importação.
4. Marque as opções necessárias nas regras de importação.
5. Clique em *Importar* no canto inferior direito do formulário de configuração.



Regras de importação:

Regra	Descrição
Atualizar existentes	Os mapas existentes serão atualizados usando os dados do arquivo de importação. Caso contrário, eles não serão atualizados.
Criar novos	Novos mapas serão criados usando os dados do arquivo de importação. Caso contrário, eles não serão criados.

Se você desmarcar ambas as opções de mapa e marcar as opções respectivas para imagens, apenas as imagens serão importadas.
A importação de imagens está disponível apenas para usuários *Super admin*.

Uma mensagem de sucesso ou falha da importação será exibida no frontend.

> **Warning:**
> Se substituir uma imagem existente, isso afetará todos os mapas que estiverem usando essa imagem.

Formato de exportação

Exportar para YAML:

```yaml
zabbix_export:
  version: '7.0'
  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: {  }
```

Elementos exportados Os elementos exportados são explicados na tabela abaixo.

Elemento	Tipo	Descrição
images		Elemento raiz para imagens.
name	string	Nome único da imagem.
imagetype	integer	Tipo de imagem. Valores possíveis: 1 - Imagem; 2 - Plano de fundo.
encodedImage	string	Imagem codificada em Base64.
maps		Elemento raiz para mapas.

Mapas

Elemento	Tipo	Descrição
name	string	Nome exclusivo do mapa.
width	integer	Largura do mapa, em pixels.
height	integer	Altura do mapa, em pixels.
label_type	integer	Tipo de rótulo do elemento do mapa. Valores possíveis: 0 - Rótulo; 1 - Endereço IP do host; 2 - Nome do elemento; 3 - Apenas status; 4 - Nada.
label_location	integer	Localização do rótulo do elemento do mapa por padrão. Valores possíveis: 0 - Inferior; 1 - Esquerda; 2 - Direita; 3 - Superior.
highlight	integer	Ativar realce de ícone para triggers ativas e status de host. Valores possíveis: 0 - Não; 1 - Sim.

Elemento	Tipo	Descrição
expandproblem	integer	Exibir trigger de problema para elementos com um único problema. Valores possíveis: 0 - Não; 1 - Sim.
markelements	integer	Destacar elementos do mapa que mudaram recentemente seu status. Valores possíveis: 0 - Não; 1 - Sim.
show_unack	integer	Exibição de problemas. Valores possíveis: 0 - Contagem de todos os problemas; 1 - Contagem de problemas não reconhecidos; 2 - Contagem de problemas reconhecidos e não reconhecidos separadamente.
severity_min	integer	Severidade mínima do trigger para mostrar no mapa por padrão. Valores possíveis: 0 - Não classificado; 1 - Informação; 2 - Aviso; 3 - Médio; 4 - Alto; 5 - Desastre.
show_suppressed	integer	Exibir problemas que, de outra forma, seriam suprimidos (não exibidos) devido à manutenção do host. Valores possíveis: 0 - Não; 1 - Sim.
grid_size	integer	Tamanho da célula de uma grade do mapa em pixels. Suportado se `grid_show` estiver definido como 0. Valores possíveis: 20, 40, 50, 75 ou 100.
grid_show	integer	Exibir uma grade na configuração do mapa. Valores possíveis: 0 - Sim; 1 - Não.
grid_align	integer	Alinhar automaticamente os ícones na configuração do mapa. Valores possíveis: 0 - Sim; 1 - Não.
label_format	integer	Usar configuração avançada de rótulo. Valores possíveis: 0 - Não; 1 - Sim.
label_type_host	integer	Exibir rótulo como rótulo de host. Suportado se `label_format` estiver definido como 1. Valores possíveis: 0 - Rótulo; 1 - Endereço IP do host; 2 - Nome do elemento; 3 - Apenas status; 4 - Nada; 5 - Rótulo personalizado.
label_type_hostgroup	integer	Exibir rótulo como rótulo de grupo de hosts. Suportado se `label_format` estiver definido como 1. Valores possíveis: 0 - Rótulo; 2 - Nome do elemento; 3 - Apenas status; 4 - Nada; 5 - Rótulo personalizado.

Elemento	Tipo	Descrição
label_type_trigger	integer	Exibir rótulo como rótulo de trigger. Suportado se `label_format` estiver definido como 1. Valores possíveis: 0 - Rótulo; 2 - Nome do elemento; 3 - Apenas status; 4 - Nada; 5 - Rótulo personalizado.
label_type_map	integer	Exibir rótulo como rótulo de mapa. Suportado se `label_format` estiver definido como 1. Valores possíveis: 0 - Rótulo; 2 - Nome do elemento; 3 - Apenas status; 4 - Nada; 5 - Rótulo personalizado.
label_type_image	integer	Exibir rótulo como rótulo de imagem. Suportado se `label_format` estiver definido como 1. Valores possíveis: 0 - Rótulo; 2 - Nome do elemento; 4 - Nada; 5 - Rótulo personalizado.
label_string_host	string	Rótulo personalizado para elementos de host. Suportado se `label_type_host` estiver definido como 5.
label_string_hostgroup	string	Rótulo personalizado para elementos de grupo de hosts. Suportado se `label_type_hostgroup` estiver definido como 5.
label_string_trigger	string	Rótulo personalizado para elementos de trigger. Suportado se `label_type_trigger` estiver definido como 5.
label_string_map	string	Rótulo personalizado para elementos de mapa. Suportado se `label_type_map` estiver definido como 5.
label_string_image	string	Rótulo personalizado para elementos de imagem. Suportado se `label_type_image` estiver definido como 5.
expand_macros	integer	Expandir macros em rótulos na configuração do mapa. Valores possíveis: 0 - Não; 1 - Sim.
background		Elemento raiz para imagem de fundo (se houver). Suportado se `imagetype` estiver definido como 2.
name	string	Nome da imagem de fundo.
iconmap		Elemento raiz para mapeamento de ícones (se houver).
name	string	Nome do mapeamento de ícones.
urls		Elemento raiz para URLs usadas por mapas ou cada elemento do mapa.
name	string	Nome do link.
url	string	URL do link.
elementtype	integer	Tipo de item do mapa ao qual o link pertence. Valores possíveis: 0 - Host; 1 - Mapa; 2 - Trigger; 3 - Grupo de hosts; 4 - Imagem.
selements		Elemento raiz para elementos do mapa.
shapes		Elemento raiz para formas do mapa.
type	integer	Tipo de forma. Valores possíveis: 0 - Retângulo; 1 - Elipse.
x	integer	Coordenadas X da forma em pixels.
y	integer	Coordenadas Y da forma em pixels.
width	integer	Largura da forma.

Elemento		Tipo	Descrição
	height	integer	Altura da forma.
	text	string	Texto dentro da forma.
	font	integer	Estilo da fonte do texto. Valores possíveis: 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	Tamanho da fonte em pixels.
	font_color	string	Cor da fonte representada em código hexadecimal.
	text_halign	integer	Alinhamento horizontal do texto. Valores possíveis: 0 - Centro; 1 - Esquerda; 2 - Direita.
	text_valign	integer	Alinhamento vertical do texto. Valores possíveis: 0 - Meio; 1 - Superior; 2 - Inferior.
	border_type	integer	Tipo de borda para a forma. Valores possíveis: 0 - Nenhum; 1 - Linha grossa; 2 - Linha pontilhada; 3 - Linha tracejada.
	border_width	integer	Largura da borda em pixels.
	border_color	string	Cor da borda representada em código hexadecimal.
	background_color	string	Cor de fundo (preenchimento) representada em código hexadecimal.
	zindex	integer	Valor para ordenação de todas as formas e linhas (z-index).
lines			Elemento raiz para linhas do mapa.
	x1	integer	Coordenadas X do ponto 1 da linha em pixels.
	y1	integer	Coordenadas Y do ponto 1 da linha em pixels.
	x2	integer	Coordenadas X do ponto 2 da linha em pixels.
	y2	integer	Coordenadas Y do ponto 2 da linha em pixels.
	line_type	integer	Tipo de linha. Valores possíveis: 0 - Nenhum; 1 - Linha grossa; 2 - Linha pontilhada; 3 - Linha tracejada.
	line_width	integer	Largura da linha em pixels.
	line_color	string	Cor da linha representada em código hexadecimal.
	zindex	integer	Valor para ordenação de todas as formas e linhas (z-index).
links			Elemento raiz para links entre elementos do mapa.
	drawtype	integer	Estilo do link. Valores possíveis: 0 - Linha; 2 - Linha grossa; 3 - Linha pontilhada; 4 - Linha tracejada.
	color	string	Cor do link (6 símbolos, hex).

Elemento	Tipo	Descrição
label	string	Rótulo do link.
selementid1	id	ID de um elemento a ser conectado.
selementid2	id	ID do outro elemento a ser conectado.
linktriggers		Elemento raiz para indicadores de status do link.

Note:
Veja também: Objeto mapa (consulte a propriedade relevante com o nome correspondente).

Elementos do mapa

Elemento	Tipo	Descrição
elementtype	integer	Tipo de elemento do mapa. Valores possíveis: 0 - Host; 1 - Mapa; 2 - Trigger; 3 - Grupo de hosts; 4 - Imagem.
elements		Elemento raiz para entidades Zabbix (host, grupo de hosts, mapa, etc.) que são representadas no mapa. Todas as entidades têm seu respectivo elemento (`host`, etc.; veja, por exemplo, Formato de exportação).
label	string	Rótulo do ícone.
label_location	integer	Localização do rótulo. Valores possíveis: -1 - Usar padrão do mapa; 0 - Inferior; 1 - Esquerda; 2 - Direita; 3 - Superior.
x	integer	Localização no eixo X.
y	integer	Localização no eixo Y.
elementsubtype	integer	Subtipo do elemento. Suportado se `elementtype` for definido como 3. Valores possíveis: 0 - Grupo de hosts único; 1 - Todos os grupos de hosts.
areatype	integer	Tamanho da área. Suportado se `elementtype` for definido como 1. Valores possíveis: 0 - Igual ao mapa inteiro; 1 - Tamanho personalizado.
width	integer	Largura da área. Suportado se `areatype` for definido como 1.
height	integer	Altura da área. Suportado se `areatype` for definido como 1.
viewtype	integer	Algoritmo de posicionamento da área. Suportado se `elementsubtype` for definido como 1. Valores possíveis: 0 - Distribuir uniformemente na área.
use_iconmap	integer	Usar mapeamento de ícone para este elemento. Relevante apenas se o mapeamento de ícone estiver ativado no nível do mapa. Valores possíveis: 0 - Não; 1 - Sim.
selementid	id	ID exclusivo do registro do elemento.
icon_off		Elemento raiz para a imagem a ser usada quando o elemento estiver no status 'OK'.
name	string	Nome exclusivo da imagem.

Elemento	Tipo	Descrição
icon_on		Elemento raiz para a imagem a ser usada quando o elemento estiver no status 'Problema'.
name	string	Nome exclusivo da imagem.
icon_disabled		Elemento raiz para a imagem a ser usada quando o elemento estiver desativado.
name	string	Nome exclusivo da imagem.
icon_maintenance		Elemento raiz para a imagem a ser usada quando o elemento estiver em manutenção.
name	string	Nome exclusivo da imagem.
urls		Elemento raiz para URLs usadas por mapas ou por cada elemento do mapa.
name	string	Nome do link.
url	string	URL do link.
evaltype	integer	Tipo de avaliação para tags.
tags		Elemento raiz para tags de problema (para elementos de host e grupo de hosts). Se as tags forem fornecidas, apenas problemas com essas tags serão exibidos no mapa.
tag	string	Nome da tag.
value	string	Valor da tag.
operator	integer	Operador.

Note:
Veja também: Objeto de elemento do mapa (consulte a propriedade relevante com o nome correspondente).

Indicadores de status de link de mapa

Elemento	Tipo	Descrição
drawtype	integer	Estilo do link quando o trigger está no estado 'Problema'. Valores possíveis: 0 - Linha; 2 - Linha em negrito; 3 - Linha pontilhada; 4 - Linha tracejada.
color	string	Cor do link (6 símbolos, hex) quando o trigger está no estado 'Problema'.
trigger		Elemento raiz para o trigger usado para indicar o status do link.
description	string	Nome do trigger.
expression	string	Expressão do trigger.
recovery_expression	string	Expressão de recuperação do trigger.

Note:
Veja também: Objeto de trigger de link de mapa (consulte a propriedade relevante com o nome correspondente).

5 Tipos de mídia

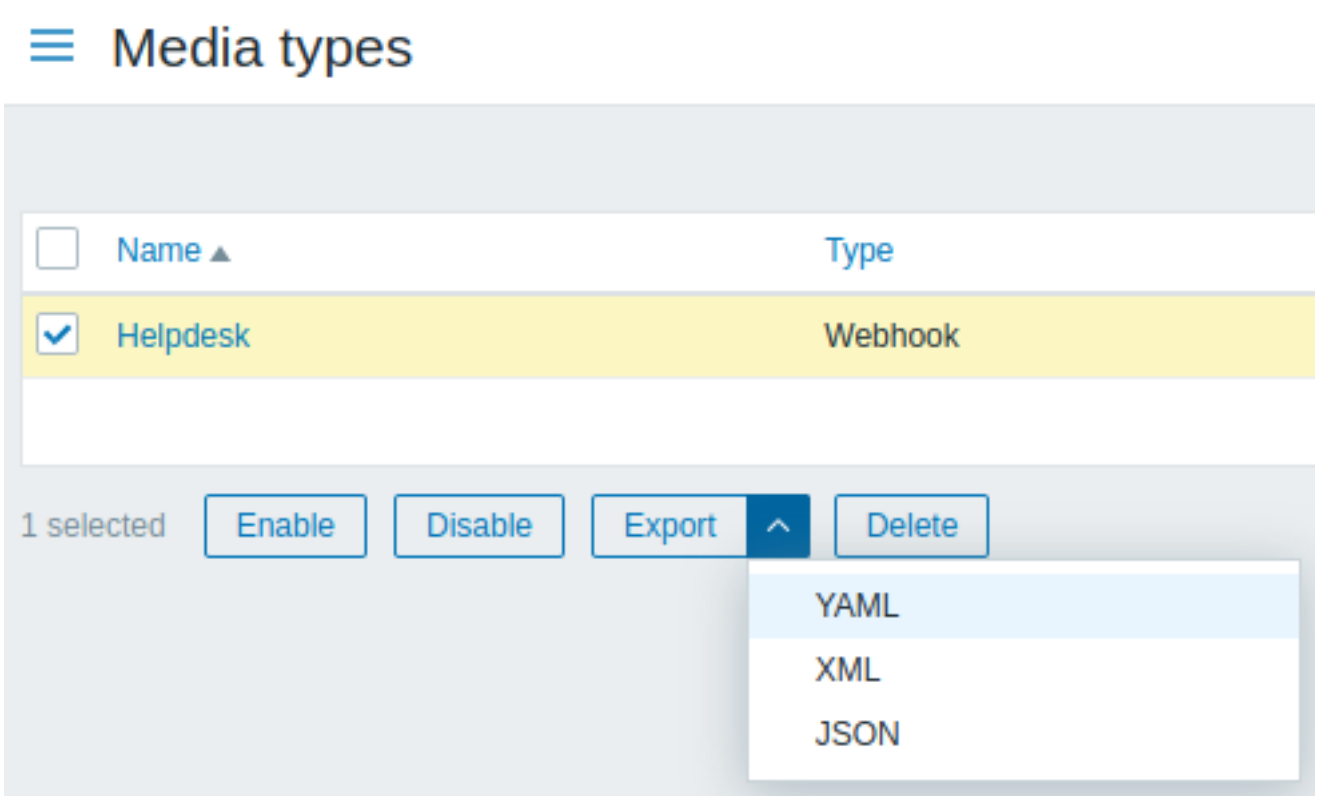
Visão geral

Os tipos de mídia são exportados com todos os objetos relacionados e relações de objetos.

Exportando

Para exportar tipos de mídia, faça o seguinte:

1. Vá para *Alertas* → *Tipos de mídia*.
2. Marque as caixas de seleção dos tipos de mídia a serem exportados.
3. Clique em *Exportar* abaixo da lista.



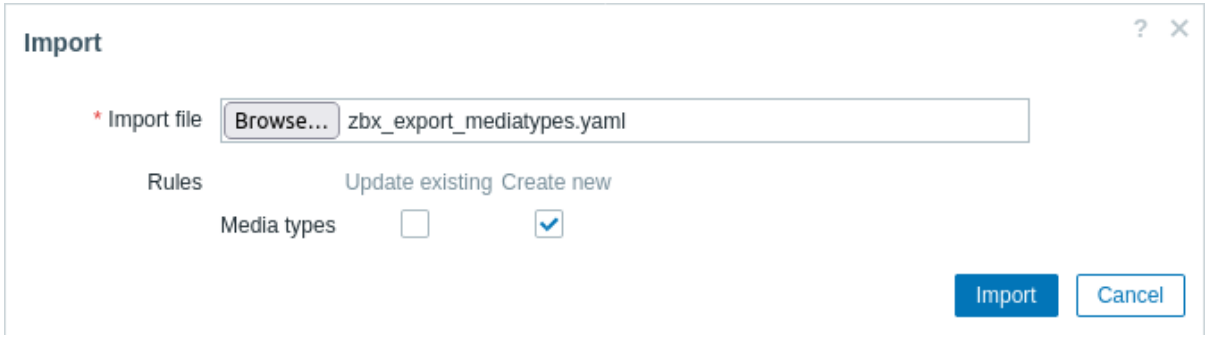
Dependendo do formato selecionado, os tipos de mídia são exportados para um arquivo local com um nome padrão:

- `zabbix_export_mediatypes.yaml` - na exportação YAML (opção padrão para exportação);
- `zabbix_export_mediatypes.xml` - na exportação XML;
- `zabbix_export_mediatypes.json` - na exportação JSON.

Importando

Para importar tipos de mídia, faça o seguinte:

1. Vá para *Alertas* → *Tipos de mídia*.
2. Clique em *Importar* no canto superior direito.
3. Selecione o arquivo de importação.
4. Marque as opções necessárias nas regras de importação.
5. Clique em *Importar* no canto inferior direito do formulário de configuração.



Regras de importação:

Regra	Descrição
Atualizar existentes	Os elementos existentes serão atualizados usando os dados do arquivo de importação. Caso contrário, eles não serão atualizados.
Criar novos	Novos elementos serão criados usando os dados do arquivo de importação. Caso contrário, eles não serão criados.

Uma mensagem de sucesso ou falha da importação será exibida no frontend.

Formato de exportação

Exportar para YAML:

```
zabbix_export:
  version: '7.0'
  media_types:
    - name: Pushover
      type: WEBHOOK
      parameters:
        - name: endpoint
          value: 'https://api.pushover.net/1/messages.json'
        - name: eventid
          value: '{EVENT.ID}'
        - name: event_nseverity
          value: '{EVENT.NSEVERITY}'
        - name: event_source
          value: '{EVENT.SOURCE}'
        - name: event_value
          value: '{EVENT.VALUE}'
        - name: expire
          value: '1200'
        - name: message
          value: '{ALERT.MESSAGE}'
        - name: priority_average
          value: '0'
        - name: priority_default
          value: '0'
        - name: priority_disaster
          value: '0'
        - name: priority_high
          value: '0'
        - name: priority_information
          value: '0'
        - name: priority_not_classified
          value: '0'
        - name: priority_warning
          value: '0'
        - name: retry
          value: '60'
        - name: title
          value: '{ALERT.SUBJECT}'
        - name: token
          value: '<PUSHOVER TOKEN HERE>'
        - name: triggerid
          value: '{TRIGGER.ID}'
        - name: url
          value: '{$ZABBIX.URL}'
        - name: url_title
          value: Zabbix
        - name: user
          value: '{ALERT.SENDTO}'
      status: DISABLED
      max_sessions: '0'
      script: |
        try {
            var params = JSON.parse(value),
                request = new HttpRequest(),
                data,
                response,
                severities = [
                    {name: 'not_classified', color: '#97AAB3'},
                    {name: 'information', color: '#7499FF'},
                    {name: 'warning', color: '#FFC859'},
                    {name: 'average', color: '#FFA059'},
                    {name: 'high', color: '#E97659'},
```

```
            {name: 'disaster', color: '#E45959'},
            {name: 'resolved', color: '#009900'},
            {name: 'default', color: '#000000'}
        ],
        priority;

    if (typeof params.HTTPProxy === 'string' && params.HTTPProxy.trim() !== '') {
        request.setProxy(params.HTTPProxy);
    }

    if ([0, 1, 2, 3].indexOf(parseInt(params.event_source)) === -1) {
        throw 'Incorrect "event_source" parameter given: "' + params.event_source + '".\nMust be 0
    }

    if (params.event_value !== '0' && params.event_value !== '1'
        && (params.event_source === '0' || params.event_source === '3')) {
        throw 'Incorrect "event_value" parameter given: ' + params.event_value + '\nMust be 0 or 1
    }

    if ([0, 1, 2, 3, 4, 5].indexOf(parseInt(params.event_nseverity)) === -1) {
        params.event_nseverity = '7';
    }

    if (params.event_value === '0') {
        params.event_nseverity = '6';
    }

    priority = params['priority_' + severities[params.event_nseverity].name] || params.priority_de

    if (isNaN(priority) || priority < -2 || priority > 2) {
        throw '"priority" should be -2..2';
    }

    if (params.event_source === '0' && isNaN(params.triggerid)) {
        throw 'field "triggerid" is not a number';
    }

    if (isNaN(params.eventid)) {
        throw 'field "eventid" is not a number';
    }

    if (typeof params.message !== 'string' || params.message.trim() === '') {
        throw 'field "message" cannot be empty';
    }

    data = {
        token: params.token,
        user: params.user,
        title: params.title,
        message: params.message,
        url: (params.event_source === '0')
            ? params.url + '/tr_events.php?triggerid=' + params.triggerid + '&eventid=' + params.e
            : params.url,
        url_title: params.url_title,
        priority: priority
    };

    if (priority == 2) {
        if (isNaN(params.retry) || params.retry < 30) {
            throw 'field "retry" should be a number with value of at least 30 if "priority" is set
        }
```

```
            if (isNaN(params.expire) || params.expire > 10800) {
                throw 'field "expire" should be a number with value of at most 10800 if "priority" is
            }

            data.retry = params.retry;
            data.expire = params.expire;
        }

        data = JSON.stringify(data);
        Zabbix.log(4, '[ Pushover Webhook ] Sending request: ' + params.endpoint + '\n' + data);

        request.addHeader('Content-Type: application/json');
        response = request.post(params.endpoint, data);

        Zabbix.log(4, '[ Pushover Webhook ] Received response with status code ' + request.getStatus()

        if (response !== null) {
            try {
                response = JSON.parse(response);
            }
            catch (error) {
                Zabbix.log(4, '[ Pushover Webhook ] Failed to parse response received from Pushover');
                response = null;
            }
        }

        if (request.getStatus() != 200 || response === null || typeof response !== 'object' || respons
            if (response !== null && typeof response === 'object' && typeof response.errors === 'objec
                    && typeof response.errors[0] === 'string') {
                throw response.errors[0];
            }
            else {
                throw 'Unknown error. Check debug log for more information.';
            }
        }

        return 'OK';
    }
    catch (error) {
        Zabbix.log(4, '[ Pushover Webhook ] Pushover notification failed: ' + error);
        throw 'Pushover notification failed: ' + error;
    }
description: |
    Consulte o guia de configuração aqui: https://git.zabbix.com/projects/ZBX/repos/zabbix/browse/temp

    Defina o parâmetro token com a chave do seu aplicativo Pushover.
    Ao atribuir o meio Pushover ao usuário do Zabbix - adicione a chave do usuário no campo enviar par
message_templates:
    - event_source: TRIGGERS
      operation_mode: PROBLEM
      subject: 'Problema: {EVENT.NAME}'
      message: |
        Problema iniciado às {EVENT.TIME} em {EVENT.DATE}
        Nome do problema: {EVENT.NAME}
        Host: {HOST.NAME}
        Severidade: {EVENT.SEVERITY}
        Dados operacionais: {EVENT.OPDATA}
        ID do problema original: {EVENT.ID}
        {TRIGGER.URL}
    - event_source: TRIGGERS
      operation_mode: RECOVERY
      subject: 'Resolvido em {EVENT.DURATION}: {EVENT.NAME}'
```

```
      message: |
        O problema foi resolvido às {EVENT.RECOVERY.TIME} em {EVENT.RECOVERY.DATE}
        Nome do problema: {EVENT.NAME}
        Duração do problema: {EVENT.DURATION}
        Host: {HOST.NAME}
        Severidade: {EVENT.SEVERITY}
        ID do problema original: {EVENT.ID}
        {TRIGGER.URL}
    - event_source: TRIGGERS
      operation_mode: UPDATE
      subject: 'Problema atualizado em {EVENT.AGE}: {EVENT.NAME}'
      message: |
        {USER.FULLNAME} {EVENT.UPDATE.ACTION} problema em {EVENT.UPDATE.DATE} {EVENT.UPDATE.TIME}.
        {EVENT.UPDATE.MESSAGE}

        O status atual do problema é {EVENT.STATUS}, idade é {EVENT.AGE}, reconhecido: {EVENT.ACK.STAT
    - event_source: DISCOVERY
      operation_mode: PROBLEM
      subject: 'Descoberta: {DISCOVERY.DEVICE.STATUS} {DISCOVERY.DEVICE.IPADDRESS}'
      message: |
        Regra de descoberta: {DISCOVERY.RULE.NAME}

        IP do dispositivo: {DISCOVERY.DEVICE.IPADDRESS}
        DNS do dispositivo: {DISCOVERY.DEVICE.DNS}
        Status do dispositivo: {DISCOVERY.DEVICE.STATUS}
        Uptime do dispositivo: {DISCOVERY.DEVICE.UPTIME}

        Nome do serviço do dispositivo: {DISCOVERY.SERVICE.NAME}
        Porta do serviço do dispositivo: {DISCOVERY.SERVICE.PORT}
        Status do serviço do dispositivo: {DISCOVERY.SERVICE.STATUS}
        Uptime do serviço do dispositivo: {DISCOVERY.SERVICE.UPTIME}
    - event_source: AUTOREGISTRATION
      operation_mode: PROBLEM
      subject: 'Autoregistro: {HOST.HOST}'
      message: |
        Nome do host: {HOST.HOST}
        IP do host: {HOST.IP}
        Porta do agent: {HOST.PORT}
```

Elementos exportados

Os elementos exportados são explicados na tabela abaixo.

Elemento	Tipo	Descrição
name	string	(obrigatório) Nome do tipo de mídia.
type	string	(obrigatório) Transporte usado pelo tipo de mídia. Valores possíveis:[1] EMAIL (0), SMS (1), SCRIPT (2), WEBHOOK (4).
status	string	Se o tipo de mídia está habilitado. Valores possíveis:[1] ENABLED (0, padrão), DISABLED (1)
max_sessions	integer	O número máximo de alertas que podem ser processados em paralelo. Valores possíveis para SMS:[1] 1 (padrão). Valores possíveis para outros tipos de mídia:[1] 0-100 (onde 0 - ilimitado).
attempts	integer	O número máximo de tentativas para enviar um alerta. Valores possíveis:[1] 1-10 (padrão: 3).
attempt_interval	string	O intervalo (usando segundos ou sufixo de tempo) entre as tentativas de repetição. Valores possíveis:[1] 0-60s (padrão: 10s).
description	string	Descrição do tipo de mídia.
message_templates		Elemento raiz para modelos de mensagem do tipo de mídia.
event_source	string	(obrigatório) Fonte do evento. Valores possíveis:[1] TRIGGERS (0), DISCOVERY (1), AUTOREGISTRATION (2), INTERNAL (3), SERVICE (4).
operation_mode	string	Modo de operação. Valores possíveis:[1] PROBLEM (0), RECOVERY (1), UPDATE (2).

Elemento	Tipo	Descrição
subject	string	Assunto da mensagem.
message	string	Corpo da mensagem.

Note:
Veja também: Objeto media type (consulte a propriedade relevante com o nome correspondente).

Email

Os seguintes elementos adicionais são exportados apenas para o tipo de mídia *Email*.

Elemento	Tipo	Descrição
provider	string	Provedor de email.
smtp_server	string	Servidor SMTP.
smtp_port	integer	Porta do servidor SMTP para conectar. Padrão: 25.
smtp_helo	string	SMTP helo.
smtp_email	string	Endereço de email a partir do qual as notificações serão enviadas.
smtp_security	string	Nível de segurança da conexão SMTP a ser usado. Valores possíveis:[1] NONE (0, padrão), STARTTLS (1), SSL_OR_TLS (2).
smtp_verify_host	string	Verificação de host SSL para SMTP. Valores possíveis:[1] NO (0, padrão), YES (1).
smtp_verify_peer	string	Verificação de peer SSL para SMTP. Valores possíveis:[1] NO (0, padrão), YES (1).
smtp_authentication	string	Método de autenticação SMTP a ser usado. Valores possíveis:[1] NONE (0, padrão), PASSWORD (1).
username	string	Nome de usuário.
password	string	Senha de autenticação.
message_format	string	Formato da mensagem. Valores possíveis:[1] TEXT (0), HTML (1, padrão).

Note:
Veja também: Objeto Media type (consulte a propriedade relevante com o nome correspondente).

SMS

Os seguintes elementos adicionais são exportados apenas para o tipo de mídia *SMS*.

Elemento	Tipo	Descrição
gsm_modem	string	(obrigatório) Nome do dispositivo serial do modem GSM.

Note:
Veja também: Objeto de tipo de mídia (consulte a propriedade relevante com o nome correspondente).

Script

Os seguintes elementos adicionais são exportados apenas para o tipo de mídia *Script*.

Elemento	Tipo	Descrição
script name	string	(obrigatório) Nome do script.
parameters		Elemento raiz para parâmetros do script.
sortorder	string	(obrigatório) Ordem dos parâmetros do script passados para o script como argumentos de linha de comando.
value	string	Valor do parâmetro do script.

Note:
Veja também: Objeto media type (consulte a propriedade relevante com o mesmo nome).

Webhook

Os seguintes elementos adicionais são exportados apenas para o tipo de mídia *Webhook*.

Elemento	Tipo	Descrição
script	string	Script.
timeout	string	Intervalo de tempo limite para requisição HTTP do script Javascript. Valores possíveis:[1] 1-60s (padrão: 30s).
process_tags	string	Se as tags retornadas devem ser processadas. Valores possíveis:[1] NO (0, padrão), YES (1).
show_event_menu	string	Indica a presença de uma entrada no menu de eventos se a macro {EVENT.TAGS.*} foi resolvida com sucesso nos campos `event_menu_url` e `event_menu_name`. Valores possíveis:[1] NO (0, padrão), YES (1).
event_menu_url	string	URL da entrada do menu de eventos. Suporta a macro {EVENT.TAGS.*}.
event_menu_name	string	Nome da entrada do menu de eventos. Suporta a macro {EVENT.TAGS.*}.
parameters		Elemento raiz para parâmetros do tipo de mídia webhook.
name	string	(obrigatório) Nome do parâmetro do webhook.
value	string	Valor do parâmetro do webhook.

Note:
Veja também: Objeto media type (consulte a propriedade relevante com o nome correspondente).

Notas de rodapé

[1] Os valores inteiros da API entre colchetes, por exemplo, ENABLED (0), são mencionados apenas para referência. Para obter mais informações, consulte a página do objeto da API vinculada na entrada da tabela ou no final de cada seção.

15. Descoberta

Use a barra lateral para acessar o conteúdo da seção Descoberta.

1 descoberta de rede

Visão geral

O Zabbix oferece uma funcionalidade de descoberta automática de rede que é eficaz e muito flexível.

Com a descoberta de rede devidamente configurada, você pode:

- acelerar a implantação do Zabbix
- simplificar a administração
- usar o Zabbix em ambientes que mudam rapidamente sem administração excessiva

A descoberta de rede do Zabbix é baseada nas seguintes informações:

- Intervalos de IP
- Disponibilidade de serviços externos (FTP, SSH, WEB, POP3, IMAP, TCP, etc)
- Informações recebidas do agent do Zabbix (apenas o modo não criptografado é suportado)
- Informações recebidas do agent SNMP

Ela NÃO fornece:

- Descoberta da topologia da rede

A descoberta de rede basicamente consiste em duas fases: descoberta e ações.

Descoberta

O Zabbix periodicamente escaneia os intervalos de IP definidos nas regras de descoberta de rede. A frequência da verificação é configurável para cada regra individualmente.

Cada regra tem um conjunto de verificações de serviço definidas para serem realizadas no intervalo de IP.

As regras de descoberta são processadas pelo gerenciador de descoberta. O gerenciador de descoberta cria um trabalho para cada regra com uma lista de tarefas (verificações de rede). As verificações de rede são realizadas em paralelo pelos workers de descoberta disponíveis (o número é configurável no frontend para cada regra). Apenas verificações com o mesmo IP e porta são agendadas sequencialmente porque alguns dispositivos não aceitam conexões paralelas na mesma porta.

O tamanho da fila de verificações de rede é limitado a 2.000.000 ou aproximadamente 4 GB de memória. Se a fila ficar cheia, a regra de descoberta será ignorada e uma mensagem de aviso será impressa no log. Você pode usar o item interno `zabbix[discovery_queue]` para monitorar o número de verificações de descoberta na fila.

As verificações de descoberta são processadas independentemente das outras verificações. Se alguma verificação não encontrar um serviço (ou falhar), as outras verificações ainda serão processadas.

Note:
Se uma regra de descoberta for alterada durante a execução, a execução atual da descoberta será abortada.

Cada verificação de um serviço e de um host (IP) realizada pelo módulo de descoberta de rede gera um evento de descoberta.

Evento	Resultado da verificação do serviço
Serviço Descoberto	O serviço está 'ativo' após estar 'inativo' ou quando descoberto pela primeira vez.
Serviço Ativo	O serviço está 'ativo', após já estar 'ativo'.
Serviço Perdido	O serviço está 'inativo' após estar 'ativo'.
Serviço Inativo	O serviço está 'inativo', após já estar 'inativo'.
Host Descoberto	Pelo menos um serviço de um host está 'ativo' após todos os serviços desse host estarem 'inativos' ou um serviço é descoberto que pertence a um host não registrado.
Host Ativo	Pelo menos um serviço de um host está 'ativo', após pelo menos um serviço já estar 'ativo'.
Host Perdido	Todos os serviços de um host estão 'inativos' após pelo menos um estar 'ativo'.
Host Inativo	Todos os serviços de um host estão 'inativos', após já estarem 'inativos'.

Ações

Eventos de descoberta podem ser a base para ações relevantes, tais como:

- Enviar notificações
- Adicionar/remover hosts
- Habilitar/desabilitar hosts
- Adicionar hosts a um grupo
- Remover hosts de um grupo
- Adicionar tags a um host
- Remover tags de um host
- Vincular um template a hosts/desvincular um template de hosts
- Executar scripts remotos

Essas ações podem ser configuradas com base no tipo de dispositivo, IP, status, tempo de atividade/inatividade, etc. Para detalhes completos sobre como configurar ações para eventos baseados em descoberta de rede, consulte as páginas de operações e condições de ação.

Como as ações de descoberta de rede são baseadas em eventos, elas serão acionadas tanto quando um host descoberto estiver online quanto quando estiver offline. É altamente recomendado adicionar uma condição de ação *Discovery status: up* para evitar que ações como *Add host* sejam acionadas em eventos *Service Lost*/*Service Down*. Caso contrário, se um host descoberto for removido manualmente, ele ainda gerará eventos *Service Lost*/*Service Down* e será recriado durante o próximo ciclo de descoberta.

Note:
Vincular templates a um host descoberto falhará coletivamente se qualquer um dos templates vinculáveis tiver uma entidade única (por exemplo, chave de item) que seja igual a uma entidade única (por exemplo, chave de item) já existente no host ou em outro dos templates vinculáveis.

Criação de host

Um host é adicionado se a operação *Adicionar host* for selecionada. Um host também é adicionado, mesmo que a operação *Adicionar host* esteja ausente, se você selecionar operações que resultem em ações em um host. Essas operações são:

- habilitar host
- desabilitar host
- adicionar host a um grupo de hosts
- vincular template a um host

Os hosts criados são adicionados ao grupo *Hosts descobertos* (por padrão, configurável em *Administração* > *Geral* > *Outros*). Se você deseja que os hosts sejam adicionados a outro grupo, adicione uma operação *Remover de grupos de hosts* (especificando "Hosts descobertos") e também adicione uma operação *Adicionar a grupos de hosts* (especificando outro grupo de hosts), porque um host deve pertencer a um grupo de hosts.

O endereço IP do dispositivo descoberto, juntamente com a fonte da descoberta (Zabbix server, Zabbix proxy ou grupo de proxy) e o tipo de interface, é usado como critério para encontrar um host no sistema. Se já existir um host com o mesmo endereço IP, tipo de interface e fonte de descoberta, esse host será o alvo para a execução das operações. Se a fonte de descoberta for diferente, a entidade descoberta é tratada como um host diferente e um novo host pode ser criado.

Se o endereço IP do host descoberto for alterado ou a interface for excluída, um novo host será criado na próxima descoberta.

Nomeação de hosts

Ao adicionar hosts, o nome do host é o resultado da pesquisa reversa de DNS ou do endereço IP se a pesquisa reversa falhar. A pesquisa é realizada a partir do Zabbix server ou Zabbix proxy, dependendo de qual está realizando a descoberta. Se a pesquisa falhar no proxy, ela não será repetida no server. Se já existir um host com esse nome, o próximo host terá **_2** anexado ao nome, depois **_3** e assim por diante.

Também é possível substituir a pesquisa de DNS/IP e, em vez disso, usar o valor de um item para o nome do host, por exemplo:

- Você pode descobrir vários servidores com o Zabbix agent em execução usando um item do Zabbix agent para descoberta e atribuir nomes apropriados a eles automaticamente, com base no valor da string retornada por esse item
- Você pode descobrir vários dispositivos de rede SNMP usando um item do SNMP agent para descoberta e atribuir nomes apropriados a eles automaticamente, com base no valor da string retornada por esse item

Se o nome do host foi definido usando o valor de um item, ele não é atualizado durante as verificações de descoberta subsequentes. Se não for possível definir o nome do host usando o valor de um item, o valor padrão (nome DNS) será usado.

Se já existir um host com o endereço IP descoberto e a fonte de descoberta (Zabbix server, proxy ou grupo de proxy) não tiver sido alterada, um novo host não será criado. Se a fonte de descoberta for diferente, a entidade descoberta é tratada como distinta e um novo host pode ser criado. No entanto, se a ação de descoberta contiver operações (vincular template, adicionar ao grupo de hosts, etc.), elas serão executadas no host existente que corresponde pelo endereço IP, tipo de interface e fonte de descoberta.

Remoção de host

Hosts descobertos por uma regra de descoberta de rede são removidos automaticamente de *Monitoramento* > *Descoberta* se uma entidade descoberta não estiver mais no intervalo de IPs da regra. Os hosts são removidos imediatamente.

Criação de interface ao adicionar hosts

Quando hosts são adicionados como resultado da descoberta de rede, eles recebem interfaces criadas de acordo com estas regras:

- Os serviços detectados - por exemplo, se uma verificação SNMP for bem-sucedida, uma interface SNMP será criada.
- Se um host respondeu tanto a solicitações do agent Zabbix quanto a solicitações SNMP, ambos os tipos de interfaces serão criados.
- Se os critérios de exclusividade forem dados retornados pelo agent Zabbix ou SNMP, a primeira interface encontrada para um host será criada como a interface padrão. Outros endereços IP serão adicionados como interfaces adicionais. As condições da ação (como IP do host) não afetam a adição de interfaces. *Observe* que isso funcionará se todas as interfaces forem descobertas pela mesma regra de descoberta. Se uma regra de descoberta diferente descobrir uma interface diferente do mesmo host, um host adicional será adicionado.
- Se um host respondeu apenas a verificações do agent, ele será criado apenas com uma interface agent. Se ele começar a responder ao SNMP posteriormente, interfaces SNMP adicionais serão adicionadas.
- Se 3 hosts separados foram inicialmente criados, tendo sido descobertos pelo critério de exclusividade "IP", e então a regra de descoberta for modificada para que os hosts A, B e C tenham resultados idênticos de critério de exclusividade, B e C são criados como interfaces adicionais para A, o primeiro host. Os hosts individuais B e C permanecem. Em *Monitoramento* > *Descoberta*, as interfaces adicionadas serão exibidas na coluna "Dispositivo descoberto", em fonte preta e indentadas, mas a coluna "Host monitorado" exibirá apenas A, o primeiro host criado. "O tempo de atividade/inatividade" não é medido para IPs que são considerados interfaces adicionais.

Alterando a configuração do proxy

Os hosts descobertos por diferentes proxies nem sempre são tratados como hosts diferentes. A descoberta e as verificações de exclusividade dependem da estrutura do grupo de proxies: quando um proxy executa uma regra de descoberta e cria um host, esse host é adicionado ao grupo de proxy pai do proxy, não atribuído ao próprio proxy. Quando o Zabbix avalia a exclusividade do IP durante a descoberta, ele verifica os hosts monitorados pelo grupo de proxy pai. Os hosts monitorados por proxies individuais

dentro desse grupo (incluindo o proxy que executou a descoberta) são ignorados para a verificação de exclusividade, o que pode resultar em hosts duplicados se vários proxies monitorarem sub-redes sobrepostas.

Embora esse comportamento permita que a descoberta opere em intervalos de IP sobrepostos usados por diferentes sub-redes, alterar o proxy atribuído a uma sub-rede já monitorada é mais complicado porque as alterações de proxy devem ser aplicadas de forma consistente aos hosts descobertos e à associação do grupo de proxy pai para evitar duplicatas.

Por exemplo, os passos para substituir o proxy em uma regra de descoberta:

1. desabilite a regra de descoberta
2. sincronize a configuração do proxy
3. substitua o proxy na regra de descoberta
4. substitua o proxy para todos os hosts descobertos por esta regra (garanta que os hosts no grupo de proxy pai e quaisquer hosts monitorados por proxies individuais nesse grupo sejam atualizados para evitar duplicatas)
5. habilite a regra de descoberta

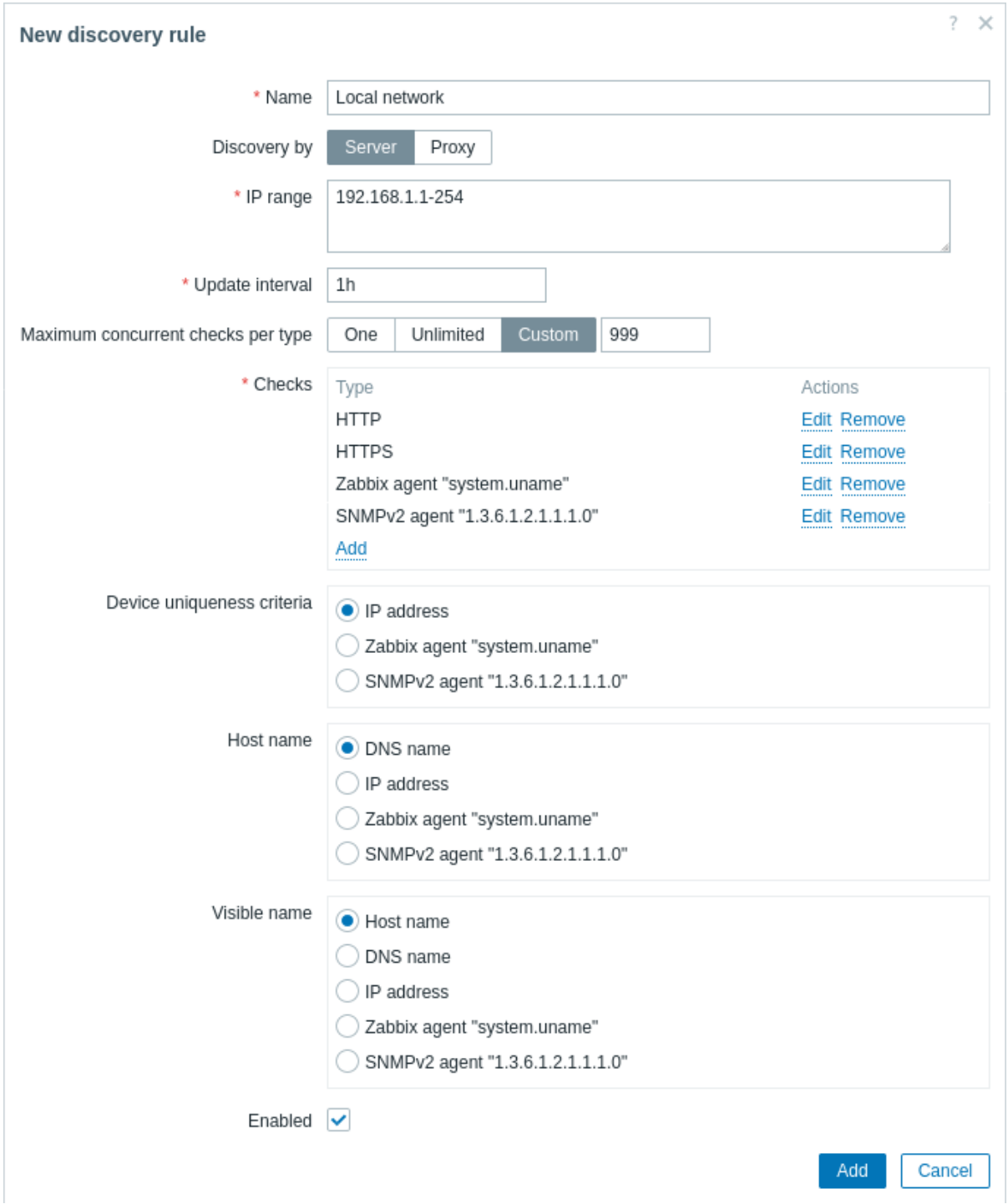
1 Configurando uma regra de descoberta de rede

Visão geral

Para configurar uma regra de descoberta de rede usada pelo Zabbix para descobrir hosts e serviços:

- Vá para *Coleta de dados* → *Descoberta*
- Clique em *Criar regra de descoberta* (ou no nome da regra para editar uma existente)
- Edite os atributos da regra de descoberta

Atributos da regra



Todos os campos obrigatórios estão marcados com um asterisco vermelho.

Parâmetro	Descrição
Nome	Nome exclusivo da regra. Por exemplo, ”Rede local”.
Descoberta por	A descoberta é realizada por: **Servidor** - pelo Zabbix server **Proxy** - pelo Zabbix proxy (selecionado no campo de nome do proxy)
Intervalo de IP	O intervalo de endereços IP para descoberta. Pode ter os seguintes formatos: IP único: 192.168.1.33 Intervalo de endereços IP: 192.168.1-10.1-255. O intervalo é limitado pelo número total de endereços cobertos (menos de 64K). Máscara de IP: 192.168.4.0/24 máscaras de IP suportadas: /16 - /30 para endereços IPv4 /112 - /128 para endereços IPv6 Lista: 192.168.1.1-255, 192.168.2.1-100, 192.168.2.200, 192.168.4.0/24 Este campo suporta espaços, tabulação e múltiplas linhas.

Parâmetro	Descrição
Intervalo de atualização	Este parâmetro define com que frequência o Zabbix executará a regra. O intervalo é medido após a execução da instância de descoberta anterior terminar, para que não haja sobreposição. Sufixos de tempo são suportados, por exemplo, 30s, 1m, 2h, 1d. Macros de usuário são suportadas. *Observe* que se uma macro de usuário for usada e seu valor for alterado (por exemplo, 1w → 1h), a próxima verificação será executada de acordo com o valor anterior (muito no futuro com os valores de exemplo).
Máximo de verificações simultâneas por tipo	Defina o número máximo de threads de descoberta (workers) por verificação de serviço para processamento paralelo das verificações de descoberta: **Um** - um thread **Ilimitado** - número ilimitado de threads (mas não mais do que no parâmetro StartDiscoverers) **Personalizado** - defina um número personalizado de threads (0-999) Observe que todas as regras de descoberta com verificações de serviço SNMPv3 assíncronas são processadas por um worker devido às peculiaridades da implementação do libsnmp, ou seja, aumentar o número de workers não aumentará a velocidade da descoberta.
Verificações	O Zabbix usará esta lista de verificações para descoberta. Clique em Add para configurar uma nova verificação em uma janela pop-up. Verificações suportadas: SSH, LDAP, SMTP, FTP, HTTP, HTTPS, POP, NNTP, IMAP, TCP, Telnet, Zabbix agent, SNMPv1 agent, SNMPv2 agent, SNMPv3 agent, ICMP ping. Uma descoberta baseada em protocolo usa a funcionalidade **net.tcp.service[]** para testar cada host, exceto para SNMP, que consulta um OID SNMP. O Zabbix agent é testado consultando um item em modo não criptografado. Consulte itens de agent para mais detalhes. O parâmetro 'Portas' pode ser um dos seguintes: Porta única: 22 Intervalo de portas: 22-45 Lista: 22-45,55,60-70 Desde o Zabbix 7.0, todas as verificações de serviço são realizadas de forma assíncrona, exceto as verificações LDAP. Desde o Zabbix 7.0, a verificação HTTP/HTTPs é feita via libcurl. Se o Zabbix server/proxy for compilado sem libcurl, as verificações HTTP funcionarão como nas versões anteriores (ou seja, como verificações TCP), mas as verificações HTTPS não funcionarão.
Critérios de exclusividade do dispositivo	Os critérios de exclusividade podem ser: **Endereço IP** - não processe vários dispositivos de IP único. Se um dispositivo com o mesmo IP já existir, ele será considerado já descoberto e um novo host não será adicionado. **<verificação de descoberta>** - verificação do Zabbix agent ou do SNMP agent. Observe que os critérios de exclusividade usados durante a descoberta não são os mesmos que a identificação do host no sistema ao executar ações. Os critérios de exclusividade durante a descoberta definem se dois ou mais dispositivos descobertos são iguais (ou diferentes), enquanto apenas o endereço IP é o critério para identificação do host no Zabbix (consulte Criação de host).
Nome do host	Defina o nome técnico do host criado usando: **Nome DNS** - nome DNS (padrão) **Endereço IP** - endereço IP **<verificação de descoberta>** - valor de string recebido da verificação de descoberta (por exemplo, Zabbix agent, verificação do SNMP agent) Veja também: Nomeação de host.
Nome visível	Defina o nome visível do host criado usando: **Nome do host** - nome técnico do host (padrão) **Nome DNS** - nome DNS **Endereço IP** - endereço IP **<verificação de descoberta>** - valor de string recebido da verificação de descoberta (por exemplo, Zabbix agent, verificação do SNMP agent) Veja também: Nomeação de host.
Habilitado	Com a caixa de seleção marcada, a regra está ativa e será executada pelo Zabbix server. Se desmarcada, a regra não está ativa. Não será executada.

Excedendo o limite de descritores de arquivos

No caso de um grande número de verificações simultâneas, é possível esgotar o limite de descritores de arquivos para o gerenciador de descoberta.

O número de descritores de arquivos necessários para a detecção equivale ao número de discovery workers * 1000. Por padrão, existem 5 discovery workers, enquanto o limite soft do sistema é de aproximadamente 1024.

Se esse limite for atingido, o Zabbix reduzirá o número padrão de verificações simultâneas por tipo para cada worker e escreverá um aviso no arquivo de log. Mas, se o usuário definiu um valor maior para *Verificações simultâneas máximas por tipo* do que o valor calculado pelo Zabbix, o Zabbix usará o valor definido pelo usuário para um worker.

Um cenário da vida real

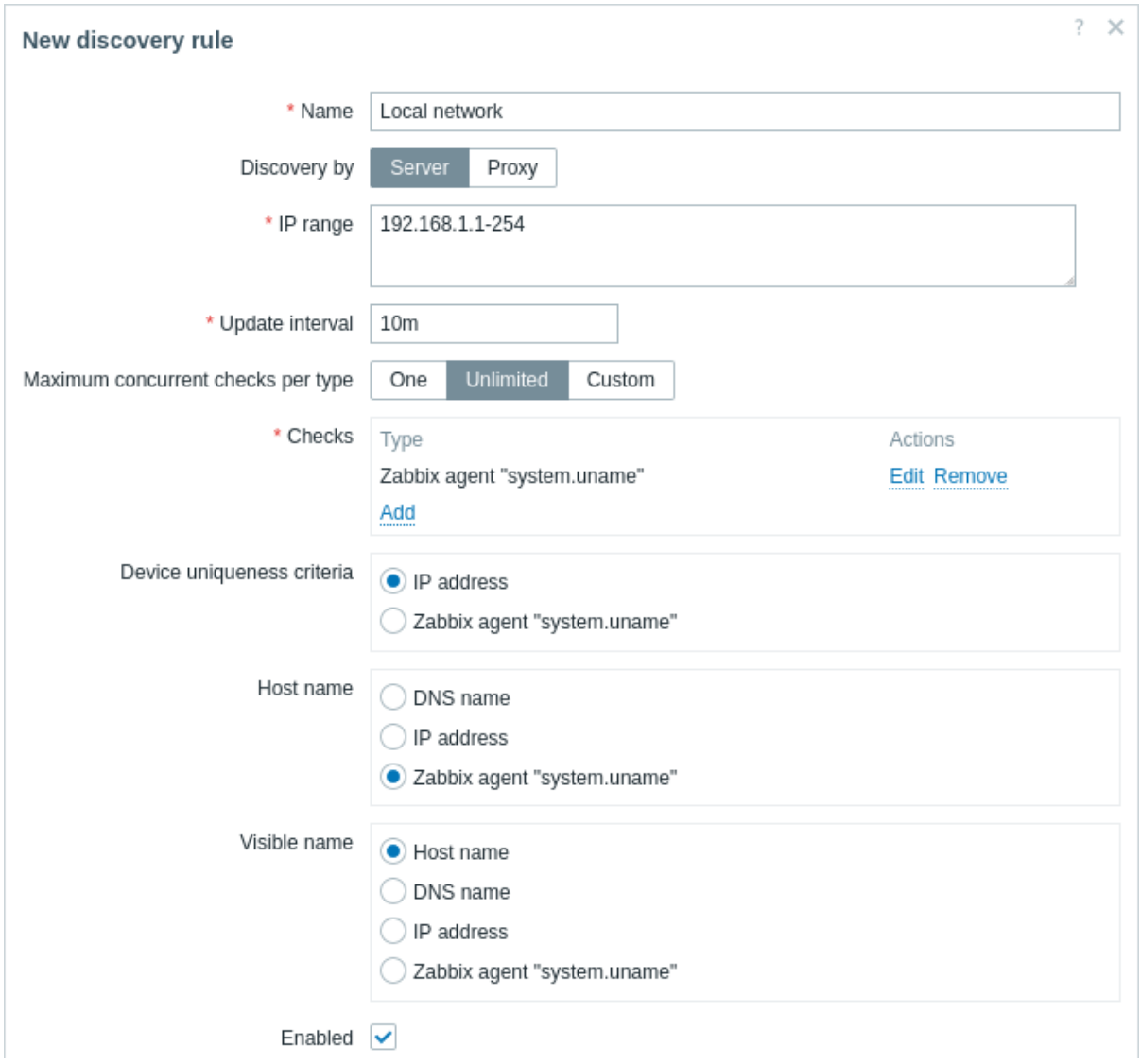
Neste exemplo, gostaríamos de configurar a descoberta de rede para o local rede com um intervalo de IP de 192.168.1.1-192.168.1.254.

Em nosso cenário queremos:

- descubra os hosts que possuem o agente Zabbix em execução
- execute a descoberta a cada 10 minutos
- adicionar um host ao monitoramento se o tempo de atividade do host for superior a 1 hora
- remover hosts se o tempo de inatividade do host for superior a 24 horas
- adicionar hosts Linux ao grupo ”servidores Linux”
- adicionar hosts Windows ao grupo ”servidores Windows”
- use o modelo *Linux* para hosts Linux
- use o modelo *Windows* para hosts Windows

Passo 1

Definindo uma regra de descoberta de rede para nosso intervalo de IPs.



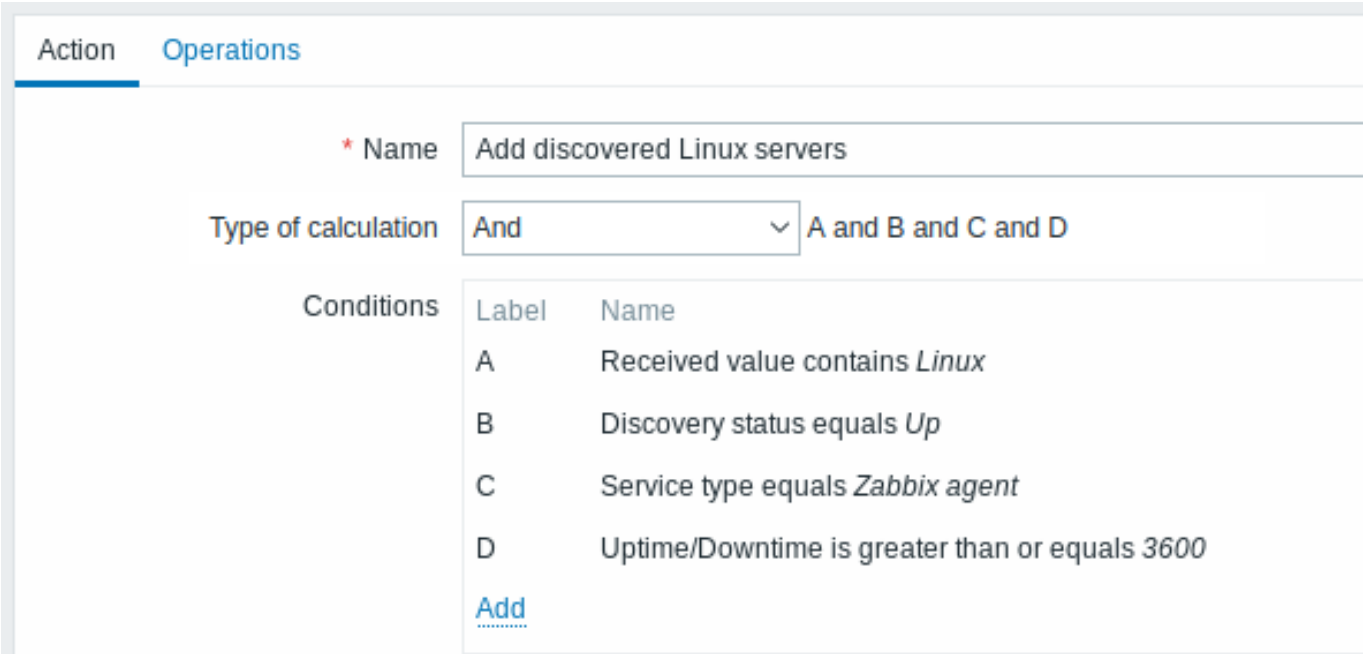
O Zabbix tentará descobrir hosts no intervalo de IPs 192.168.1.1-192.168.1.254 conectando-se aos agents do Zabbix e obtendo o valor da chave **system.uname**. O valor recebido do agent pode ser usado para nomear os hosts e também para aplicar diferentes ações para diferentes sistemas operacionais. Por exemplo, vincular servidores Windows ao template *Windows*, servidores Linux ao template *Linux*.

A regra será executada a cada 10 minutos.

Quando esta regra for adicionada, o Zabbix iniciará automaticamente a descoberta e a geração de eventos baseados em descoberta para processamento posterior.

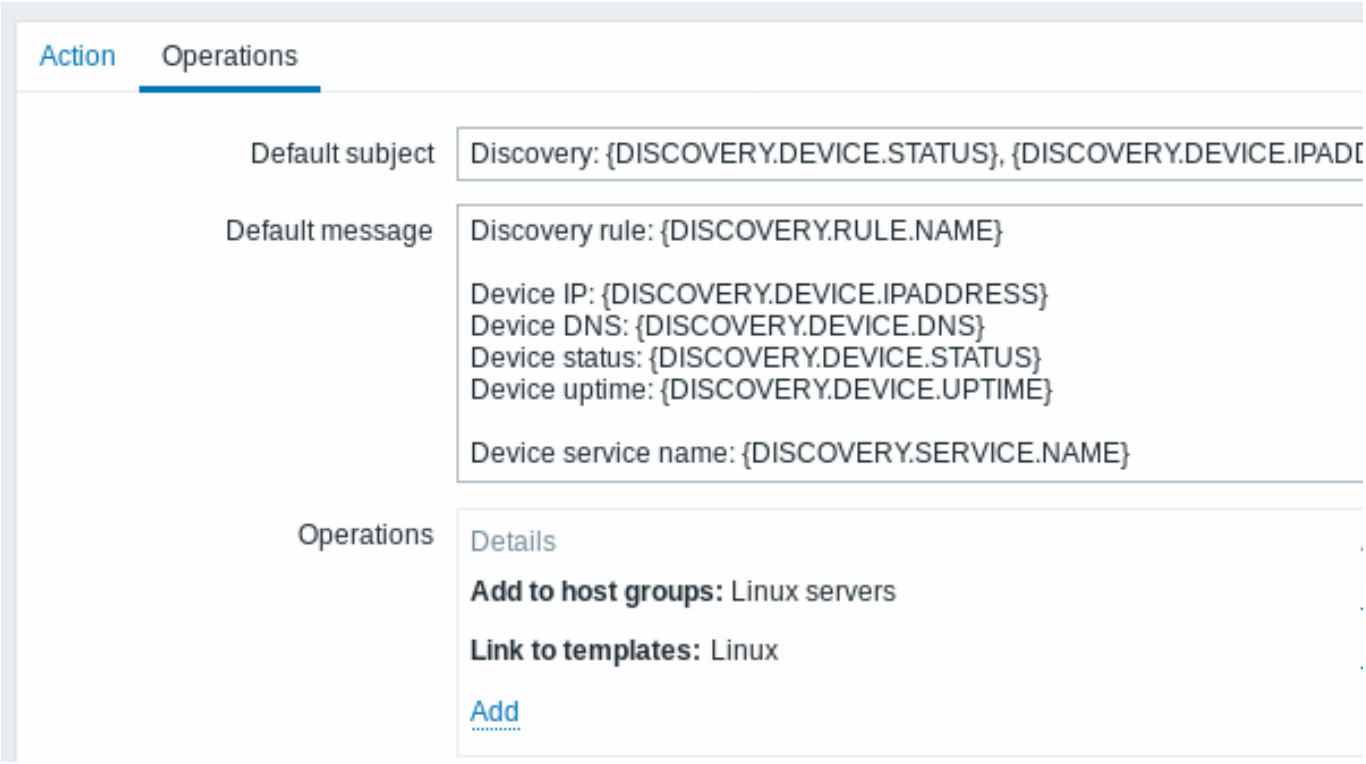
Passo 2

Definindo uma ação de descoberta para adicionando os servidores Linux descobertos ao respectivo grupo/modelo.



A ação será ativada se:

- o serviço ”agente Zabbix” está ”ativo”
- o valor de system.uname (a chave do agente Zabbix que usamos na regra definição) contém ”Linux”
- O tempo de atividade é de 1 hora (3600 segundos) ou mais

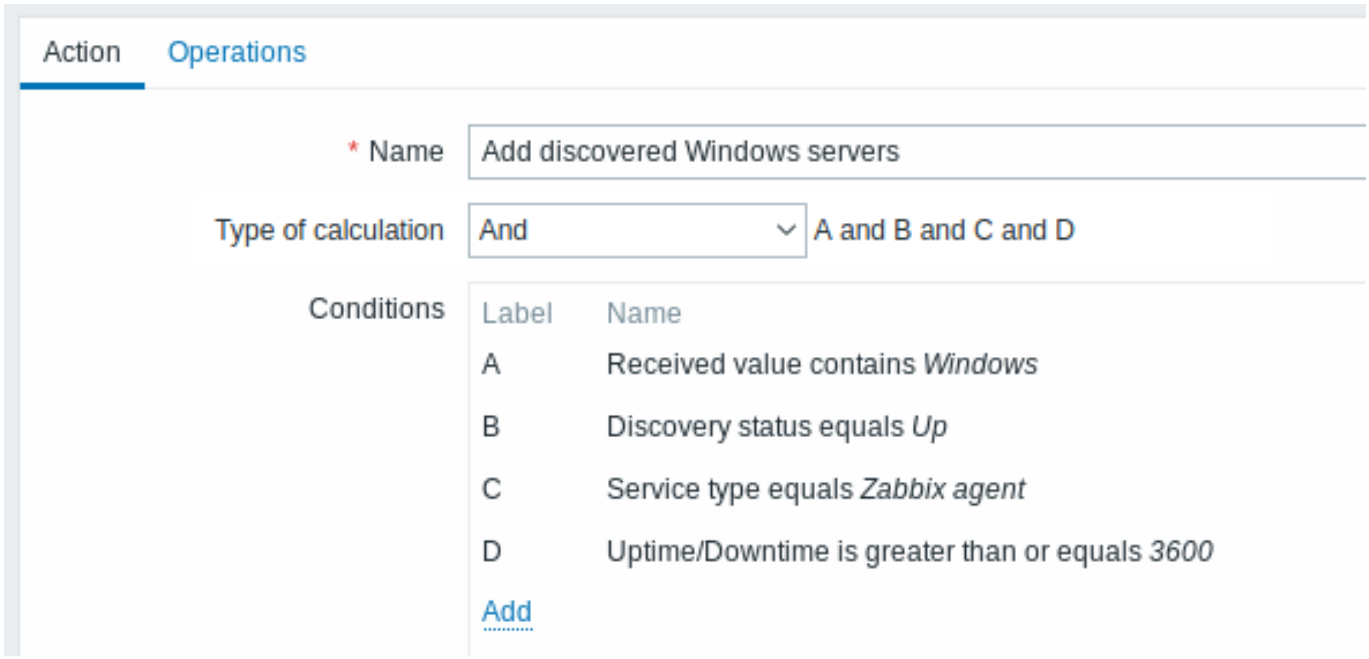


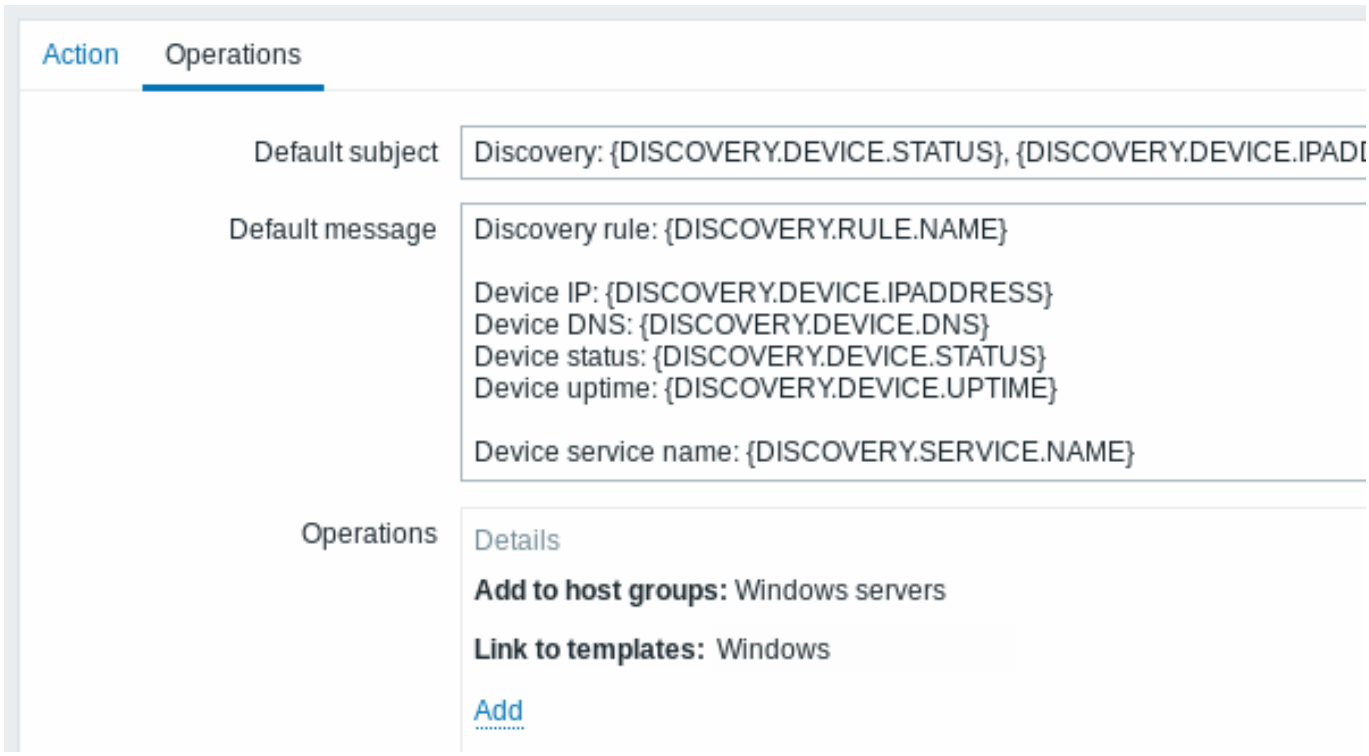
A ação executará as seguintes operações:

- adicione o host descoberto ao grupo ”servidores Linux” (e também adicione host se não foi adicionado anteriormente)
- vincule o host ao modelo *Linux*. Zabbix irá iniciar automaticamente monitorando o host usando itens e gatilhos do ”Linux” modelo.

Etapa 3

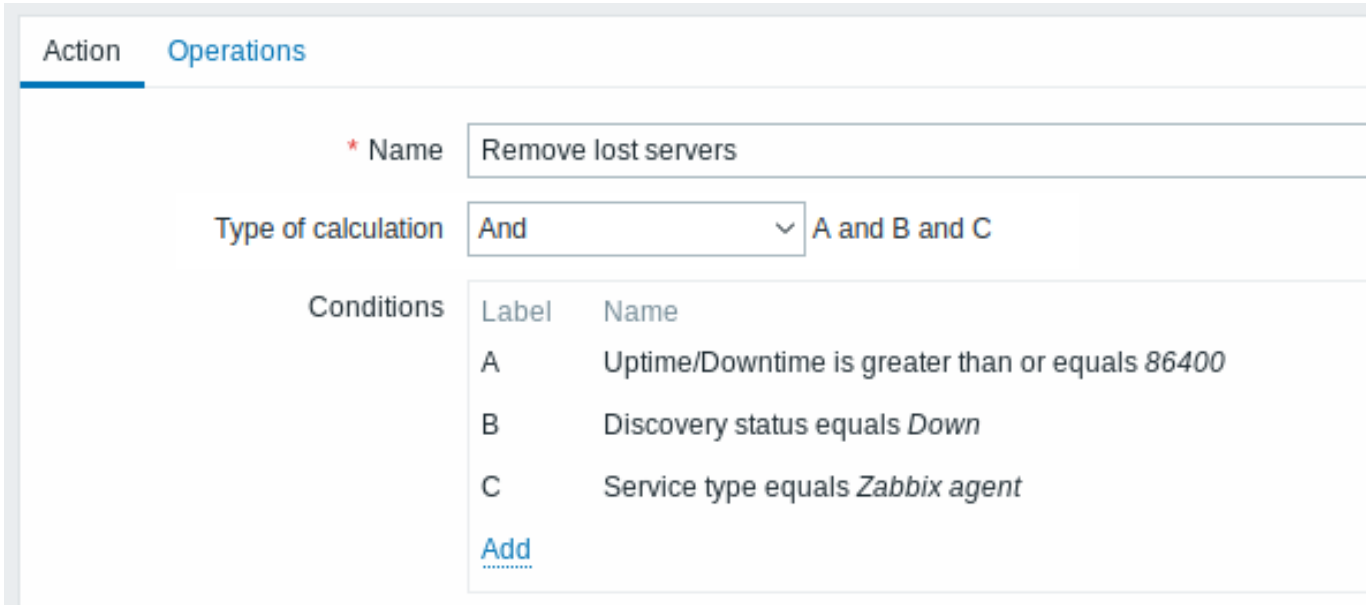
Definir uma ação de descoberta para adicionar os servidores Windows descobertos a o respectivo grupo/modelo.

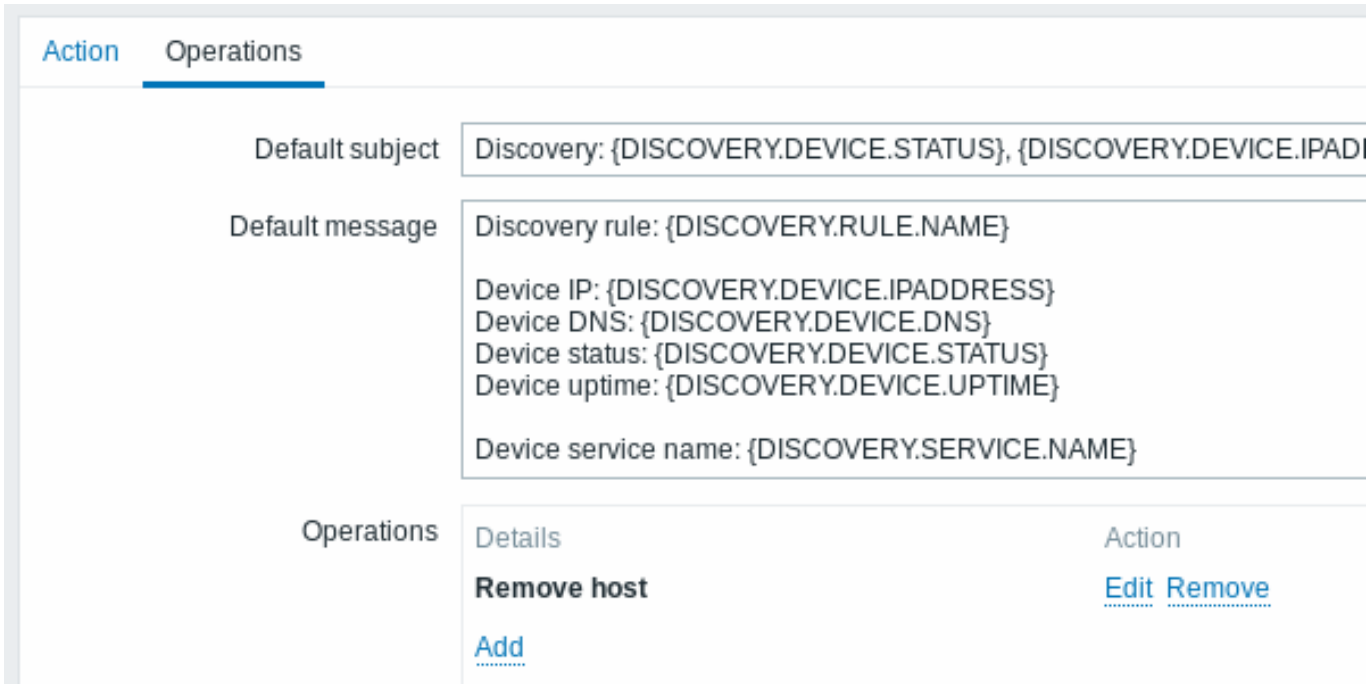




Passo 4

Definindo uma ação de descoberta para remover servidores perdidos.





Um servidor será removido se o serviço ”agente Zabbix” estiver ’inativo’ por mais mais de 24 horas (86400 segundos).

2 Registro automático de agente ativo

Visão geral

É possível permitir a autorregistro de agent ativos do Zabbix, após o qual o server pode começar a monitorá-los. Dessa forma, novos hosts podem ser adicionados para monitoramento sem configurá-los manualmente no server.

O autorregistro pode ocorrer quando um agent ativo, até então desconhecido, solicita verificações.

Esse recurso pode ser muito útil para o monitoramento automático de novos nós em Cloud. Assim que você tiver um novo nó na Cloud, o Zabbix começará automaticamente a coleta de dados de desempenho e disponibilidade do host.

O autorregistro de agent ativo também suporta o monitoramento de hosts adicionados com verificações passivas. Quando o agent ativo solicita verificações, desde que os parâmetros de configuração `ListenIP` ou `ListenPort` estejam definidos no arquivo de configuração, eles são enviados ao server. Se vários endereços IP forem especificados, o primeiro é enviado ao server.

O server, ao adicionar o novo host autorregistrado, usa o endereço IP e a porta recebidos para configurar o agent. Se nenhum valor de endereço IP for recebido, será usado o endereço da conexão recebida. Se nenhum valor de porta for recebido, será usada a porta 10050.

É possível especificar que o host deve ser autorregistrado com um nome DNS como interface padrão do agent.

O autorregistro é reexecutado:

- se as informações de metadados do host forem alteradas:
  - devido à alteração do HostMetadata e reinicialização do agent
  - devido à alteração do valor retornado por HostMetadataItem
- para hosts criados manualmente sem metadados
- se um host for alterado manualmente para ser monitorado por outro proxy do Zabbix
- se o autorregistro para o mesmo host vier de um novo proxy do Zabbix

O intervalo de heartbeat do autorregistro de agent ativo para o Zabbix server e o Zabbix proxy é de 120 segundos. Portanto, caso um host descoberto seja excluído, o autorregistro será reexecutado em 120 segundos.

Configuração

Especificar o servidor

Certifique-se de que o Zabbix server está identificado no arquivo de configuração do agent - `zabbix_agentd.conf`:

```
ServerActive=10.0.0.1
```

A menos que você defina especificamente um *Hostname* em `zabbix_agentd.conf`, o nome do host do sistema onde o agent está localizado será usado pelo server para nomear o host. O nome do host do sistema no Linux pode ser obtido executando o comando `hostname`.

Se *Hostname* for definido na configuração do Zabbix agent como uma lista de hosts separada por vírgulas, hosts serão criados para todos os nomes de host listados.

Reinicie o agent após fazer qualquer alteração no arquivo de configuração.

Ação para autoregistro de agent ativo

Quando o server recebe uma solicitação de autoregistro de um agent, ele chama uma ação. Uma ação com fonte de evento "Autoregistration" deve ser configurada para o autoregistro de agent.

> **Note:**
> Configurar a descoberta de rede não é necessário para que agents ativos se autoregistrem.

No frontend do Zabbix, vá para *Alertas* → *Ações*, selecione *Ações de autoregistro* e clique em *Criar ação*:

- Na aba Ação, dê um nome para sua ação
- Opcionalmente, especifique as condições. Você pode fazer uma correspondência de substring ou expressão regular nas condições para nome do host/metadados do host. Se você for usar a condição "Host metadata", veja a próxima seção.
- Na aba Operações, adicione as operações relevantes, como - 'Adicionar host', 'Adicionar ao grupo de hosts' (por exemplo, *Discovered hosts*), 'Vincular templates', etc.

> **Note:**
> Se os hosts que irão se autoregistrar provavelmente serão suportados apenas para monitoramento ativo (como hosts que estão protegidos por firewall do seu Zabbix server), talvez você queira criar um template específico como *Template_Linux-active* para vincular.

Os hosts criados são adicionados ao grupo *Discovered hosts* (por padrão, configurável em *Administração* > *Geral* > *Outros*). Se você deseja que os hosts sejam adicionados a outro grupo, adicione uma operação *Remover do grupo de hosts* (especificando "Discovered hosts") e também adicione uma operação *Adicionar ao grupo de hosts* (especificando outro grupo de hosts), porque um host deve pertencer a um grupo de hosts.

Autoregistro seguro

Uma forma segura de autoregistro é possível configurando a autenticação baseada em PSK com conexões criptografadas.

O nível de criptografia é configurado globalmente em *Administração* > *Geral* > *Autoregistro*. É possível selecionar sem criptografia, criptografia TLS com autenticação PSK ou ambos (de modo que alguns hosts possam se registrar sem criptografia enquanto outros através de criptografia).

A autenticação por PSK é verificada pelo Zabbix server antes de adicionar um host. Se for bem-sucedido, o host é adicionado e as *Conexões de/para o host* são definidas apenas como 'PSK' com identidade/chave pré-compartilhada igual à configuração global de autoregistro.

Attention:
Para garantir a segurança do autoregistro em instalações que usam proxies, a criptografia entre o Zabbix server e o proxy deve ser ativada.

Usando DNS como interface padrão

Os parâmetros de configuração HostInterface e HostInterfaceItem parâmetros de configuração permitem especificar um valor personalizado para a interface do host durante a autorregistro.

Mais especificamente, eles são úteis se o host deve ser autorregistrado com um nome DNS como a interface agent padrão em vez de seu endereço IP. Nesse caso, o nome DNS deve ser especificado ou retornado como o valor dos parâmetros HostInterface ou HostInterfaceItem. Se o valor de um desses parâmetros mudar — por exemplo, de um endereço IP para um nome DNS ou vice-versa — a interface padrão do host autorregistrado será atualizada de acordo. Essa atualização é aplicada ao host existente, não criando um novo. Para enviar o novo valor, o agent deve ser reiniciado para que ele reinicie o processo de autorregistro.

Note:
Se os parâmetros HostInterface ou HostInterfaceItem não estiverem configurados, o parâmetro listen_dns será usado. Esse valor é determinado realizando uma pesquisa reversa de DNS do endereço IP do agent. Se o DNS reverso não estiver devidamente configurado ou retornar um nome inválido, isso pode resultar em autorregistro incorreto ou com falha devido a um valor de interface inválido.

Usando metadados do host

Quando o agent envia uma solicitação de autorregistro para o server, ele envia seu hostname. Em alguns casos (por exemplo, nós da nuvem Amazon) um hostname não é suficiente para o Zabbix server diferenciar os hosts descobertos. Os metadados do host podem ser usados opcionalmente para enviar outras informações de um agent para o server.

Os metadados do host são configurados no arquivo de configuração do agent - `zabbix_agentd.conf`. Existem 2 maneiras de especificar os metadados do host no arquivo de configuração:

```
HostMetadata
HostMetadataItem
```

Veja a descrição das opções no link acima.

O parâmetro HostMetadataItem pode retornar até 65535 pontos de código UTF-8. Um valor mais longo será truncado.

Observe que, no MySQL, o comprimento máximo efetivo em caracteres será menor se o valor retornado contiver caracteres multibyte. Por exemplo, um valor contendo apenas caracteres de 3 bytes será limitado a 21844 caracteres no total, enquanto um valor contendo apenas caracteres de 4 bytes será limitado a 16383 símbolos.

Attention:
Uma tentativa de autorregistro ocorre toda vez que um agent ativo envia uma solicitação para atualizar as verificações ativas para o server. O atraso entre as solicitações é especificado no parâmetro RefreshActiveChecks do agent. A primeira solicitação é enviada imediatamente após o agent ser reiniciado.

Exemplos

Autoregistro por SO usando HostMetadata

Suponha que você queira que os hosts sejam registrados automaticamente pelo servidor Zabbix. Você possui agents Zabbix ativos (consulte a seção ”Configuração” acima) em sua rede. Existem hosts Windows e hosts Linux em sua rede e você tem os templates ”Linux by Zabbix agent” e ”Windows by Zabbix agent” disponíveis em seu frontend Zabbix. Portanto, no registro do host, você gostaria que o template Linux/Windows apropriado fosse aplicado ao host que está sendo registrado. Por padrão, apenas o nome do host é enviado ao servidor no autoregistro, o que pode não ser suficiente. Para garantir que o template correto seja aplicado ao host, você deve usar a metadados do host.

Configuração do frontend

A primeira coisa a fazer é configurar o frontend. Crie 2 ações. A primeira ação:

- Nome: Autoregistro de host Linux
- Condições: Metadados do host contém *Linux*
- Operações: Vincular templates: Linux por Zabbix agent

Note:
Você pode pular a operação ”Adicionar host” neste caso. Vincular um template a um host requer adicionar o host primeiro, então o server fará isso automaticamente.

A segunda ação:

- Nome: Autoregistro de host Windows
- Condições: Metadados do host contém *Windows*
- Operações: Vincular templates: Windows por Zabbix agent

Configuração do agent

Agora você precisa configurar os agents. Adicione a seguinte linha aos arquivos de configuração do agent:

```
HostMetadataItem=system.uname
```

Dessa forma, você garante que os metadados do host conterão "Linux" ou "Windows", dependendo do host em que o agent está sendo executado. Um exemplo de metadados do host neste caso:

```
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
```

Não se esqueça de reiniciar o agent após fazer qualquer alteração no arquivo de configuração.

Usando HostMetadata para controlar o registro automático e os templates

Etapa 1 - Proteger o registro com HostMetadata

Usando metadados do host para permitir alguma proteção básica contra o registro de hosts indesejados.

Configuração do frontend

Crie uma ação no frontend, usando algum código secreto difícil de adivinhar para impedir hosts indesejados:

- Nome: Ação de autorregistro Linux
- Condições:
  - Tipo de cálculo: AND
  - Condição (A): Metadados do host contém //Linux//
  - Condição (B): Metadados do host contém //21df83bf21bf0be663090bb8d4128558ab9b95fba66a6dbf834f8b91ae5e08ae//
- Operações:
  - Enviar mensagem para usuários: Admin via todos os meios
  - Adicionar aos grupos de hosts: Servidores Linux
  - Associar templates: Linux pelo agent Zabbix

Observe que este método sozinho não fornece proteção forte porque os dados são transmitidos em texto simples. É necessário recarregar o cache de configuração para que as alterações tenham efeito imediato.

Configuração do agent

Adicione a seguinte linha ao arquivo de configuração do agent:

```
HostMetadata=Linux    21df83bf21bf0be663090bb8d4128558ab9b95fba66a6dbf834f8b91ae5e08ae
```

onde "Linux" é uma plataforma e o restante da string é o texto secreto difícil de adivinhar.

Não se esqueça de reiniciar o agent após fazer qualquer alteração no arquivo de configuração.

Etapa 2 - Adicionar template ao host registrado

É possível adicionar templates adicionais para um host já registrado. Neste caso, o template MySQL by Zabbix agent será vinculado apenas aos hosts cujo HostMetadata contenha o token `MySQL`.

Configuração do frontend

Atualize a ação no frontend:

- Nome: Autoregistration action Linux
- Condições:
  - Tipo de cálculo: AND
  - Condição (A): Host metadata contém Linux
  - Condição (B): Host metadata contém 21df83bf21bf0be663090bb8d4128558ab9b95fba66a6dbf834f8b91ae5e08ae
  - Condição (C): Host metadata contém MySQL
- Operações:
  - Enviar mensagem para usuários: Admin via todos os meios
  - Adicionar aos grupos de hosts: Linux servers
  - Associar templates: Linux by Zabbix agent
  - Associar templates: MySQL by Zabbix Agent

Configuração do agent

Atualize a seguinte linha no arquivo de configuração do agent:

```
HostMetadata=MySQL on Linux 21df83bf21bf0be663090bb8d4128558ab9b95fba66a6dbf834f8b91ae5e08ae
```

Não se esqueça de reiniciar o agent após fazer qualquer alteração no arquivo de configuração.

3 Descoberta de baixo nível

Visão geral A descoberta de baixo nível (LLD) fornece uma maneira de criar automaticamente items, triggers e gráficos para diferentes entidades em um host. Por exemplo, o Zabbix pode começar a monitorar automaticamente sistemas de arquivos ou interfaces de rede em sua máquina, sem a necessidade de criar items para cada sistema de arquivos ou interface de rede manualmente. A LLD também pode criar hosts, por exemplo, para preencher máquinas virtuais descobertas em um hipervisor. Além disso, é possível configurar o Zabbix para remover automaticamente entidades desnecessárias com base nos resultados reais da descoberta realizada periodicamente.

Um usuário pode definir seus próprios tipos de descoberta, desde que siga um protocolo JSON específico.

A arquitetura geral do processo de descoberta é a seguinte.

Primeiro, um usuário cria uma regra de descoberta em *Coleta de dados* > *Templates*, na coluna *Descoberta*. Uma regra de descoberta consiste em (1) um item que descobre as entidades necessárias (por exemplo, sistemas de arquivos ou interfaces de rede) e (2) protótipos de items, triggers e gráficos que devem ser criados com base no valor desse item.

Um item que descobre as entidades necessárias é como um item regular visto em outros lugares: o server solicita a um agent Zabbix (ou qualquer que seja o tipo do item definido) um valor para esse item, o agent responde com um valor textual. A diferença é que o valor que o agent responde deve conter uma lista de entidades descobertas em formato JSON. Embora os detalhes desse formato sejam importantes apenas para implementadores de verificações de descoberta personalizadas, é necessário saber que o valor retornado contém uma lista de pares macro → valor. Por exemplo, o item "net.if.discovery" pode retornar dois pares: "{#IFNAME}" → "lo" e "{#IFNAME}" → "eth0".

Essas macros são usadas em nomes, chaves e outros campos de protótipo, onde são então substituídas pelos valores recebidos para criar items, triggers, gráficos ou até mesmo hosts reais para cada entidade descoberta. Veja a lista completa de opções para usar macros LLD.

Quando o server recebe um valor para um item de descoberta, ele analisa os pares macro → valor e, para cada par, gera items, triggers e gráficos reais, com base em seus protótipos. No exemplo com "net.if.discovery" acima, o server geraria um conjunto de items, triggers e gráficos para a interface de loopback "lo" e outro conjunto para a interface "eth0".

Observe que desde o **Zabbix 4.2**, o formato do JSON retornado pelas regras de descoberta de baixo nível foi alterado. Não é mais esperado que o JSON contenha o objeto "data". A descoberta de baixo nível agora aceitará um JSON normal contendo um array, para suportar novos recursos, como o pré-processamento do valor do item e caminhos personalizados para valores de macros de descoberta de baixo nível em um documento JSON.

As chaves de descoberta integradas foram atualizadas para retornar um array de linhas LLD na raiz do documento JSON. O Zabbix extrairá automaticamente uma macro e um valor se um campo de array usar a sintaxe {#MACRO} como chave. Quaisquer novas verificações de descoberta nativas usarão a nova sintaxe sem os elementos "data". Ao processar um valor de descoberta de baixo nível, primeiro a raiz é localizada (array em `$.` ou `$.data`).

Embora o elemento "data" tenha sido removido de todos os items nativos relacionados à descoberta, para compatibilidade com versões anteriores, o Zabbix ainda aceitará a notação JSON com um elemento "data", embora seu uso seja desencorajado. Se o JSON contiver um objeto com apenas um elemento de array "data", então ele extrairá automaticamente o conteúdo do elemento usando JSONPath `$.data`. A descoberta de baixo nível agora aceita macros LLD opcionais definidas pelo usuário com um caminho personalizado especificado na sintaxe JSONPath.

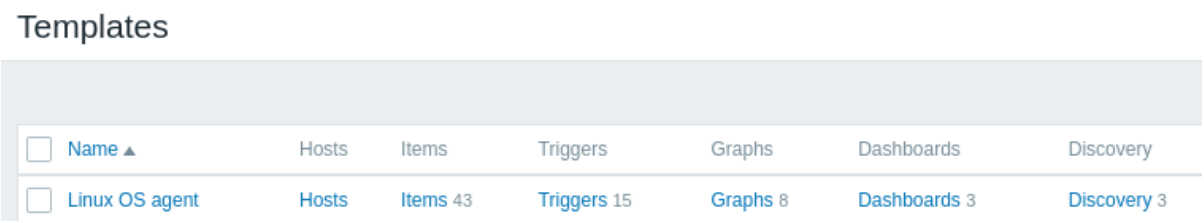
Warning:
Como resultado das alterações acima, agents mais recentes não poderão mais trabalhar com um server Zabbix mais antigo.

Veja também: Entidades descobertas

Configurando a descoberta de baixo nível Vamos ilustrar a descoberta de baixo nível com base em um exemplo de descoberta de sistema de arquivos.

Para configurar a descoberta, faça o seguinte:

- Vá para: *Coleta de dados* > *Templates* ou *Hosts*.
- Clique em *Descoberta* na linha do template/host apropriado.



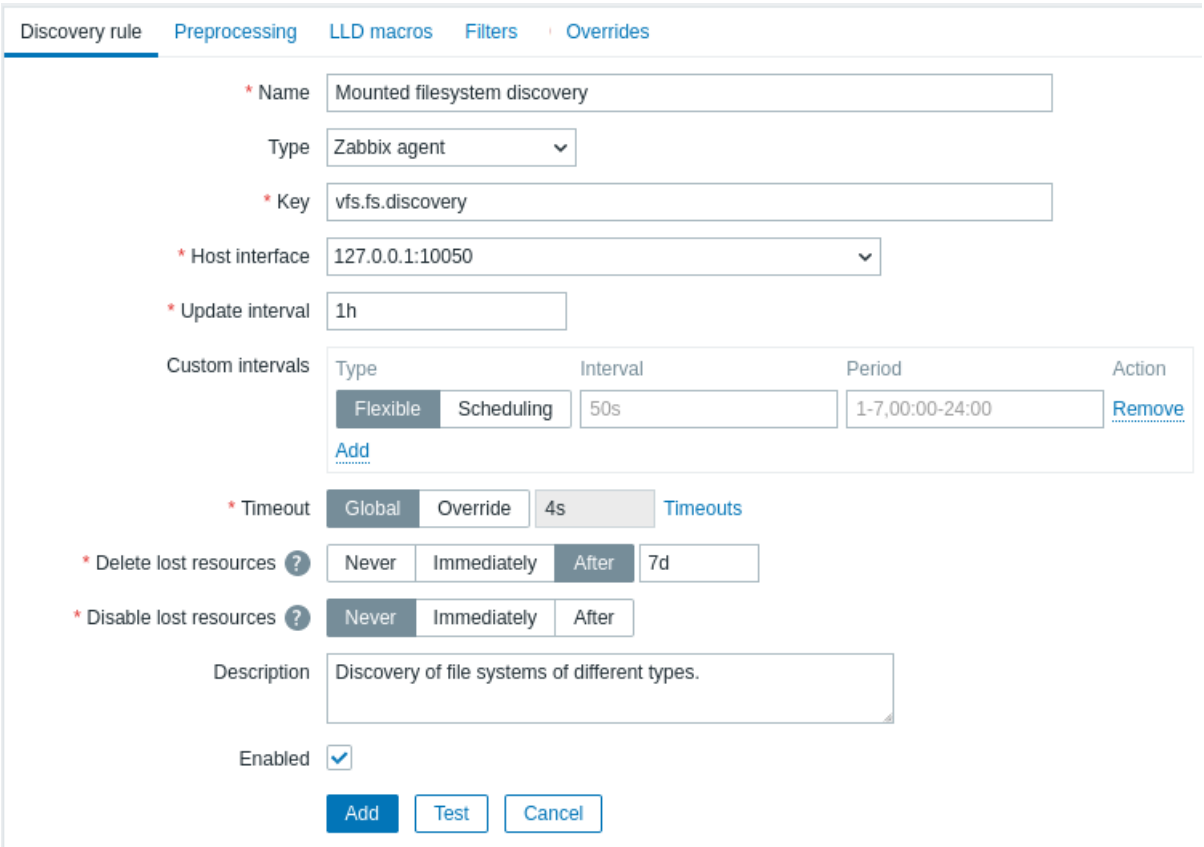
- Clique em *Criar regra de descoberta* no canto superior direito da tela.
- Preencha o formulário da regra de descoberta com os detalhes necessários.

Regra de descoberta

O formulário de regra de descoberta contém cinco abas, representando, da esquerda para a direita, o fluxo de dados durante a descoberta:

- *Regra de descoberta* - especifica, principalmente, o item interno ou script personalizado para obter os dados de descoberta.
- *Pré-processamento* - aplica algum pré-processamento aos dados descobertos.
- *Macros LLD* - permite extrair alguns valores de macro para usar em items, triggers, etc. descobertos.
- *Filtros* - permite filtrar os valores descobertos.
- *Substituições* - permite modificar items, triggers, gráficos ou protótipos de host ao aplicar a objetos descobertos específicos.

A aba **Regra de descoberta** contém a chave do item a ser usada para descoberta (bem como alguns atributos gerais da regra de descoberta):



Todos os campos obrigatórios estão marcados com um asterisco vermelho.

Parâmetro	Descrição
Nome	Nome da regra de descoberta.
Tipo	O tipo de verificação para realizar a descoberta. Neste exemplo estamos usando um tipo de item *Zabbix agent*. A regra de descoberta também pode ser um item dependente, dependendo de um item regular. Não pode depender de outra regra de descoberta. Para um item dependente, selecione o respectivo tipo (*Item dependente*) e especifique o item mestre no campo 'Item mestre'. O item mestre deve existir.

Parâmetro	Descrição
Chave	Insira a chave do item de descoberta (até 2048 caracteres). Por exemplo, você pode usar a chave de item interna ”vfs.fs.discovery” para retornar uma string JSON com a lista de sistemas de arquivos presentes no computador, seus tipos e opções de montagem. Observe que outra opção para descoberta de sistemas de arquivos é usar os resultados de descoberta pela chave de agent ”vfs.fs.get” (veja exemplo).
Intervalo de atualização	Este campo especifica com que frequência o Zabbix realiza a descoberta. No início, quando você está apenas configurando a descoberta de sistemas de arquivos, pode desejar definir um intervalo pequeno, mas, uma vez que saiba que está funcionando, pode definir para 30 minutos ou mais, pois os sistemas de arquivos geralmente não mudam com frequência. Sufixos de tempo são suportados, por exemplo, 30s, 1m, 2h, 1d. Macros de usuário são suportadas. *Nota*: O intervalo de atualização só pode ser definido como ’0’ se existirem intervalos personalizados com valor diferente de zero. Se definido como ’0’, e existir um intervalo personalizado (flexível ou agendado) com valor diferente de zero, o item será coletado durante a duração do intervalo personalizado. Novas regras de descoberta serão verificadas em até 60 segundos após sua criação, a menos que tenham Agendamento ou Intervalo de atualização flexível e o *Intervalo de atualização* esteja definido como 0. *Nota* que para uma regra de descoberta existente, a descoberta pode ser realizada imediatamente pressionando o botão *Executar agora*.
Intervalos personalizados	Você pode criar regras personalizadas para checar o item: **Flexível** - cria uma exceção ao *Intervalo de atualização* (intervalo com frequência diferente) **Agendamento** - cria um agendamento de coleta personalizado. Para informações detalhadas, veja Intervalos personalizados.
Timeout	Defina o timeout da verificação de descoberta. Selecione a opção de timeout: **Global** - o timeout do proxy/global é usado (exibido no campo *Timeout* desabilitado); **Substituir** - um timeout personalizado é usado (definido no campo *Timeout*; intervalo permitido: 1 - 600s). Sufixos de tempo, por exemplo, 30s, 1m, e macros de usuário são suportados. Clicar no link *Timeouts* permite configurar os timeouts do proxy ou globais (se um proxy não for usado). Observe que o link *Timeouts* é visível apenas para usuários do tipo *Super admin* com permissões para as seções do frontend *Administração* > *Geral* ou *Administração* > *Proxies*.
Excluir recursos perdidos	Especifique em quanto tempo a entidade descoberta será excluída assim que seu status de descoberta se tornar ”Não descoberto mais”: *Nunca* - não será excluída; *Imediatamente* - será excluída imediatamente; *Após* - será excluída após o período de tempo especificado. O valor deve ser maior que o valor de *Desabilitar recursos perdidos*. Sufixos de tempo são suportados, por exemplo, 2h, 1d. Macros de usuário são suportadas. *Nota*: Usar ”Imediatamente” não é recomendado, pois apenas editar o filtro incorretamente pode acabar excluindo a entidade com todos os dados históricos. Observe que recursos desabilitados manualmente não serão excluídos pela descoberta de baixo nível.
Desabilitar recursos perdidos	Especifique em quanto tempo a entidade descoberta será desabilitada assim que seu status de descoberta se tornar ”Não descoberto mais”: *Nunca* - não será desabilitada; *Imediatamente* - será desabilitada imediatamente; *Após* - será desabilitada após o período de tempo especificado. O valor deve ser maior que o intervalo de atualização da regra de descoberta. Observe que recursos desabilitados automaticamente serão habilitados novamente, se redescobertos pela descoberta de baixo nível. Recursos desabilitados manualmente não serão habilitados novamente se redescobertos. Este campo não é exibido se *Excluir recursos perdidos* estiver definido como ”Imediatamente”. Sufixos de tempo são suportados, por exemplo, 2h, 1d. Macros de usuário são suportadas.
Descrição	Insira uma descrição.
Habilitado	Se marcado, a regra será processada.

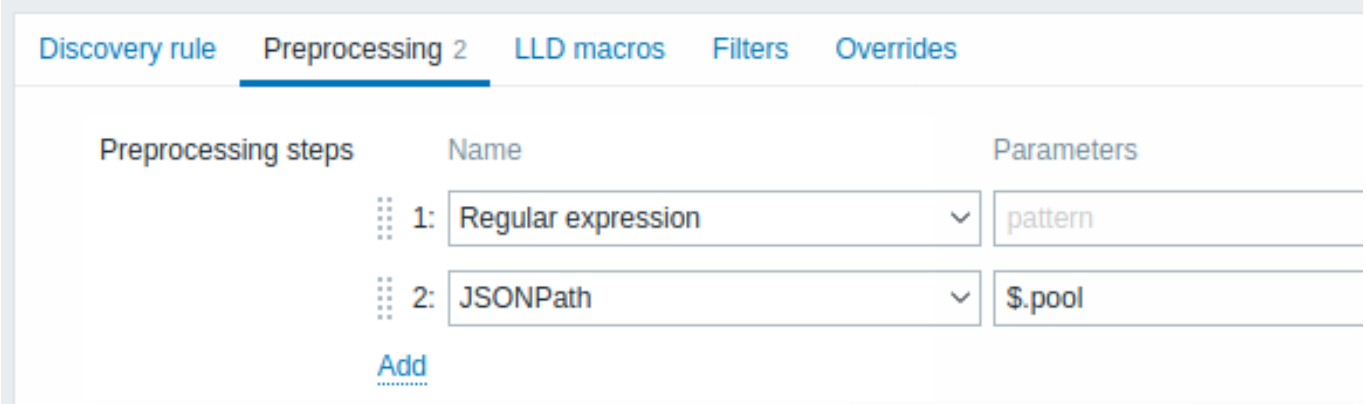
Note:
O histórico da regra de descoberta não é preservado.

Pré-processamento

A aba **Pré-processamento** permite definir regras de transformação a serem aplicadas ao resultado da descoberta. Uma ou várias transformações são possíveis nesta etapa. As transformações são executadas na ordem em que são definidas. Todo o pré-processamento é feito pelo Zabbix server.

Veja também:

- Detalhes do pré-processamento
- Testando o pré-processamento



Tipo		
	Transformação	Descrição
Texto		
	Expressão regular	Corresponde o valor recebido à expressão regular <pattern> e substitui o valor pelo <output> extraído. A expressão regular suporta a extração de no máximo 10 grupos capturados com a sequência \N. Parâmetros: **pattern** - expressão regular **output** - modelo de formatação de saída. Uma sequência de escape \N (onde N=1...9) é substituída pelo N-ésimo grupo correspondente. Uma sequência de escape \0 é substituída pelo texto correspondente. Se você marcar a caixa *Personalizar em caso de falha*, é possível especificar opções personalizadas de tratamento de erro: descartar o valor, definir um valor especificado ou definir uma mensagem de erro especificada.
	Substituir	Encontra a string de pesquisa e a substitui por outra (ou nada). Todas as ocorrências da string de pesquisa serão substituídas. Parâmetros: **string de pesquisa** - a string a ser encontrada e substituída, diferencia maiúsculas de minúsculas (obrigatório) **substituição** - a string para substituir a string de pesquisa. A string de substituição também pode estar vazia, permitindo efetivamente excluir a string de pesquisa quando encontrada. É possível usar sequências de escape para pesquisar ou substituir quebras de linha, retorno de carro, tabulações e espaços "\n \r \t \s"; a barra invertida pode ser escapada como "\\" e as sequências de escape podem ser escapadas como "\\n". O escape de quebras de linha, retorno de carro e tabulações é feito automaticamente durante a descoberta de baixo nível.
Dados estruturados		
	JSONPath	Extrai valor ou fragmento de dados JSON usando a funcionalidade JSONPath. Se você marcar a caixa *Personalizar em caso de falha*, é possível especificar opções personalizadas de tratamento de erro: descartar o valor, definir um valor especificado ou definir uma mensagem de erro especificada.

Tipo	

XML XPath	Extrai valor ou fragmento de dados XML usando a funcionalidade XPath.
	Para que esta opção funcione, o Zabbix server deve ser compilado com suporte a libxml.
	Exemplos:
	`number(/document/item/value)` extrairá 10 de `<document><item><value>10</value></item></document>`
	`number(/document/item/@attribute)` extrairá 10 de `<document><item attribute="10"></item></document>`
	`/document/item` extrairá `<item><value>10</value></item>` de `<document><item><value>10</value></item></document>`
	Observe que namespaces não são suportados.
	Se você marcar a caixa *Personalizar em caso de falha*, é possível especificar opções personalizadas de tratamento de erro: descartar o valor, definir um valor especificado ou definir uma mensagem de erro especificada.

CSV para JSON	Converte dados de arquivo CSV para o formato JSON. Para mais informações, veja: Pré-processamento CSV para JSON.
XML para JSON	Converte dados em formato XML para JSON. Para mais informações, veja: Regras de serialização. Se você marcar a caixa *Personalizar em caso de falha*, é possível especificar opções personalizadas de tratamento de erro: descartar o valor, definir um valor especificado ou definir uma mensagem de erro especificada.
SNMP	
Valor SNMP walk	Extrai o valor pelo OID/nome MIB especificado e aplica opções de formatação: **Inalterado** - retorna Hex-STRING como string hexadecimal não escapada (*observe* que dicas de exibição ainda são aplicadas); **UTF-8 de Hex-STRING** - converte Hex-STRING para string UTF-8; **MAC de Hex-STRING** - converte Hex-STRING para string de endereço MAC (que terá `' '` substituído por `':'`); **Inteiro de BITS** - converte os primeiros 8 bytes de uma string de bits expressa como uma sequência de caracteres hexadecimais (por exemplo, "1A 2B 3C 4D") em um inteiro sem sinal de 64 bits. Em strings de bits com mais de 8 bytes, os bytes subsequentes serão ignorados. Se você marcar a caixa *Personalizar em caso de falha*, é possível especificar opções personalizadas de tratamento de erro: descartar o valor, definir um valor especificado ou definir uma mensagem de erro especificada.
SNMP walk para JSON	Converte valores SNMP para JSON. Especifique um nome de campo no JSON e o caminho OID SNMP correspondente. Os valores dos campos serão preenchidos pelos valores no caminho OID SNMP especificado. Você pode usar esta etapa de pré-processamento para descoberta de OIDs SNMP. Opções de formatação de valor semelhantes às da etapa *Valor SNMP walk* estão disponíveis. Se você marcar a caixa *Personalizar em caso de falha*, é possível especificar opções personalizadas de tratamento de erro: descartar o valor, definir um valor especificado ou definir uma mensagem de erro especificada.
Valor SNMP get	Aplica opções de formatação ao valor SNMP get: **UTF-8 de Hex-STRING** - converte Hex-STRING para string UTF-8; **MAC de Hex-STRING** - converte Hex-STRING para string de endereço MAC (que terá `' '` substituído por `':'`); **Inteiro de BITS** - converte os primeiros 8 bytes de uma string de bits expressa como uma sequência de caracteres hexadecimais (por exemplo, "1A 2B 3C 4D") em um inteiro sem sinal de 64 bits. Em strings de bits com mais de 8 bytes, os bytes subsequentes serão ignorados. Se você marcar a caixa *Personalizar em caso de falha*, é possível especificar opções personalizadas de tratamento de erro: descartar o valor, definir um valor especificado ou definir uma mensagem de erro especificada.
Scripts personalizados	
JavaScript	Digite o código JavaScript no editor modal que é aberto ao clicar no campo de parâmetro ou no ícone de lápis ao lado dele. Observe que o comprimento do JavaScript disponível depende do banco de dados usado. Para mais informações, veja: Pré-processamento Javascript

Tipo	
Validação	
Não corresponde à expressão regular	Especifique uma expressão regular à qual um valor não deve corresponder. Ex.: `Error:(.*?)\.` Se você marcar a caixa *Personalizar em caso de falha*, é possível especificar opções personalizadas de tratamento de erro: descartar o valor, definir um valor especificado ou definir uma mensagem de erro especificada.
Verificar erro em JSON	Verifica uma mensagem de erro em nível de aplicação localizada em JSONPath. Interrompe o processamento se for bem-sucedido e a mensagem não estiver vazia; caso contrário, continua o processamento com o valor que estava antes desta etapa de pré-processamento. Observe que esses erros de serviço externo são relatados ao usuário como estão, sem adicionar informações da etapa de pré-processamento. Ex.: `$.errors`. Se um JSON como `{"errors":"e1"}` for recebido, a próxima etapa de pré-processamento não será executada. Se você marcar a caixa *Personalizar em caso de falha*, é possível especificar opções personalizadas de tratamento de erro: descartar o valor, definir um valor especificado ou definir uma mensagem de erro especificada.
Verificar erro em XML	Verifica uma mensagem de erro em nível de aplicação localizada em Xpath. Interrompe o processamento se for bem-sucedido e a mensagem não estiver vazia; caso contrário, continua o processamento com o valor que estava antes desta etapa de pré-processamento. Observe que esses erros de serviço externo são relatados ao usuário como estão, sem adicionar informações da etapa de pré-processamento. Nenhum erro será relatado em caso de falha ao analisar XML inválido. Se você marcar a caixa *Personalizar em caso de falha*, é possível especificar opções personalizadas de tratamento de erro: descartar o valor, definir um valor especificado ou definir uma mensagem de erro especificada.
Corresponde à expressão regular	Especifique uma expressão regular à qual um valor deve corresponder. Se você marcar a caixa *Personalizar em caso de falha*, é possível especificar opções personalizadas de tratamento de erro: descartar o valor, definir um valor especificado ou definir uma mensagem de erro especificada.
Limitação	
Descartar inalterado com heartbeat	Descarta um valor se ele não tiver mudado dentro do período de tempo definido (em segundos). Valores inteiros positivos são suportados para especificar os segundos (mínimo - 1 segundo). Sufixos de tempo podem ser usados neste campo (por exemplo, 30s, 1m, 2h, 1d). Macros de usuário e macros de descoberta de baixo nível podem ser usadas neste campo. Apenas uma opção de limitação pode ser especificada para um item de descoberta. Ex.: `1m`. Se um texto idêntico for passado para esta regra duas vezes em 60 segundos, ele será descartado. *Nota*: Alterar protótipos de item não redefine a limitação. A limitação é redefinida apenas quando as etapas de pré-processamento são alteradas.
Prometheus	
Prometheus para JSON	Converte as métricas Prometheus necessárias para JSON. Veja Verificações Prometheus para mais detalhes.

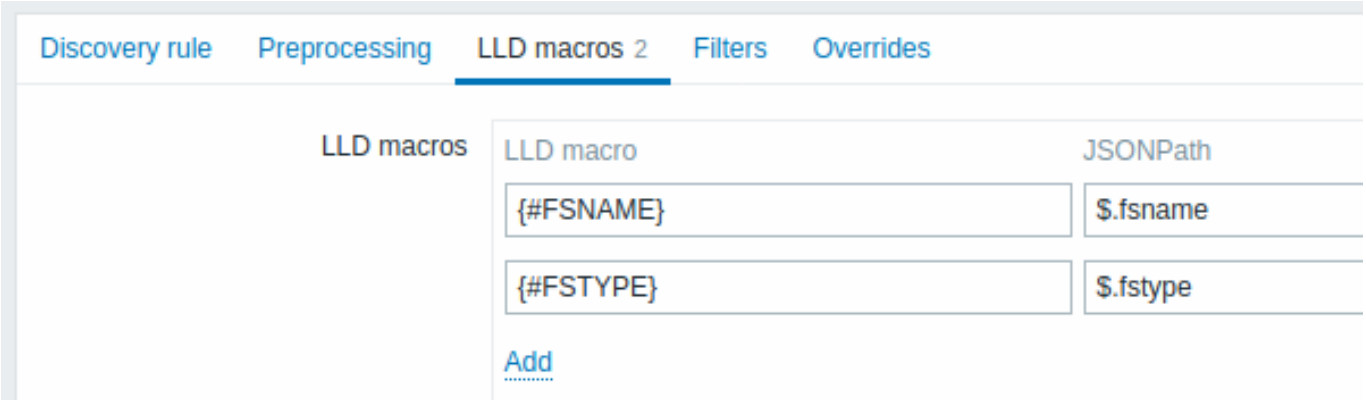
Observe que se a regra de descoberta foi aplicada ao host via template, o conteúdo desta aba é somente leitura.

Macros personalizadas

A aba **Macros LLD** permite especificar macros de descoberta de baixo nível personalizadas.

Macros personalizadas são úteis em casos em que o JSON retornado não possui as macros necessárias já definidas. Então, por exemplo:

- A chave nativa `vfs.fs.discovery` para descoberta de sistemas de arquivos retorna um JSON com algumas macros LLD predefinidas, como {#FSNAME}, {#FSTYPE}. Essas macros podem ser usadas diretamente em protótipos de item, trigger (veja as seções subsequentes da página); não é necessário definir macros personalizadas;
- O item agent `vfs.fs.get` também retorna um JSON com dados do sistema de arquivos, mas sem nenhuma macro LLD predefinida. Neste caso, você pode definir as macros manualmente e mapeá-las para os valores no JSON usando JSONPath:

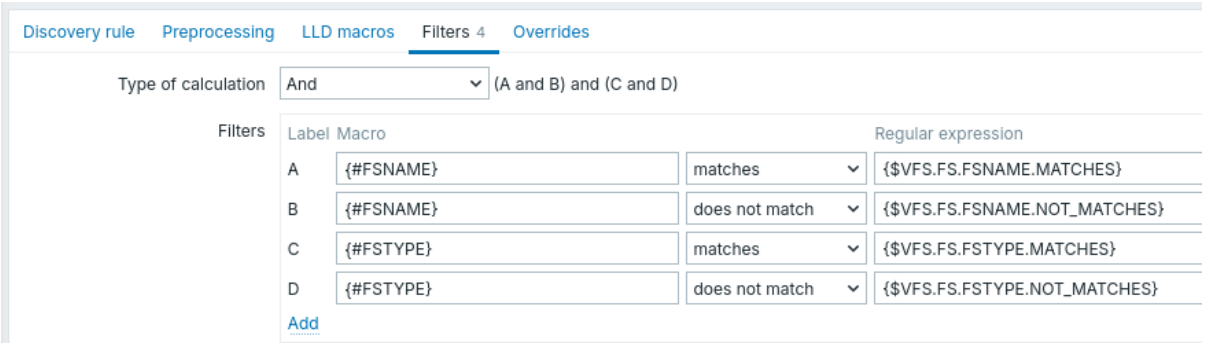


Os valores extraídos podem ser usados em items, triggers, etc. descobertos. Observe que os valores serão extraídos do resultado da descoberta e de quaisquer etapas de pré-processamento até o momento.

Parâmetro	Descrição
Macro LLD	Nome da macro de descoberta de baixo nível, usando a seguinte sintaxe: `{#MACRO}`.
JSONPath	Caminho usado para extrair o valor da macro LLD de uma linha LLD, usando a sintaxe JSONPath. Os valores extraídos do JSON retornado são usados para substituir as macros LLD nos campos dos protótipos de item, trigger, etc. O JSONPath pode ser especificado usando a notação de ponto ou a notação de colchetes. A notação de colchetes deve ser usada em caso de caracteres especiais e Unicode, como `$['unicode + special chars #1']['unicode + special chars #2']`. Por exemplo, `$.foo` irá extrair ”bar” e ”baz” deste JSON: `[{"foo":"bar"}, {"foo":"baz"}]` Observe que `$.foo` também irá extrair ”bar” e ”baz” deste JSON: `{"data":[{"foo":"bar"}, {"foo":"baz"}]}` porque um único objeto ”data” é processado automaticamente (para compatibilidade com versões anteriores com a implementação de descoberta de baixo nível nas versões do Zabbix anteriores à 4.2).

Filtro

Um filtro pode ser usado para gerar itens, triggers e gráficos reais apenas para entidades que correspondam aos critérios. A aba **Filtros** contém definições de filtro de regra de descoberta, permitindo filtrar valores de descoberta:



Parâmetro	Descrição
Tipo de cálculo	As seguintes opções para calcular filtros estão disponíveis: **E** - todos os filtros devem ser aprovados; **Ou** - basta que um filtro seja aprovado; **E/Ou** - usa *E* com nomes de macro diferentes e *Ou* com o mesmo nome de macro; **Expressão personalizada** - oferece a possibilidade de definir um cálculo personalizado de filtros. A fórmula deve incluir todos os filtros da lista. Limitado a 255 símbolos.

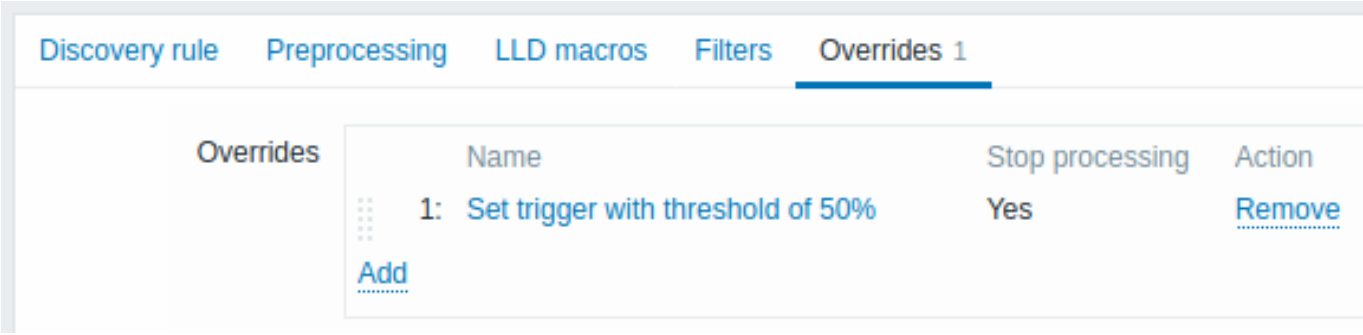
Parâmetro	Descrição
Filtros	Os seguintes operadores de condição de filtro estão disponíveis: *corresponde*, *não corresponde*, *existe*, *não existe*. Os operadores *corresponde* e *não corresponde* esperam uma Expressão Regular Compatível com Perl (PCRE). Por exemplo, se você estiver interessado apenas nos sistemas de arquivos C:, D: e E:, você pode colocar {#FSNAME} em "Macro" e a expressão regular "^C|^D|^E" no campo de texto "Expressão regular". A filtragem também é possível por tipos de sistema de arquivos usando a macro {#FSTYPE} (por exemplo, "^ext|^reiserfs") e por tipos de unidade (suportado apenas pelo agent do Windows) usando a macro {#FSDRIVETYPE} (por exemplo, "fixed"). Você pode inserir uma expressão regular ou referenciar uma expressão regular global no campo "Expressão regular". Para testar uma expressão regular, você pode usar "grep -E", por exemplo: `for f in ext2 nfs reiserfs smbfs; do echo $f | grep -E '^ext|^reiserfs' || echo "SKIP: $f"; done` Os operadores *existe* e *não existe* permitem filtrar entidades com base na presença ou ausência da macro LLD especificada na resposta. Observe que se uma macro do filtro estiver ausente na resposta, a entidade encontrada será ignorada, a menos que uma condição "não existe" seja especificada para essa macro. Um aviso será exibido se a ausência de uma macro afetar o resultado da expressão. Por exemplo, se {#B} estiver ausente em: {#A} corresponde a 1 e {#B} corresponde a 2 - exibirá um aviso {#A} corresponde a 1 ou {#B} corresponde a 2 - sem aviso

Warning:
Um erro ou um erro de digitação na expressão regular usada na regra LLD (por exemplo, uma expressão regular incorreta em "Sistemas de arquivos para descoberta") pode causar a exclusão de milhares de elementos de configuração, valores históricos e eventos para muitos hosts.

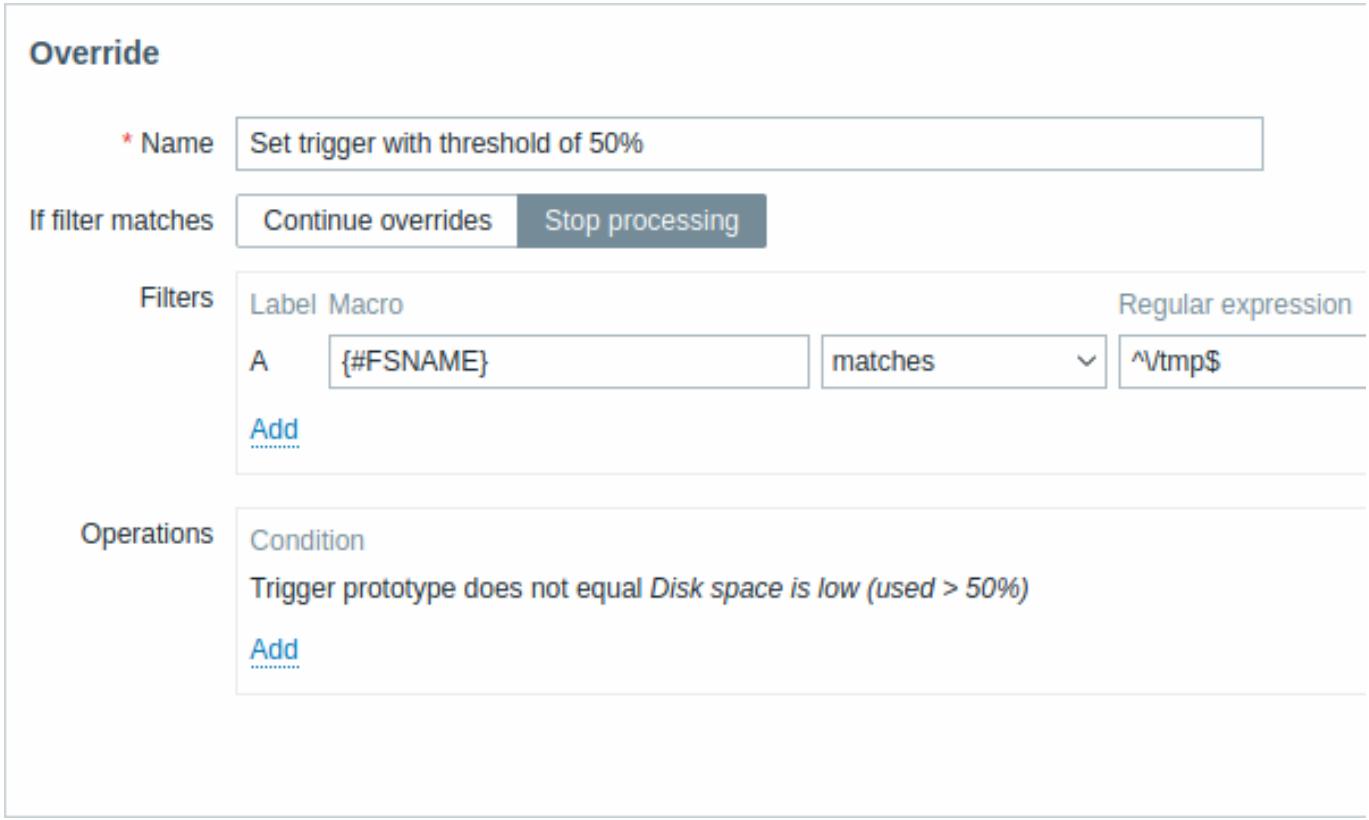
Attention:
O banco de dados Zabbix no MySQL deve ser criado como case-sensitive se nomes de sistemas de arquivos que diferem apenas por maiúsculas/minúsculas precisarem ser descobertos corretamente.

Substituição

A aba **Substituições** permite definir regras para modificar a lista de protótipos de item, trigger, gráfico e host ou seus atributos para objetos descobertos que atendam a determinados critérios.



As substituições (se houver) são exibidas em uma lista reordenável por arrastar e soltar e executadas na ordem em que são definidas. Para configurar os detalhes de uma nova substituição, clique em Add no bloco *Substituições*. Para editar uma substituição existente, clique no nome da substituição. Uma janela pop-up será aberta permitindo editar os detalhes da regra de substituição.

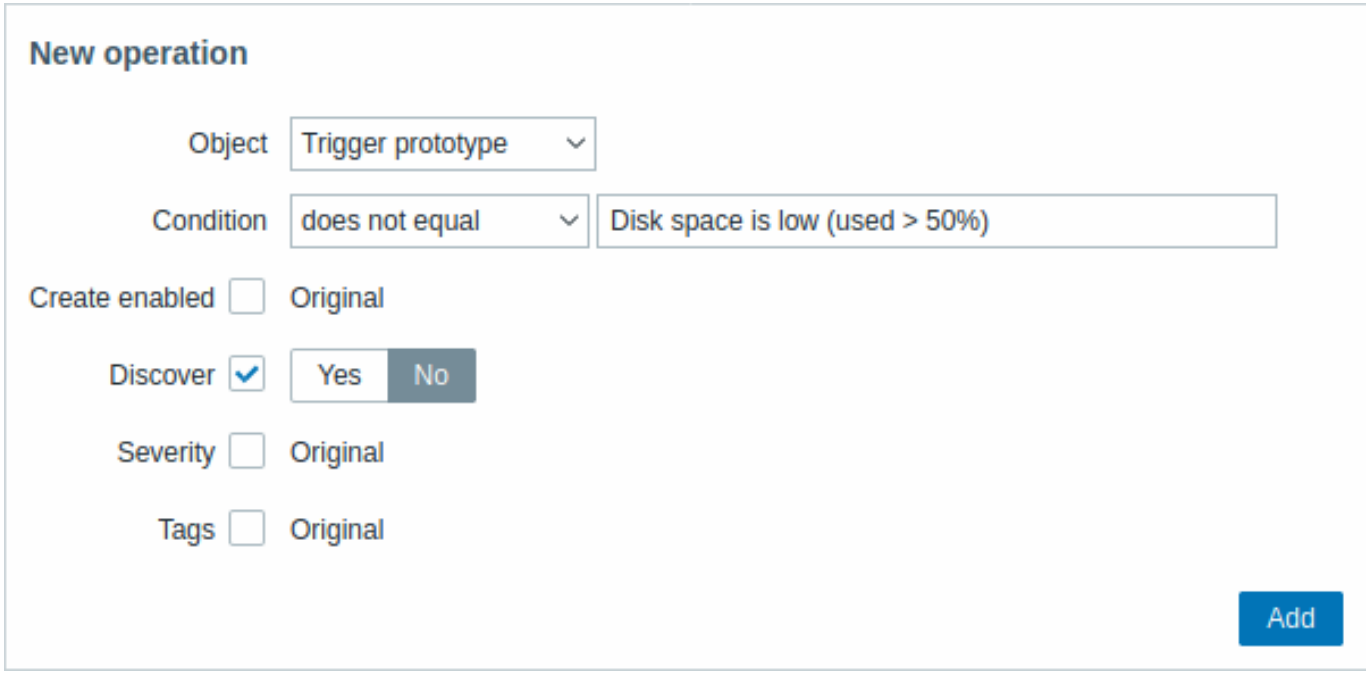


Todos os parâmetros obrigatórios são marcados com asteriscos vermelhos.

Parâmetro	Descrição
Nome	Um nome de substituição exclusivo (por regra LLD).
Se o filtro corresponder	Define se as próximas substituições devem ser processadas quando as condições do filtro forem atendidas: **Continuar substituições** - as substituições subsequentes serão processadas. **Parar processamento** - as operações das substituições anteriores (se houver) e desta substituição serão executadas, as substituições subsequentes serão ignoradas para as linhas LLD correspondentes.
Filtros	Determina a quais entidades descobertas a substituição deve ser aplicada. Os filtros de substituição são processados após os filtros da regra de descoberta e têm a mesma funcionalidade.
Operações	As operações de substituição são exibidas com estes detalhes: **Condição** - um tipo de objeto e uma condição a ser atendida para o nome do objeto; por exemplo: O protótipo de trigger não é igual a *Disk space is low (used > 50%)*. **Ação** - são exibidos links para editar e remover uma operação.

Configurando uma operação

Para configurar os detalhes de uma nova operação, clique em Add no bloco Operações. Para editar uma operação existente, clique em Edit ao lado da operação. Uma janela pop-up será aberta onde você pode editar os detalhes da operação.



Parâmetro	Descrição
Objeto	Quatro tipos de objetos estão disponíveis: Protótipo de item Protótipo de trigger Protótipo de gráfico Protótipo de host
Condição	Permite filtrar as entidades às quais a operação deve ser aplicada.
Operador	Operadores suportados: **igual** - aplicar a este protótipo **diferente** - aplicar a todos os protótipos, exceto este **contém** - aplicar se o nome do protótipo contiver esta string **não contém** - aplicar se o nome do protótipo não contiver esta string **corresponde** - aplicar se o nome do protótipo corresponder à expressão regular **não corresponde** - aplicar se o nome do protótipo não corresponder à expressão regular
Padrão	Uma expressão regular ou uma string para pesquisar.
Objeto: *Protótipo de item*	
Criar habilitado	Quando a caixa de seleção estiver marcada, os botões aparecerão, permitindo substituir as configurações originais do protótipo de item: *Sim* - o item será adicionado em estado habilitado. *Não* - o item será adicionado a uma entidade descoberta, mas em estado desabilitado.
Descobrir	Quando a caixa de seleção estiver marcada, os botões aparecerão, permitindo substituir as configurações originais do protótipo de item: *Sim* - o item será adicionado. *Não* - o item não será adicionado.
Intervalo de atualização	Quando a caixa de seleção estiver marcada, duas opções aparecerão, permitindo definir um intervalo diferente para o item: *Atraso* - Intervalo de atualização do item. Macros de usuário e sufixos de tempo (por exemplo, 30s, 1m, 2h, 1d) são suportados. Deve ser definido como 0 se *Intervalo personalizado* for usado. *Intervalo personalizado* - clique em Add para especificar intervalos flexíveis/agendados. Para informações detalhadas, consulte Intervalos personalizados.

Parâmetro	Descrição
Histórico	Quando a caixa de seleção estiver marcada, os botões aparecerão, permitindo definir um período de armazenamento de histórico diferente para o item: *Não armazenar* - se selecionado, o histórico não será armazenado. *Armazenar até* - se selecionado, um campo de entrada para especificar o período de armazenamento aparecerá à direita. Macros de usuário e macros LLD são suportados.
Tendências	Quando a caixa de seleção estiver marcada, os botões aparecerão, permitindo definir um período de armazenamento de tendências diferente para o item: *Não armazenar* - se selecionado, as tendências não serão armazenadas. *Armazenar até* - se selecionado, um campo de entrada para especificar o período de armazenamento aparecerá à direita. Macros de usuário e macros LLD são suportados.
Tags	Quando a caixa de seleção estiver marcada, um novo bloco aparecerá, permitindo especificar pares tag-valor. Essas tags serão adicionadas às tags especificadas no protótipo de item, mesmo que os nomes das tags coincidam.
Objeto: *Protótipo de trigger*	
Criar habilitado	Quando a caixa de seleção estiver marcada, os botões aparecerão, permitindo substituir as configurações originais do protótipo de trigger: *Sim* - o trigger será adicionado em estado habilitado. *Não* - o trigger será adicionado a uma entidade descoberta, mas em estado desabilitado.
Descobrir	Quando a caixa de seleção estiver marcada, os botões aparecerão, permitindo substituir as configurações originais do protótipo de trigger: *Sim* - o trigger será adicionado. *Não* - o trigger não será adicionado.
Gravidade	Quando a caixa de seleção estiver marcada, os botões de gravidade do trigger aparecerão, permitindo modificar a gravidade do trigger.
Tags	Quando a caixa de seleção estiver marcada, um novo bloco aparecerá, permitindo especificar pares tag-valor. Essas tags serão adicionadas às tags especificadas no protótipo de trigger, mesmo que os nomes das tags coincidam.
Objeto: *Protótipo de gráfico*	
Descobrir	Quando a caixa de seleção estiver marcada, os botões aparecerão, permitindo substituir as configurações originais do protótipo de gráfico: *Sim* - o gráfico será adicionado. *Não* - o gráfico não será adicionado.
Objeto: *Protótipo de host*	
Criar habilitado	Quando a caixa de seleção estiver marcada, os botões aparecerão, permitindo substituir as configurações originais do protótipo de host: *Sim* - o host será criado em estado habilitado. *Não* - o host será criado em estado desabilitado.
Descobrir	Quando a caixa de seleção estiver marcada, os botões aparecerão, permitindo substituir as configurações originais do protótipo de host: *Sim* - o host será descoberto. *Não* - o host não será descoberto.

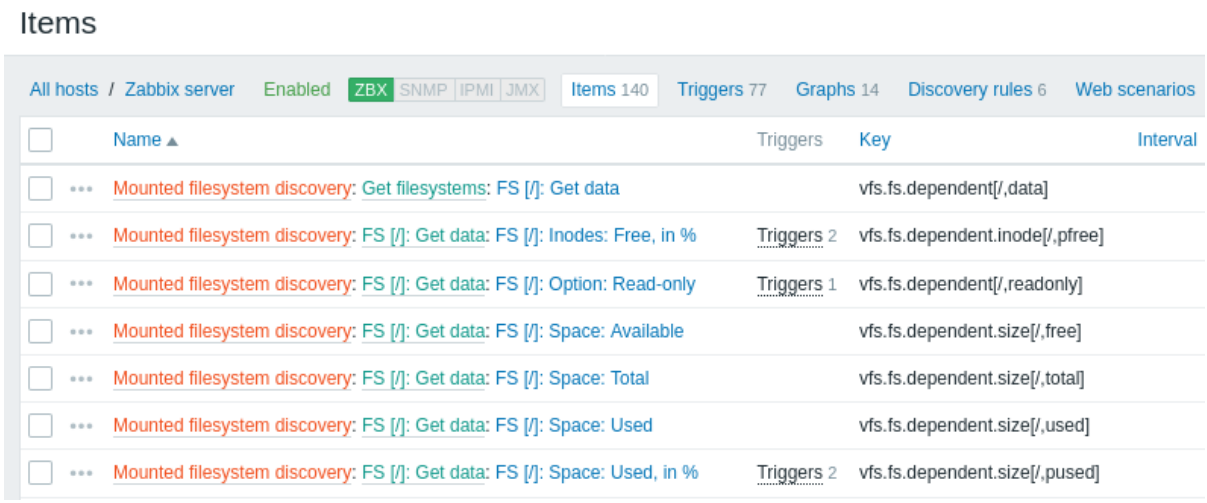
Parâmetro	Descrição
Vincular templates	Quando a caixa de seleção estiver marcada, um campo de entrada para especificar templates aparecerá. Comece a digitar o nome do template ou clique em *Selecionar* ao lado do campo e selecione os templates da lista em uma janela pop-up. Os templates desta substituição são adicionados a todos os templates já vinculados ao protótipo de host.
Tags	Quando a caixa de seleção estiver marcada, um novo bloco aparecerá, permitindo especificar pares tag-valor. Essas tags serão adicionadas às tags especificadas no protótipo de host, mesmo que os nomes das tags coincidam.
Inventário do host	Quando a caixa de seleção estiver marcada, os botões aparecerão, permitindo selecionar um modo de inventário diferente para o protótipo de host: *Desabilitado* - não preencher o inventário do host *Manual* - fornecer detalhes manualmente *Automático* - preencher automaticamente os dados do inventário do host com base nas métricas coletadas.

Botões do formulário

Os botões na parte inferior do formulário permitem realizar várias operações.

Add	Adiciona uma regra de descoberta. Este botão está disponível apenas para novas regras de descoberta.
Update	Atualiza as propriedades de uma regra de descoberta. Este botão está disponível apenas para regras de descoberta já existentes.
Clone	Cria outra regra de descoberta com base nas propriedades da regra de descoberta atual.
Execute now	Executa a descoberta com base na regra de descoberta imediatamente. A regra de descoberta já deve existir. Veja mais detalhes. *Observe* que ao executar a descoberta imediatamente, o cache de configuração não é atualizado, portanto, o resultado não refletirá alterações muito recentes na configuração da regra de descoberta.
Test	Testa a configuração da regra de descoberta. Use este botão para verificar as configurações (como conectividade e correção de parâmetros) sem aplicar permanentemente nenhuma alteração.
Delete	Exclui a regra de descoberta.
Cancel	Cancela a edição das propriedades da regra de descoberta.

Entidades descobertas As capturas de tela abaixo ilustram como os items, triggers e gráficos descobertos aparecem na configuração do host. As entidades descobertas são prefixadas com um link laranja para a regra de descoberta de onde vieram.



Observe que as entidades descobertas não serão criadas caso já existam entidades com os mesmos critérios de exclusividade, por exemplo, um item com a mesma chave ou um gráfico com o mesmo nome. Uma mensagem de erro é exibida neste caso no frontend informando que a regra de descoberta de baixo nível não pôde criar determinadas entidades. A própria regra de descoberta, no entanto, não ficará como não suportada porque alguma entidade não pôde ser criada e teve que ser ignorada. A regra de descoberta continuará criando/atualizando outras entidades.

Se uma entidade descoberta (host, sistema de arquivos, interface, etc) parar de ser descoberta (ou não passar mais no filtro), as entidades que foram criadas com base nela podem ser automaticamente desativadas e eventualmente excluídas.

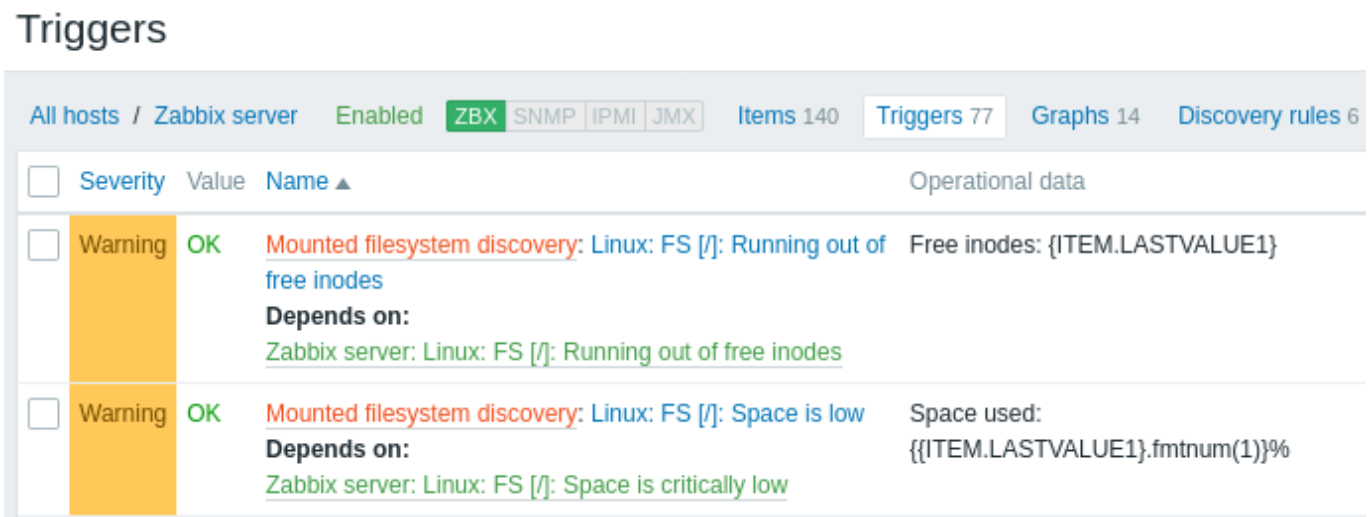
Recursos perdidos podem ser automaticamente desativados com base no valor do parâmetro *Desativar recursos perdidos*. Isso afeta hosts, items e triggers perdidos.

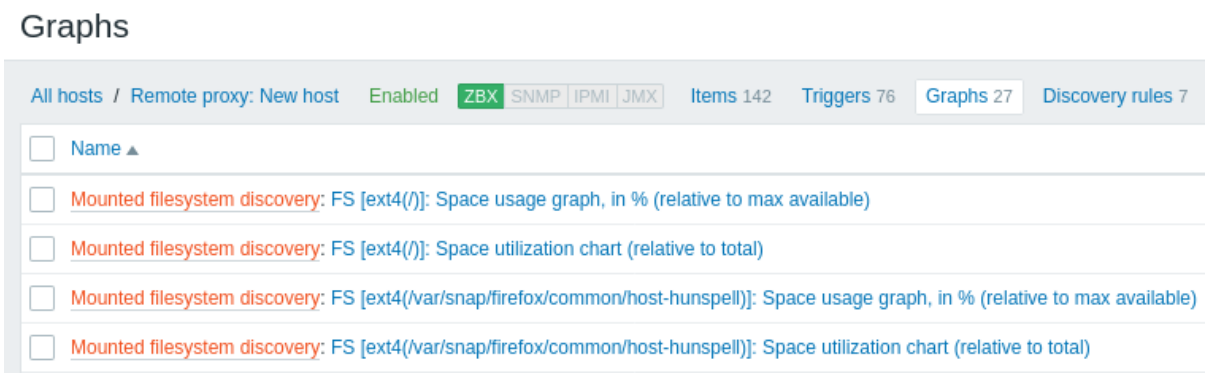
Recursos perdidos podem ser automaticamente excluídos com base no valor do parâmetro *Excluir recursos perdidos*. Isso afeta hosts, grupos de hosts, items, triggers e gráficos perdidos.

Quando entidades descobertas se tornam 'Não descobertas mais', um indicador de tempo de vida é exibido na lista de entidades. Mova o ponteiro do mouse sobre ele e uma mensagem será exibida indicando os detalhes do status.

Se as entidades foram marcadas para exclusão, mas não foram excluídas no momento esperado (regra de descoberta ou host do item desativado), elas serão excluídas na próxima vez que a regra de descoberta for processada.

Entidades que contêm outras entidades, que estão marcadas para exclusão, não serão atualizadas se forem alteradas no nível da regra de descoberta. Por exemplo, triggers baseados em LLD não serão atualizados se contiverem items que estão marcados para exclusão.





Outros tipos de descoberta Mais detalhes e tutoriais sobre outros tipos de descoberta prontos para uso estão disponíveis nas seguintes seções:

- descoberta de interfaces de rede
- descoberta de CPUs e núcleos de CPU
- descoberta de OIDs SNMP
- descoberta de objetos JMX;
- descoberta usando consultas SQL ODBC
- descoberta de serviços do Windows
- descoberta de interfaces de host no Zabbix

Para mais detalhes sobre o formato JSON para items de descoberta e um exemplo de como implementar seu próprio descobridor de sistema de arquivos como um script Perl, veja criando regras LLD personalizadas.

1 Protótipos de item

Uma vez que uma regra é criada, vá para os itens dessa regra e pressione "Criar protótipo de item" para criar um protótipo de item.

Observe como a macro {#FSNAME} é usada onde um nome de sistema de arquivos é necessário. O uso de uma macro de descoberta de baixo nível é obrigatório na chave do item para garantir que a descoberta seja processada corretamente. Quando a regra de descoberta for processada, essa macro será substituída pelo sistema de arquivos descoberto.

> **Note:**
> Quando os parâmetros da chave do item não são usados, coloque a macro LLD dentro dos colchetes da chave do item `[...]` (por exemplo, `v_[{#MACRO}]`).

Macros de descoberta de baixo nível e macros de usuário são suportadas na configuração do protótipo de item e nos parâmetros de pré-processamento do valor do item. Observe que, quando usadas em intervalos de atualização, uma única macro deve preencher todo o campo. Múltiplas macros em um campo ou macros misturadas com texto não são suportadas.

> **Note:**
> O escape específico para o contexto de macros de descoberta de baixo nível é realizado para uso seguro em parâmetros de pré-processamento de expressões regulares e XPath.

Atributos específicos para protótipos de item:

Parâmetro	Descrição
Criar habilitado	Se marcado, o item será adicionado em estado habilitado. Se desmarcado, o item será adicionado a uma entidade descoberta, mas em estado desabilitado.
Descobrir	Se marcado (padrão), o item será adicionado a uma entidade descoberta. Se desmarcado, o item não será adicionado a uma entidade descoberta, a menos que essa configuração seja substituída na regra de descoberta.

Podemos criar vários protótipos de item para cada métrica de sistema de arquivos que nos interessa:

Clique no ícone de três pontos para abrir o menu para o protótipo de item específico com estas opções:

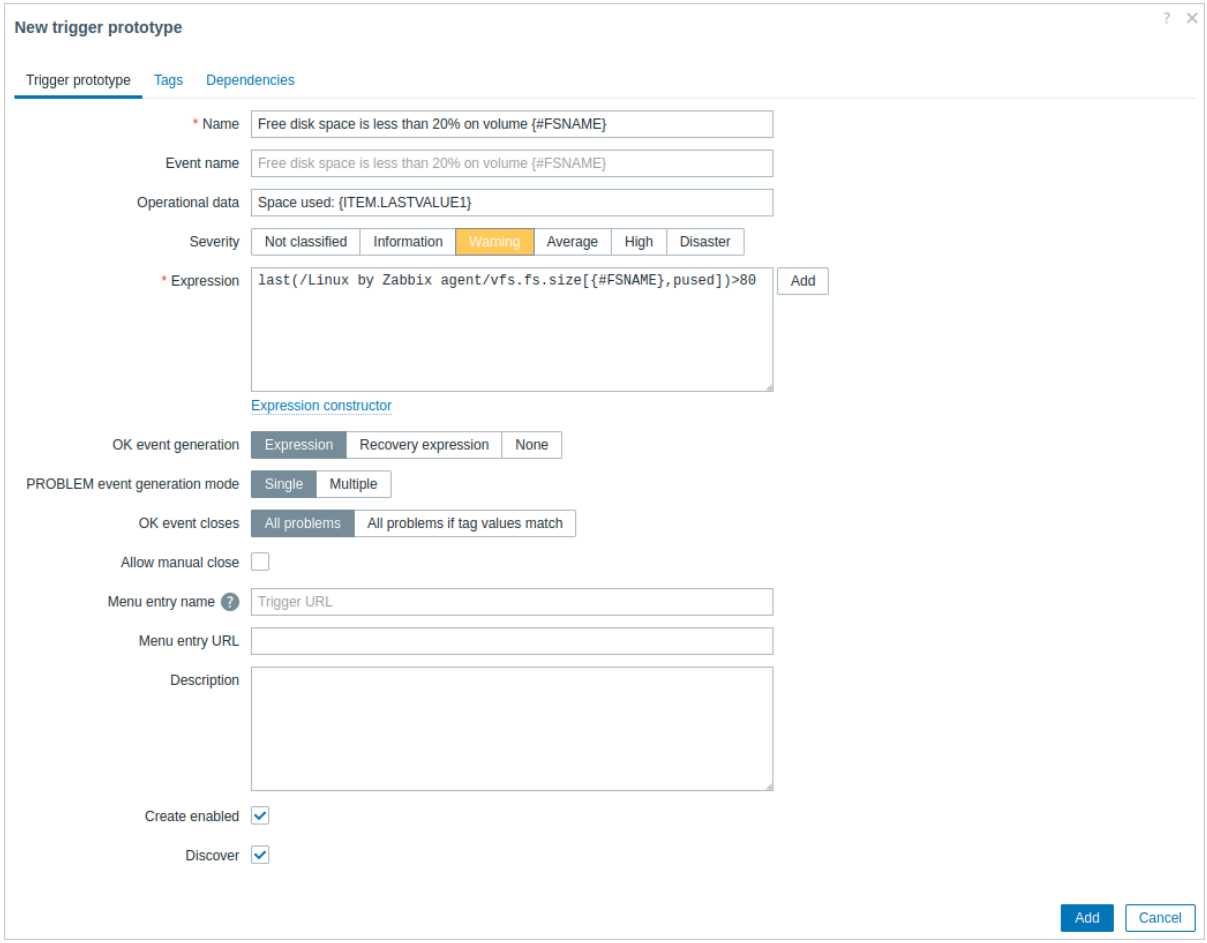
- *Criar protótipo de trigger* - criar um protótipo de trigger com base neste protótipo de item

- *Protótipos de trigger* - clique para ver uma lista com links para protótipos de trigger já configurados deste protótipo de item
- *Criar item dependente* - criar um item dependente para este protótipo de item

A opção de *Atualização em massa* está disponível se você quiser atualizar as propriedades de vários protótipos de item de uma só vez.

2 Protótipos de trigger

Criamos protótipos de trigger de forma semelhante aos protótipos de item:

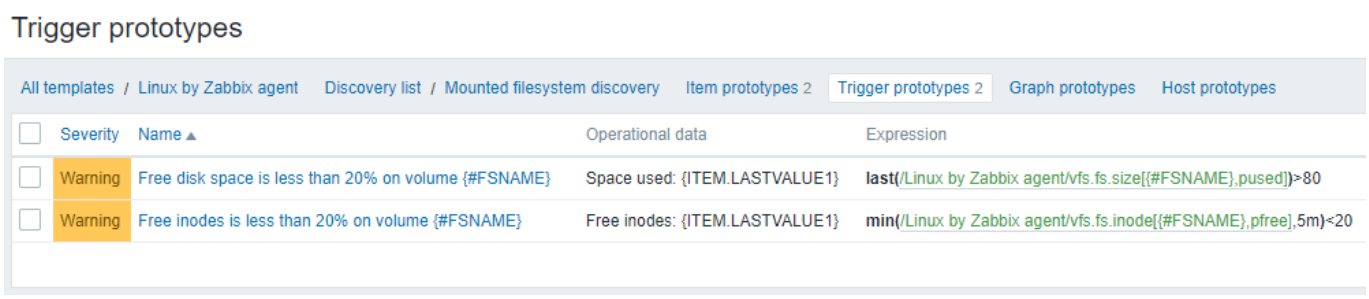


Atributos específicos para protótipos de trigger:

Parâmetro	Descrição
Criar habilitado	Se marcado, o trigger será adicionado em estado habilitado. Se desmarcado, o trigger será adicionado a uma entidade descoberta, mas em estado desabilitado.
Descobrir	Se marcado (padrão), o trigger será adicionado a uma entidade descoberta. Se desmarcado, o trigger não será adicionado a uma entidade descoberta, a menos que esta configuração seja substituída na regra de descoberta.

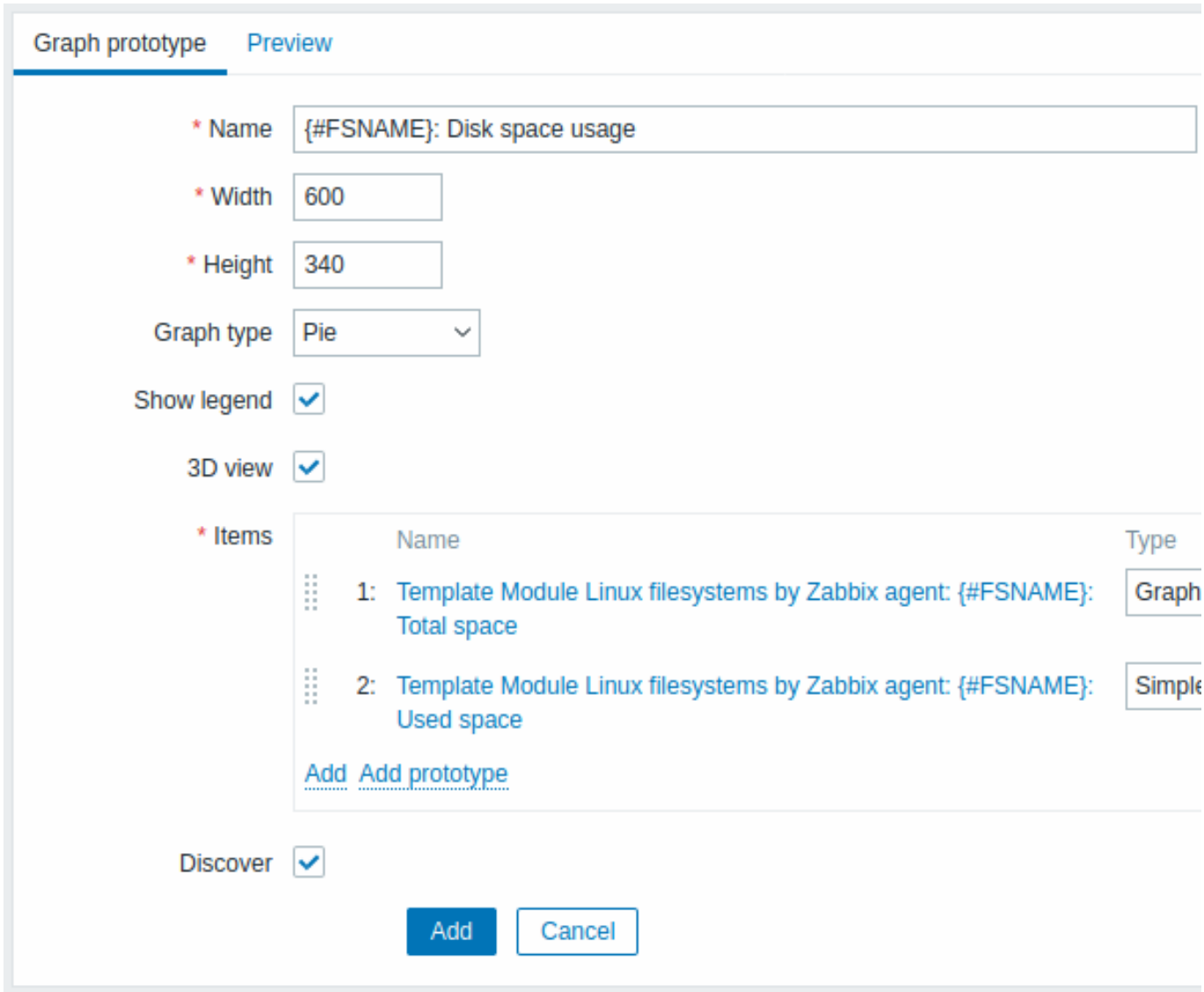
Quando triggers reais são criados a partir dos protótipos, pode ser necessário ser flexível quanto à constante ('20' em nosso exemplo) usada para comparação na expressão. Veja como macros de usuário com contexto podem ser úteis para alcançar tal flexibilidade.

Você pode definir dependências entre protótipos de trigger. Para isso, vá até a aba *Dependências*. Um protótipo de trigger pode depender de outro protótipo de trigger da mesma regra de descoberta de baixo nível (LLD) ou de um trigger regular. Um protótipo de trigger não pode depender de um protótipo de trigger de uma regra LLD diferente ou de um trigger criado a partir de um protótipo de trigger. O protótipo de trigger de host não pode depender de um trigger de um template.



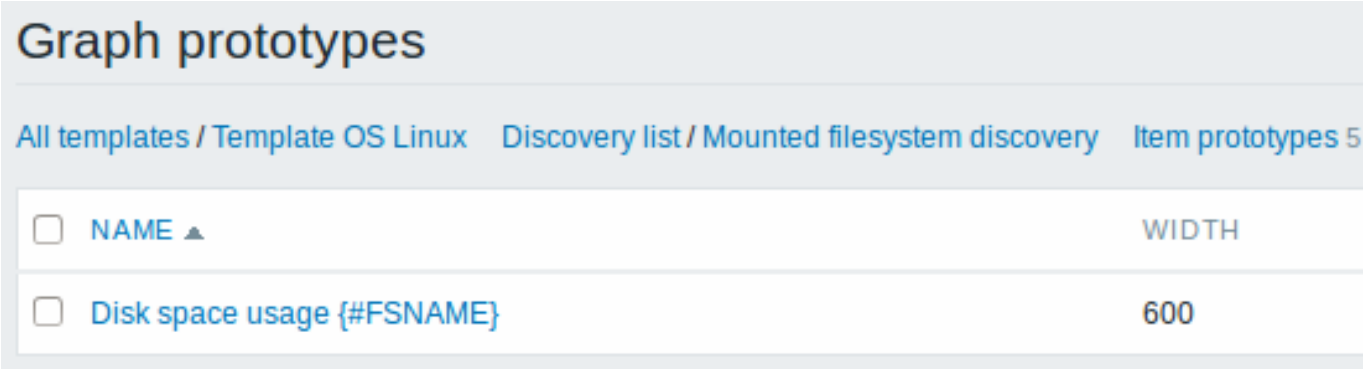
3 Protótipos de gráficos

Também podemos criar protótipos de gráficos:

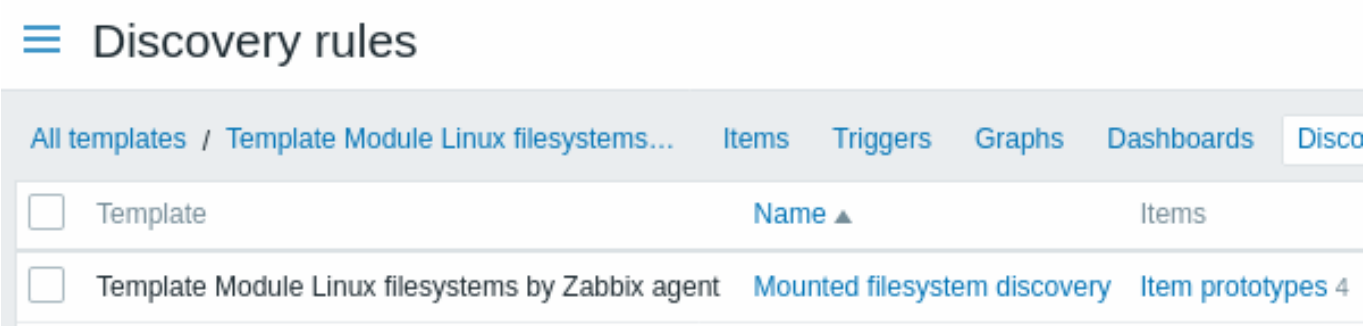


Atributos que são específicos para protótipos de gráficos:

Parâmetro	Descrição
Descobrir	Se marcado (padrão), o gráfico será adicionado a uma entidade descoberta. Se desmarcado, o gráfico não será adicionado a uma entidade descoberta, a menos que essa configuração seja substituída na regra de descoberta.



Por fim, criamos uma regra de descoberta que fica como mostrado abaixo. Ela possui cinco protótipos de item, dois protótipos de trigger e um protótipo de gráfico.



4 Protótipos de host

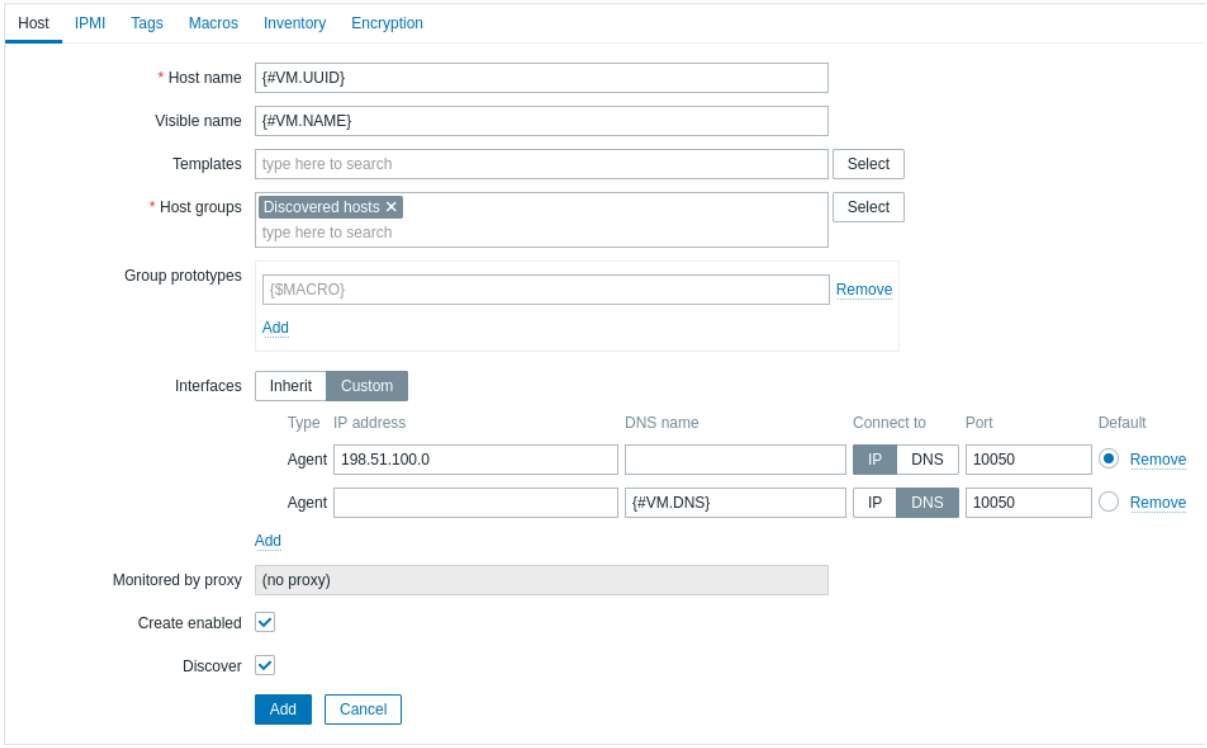
Protótipos de host são modelos para criar hosts por meio de regras de descoberta de baixo nível. Antes de serem descobertos como hosts, esses protótipos não podem ter items e triggers, exceto aqueles vinculados a partir de templates.

Configuração

Protótipos de host são configurados em regras de descoberta de baixo nível.

Para criar um protótipo de host:

1. Vá para *Coleta de dados → Hosts*.
2. Clique em *Descoberta* para o host desejado para navegar até a lista de regras de descoberta de baixo nível configuradas para esse host.
3. Clique em *Protótipos de host* para a regra de descoberta desejada.
4. Clique no botão *Criar protótipo de host* no canto superior direito.



Protótipos de host possuem os mesmos parâmetros que hosts regulares; no entanto, os seguintes parâmetros suportam configurações diferentes ou adicionais:

Parâmetro	Descrição
Nome do host	Este parâmetro deve conter pelo menos uma macro de descoberta de baixo nível para garantir nomes de host exclusivos para os hosts criados.
Nome visível	Macros de descoberta de baixo nível são suportadas.
Protótipos de grupo	Permite especificar protótipos de grupo de host usando macros de descoberta de baixo nível. Com base nos protótipos de grupo especificados, grupos de host serão descobertos, criados e vinculados aos hosts criados; grupos descobertos que já tenham sido criados por outras regras de descoberta de baixo nível também serão vinculados aos hosts criados. No entanto, grupos de host descobertos que correspondam a grupos de host criados manualmente não serão vinculados aos hosts criados.
Interfaces	Defina se os hosts descobertos herdam o IP do host ao qual a regra de descoberta pertence (padrão) ou obtêm interfaces personalizadas. Macros de descoberta de baixo nível e macros de usuário são suportadas.
Criar habilitado	Defina o status dos hosts descobertos; se desmarcado, os hosts serão criados como desabilitados.
Descobrir	Defina se os hosts serão criados a partir do protótipo de host; se desmarcado, os hosts não serão criados a partir do protótipo de host (a menos que esta configuração seja substituída na regra de descoberta de baixo nível).

Note:
Macros de descoberta de baixo nível também são suportadas para valores de tags e valores de macros de usuário de protótipo de host.
 Mapas de valor não são suportados para protótipos de host.

Para um exemplo de como configurar um protótipo de host, consulte *Monitoramento de máquina virtual*.

Interfaces do host

Para adicionar interfaces personalizadas, altere o seletor *Interfaces* de ”Herdar” para ”Personalizado”. Clique em Add e selecione o tipo de interface - Zabbix agent, SNMP, JMX, IPMI.

Note:
Se *Personalizado* for selecionado, mas nenhuma interface tiver sido definida, os hosts serão criados sem interfaces.
 Se *Herdar* for selecionado e o protótipo de host pertencer a um template, todos os hosts descobertos herdarão a interface do host ao qual o template está vinculado.

Se várias interfaces personalizadas forem especificadas, a interface primária pode ser definida na coluna *Padrão*.

Para um exemplo de como configurar uma interface de host personalizada, consulte *Exemplo de configuração de monitoramento VMware*.

Warning:
Um host só será criado se uma interface de host contiver dados corretos.

Hosts descobertos

Na lista de hosts, os hosts descobertos são prefixados com o nome da regra de descoberta que os criou.

Os hosts descobertos herdam a maioria dos parâmetros dos protótipos de host como *somente leitura*. Apenas os seguintes parâmetros de host descobertos podem ser configurados:

- *Templates* - vincular novos templates ou desvincular aqueles adicionados manualmente. Os templates herdados de um protótipo de host não podem ser desvinculados.
- *Descrição* - adicionar ou editar a descrição do host.
- *Status* - habilitar ou desabilitar o host.
- *Tags* - adicionar novas tags ou remover aquelas adicionadas manualmente. As tags herdadas de um protótipo de host não podem ser removidas. Observe que as tags não podem ser duplicadas (tags com o mesmo nome e valor). Se uma nova tag for adicionada ao protótipo de host e corresponder a uma tag adicionada manualmente em um host descoberto, a tag adicionada manualmente será substituída pela tag herdada durante a descoberta.
- *Macros* - adicionar novas macros de host ou remover aquelas adicionadas manualmente, bem como as herdadas de um protótipo de host; alterar valores e tipos de macros.

Note:
Os hosts descobertos também herdam macros de usuário do host onde a regra de descoberta está configurada. Essas macros herdadas podem ser removidas, e seus valores e tipos podem ser alterados.

Os hosts descobertos podem ser excluídos manualmente. Observe, no entanto, que eles serão descobertos novamente se a descoberta estiver habilitada para eles.

Hosts que não são mais descobertos podem ser:

- desabilitados automaticamente (com base no valor *Desabilitar recursos perdidos* da regra de descoberta)
- excluídos automaticamente (com base no valor *Excluir recursos perdidos* da regra de descoberta).

Note:
O Zabbix não suporta protótipos de host aninhados, ou seja, protótipos de host em hosts descobertos por regras de descoberta de baixo nível.

5 Notas sobre descoberta de baixo nível

Usando macros LLD em contextos de macro de usuário

As macros LLD podem ser usadas dentro do contexto da macro do usuário, por exemplo, em acionar protótipos.

Várias regras de LLD para o mesmo item

É possível definir várias regras de descoberta de baixo nível com o mesmo item de descoberta.

Para isso, você precisa definir o parâmetro Alias do agent parâmetro, permitindo usar chaves de item de descoberta alteradas em diferentes regras de descoberta, por exemplo `vfs.fs.discovery[foo]`, `vfs.fs.discovery[bar]`, etc.

Limites de dados para valores de retorno

Não há limite para os dados JSON da regra de descoberta de baixo nível se forem recebidos diretamente pelo Zabbix server. Isso ocorre porque os valores de retorno são processados sem serem armazenados em um banco de dados.

Também não há limite para regras de descoberta de baixo nível personalizadas. No entanto, se os dados da regra de descoberta de baixo nível personalizada forem recuperados usando um parâmetro de usuário, o limite de valor de retorno do parâmetro de usuário se aplica.

Se os dados tiverem que passar pelo Zabbix proxy, eles terão que ser armazenados no banco de dados. Nesse caso, os limites do banco de dados se aplicam.

6 Regras de descoberta

Use a barra lateral para ver exemplos de configuração de regras de descoberta para vários casos.

1 Descoberta de sistemas de arquivos montados

Visão geral

É possível descobrir sistemas de arquivos montados e suas propriedades:

- nome do ponto de montagem
- tipo de sistema de arquivos
- tamanho do sistema de arquivos
- estatísticas de inode
- opções de montagem

Para isso, você pode usar uma combinação de:

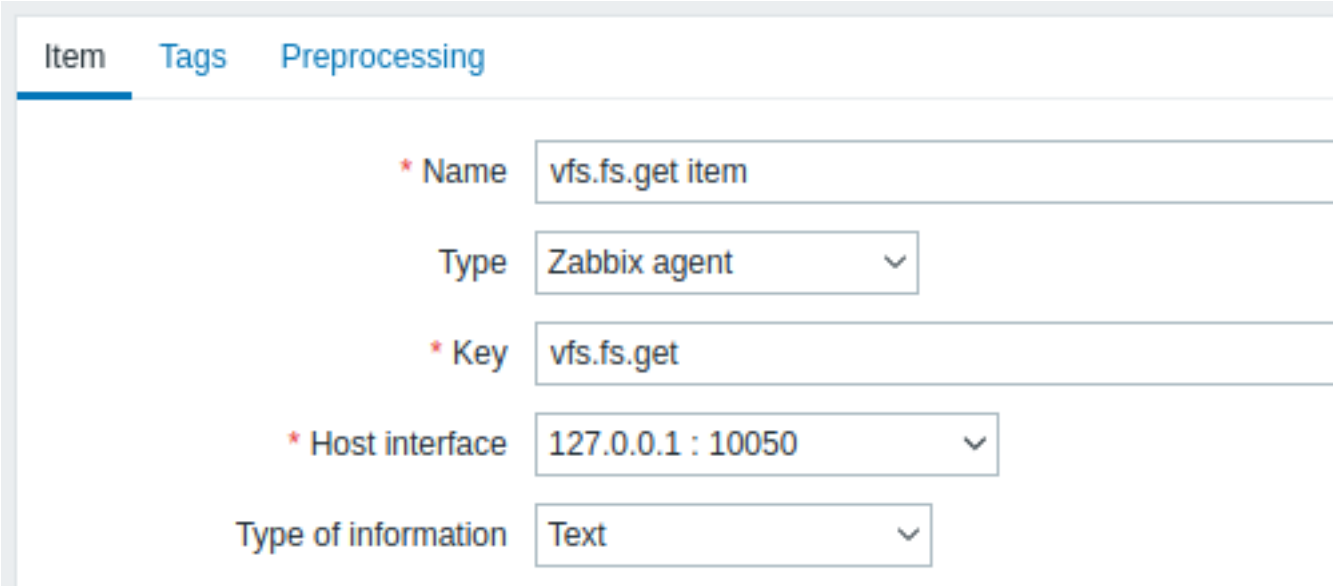
- o item agent `vfs.fs.get` como item mestre
- regra de descoberta de baixo nível dependente e protótipos de item

Comfiguração

Item mestre

Crie um item do agent Zabbix usando a seguinte chave:

`vfs.fs.get`



Defina o tipo de informação como ”Texto” para possíveis grandes dados JSON.

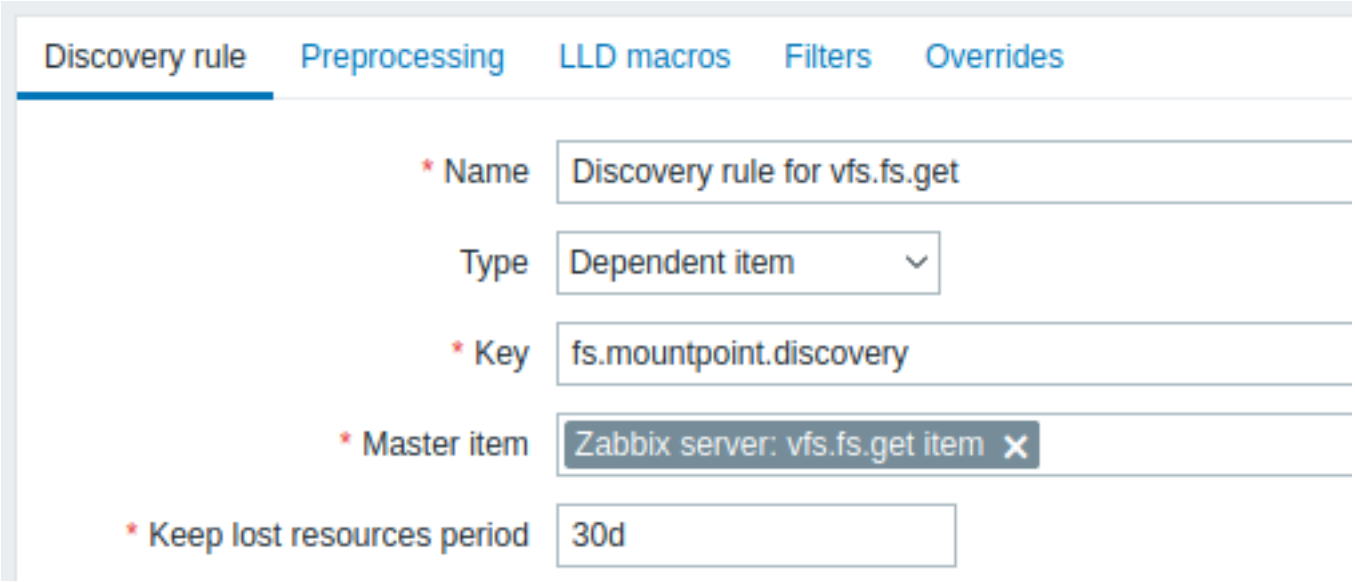
Os dados retornados por este item conterão algo como o seguinte para um sistema de arquivos montado:

```
[
  {
    "fsname": "/",
    "fstype": "ext4",
    "bytes": {
      "total": 249405239296,
      "free": 24069537792,
      "used": 212595294208,
      "pfree": 10.170306,
      "pused": 89.829694
    },
    "inodes": {
      "total": 15532032,
      "free": 12656665,
      "used": 2875367,
```

```
      "pfree": 81.487503,
      "pused": 18.512497
    },
    "options": "rw,noatime,errors=remount-ro"
  }
]
```

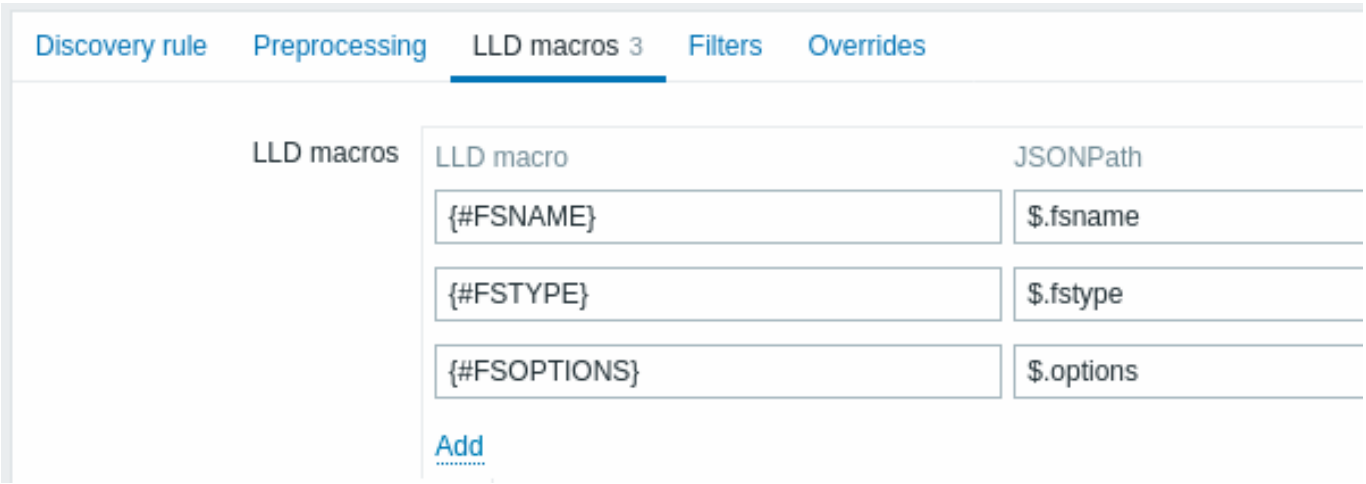
Regra LLD dependente

Crie uma regra de descoberta de baixo nível do tipo ”item dependente”:

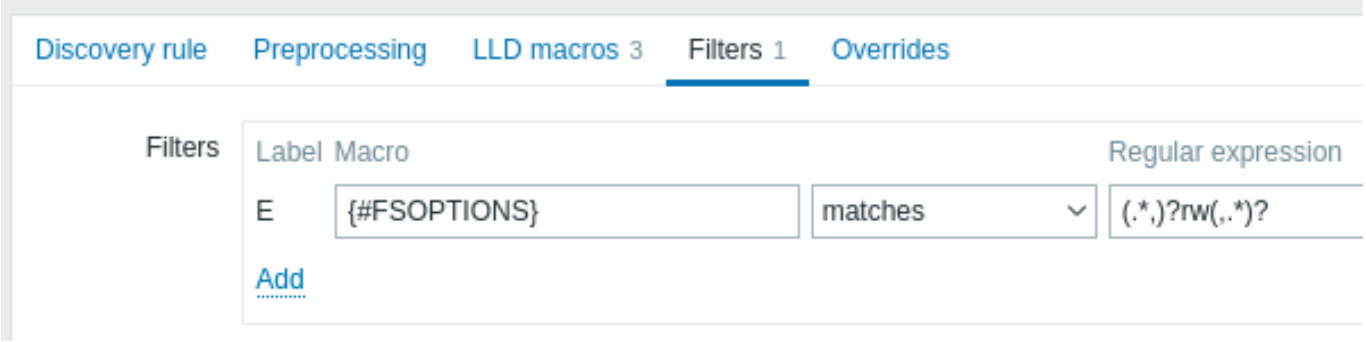


Como item mestre, selecione o item `vfs.fs.get` que criamos.

Na aba ”Macros LLD”, defina macros personalizadas com o respectivo JSONPath:

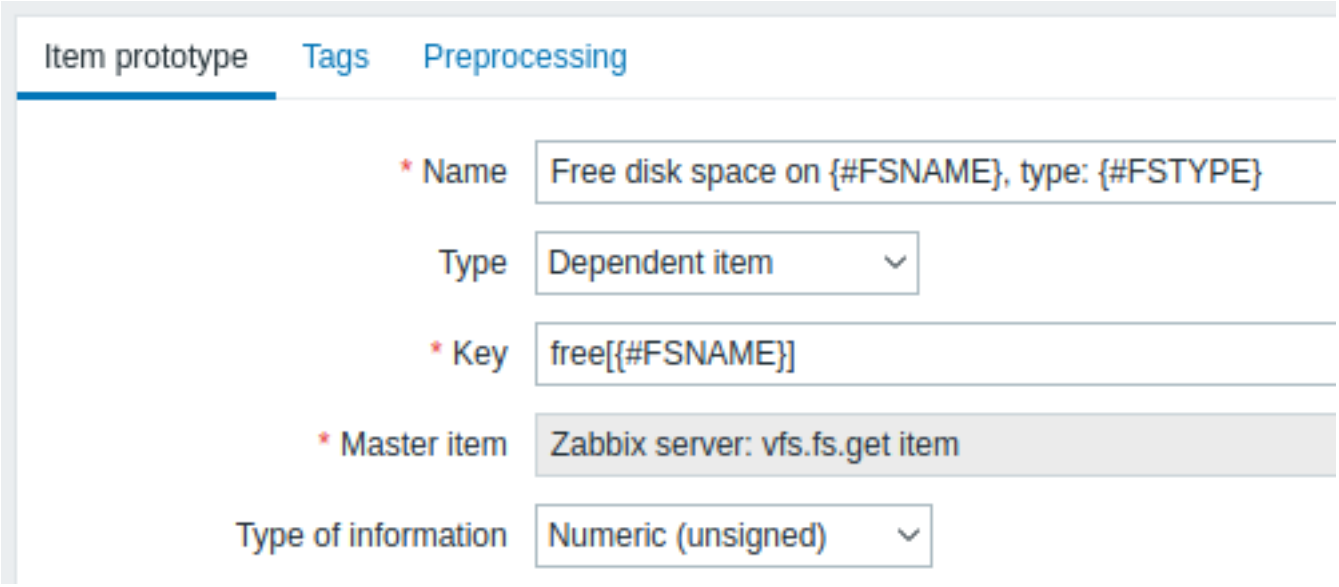


Na aba ”Filtros”, você pode adicionar uma expressão regular que filtra apenas sistemas de arquivos **leitura-escrita**:



Protótipo de item dependente

Crie um protótipo de item com o tipo ”Item dependente” nesta regra LLD. Como item mestre para este protótipo selecione o item `vfs.fs.get` que criamos.



Observe o uso de macros personalizadas no nome e na chave do protótipo do item:

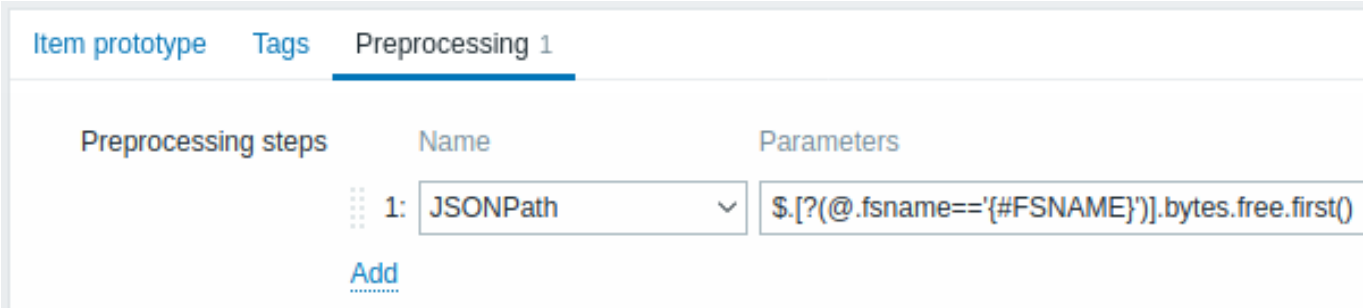
- *Nome*: Espaço livre em disco em {#FSNAME}, digite: {#FSTYPE}
- *Chave*: Grátis[{#FSNAME}]

Como tipo de informação, use:

- *Numérico (sem sinal)* para métricas como ’gratuito’, ’total’, ’usado’
- *Numeric (float)* para métricas como ’pfree’, ’pused’ (porcentagem)

Na guia ”Pré-processamento” do protótipo do item, selecione JSONPath e use o seguinte expressão JSONPath como parâmetro:

`$.[?(@.fsname=='{#FSNAME}')].bytes.free.first()`



Quando a descoberta for iniciada, será criado um item por cada ponto de montagem. Este item retornará o número de bytes livres para o ponto de montagem fornecido.

2 Descoberta de interfaces de rede

De maneira semelhante ao arquivo sistemas são descobertos, é possível também descobrir interfaces de rede.

Chave do item

A chave do item a ser usada na regra de descoberta é

`net.if.discovery`

Macros compatíveis

Você pode usar a macro {#IFNAME} na regra de descoberta filtro e protótipos de itens, gatilhos e gráficos.

Exemplos de protótipos de itens que você pode querer criar com base em ”net.if.discovery”:

- ”net.if.in[{#IFNAME},bytes]”,
- ”net.if.out[{#IFNAME},bytes]”.

Observe que no Windows {#IFGUID} também é retornado.

3 Descoberta de CPUs e núcleos de CPU

De maneira semelhante ao arquivo sistemas são descobertos, é possível descobrir também CPUs e núcleos de CPU.

Chave do item

A chave do item a ser usada na regra de descoberta é

`system.cpu.discovery`

Macros suportadas

Esta chave de descoberta retorna duas macros - {#CPU.NUMBER} e {#CPU.STATUS} identificando o número de ordem da CPU e o status, respectivamente. Ela enumera os processadores de 0 a N - 1, onde N é o número total de processadores.

```
[
    {
        "{#CPU.NUMBER}": 0,
        "{#CPU.STATUS}": "online"
    },
    {
        "{#CPU.NUMBER}": 1,
        "{#CPU.STATUS}": "offline"
    },
    {
        "{#CPU.NUMBER}": 2,
        "{#CPU.STATUS}": "unknown" /* "unknown" só existe no Windows */
    },
    {
        "{#CPU.NUMBER}": 3,
        "{#CPU.STATUS}": "online"
    }
]
```

Observe que não é possível fazer uma distinção clara entre processadores físicos reais, núcleos e hyperthreads. {#CPU.STATUS} em sistemas Linux, UNIX e BSD retorna o status do processador, que pode ser ”online” ou ”offline”. Em sistemas Windows, essa mesma macro pode representar um terceiro valor - ”unknown” - que indica que um processador foi detectado, mas ainda não foram coletadas informações para ele.

A descoberta de CPU depende do processo coletor do agent para permanecer consistente com os dados fornecidos pelo coletor e economizar recursos na obtenção dos dados. Isso faz com que essa chave de item não funcione com o parâmetro de linha de comando de teste (-t) do binário do agent, que retornará um status NOT_SUPPORTED e uma mensagem indicando que o processo coletor não foi iniciado.

Protótipos de item que podem ser criados com base na descoberta de CPU incluem, por exemplo:

- `system.cpu.util[{#CPU.NUMBER},<type>,<mode>]`
- `system.hw.cpu[{#CPU.NUMBER},<info>]`

Para uma descrição detalhada da chave de item, consulte Chaves de item do Zabbix agent.

Observações sobre itens de contadores de desempenho descobertos em sistemas NUMA

Alguns contadores de desempenho do Windows podem não estar disponíveis para alguns dos processadores lógicos em sistemas NUMA.

Por exemplo, itens descobertos com um protótipo de item com a chave abaixo podem funcionar apenas para o primeiro nó NUMA. Itens para os outros nós NUMA podem estar em estado não suportado.

`perf_counter[\Processor({#CPU.NUMBER})\% Processor Time,60]`

Além disso, uma regra de descoberta com a chave ”system.cpu.discovery” não fornece nenhum valor de macro de descoberta de baixo nível para substituir nesses contadores de desempenho, como:

`perf_counter["\Processor Information(<NUMA node index>,<CPU index in NUMA node>)\% Processor Time",60]`

Enquanto os itens system.cpu.util dependem de diferentes contadores de desempenho para monitorar a porcentagem de utilização da CPU, dependendo da contagem de grupos de processadores e da contagem de processadores. Além disso, existe um bug raro relacionado a grupos de processadores no Windows.

4 Descoberta de OIDs SNMP

Visão geral

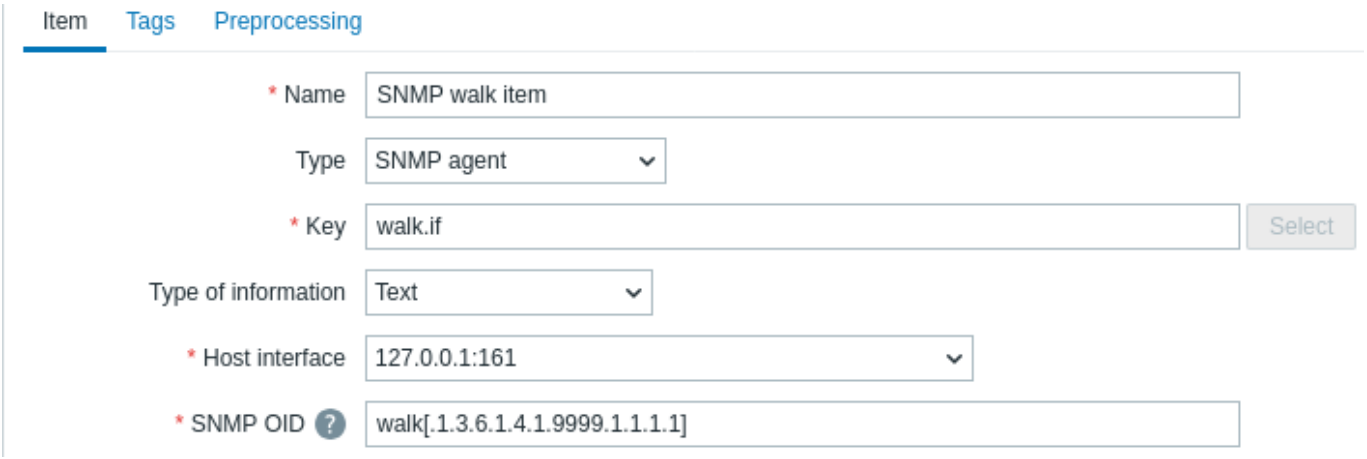
Nesta seção, realizaremos uma descoberta de baixo nível em um dispositivo SNMP.

Esse método de descoberta de OIDs SNMP é suportado desde o Zabbix server/proxy 6.4.

Configuração de exemplo

1. Crie um item de agente SNMP com uma chave como:

```
walk[.1.3.6.1.4.1.9999.1.1.1.1]
```



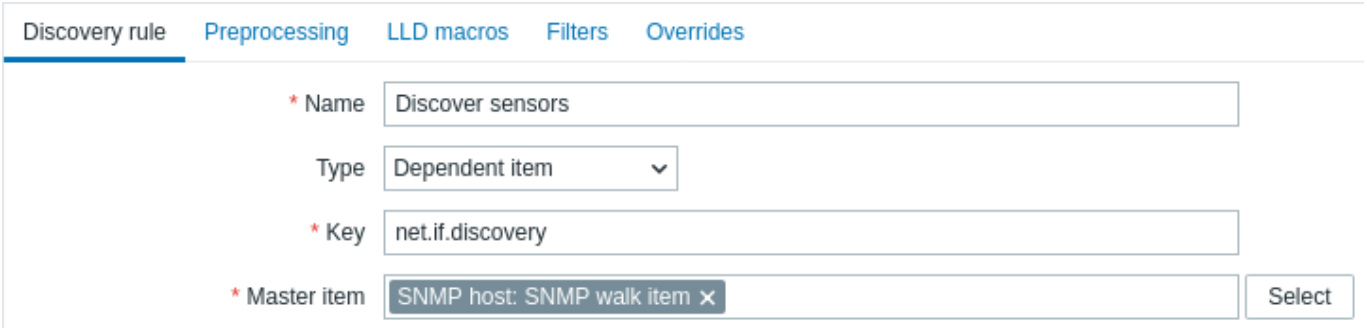
Este item executa uma única varredura de tabela SNMP e retorna todas as entradas da tabela em uma única solicitação, em um formato que corresponde à saída do utilitário `snmpwalk` com as opções de formatação `-Oe -Ot -On`.

Ele retornará o seguinte valor de texto multilinha:

```
.1.3.6.1.4.1.9999.1.1.1.1.1.1 = STRING: "Temperature Sensor"
.1.3.6.1.4.1.9999.1.1.1.1.2.1 = STRING: "temp"
.1.3.6.1.4.1.9999.1.1.1.1.3.1 = 100
.1.3.6.1.4.1.9999.1.1.1.1.1.2 = STRING: "Humidity Sensor"
.1.3.6.1.4.1.9999.1.1.1.1.2.2 = STRING: "humidity"
.1.3.6.1.4.1.9999.1.1.1.1.3.2 = 200
```

2. Crie uma regra de descoberta:

- No campo *Nome*, insira um nome descritivo para a regra de descoberta (por exemplo, ”Descobrir sensores”).
- No campo *Tipo*, selecione ”Item dependente”.
- No campo *Chave*, insira uma chave descritiva (por exemplo, ”net.if.discovery”).
- No campo *Item mestre*, selecione ”SNMP walk item”.



3. Na guia *Pré-processamento*, adicione uma etapa de pré-processamento com o nome ”SNMP walk to JSON” no menu suspenso *Nome* com 3 parâmetros:

- *Nome do campo*: ”{#SENSORNAME}”; *Prefixo OID*: ”.1.3.6.1.4.1.9999.1.1.1.1.1”: *Formato*: ”Inalterado”.
- *Nome do campo*: ”{#SENSORTYPE}”; *Prefixo OID*: ”.1.3.6.1.4.1.9999.1.1.1.1.2”: *Formato*: ”Inalterado”.
- *Nome do campo*: ”{#SENSORVALUE}”; *Prefixo OID*: ”.1.3.6.1.4.1.9999.1.1.1.1.3”: *Formato*: ”Inalterado”.

Após o pré-processamento, a regra de descoberta retorna um array JSON de conjuntos de macros.

Por exemplo:

```
[
    {
        "{#SNMPINDEX}": "1",
        "{#SENSORNAME}": "Temperature Sensor",
        "{#SENSORTYPE}": "temp",
        "{#SENSORVALUE}": "100"
    },
    {
        "{#SNMPINDEX}": "2",
        "{#SENSORNAME}": "Humidity Sensor",
        "{#SENSORTYPE}": "humidity",
        "{#SENSORVALUE}": "200"
    }
]
```

Cada objeto representa um sensor descoberto e fornece macros como `{#SNMPINDEX}`, `{#SENSORNAME}`, `{#SENSORTYPE}` e `{#SENSORVALUE}`.

Eles são agrupados pelo índice SNMP, que é o sufixo numérico no final de cada OID (por exemplo, .1, .2) — esse índice identifica exclusivamente cada linha na tabela SNMP e é extraído automaticamente como `{#SNMPINDEX}`.

4. Na regra de descoberta, crie um ou mais protótipos de item (com a regra de descoberta como item mestre).

Por exemplo, item dependente do valor do sensor:

- No campo *Nome*, insira "Sensor {#SNMPINDEX}: {#SENSORNAME}".
- No campo *Tipo*, selecione "Item dependente".
- No campo *Chave*, insira "sensor.value[{#SNMPINDEX}]".
- No campo *Item mestre*, selecione "SNMP walk item".

Na guia *Pré-processamento*, adicione uma etapa de pré-processamento com o nome "SNMP walk value" com o OID ".1.3.6.1.4.1.9999.1.1.1.1.3.{#SNMPINDEX}" no campo *Parâmetro*. *Formato*: "Inalterado".

Os seguintes itens serão descobertos:

Nome	Chave	OID de onde o valor é extraído	Valor do item
Sensor 1: Temperature Sensor	sensor.value[1]	1.3.6.1.4.1.9999.1.1.1.1.3.1	100
Sensor 2: Humidity Sensor	sensor.value[2]	1.3.6.1.4.1.9999.1.1.1.1.3.2	200

Quando a regra de descoberta é executada, itens como `sensor.value[1]`, `sensor.value[2]` são criados.

Cada item dependente extrai seu valor do resultado do SNMP walk do item mestre usando pré-processamento, sem executar solicitações SNMP separadas.

5. Referencie protótipos de itens dependentes em protótipos de trigger usando as mesmas macros da regra de descoberta. Exemplo:

```
{Template_Sensor:sensor.value[{#SNMPINDEX}].last()} > 75
```

Isso produz um trigger para cada sensor descoberto (por exemplo, sensor.value[1], sensor.value[2]) e dispara se o valor mais recente (temperatura ou umidade) exceder 75.

6. Inclua itens dependentes para cada entidade descoberta. Exemplo de chave de item de gráfico:

```
sensor.value[{#SNMPINDEX}]
```

Um gráfico é criado por {#SNMPINDEX}, plotando temperatura e umidade ao longo do tempo.

Esta configuração executa apenas uma única solicitação SNMP walk por ciclo de coleta, independentemente do número de itens descobertos. Todos os itens dependentes extraem seus valores do resultado do SNMP walk mestre usando pré-processamento, reduzindo significativamente o tráfego e a carga SNMP.

Índices dinâmicos com walk[]

Os índices dinâmicos (por exemplo, índices de interface) podem mudar quando o hardware é reconfigurado. Para acomodar esse comportamento, uma regra de descoberta mestre SNMP walk é criada com uma chave como:

```
walk[1.3.6.1.2.1.2.2.1.10]
```

Após o pré-processamento SNMP walk para JSON, o resultado pode se parecer com:

```
[
    {
        "{#SNMPINDEX}": "2",
        "{#VALUE}": "123456"
    },
    {
        "{#SNMPINDEX}": "3",
        "{#VALUE}": "654321"
    }
]
```

Um protótipo de item dependente usa a macro {#SNMPINDEX} para construir a chave:

```
net.if.in[{#SNMPINDEX}]
```

O pré-processamento para esse protótipo inclui o nome ”SNMP walk value” com o OID ”1.3.6.1.2.1.2.2.1.10.{#SNMPINDEX}” no campo *Parameter*. *Format*: ”Unchanged”.

Em tempo de execução, itens reais como net.if.in[2] e net.if.in[3] são criados. Se um determinado índice de interface mudar (por exemplo, se o índice 2 for substituído por 5 na tabela SNMP), então na próxima execução da regra de descoberta:

- O antigo item dependente net.if.in[2] é marcado como ”perdido” ou removido, e nenhum novo dado é coletado para esse item.
- Um novo item dependente net.if.in[5] é criado, começando com um histórico vazio.
- Os dados históricos de net.if.in[2] não são movidos automaticamente para net.if.in[5].

Exemplo de protótipo de trigger:

```
{Template_Interface:net.if.in[{#SNMPINDEX}].last()} > 1000000000
```

Exemplo de protótipo de gráfico inclui itens:

```
net.if.in[{#SNMPINDEX}]
net.if.out[{#SNMPINDEX}]
```

Essa configuração garante o monitoramento confiável de tabelas com índices dinâmicos, minimizando o tráfego SNMP—apenas um único SNMP walk por ciclo de coleta é necessário, com protótipos de itens dependentes extraindo os valores necessários.

Entidades descobertas

Quando o server é executado, ele criará itens dependentes reais, triggers e gráficos com base nos valores que a regra de descoberta SNMP retorna.

5 Descoberta de OIDs SNMP (legado)

Visão geral

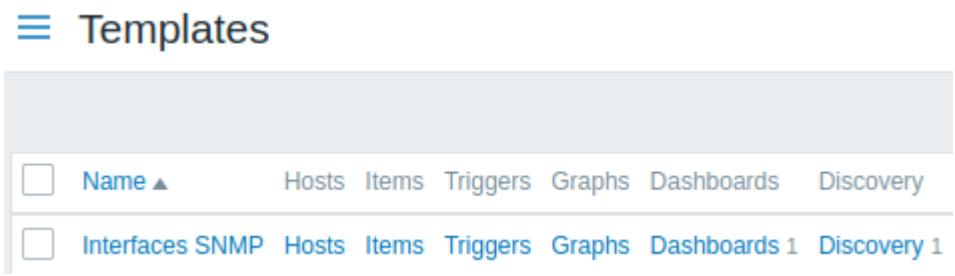
Nesta seção vamos realizar um SNMP discovery em um switch.

Chave do item

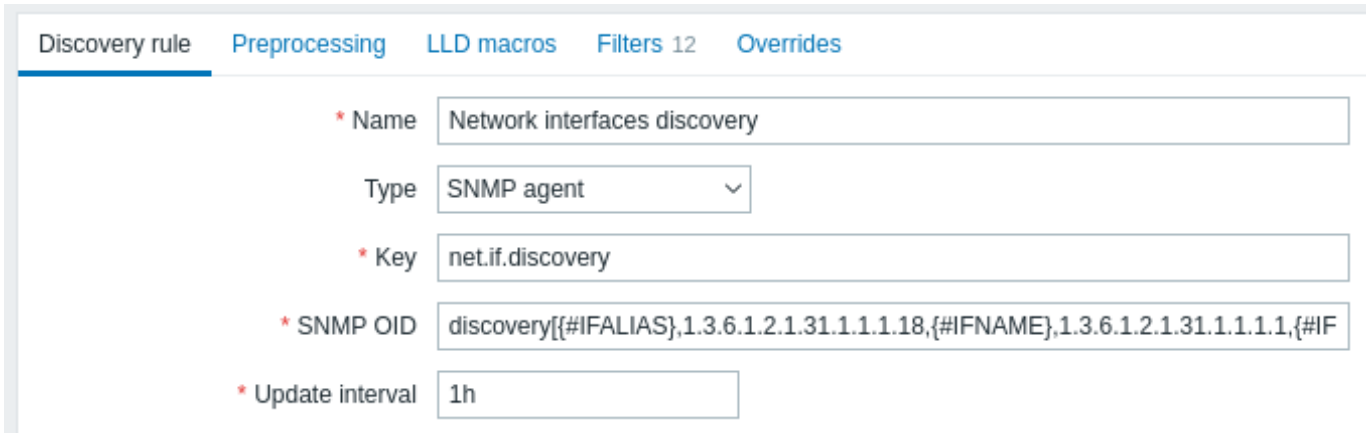
Diferentemente da descoberta de sistema de arquivos e de interface de rede, o item não precisa necessariamente ter uma chave "snmp.discovery" - o tipo de item SNMP agent é suficiente.

Para configurar a regra de descoberta, faça o seguinte:

- Vá para: *Coleta de dados* → *Templates*
- Clique em *Descoberta* na linha do template apropriado



- Clique em *Criar regra de descoberta* no canto superior direito da tela
- Preencha o formulário da regra de descoberta com os detalhes necessários, conforme a captura de tela abaixo



Todos os campos obrigatórios estão marcados com um asterisco vermelho.

Os OIDs a serem descobertos são definidos no campo SNMP OID no seguinte formato: `discovery[{#MACRO1}, oid1, {#MACRO2}, oid2, …,]`

onde *{#MACRO1}*, *{#MACRO2}* ... são nomes de macros lld válidos e *oid1*, *oid2*... são OIDs capazes de gerar valores significativos para essas macros. Uma macro interna *{#SNMPINDEX}* contendo o índice do OID descoberto é aplicada às entidades descobertas. As entidades descobertas são agrupadas pelo valor da macro *{#SNMPINDEX}*.

Para entender o que queremos dizer, vamos executar alguns snmpwalks em nosso switch:

```
$ snmpwalk -v 2c -c public 192.168.1.1 IF-MIB::ifDescr
IF-MIB::ifDescr.1 = STRING: WAN
IF-MIB::ifDescr.2 = STRING: LAN1
IF-MIB::ifDescr.3 = STRING: LAN2

$ snmpwalk -v 2c -c public 192.168.1.1 IF-MIB::ifPhysAddress
IF-MIB::ifPhysAddress.1 = STRING: 8:0:27:90:7a:75
IF-MIB::ifPhysAddress.2 = STRING: 8:0:27:90:7a:76
IF-MIB::ifPhysAddress.3 = STRING: 8:0:27:2b:af:9e
```

E definir o SNMP OID para: `discovery[{#IFDESCR}, ifDescr, {#IFPHYSADDRESS}, ifPhysAddress]`

Agora esta regra irá descobrir entidades com macros {#IFDESCR} definidas como **WAN**, **LAN1** e **LAN2**, macros {#IFPHYSADDRESS} definidas como **8:0:27:90:7a:75**, **8:0:27:90:7a:76** e **8:0:27:2b:af:9e**, macros {#SNMPINDEX} definidas como os índices dos OIDs descobertos **1**, **2** e **3**:

```
[
    {
        "{#SNMPINDEX}": "1",
        "{#IFDESCR}": "WAN",
        "{#IFPHYSADDRESS}": "8:0:27:90:7a:75"
```

```
    },
    {
        "{#SNMPINDEX}": "2",
        "{#IFDESCR}": "LAN1",
        "{#IFPHYSADDRESS}": "8:0:27:90:7a:76"
    },
    {
        "{#SNMPINDEX}": "3",
        "{#IFDESCR}": "LAN2",
        "{#IFPHYSADDRESS}": "8:0:27:2b:af:9e"
    }
]
```

Se uma entidade não tiver o OID especificado, a macro correspondente será omitida para essa entidade. Por exemplo, se tivermos os seguintes dados:

```
ifDescr.1 "Interface #1"
ifDescr.2 "Interface #2"
ifDescr.4 "Interface #4"

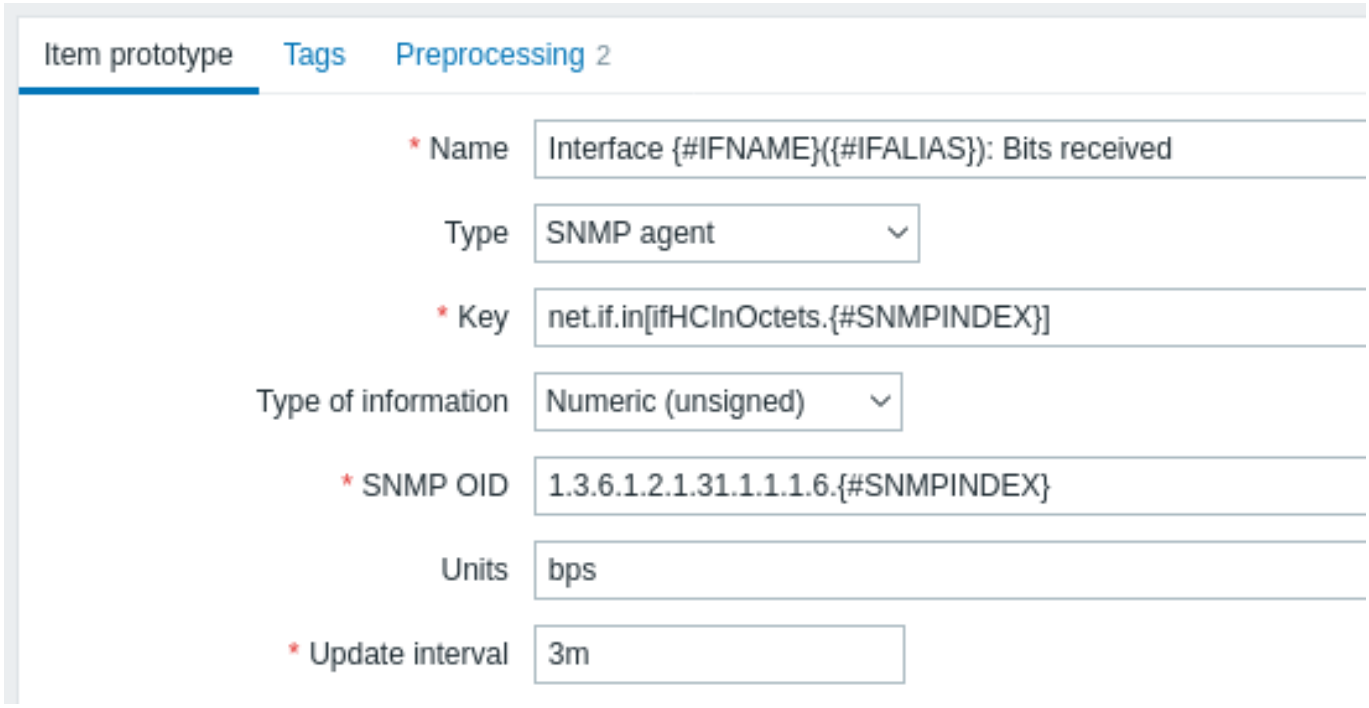
ifAlias.1 "eth0"
ifAlias.2 "eth1"
ifAlias.3 "eth2"
ifAlias.5 "eth4"
```

Então, neste caso, a descoberta SNMP `discovery[{#IFDESCR}, ifDescr, {#IFALIAS}, ifAlias]` retornará a seguinte estrutura:

```
[
    {
        "{#SNMPINDEX}": 1,
        "{#IFDESCR}": "Interface #1",
        "{#IFALIAS}": "eth0"
    },
    {
        "{#SNMPINDEX}": 2,
        "{#IFDESCR}": "Interface #2",
        "{#IFALIAS}": "eth1"
    },
    {
        "{#SNMPINDEX}": 3,
        "{#IFALIAS}": "eth2"
    },
    {
        "{#SNMPINDEX}": 4,
        "{#IFDESCR}": "Interface #4"
    },
    {
        "{#SNMPINDEX}": 5,
        "{#IFALIAS}": "eth4"
    }
]
```

Protótipos de item

A captura de tela a seguir ilustra como podemos usar essas macros em protótipos de item:



Você pode criar quantos protótipos de item forem necessários:

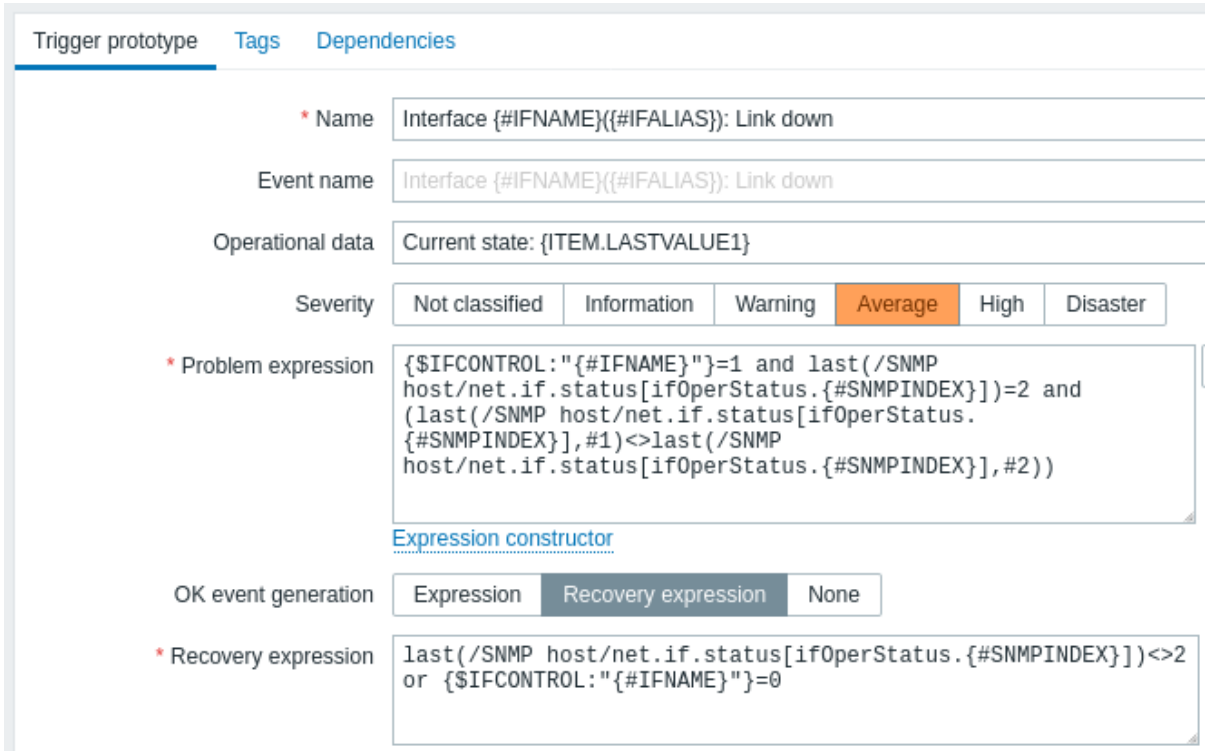
Item prototypes

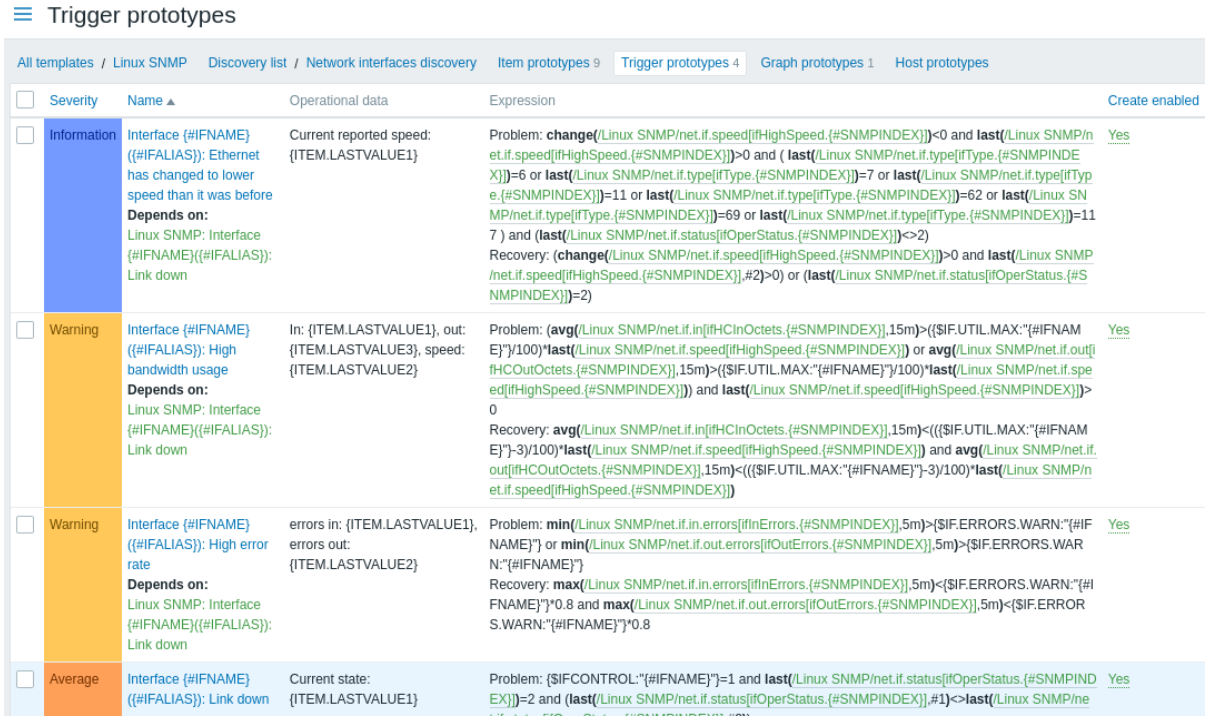
All templates / Linux SNMP Discovery list / Network interfaces discovery Item prototypes 9 Trigger prototypes 4 Graph prototypes 1 Host prototypes

Name ▲	Key	Interval	History	Trends	Type	Create enabled
Interface {#IFNAME}({#IFALIAS}): Bits received	net.if.in[ifHCInOctets.{#SNMPINDEX}]	3m	7d	365d	SNMP agent	Yes
Interface {#IFNAME}({#IFALIAS}): Bits sent	net.if.out[ifHCOutOctets.{#SNMPINDEX}]	3m	7d	365d	SNMP agent	Yes
Interface {#IFNAME}({#IFALIAS}): Inbound packets discarded	net.if.in.discards[ifInDiscards.{#SNMPINDEX}]	3m	7d	365d	SNMP agent	Yes
Interface {#IFNAME}({#IFALIAS}): Inbound packets with errors	net.if.in.errors[ifInErrors.{#SNMPINDEX}]	3m	7d	365d	SNMP agent	Yes
Interface {#IFNAME}({#IFALIAS}): Interface type	net.if.type[ifType.{#SNMPINDEX}]	1h	7d	0d	SNMP agent	Yes
Interface {#IFNAME}({#IFALIAS}): Operational status	net.if.status[ifOperStatus.{#SNMPINDEX}]	1m	7d	0	SNMP agent	Yes
Interface {#IFNAME}({#IFALIAS}): Outbound packets discarded	net.if.out.discards[ifOutDiscards.{#SNMPINDEX}]	3m	7d	365d	SNMP agent	Yes
Interface {#IFNAME}({#IFALIAS}): Outbound packets with errors	net.if.out.errors[ifOutErrors.{#SNMPINDEX}]	3m	7d	365d	SNMP agent	Yes
Interface {#IFNAME}({#IFALIAS}): Speed	net.if.speed[ifHighSpeed.{#SNMPINDEX}]	5m	7d	0d	SNMP agent	Yes

Protótipos de trigger

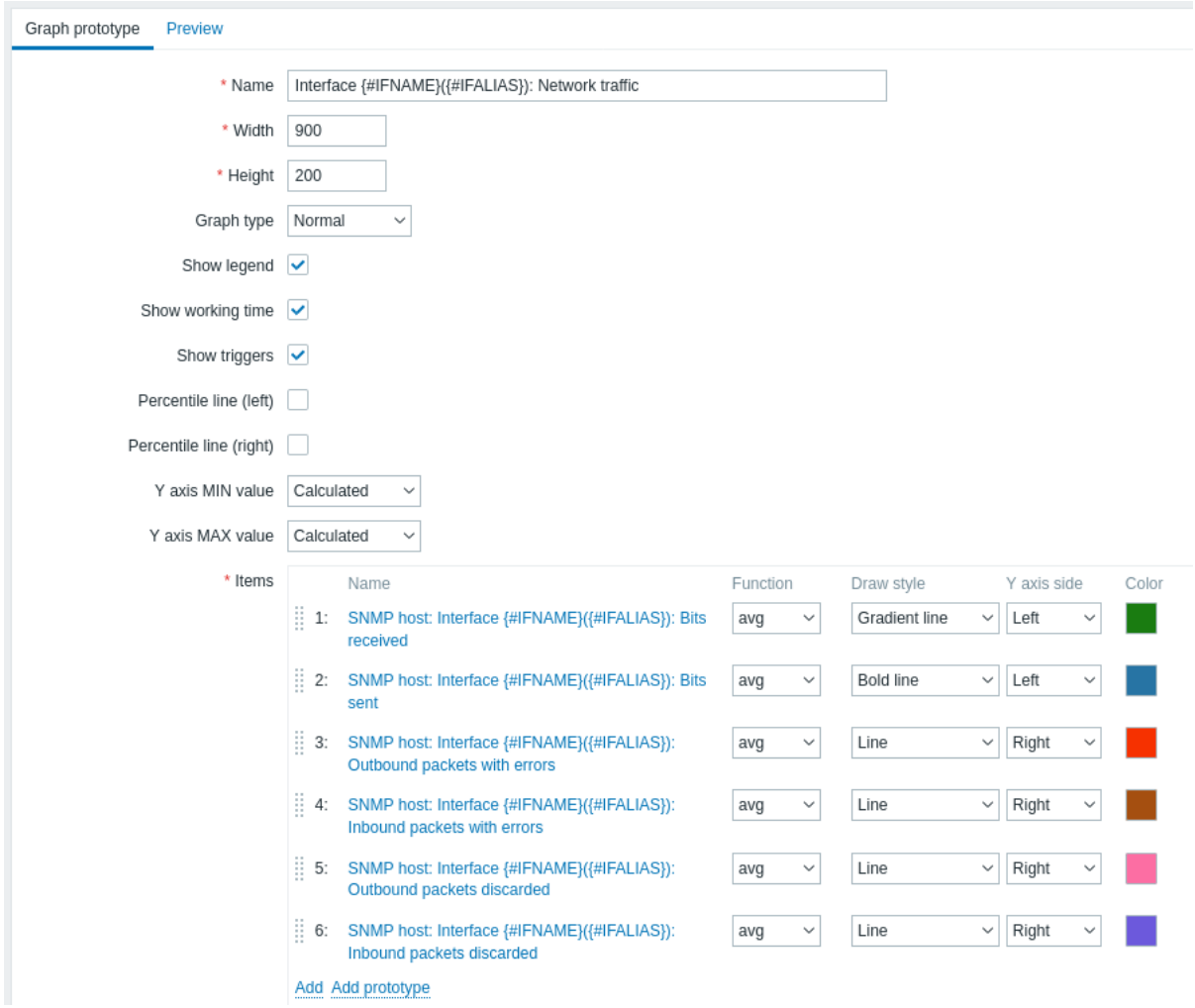
A captura de tela a seguir ilustra como podemos usar essas macros em protótipos de trigger:

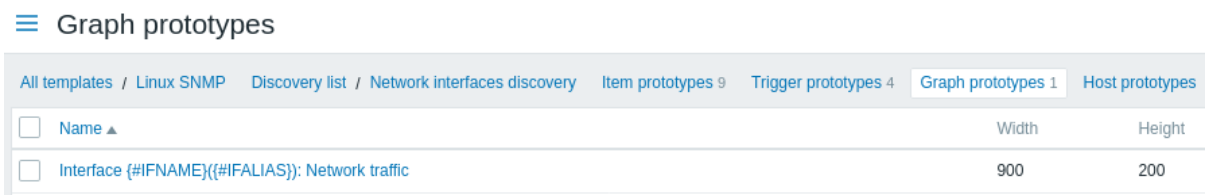




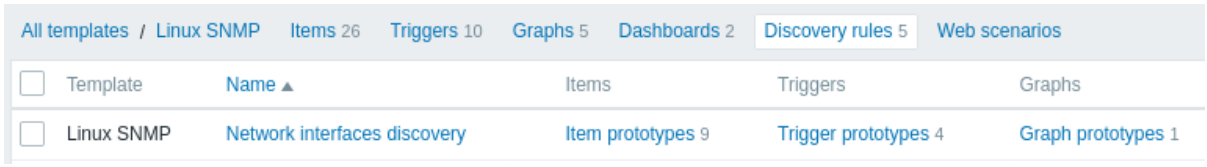
Protótipos de gráficos

A captura de tela a seguir ilustra como podemos usar essas macros em protótipos de gráficos:





Um resumo da nossa regra de descoberta:



Entidades descobertas

Quando o server é executado, ele criará itens, triggers e gráficos reais com base nos valores que a regra de descoberta SNMP retorna. Na configuração do host, eles são prefixados com um link laranja para a regra de descoberta da qual eles vêm.

6 Descoberta de objetos JMX

Visão geral

É possível descobrir todos JMX MBeans ou atributos MBean ou para especificar um padrão para a descoberta desses objetos.

É obrigatório entender a diferença entre um MBean e um MBean atributos para configuração de regras de descoberta. Um MBean é um objeto que pode representar um dispositivo, um aplicativo ou qualquer recurso que precise ser gerenciou.

Por exemplo, existe um MBean que representa um servidor web. Seu atributos são contagem de conexão, contagem de threads, tempo limite de solicitação, http cache de arquivos, uso de memória, etc. linguagem abrangente podemos definir uma máquina de café como um MBean que possui os seguintes atributos a serem monitorados: quantidade de água por xícara, consumo médio

de água por um determinado período de tempo, número de grãos de café necessários por xícara, grãos de café e tempo de recarga de água, etc.

Chave do item

Na configuração da regra de descoberta, selecione **JMX agent** no campo *Tipo*.

Duas chaves de item são suportadas para descoberta de objetos JMX - jmx.discovery[] e jmx.get[]:

Chave do item	Valor de retorno	Parâmetros	Comentário
jmx.discovery[<modo de descoberta>,<nome do objeto>,<descrição curta única>]	Este item retorna um array JSON com macros LLD descrevendo objetos MBean ou seus atributos.	**modo de descoberta** - um dos seguintes: *attributes* (recuperar atributos JMX MBean, padrão) ou *beans* (recuperar JMX MBeans) **nome do objeto** - padrão de nome do objeto (veja a documentação) identificando os nomes dos MBeans a serem recuperados (vazio por padrão, recuperando todos os beans registrados) **descrição curta única** - uma descrição única que permite múltiplos itens JMX com o mesmo modo de descoberta e nome do objeto no host (opcional)	Exemplos: → jmx.discovery - recupera todos os atributos JMX MBean → jmx.discovery[beans] - recupera todos os JMX MBeans → jmx.discovery[attributes,"*:type=GarbageCollector,name - recupera todos os atributos do coletor de lixo → jmx.discovery[beans,"*:type=GarbageCollector,name=* - recupera todos os coletores de lixo Existem algumas limitações sobre quais propriedades MBean este item pode retornar com base nos caracteres limitados que são suportados na geração do nome da macro (os caracteres suportados podem ser descritos pela seguinte expressão regular: `A-Z0-9_\.`). Portanto, por exemplo, para descobrir propriedades MBean com uma palavra hifenizada ou caracteres não ASCII, você precisa usar `jmx.get[]`.
jmx.get[<modo de descoberta>,<nome do objeto>,<descrição curta única>]			

Chave do item			
	Este item retorna um array JSON com objetos MBean ou seus atributos. Comparado ao `jmx.discovery[]` ele não define macros LLD.	**modo de descoberta** - um dos seguintes: *attributes* (recuperar atributos JMX MBean, padrão) ou *beans* (recuperar JMX MBeans) **nome do objeto** - padrão de nome do objeto (veja a documentação) identificando os nomes dos MBeans a serem recuperados (vazio por padrão, recuperando todos os beans registrados) **descrição curta única** - uma descrição única que permite múltiplos itens JMX com o mesmo modo de descoberta e nome do objeto no host (opcional)	Ao usar este item, é necessário definir macros de descoberta de baixo nível personalizadas, apontando para valores extraídos do JSON retornado usando JSONPath.

Attention:
Se nenhum parâmetro for passado, todos os atributos MBean do JMX serão solicitados. Não especificar parâmetros para a descoberta JMX ou tentar receber todos os atributos para um intervalo amplo como `*:type=*,name=*` pode levar a potenciais problemas de desempenho.

Usando jmx.discovery

Este item retorna um objeto JSON com macros de descoberta de baixo nível descrevendo objetos ou atributos MBean. Por exemplo, na descoberta de atributos MBean (reformatado para maior clareza):

```
[
    {
        "{#JMXVALUE}":"0",
        "{#JMXTYPE}":"java.lang.Long",
        "{#JMXOBJ}":"java.lang:type=GarbageCollector,name=PS Scavenge",
        "{#JMXDESC}":"java.lang:type=GarbageCollector,name=PS Scavenge,CollectionCount",
        "{#JMXATTR}":"CollectionCount"
    },
    {
        "{#JMXVALUE}":"0",
        "{#JMXTYPE}":"java.lang.Long",
        "{#JMXOBJ}":"java.lang:type=GarbageCollector,name=PS Scavenge",
        "{#JMXDESC}":"java.lang:type=GarbageCollector,name=PS Scavenge,CollectionTime",
        "{#JMXATTR}":"CollectionTime"
    },
    {
        "{#JMXVALUE}":"true",
        "{#JMXTYPE}":"java.lang.Boolean",
        "{#JMXOBJ}":"java.lang:type=GarbageCollector,name=PS Scavenge",
        "{#JMXDESC}":"java.lang:type=GarbageCollector,name=PS Scavenge,Valid",
        "{#JMXATTR}":"Valid"
    },
    {
        "{#JMXVALUE}":"PS Scavenge",
        "{#JMXTYPE}":"java.lang.String",
        "{#JMXOBJ}":"java.lang:type=GarbageCollector,name=PS Scavenge",
        "{#JMXDESC}":"java.lang:type=GarbageCollector,name=PS Scavenge,Name",
        "{#JMXATTR}":"Name"
    },
    {
        "{#JMXVALUE}":"java.lang:type=GarbageCollector,name=PS Scavenge",
        "{#JMXTYPE}":"javax.management.ObjectName",
```

```
        "{#JMXOBJ}":"java.lang:type=GarbageCollector,name=PS Scavenge",
        "{#JMXDESC}":"java.lang:type=GarbageCollector,name=PS Scavenge,ObjectName",
        "{#JMXATTR}":"ObjectName"
    }
]
```

Na descoberta de MBeans (reformatado para maior clareza):

```
[
    {
        "{#JMXDOMAIN}":"java.lang",
        "{#JMXTYPE}":"GarbageCollector",
        "{#JMXOBJ}":"java.lang:type=GarbageCollector,name=PS Scavenge",
        "{#JMXNAME}":"PS Scavenge"
    }
]
```

Macros compatíveis

As seguintes macros têm suporte para uso na regra de descoberta filtro e protótipos de itens, gatilhos e gráficos:

Macro	Descrição
Descoberta de atributos MBean	
{#JMXVALUE}	Valor do atributo.
{#JMXTYPE}	Tipo de atributo.
{#JMXOBJ}	Nome do objeto.
{#JMXDESC}	Nome do objeto incluindo o nome do atributo.
{#JMXATTR}	Nome do atributo.
Descoberta de MBeans	
{#JMXDOMAIN}	domínio MBean. (*Nome reservado Zabbix*)
{#JMXOBJ}	Nome do objeto. (*Nome reservado Zabbix*)
{#JMX<propriedade da chave>}	Propriedades MBean (como {#JMXTYPE}, {#JMXNAME}) (consulte Limitações abaixo).

Limitações

Existem algumas limitações associadas ao algoritmo de criação de nomes de macros LLD a partir de nomes de propriedades MBean:

- os nomes dos atributos são convertidos para maiúsculas
- os nomes dos atributos são ignorados (nenhuma macro LLD é gerada) se consistirem em caracteres não suportados para nomes de macros LLD. Os caracteres suportados podem ser descritos pela seguinte expressão regular: `A-Z0-9_\.`
- se um atributo for chamado ”obj” ou ”domain”, eles serão ignorados devido à sobreposição com os valores das propriedades reservadas do Zabbix {#JMXOBJ} e {#JMXDOMAIN}

Considere este exemplo de jmx.discovery (com modo ”beans”). O MBean tem as seguintes propriedades definidas (algumas das quais serão ignoradas; veja abaixo):

```
name=test
  =Type
attributes []=1,2,3
Name=NameOfTheTest
domAin=some
```

Como resultado da descoberta JMX, as seguintes macros LLD serão geradas:

- {#JMXDOMAIN} - interna do Zabbix, descrevendo o domínio do MBean
- {#JMXOBJ} - interna do Zabbix, descrevendo o objeto MBean
- {#JMXNAME} - criada a partir da propriedade ”name”

As propriedades ignoradas são:

- тип : seu nome contém caracteres não suportados (não-ASCII)
- attributes[] : seu nome contém caracteres não suportados (colchetes não são suportados)
- Name : já está definida (name=test)
- domAin : é um nome reservado do Zabbix

Exemplos

Vamos revisar dois exemplos mais práticos de criação de uma regra de LLD com o uso de MBean. Para entender melhor a diferença entre uma regra de LLD que coleta MBeans e uma regra de LLD que coleta atributos de MBean, por favor, veja a tabela a seguir:

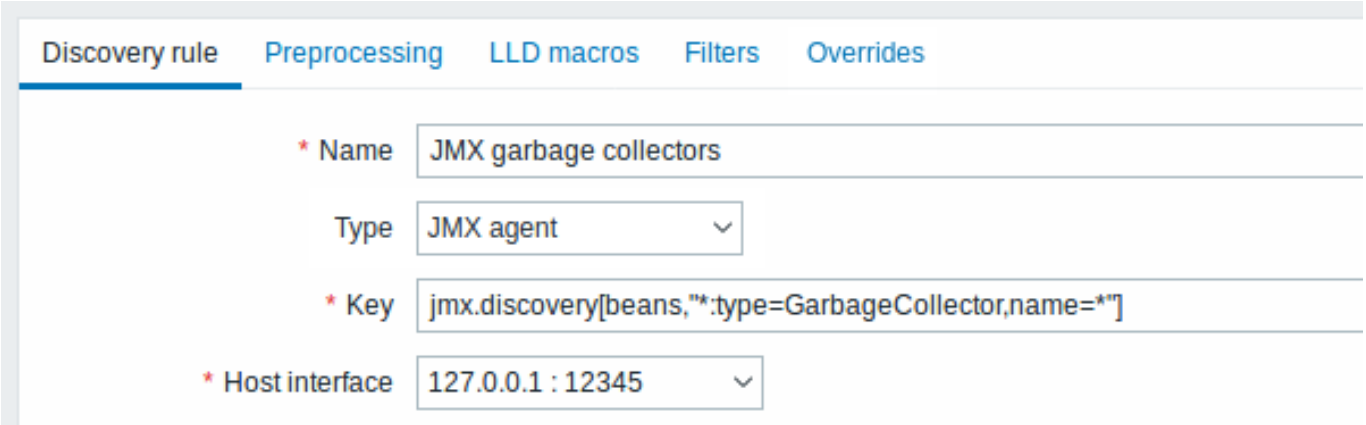
MBean1	**MBean2**	**MBean3**
MBean1Attribute1	MBean2Attribute1	MBean3Attribute1
MBean1Attribute2	MBean2Attribute2	MBean3Attribute2
MBean1Attribute3	MBean2Attribute3	MBean3Attribute3

Exemplo 1: Descobrindo MBeans

Esta regra retornará 3 objetos: a linha superior da coluna: MBean1, MBean2, MBean3.

Para mais informações sobre objetos, consulte a tabela de macros suportadas, seção *Descoberta de MBeans*.

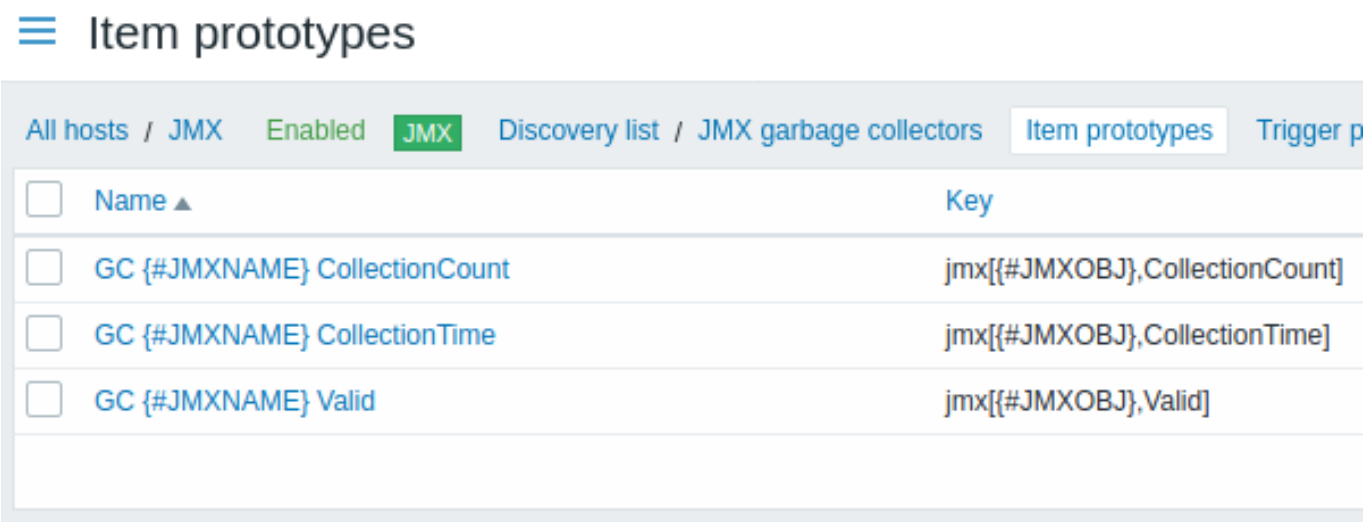
A configuração da regra de descoberta coletando MBeans (sem os atributos) fica assim:



A chave usada aqui:

```
jmx.discovery[beans,"*:type=GarbageCollector,name=*"]
```

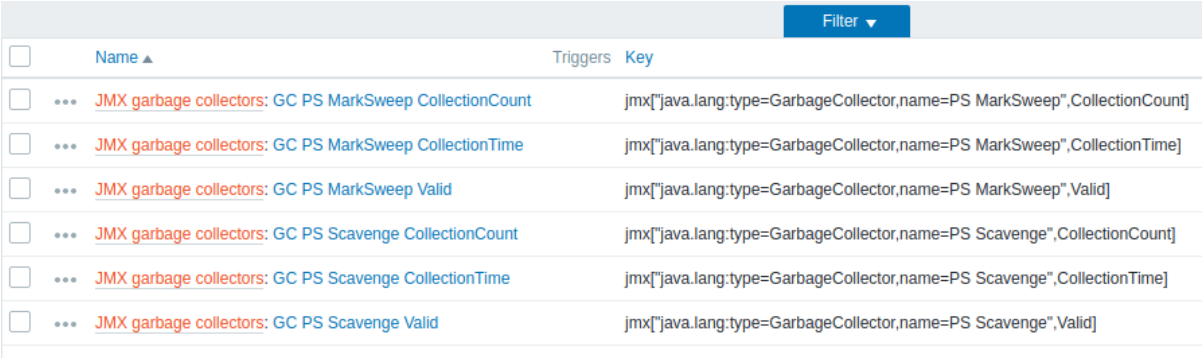
Todos os coletores de lixo sem atributos serão descobertos. Como os coletores de lixo possuem o mesmo conjunto de atributos, podemos usar os atributos desejados nos protótipos de item da seguinte forma:



As chaves usadas aqui:

```
jmx[{#JMXOBJ},CollectionCount]
jmx[{#JMXOBJ},CollectionTime]
jmx[{#JMXOBJ},Valid]
```

A regra de descoberta LLD resultará em algo próximo a isto (itens são descobertos para dois coletores de lixo):



Exemplo 2: Descobrindo atributos do MBean

Esta regra retornará 9 objetos com os seguintes campos: MBean1Attribute1, MBean2Attribute1, MBean3Attribute1,MBean1Attribute2,MBean2 MBean3Attribute2, MBean1Attribute3, MBean2Attribute3, MBean3Attribute3.

Para mais informações sobre objetos, consulte a tabela de macros suportadas, seção *Descoberta de atributos do MBean*.

A configuração da regra de descoberta coletando atributos do MBean é semelhante à seguinte:

lld_rule_mbean_attr.png

A chave usada aqui:

`jmx.discovery[attributes,"*:type=GarbageCollector,name=*"]`

Todos os coletores de lixo com um único atributo de item serão descobertos.

lld_rule_mbean_attr_prototypes.png

Neste caso específico, um item será criado a partir do protótipo para cada atributo do MBean. A principal desvantagem desta configuração é que a criação de trigger a partir de protótipos de trigger é impossível, pois há apenas um protótipo de item para todos os atributos. Portanto, esta configuração pode ser usada para coleta de dados, mas não é recomendada para monitoramento automático.

Usando jmx.get

`jmx.get[]` é semelhante ao item `jmx.discovery[]`, mas não transforma as propriedades do objeto Java em nomes de macros de descoberta de baixo nível e, portanto, pode retornar valores sem as limitações associadas à geração de nomes de macros LLD, como hífens ou caracteres não ASCII.

Ao usar `jmx.get[]` para descoberta, macros de descoberta de baixo nível podem ser definidas separadamente na guia personalizada de macro LLD da configuração da regra de descoberta, usando JSONPath para apontar para os valores necessários.

Descobrindo MBeans

Item de descoberta: `jmx.get[beans,"com.example:type=*,*"]`

Resposta:

```
[
    {
        "object": "com.example:type=Hello,data-src=data-base,  =    ",
        "domínio": "com.exemplo",
        "propriedades": {
            "data-src": "base de dados",
            "  ": "    ",
            "tipo": "Olá"
        }
    },
    {
        "object": "com.example:type=Atomic",
        "domínio": "com.exemplo",
        "propriedades": {
            "tipo": "Atômica"
        }
    }
]
```

Descobrindo os atributos do MBean

Item de descoberta: `jmx.get[attributes,"com.example:type=*,*"]`

Resposta:

```
[
    {
        "object": "com.example:type="*",
        "domínio": "com.exemplo",
        "propriedades": {
            "tipo": "Simples"
        }
    },
    {
        "object": "com.zabbix:type=yes,domain=zabbix.com,data-source=/dev/rand,  =     ,obj=true",
        "domínio": "com.zabbix",
        "propriedades": {
            "tipo": "Olá",
            "domínio": "com.exemplo",
            "fonte de dados": "/dev/rand",
            "  ": "     ",
            "obj": verdadeiro
        }
    }
]
```

7 Descoberta de sensores IPMI

Visão geral

É possível descobrir automaticamente sensores IPMI.

Para isso, você pode usar uma combinação de:

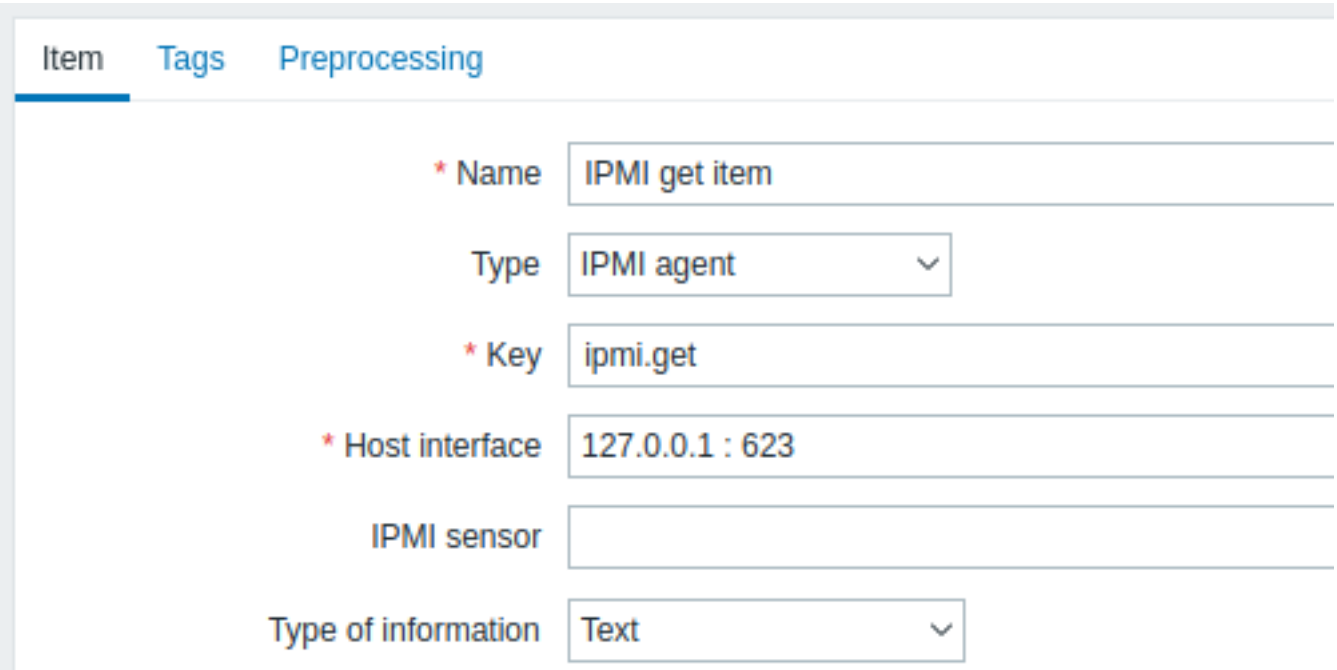
- o item IPMI `ipmi.get` como item mestre
- regra de descoberta de baixo nível dependente e protótipos de item

1 Configuração

Item mestre

Crie um item IPMI usando a seguinte chave:

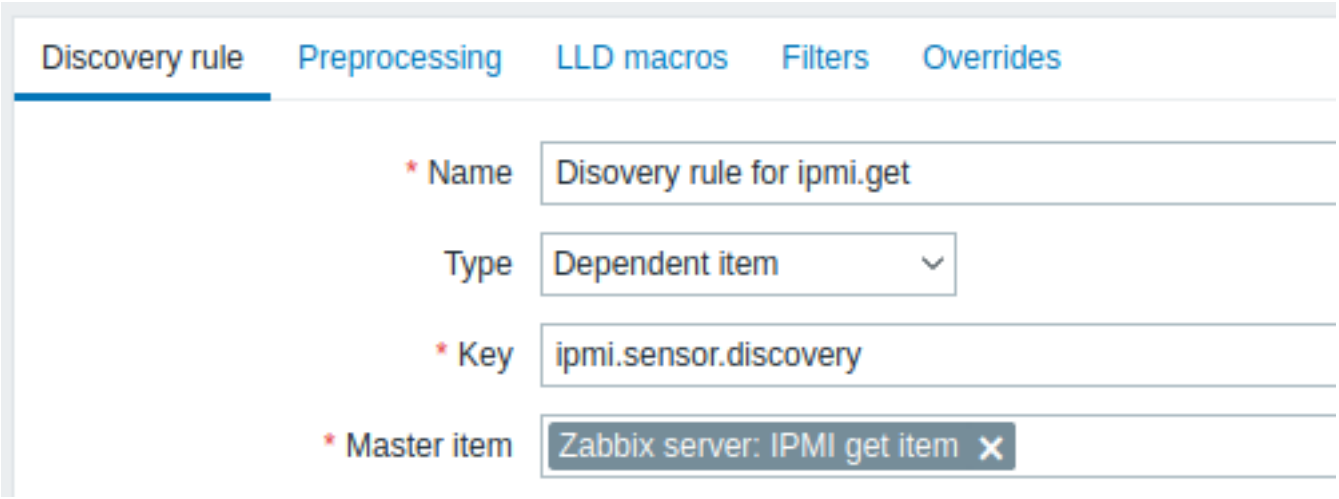
`ipmi.get`



Defina o tipo de informação como ”Texto” para dados JSON possivelmente grandes.

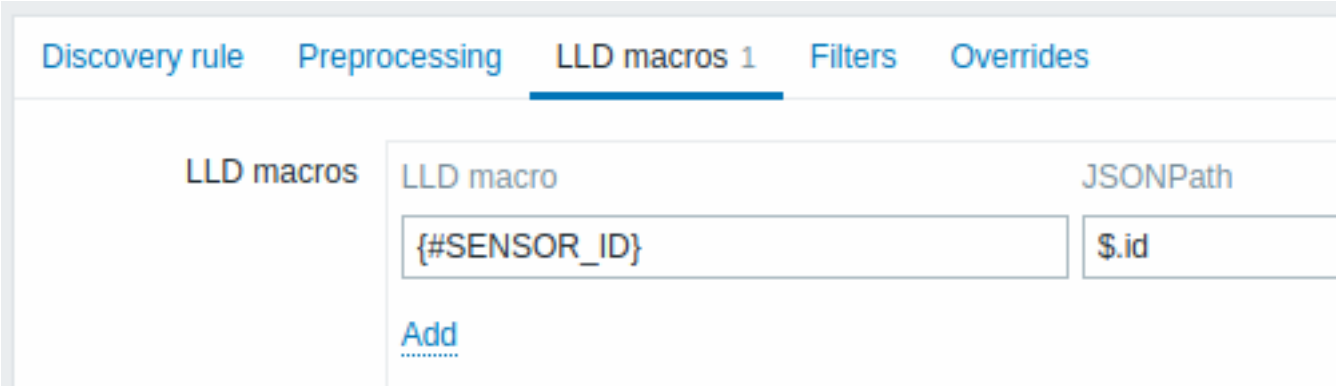
Regra LLD dependente

Crie uma regra de descoberta de baixo nível como tipo ”Item dependente”:



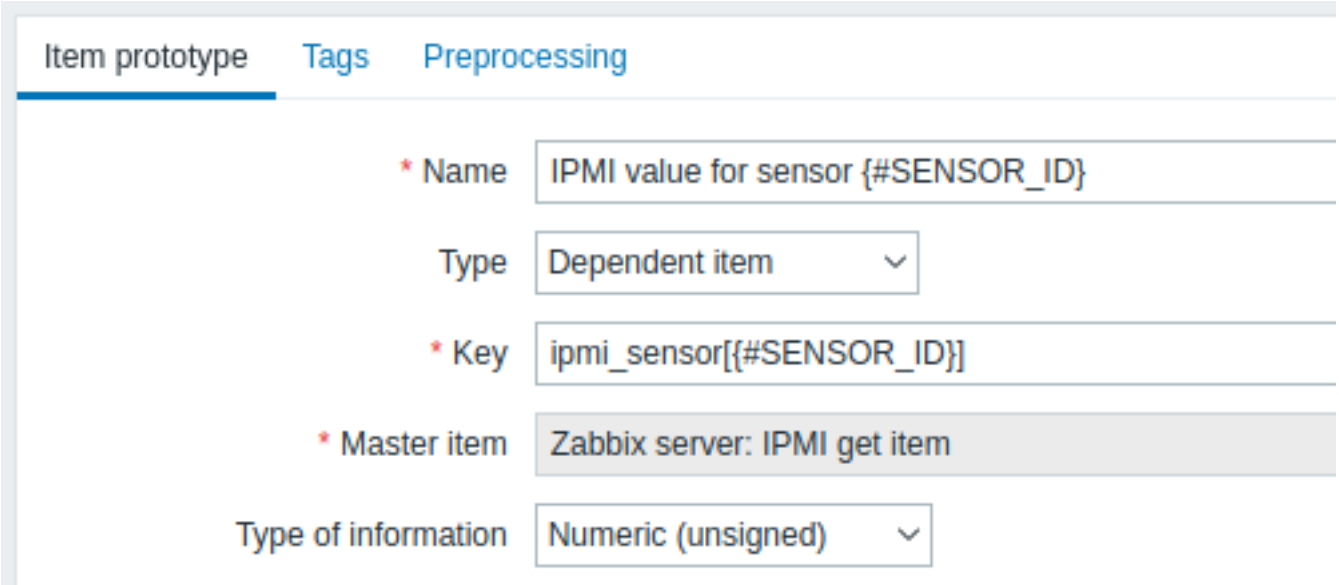
Como item mestre selecione o item `ipmi.get` que criamos.

Na aba ”Macros LLD” defina uma macro personalizada com o correspondente Caminho JSON:



Protótipo de item dependente

Crie um protótipo de item com o tipo ”Item dependente” nesta regra LLD. Como item mestre para este protótipo selecione o item `ipmi.get` que criamos.



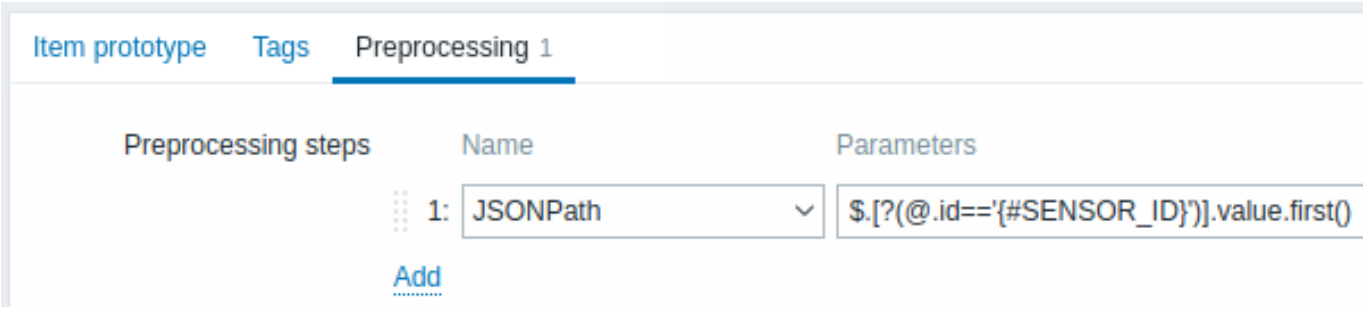
Observe o uso da macro {#SENSOR_ID} no nome do protótipo do item e chave:

- *Nome*: valor IPMI para o sensor {#SENSOR_ID}
- *Chave*: ipmi_sensor[{#SENSOR_ID}]

Como tipo de informação, *Numérico (sem sinal)*.

Na guia ”Pré-processamento” do protótipo do item, selecione JSONPath e use o seguinte expressão JSONPath como parâmetro:

`$.[?(@.id=='{#SENSOR_ID}')].value.first()`



Quando a descoberta for iniciada, será criado um item para cada sensor IPMI. Este item retornará o valor inteiro do sensor fornecido.

8 Descoberta de serviços systemd

Visão geral

É possível descobrir unidades systemd (serviços, por padrão) com Zabbix.

Chave do item

O item a ser usado na descoberta regra é a

`systemd.unit.discovery`

::: não importante este item é suportado apenas no agente Zabbix 2. :::

Este item retorna um JSON com informações sobre unidades systemd, por exemplo:

```
[{
    "{#UNIT.NAME}": "mysqld.service",
    "{#UNIT.DESCRIPTION}": "Servidor MySQL",
    "{#UNIT.LOADSTATE}": "carregado",
    "{#UNIT.ACTIVESTATE}": "ativo",
    "{#UNIT.SUBSTATE}": "executando",
    "{#UNIT.FOLLOWED}": "",
    "{#UNIT.PATH}": "/org/freedesktop/systemd1/unit/mysqld_2eservice",
    "{#UNIT.JOBID}": 0,
    "{#UNIT.JOBTYPE}": ""
    "{#UNIT.JOBPATH}": "/",
    "{#UNIT.UNITFILESTATE}": "habilitado"
}, {
    "{#UNIT.NAME}": "systemd-journald.socket",
    "{#UNIT.DESCRIPTION}": "Soquete de diário",
    "{#UNIT.LOADSTATE}": "carregado",
    "{#UNIT.ACTIVESTATE}": "ativo",
    "{#UNIT.SUBSTATE}": "executando",
    "{#UNIT.FOLLOWED}": "",
    "{#UNIT.PATH}": "/org/freedesktop/systemd1/unit/systemd_2djournald_2esocket",
    "{#UNIT.JOBID}": 0,
    "{#UNIT.JOBTYPE}": "",
    "{#UNIT.JOBPATH}": "/"
    "{#UNIT.UNITFILESTATE}": "habilitado"
}]
```

Descoberta de unidades systemd desabilitadas

Também é possível descobrir unidades systemd **desabilitadas**. Neste caso, três macros são retornadas no JSON resultante:

- {#UNIT.PATH}
- {#UNIT.ACTIVESTATE}
- {#UNIT.UNITFILESTATE}.

Attention:
Para que items e triggers sejam criados a partir de protótipos para unidades systemd desabilitadas, certifique-se de ajustar (ou remover) os filtros LLD proibitivos para {#UNIT.ACTIVESTATE} e {#UNIT.UNITFILESTATE}.

Macros compatíveis

As seguintes macros têm suporte para uso na regra de descoberta filtro e protótipos de itens, gatilhos e gráficos:

Macro	Descrição
{#UNIT.NAME}	Nome da unidade primária.
{#UNIT.DESCRIPTION}	Descrição legível por humanos.
{#UNIT.LOADSTATE}	Estado de carregamento (ou seja, se o arquivo de unidade foi carregado com sucesso)
{#UNIT.ACTIVESTATE}	Estado ativo (ou seja, se a unidade está iniciada ou não)
{#UNIT.SUBSTATE}	Subestado (uma versão mais refinada do estado ativo que é específico para o tipo de unidade, que o estado ativo não é)
{#UNIT.FOLLOWED}	Unidade que está sendo seguida em seu estado por esta unidade, se houver; caso contrário, uma string vazia.
{#UNIT.PATH}	Caminho do objeto da unidade.
{#UNIT.JOBID}	ID de trabalho numérico se houver um trabalho na fila para a unidade de trabalho; 0 caso contrário.
{#UNIT.JOBTYPE}	Tipo de trabalho.
{#UNIT.JOBPATH}	Caminho do objeto de trabalho.
{#UNIT.UNITFILESTATE}	O estado de instalação do arquivo de unidade.

Protótipos de item

Protótipos de item que podem ser criados com base na descoberta de serviços systemd incluem, por exemplo:

- Nome do item: `{#UNIT.DESCRIPTION} active state info`; chave do item: `systemd.unit.info["{#UNIT.NAME}"]`
- Nome do item: `{#UNIT.DESCRIPTION} load state info`; chave do item: `systemd.unit.info["{#UNIT.NAME}",LoadState`

9 Descoberta de serviços do Windows

Visão geral

De maneira semelhante ao arquivo sistemas são descobertos, é possível descobrir também os serviços do Windows.

Chave do item

O item a ser usado na regra de descoberta é

`service.discovery`

Macros suportadas

As seguintes macros são suportadas para uso na regra de descoberta (filtro) e nos protótipos de items, triggers e gráficos:

Macro	Descrição
{#SERVICE.NAME}	Nome do serviço.
{#SERVICE.DISPLAYNAME}	Nome do serviço exibido.
{#SERVICE.DESCRIPTION}	Descrição do serviço.
{#SERVICE.STATE}	Valor numérico do estado do serviço. Veja o item service.info para detalhes.
{#SERVICE.STATENAME}	Nome do estado do serviço. Veja o item service.info para detalhes.
{#SERVICE.PATH}	Caminho do serviço.
{#SERVICE.USER}	Usuário do serviço.
{#SERVICE.STARTUP}	Valor numérico do tipo de inicialização do serviço. Veja o item service.info para detalhes.
{#SERVICE.STARTUPNAME}	Nome do tipo de inicialização do serviço. Veja o item service.info para detalhes.

Macro	Descrição
{#SERVICE.STARTUPTRIGGER}	Valor numérico para indicar se o tipo de inicialização do serviço possui: 0 - sem triggers de inicialização 1 - possui triggers de inicialização É útil para descobrir tipos de inicialização de serviço como *Automático (início por trigger)*, *Automático atrasado (início por trigger)* e *Manual (início por trigger)*.

Com base na descoberta de serviços do Windows, você pode criar um protótipo de item como

`service.info[{#SERVICE.NAME},<param>]`

onde `param` aceita os seguintes valores: *state*, *displayname*, *path*, *user*, *startup* ou *description*.

Por exemplo, para obter o nome exibido de um serviço, você pode usar um item "service.info[{#SERVICE.NAME},displayname]". Se o valor de `param` não for especificado ("service.info[{#SERVICE.NAME}]"), o parâmetro padrão *state* será usado.

10 Descoberta de instâncias de contadores de desempenho do Windows

Visão geral

É possível descobrir instâncias de objetos de contadores de desempenho do Windows. Isso é útil para contadores de desempenho com várias instâncias.

Chave do item

Para configurar a regra de descoberta, use o seguinte item:

- `perf_instance.discovery[object]`

Observe que o nome do objeto pode estar localizado. Por exemplo:

```
perf_instance.discovery[Processor] # O nome do objeto está em inglês.
perf_instance.discovery[Processador] # O nome do objeto está em português.
```

Como alternativa, para garantir que o nome do objeto seja fornecido em inglês, independentemente da localização do SO, use o seguinte item:

- `perf_instance_en.discovery[object]`

Por exemplo:

```
perf_instance_en.discovery[Processor]
perf_instance_en.discovery[Memory]
```

Macros suportadas

O processo de descoberta retornará todas as instâncias do objeto especificado na macro {#INSTANCE}:

```
[
    {"{#INSTANCE}":"0"},
    {"{#INSTANCE}":"1"},
    {"{#INSTANCE}":"_Total"}
]
```

Esta macro pode ser usada nos protótipos de items `perf_counter[]` e `perf_counter_en[]`.

Por exemplo, se a chave do item usada na regra de descoberta for `perf_instance.discovery[Processor]`, você pode criar o seguinte protótipo de item:

```
perf_counter["\Processor({#INSTANCE})\% Processor Time"]
```

Nota:

- Se o objeto especificado não for encontrado ou não suportar instâncias variáveis, o item de descoberta se tornará NOTSUPPORTED.
- Se o objeto especificado suportar instâncias variáveis, mas atualmente não tiver nenhuma instância, um array JSON vazio será retornado.
- Instâncias duplicadas serão ignoradas.

11 Descoberta usando consultas WMI

Visão geral

WMI é uma interface poderosa no Windows que pode ser usada para recuperar várias informações sobre componentes do Windows, serviços, estado e software instalado.

Ele pode ser usado para descoberta de disco físico e seus dados de desempenho coleção, descoberta de interface de rede, descoberta de convidado Hyper-V, monitorando os serviços do Windows e muitas outras coisas no sistema operacional Windows.

Este tipo de baixo nível discovery é feito usando WQL consultas cujos resultados são transformados automaticamente em um objeto JSON adequado para descoberta de baixo nível.

Chave do item

O item a ser usado na regra de descoberta é

`wmi.getall[<namespace>,<query>]`

Este item transforma o resultado da consulta em um array JSON. Por exemplo:

`select * from Win32_DiskDrive where Name like '%PHYSICALDRIVE%'`

pode retornar algo assim:

```
[
    {
        "DeviceID" : "\\.\PHYSICALDRIVE0",
        "BytesPerSector" : 512,
        "Capabilities" : [
            3,
            4
        ],
        "CapabilityDescriptions" : [
            "Random Access",
            "Supports Writing"
        ],
        "Caption" : "VBOX HARDDISK ATA Device",
        "ConfigManagerErrorCode" : 0,
        "ConfigManagerUserConfig" : "False",
        "CreationClassName" : "Win32_DiskDrive",
        "Description" : "Disk drive",
        "FirmwareRevision" : "1.0",
        "Index" : 0,
        "InterfaceType" : "IDE"
    },
    {
        "DeviceID" : "\\.\PHYSICALDRIVE1",
        "BytesPerSector" : 512,
        "Capabilities" : [
            3,
            4
        ],
        "CapabilityDescriptions" : [
            "Random Access",
            "Supports Writing"
        ],
        "Caption" : "VBOX HARDDISK ATA Device",
        "ConfigManagerErrorCode" : 0,
        "ConfigManagerUserConfig" : "False",
        "CreationClassName" : "Win32_DiskDrive",
        "Description" : "Disk drive",
        "FirmwareRevision" : "1.0",
        "Index" : 1,
        "InterfaceType" : "IDE"
    }
]
```

Macros de descoberta de baixo nível

Mesmo que nenhuma macro de descoberta de baixo nível seja criada no retorno JSON, essas macros podem ser definidas pelo usuário como uma etapa adicional, usando o LDL personalizado macro funcionalidade com JSONPath apontando para os valores descobertos no retornou JSON.

As macros podem ser usadas para criar protótipos de itens, gatilhos, etc.

12 Descoberta usando consultas SQL ODBC

Visão geral

Este tipo de descoberta de baixo nível é feito usando consultas SQL, cujos resultados são automaticamente transformados em um objeto JSON adequado para descoberta de baixo nível.

Chave do item

As consultas SQL são executadas usando um tipo de item ”Database monitor”. Portanto, a maioria das instruções na página de monitoramento ODBC se aplica para obter uma regra de descoberta ”Database monitor” funcional.

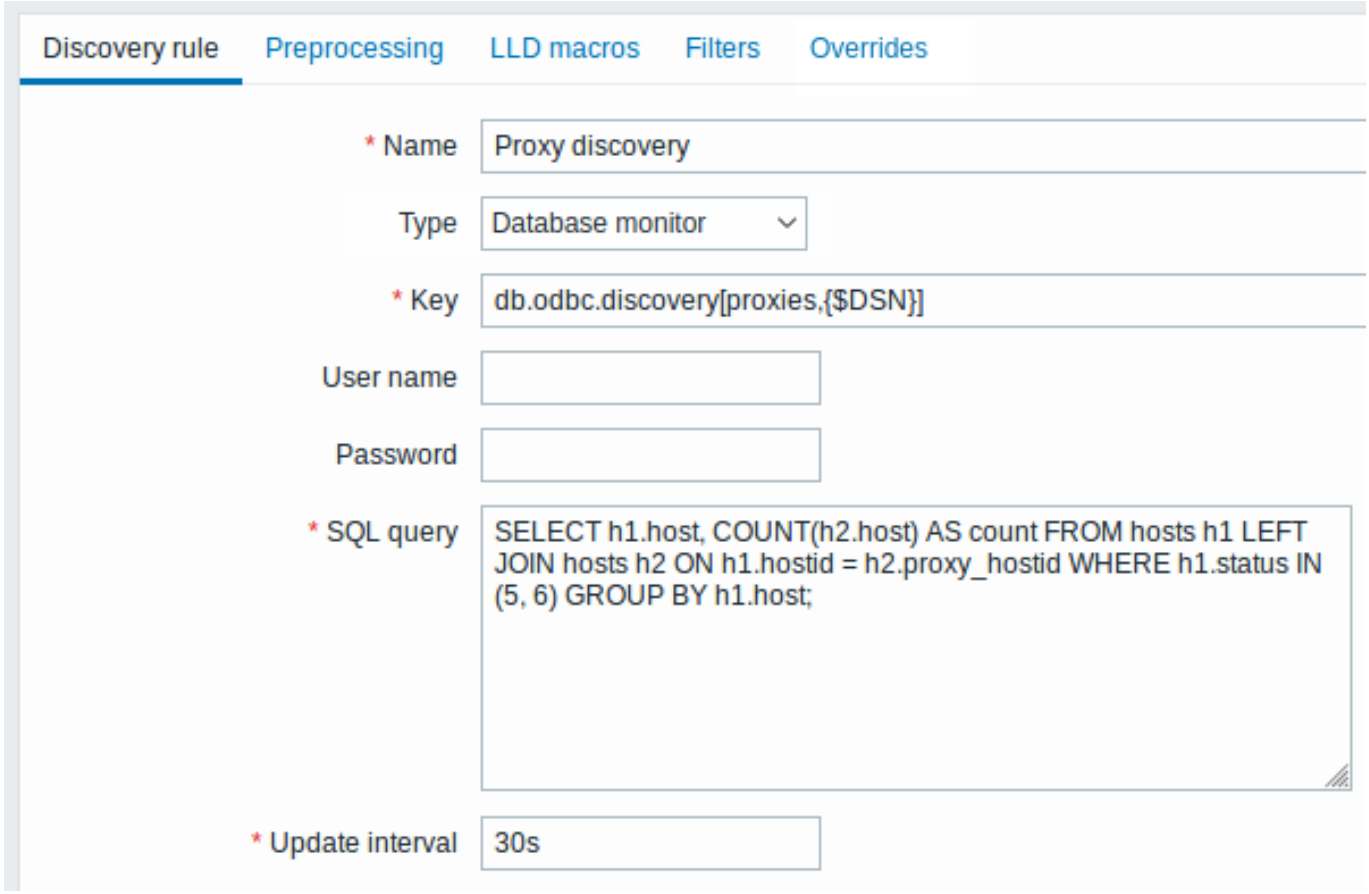
Duas chaves de item podem ser usadas em regras de descoberta ”Database monitor”:

- **db.odbc.discovery**[<descrição curta exclusiva>,<dsn>,<string de conexão>] - este item transforma o resultado da consulta SQL em um array JSON, transformando os nomes das colunas do resultado da consulta em nomes de macros de descoberta de baixo nível emparelhados com os valores dos campos descobertos. Essas macros podem ser usadas na criação de protótipos de item, trigger, etc. Veja também: Usando db.odbc.discovery.
- **db.odbc.get**[<descrição curta exclusiva>,<dsn>,<string de conexão>] - este item transforma o resultado da consulta SQL em um array JSON, mantendo os nomes originais das colunas do resultado da consulta como um nome de campo no JSON emparelhado com os valores descobertos. Comparado ao `db.odbc.discovery[]`, este item não cria macros de descoberta de baixo nível no JSON retornado, portanto, não há necessidade de verificar se os nomes das colunas podem ser nomes de macros válidos. As macros de descoberta de baixo nível podem ser definidas como uma etapa adicional, se necessário, usando a funcionalidade de macro LLD personalizada com JSONPath apontando para os valores descobertos no JSON retornado. Veja também: Usando db.odbc.get.

Usando db.odbc.discovery

Como um exemplo prático para ilustrar como a consulta SQL é transformada em JSON, vamos considerar a descoberta de baixo nível de proxies Zabbix realizando uma consulta ODBC no banco de dados Zabbix. Isso é útil para a criação automática de itens internos ”zabbix[proxy,<name>,lastaccess]” para monitorar quais proxies estão ativos.

Vamos começar com a configuração da regra de descoberta:



Todos os campos obrigatórios estão marcados com um asterisco vermelho.

Aqui, a seguinte consulta direta no banco de dados Zabbix é usada para selecionar todos os proxies Zabbix, juntamente com o número de hosts que eles estão monitorando. O número de hosts pode ser usado, por exemplo, para filtrar proxies vazios:

```
mysql> SELECT h1.host, COUNT(h2.host) AS count FROM hosts h1 LEFT JOIN hosts h2 ON h1.hostid = h2.proxyid
+---------+-------+
| host    | count |
+---------+-------+
| Japan 1 |     5 |
| Japan 2 |    12 |
| Latvia  |     3 |
+---------+-------+
3 rows in set (0.01 sec)
```

Pelo funcionamento interno do item "db.odbc.discovery[,{$DSN}]", o resultado desta consulta é automaticamente transformado no seguinte JSON:

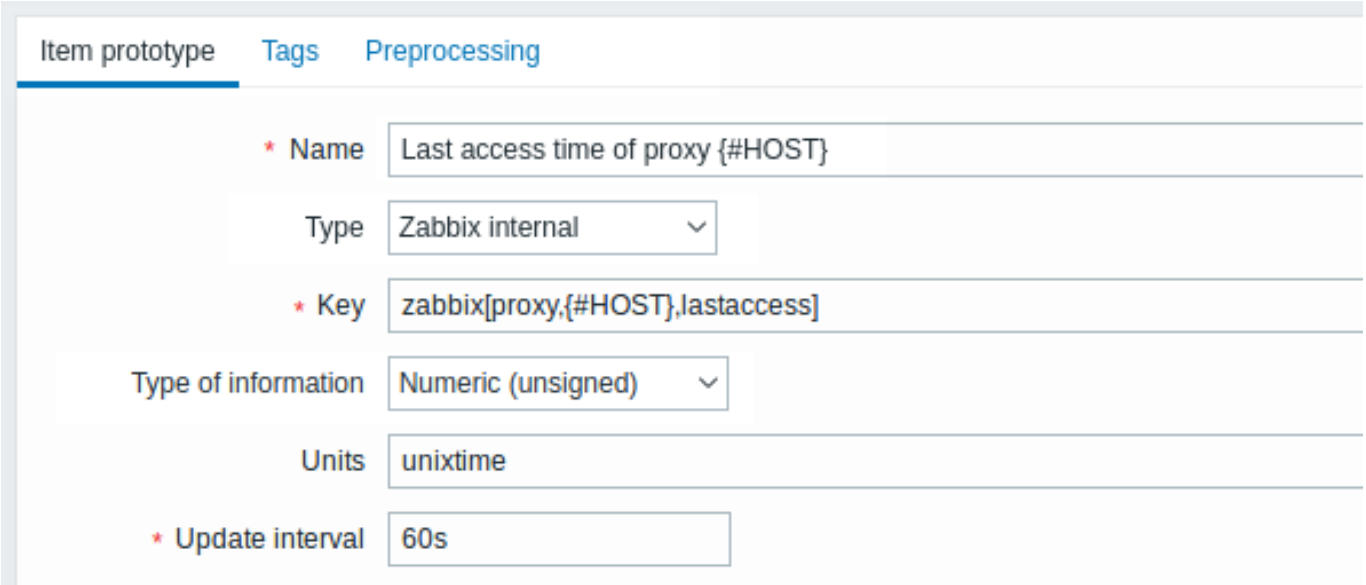
```
[
    {
        "{#HOST}": "Japan 1",
        "{#COUNT}": "5"
    },
    {
        "{#HOST}": "Japan 2",
        "{#COUNT}": "12"
    },
    {
        "{#HOST}": "Latvia",
        "{#COUNT}": "3"
    }
]
```

Pode-se ver que os nomes das colunas tornam-se nomes de macros e as linhas selecionadas tornam-se os valores dessas macros.

Note:
Se não for óbvio como um nome de coluna seria transformado em um nome de macro, sugere-se usar aliases de coluna como "COUNT(h2.host) AS count" no exemplo acima.
Caso um nome de coluna não possa ser convertido em um nome de macro válido, a regra de descoberta torna-se não suportada, com a mensagem de erro detalhando o número da coluna com problema. Se for desejada ajuda adicional, os nomes das colunas obtidas são fornecidos com DebugLevel=4 no arquivo de log do servidor Zabbix:

```
$ grep db.odbc.discovery /tmp/zabbix_server.log
 ...
 23876:20150114:153410.856 In db_odbc_discovery() query:'SELECT h1.host, COUNT(h2.host) FROM hosts h1 L
 23876:20150114:153410.860 db_odbc_discovery() column[1]:'host'
 23876:20150114:153410.860 db_odbc_discovery() column[2]:'COUNT(h2.host)'
 23876:20150114:153410.860 End of db_odbc_discovery():NOTSUPPORTED
 23876:20150114:153410.860 Item [Zabbix server:db.odbc.discovery[proxies,{$DSN}]] error: Cannot convert
```

Agora que entendemos como uma consulta SQL é transformada em um objeto JSON, podemos usar a macro {#HOST} em protótipos de item:



Uma vez que a descoberta seja realizada, um item será criado para cada proxy:

Usando db.odbc.get

Usando `db.odbc.get[,{$DSN}]` e o seguinte exemplo de SQL:

```
mysql> SELECT h1.host, COUNT(h2.host) AS count FROM hosts h1 LEFT JOIN hosts h2 ON h1.hostid = h2.proxyid
+---------+-------+
| host    | count |
+---------+-------+
| Japan 1 |     5 |
| Japan 2 |    12 |
| Latvia  |     3 |
+---------+-------+
3 rows in set (0.01 sec)
```

este JSON será retornado:

```
[
    {
        "host": "Japan 1",
        "count": "5"
    },
    {
        "host": "Japan 2",
        "count": "12"
    },
    {
        "host": "Latvia",
        "count": "3"
    }
]
```

Como você pode ver, não há macros de descoberta de baixo nível aí. No entanto, macros de descoberta de baixo nível personalizadas podem ser criadas na aba Macros de LLD de uma regra de descoberta usando JSONPath, por exemplo:

```
{#HOST} → $.host
```

Agora esta macro {#HOST} pode ser usada em protótipos de item:

item_prototype_odbc.png

13 Descoberta usando dados do Prometheus

Visão geral

Os dados fornecidos no formato de linha do Prometheus podem ser usados para descoberta.

Consulte verificações do Prometheus para detalha como a consulta de dados do Prometheus é implementada no Zabbix.

Configuração

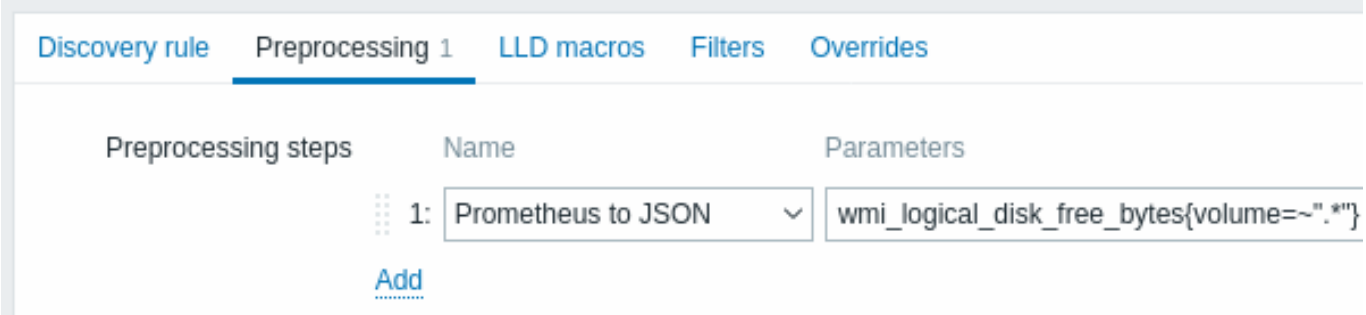
A regra de descoberta de baixo nível deve ser criada como um dependente item para o mestre HTTP item que coleta dados do Prometheus.

Prometheus para JSON

Na regra de descoberta, vá para a guia Pré-processamento e selecione o Opção de pré-processamento *Prometheus to JSON*. Os dados no formato JSON são necessário para descoberta e a opção de pré-processamento *Prometheus to JSON* retornará exatamente isso, com os seguintes atributos:

- nome da métrica
- valor métrico
- ajuda (se presente)
- tipo (se houver)
- etiquetas (se houver)
- linha crua

Por exemplo, consultando `wmi_logical_disk_free_bytes`:



dessas linhas do Prometheus:

```
# HELP wmi_logical_disk_free_bytes Espaço livre em bytes (LogicalDisk.PercentFreeSpace)
# TYPE medidor wmi_logical_disk_free_bytes
wmi_logical_disk_free_bytes{volume="C:"} 3.5180249088e+11
wmi_logical_disk_free_bytes{volume="D:"} 2.627731456e+09
```

```
wmi_logical_disk_free_bytes{volume="HarddiskVolume4"} 4.59276288e+08
```

retornará:


```
[
    {
        "name": "wmi_logical_disk_free_bytes",
        "help": "Espaço livre em bytes (LogicalDisk.PercentFreeSpace)",
        "tipo": "medidor",
        "etiquetas": {
            "volume": "C:"
         },
        "valor": "3.5180249088e+11",
        "line_raw": "wmi_logical_disk_free_bytes{volume=\"C:\"} 3.5180249088e+11"
    },
    {
        "name": "wmi_logical_disk_free_bytes",
        "help": "Espaço livre em bytes (LogicalDisk.PercentFreeSpace)",
        "tipo": "medidor",
        "etiquetas": {
            "volume": "D:"
         },
        "valor": "2.627731456e+09",
        "line_raw": "wmi_logical_disk_free_bytes{volume=\"D:\"} 2.627731456e+09"
    },
    {
        "name": "wmi_logical_disk_free_bytes",
        "help": "Espaço livre em bytes (LogicalDisk.PercentFreeSpace)",
        "tipo": "medidor",
        "etiquetas": {
            "volume": "HarddiskVolume4"
         },
        "valor": "4.59276288e+08",
        "line_raw": "wmi_logical_disk_free_bytes{volume=\"HarddiskVolume4\"} 4.59276288e+08"
    }
]
```

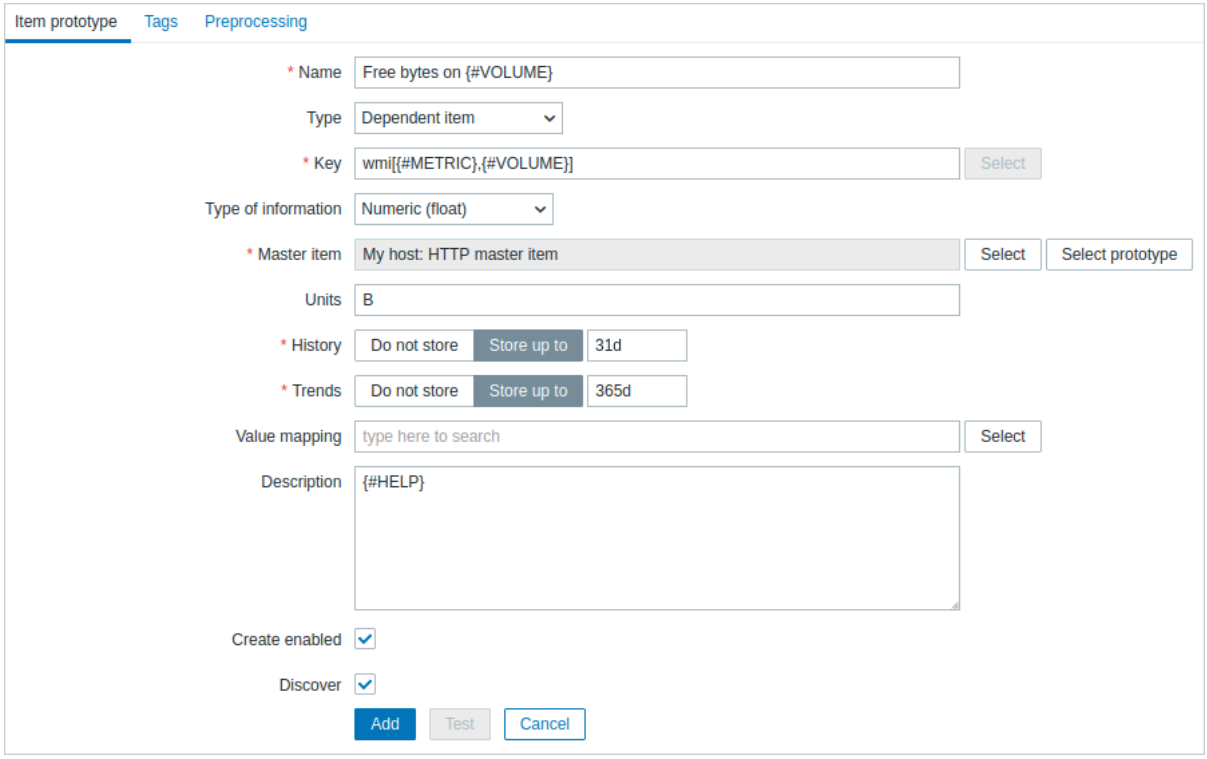

Mapeamento de macros LLD

Em seguida, você deve ir para a guia de macros LLD e fazer o seguinte mapeamentos:

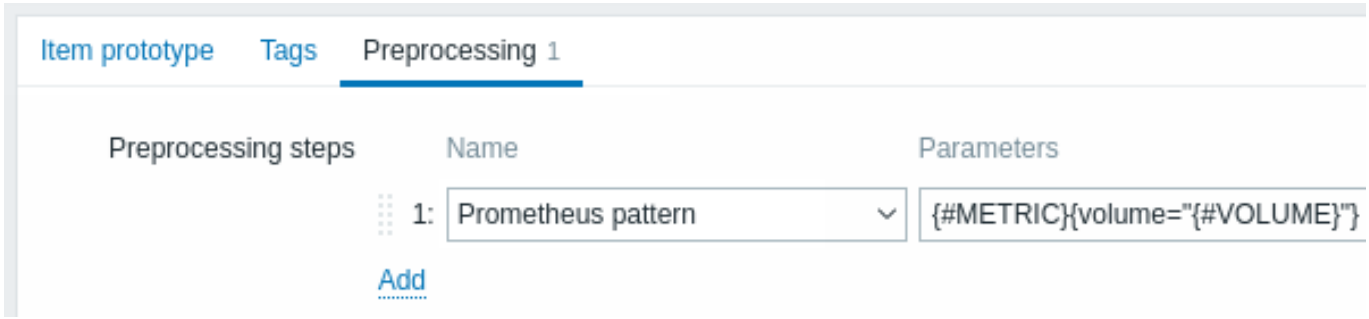
```
{#VOLUME}=$.labels['volume']
{#METRIC}=$['name']
{#HELP}=$['help']
```

Protótipo de item

Você pode querer criar um protótipo de item como este:



com opções de pré-processamento:



14 Descoberta de dispositivos de bloco

De forma semelhante à descoberta de sistemas de arquivos, também é possível descobrir dispositivos de bloco e seu tipo.

Chave do item

A chave do item a ser usada na regra de descoberta é

`vfs.dev.discovery`

Este item é suportado apenas em plataformas Linux.

Você pode criar regras de descoberta usando este item de descoberta e:

- filtro: **{#DEVNAME} corresponde a** `sd[\D]$` - para descobrir dispositivos chamados "sd0", "sd1", "sd2", ...
- filtro: **{#DEVTYPE} corresponde a** `disk` **E {#DEVNAME} não corresponde a** `^loop.*` - para descobrir dispositivos do tipo disco cujo nome não começa com "loop"

Macros compatíveis

Esta chave de descoberta retorna duas macros - {#DEVNAME} e {#DEVTYPE} identificando o nome e o tipo do dispositivo de bloco respectivamente, por exemplo:

```
[
   {
      "{#DEVNAME}":"loop1",
      "{#DEVTYPE}":"disco"
   },
   {
      "{#DEVNAME}":"dm-0",
```

```
        "{#DEVTYPE}":"disco"
    },
    {
        "{#DEVNAME}":"sda",
        "{#DEVTYPE}":"disco"
    },
    {
        "{#DEVNAME}":"sda1",
        "{#DEVTYPE}":"partição"
    }
]
```

A descoberta de dispositivo de bloco permite usar `vfs.dev.read[]` e itens `vfs.dev.write[]` para criar protótipos de itens usando o {#DEVNAME} macro, por exemplo:

- ”vfs.dev.read[{#DEVNAME},sps]”
- ”vfs.dev.write[{#DEVNAME},sps]”

{#DEVTYPE} destina-se à filtragem de dispositivos.

15 Descoberta de interfaces de host no Zabbix

Visão geral

É possível descobrir todos interfaces configuradas no frontend do Zabbix para um host.

Chave do item

O item a ser usado na regra de descoberta é o item interno do Zabbix:

`zabbix[host,discovery,interfaces]`

Este item retorna um JSON com a descrição das interfaces, incluindo:

- Endereço IP/nome DNS (dependendo da configuração “Conectar a” do host)
- Número da porta
- Tipo de interface (agent Zabbix, SNMP, JMX, IPMI)
- Se é a interface padrão ou não
- Se o recurso de requisição em lote está habilitado - apenas para interfaces SNMP.

Por exemplo:

```
[{"{#IF.CONN}":"192.168.3.1","{#IF.IP}":"192.168.3.1","{#IF.DNS}":"","{#IF.PORT}":"10050","{#IF.TYPE}":"AG
```

Com múltiplas interfaces, seus registros em JSON são ordenados por:

- Tipo de interface,
- Padrão - a interface padrão é colocada antes das interfaces não padrão,
- ID da interface (em ordem crescente).

Macros suportadas

As seguintes macros são suportadas para uso na regra de descoberta (filtro) e protótipos de items, triggers e gráficos:

Macro	Descrição
{#IF.CONN}	Endereço IP da interface ou nome DNS do host.
{#IF.IP}	Endereço IP da interface.
{#IF.DNS}	Nome DNS do host da interface.
{#IF.PORT}	Número da porta da interface.
{#IF.TYPE}	Tipo de interface (”AGENT”, ”SNMP”, ”JMX” ou ”IPMI”).
{#IF.DEFAULT}	Status padrão para a interface: 0 - não é a interface padrão 1 - interface padrão
{#IF.SNMP.BULK}	Status do processamento SNMP bulk para a interface: 0 - desabilitado 1 - habilitado Esta macro é retornada apenas se o tipo de interface for “SNMP”.

7 Regras LLD personalizadas

Visão geral

Também é possível criar uma regra de LLD completamente personalizada, descobrindo qualquer tipo de entidade - por exemplo, bancos de dados em um servidor de banco de dados.

Para isso, um item personalizado deve ser criado que retorne uma string JSON, especificando os objetos encontrados e, opcionalmente, algumas propriedades deles. A quantidade de macros por entidade não é limitada - enquanto as regras de descoberta internas retornam uma ou duas macros (por exemplo, duas para descoberta de sistema de arquivos), é possível retornar mais.

Exemplo

O formato de string JSON necessário é melhor ilustrado com um exemplo. Suponha que estamos executando um agent Zabbix 1.8 antigo (um que não suporta "vfs.fs.discovery"), mas ainda precisamos descobrir sistemas de arquivos. Aqui está um script Perl simples para Linux que descobre sistemas de arquivos montados e gera a saída em JSON, incluindo o nome e o tipo do sistema de arquivos. Uma maneira de usá-lo seria como um UserParameter com a chave "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:
Os símbolos permitidos para nomes de macros LLD são **0-9** , **A-Z** , **_** , **.** Letras minúsculas não são suportadas nos nomes.

Um exemplo de sua saída (reformatado para maior clareza) é mostrado abaixo. O JSON para verificações de descoberta personalizadas deve seguir o mesmo formato.

```
[
    { "{#FSNAME}":"/",                          "{#FSTYPE}":"rootfs"   },
    { "{#FSNAME}":"/sys",                       "{#FSTYPE}":"sysfs"    },
    { "{#FSNAME}":"/proc",                      "{#FSTYPE}":"proc"     },
    { "{#FSNAME}":"/dev",                       "{#FSTYPE}":"devtmpfs" },
    { "{#FSNAME}":"/dev/pts",                   "{#FSTYPE}":"devpts"   },
    { "{#FSNAME}":"/lib/init/rw",               "{#FSTYPE}":"tmpfs"    },
    { "{#FSNAME}":"/dev/shm",                   "{#FSTYPE}":"tmpfs"    },
    { "{#FSNAME}":"/home",                      "{#FSTYPE}":"ext3"     },
    { "{#FSNAME}":"/tmp",                       "{#FSTYPE}":"ext3"     },
    { "{#FSNAME}":"/usr",                       "{#FSTYPE}":"ext3"     },
    { "{#FSNAME}":"/var",                       "{#FSTYPE}":"ext3"     },
    { "{#FSNAME}":"/sys/fs/fuse/connections",   "{#FSTYPE}":"fusectl"  }
]
```

No exemplo anterior, é necessário que as chaves correspondam aos nomes das macros LLD usadas nos protótipos, a alternativa é extrair os valores das macros LLD usando JSONPath {#FSNAME} → $.fsname e {#FSTYPE} → $.fstype, tornando assim possível tal script:

```
####!/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";
```

Um exemplo de sua saída (reformatado para maior clareza) é mostrado abaixo. O JSON para verificações de descoberta personalizadas deve seguir o mesmo formato.

```
[
    { "fsname":"/",                          "fstype":"rootfs"   },
    { "fsname":"/sys",                       "fstype":"sysfs"    },
    { "fsname":"/proc",                      "fstype":"proc"     },
    { "fsname":"/dev",                       "fstype":"devtmpfs" },
    { "fsname":"/dev/pts",                   "fstype":"devpts"   },
    { "fsname":"/lib/init/rw",               "fstype":"tmpfs"    },
    { "fsname":"/dev/shm",                   "fstype":"tmpfs"    },
    { "fsname":"/home",                      "fstype":"ext3"     },
    { "fsname":"/tmp",                       "fstype":"ext3"     },
    { "fsname":"/usr",                       "fstype":"ext3"     },
    { "fsname":"/var",                       "fstype":"ext3"     },
    { "fsname":"/sys/fs/fuse/connections",   "fstype":"fusectl"  }
]
```

Então, no campo ”Filtro” da regra de descoberta, poderíamos especificar ”{#FSTYPE}” como uma macro e ”rootfs|ext3” como uma expressão regular.

Note:
Você não precisa usar nomes de macro FSNAME/FSTYPE com regras LLD personalizadas, você é livre para usar os nomes que quiser. Caso o JSONPath seja usado, a linha LLD será um elemento de array que pode ser um objeto, mas também pode ser outro array ou um valor.

Observe que, se estiver usando um parâmetro de usuário, o valor de retorno é limitado a 16 MB. Para mais detalhes, consulte limites de dados para valores de retorno do LLD.

16 Monitoração distribuída

Visão geral O Zabbix fornece uma maneira eficaz e confiável de monitorar uma infraestrutura de TI distribuída usando Zabbix proxies.

Os proxies podem ser usados para coletar dados localmente em nome de um Zabbix server centralizado e, em seguida, reportar os dados ao server.

Recursos do proxy

Ao decidir usar ou não um proxy, várias considerações devem ser levadas em conta.

	Proxy
Leve	**Sim**
GUI	Não
Funciona independentemente	**Sim**
Fácil manutenção	**Sim**
Criação automática de banco de dados	**Sim**[1]
Administração local	Não
Pronto para hardware embarcado	**Sim**
Conexões TCP unidirecionais	**Sim**
Configuração centralizada	**Sim**
Gera notificações	Não

[1] O recurso de criação automática de banco de dados funciona apenas com SQLite. Outros bancos de dados suportados requerem configuração manual.

Attention:
O proxy Zabbix não reconhece períodos de manutenção; veja Cálculo de filas durante a manutenção para mais detalhes.

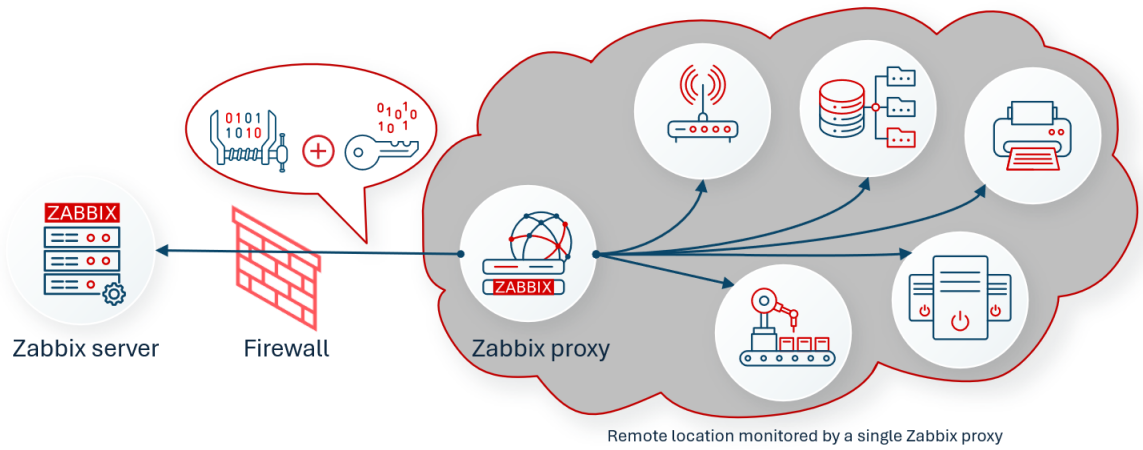
1 Proxies

Visão geral Um proxy Zabbix pode coletar dados de desempenho e disponibilidade em nome do Zabbix server. Dessa forma, um proxy pode assumir parte da carga de coleta de dados e aliviar o Zabbix server.

Além disso, usar um proxy é a maneira mais fácil de implementar o monitoramento centralizado e distribuído, quando todos os agents e proxies reportam para um único Zabbix server e todos os dados são coletados centralmente.

Um proxy Zabbix pode ser usado para:

- Monitorar locais remotos
- Monitorar locais com comunicações não confiáveis
- Aliviar o Zabbix server ao monitorar milhares de dispositivos
- Simplificar a manutenção do monitoramento distribuído



Remote location monitored by a single Zabbix proxy

O proxy requer apenas uma conexão TCP com o Zabbix server. Dessa forma, é mais fácil contornar um firewall, pois você só precisa configurar uma regra de firewall.

Attention:
O proxy Zabbix deve usar um banco de dados separado. Apontá-lo para o banco de dados do Zabbix server irá quebrar a configuração.

Todos os dados coletados pelo proxy são armazenados localmente antes de serem transmitidos para o server. Dessa forma, nenhum dado é perdido devido a problemas temporários de comunicação com o server. Os parâmetros *ProxyLocalBuffer* e *ProxyOfflineBuffer* no arquivo de configuração do proxy controlam por quanto tempo os dados são mantidos localmente.

Attention:
Pode acontecer de um proxy, que recebe as últimas alterações de configuração diretamente do banco de dados do Zabbix server, ter uma configuração mais atualizada do que o Zabbix server, cuja configuração pode não ser atualizada tão rapidamente devido ao valor de CacheUpdateFrequency. Como resultado, o proxy pode começar a coletar dados e enviá-los para o Zabbix server, que ignora esses dados.

O proxy Zabbix é um coletor de dados. Ele não calcula triggers, processa eventos ou envia alertas. Para uma visão geral da funcionalidade do proxy, consulte a tabela a seguir:

Função		Suportado pelo proxy
Items		
	Zabbix agent checks	**Sim**
	Zabbix agent checks (active)	**Sim** [1]
	Simple checks	**Sim**
	Trapper items	**Sim**
	SNMP checks	**Sim**
	SNMP traps	**Sim**
	IPMI checks	**Sim**
	JMX checks	**Sim**
	Log file monitoring	**Sim**
	Internal checks	**Sim**
	SSH checks	**Sim**
	Telnet checks	**Sim**
	External checks	**Sim**
	Dependent items	**Sim**
	Script items	**Sim**
	Browser items	**Sim**
Monitoramento web embutido		**Sim**
Pré-processamento de valor de item		**Sim**
Descoberta de rede		**Sim**
Autoregistro de agent ativo		**Sim**
Descoberta de baixo nível		**Sim** [2]
Comandos remotos		**Sim**
Cálculo de triggers		*Não*
Processamento de eventos		*Não*
Correlação de eventos		*Não*
Envio de alertas		*Não*

Note:
[1] Para garantir que um agent solicite ao proxy (e não ao server) as verificações ativas, o proxy deve estar listado no parâmetro **ServerActive** no arquivo de configuração do agent.
 [2] Para LLD, o proxy Zabbix apenas coleta e pré-processa os dados e, em seguida, os envia para o Zabbix server para processamento adicional.

Proteção contra sobrecarga

Se o Zabbix server ficar fora do ar por algum tempo e os proxies coletarem muitos dados, e então o server for iniciado, ele pode ficar sobrecarregado (o uso do cache de histórico permanece em 95-100% por algum tempo). Essa sobrecarga pode resultar em uma queda de desempenho, onde as verificações são processadas mais lentamente do que deveriam. A proteção contra esse cenário foi implementada para evitar problemas que surgem devido à sobrecarga do cache de histórico.

Quando o cache de histórico do Zabbix server está cheio, o acesso de gravação ao cache de histórico é reduzido, paralisando os processos de coleta de dados do server. O caso mais comum de sobrecarga do cache de histórico é após o tempo de inatividade do server, quando os proxies estão enviando os dados coletados. Para evitar isso, foi adicionada a limitação de proxy (atualmente não pode ser desabilitada).

O Zabbix server deixará de aceitar dados dos proxies quando o uso do cache de histórico atingir 80%. Em vez disso, esses proxies serão colocados em uma lista de limitação. Isso continuará até que o uso do cache caia para 60%. Agora o server começará a aceitar dados dos proxies um por um, definidos pela lista de limitação. Isso significa que o primeiro proxy que tentou enviar dados durante o período de limitação será atendido primeiro e, até que termine, o server não aceitará dados de outros proxies.

Esse modo de limitação continuará até que o uso do cache atinja 80% novamente, caia para 20% ou a lista de limitação fique vazia. No primeiro caso, o server deixará de aceitar dados do proxy novamente. Nos outros dois casos, o server voltará a funcionar normalmente, aceitando dados de todos os proxies.

As informações acima podem ser ilustradas na tabela a seguir:

Uso do cache de gravação do histórico	Modo do Zabbix server	Ação do Zabbix server
Atinge 80%	Aguardando	Para de aceitar dados do proxy, mas mantém uma *lista de limitação* (lista priorizada de proxies a serem contatados posteriormente).
Cai para 60%	Limitado	Começa a processar a lista de limitação, mas ainda não aceita dados do proxy.
Cai para 20%	Normal	Descarta a lista de limitação e começa a aceitar dados do proxy normalmente.

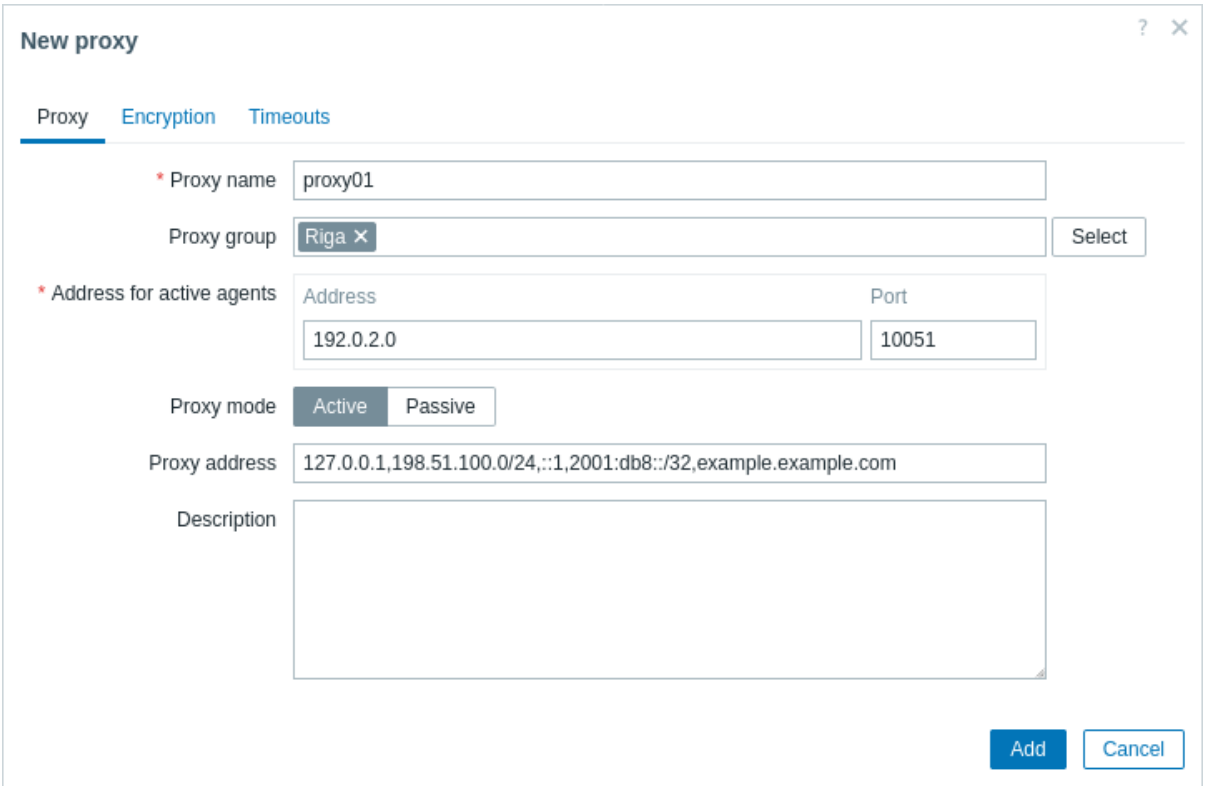
Você pode usar o item interno zabbix[wcache,history,pused] para correlacionar esse comportamento do Zabbix server com uma métrica.

Configuração Depois de instalar e configurar um proxy, é hora de configurá-lo no frontend do Zabbix.

Adicionando proxies

Para configurar um proxy no frontend do Zabbix:

- Vá para: *Administração* → *Proxies*
- Clique em *Criar proxy*



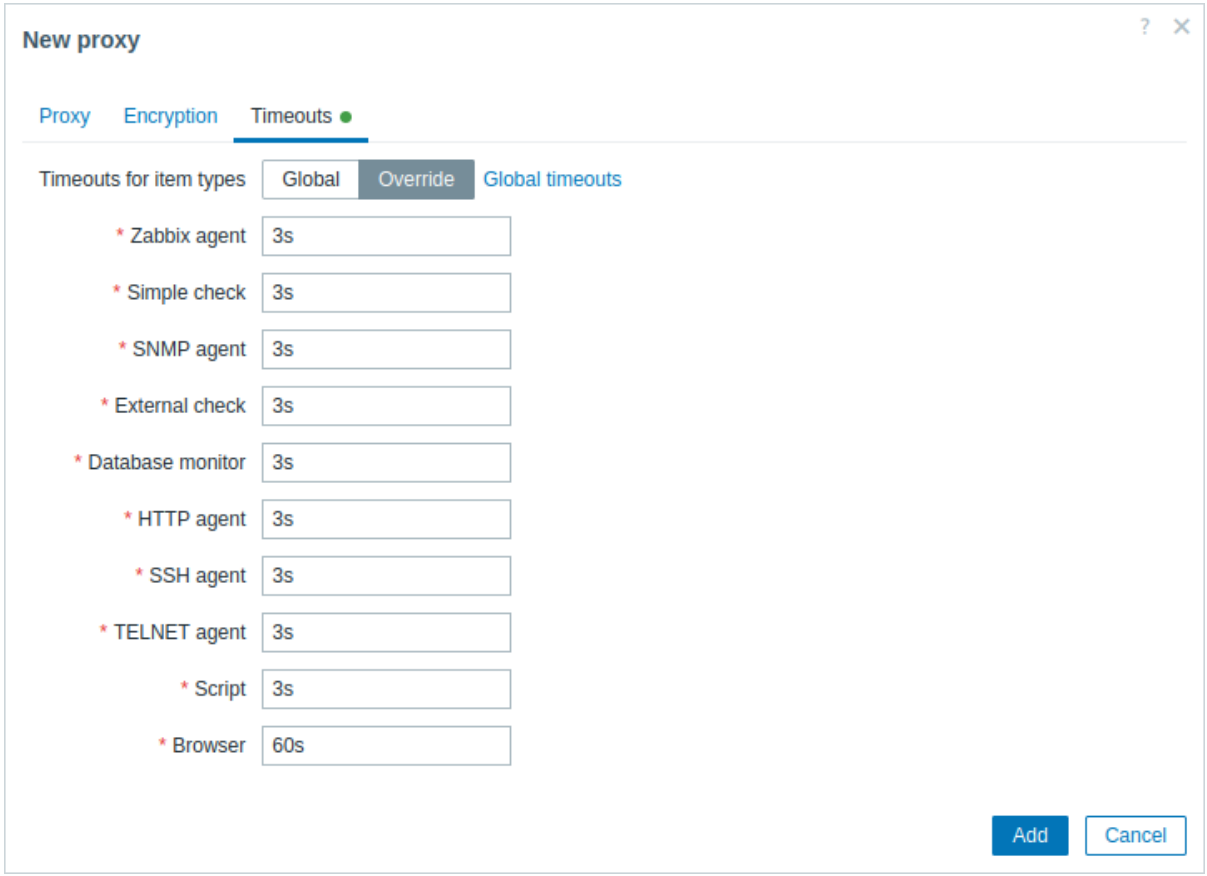
Parâmetro	Descrição
Nome do proxy	Digite o nome do proxy. Deve ser o mesmo nome do parâmetro *Hostname* no arquivo de configuração do proxy.
Grupo de proxy	Selecione um grupo de proxy para balanceamento de carga/alta disponibilidade do proxy.
Endereço para agents ativos	Digite o endereço ao qual os agents ativos monitorados ou senders devem se conectar. Suportado **apenas** para agents Zabbix 7.0 ou posterior. Este endereço é usado para conectar tanto proxies ativos quanto passivos. Este campo só está disponível se um grupo de proxy for selecionado no campo *Grupo de proxy*.
Endereço	Endereço IP/nome DNS para conectar.
Porta	Número da porta TCP (10051 por padrão) para conectar. Macros de usuário são suportadas.

Parâmetro	Descrição
Modo do proxy	Selecione o modo do proxy. **Ativo** - o proxy irá se conectar ao Zabbix server e solicitar dados de configuração **Passivo** - o Zabbix server se conecta ao proxy *Nota* que, sem comunicações criptografadas, os dados de configuração do proxy (sensíveis) podem ficar disponíveis para partes que tenham acesso à porta trapper do Zabbix server ao usar um proxy ativo. Isso é possível porque qualquer um pode se passar por um proxy ativo e solicitar dados de configuração se a autenticação não ocorrer ou se os endereços do proxy não forem limitados no campo *Endereço do proxy*.
Endereço do proxy	Se especificado, as solicitações do proxy ativo só serão aceitas desta lista de endereços IP separados por vírgula, opcionalmente em notação CIDR, ou nomes DNS do proxy Zabbix ativo. Este campo só está disponível se um proxy ativo for selecionado no campo *Modo do proxy*. Macros não são suportadas.
Interface	Digite os detalhes da interface para um proxy passivo. Este campo só está disponível se um proxy passivo for selecionado no campo Modo do proxy.
Endereço	Endereço IP/nome DNS do proxy passivo.
Porta	Número da porta TCP do proxy passivo (10051 por padrão). Macros de usuário são suportadas.
Descrição	Digite a descrição do proxy.

A aba **Criptografia** permite exigir conexões criptografadas com o proxy.

Parâmetro	Descrição
Conexões para o proxy	Como o server se conecta ao proxy passivo: sem criptografia (padrão), usando PSK (chave pré-compartilhada) ou certificado.
Conexões do proxy	Selecione que tipo de conexões são permitidas do proxy ativo. Vários tipos de conexão podem ser selecionados ao mesmo tempo (útil para testes e troca para outro tipo de conexão). O padrão é ”Sem criptografia”.
Emissor	Emissor permitido do certificado. O certificado é primeiro validado com a CA (autoridade certificadora). Se for válido, assinado pela CA, então o campo *Emissor* pode ser usado para restringir ainda mais a CA permitida. Este campo é opcional, destinado ao uso se sua instalação do Zabbix usar certificados de várias CAs.
Sujeito	Sujeito permitido do certificado. O certificado é primeiro validado com a CA. Se for válido, assinado pela CA, então o campo *Sujeito* pode ser usado para permitir apenas um valor da string *Sujeito*. Se este campo estiver vazio, qualquer certificado válido assinado pela CA configurada é aceito.
Identidade PSK	String de identidade da chave pré-compartilhada. Não coloque informações sensíveis na identidade PSK, ela é transmitida sem criptografia pela rede para informar ao receptor qual PSK usar.
PSK	Chave pré-compartilhada (hex-string). Comprimento máximo: 512 dígitos hexadecimais (PSK de 256 bytes) se o Zabbix usar a biblioteca GnuTLS ou OpenSSL, 64 dígitos hexadecimais (PSK de 32 bytes) se o Zabbix usar a biblioteca mbed TLS (PolarSSL). Exemplo: 1f87b595725ac58dd977beef14b97461a7c1045b9a1c963065002c5473194952

A aba **Timeouts** permite substituir os timeouts globais para tipos de item que suportam isso.



Parâmetro	Descrição
Timeouts para tipos de item	Selecione a opção de timeout: **Global** - o timeout global é usado (exibido no campo *Timeout* esmaecido para cada tipo de item); **Substituir** - o timeout personalizado é usado (definido no campo *Timeout* para cada tipo de item). Faixa permitida: 1 - 600s (padrão: herdado dos timeouts globais). Sufixos de tempo, por exemplo, 30s, 1m, e macros de usuário são suportados. Clicar no link *Timeouts globais* permite configurar os timeouts globais. Observe que o link *Timeouts globais* é visível apenas para usuários do tipo *Super admin* com permissões para a seção de frontend *Administração* → *Geral*. Tipos de item suportados: - Zabbix agent (verificações passivas e ativas) - Verificação simples (exceto itens `icmpping*`, `vmware.*`) - SNMP agent (apenas para itens SNMP `walk[OID]` e `get[OID]`) - Verificação externa - Monitor de banco de dados - HTTP agent - SSH agent - TELNET agent - Script - Browser Observe que os timeouts definidos em **Substituir** prevalecerão sobre os globais, mas serão substituídos por timeouts individuais de item se estes forem definidos na configuração do item.

Note:

Se a versão principal do proxy não corresponder à versão principal do server, o ícone será exibido ao lado de *Timeouts para tipos de item*, com a mensagem ao passar o mouse "Timeouts desabilitados porque as versões do proxy e do server não correspondem". Nesses casos, o proxy usará o parâmetro `Timeout` do arquivo de configuração do proxy.

O formulário de edição de um proxy existente possui os seguintes botões adicionais:

- *Atualizar configuração* - atualiza a configuração do proxy
- *Clonar* - cria um novo proxy com base nas propriedades do proxy existente
- *Excluir* - exclui o proxy

Configuração do host

Você pode especificar que um host individual deve ser monitorado por um proxy ou grupo de proxies no formulário de configuração do host, usando o campo *Monitorado por*.

A atualização em massa de hosts é outra maneira de especificar que hosts devem ser monitorados por um proxy ou grupo de proxies.

1 Sincronização da configuração de monitoramento

Esta página fornece detalhes da atualização da configuração do monitoramento do Zabbix Proxy, ou seja, como as mudanças realizadas no monitoramento do lado do Zabbix Server são sincronizadas com o Zabbix Proxy.

Atualização incremental

A atualização da configuração do proxy é incremental. Durante uma sincronização de configuração, apenas as entidades modificadas são atualizadas (assim, se nenhuma entidade foi modificada, nada será enviado). Essa abordagem permite economizar recursos e definir um intervalo menor (quase instantâneo) para a atualização da configuração do proxy.

As alterações de configuração do proxy são rastreadas usando números de revisão. Apenas as entidades com revisões maiores que a revisão da configuração do proxy são incluídas nos dados de configuração enviados ao proxy.

As entidades para uma sincronização de configuração são as seguintes:

Entidade	Detalhes
autoregistration tls data	Todos os dados TLS de autorregistro.
expressions	Todas as expressões (expressões regulares, tabelas de expressões).
global configuration	Configuração global definida na tabela 'config'
host	Todas as propriedades, interfaces, inventário, items, pré-processamento de items, parâmetros de items, cenários web de um host.
host macros	Todas as macros definidas em um host e todos os IDs de template vinculados a ele.
proxy discovery rule	Regras de descoberta e verificações atribuídas a um proxy.

Isso significa:

- Se um item for alterado em um **host**, toda a configuração desse host será sincronizada.
- Se uma **expressão regular** for alterada, todas as expressões regulares serão sincronizadas.

Uma exceção são as macros de host, que também são enviadas se qualquer coisa no host tiver sido alterada.

O comando `-R config_cache_reload` no proxy também iniciará uma atualização incremental.

Observe que uma sincronização completa da configuração ocorrerá ao iniciar/reiniciar um proxy, failover de HA, se o token de sessão tiver sido alterado ou se a atualização da configuração falhar no proxy, por exemplo, se a conexão for interrompida ao receber os dados de configuração.

Parâmetros de Configuração:

O parâmetro **ProxyConfigFrequency** define com qual frequência o Zabbix Proxy deverá sincronizar com o Zabbix Server (por padrão, este valor está definido como 10 segundos).

Observe que ProxyConfigFrequency é:

- Parâmetro do Zabbix Server para os proxies passivos;
- Parâmetro do Zabbix Proxy para proxies ativos.

Em proxies ativos, **ProxyConfigFrequency** é um parâmetro novo, desde o Zabbix 6.4 e precisa ser utilizado no lugar do agora obsoleto **ConfigFrequency**.

::: noteimportantSe ambos os parâmetros ProxyConfigFrequency e ConfigFrequency forem utilizados, o proxy irá registrar um erro no log, e irá encerrar o processo.:::

Cálculo das filas durante a manutenção

> **Attention:**
> O proxy Zabbix não tem conhecimento dos períodos de manutenção; veja Cálculo das filas durante a manutenção para detalhes.

2 Balanceamento de carga e alta disponibilidade do proxy

Visão geral

Os proxies do Zabbix podem ser organizados em grupos de proxies para permitir balanceamento de carga e alta disponibilidade de proxies.

O balanceamento de carga e alta disponibilidade de proxies é a redistribuição automática de hosts entre proxies dentro de um grupo de proxies:

- Se um proxy ficar offline, seus hosts serão movidos para outros proxies, mantendo assim alta disponibilidade do proxy.
- Se um proxy tiver um número muito maior/menor de hosts do que outros proxies, seus hosts serão movidos para outros proxies para equilibrar a carga do proxy.

A redistribuição de hosts funciona apenas entre proxies em um grupo que atendam às seguintes condições:

- Os proxies estão executando o Zabbix 7.0 ou posterior.
- A versão do proxy corresponde à versão do Zabbix server. Se estiver usando o Zabbix agent (passivo), a versão do proxy deve corresponder à versão do agent. Agents ativos exigem apenas o Zabbix 7.0 ou posterior.
- O grupo de proxies possui um estado online.
- Os hosts estão configurados para serem monitorados por um grupo de proxies em vez de proxies individuais.

> **Note:**
> A integridade do grupo de proxies pode ser monitorada com verificações internas por qualquer host atribuído a um grupo de proxies. No entanto, para monitorar a integridade de um único proxy em um grupo, atribua o host a esse proxy; caso contrário, os resultados podem ser inconsistentes.

Redistribuição de hosts

O balanceamento de carga e alta disponibilidade do proxy é gerenciado pelo Zabbix server por meio do gerenciador de grupo de proxy, que monitora continuamente o estado de todos os proxies em cada grupo de proxy e sua distribuição de hosts.

A alta disponibilidade do proxy dentro de um grupo é garantida por meio do failover do proxy: quando um proxy fica offline, seus hosts são imediatamente redistribuídos para outros proxies. O balanceamento de carga do proxy também ocorre, pois os hosts são realocados para proxies com menos hosts atribuídos.

Além disso, o balanceamento de carga do proxy é acionado quando a contagem de hosts do proxy difere da média do grupo em pelo menos 10 hosts e um fator de 2 (excesso ou déficit de hosts). Se o desequilíbrio persistir após um período de carência (10 x failover delay), o grupo de proxy é colocado na fila para redistribuição de hosts.

O gerenciador de grupo de proxy redistribui os hosts usando a seguinte lógica:

1. Calcule o número médio de hosts por proxy.
2. Para proxies com excesso de hosts — mova os hosts excedentes para o pool não atribuído de proxies.
3. Para proxies com déficit de hosts — calcule quantos hosts são necessários para atingir o equilíbrio.
4. Remova o número necessário de hosts dos proxies com mais hosts.
5. Mova os hosts não atribuídos para os proxies com menos hosts.

Exemplos de redistribuição de hosts:

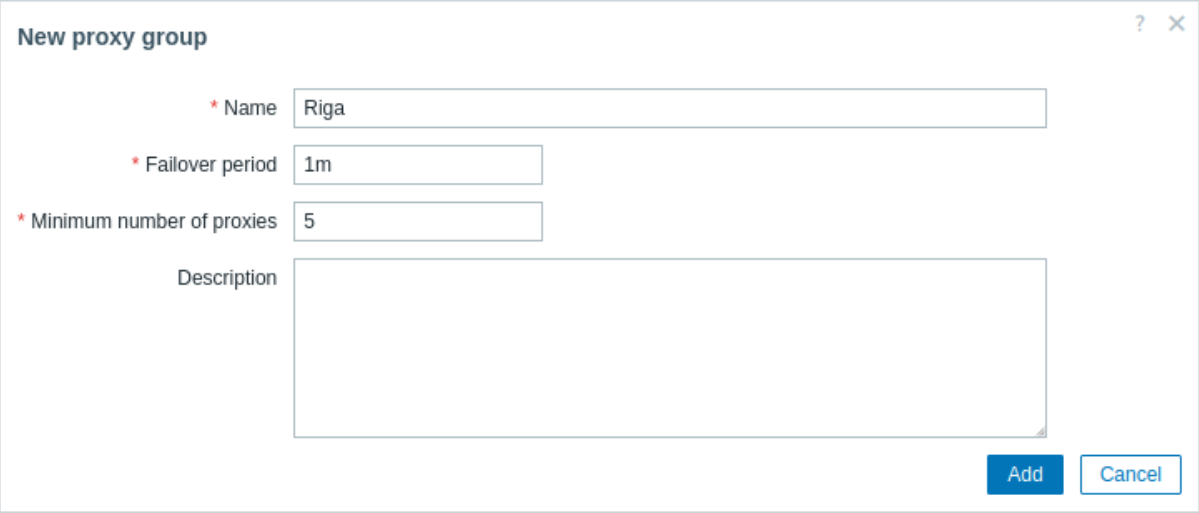
Hosts no proxy	Média do grupo	Reatribuição de hosts
100	50	Sim
60	50	Não
40	50	Não
25	50	Sim
15	5	Sim
10	5	Não

Attention:
Ter menos de 10 hosts monitorados por um grupo de proxy pode levar a uma distribuição desigual de hosts entre os proxies do grupo.

Configurando um grupo de proxies

Para configurar um grupo de proxies no frontend do Zabbix:

1. Vá para *Administração* > *Grupos de proxies*
2. Clique em *Criar grupo de proxies*



Parâmetro	Descrição
Nome	Nome do grupo de proxies.
Período de failover	Período em segundos durante o qual um proxy no grupo de proxies deve se comunicar com o Zabbix server para ser considerado online (padrão: 1m; intervalo: 10s–15m). Se o proxy não se comunicar dentro desse período, o estado do proxy será alterado para *Offline* e seus hosts serão imediatamente redistribuídos para outros proxies. O balanceamento de carga do proxy começa após 10 x esse período. Suporta sufixos de tempo (por exemplo, 30s, 1m) e macros de usuário.
Número mínimo de proxies	Número mínimo de proxies online necessários para manter o grupo de proxies online (padrão: 1; intervalo: 1–1000). Suporta macros de usuário. Esse valor deve ser menor que o número total de proxies no grupo. Por exemplo, em um grupo de 10 proxies, definir o mínimo para 10 fará com que o grupo fique offline se algum proxy falhar. Observe que proxies online em um grupo offline continuam funcionando normalmente, mas o balanceamento de carga/alta disponibilidade não ocorrerá.
Descrição	Descrição do grupo de proxies.
Proxies	Exibe uma lista de até cinco proxies (como links ou em texto simples, dependendo das permissões do usuário para proxies) ao editar um grupo com proxies.

Configurando o balanceamento de carga do proxy

Para usar o balanceamento de carga do proxy, você precisa configurar um grupo de proxies no frontend do Zabbix (veja acima) e garantir que os hosts sejam monitorados por um grupo de proxies, e não por proxies individuais (você pode usar a atualização em massa de hosts para mover hosts de proxies para o grupo de proxies).

Se estiver usando o agent Zabbix, também configure-o da seguinte forma:

- Para cheques passivos, liste todos os proxies do grupo de proxies no parâmetro Server.
- Para cheques ativos, recomenda-se listar todos os proxies do grupo de proxies ou o server Zabbix no parâmetro ServerActive. Observe que apenas o agent Zabbix 7.0 (ou posterior) funcionará com grupos de proxies em seu modo ativo.

Se o parâmetro `ServerActive` incluir apenas um proxy de um grupo de proxies (ou o server Zabbix), o agent ainda poderá se conectar ao proxy correto. Quando o serviço do agent é iniciado e se conecta ao proxy especificado, o agent receberá e armazenará em cache a lista completa de IPs dos proxies e sua carga atual dentro do grupo. Em seguida, os cheques ativos serão redirecionados para o proxy online correto para o host, com base na atribuição atual de proxy-host dentro do grupo de proxies.

Warning:
Ter apenas um único proxy especificado no parâmetro `ServerActive` do agent Zabbix pode levar à perda de dados de monitoramento se o agent for iniciado/reiniciado enquanto o proxy especificado estiver offline.

Note:
Ao usar o Zabbix sender, as solicitações de dados também são redirecionadas para o proxy online correto para o host, com base na atribuição atual de proxy-host dentro do grupo de proxies. No entanto, se você estiver enviando valores de vários hosts a partir de um arquivo de entrada, use a opção `-g` para evitar o envio de dados para o proxy errado.

O agent Zabbix também deve ser capaz de se conectar a todos os proxies do grupo de proxies através do firewall. Caso contrário, os cheques ativos podem travar ou falhar durante o redirecionamento ou failover. Por exemplo:

- Durante cheques ativos, um proxy pode redirecionar o agent para outro proxy. Se esse proxy estiver bloqueado por um firewall, a comunicação ficará travada enquanto aguarda uma resposta.
- Em configurações de alta disponibilidade estáveis, sem rebalanceamento recente, os agents podem nunca contatar proxies de backup. Se as regras do firewall mudaram e não foram testadas, o failover pode falhar.

Testando o balanceamento de carga do proxy

Para testar o balanceamento de carga do proxy:

1. Configure um grupo de proxies.
2. Certifique-se de que o grupo de proxies está com o estado online.
3. Certifique-se de que os hosts estão sendo monitorados por um grupo de proxies, e não por proxies individuais (você pode usar a atualização em massa de hosts para mover hosts de proxies para o grupo de proxies).
4. Aguarde alguns segundos para a atualização da configuração e distribuição dos hosts entre os proxies do grupo. Observe a alteração atualizando a lista de hosts em *Administração* > *Proxies*.

Notas importantes

- Traps SNMP não são suportados por proxies em um grupo de proxies.
- Verificações que dependem de configuração externa (por exemplo, scripts para verificações externas ou configuração ODBC para verificações de banco de dados) devem ter a mesma configuração em todos os proxies do grupo de proxies.
- Verificações de banco de dados requerem permissões estendidas no objeto/servidor de banco de dados.
- Hosts VMware monitorados por um grupo de proxies serão distribuídos aleatoriamente entre os proxies do grupo. Isso faz com que cada proxy armazene em cache todos os dados do VMware, resultando em carga adicional no vCenter.
- Hosts criados com base em dados de autorregistro de um proxy em um grupo de proxies são definidos para serem monitorados por aquele grupo de proxies. No entanto, hosts criados com base em dados de descoberta de rede de um proxy em um grupo de proxies são definidos para serem monitorados por aquele proxy.

17. Criptografia

Visão geral O Zabbix suporta comunicações criptografadas entre os componentes do Zabbix usando o protocolo Transport Layer Security (TLS) v.1.2 e 1.3 (dependendo da biblioteca de criptografia). A criptografia baseada em certificado e baseada em chave pré-compartilhada é suportada.

A criptografia pode ser configurada para conexões:

- Entre o Zabbix server, Zabbix proxy, Zabbix agent, Zabbix web service, utilitários zabbix_sender e zabbix_get
- Para o banco de dados Zabbix a partir do Zabbix frontend e server/proxy

A criptografia é opcional e configurável para componentes individuais:

- Alguns proxies e agents podem ser configurados para usar criptografia baseada em certificado com o server, enquanto outros podem usar criptografia baseada em chave pré-compartilhada, e ainda outros continuam com comunicações não criptografadas (como antes)
- O server (proxy) pode usar diferentes configurações de criptografia para diferentes hosts

Os programas daemon do Zabbix usam uma porta de escuta para conexões de entrada criptografadas e não criptografadas. Adicionar uma criptografia não requer a abertura de novas portas em firewalls.

Limitações

- As chaves privadas são armazenadas em texto simples em arquivos legíveis pelo Zabbix componentes durante a inicialização

- Chaves pré-compartilhadas são inseridas no frontend do Zabbix e armazenadas no Zabbix banco de dados em texto simples
- A criptografia integrada não protege as comunicações:
  - Entre o servidor web rodando o frontend Zabbix e a web do usuário navegador
  - Entre o frontend Zabbix e o servidor Zabbix
- Atualmente, cada conexão criptografada é aberta com um handshake TLS completo, nenhum cache de sessão e tickets são implementados
- A adição de criptografia aumenta o tempo para verificações e ações de itens, dependendo da latência da rede:
  - Por exemplo, se o atraso do pacote for de 100ms, abrir um TCP conexão e envio de solicitação não criptografada leva cerca de 200ms. Com criptografia são adicionados cerca de 1000 ms para estabelecer o TLS conexão;
  - Os tempos limite podem precisar ser aumentados, caso contrário, alguns itens e ações que executam scripts remotos em agentes podem funcionar com conexões não criptografadas, mas falham com o tempo limite com criptografado.
- A criptografia não é suportada por network descoberta. Verificações do agente Zabbix realizada pela descoberta de rede não será criptografada e se o Zabbix agente está configurado para rejeitar conexões não criptografadas, tais verificações não terá sucesso.

Compilando o Zabbix com suporte a criptografia Para suportar criptografia, o Zabbix deve ser compilado e vinculado com uma das bibliotecas de criptografia suportadas:

- GnuTLS - a partir da versão 3.1.18
- OpenSSL - versões 1.0.1, 1.0.2, 1.1.0, 1.1.1, 3.0.x
- LibreSSL - testado com as versões 2.7.4, 2.8.2:
  - LibreSSL 2.6.x não é suportado
  - LibreSSL é suportado como um substituto compatível do OpenSSL; as novas funções de API específicas do LibreSSL `tls_*()` não são usadas. Componentes do Zabbix compilados com LibreSSL não poderão usar PSK, apenas certificados podem ser usados.

Note:
Você pode saber mais sobre como configurar SSL para o frontend do Zabbix consultando estas melhores práticas.

A biblioteca é selecionada especificando a respectiva opção para o script "configure":

- `--with-gnutls[=DIR]`
- `--with-openssl[=DIR]` (também usado para LibreSSL)

Por exemplo, para configurar os fontes para o server e o agent com *OpenSSL* você pode usar algo como:

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

Diferentes componentes do Zabbix podem ser compilados com diferentes bibliotecas de criptografia (por exemplo, um server com *OpenSSL*, um agent com *GnuTLS*).

Attention:
Se você planeja usar chaves pré-compartilhadas (PSK), considere usar as bibliotecas *GnuTLS* ou *OpenSSL 1.1.0* (ou mais recentes) nos componentes do Zabbix que usam PSKs. As bibliotecas *GnuTLS* e *OpenSSL 1.1.0* suportam ciphersuites PSK com Perfect Forward Secrecy. Versões mais antigas da biblioteca *OpenSSL* (1.0.1, 1.0.2c) também suportam PSKs, mas os ciphersuites PSK disponíveis não fornecem Perfect Forward Secrecy.

Gerenciamento de criptografia de conexão As conexões no Zabbix podem usar:

- sem criptografia (padrão)
- Criptografia baseada em certificado RSA
- Criptografia baseada em PSK

Existem dois parâmetros importantes usados para especificar a criptografia entre os componentes do Zabbix:

- TLSConnect - especifica qual criptografia usar para conexões de saída (sem criptografia, PSK ou certificado)
- TLSAccept - especifica quais tipos de conexões são permitidos para conexões de entrada (sem criptografia, PSK ou certificado). Um ou mais valores podem ser especificados.

`TLSConnect` é usado nos arquivos de configuração do proxy Zabbix (no modo ativo, especifica apenas conexões para o server) e do agent Zabbix (para verificações ativas). No frontend do Zabbix, o equivalente ao TLSConnect é o campo *Conexões para o host* em *Coleta de dados → Hosts → <algum host> → aba Criptografia* e o campo *Conexões para o proxy* em *Administração → Proxies → <algum proxy> → aba Criptografia*. Se o tipo de criptografia configurado para a conexão falhar, nenhum outro tipo de criptografia será tentado.

`TLSAccept` é usado nos arquivos de configuração do proxy Zabbix (no modo passivo, especifica apenas conexões do server) e do agent Zabbix (para verificações passivas). No frontend do Zabbix, o equivalente ao TLSAccept é o campo *Conexões do host*

em *Coleta de dados → Hosts → <algum host> → aba Criptografia* e o campo *Conexões do proxy* em *Administração → Proxies → <algum proxy> → aba Criptografia*.

Normalmente, você configura apenas um tipo de criptografia para conexões de entrada. Mas você pode querer alternar o tipo de criptografia, por exemplo, de sem criptografia para baseada em certificado, com o mínimo de tempo de inatividade e possibilidade de rollback. Para isso:

- Defina `TLSAccept=unencrypted,cert` no arquivo de configuração do agent e reinicie o agent Zabbix
- Teste a conexão com o zabbix_get para o agent usando certificado. Se funcionar, você pode reconfigurar a criptografia para esse agent no frontend do Zabbix na aba *Coleta de dados → Hosts → <algum host> → Criptografia* definindo *Conexões para o host* como "Certificado".
- Quando o cache de configuração do server for atualizado (e a configuração do proxy for atualizada se o host for monitorado por proxy), as conexões para esse agent serão criptografadas
- Se tudo funcionar como esperado, você pode definir `TLSAccept=cert` no arquivo de configuração do agent e reiniciar o agent Zabbix. Agora o agent aceitará apenas conexões criptografadas baseadas em certificado. Conexões sem criptografia e baseadas em PSK serão rejeitadas.

Da mesma forma, funciona no server e no proxy. Se no frontend do Zabbix, na configuração do host, *Conexões do host* estiver definido como "Certificado", então apenas conexões criptografadas baseadas em certificado serão aceitas do agent (verificações ativas) e do zabbix_sender (itens trapper).

Muito provavelmente, você configurará as conexões de entrada e saída para usar o mesmo tipo de criptografia ou nenhuma criptografia. Mas tecnicamente é possível configurar de forma assimétrica, por exemplo, criptografia baseada em certificado para entrada e baseada em PSK para saída.

A configuração de criptografia para cada host é exibida no frontend do Zabbix, em *Coleta de dados → Hosts* na coluna *Criptografia do agent*. Por exemplo:

Exemplo	Conexões para o host	Conexões permitidas do host	Conexões rejeitadas do host
NONE	Sem criptografia	Sem criptografia	Criptografadas, baseadas em certificado e baseadas em PSK
CERT NONE PSK CERT	Criptografada, baseada em certificado	Criptografada, baseada em certificado	Sem criptografia e criptografada baseada em PSK
PSK NONE PSK CERT	Criptografada, baseada em PSK	Criptografada, baseada em PSK	Sem criptografia e criptografada baseada em certificado
PSK NONE PSK CERT	Criptografada, baseada em PSK	Sem criptografia e criptografada baseada em PSK	Criptografada baseada em certificado
CERT NONE PSK CERT	Criptografada, baseada em certificado	Sem criptografia, baseada em PSK ou baseada em certificado	-

Attention:
As conexões não são criptografadas por padrão. A criptografia deve ser configurada para cada host e proxy individualmente.

zabbix_get e zabbix_sender com criptografia Veja zabbix_get e zabbix_sender manpages para usá-los com criptografia.

Ciphersuites As ciphersuites por padrão são configuradas internamente durante a inicialização do Zabbix.

Também são suportadas ciphersuites configuradas pelo usuário para GnuTLS e OpenSSL. Os usuários podem configurar ciphersuites de acordo com suas políticas de segurança. O uso deste recurso é opcional (as ciphersuites padrão internas ainda funcionam).

Para bibliotecas criptográficas compiladas com configurações padrão, as regras internas do Zabbix normalmente resultam nas seguintes ciphersuites (em ordem da maior para a menor prioridade):

Biblioteca	Ciphersuites de certificado	Ciphersuites PSK
GnuTLS 3.1.18	TLS_ECDHE_RSA_AES_128_GCM_SHA256 TLS_ECDHE_RSA_AES_128_CBC_SHA256 TLS_ECDHE_RSA_AES_128_CBC_SHA1 TLS_RSA_AES_128_GCM_SHA256 TLS_RSA_AES_128_CBC_SHA256 TLS_RSA_AES_128_CBC_SHA1	TLS_ECDHE_PSK_AES_128_CBC_SHA256 TLS_ECDHE_PSK_AES_128_CBC_SHA1 TLS_PSK_AES_128_GCM_SHA256 TLS_PSK_AES_128_CBC_SHA256 TLS_PSK_AES_128_CBC_SHA1

Biblioteca	Ciphersuites de certificado	Ciphersuites PSK
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

Ciphersuites configuradas pelo usuário Os critérios internos de seleção de ciphersuite podem ser substituídos por ciphersuites configuradas pelo usuário.

> **Attention:**
> Ciphersuites configuradas pelo usuário são um recurso destinado a usuários avançados que entendem ciphersuites TLS, sua segurança e as consequências de erros, e que estão confortáveis com a solução de problemas de TLS.

Os critérios internos de seleção de ciphersuite podem ser substituídos usando os seguintes parâmetros:

Escopo de substituição	Parâmetro	Valor	Descrição
Seleção de ciphersuite para certificados	TLSCipherCert13	Cipher strings válidas do OpenSSL 1.1.1 para o protocolo TLS 1.3 (seus valores são passados para a função OpenSSL `SSL_CTX_set_ciphersuites()`).	Critérios de seleção de ciphersuite baseados em certificado para TLS 1.3 Apenas OpenSSL 1.1.1 ou mais recente.
	TLSCipherCert	Cipher strings válidas do OpenSSL para TLS 1.2 ou priority strings válidas do GnuTLS. Seus valores são passados para as funções `SSL_CTX_set_cipher_list()` ou `gnutls_priority_init()`, respectivamente.	Critérios de seleção de ciphersuite baseados em certificado para TLS 1.2/1.3 (GnuTLS), TLS 1.2 (OpenSSL)
Seleção de ciphersuite para PSK	TLSCipherPSK13	Cipher strings válidas do OpenSSL 1.1.1 para o protocolo TLS 1.3 (seus valores são passados para a função OpenSSL `SSL_CTX_set_ciphersuites()`).	Critérios de seleção de ciphersuite baseados em PSK para TLS 1.3 Apenas OpenSSL 1.1.1 ou mais recente.
	TLSCipherPSK	Cipher strings válidas do OpenSSL para TLS 1.2 ou priority strings válidas do GnuTLS. Seus valores são passados para as funções `SSL_CTX_set_cipher_list()` ou `gnutls_priority_init()`, respectivamente.	Critérios de seleção de ciphersuite baseados em PSK para TLS 1.2/1.3 (GnuTLS), TLS 1.2 (OpenSSL)

Escopo de substituição	Parâmetro	Valor	Descrição
Lista combinada de ciphersuite para certificado e PSK	TLSCipherAll13	Cipher strings válidas do OpenSSL 1.1.1 para o protocolo TLS 1.3 (seus valores são passados para a função OpenSSL `SSL_CTX_set_ciphersuites()`).	Critérios de seleção de ciphersuite para TLS 1.3 Apenas OpenSSL 1.1.1 ou mais recente.
	TLSCipherAll	Cipher strings válidas do OpenSSL para TLS 1.2 ou priority strings válidas do GnuTLS. Seus valores são passados para as funções `SSL_CTX_set_cipher_list()` ou `gnutls_priority_init()`, respectivamente.	Critérios de seleção de ciphersuite para TLS 1.2/1.3 (GnuTLS), TLS 1.2 (OpenSSL)

Para substituir a seleção de ciphersuite nas utilitários zabbix_get e zabbix_sender - use os parâmetros de linha de comando:

- `--tls-cipher13`
- `--tls-cipher`

Os novos parâmetros são opcionais. Se um parâmetro não for especificado, o valor padrão interno será usado. Se um parâmetro for definido, ele não pode estar vazio.

Se a configuração de um valor TLSCipher* na biblioteca de criptografia falhar, o server, proxy ou agent não será iniciado e um erro será registrado.

É importante entender quando cada parâmetro é aplicável.

Conexões de saída

O caso mais simples são as conexões de saída:

- Para conexões de saída com certificado - use TLSCipherCert13 ou TLSCipherCert
- Para conexões de saída com PSK - use TLSCipherPSK13 ou TLSCipherPSK
- No caso das utilitários zabbix_get e zabbix_sender, os parâmetros de linha de comando `--tls-cipher13` ou `--tls-cipher` podem ser usados (a criptografia é especificada de forma inequívoca com o parâmetro `--tls-connect`)

Conexões de entrada

É um pouco mais complicado com conexões de entrada porque as regras são específicas para componentes e configuração.

Para o **agent** do Zabbix:

Configuração de conexão do agent	Configuração de cifra
TLSConnect=cert	TLSCipherCert, TLSCipherCert13
TLSConnect=psk	TLSCipherPSK, TLSCipherPSK13
TLSAccept=cert	TLSCipherCert, TLSCipherCert13
TLSAccept=psk	TLSCipherPSK, TLSCipherPSK13
TLSAccept=cert,psk	TLSCipherAll, TLSCipherAll13

Para o **server** e **proxy** do Zabbix:

Configuração de conexão	Configuração de cifra
Conexões de saída usando PSK	TLSCipherPSK, TLSCipherPSK13
Conexões de entrada usando certificados	TLSCipherAll, TLSCipherAll13
Conexões de entrada usando PSK se o server não tiver certificado	TLSCipherPSK, TLSCipherPSK13
Conexões de entrada usando PSK se o server tiver certificado	TLSCipherAll, TLSCipherAll13

Algum padrão pode ser visto nas duas tabelas acima:

- TLSCipherAll e TLSCipherAll13 só podem ser especificados se uma lista combinada de cifras baseada em certificado **e** PSK for usada. Existem dois casos em que isso ocorre: server (proxy) com um certificado configurado (cifras PSK são sempre configuradas no server, proxy se a biblioteca de criptografia suportar PSK), agent configurado para aceitar conexões de entrada baseadas em certificado e PSK
- nos outros casos, TLSCipherCert* e/ou TLSCipherPSK* são suficientes

As tabelas a seguir mostram os valores padrão internos do `TLSCipher*`. Eles podem ser um bom ponto de partida para seus próprios valores personalizados.

Parâmetro	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

Parâmetro	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] Os valores padrão são diferentes para versões mais antigas do OpenSSL (1.0.1, 1.0.2, 1.1.0), para LibreSSL e se o OpenSSL for compilado sem suporte a PSK.

Exemplos de cifras configuradas pelo usuário

Veja abaixo os seguintes exemplos de cifras configuradas pelo usuário:

- Testando strings de cifra e permitindo apenas cifras PFS
- Trocando de AES128 para AES256

Testando strings de cifra e permitindo apenas ciphersuites PFS

Para ver quais ciphersuites foram selecionadas, você precisa definir 'DebugLevel=4' no arquivo de configuração ou usar a opção `-vv` para o zabbix_sender.

Pode ser necessário experimentar com os parâmetros `TLSCipher*` antes de obter as ciphersuites desejadas. É inconveniente reiniciar o Zabbix server, proxy ou agent várias vezes apenas para ajustar os parâmetros `TLSCipher*`. Opções mais convenientes são usar o zabbix_sender ou o comando `openssl`. Vamos mostrar ambos.

1. Usando zabbix_sender.

Vamos criar um arquivo de configuração de teste, por exemplo `/home/zabbix/test.conf`, com a sintaxe de um arquivo `zabbix_agentd.conf`:

```
Hostname=nonexisting
ServerActive=nonexisting

TLSConnect=cert
TLSCAFile=/home/zabbix/ca.crt
TLSCertFile=/home/zabbix/agent.crt
TLSKeyFile=/home/zabbix/agent.key
TLSPSKIdentity=nonexisting
TLSPSKFile=/home/zabbix/agent.psk
```

Você precisa de certificados CA e agent válidos e PSK para este exemplo. Ajuste os caminhos e nomes dos arquivos de certificado e PSK para o seu ambiente.

Se você não estiver usando certificados, mas apenas PSK, pode criar um arquivo de teste mais simples:

```
Hostname=nonexisting
ServerActive=nonexisting
```

```
TLSConnect=psk
TLSPSKIdentity=nonexisting
TLSPSKFile=/home/zabbix/agentd.psk
```

As ciphersuites selecionadas podem ser vistas executando o zabbix_sender (exemplo compilado com 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
```

Aqui você vê as ciphersuites selecionadas por padrão. Esses valores padrão são escolhidos para garantir a interoperabilidade com agents Zabbix executando em sistemas com versões mais antigas do OpenSSL (a partir de 1.0.1).

Com sistemas mais recentes, você pode optar por aumentar a segurança permitindo apenas algumas ciphersuites, por exemplo, apenas ciphersuites com PFS (Perfect Forward Secrecy). Vamos tentar permitir apenas ciphersuites com PFS usando os parâmetros `TLSCipher*`.

Attention:
O resultado não será interoperável com sistemas usando OpenSSL 1.0.1 e 1.0.2, se PSK for usado. A criptografia baseada em certificado deve funcionar.

Adicione duas linhas ao arquivo de configuração `test.conf`:

```
TLSCipherCert=EECDH+aRSA+AES128
TLSCipherPSK=kECDHEPSK+AES128
```

e teste novamente:

```
$ 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
```

As listas "certificate ciphersuites" e "PSK ciphersuites" mudaram - elas estão mais curtas do que antes, contendo apenas ciphersuites TLS 1.3 e ciphersuites TLS 1.2 ECDHE-*, como esperado.

2. TLSCipherAll e TLSCipherAll13 não podem ser testados com zabbix_sender; eles não afetam o valor "certificate and PSK ciphersuites" mostrado no exemplo acima. Para ajustar TLSCipherAll e TLSCipherAll13, você precisa experimentar com o agent, proxy ou server.

Portanto, para permitir apenas ciphersuites PFS, pode ser necessário adicionar até três parâmetros

```
TLSCipherCert=EECDH+aRSA+AES128
TLSCipherPSK=kECDHEPSK+AES128
TLSCipherAll=EECDH+aRSA+AES128:kECDHEPSK+AES128
```

ao zabbix_agentd.conf, zabbix_proxy.conf e zabbix_server_conf se cada um deles tiver um certificado configurado e o agent também tiver PSK.

Se o seu ambiente Zabbix usar apenas criptografia baseada em PSK e não certificados, então apenas um:

```
TLSCipherPSK=kECDHEPSK+AES128
```

Agora que você entende como funciona, pode testar a seleção de ciphersuite até mesmo fora do Zabbix, com o comando `openssl`. Vamos testar todos os três valores de parâmetro `TLSCipher*`:

```
$ openssl ciphers EECDH+aRSA+AES128 | sed 's/:/ /g'
TLS_AES_256_GCM_SHA384 TLS_CHACHA20_POLY1305_SHA256 TLS_AES_128_GCM_SHA256 ECDHE-RSA-AES128-GCM-SHA256 E
$ openssl ciphers kECDHEPSK+AES128 | sed 's/:/ /g'
TLS_AES_256_GCM_SHA384 TLS_CHACHA20_POLY1305_SHA256 TLS_AES_128_GCM_SHA256 ECDHE-PSK-AES128-CBC-SHA256 E
$ openssl ciphers EECDH+aRSA+AES128:kECDHEPSK+AES128 | sed 's/:/ /g'
TLS_AES_256_GCM_SHA384 TLS_CHACHA20_POLY1305_SHA256 TLS_AES_128_GCM_SHA256 ECDHE-RSA-AES128-GCM-SHA256 E
```

Você pode preferir `openssl ciphers` com a opção `-V` para uma saída mais detalhada:

```
$ 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
```

Da mesma forma, você pode testar as strings de prioridade para GnuTLS:

```
$ gnutls-cli -l --priority=NONE:+VERS-TLS1.2:+ECDHE-RSA:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+CURVE-A
Cipher suites for NONE:+VERS-TLS1.2:+ECDHE-RSA:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+CURVE-ALL:+COMP-
TLS_ECDHE_RSA_AES_128_GCM_SHA256                        0xc0, 0x2f      TLS1.2
TLS_ECDHE_RSA_AES_128_CBC_SHA256                        0xc0, 0x27      TLS1.2

Protocols: VERS-TLS1.2
Ciphers: AES-128-GCM, AES-128-CBC
MACs: AEAD, SHA256
Key Exchange Algorithms: ECDHE-RSA
Groups: GROUP-SECP256R1, GROUP-SECP384R1, GROUP-SECP521R1, GROUP-X25519, GROUP-X448, GROUP-FFDHE2048, GR
PK-signatures: SIGN-RSA-SHA256, SIGN-RSA-PSS-SHA256, SIGN-RSA-PSS-RSAE-SHA256, SIGN-ECDSA-SHA256, SIGN-E
```

Mudando de AES128 para AES256

O Zabbix usa o AES128 como padrão interno para dados. Vamos supor que você está usando certificados e deseja mudar para AES256, no OpenSSL 1.1.1.

Isso pode ser feito adicionando os respectivos parâmetros em `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
TLSipherPSK13=TLS_CHACHA20_POLY1305_SHA256
TLSCipherPSK=kECDHEPSK+AES256:-SHA1
TLSipherAll13=TLS_AES_256_GCM_SHA384
TLSCipherAll=EECDH+aRSA+AES256:-SHA1:-SHA384
```

::: não importante Embora apenas conjuntos de criptografia relacionados a certificados será usado, os parâmetros `TLSCipherPSK*` também são definidos para evitar seus valores padrão que incluem cifras menos seguras para interoperabilidade. Ciphersuites PSK não podem ser completamente desabilitados em servidor/proxy. :::

E em `zabbix_agentd.conf`:

```
TLSConnect=cert
TLSAccept=cert
TLSCAFile=/home/zabbix/ca.crt
TLSCertFile=/home/zabbix/agent.crt
TLSKeyFile=/home/zabbix/agent.key
TLSCipherCert13=TLS_AES_256_GCM_SHA384
TLSCipherCert=EECDH+aRSA+AES256:-SHA1:-SHA384
```

1 Usando certificados

Visão geral

O Zabbix pode usar certificados RSA no formato PEM, assinados por uma autoridade certificadora (CA) pública ou interna.

A verificação do certificado é realizada em relação a um certificado CA pré-configurado. Opcionalmente, Listas de Revogação de Certificados (CRL) podem ser usadas.

Cada componente do Zabbix pode ter apenas um certificado configurado.

Para obter mais informações sobre como configurar e operar uma CA interna, gerar e assinar solicitações de certificado e revogar certificados, consulte tutoriais como o OpenSSL PKI Tutorial v2.0.

Considere e teste cuidadosamente suas extensões de certificado. Para mais detalhes, consulte Limitações no uso de extensões de certificado X.509 v3.

Parâmetros de configuração de certificado

Os seguintes parâmetros de configuração são suportados para configurar certificados nos componentes do Zabbix.

Parâmetro	Obrigatório	Descrição
TLSCAFile	sim	Caminho completo de um arquivo contendo os certificados da(s) CA(s) de nível superior para verificação do certificado do par. Se estiver usando uma cadeia de certificados com vários membros, ordene os certificados com os certificados da(s) CA(s) de nível inferior primeiro, seguidos pelos certificados da(s) CA(s) de nível superior. Certificados de várias CAs podem ser incluídos em um único arquivo.
TLSCRLFile	não	Caminho completo de um arquivo contendo as Listas de Revogação de Certificados (CRL).
TLSCertFile	sim	Caminho completo de um arquivo contendo o certificado. Se estiver usando uma cadeia de certificados com vários membros, ordene os certificados com o certificado do server, proxy ou agent primeiro, seguido pelos certificados da(s) CA(s) de nível inferior e concluído pelos certificados da(s) CA(s) de nível superior.
TLSKeyFile	sim	Caminho completo de um arquivo contendo a chave privada. Certifique-se de que este arquivo seja legível apenas pelo usuário do Zabbix definindo os direitos de acesso apropriados.
TLSServerCertIssuer	não	Emissor do certificado do server permitido.
TLSServerCertSubject	não	Assunto do certificado do server permitido.

Exemplos de configuração

Após configurar os certificados necessários, configure os componentes do Zabbix para usar criptografia baseada em certificados.

Abaixo estão as etapas detalhadas para configurar:

- Zabbix server
- Zabbix proxy
- Zabbix agent

Zabbix server

1. Prepare o arquivo do certificado da CA.

Para verificar os certificados do par, o Zabbix server deve ter acesso ao arquivo que contém os certificados da CA raiz autoassinados de nível superior. Por exemplo, se forem necessários certificados de duas CAs raiz independentes, coloque-os em um arquivo em `/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. Coloque o certificado/cadeia de certificados do Zabbix server em um arquivo, por exemplo, em `/home/zabbix/zabbix_server.crt`. O primeiro certificado é o certificado do Zabbix server, seguido pelo certificado da CA intermediária:

```
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:
Use apenas os atributos mencionados acima para os certificados de cliente e servidor para evitar afetar o processo de verificação do certificado. Por exemplo, o OpenSSL pode falhar ao estabelecer uma conexão criptografada se as extensões *X509v3 Subject Alternative Name* ou *Netscape Cert Type* forem usadas. Para mais informações, consulte Limitações no uso de extensões de certificado X.509 v3.

3. Coloque a chave privada do Zabbix server em um arquivo, por exemplo, em `/home/zabbix/zabbix_server.key`:

```
-----BEGIN PRIVATE KEY-----
MIIEwAIBADANBgkqhkiG9w0BAQEFAASCBKowggSmAgEAAoIBAQC9tIXIJoVnNXDl
...
IJLkhbybBYEf47MLhffWa7XvZTY=
-----END PRIVATE KEY-----
```

4. Edite os parâmetros de configuração TLS no arquivo de configuração do Zabbix server:

```
TLSCAFile=/home/zabbix/zabbix_ca_file.crt
TLSCertFile=/home/zabbix/zabbix_server.crt
TLSKeyFile=/home/zabbix/zabbix_server.key
```

Zabbix proxy

1. Prepare os arquivos com os certificados da CA de nível superior, o certificado/cadeia de certificados do Zabbix proxy e a chave privada conforme descrito na seção Zabbix server. Em seguida, edite os parâmetros `TLSCAFile`, `TLSCertFile` e `TLSKeyFile` no arquivo de configuração do Zabbix proxy de acordo.

2. Edite os parâmetros TLS adicionais no arquivo de configuração do Zabbix proxy:

- Para proxy ativo: `TLSConnect=cert`
- Para proxy passivo: `TLSAccept=cert`

Note:
Para melhorar a segurança do proxy, você também pode definir os parâmetros `TLSServerCertIssuer` e `TLSServerCertSubject`. Para mais informações, consulte Restringindo emissor e assunto do certificado permitido.

Os parâmetros TLS no arquivo de configuração final do proxy podem ser semelhantes a:

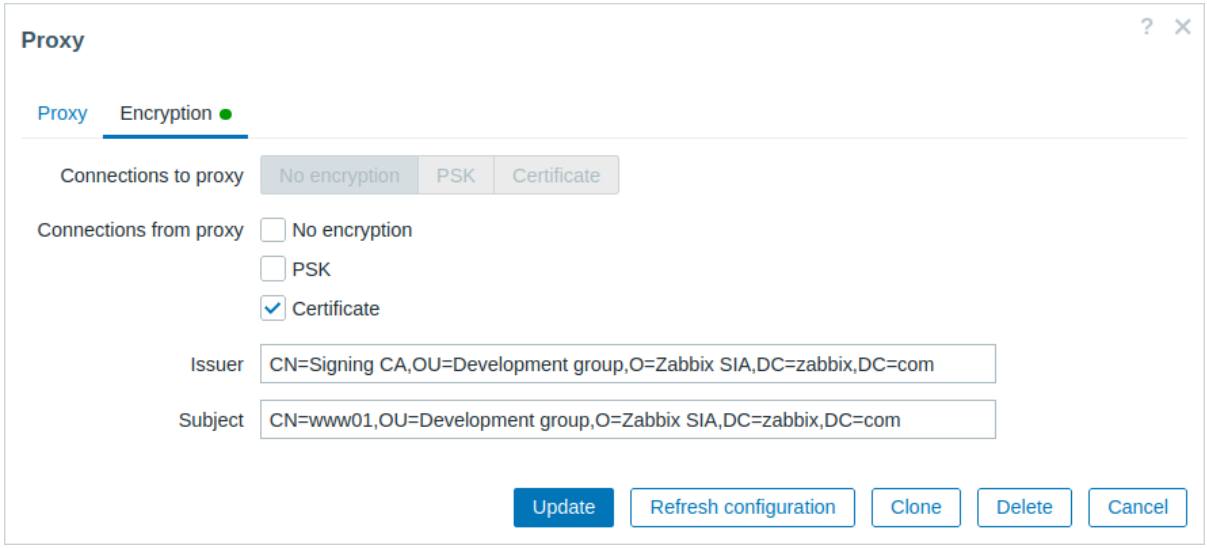
```
TLSConnect=cert
TLSAccept=cert
TLSCAFile=/home/zabbix/zabbix_ca_file.crt
TLSServerCertIssuer=CN=Signing CA,OU=Development group,O=Zabbix SIA,DC=zabbix,DC=com
TLSServerCertSubject=CN=Zabbix server,OU=Development group,O=Zabbix SIA,DC=zabbix,DC=com
TLSCertFile=/home/zabbix/zabbix_proxy.crt
TLSKeyFile=/home/zabbix/zabbix_proxy.key
```

3. Configure a criptografia para este proxy no frontend do Zabbix:

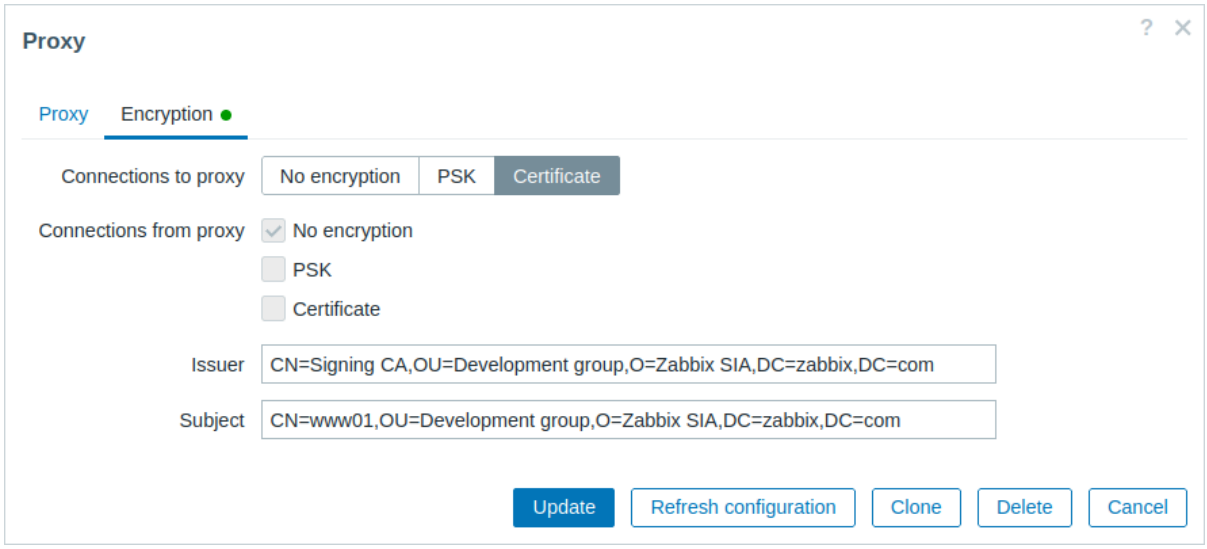
- Vá para: *Administração → Proxies*.
- Selecione o proxy e clique na guia *Criptografia*.

Nos exemplos abaixo, os campos *Emissor* e *Assunto* estão preenchidos. Para mais informações sobre por que e como usar esses campos, consulte Restringindo emissor e assunto do certificado permitido.

Para proxy ativo:



Para proxy passivo:



Zabbix agent

1. Prepare os arquivos com os certificados CA de nível superior, o certificado/cadeia de certificados do Zabbix agent e a chave privada conforme descrito na seção Zabbix server. Em seguida, edite os parâmetros `TLSCAFile`, `TLSCertFile` e `TLSKeyFile` no arquivo de configuração do Zabbix agent de acordo.

2. Edite os parâmetros TLS adicionais no arquivo de configuração do Zabbix agent:

- Para agent ativo: `TLSConnect=cert`
- Para agent passivo: `TLSAccept=cert`

Note:
Para melhorar a segurança do agent, você pode definir os parâmetros `TLSServerCertIssuer` e `TLSServerCertSubject`. Para mais informações, consulte Restringindo o emissor e o assunto do certificado permitido.

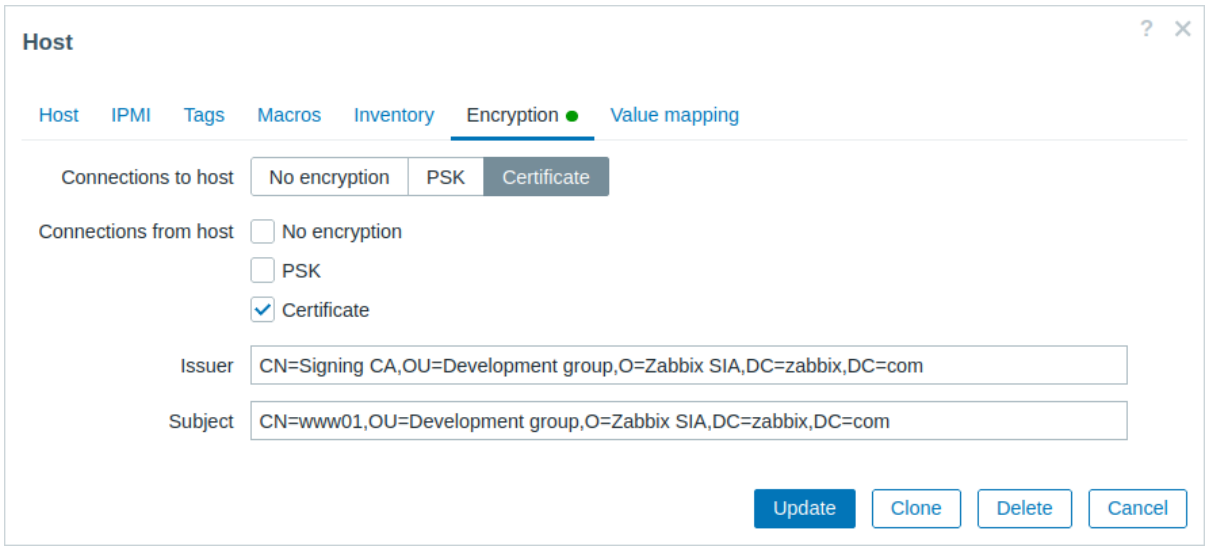
Os parâmetros TLS no arquivo de configuração final do agent podem ser semelhantes ao exemplo abaixo. Observe que o exemplo assume que o host é monitorado por um proxy, portanto, ele é especificado como o Subject do certificado:

```
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. Configure a criptografia no frontend do Zabbix para o host monitorado por este agent.

- Vá para: *Coleta de dados* → *Hosts*.
- Selecione o host e clique na guia *Criptografia*.

No exemplo abaixo, os campos *Emissor* e *Assunto* estão preenchidos. Para mais informações sobre por que e como usar esses campos, consulte Restringindo o emissor e o assunto do certificado permitido.



Zabbix web service

1. Prepare os arquivos com os certificados CA de nível superior, o certificado/cadeia de certificados do Zabbix web service e a chave privada, conforme descrito na seção Zabbix server. Em seguida, edite os parâmetros `TLSCAFile`, `TLSCertFile` e `TLSKeyFile` no arquivo de configuração do Zabbix web service de acordo.

2. Edite um parâmetro TLS adicional no arquivo de configuração do Zabbix web service: `TLSAccept=cert`

Os parâmetros TLS no arquivo de configuração final do web service podem ser semelhantes a:

```
TLSAccept=cert
TLSCAFile=/home/zabbix/zabbix_ca_file.crt
TLSCertFile=/home/zabbix/zabbix_web_service.crt
TLSKeyFile=/home/zabbix/zabbix_web_service.key
```

3. Configure o Zabbix server para se conectar ao Zabbix web service configurado com TLS, editando o parâmetro `WebServiceURL` no arquivo de configuração do Zabbix server:

```
WebServiceURL=https://example.com:443/report
```

Restringindo o emissor e o assunto do certificado permitido

Quando dois componentes do Zabbix (por exemplo, server e agent) estabelecem uma conexão TLS, eles validam os certificados um do outro. Se um certificado do par for assinado por uma CA confiável (com um certificado de nível superior pré-configurado em `TLSCAFile`), for válido, não tiver expirado e passar em outras verificações, a comunicação entre os componentes pode prosseguir. Nesse caso mais simples, o emissor e o assunto do certificado não são verificados.

No entanto, isso apresenta um risco: qualquer pessoa com um certificado válido pode se passar por outra pessoa (por exemplo, um certificado de host pode ser usado para se passar por um server). Embora isso possa ser aceitável em ambientes pequenos, onde os certificados são assinados por uma CA interna dedicada e o risco de falsificação é baixo, pode não ser suficiente em ambientes maiores ou mais sensíveis à segurança.

Se sua CA de nível superior emitir certificados que não devem ser aceitos pelo Zabbix ou se você quiser reduzir o risco de falsificação, poderá restringir os certificados permitidos especificando seu emissor e assunto.

Por exemplo, no arquivo de configuração do proxy do Zabbix, você pode especificar:

```
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
```

Com essas configurações, um proxy ativo não se comunicará com um server Zabbix cujo certificado tenha um emissor ou assunto diferente. Da mesma forma, um proxy passivo não aceitará solicitações de tal server.

Regras para correspondência de strings `Issuer` e `Subject`

As regras para correspondência de strings `Issuer` e `Subject` são as seguintes:

- As strings `Issuer` e `Subject` são verificadas independentemente. Ambas são opcionais.

- Uma string não especificada significa que qualquer string é aceita.
- As strings são comparadas *como estão* e devem corresponder exatamente.
- Caracteres UTF-8 são suportados. No entanto, curingas (*) ou expressões regulares não são suportados.
- Os seguintes requisitos da RFC 4514 são implementados - caracteres que exigem escape (com uma barra invertida '\', U+005C):
  - em qualquer lugar da string: '"' (U+0022), '+' (U+002B), ',' (U+002C), ';' (U+003B), '<' (U+003C), '>' (U+003E), '\\' (U+005C);
  - no início da string: espaço (' ', U+0020) ou cerquilha ('#', U+0023);
  - no final da string: espaço (' ', U+0020).
- Caracteres nulos (U+0000) não são suportados. Se um caractere nulo for encontrado, a correspondência falhará.
- Os padrões RFC 4517 e RFC 4518 não são suportados.

Por exemplo, se as strings de organização (O) de `Issuer` e `Subject` contiverem espaços à direita e a string de unidade organizacional (OU) de `Subject` contiver aspas duplas, esses caracteres devem ser escapados:

```
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
```

Ordem e formatação dos campos

O Zabbix segue as recomendações da RFC 4514, que especifica uma ordem "reversa" para esses campos, começando pelos campos de nível mais baixo (CN), passando pelos campos de nível intermediário (OU, O) e concluindo com os campos de nível mais alto (DC).

```
TLSServerCertIssuer=CN=Signing CA,OU=Development group,O=Zabbix SIA,DC=zabbix,DC=com
TLSServerCertSubject=CN=Zabbix proxy,OU=Development group,O=Zabbix SIA,DC=zabbix,DC=com
```

Por outro lado, o OpenSSL exibe por padrão as strings `Issuer` e `Subject` na ordem do nível mais alto para o mais baixo. No exemplo a seguir, os campos `Issuer` e `Subject` começam com o campo de nível mais alto (DC) e terminam com o campo de nível mais baixo (CN). A formatação com espaços e separadores de campos também varia de acordo com as opções utilizadas e, portanto, não corresponderá ao formato exigido pelo Zabbix.

```
$ openssl x509 -noout -in /home/zabbix/zabbix_proxy.crt -issuer -subject
issuer= /DC=com/DC=zabbix/O=Zabbix SIA/OU=Development group/CN=Signing CA
subject= /DC=com/DC=zabbix/O=Zabbix SIA/OU=Development group/CN=Zabbix proxy

$ openssl x509 -noout -text -in /home/zabbix/zabbix_proxy.crt
Certificate:
    ...
        Issuer: DC=com, DC=zabbix, O=Zabbix SIA, OU=Development group, CN=Signing CA
        ...
        Subject: DC=com, DC=zabbix, O=Zabbix SIA, OU=Development group, CN=Zabbix proxy
```

Para formatar corretamente as strings *Issuer* e *Subject* para o Zabbix, invoque o OpenSSL com as seguintes opções:

```
$ 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
```

A saída estará então em ordem reversa, separada por vírgulas e utilizável nos arquivos de configuração do Zabbix e no frontend:

```
issuer=CN=Signing CA,OU=Development group,O=Zabbix SIA,DC=zabbix,DC=com
subject=CN=Zabbix proxy,OU=Development group,O=Zabbix SIA,DC=zabbix,DC=com
```

Limitações no uso de extensões de certificado X.509 v3

Ao implementar certificados X.509 v3 no Zabbix, certas extensões podem não ser totalmente suportadas ou podem resultar em comportamento inconsistente.

Extensão Subject Alternative Name

O Zabbix não suporta a extensão *Subject Alternative Name*, que é usada para especificar nomes DNS alternativos, como endereços IP ou endereços de e-mail. O Zabbix só pode validar o valor no campo *Subject* do certificado (consulte Restringindo o emissor e o assunto do certificado permitido). Se os certificados incluírem o campo `subjectAltName`, o resultado da validação do certificado pode variar dependendo dos toolkits de criptografia específicos usados para compilar os componentes do Zabbix. Como resultado, o Zabbix pode aceitar ou rejeitar certificados com base nessas combinações.

Extensão Extended Key Usage

O Zabbix suporta a extensão *Extended Key Usage*. No entanto, se usada, geralmente é necessário que ambos os atributos

clientAuth (para autenticação de cliente TLS WWW) e *serverAuth* (para autenticação de servidor TLS WWW) sejam especificados. Por exemplo:

- Em verificações passivas, onde o agent Zabbix opera como um servidor TLS, o atributo *serverAuth* deve ser incluído no certificado do agent.
- Para verificações ativas, onde o agent opera como um cliente TLS, o atributo *clientAuth* deve ser incluído no certificado do agent.

Embora o GnuTLS possa emitir um aviso para violações de uso de chave, normalmente permite que a comunicação prossiga apesar desses avisos.

Extensão Name Constraints

O suporte para a extensão *Name Constraints* varia entre os toolkits de criptografia. Certifique-se de que o toolkit escolhido suporte essa extensão. Essa extensão pode restringir o Zabbix de carregar certificados CA se esta seção for marcada como crítica, dependendo do toolkit específico em uso.

Listas de Revogação de Certificados (CRL)

Se um certificado for comprometido, a Autoridade Certificadora (CA) pode revogá-lo incluindo o certificado em uma Lista de Revogação de Certificados (CRL). As CRLs são gerenciadas por meio de arquivos de configuração e podem ser especificadas usando o parâmetro `TLSCRLFile` nos arquivos de configuração do server, proxy e agent. Por exemplo:

```
TLSCRLFile=/home/zabbix/zabbix_crl_file.crt
```

Neste caso, `zabbix_crl_file.crt` pode conter CRLs de várias CAs e pode ser semelhante a isto:

```
-----BEGIN X509 CRL-----
MIIB/DCB5QIBATANBgkqhkiG9w0BAQUFADCBgTETMBEGCgmSJomT8ixkARkWA2Nv
...
treZeUPjb7LSmZ3K2hpbZN7SoOZcAoHQ3GWd9npuctg=
-----END X509 CRL-----
-----BEGIN X509 CRL-----
MIIB+TCB4gIBATANBgkqhkiG9w0BAQUFADB/MRMwEQYKCZImiZPyLGQBGRYDY29t
...
CAEebS2CND3ShBedZ8YSil59O6JvaDP61lR5lNs=
-----END X509 CRL-----
```

O arquivo CRL é carregado apenas quando o Zabbix é iniciado. Para atualizar a CRL, reinicie o Zabbix.

> **Attention:**
> Se os componentes do Zabbix forem compilados com OpenSSL e as CRLs forem usadas, certifique-se de que cada CA de nível superior e intermediária nas cadeias de certificados tenha uma CRL correspondente (mesmo que esteja vazia) incluída no `TLSCRLFile`.

2 Usando chaves pré-compartilhadas

Visão geral

Cada chave pré-compartilhada (PSK) no Zabbix é na verdade um par de:

- string de identidade PSK não secreta,
- valor da string PSK secreta.

A string de identidade PSK é uma string UTF-8 não vazia. Por exemplo, "ID PSK 001 Zabbix agentd". É um nome único pelo qual este PSK específico é referido pelos componentes Zabbix. Não coloque informações confidenciais em Cadeia de identidade PSK - é transmitida pela rede sem criptografia.

O valor PSK é uma sequência de dígitos hexadecimais difícil de adivinhar, por exemplo, "e560cb0d918d26d31b4f642181f5f570ad89a3909311

Limites de tamanho

Existem limites de tamanho para a identidade e valor do PSK no Zabbix, em alguns casos uma biblioteca de criptografia pode ter um limite inferior:

Componente	Tamanho máximo da identidade PSK	Tamanho mínimo do valor PSK	Tamanho máximo do valor PSK
Zabbix	128 caracteres UTF-8	128 bits (16 bytes de PSK, inseridos como 32 dígitos hexadecimais)	2048 bits (256 bytes de PSK, inseridos como 512 dígitos hexadecimais)
GnuTLS	128 bytes (pode incluir caracteres UTF-8)	-	2048 bits (256 bytes de PSK, inseridos como 512 dígitos hexadecimais)
OpenSSL 1.0.x, 1.1.0	127 bytes (pode incluir caracteres UTF-8)	-	2048 bits (256 bytes de PSK, inseridos como 512 dígitos hexadecimais)
OpenSSL 1.1.1	127 bytes (pode incluir caracteres UTF-8)	-	512 bits (64 bytes de PSK, inseridos como 128 dígitos hexadecimais)
OpenSSL 1.1.1a e posteriores	127 bytes (pode incluir caracteres UTF-8)	-	2048 bits (256 bytes de PSK, inseridos como 512 dígitos hexadecimais)

Attention:
O frontend do Zabbix permite configurar uma string de identidade PSK com até 128 caracteres e um PSK de até 2048 bits, independentemente das bibliotecas de criptografia utilizadas.
 Se alguns componentes do Zabbix suportarem limites inferiores, é responsabilidade do usuário configurar a identidade e o valor do PSK com o comprimento permitido para esses componentes.
 Exceder os limites de comprimento resulta em falhas de comunicação entre os componentes do Zabbix.

Antes que o servidor Zabbix se conecte ao agent usando PSK, o servidor procura a identidade e o valor do PSK configurados para esse agent no banco de dados (na verdade, no cache de configuração). Ao receber uma conexão, o agent usa a identidade e o valor do PSK de seu arquivo de configuração. Se ambas as partes tiverem a mesma string de identidade e valor do PSK, a conexão poderá ser bem-sucedida.

Attention:
Cada identidade PSK deve ser emparelhada com apenas um valor. É responsabilidade do usuário garantir que não existam dois PSKs com a mesma string de identidade, mas valores diferentes. Não fazer isso pode levar a erros imprevisíveis ou interrupções na comunicação entre os componentes do Zabbix que usam PSKs com essa string de identidade PSK.

Gerando PSK

Por exemplo, um PSK de 256 bits (32 bytes) pode ser gerado usando o seguintes comandos:

- com *OpenSSL*:

```
$ openssl rand -hex 32
af8ced32dfe8714e548694e2d29e1a14ba6fa13f216cb35c19d0feb1084b0429
```

- com *GnuTLS*:

```
$ psktool -u psk_identity -p database.psk -s 32
Gerando uma chave aleatória para o usuário 'psk_identity'
Chave armazenada em database.psk

$ cat database.psk
psk_identity:9b8eafedfaae00cece62e85d5f4792c7d9c9bcc851b23216a1d300311cc4f7cb
```

Observe que "psktool" acima gera um arquivo de banco de dados com uma identidade PSK e seu PSK associado. O Zabbix espera apenas um PSK no arquivo PSK, então a string de identidade e os dois pontos (':') devem ser removidos do arquivo.

Configurando PSK para comunicação server-agent (exemplo)

No host do agent, escreva o valor do PSK em um arquivo, por exemplo, `/home/zabbix/zabbix_agentd.psk`. O arquivo deve conter o PSK na primeira linha de texto, por exemplo:

```
1f87b595725ac58dd977beef14b97461a7c1045b9a1c963065002c5473194952
```

Defina as permissões de acesso ao arquivo PSK - ele deve ser legível apenas pelo usuário Zabbix.

Edite os parâmetros TLS no arquivo de configuração do agent `zabbix_agentd.conf`, por exemplo, defina:

```
TLSConnect=psk
TLSAccept=psk
TLSPSKFile=/home/zabbix/zabbix_agentd.psk
TLSPSKIdentity=PSK 001
```

O agent irá se conectar ao server (cheques ativos) e aceitará conexões do server e do `zabbix_get` apenas usando PSK. A identidade do PSK será "PSK 001".

Reinicie o agent. Agora você pode testar a conexão usando o `zabbix_get`, por exemplo:

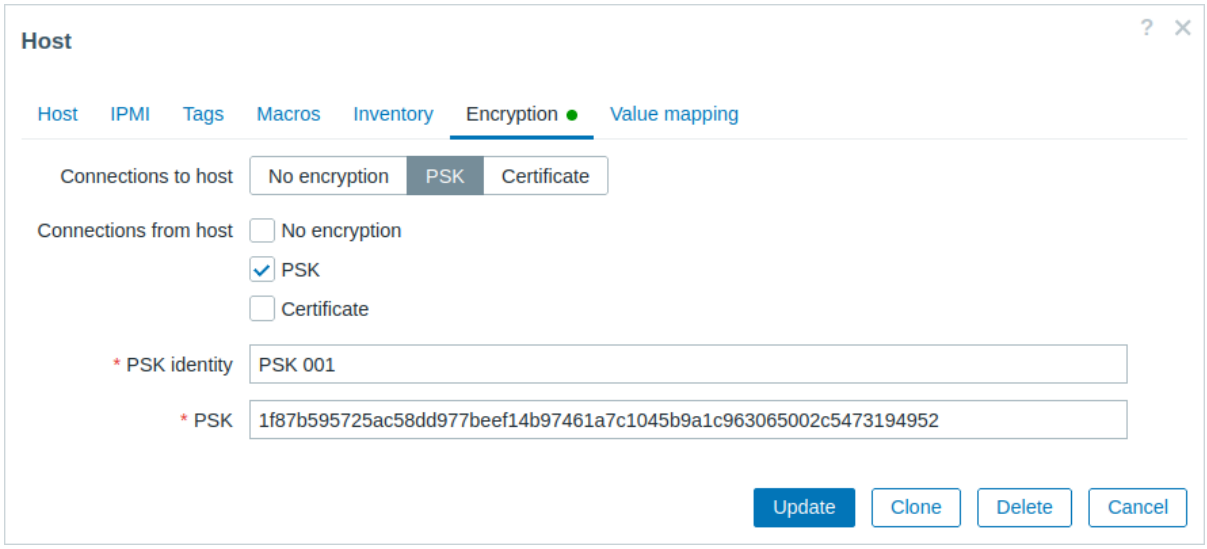
```
zabbix_get -s 127.0.0.1 -k "system.cpu.load[all,avg1]" --tls-connect=psk --tls-psk-identity="PSK 001" --tl
```

(Para minimizar o tempo de inatividade, veja como alterar o tipo de conexão em Gerenciamento de criptografia de conexão).

Configure a criptografia PSK para este agent no frontend do Zabbix:

- Vá para: *Coleta de dados → Hosts*
- Selecione o host e clique na aba **Criptografia**

Exemplo:



Todos os campos obrigatórios estão marcados com um asterisco vermelho.

Quando o cache de configuração for sincronizado com o banco de dados, as novas conexões usarão PSK. Verifique os arquivos de log do server e do agent para mensagens de erro.

Configurando PSK para servidor - comunicação proxy ativa (exemplo)

No proxy, grave o valor PSK em um arquivo, por exemplo, `/home/zabbix/zabbix_proxy.psk`. O arquivo deve conter PSK no primeiro cadeia de texto, por exemplo:

```
e560cb0d918d26d31b4f642181f5f570ad89a390931102e5391d08327ba434e9
```

Defina os direitos de acesso ao arquivo PSK - ele deve ser legível apenas pelo usuário do Zabbix.

Edite os parâmetros TLS no arquivo de configuração de proxy `zabbix_proxy.conf`, para exemplo, defina:

```
TLSConnect=psk
TLSPSKFile=/home/zabbix/zabbix_proxy.psk
TLSPSKIdentity=PSK 002
```

O proxy se conectará ao servidor usando o PSK. A identidade PSK será "PSK 002".

(Para minimizar o tempo de inatividade, veja como alterar o tipo de conexão em Connection criptografia gerenciamento).

Configure o PSK para este proxy no frontend do Zabbix. Vamos para *Administração→Proxies*, selecione o proxy, vá para a aba "Criptografia". Dentro "Conexões do proxy" marcam PSK. Cole no campo "Identidade PSK" "PSK 002" e "e560cb0d918d26d31b4f642181f5f570ad8
em campo "PSK". Clique em "Atualizar".

Reinicie o proxy. Ele começará a usar conexões criptografadas baseadas em PSK para servidor. Verifique os arquivos de log do servidor e do proxy para mensagens de erro.

Para um proxy passivo, o procedimento é muito semelhante. A única diferença - defina `TLSAccept=psk` no arquivo de configuração do proxy e defina "Conexões para proxy" no frontend do Zabbix para PSK.

3 Solução de problemas

Recomendações gerais

- Comece entendendo qual componente atua como cliente TLS e qual atua como servidor TLS no caso do problema.
 O Zabbix server, proxies e agents, dependendo da interação entre eles, podem funcionar como servidores e clientes TLS.
 Por exemplo, o Zabbix server conectando-se ao agent para uma verificação passiva, atua como um cliente TLS. O agent está no papel de servidor TLS.
 O Zabbix agent, solicitando uma lista de verificações ativas ao proxy, atua como um cliente TLS. O proxy está no papel de servidor TLS.
 As utilitários `zabbix_get` e `zabbix_sender` sempre atuam como clientes TLS.
- O Zabbix usa autenticação mútua.
 Cada lado verifica seu par e pode recusar a conexão.
 Por exemplo, o Zabbix server conectando-se ao agent pode fechar a conexão imediatamente se o certificado do agent for inválido. E vice-versa - o Zabbix agent aceitando uma conexão do server pode fechar a conexão se o server não for confiável para o agent.
- Examine os arquivos de log em ambos os lados - no cliente TLS e no servidor TLS.
 O lado que recusa a conexão pode registrar um motivo preciso pelo qual ela foi recusada. O outro lado geralmente relata um erro mais geral (por exemplo, ”Connection closed by peer”, ”connection was non-properly terminated”).
- Às vezes, a criptografia mal configurada resulta em mensagens de erro confusas que não apontam para a causa real.
 Nas subseções abaixo, tentamos fornecer uma coleção (longe de ser exaustiva) de mensagens e possíveis causas que podem ajudar na solução de problemas.
 Observe que diferentes toolkits de criptografia (OpenSSL, GnuTLS) geralmente produzem mensagens de erro diferentes nas mesmas situações problemáticas.
 Às vezes, as mensagens de erro dependem até mesmo da combinação específica de toolkits de criptografia em ambos os lados.

1 Tipo de conexão ou problemas de permissão

O servidor está configurado para se conectar com o PSK ao agente, mas o agente aceita apenas conexões não criptografadas

No log do servidor ou proxy (com *GnuTLS* 3.3.16)

```
Falha ao obter valor do agente: zbx_tls_connect(): gnutls_handshake() falhou: \
    -110 A conexão TLS foi finalizada incorretamente.
```

No log do servidor ou proxy (com *OpenSSL* 1.0.2c)

```
Falha ao obter valor do agente: conexão TCP bem-sucedida, não é possível estabelecer TLS para [[127.0.0.1
    Conexão fechada por peer. Verifique os tipos de conexão permitidos e os direitos de acesso
```

Um lado se conecta com o certificado, mas o outro lado aceita apenas PSK ou vice-versa

Em qualquer log (com *GnuTLS*):

```
falhou ao aceitar uma conexão de entrada: de 127.0.0.1: zbx_tls_accept(): gnutls_handshake() falhou:\
    -21 Não foi possível negociar um conjunto de cifras suportado.
```

Em qualquer log (com *OpenSSL* 1.0.2c):

```
falhou ao aceitar uma conexão de entrada: de 127.0.0.1: o handshake TLS retornou o código de erro 1:\
    arquivo .\ssl\s3_srvr.c linha 1411: erro:1408A0C1:rotinas SSL:ssl3_get_client_hello:nenhuma cifra com
    TLS escreve alerta fatal "falha de handshake"
```

Tentando usar o remetente Zabbix compilado com suporte a TLS para enviar dados para o servidor/proxy Zabbix compilado sem TLS

No log do lado da conexão:

Linux:

```
...Em zbx_tls_init_child()
...Biblioteca OpenSSL (versão OpenSSL 1.1.1 11 de setembro de 2018) inicializada
...
...Em zbx_tls_connect(): psk_identity:"PSK teste remetente"
...Fim do erro zbx_tls_connect():FAIL: 'conexão fechada por peer'
...send value error: TCP bem-sucedido, não é possível estabelecer TLS para [[localhost]:10051]: conexão f
```

Janelas:

```
...Biblioteca OpenSSL (versão OpenSSL 1.1.1a 20 de novembro de 2018) inicializada
...
...Em zbx_tls_connect(): psk_identity:"PSK teste remetente"
```

```
...zbx_psk_client_cb() solicitou a identidade PSK "PSK teste remetente"
...Fim do erro zbx_tls_connect():FAIL: 'Erro de E/S SSL_connect(): [0x00000000] A operação foi concluída
...send value error: TCP bem-sucedido, não é possível estabelecer TLS para [[192.168.1.2]:10051]: SSL_con
```

No log do lado da aceitação:

```
... falhou ao aceitar uma conexão de entrada: de 127.0.0.1: suporte para TLS não foi compilado em
```

Um lado se conecta ao PSK, mas o outro lado usa LibreSSL ou foi compilado sem suporte de criptografia

LibreSSL não suporta PSK.

No log do lado da conexão:

```
...TCP bem-sucedido, não é possível estabelecer TLS para [[192.168.1.2]:10050]: SSL_connect() Erro de E/S
```

No log do lado da aceitação:

```
...falha ao aceitar uma conexão de entrada: de 192.168.1.2: suporte para PSK não foi compilado em
```

Na interface do Zabbix:

```
Falha ao obter valor do agente: TCP bem-sucedido, não é possível estabelecer TLS para [[192.168.1.2]:1005
```

Um lado se conecta ao PSK, mas o outro lado usa OpenSSL com suporte a PSK desativado

No log do lado da conexão:

```
...TCP bem-sucedido, não é possível estabelecer TLS para [[192.168.1.2]:10050]: SSL_connect() definir cód
```

No log do lado da aceitação:

```
...falha ao aceitar uma conexão de entrada: de 192.168.1.2: código de resultado do conjunto de handshake
```

2 Problemas de certificado

OpenSSL usado com CRLs e para algumas CA na cadeia de certificados sua CRL não está incluída em `TLSCRLFile`

No log do servidor TLS no caso de *OpenSSL* peer:

```
falhou ao aceitar uma conexão de entrada: de 127.0.0.1: handshake TLS com 127.0.0.1 retornou o código de
    arquivo s3_srvr.c linha 3251: erro:14089086: rotinas SSL:ssl3_get_client_certificate:certificação de
    TLS escreve alerta fatal "CA desconhecida"
```

No log do servidor TLS no caso do par *GnuTLS*:

```
falhou ao aceitar uma conexão de entrada: de 127.0.0.1: handshake TLS com 127.0.0.1 retornou o código de
    arquivo rsa_pk1.c linha 103: erro:0407006A: rotinas rsa:RSA_padding_check_PKCS1_type_1:\
    tipo de bloco não é 01 arquivo rsa_eay.c linha 705: erro:04067072: rotinas rsa:RSA_EAY_PUBLIC_DECRYPT
```

CRL expirou ou expira durante a operação do servidor

OpenSSL, no log do servidor:

- antes do vencimento:

```
não é possível conectar ao proxy "proxy-openssl-1.0.1e": TCP bem-sucedido, não é possível estabelecer TLS
    SSL_connect() retornou SSL_ERROR_SSL: arquivo s3_clnt.c linha 1253: erro:14090086:\
    Rotinas SSL:ssl3_get_server_certificate:certificação de verificação falhou:\
    TLS escreve alerta fatal "certificado revogado"
```

- após expiração:

```
não é possível conectar ao proxy "proxy-openssl-1.0.1e": TCP bem-sucedido, não é possível estabelecer TLS
    SSL_connect() retornou SSL_ERROR_SSL: arquivo s3_clnt.c linha 1253: erro:14090086:\
    Rotinas SSL:ssl3_get_server_certificate:certificação de verificação falhou:\
    Alerta fatal de gravação TLS "certificado expirado"
```

O ponto aqui é que, com CRL válida, um certificado revogado é relatado como ”certificado revogado”. Quando a CRL expira, a mensagem de erro muda para ”certificado expirou”, o que é bastante enganoso.

GnuTLS, no log do servidor:

- antes e depois do vencimento o mesmo:

```
não é possível conectar ao proxy "proxy-openssl-1.0.1e": TCP bem-sucedido, não é possível estabelecer TLS
      certificado de peer inválido: O certificado NÃO é confiável. A cadeia de certificados é revogada.
```

Certificado autoassinado, CA desconhecida

OpenSSL, no log:

```
error:'self signed certificate: SSL_connect() set result code to SSL_ERROR_SSL: file ../ssl/statem/statem_
      line 1924: error:1416F086:SSL routines:tls_process_server_certificate:certificate verify failed:\
      TLS write fatal alert "unknown CA"'
```

Isso foi observado quando o certificado do server, por engano, tinha a mesma string de Emissor e Assunto, embora tenha sido assinado por uma CA. Emissor e Assunto são iguais no certificado da CA de nível superior, mas não podem ser iguais no certificado do server. (O mesmo se aplica aos certificados do proxy e do agent.)

Para verificar se um certificado contém as mesmas entradas de Emissor e Assunto, execute:

```
openssl x509 -in <seucertificado.crt> -noout -text
```

É aceitável que o certificado raiz (de nível superior) tenha valores idênticos para Emissor e Assunto.

3 Problemas de PSK

PSK contém um número ímpar de dígitos hexadecimais

Proxy ou agent não inicia, mensagem no log do proxy ou agent:

```
invalid PSK in file "/home/zabbix/zabbix_proxy.psk"
```

Uma string de identidade PSK com mais de 128 bytes é passada para o GnuTLS

No log do lado do cliente TLS:

```
gnutls_handshake() failed: -110 The TLS connection was non-properly terminated.
```

No log do lado do servidor TLS.

```
gnutls_handshake() failed: -90 The SRP username supplied is illegal.
```

Valor PSK muito longo usado com OpenSSL 1.1.1

No log do lado da conexão:

```
...OpenSSL library (version OpenSSL 1.1.1  11 Sep 2018) initialized
...
...In zbx_tls_connect(): psk_identity:"PSK 1"
...zbx_psk_client_cb() requested PSK identity "PSK 1"
...End of zbx_tls_connect():FAIL error:'SSL_connect() set result code to SSL_ERROR_SSL: file ssl\statem\ex
```

No log do lado de aceitação:

```
...Message from 123.123.123.123 is missing header. Message ignored.
```

Esse problema normalmente ocorre ao atualizar o OpenSSL de 1.0.x ou 1.1.0 para 1.1.1 e se o valor PSK for maior que 512 bits (PSK de 64 bytes, inserido como 128 dígitos hexadecimais).

Veja também: Limites de tamanho de valor

18. Interface da Web

Visão geral Para um acesso fácil ao Zabbix de qualquer lugar e de qualquer plataforma, a interface baseada na web é fornecida.

> **Note:**
> Se estiver usando mais de uma instância de frontend, certifique-se de que os locais e bibliotecas (LDAP, SAML etc.) estejam instalados e configurados de forma idêntica para todos os frontends.

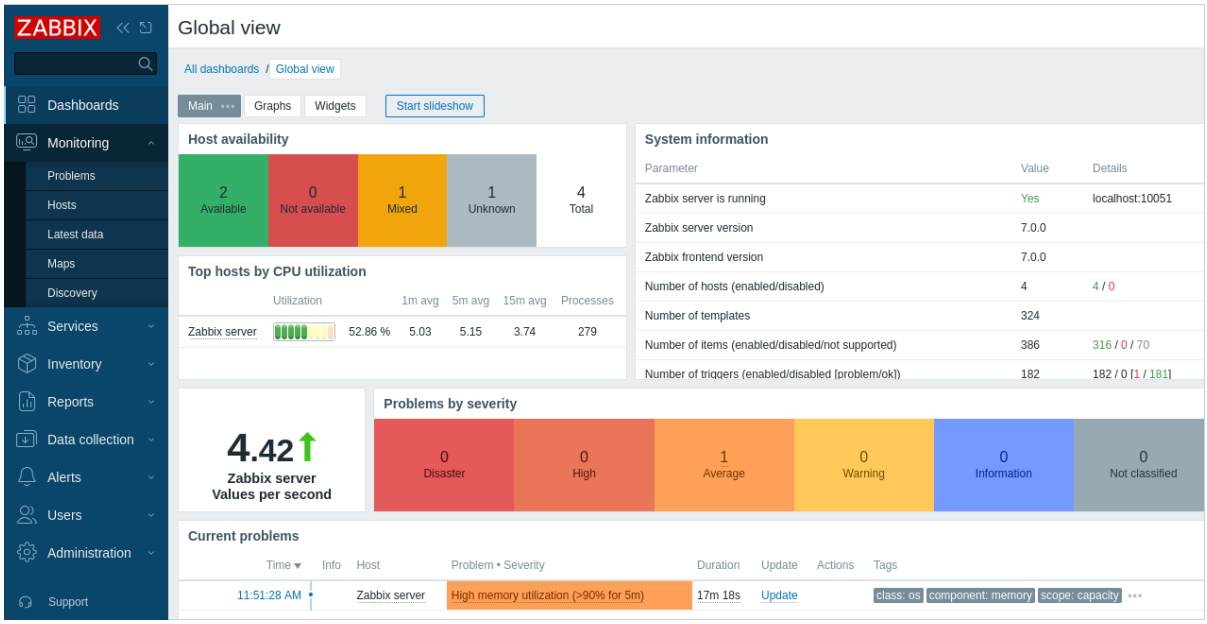
Ajuda do frontend Um link de ajuda ? é fornecido nos formulários do frontend do Zabbix com links diretos para as partes correspondentes da documentação.

1 Menu

Visão geral

Um menu vertical em uma barra lateral fornece acesso a várias seções do frontend do Zabbix.

O menu é azul escuro no tema padrão.

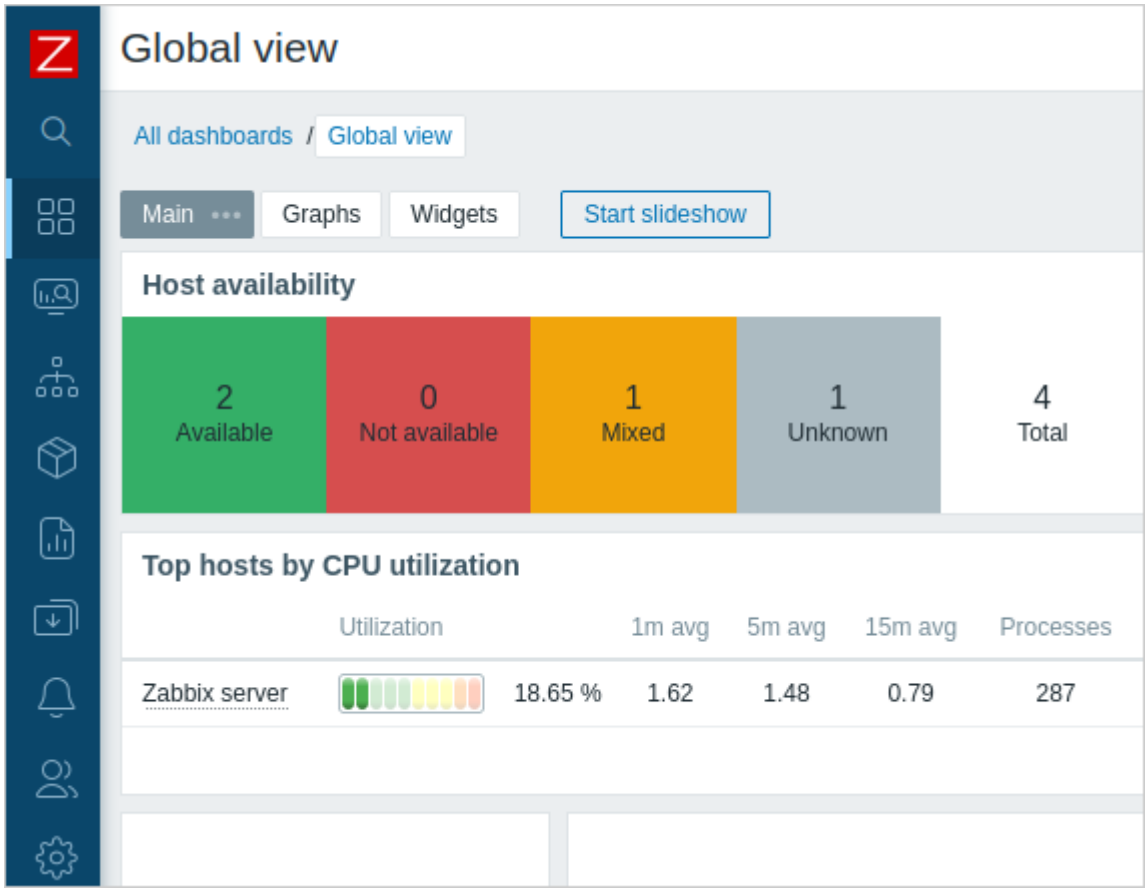


Trabalhando com o menu

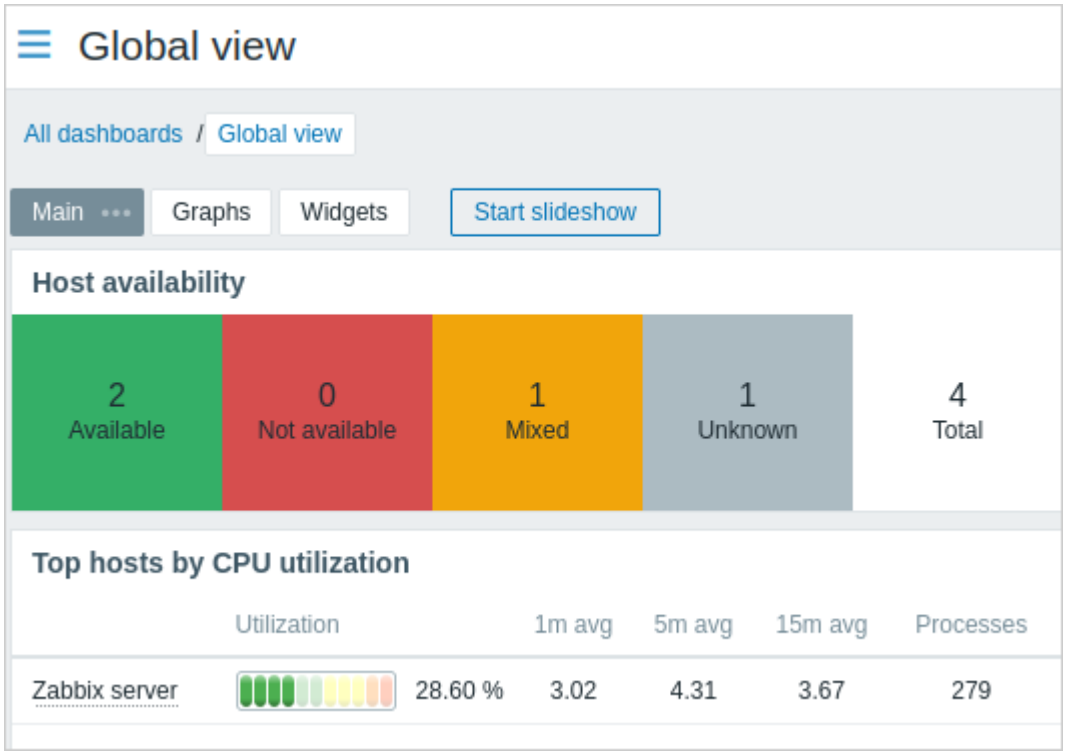
Uma caixa de pesquisa global está localizada abaixo do logotipo do Zabbix.

O menu pode ser recolhido ou ocultado completamente:

- Para recolher, clique em ao lado do logotipo do Zabbix. No menu recolhido, apenas os ícones ficam visíveis.



- Para ocultar, clique em ao lado do logotipo do Zabbix. No menu oculto, tudo fica escondido.



Menu recolhido

Quando o menu está recolhido apenas para ícones, um menu completo reaparece assim que o cursor do mouse é colocado sobre ele. Observe que ele reaparece sobre o conteúdo da página; para mover o conteúdo da página para a direita, você deve clicar no botão de expandir. Se o cursor do mouse for colocado novamente fora do menu completo, o menu será recolhido novamente após dois segundos.

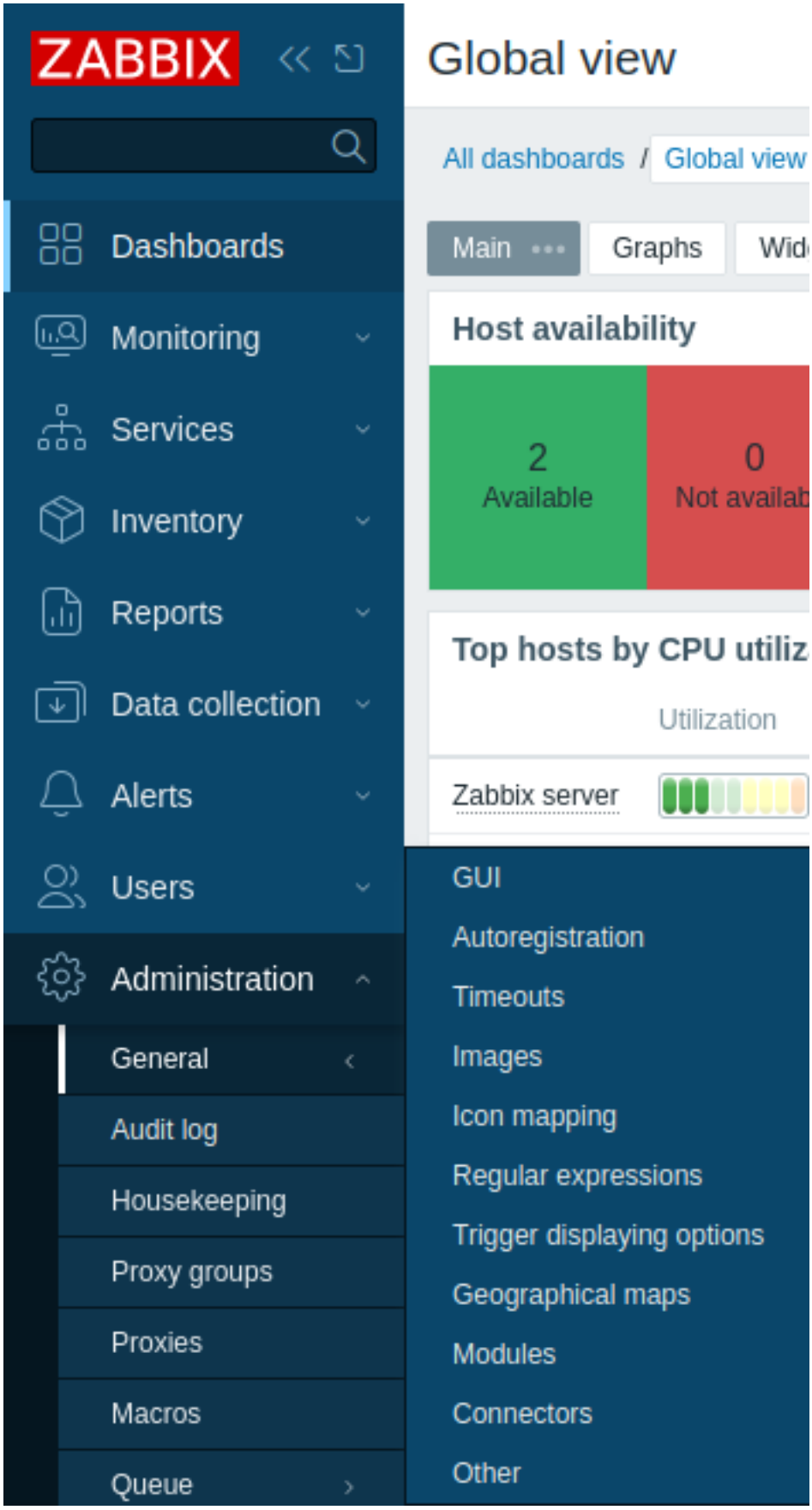
Você também pode fazer um menu recolhido reaparecer completamente pressionando a tecla Tab. Pressionar a tecla Tab repetidamente permitirá focar no próximo elemento do menu.

Menu oculto

Mesmo quando o menu está completamente oculto, um menu completo está a apenas um clique do mouse, clicando no ícone de hambúrguer . Observe que ele reaparece sobre o conteúdo da página; para mover o conteúdo da página para a direita, você deve exibir o menu clicando no botão mostrar barra lateral.

Níveis de menu

Existem até três níveis no menu.



Menus de contexto

Além do menu principal, o Zabbix fornece menus de contexto para host, item e evento para acesso rápido a entidades frequentemente usadas, como os valores mais recentes, gráfico simples, formulário de configuração, scripts relacionados ou links externos.

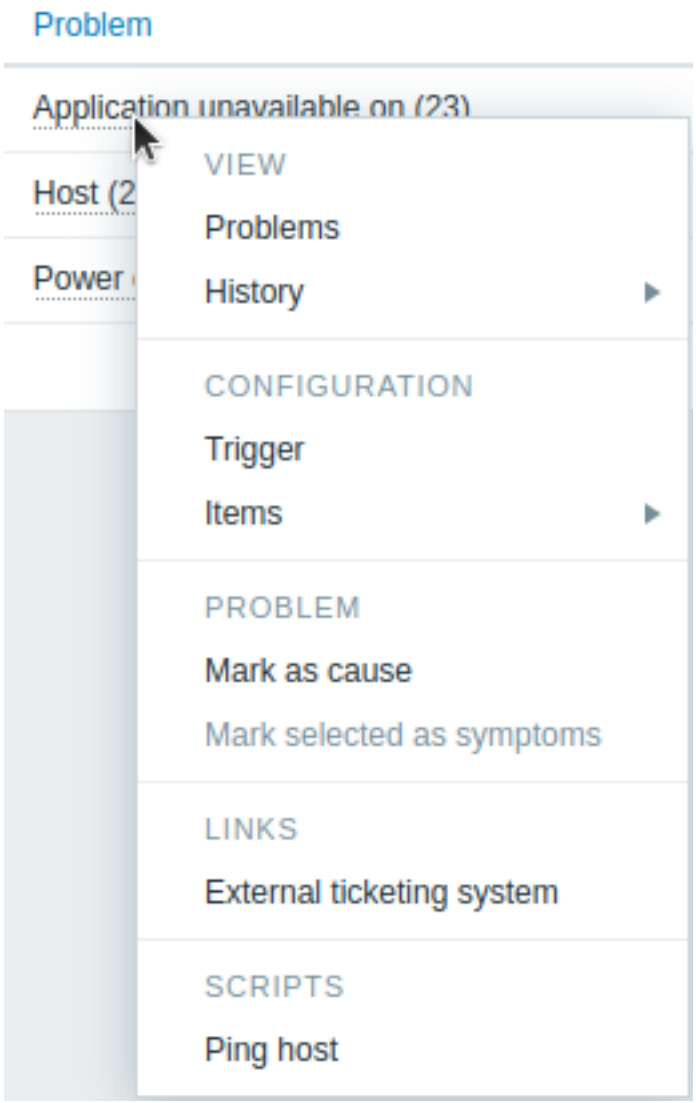
Os menus de contexto são acessíveis clicando no nome do host, item ou problema/trigger nos locais suportados.

1 Menu de evento

Visão geral

O menu de eventos contém atalhos para ações ou seções do frontend que são frequentemente necessárias para um evento.

O menu de eventos pode ser aberto clicando no nome do evento.



Conteúdo

O menu de contexto do evento possui seis seções: *Visualizar*, *Ações*, *Configuração*, *Problema*, *Links* e *Scripts*. Para as entidades que não estão configuradas, os links são desabilitados e exibidos em cinza. As seções *Scripts* e *Links* são exibidas se suas entidades estiverem configuradas.

A seção *Visualizar* contém links para:

- **Problemas** - abre a lista de problemas não resolvidos do trigger subjacente;
- **Histórico** - leva ao gráfico/histórico do item em *Últimos dados* para o(s) item(ns) subjacente(s). Se um trigger usar vários itens, links estarão disponíveis para cada um deles.

A seção *Ações* está disponível apenas em widgets de *Visão geral de triggers*. Ela contém um link para:

- **Atualizar problema** - abre a tela de atualização de problema.

A seção *Configuração* contém links para a configuração de:

- **Trigger** que disparou o problema;
- **Itens** usados na expressão do trigger.

Note:
Observe que a seção de configuração está disponível apenas para usuários Admin e Super admin.

A seção *Problema* contém as opções para:

- **Marcar como causa** - marcar o problema como causa;
- **Marcar selecionados como sintomas** - marcar os problemas selecionados como sintomas deste problema.

A seção *Links* contém links para:

- acessar uma URL do trigger configurada;
- acessar links personalizados configurados em Scripts globais (com escopo 'Ação manual de evento' e tipo 'URL');
- acessar um ticket externo configurado para o problema (veja a opção *Incluir entrada de menu de evento* ao configurar webhooks).

A seção *Scripts* contém links para executar um script global (com escopo *Ação manual de evento*). Esse recurso pode ser útil para executar scripts usados para gerenciar tickets de problemas em sistemas externos.

Locais suportados

O menu de contexto do evento é acessível clicando em um problema ou nome de evento em várias seções do frontend, por exemplo:

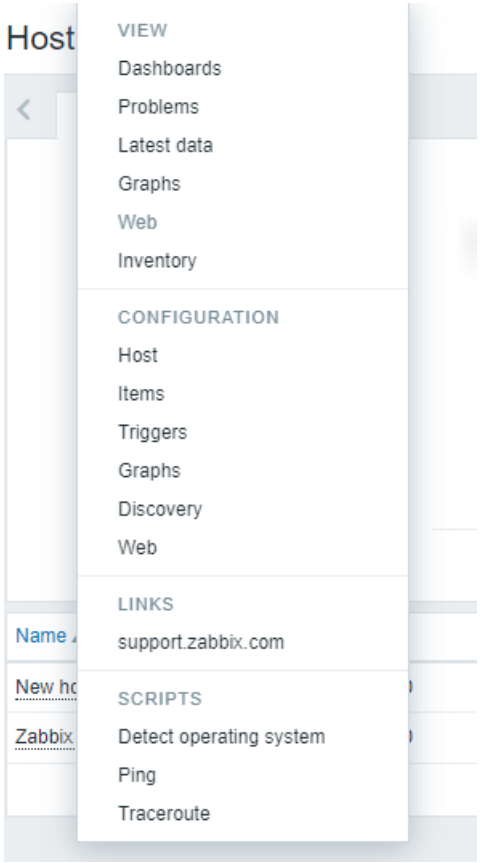
- Dashboards widgets, como o widget *Problemas*, widget *Visão geral dos triggers*, etc.
- Monitoramento → Problemas
- Monitoramento → Problemas → Detalhes do evento
- Relatórios → Top 100 triggers (scripts globais e acesso a chamados externos não são suportados neste local)

2 Menu de contexto de host

Visão geral

O menu do host contém atalhos para ações ou seções do frontend que são frequentemente necessárias para um host.

O menu do host pode ser aberto clicando no nome do host.



Conteúdo

O menu de contexto do host possui quatro seções: *Visualizar*, *Configuração*, *Links* e *Scripts*. Para as entidades que não estão configuradas, os links são desabilitados e exibidos na cor cinza. As seções *Scripts* e *Links* são exibidas se suas entidades estiverem configuradas.

A seção *Visualizar* contém links para:

- **Dashboards** - abre widgets e gráficos.
- **Problemas** - abre a seção *Problemas* com a lista de problemas não resolvidos do trigger subjacente.
- **Últimos dados** - abre a seção *Últimos dados* com a lista de todos os últimos dados do host atual.
- **Gráficos** - abre gráficos simples do host atual.
- **Web** - abre o link para os cenários web configurados.
- **Inventário** - abre o link para o inventário do host atual.

A seção *Configuração* contém links para:

- **Host** - formulário de configuração do host atual.
- **Itens** - a lista de items do host atual.
- **Triggers** - a lista de triggers do host atual.
- **Gráficos** - gráficos simples do host atual.
- **Descoberta** - a lista de regras de descoberta de baixo nível do host atual.
- **Web** - a lista de cenários web do host atual.

> **Note:**
> Observe que a seção de configuração está disponível apenas para usuários Admin e Super admin.

A seção *Links* contém links para:

- acessar uma URL do trigger configurada.
- acessar links personalizados configurados em Scripts globais (com escopo *Ação manual do host* e tipo 'URL').

A seção *Scripts* permite executar scripts globais configurados para o host atual. Esses scripts precisam ter seu escopo definido como *Ação manual do host* para estarem disponíveis no menu do host.

Locais suportados

O menu de host é acessível clicando no nome do host em várias seções do frontend, por exemplo:

- Dashboards widgets, como Visão geral dos dados, Visão geral dos triggers, Problemas, etc.
- Monitoramento → Problemas
- Monitoramento → Problemas → Detalhes do evento
- Monitoramento → Hosts
- Monitoramento → Hosts → Web Monitoring
- Monitoramento → Últimos dados
- Monitoramento → Mapas
- Inventário → Hosts
- Relatórios → Top 100 triggers

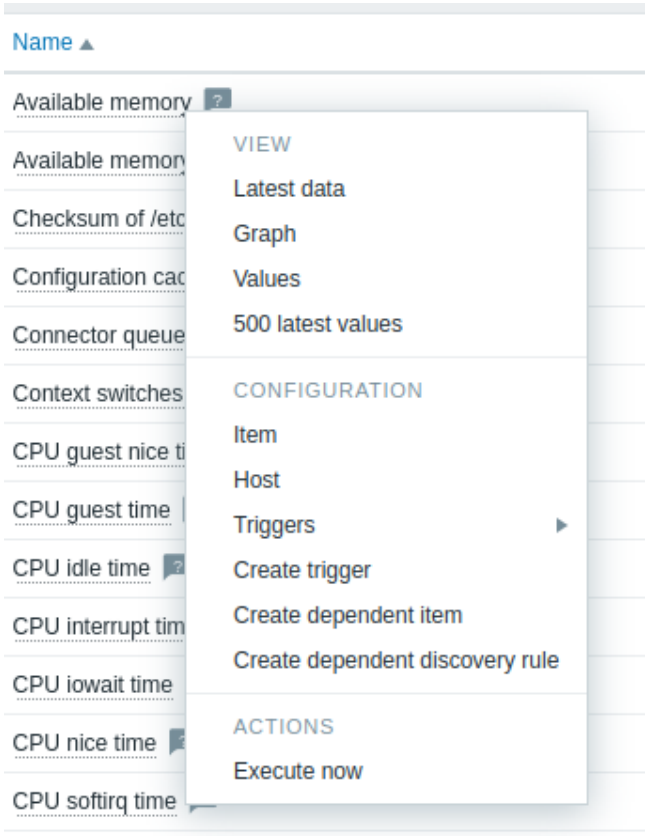
3 Menu de item

Visão geral

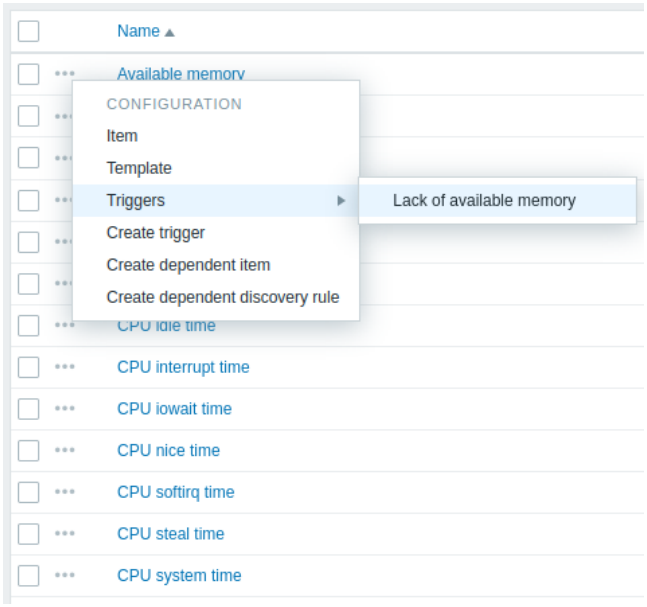
O menu de item contém atalhos para ações ou seções do frontend que são frequentemente necessárias para um item.

O menu de item pode ser aberto clicando no nome do item ou no ícone de três pontos, dependendo da seção do frontend, por exemplo:

Monitoramento → Últimos dados



Coleta de dados → Templates → Itens



Conteúdo

O menu de item possui três seções: *Visualizar*, *Configuração* e *Ações*.

A seção *Visualizar* contém as seguintes opções:

- **Últimos dados** - abre a seção *Últimos dados* filtrada pelo host e item atuais;
- **Gráfico** - abre um gráfico simples do item atual;
- **Valores** - abre a lista de todos os valores recebidos para o item atual nos últimos 60 minutos;
- **500 últimos valores** - abre a lista dos 500 últimos valores para o item atual.

A seção *Configuração* (disponível apenas para usuários do tipo *Admin* e *Super admin*) contém as seguintes opções:

- **Item** - abre o formulário de configuração do item do item atual;
- **Template** - abre o formulário de configuração do template ao qual o item pertence (desde o Zabbix 7.0.4; disponível apenas quando o menu de item é acessado em *Coleta de dados* → *Templates* → *Itens*);
- **Host** - abre o formulário de configuração do host ao qual o item pertence (desde o Zabbix 7.0.4);
- **Triggers** - ao passar o mouse, abre uma lista de triggers do item, se houver; ao clicar em um trigger, abre seu formulário de configuração;
- **Criar trigger** - abre o formulário de configuração do trigger para criar um trigger para este item;
- **Criar item dependente** - abre o formulário de configuração do item para criar um item dependente, com o item atual como item mestre;
- **Criar regra de descoberta dependente** - abre o formulário de configuração da regra de descoberta para criar uma regra de descoberta dependente, com o item atual como item mestre.

A seção *Ações* contém a seguinte opção:

- **Executar agora** - executa uma checagem para um novo valor do item imediatamente.

Locais suportados

O menu de item é acessível clicando no nome de um item em várias seções do frontend, por exemplo:

- Monitoramento → Últimos dados
- Coleta de dados → Hosts → Itens
- Coleta de dados → Hosts → Regras de descoberta → Protótipos de item

2 Seções de front-end

Estrutura do menu O menu do frontend do Zabbix possui a seguinte estrutura:

- Dashboards
- Monitoramento
  - Problemas
  - Hosts
  - Últimos dados
  - Mapas
  - Descoberta
- Serviços
  - Serviços
  - SLA
  - Relatório de SLA
- Inventário
  - Visão geral
  - Hosts
- Relatórios
  - Informações do sistema
  - Relatórios agendados
  - Relatório de disponibilidade
  - Top 100 triggers
  - Log de auditoria
  - Log de ações
  - Notificações
- Coleta de dados
  - Grupos de templates
  - Grupos de hosts
  - Templates
  - Hosts
  - Manutenção
  - Correlação de eventos
  - Descoberta
- Alertas
  - Ações
    - Ações de trigger
    - Ações de serviço
    - Ações de descoberta
    - Ações de autorregistro
    - Ações internas
  - Tipos de mídia
  - Scripts
- Usuários
  - Grupos de usuários
  - Funções de usuário
  - Usuários
  - Tokens de API
  - Autenticação
- Administração
  - Geral
    - GUI
    - Autorregistro
    - Timeouts
    - Imagens
    - Mapeamento de ícones
    - Expressões regulares
    - Opções de exibição de triggers
    - Mapas geográficos
    - Módulos
    - Conectores

    - Outros
- Log de auditoria
- Housekeeping
- Grupos de proxy
- Proxies
- Macros
- Fila
    - Visão geral da fila
    - Visão geral da fila por proxy
    - Detalhes da fila

1 Dashboards

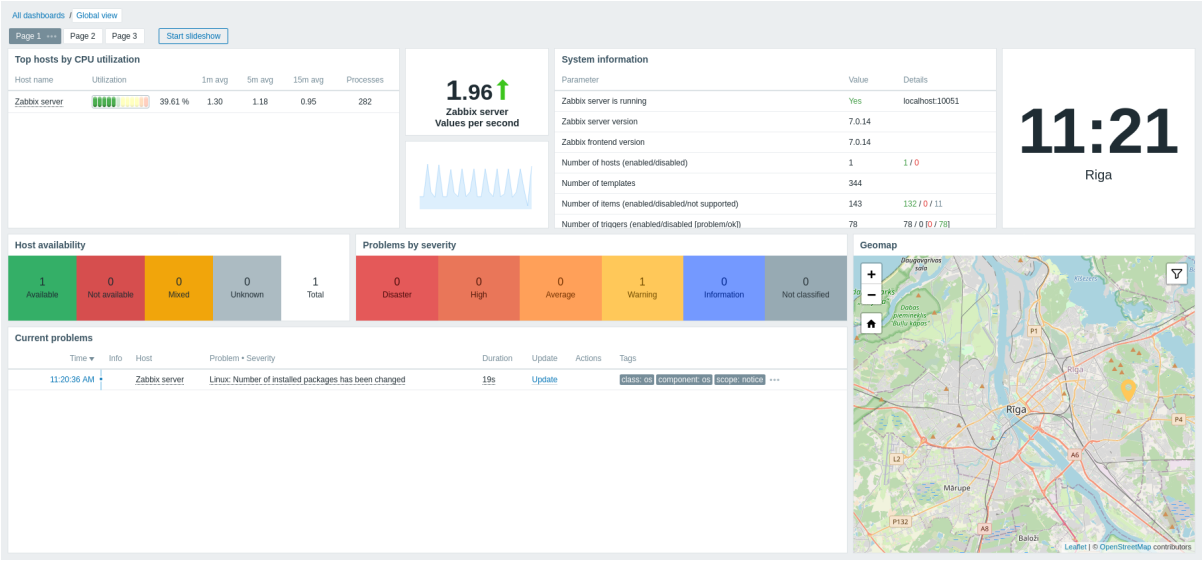
Visão geral

A seção *Dashboards* foi projetada para exibir resumos de todas as informações importantes em um **dashboard**.

Embora apenas um dashboard possa ser exibido por vez, é possível configurar vários dashboards. Cada dashboard pode conter uma ou várias páginas que podem ser alternadas em uma apresentação de slides.

Uma página de dashboard consiste em widgets e cada widget é projetado para exibir informações de um determinado tipo e fonte, que pode ser um resumo, um mapa, um gráfico, o relógio, etc.

O acesso aos hosts nos widgets depende das permissões do host.



Páginas e widgets são adicionados ao dashboard e editados no modo de edição do dashboard. As páginas podem ser visualizadas e alternadas no modo de visualização do dashboard.

O período de tempo exibido nos widgets de gráfico é controlado pelo seletor de *Período de tempo* localizado acima dos widgets. O rótulo do seletor de *Período de tempo*, localizado à direita, exibe o período de tempo atualmente selecionado. Clicar no rótulo da guia permite expandir e recolher o seletor de *Período de tempo*.

Tamanho do dashboard

A largura mínima de um dashboard é de 1200 pixels. O dashboard não será reduzido abaixo dessa largura; em vez disso, uma barra de rolagem horizontal será exibida se a janela do navegador for menor do que isso.

A largura máxima de um dashboard é a largura da janela do navegador. Os widgets do dashboard se estendem horizontalmente para se ajustar à janela. Ao mesmo tempo, um widget do dashboard não pode ser estendido horizontalmente além dos limites da janela.

Horizontalmente, o dashboard é composto por 72 colunas de largura sempre igual, que se estendem/reduzem dinamicamente (mas não para menos de 1200 pixels no total).

Verticalmente, o dashboard pode conter no máximo 64 linhas; cada linha tem uma altura fixa de 70 pixels.

Um widget pode, portanto, ter até 72 colunas de largura e 64 linhas de altura.

Visualizando dashboards

Para visualizar todos os dashboards configurados, clique em *Todos os dashboards* logo abaixo do título da seção.

Os dashboards são exibidos com uma tag de compartilhamento:

- *Meu* - indica um dashboard de propriedade do usuário atual
- *Compartilhado* - indica um dashboard público ou um dashboard privado compartilhado com qualquer usuário ou grupo de usuários

Observe que as tags de compartilhamento são exibidas apenas para os dashboards de propriedade do usuário atual. Os dashboards de outros usuários são listados sem tags de compartilhamento.

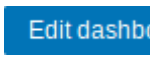
O filtro localizado à direita acima da lista permite filtrar dashboards pelo nome e por aqueles criados pelo usuário atual.

Para excluir um ou vários dashboards, marque as caixas de seleção dos respectivos dashboards e clique em *Excluir* abaixo da lista.

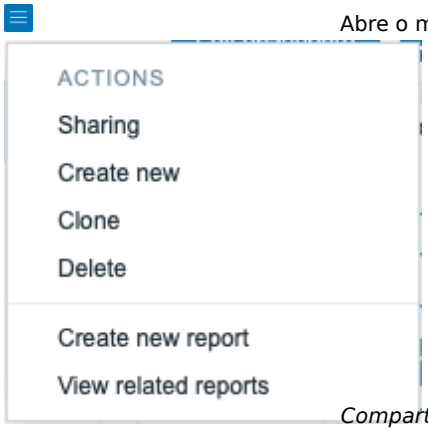
Visualizando um dashboard

Para visualizar um dashboard, clique em seu nome na lista de dashboards.

Ao **visualizar** um dashboard, as seguintes opções estão disponíveis:



Alternar para o modo de **edição** do dashboard.
O modo de edição também é aberto quando um novo dashboard está sendo criado e quando você clica no botão de edição de um widget.



Abre o menu de ações (veja as descrições das ações abaixo).

Compartilhar - editar as preferências de compartilhamento do dashboard.
Criar novo - criar um novo dashboard.
Clonar - criar um novo dashboard copiando as propriedades do existente. Primeiro, será solicitado que você insira os parâmetros do dashboard. Em seguida, o novo dashboard é aberto no modo de edição com todos os widgets do dashboard original.
Excluir - excluir o dashboard.
Criar novo relatório - abre uma janela pop-up com o formulário de configuração do relatório. Desabilitado se o usuário não tiver permissão para gerenciar relatórios agendados.
Visualizar relatórios relacionados - abre uma janela pop-up com uma lista de relatórios existentes baseados no dashboard atual. Desabilitado se não houver relatórios relacionados ou se o usuário não tiver permissão para visualizar relatórios agendados.

	Exibe apenas o conteúdo da página (modo quiosque). O modo quiosque também pode ser acessado com os seguintes parâmetros de URL: `/zabbix.php?action=dashboard.view&kiosk=1`. Para sair para o modo normal: `/zabbix.php?action=dashboard.view&kiosk=0`

Compartilhamento

Os dashboards podem ser públicos ou privados.

Dashboards públicos são visíveis para todos os usuários. Dashboards privados são visíveis apenas para seu proprietário e para os usuários/grupos de usuários adicionados nas preferências de compartilhamento.

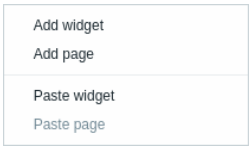
Para editar o status de compartilhamento de um dashboard, clique na opção *Compartilhamento* no menu de ações ao visualizar um único dashboard:

Parâmetro	Descrição
Tipo	Selecione o tipo de dashboard: **Privado** - o dashboard é visível apenas para grupos de usuários e usuários selecionados **Público** - o dashboard é visível para todos
Lista de compartilhamento de grupos de usuários	Selecione os grupos de usuários que terão acesso ao dashboard. Você pode permitir acesso somente leitura ou leitura e gravação.
Lista de compartilhamento de usuários	Selecione os usuários que terão acesso ao dashboard. Você pode permitir acesso somente leitura ou leitura e gravação.

Veja Permissões para dashboards para saber como as opções de compartilhamento limitam as ações disponíveis para os usuários.

Editando um dashboard

Ao **editar** um dashboard, as seguintes opções estão disponíveis:

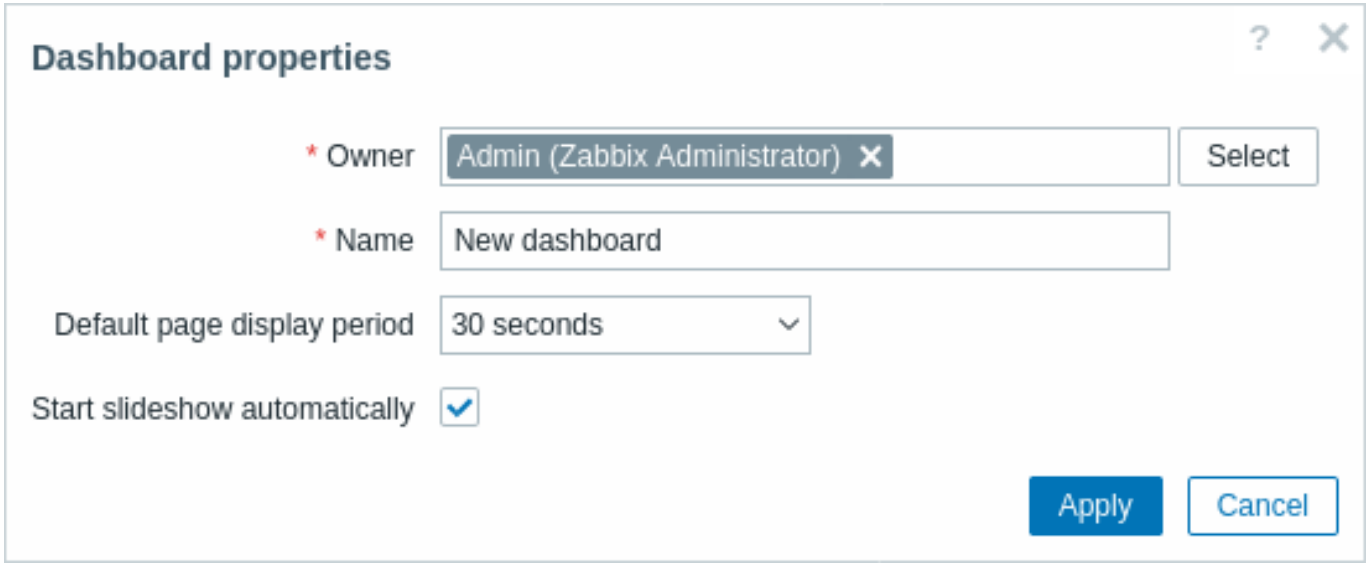
	Editar os parâmetros gerais do dashboard. Adicionar um novo widget. Clicar no botão de seta abrirá o menu de ações (veja as descrições das ações abaixo).
	*Adicionar widget* - adiciona um novo widget *Adicionar página* - adiciona uma nova página *Colar widget* - cola um widget copiado. Esta opção fica desabilitada se nenhum widget tiver sido copiado. Apenas uma entidade (widget ou página) pode ser copiada por vez. *Colar página* - cola uma página copiada. Esta opção fica desabilitada se nenhuma página tiver sido copiada.
	Salvar as alterações do dashboard. Cancelar as alterações do dashboard.

Criando um dashboard

É possível criar um novo dashboard de duas maneiras:

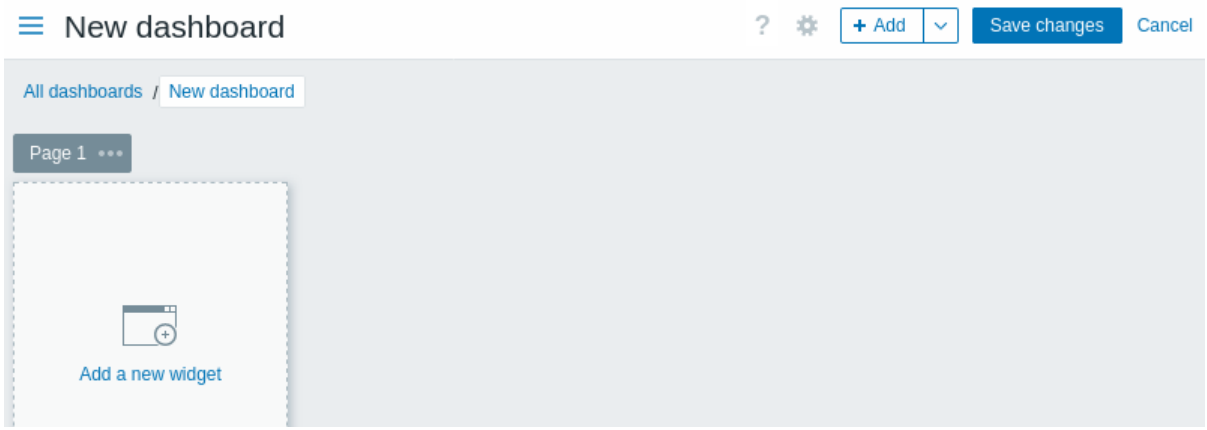
- Clique em *Criar dashboard*, ao visualizar todos os dashboards
- Selecione *Criar novo* no menu de ações, ao visualizar um único dashboard

Primeiro, será solicitado que você insira os parâmetros gerais do dashboard:



Parâmetro	Descrição
Proprietário	Selecione o usuário do sistema que será o proprietário do dashboard.
Nome	Insira o nome do dashboard.
Período padrão de exibição da página	Selecione o período pelo qual uma página do dashboard será exibida antes de alternar para a próxima página em uma apresentação de slides.
Iniciar apresentação de slides automaticamente	Marque esta caixa de seleção para executar uma apresentação de slides automaticamente quando houver mais de uma página de dashboard.

Ao clicar em *Aplicar*, um dashboard vazio será aberto:



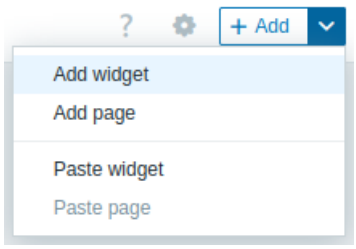
Para preencher o dashboard, você pode adicionar widgets e páginas.

Clique no botão *Salvar alterações* para salvar o dashboard. Se você clicar em *Cancelar*, o dashboard não será criado.

Adicionando widgets

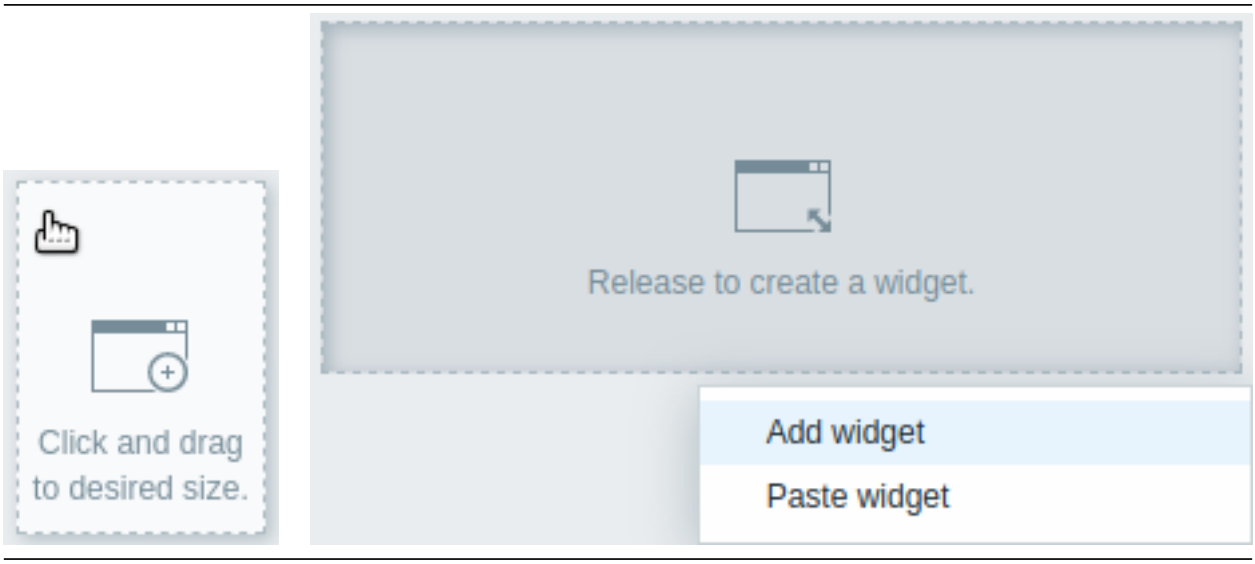
Você pode adicionar vários widgets (Log de ações, Relógio, Status de descoberta, etc.) a um dashboard. Os widgets podem ser adicionados de duas maneiras:

- Clique no botão *Adicionar* ou na seta ao lado dele e selecione *Adicionar widget* no menu de ações. Após configurar o widget, ele será adicionado em seu tamanho padrão e posicionado após qualquer widget existente.



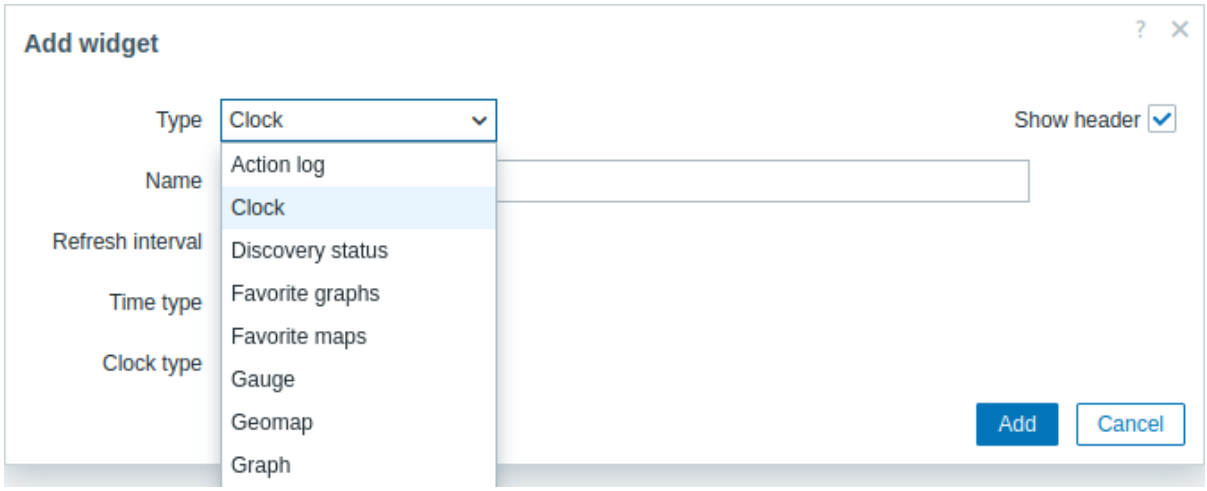
- Mova o mouse para um espaço vazio no dashboard. Observe que um espaço reservado aparece, então clique nele para abrir o formulário de configuração do widget. Após configurar o widget, ele será adicionado em seu tamanho padrão ou

redimensionado para caber no espaço disponível. Alternativamente, arraste o espaço reservado para determinar o tamanho do widget e solte-o para abrir o formulário de configuração do widget (se um widget for copiado para a área de transferência, você será solicitado primeiro a escolher entre *Adicionar widget* ou *Colar widget*).



No formulário de configuração do widget:

1. Selecione o tipo de widget.
2. Configure os parâmetros do widget.
3. Clique em *Adicionar* para adicionar o widget ao dashboard.



Editando widgets

Cada widget inclui controles em seu canto superior direito:

- - editar o widget
- - abrir o menu do widget

Após editar, clique em *Aplicar* no widget e depois em *Salvar alterações* no canto superior direito do dashboard para aplicar suas alterações.

Copiando/colando widgets

Você pode copiar e colar widgets do dashboard para criar rapidamente novos widgets com a mesma configuração. Os widgets podem ser copiados e colados dentro do mesmo dashboard ou entre dashboards abertos em abas diferentes.

Para copiar um widget, use o menu do widget. Para colar um widget copiado (disponível no modo de edição do dashboard):

- Clique na seta ao lado do botão *Adicionar* e selecione *Colar widget*.
- Alternativamente, clique em uma área vazia do dashboard e selecione *Colar widget*.

Você também pode colar um widget copiado sobre um widget existente usando a opção *Colar* no seu menu de widget.

Criando uma apresentação de slides

Uma apresentação de slides será executada automaticamente se o dashboard contiver duas ou mais páginas (consulte Adicionando páginas) e se uma das seguintes condições for verdadeira:

- A opção *Iniciar apresentação de slides automaticamente* estiver marcada nas propriedades do dashboard
- A URL do dashboard contiver o parâmetro `slideshow=1`

As páginas são alternadas de acordo com os intervalos definidos nas propriedades do dashboard e das páginas individuais. Clique em:

- *Parar apresentação de slides* - para parar a apresentação de slides
- *Iniciar apresentação de slides* - para iniciar a apresentação de slides

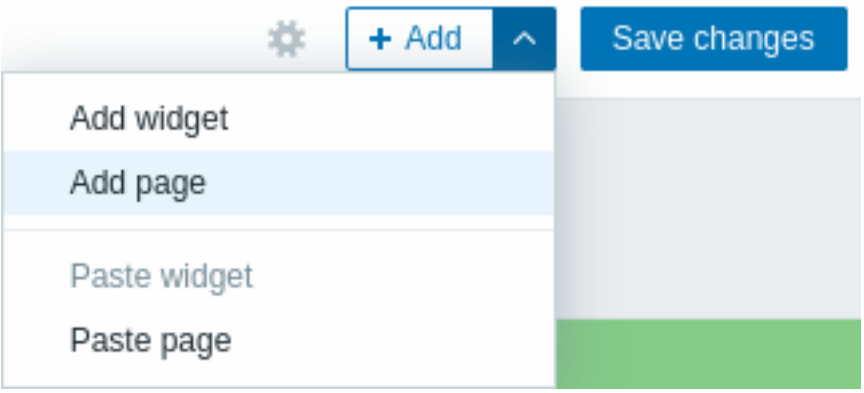
Os controles relacionados à apresentação de slides também estão disponíveis no modo quiosque (onde apenas o conteúdo da página é exibido):

- - parar apresentação de slides
- - iniciar apresentação de slides
- - voltar uma página
- - ir para a próxima página

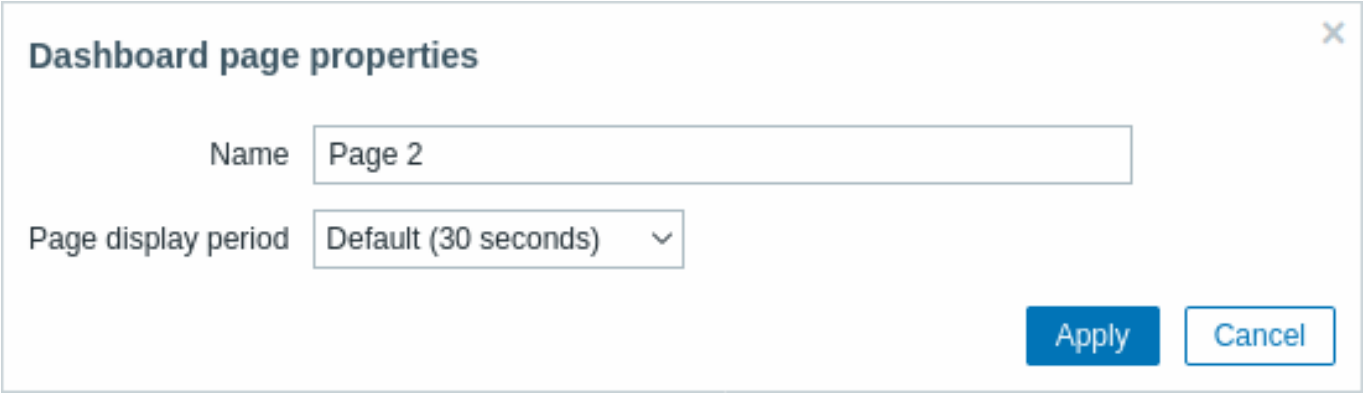
Adicionando páginas

Para adicionar uma nova página a um dashboard:

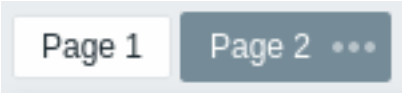
- Certifique-se de que o dashboard está no modo de edição
- Clique na seta ao lado do botão *Adicionar* e selecione a opção *Adicionar página*



- Preencha os parâmetros gerais da página e clique em *Aplicar*. Se você deixar o nome em branco, a página será adicionada com o nome `Página N`, onde 'N' é o número incremental da página. O período de exibição da página permite personalizar por quanto tempo uma página é exibida em uma apresentação de slides.



Uma nova página será adicionada, indicada por uma nova aba (*Página 2*).



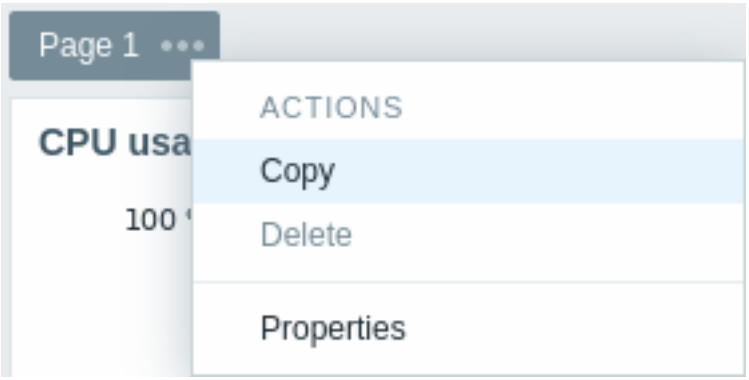
As páginas podem ser reordenadas arrastando e soltando as abas das páginas. A reordenação mantém a nomenclatura original da página. Sempre é possível ir para cada página clicando em sua aba.

Quando uma nova página é adicionada, ela está vazia. Você pode adicionar widgets a ela conforme descrito acima.

Copiando/colando páginas

As páginas do dashboard podem ser copiadas e coladas, permitindo criar uma nova página com as propriedades de uma existente. Elas podem ser coladas no mesmo dashboard ou em um dashboard diferente.

Para colar uma página existente no dashboard, primeiro copie-a, usando o menu da página:

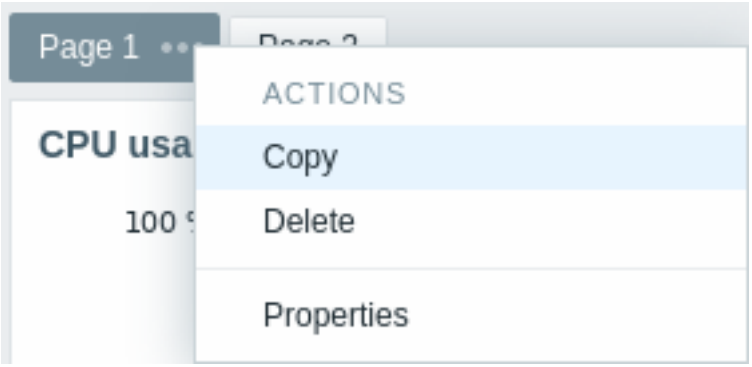


Para colar a página copiada:

- Certifique-se de que o dashboard está no modo de edição
- Clique na seta ao lado do botão *Adicionar* e selecione a opção *Colar página*

Menu da página

O menu da página pode ser aberto clicando nos três pontos ao lado do nome da página:

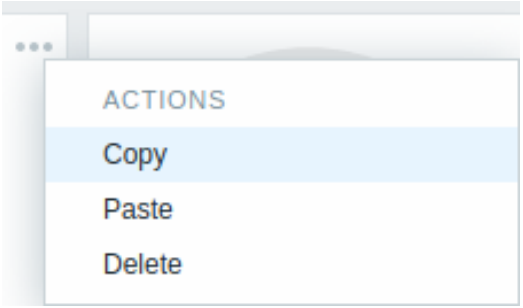
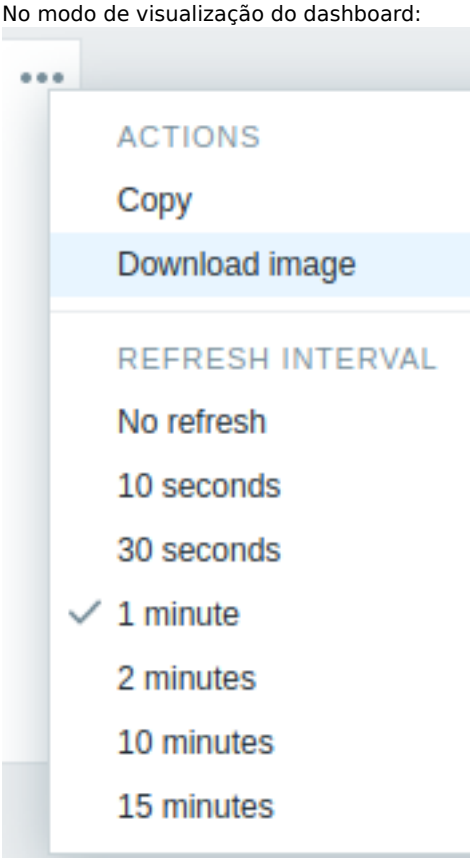


Ele contém as seguintes opções:

- *Copiar* - copia a página
- *Excluir* - exclui a página (as páginas só podem ser excluídas no modo de edição do dashboard)
- *Propriedades* - personaliza os parâmetros da página (o nome e o período de exibição da página em uma apresentação de slides)

Menu do widget

O menu do widget contém diferentes opções dependendo se o dashboard está no modo de edição ou visualização:

Menu do widget	Opções
No modo de edição do dashboard: 	*Copiar* - copia o widget *Colar* - cola um widget copiado sobre este widget Esta opção fica desabilitada se nenhum widget foi copiado. *Excluir* - exclui o widget
No modo de visualização do dashboard: 	*Copiar* - copia o widget *Baixar imagem* - baixa o widget como uma imagem PNG (disponível apenas para widgets de gráfico/gráfico clássico) *Intervalo de atualização* - seleciona a frequência de atualização do conteúdo do widget

Permissões para dashboards

As permissões para dashboards para usuários comuns e usuários do tipo 'Admin' (desde que o acesso aos dashboards e a opção *Criar e editar dashboards* estejam habilitados para a função de usuário atribuída a eles) são limitadas conforme descrito abaixo.

- Eles podem visualizar e clonar um dashboard público mesmo que nenhum usuário ou grupo de usuários seja adicionado nas preferências de compartilhamento do dashboard.
- Eles podem visualizar e clonar um dashboard privado se tiverem pelo menos direitos de *Leitura* definidos por meio das preferências de compartilhamento.
- Eles podem editar e excluir um dashboard apenas se tiverem direitos de *Leitura e gravação* definidos por meio das preferências de compartilhamento.
- Eles não podem alterar o proprietário do dashboard.

1 Dashboard widgets

Visão geral

Esta seção fornece detalhes dos parâmetros que são comuns para widgets de dashboard.

Parâmetros comuns

Os seguintes parâmetros são comuns para todos os widgets:

Nome	Digite um nome para o widget.
Intervalo de atualização	Configure o intervalo de atualização padrão. Os intervalos de atualização padrão para widgets variam de *Sem atualização* a *15 minutos*, dependendo do tipo de widget. Por exemplo: - *Sem atualização* para o widget *URL*; - *1 minuto* para o widget *Log de ações*; - *15 minutos* para o widget *Relógio*. Os intervalos de atualização podem ser definidos para um valor padrão para todos os usuários. Altere o dashboard para o modo de edição, clique no botão para editar um widget e selecione o intervalo de atualização desejado na lista suspensa. Cada usuário também pode definir seu próprio intervalo de atualização do widget. No modo de visualização do dashboard, clique nos três pontos em um widget e selecione o intervalo de atualização desejado na lista suspensa. Observe que o intervalo de atualização exclusivo do usuário tem prioridade sobre a configuração do widget e é preservado mesmo quando a configuração do widget é modificada.
Mostrar cabeçalho	Marque a caixa de seleção para exibir permanentemente o cabeçalho do widget. Quando desmarcado, o cabeçalho é ocultado para economizar espaço e torna-se visível apenas ao passar o mouse sobre o widget (tanto nos modos de visualização quanto de edição). O cabeçalho também fica semi-visível ao arrastar um widget para um novo local.

Parâmetros específicos

Para ver os parâmetros específicos de cada widget, acesse as páginas individuais dos widgets:

- Log de ações
- Relógio
- Visão geral dos dados (obsoleto; será removido na próxima versão principal)
- Status de descoberta
- Gráficos favoritos
- Mapas favoritos
- Medidor
- Geomapa
- Gráfico
- Gráfico (clássico)
- Protótipo de gráfico
- Favo de mel
- Disponibilidade do host
- Navegador de host
- Histórico do item
- Navegador de item
- Valor do item
- Mapa
- Árvore de navegação do mapa
- Gráfico de pizza
- Hosts com problemas
- Problemas
- Problemas por severidade
- Relatório de SLA
- Informações do sistema
- Principais hosts
- Principais triggers
- Visão geral das triggers
- URL
- Monitoramento web

Parâmetros dinâmicos

Vários widgets possuem parâmetros que permitem compartilhar dados de configuração entre outros widgets ou o dashboard.

Grupos de hosts, Hosts, Item, Lista de items

Os parâmetros *Grupos de hosts*, *Hosts*, *Item* e *Lista de items* (desde o Zabbix 7.0.1) permitem selecionar as respectivas entidades ou uma fonte de dados contendo grupos de hosts, hosts ou items para os quais o widget pode exibir dados.

Para os parâmetros *Grupos de hosts*, *Item* e *Lista de items*, a fonte de dados pode ser um widget compatível do mesmo dashboard.

Para o parâmetro *Hosts*, a fonte de dados pode ser um widget compatível do mesmo dashboard ou o próprio dashboard.

Note:
O widget *Mapa* também pode transmitir dados de grupo de hosts e host para widgets compatíveis. Para mais informações, consulte Comportamento do widget.

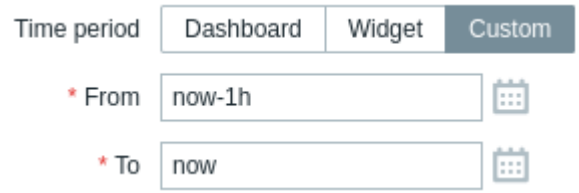
Substituir host

O parâmetro *Substituir host* permite selecionar uma fonte de dados contendo um host para o qual o widget pode exibir dados. A fonte de dados pode ser um widget compatível do mesmo dashboard ou o próprio dashboard.

- Para especificar um widget compatível, digite seu nome e selecione-o. Como alternativa, clique no botão *Selecionar* (ou no botão suspenso e, em seguida, em "Widget") para abrir um pop-up com os widgets disponíveis.
- Para especificar um dashboard, clique no botão suspenso e, em seguida, em "Dashboard". Após salvar o dashboard, o seletor *Host* aparecerá na parte superior do dashboard.

Período de tempo

O parâmetro *Período de tempo* permite selecionar uma fonte de dados contendo um período de tempo para o qual o widget pode exibir dados. A fonte de dados pode ser um widget compatível do mesmo dashboard, o próprio dashboard ou o período de tempo configurado no próprio widget.



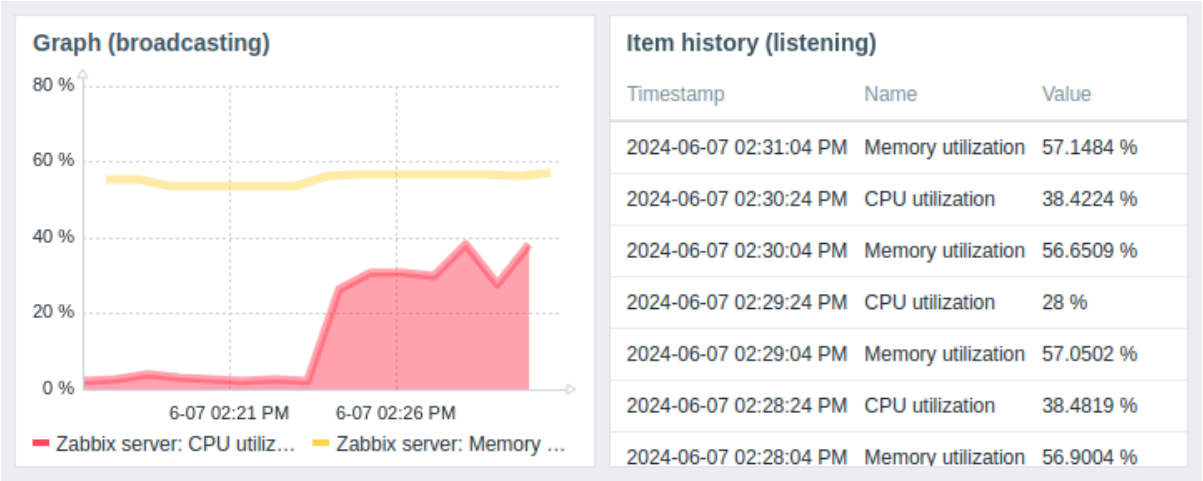
- Para especificar um widget compatível, defina *Período de tempo* como "Widget", insira seu nome e selecione-o. Alternativamente, clique no botão *Selecionar* para abrir um pop-up com os widgets disponíveis.
- Para especificar um dashboard, defina *Período de tempo* como "Dashboard". Após salvar o dashboard, o seletor de *Período de tempo* aparecerá no topo do dashboard.
- Para configurar o período de tempo no próprio widget, defina *Período de tempo* como "Personalizado" e insira ou selecione o início e o fim do período de tempo.

Note:
Independentemente da configuração de *Período de tempo* do widget, widgets compatíveis podem usá-lo como fonte de dados para o período de tempo.

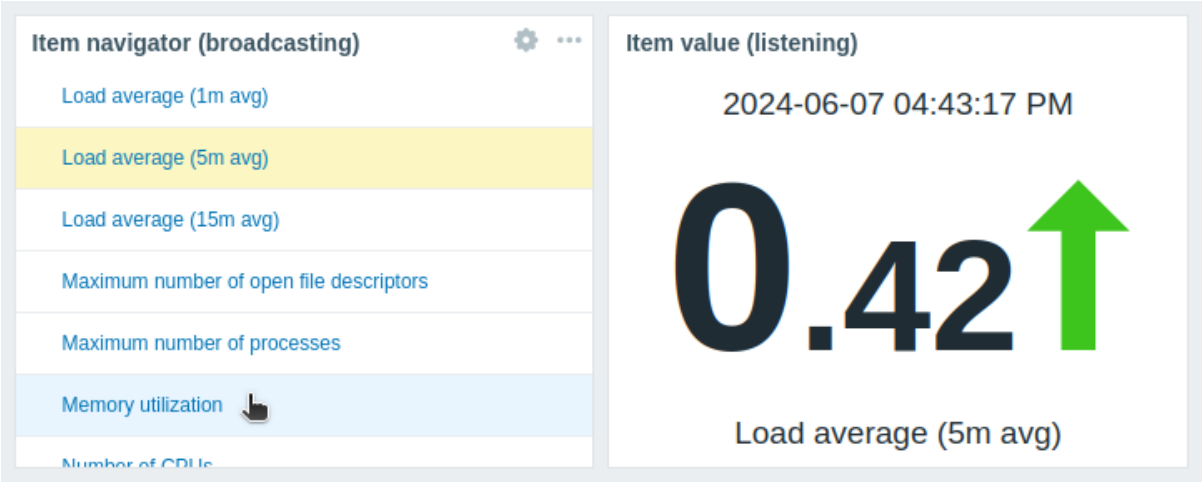
Comportamento do widget

Os widgets diferem em como **transmitem** dados para outros widgets.

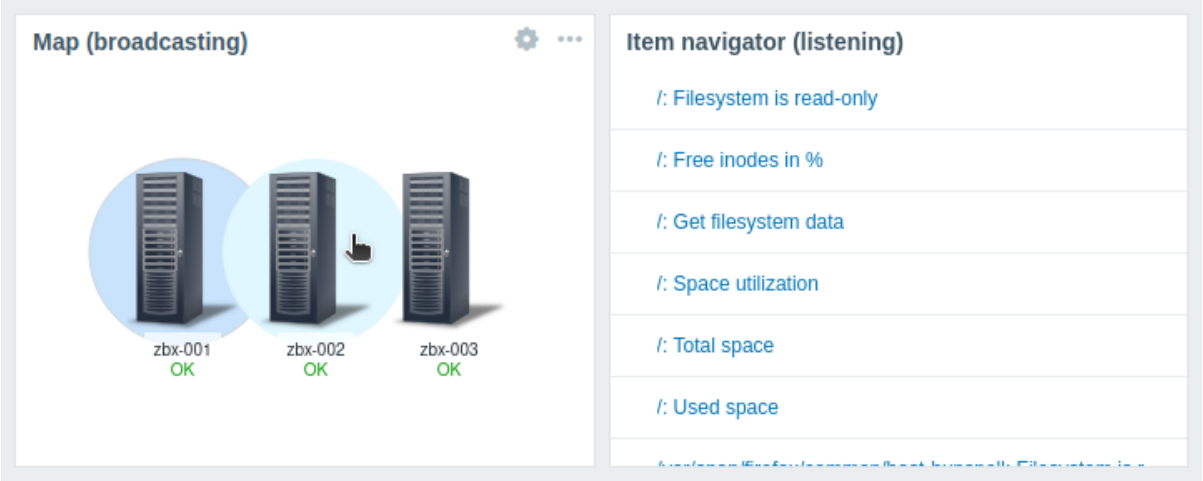
Todos os widgets capazes de transmitir dados começam a fazê-lo automaticamente após a criação. Por exemplo, o widget *Gráfico* transmite imediatamente os dados do período de tempo para os widgets ouvintes.



Widgets que suportam a seleção de entidades (por exemplo, *Navegador de item*) transmitem automaticamente os dados para a primeira entidade disponível. Por exemplo, o widget *Navegador de item* transmite dados para o primeiro item em sua lista de itens. Quando um item diferente é selecionado, ele transmite dados para esse item. Ao passar o mouse, o item é destacado em azul claro; ao selecionar, é destacado em amarelo.



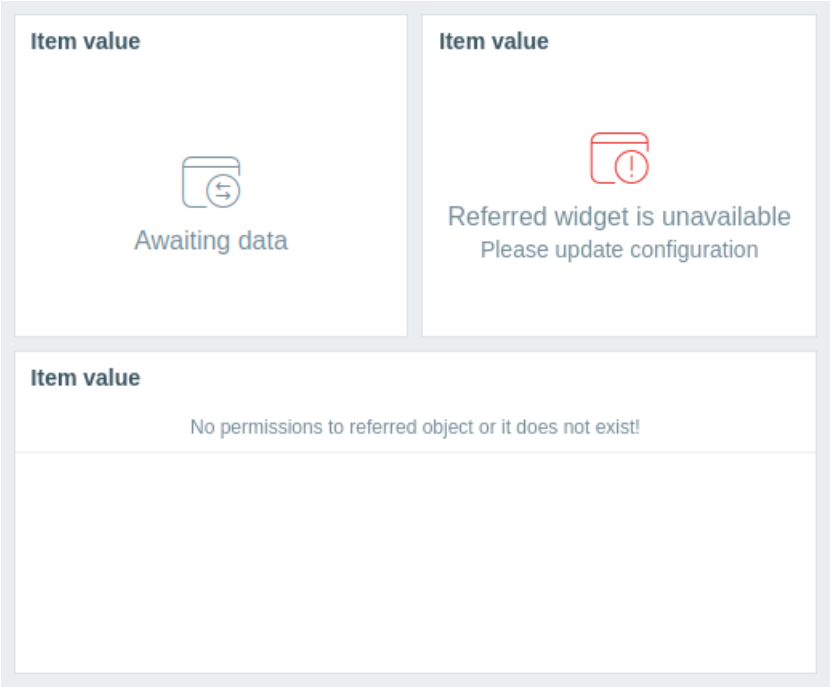
O widget *Mapa* também transmite automaticamente os dados para o primeiro elemento disponível (por exemplo, host), que está mais próximo do canto superior esquerdo da área de visualização do widget (para o widget *Geomapa*, o mais próximo do centro da área de visualização do widget). Quando um elemento diferente é selecionado, ele transmite dados para esse elemento. Ao passar o mouse, o elemento é destacado em azul claro; ao selecionar, é destacado em azul escuro.



Note:
Antes do Zabbix 7.0.7, alguns widgets transmitem dados automaticamente, enquanto outros exigem que uma entidade seja selecionada primeiro. Por exemplo, o widget *Navegador de item* transmite dados do item apenas quando um item específico é selecionado.

Os widgets também têm comportamentos específicos ao **ouvir** dados de outros widgets:

- Se o widget fonte de dados não estiver transmitindo dados, o widget ouvinte entra no estado *Aguardando dados*.
- Se o widget fonte de dados foi excluído, substituído por um widget incompatível ou movido para outra página do dashboard, o widget ouvinte entra no estado *Widget referenciado indisponível*.
- Se o host especificado na fonte de dados (widget ou dashboard) não possui a entidade configurada no widget ouvinte (item, gráfico, mapa, etc.) ou se o usuário não tem permissões para acessar o host, o widget ouvinte exibe a seguinte mensagem: *"Sem permissões para o objeto referenciado ou ele não existe!"*



Para as capacidades de transmissão e escuta de cada widget, consulte Compatibilidade de widgets.

Compatibilidade dos widgets

Alguns widgets podem transmitir dados de configuração para outros widgets, alguns podem escutar dados e alguns podem fazer ambos. Por exemplo:

- O widget *Action log* só pode recuperar dados de período de tempo dos widgets *Graph*, *Graph (classic)* e *Graph prototype*.
- O widget *Geomap* pode transmitir dados de host para widgets que os escutam (*Data overview*, *Honeycomb*, etc.) e também pode escutar dados de grupo de hosts e de host de widgets que os transmitem (*Honeycomb*, *Problem hosts*, etc.).
- O widget *Clock* não pode transmitir nem escutar nenhum dado.

A tabela a seguir descreve as capacidades de transmissão e escuta de cada widget.

Widget	Transmite	Escuta
Action log	-	Período de tempo
Clock	-	-
Data overview	-	Grupos de hosts, Hosts
Discovery status	-	-
Favorite graphs	-	-
Favorite maps	-	-
Gauge	-	Hosts, Items
Geomap	Hosts	Grupos de hosts, Hosts
Graph	Período de tempo	Items, Período de tempo
Graph (classic)	Período de tempo	Hosts, Items, Graphs, Período de tempo
Graph prototype	Período de tempo	Hosts, Período de tempo
Honeycomb	Hosts, Items	Grupos de hosts, Hosts
Host availability	-	Grupos de hosts
Host navigator	Hosts	Grupos de hosts
Item history	Items	Hosts, Período de tempo
Item navigator	Items	Grupos de hosts, Hosts
Item value	-	Hosts, Items, Período de tempo
Map	Grupos de hosts, Hosts	Maps
Map navigation tree	Maps	-
Pie chart	-	Items, Período de tempo
Problem hosts	Grupos de hosts	Grupos de hosts, Hosts

Widget	Transmite	Escuta
Problems	Events	Grupos de hosts, Hosts
Problems by severity	Grupos de hosts	Grupos de hosts, Hosts
SLA report	-	-
System information	-	-
Top hosts	Hosts	Grupos de hosts, Hosts
Top triggers	-	Período de tempo
Trigger overview	-	Grupos de hosts, Hosts
URL	-	Hosts
Web monitoring	Grupos de hosts	Grupos de hosts, Hosts

1 Log de ações

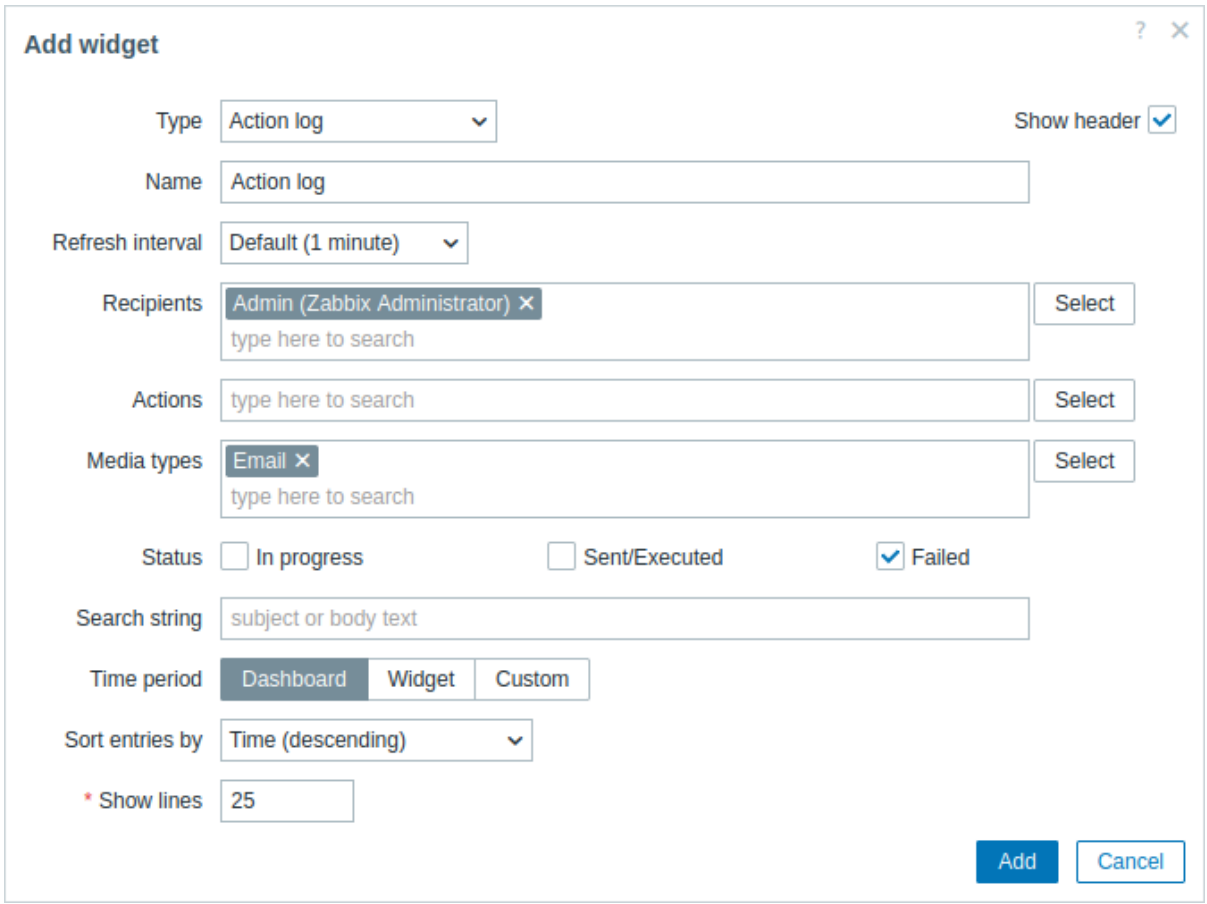
Visão geral

No widget de log de ação, você pode exibir detalhes das operações de ação (notificações, comandos remotos). Ele replica informações de *Relatórios* → *Log de ação*.

Até 1000 registros podem ser exibidos (100 antes do Zabbix 7.0.6).

Configuração

Para configurar, selecione *Action log* como tipo:



Além dos parâmetros que são comuns para todos os widgets, você pode definir as seguintes opções específicas:

Destinatários	Filtra as entradas por destinatários. Este campo é de preenchimento automático, portanto, ao começar a digitar o nome de um destinatário, será exibida uma lista suspensa de destinatários correspondentes. Se nenhum destinatário for selecionado, os detalhes das operações de ação para todos os destinatários serão exibidos.

Ações	Filtra as entradas por ações. Este campo é de preenchimento automático, portanto, ao começar a digitar o nome de uma ação, será exibida uma lista suspensa de ações correspondentes. Se nenhuma ação for selecionada, os detalhes das operações de ação para todas as ações serão exibidos.
Tipos de mídia	Filtra as entradas por tipos de mídia. Este campo é de preenchimento automático, portanto, ao começar a digitar o nome de um tipo de mídia, será exibida uma lista suspensa de tipos de mídia correspondentes. Se nenhum tipo de mídia for selecionado, os detalhes das operações de ação para todos os tipos de mídia serão exibidos.
Status	Marque a caixa de seleção para filtrar as entradas pelo respectivo status: **Em andamento** - são exibidas as operações de ação que estão em andamento; **Enviado/Executado** - são exibidas as operações de ação que enviaram uma notificação ou foram executadas; **Falhou** - são exibidas as operações de ação que falharam.
String de pesquisa	Filtra as entradas pelo conteúdo da mensagem/comando remoto. Se você inserir uma string aqui, somente as operações de ação cuja mensagem/comando remoto contenha a string inserida serão exibidas. Macros não são resolvidas.
Período de tempo	Filtra as entradas por período de tempo. Selecione a fonte de dados para o período de tempo: **Dashboard** - define o seletor *Período de tempo* como fonte de dados; **Widget** - define um widget compatível especificado no parâmetro *Widget* como fonte de dados; **Personalizado** - define o período de tempo especificado nos parâmetros *De* e *Até* como fonte de dados; se definido, um ícone de relógio será exibido no canto superior direito do widget, indicando o tempo definido ao passar o mouse.
Widget	Digite ou selecione um widget compatível como fonte de dados para o período de tempo. Este parâmetro está disponível se *Período de tempo* estiver definido como ”Widget”.
De	Digite ou selecione o início do período de tempo. A sintaxe de tempo relativo (`now`, `now/d`, `now/w-1w`, etc.) é suportada. Este parâmetro está disponível se *Período de tempo* estiver definido como ”Personalizado”.
Até	Digite ou selecione o final do período de tempo. A sintaxe de tempo relativo (`now`, `now/d`, `now/w-1w`, etc.) é suportada. Este parâmetro está disponível se *Período de tempo* estiver definido como ”Personalizado”.
Ordenar entradas por	Ordena as entradas por: **Hora** (decrescente ou crescente); **Tipo** (decrescente ou crescente); **Status** (decrescente ou crescente); **Destinatário** (decrescente ou crescente).
Exibir linhas	Define quantas linhas do log de ações serão exibidas no widget.

2 Relógio

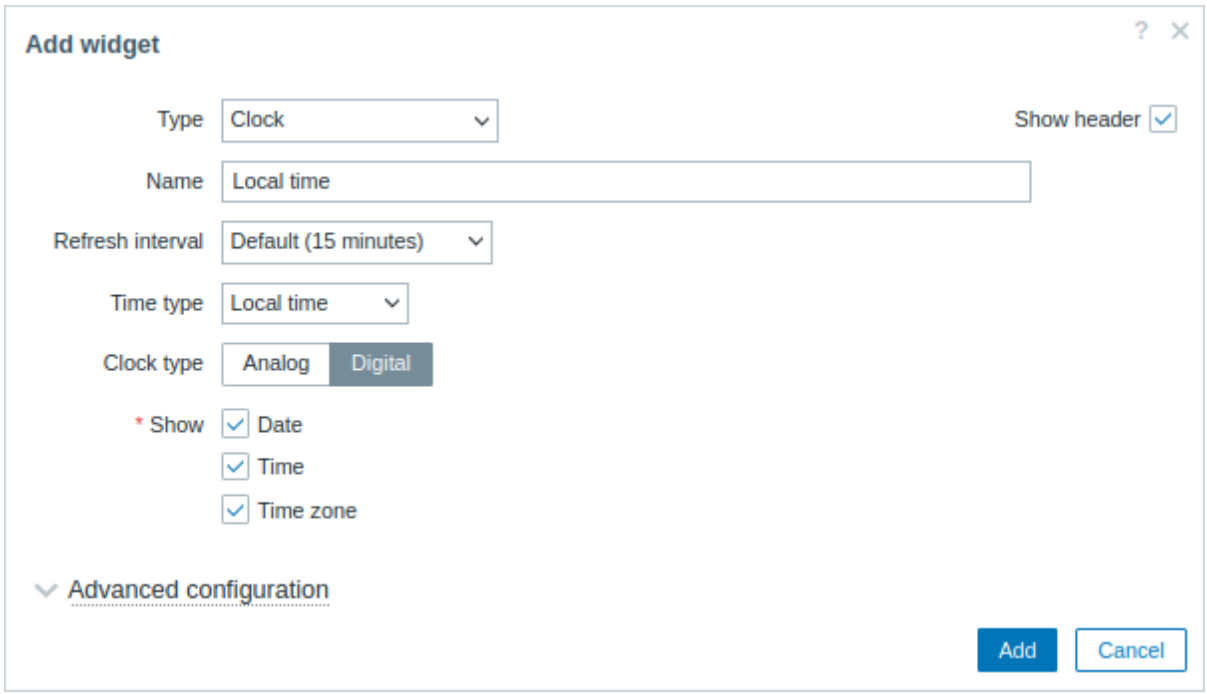
Visão geral

No widget de relógio, você pode exibir a hora local, do server ou de um host especificado.

Relógios analógicos e digitais podem ser exibidos:

Configuração

Para configurar, selecione *Relógio* como tipo:



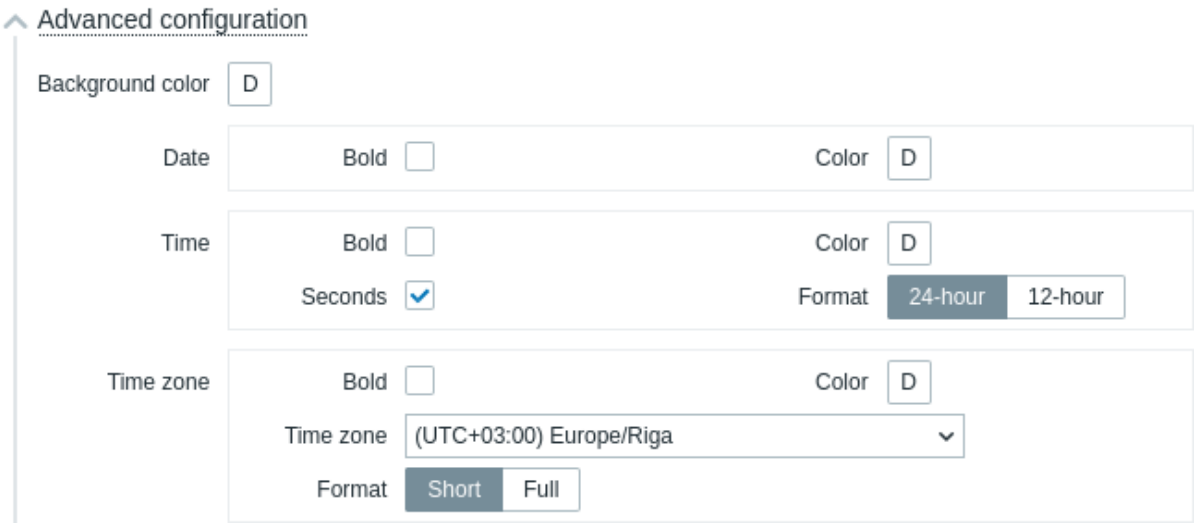
Além dos parâmetros que são comuns para todos os widgets, você pode definir as seguintes opções específicas:

Tipo de hora	Selecione hora local, do servidor ou de um host especificado. A hora do servidor será idêntica ao fuso horário definido globalmente ou para o usuário Zabbix.
Item	Selecione o item para exibir a hora. Para exibir a hora do host, use o item `system.localtime[local]`. Este item deve existir no host. Este campo está disponível apenas quando *Hora do host* é selecionado.
Tipo de relógio	Selecione o tipo de relógio: **Analógico** - relógio analógico **Digital** - relógio digital
Mostrar	Selecione as unidades de informação (data, hora, fuso horário) para exibir no relógio digital. O tamanho da unidade é dinâmico, preenchendo todo o espaço disponível do widget com base nas unidades exibidas, no tamanho do widget e na escala de exibição. Este campo está disponível apenas se "Digital" for selecionado no campo *Tipo de relógio*; pelo menos um tipo de unidade de informação deve ser selecionado.
Configuração avançada	Clique no rótulo *Configuração avançada* para exibir as opções de configuração avançada para o relógio digital. Esta seção está disponível apenas se "Digital" for selecionado no campo *Tipo de relógio*.

Configuração avançada

As opções de configuração avançada estão disponíveis na seção recolhível *Configuração avançada*, e apenas para os elementos que estão selecionados no campo *Mostrar* (veja acima).

Além disso, a configuração avançada permite alterar a cor de fundo de todo o widget.



Cor de fundo	Selecione a cor de fundo no seletor de cores. D significa cor padrão (depende do tema do frontend). Para retornar ao valor padrão, clique no botão *Usar padrão* no seletor de cores.
Data	
Negrito	Marque a caixa de seleção para exibir a data em negrito.
Cor	Selecione a cor da data no seletor de cores. D significa cor padrão (depende do tema do frontend). Para retornar ao valor padrão, clique no botão *Usar padrão* no seletor de cores.
Hora	
Negrito	Marque a caixa de seleção para exibir a hora em negrito.
Cor	Selecione a cor da hora no seletor de cores. D significa cor padrão (depende do tema do frontend). Para retornar ao valor padrão, clique no botão *Usar padrão* no seletor de cores.
Segundos	Marque a caixa de seleção para exibir os segundos. Caso contrário, apenas horas e minutos serão exibidos.
Formato	Selecione para exibir a hora em formato de 24 horas ou 12 horas.
Fuso horário	
Negrito	Marque a caixa de seleção para exibir o fuso horário em negrito.
Cor	Selecione a cor do fuso horário no seletor de cores. D significa cor padrão (depende do tema do frontend). Para retornar ao valor padrão, clique no botão *Usar padrão* no seletor de cores.

Fuso horário	Selecione o fuso horário.
Formato	Selecione para exibir o fuso horário em formato curto (por exemplo, `New York`) ou formato completo (por exemplo, `(UTC-04:00) America/New York`).

Attention:
Este widget está obsoleto e será removido na próxima versão principal. Considere usar o widget *Top hosts* em seu lugar.

Visão geral

No widget de visão geral dos dados, você pode exibir os dados mais recentes para um grupo de hosts.

A cor dos itens de problema é baseada na cor de severidade do problema, que pode ser ajustada na tela de atualização de problema.

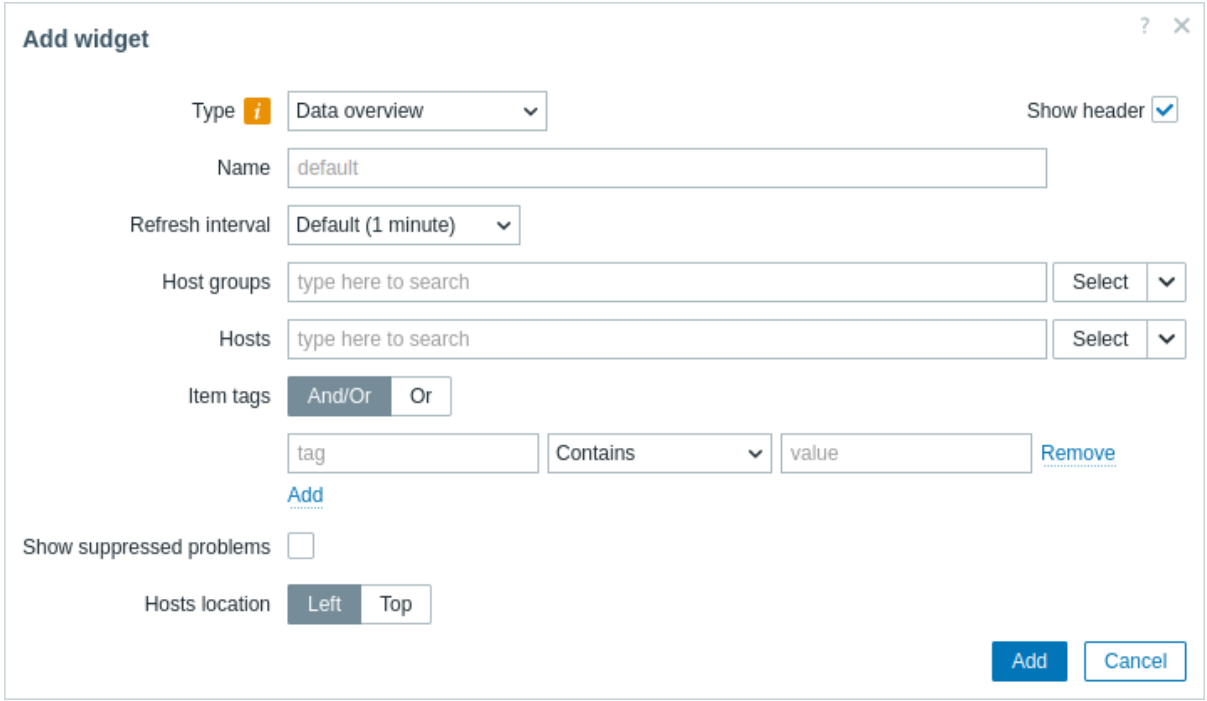
Por padrão, apenas os valores que se enquadram nas últimas 24 horas são exibidos. Esse limite foi introduzido com o objetivo de melhorar o tempo de carregamento inicial para páginas grandes de dados mais recentes. Esse limite é configurável em *Administração → Geral → GUI*, usando a opção *Período máximo de exibição do histórico*.

Clicar em um dado oferece links para alguns gráficos predefinidos ou valores mais recentes.

Observe que 50 registros são exibidos por padrão (configurável em *Administração → Geral → GUI*, usando a opção *Número máximo de colunas e linhas nas tabelas de visão geral*). Se existirem mais registros do que o configurado para exibir, uma mensagem será exibida na parte inferior da tabela, solicitando critérios de filtragem mais específicos. Não há paginação. Observe que esse limite é aplicado primeiro, antes de qualquer filtragem adicional dos dados, por exemplo, por tags.

Configuração

Para configurar, selecione *Visão geral dos dados* como tipo:



Além dos parâmetros que são comuns para todos os widgets, você pode definir as seguintes opções específicas:

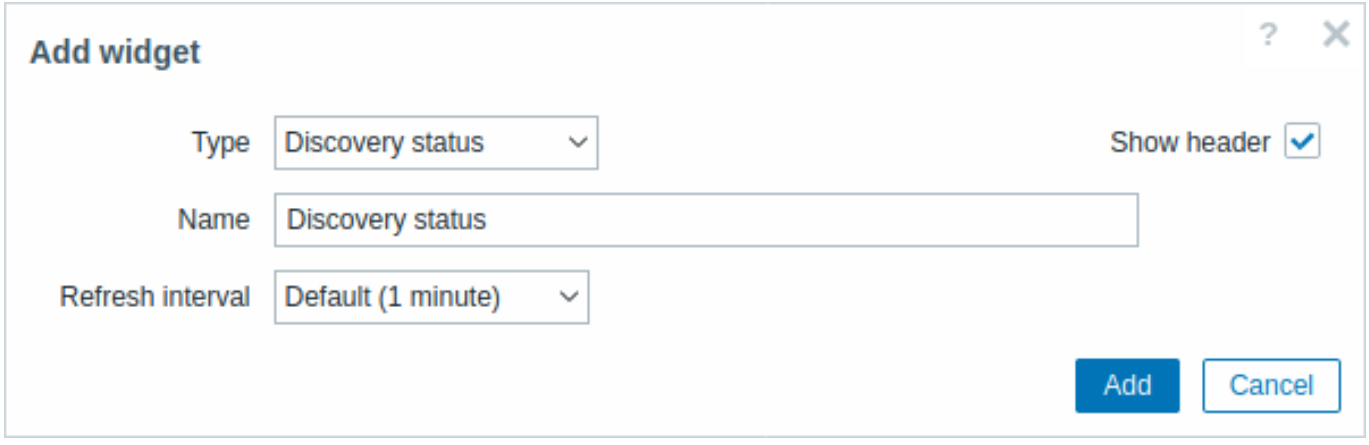
Grupos de hosts	Selecione grupos de hosts. Como alternativa, selecione um widget compatível como fonte de dados para grupos de hosts. Este campo é de preenchimento automático, portanto, ao começar a digitar o nome de um grupo, será exibida uma lista suspensa de grupos correspondentes. Este parâmetro não está disponível ao configurar o widget em um dashboard de template.

Hosts	Selecione hosts. Como alternativa, selecione um widget compatível ou o dashboard como fonte de dados para hosts. Este campo é de preenchimento automático, portanto, ao começar a digitar o nome de um host, será exibida uma lista suspensa de hosts correspondentes. Este parâmetro não está disponível ao configurar o widget em um dashboard de template.
Tags de item	Especifique tags para limitar a quantidade de dados de item exibidos no widget. É possível incluir ou excluir tags e valores de tags específicos. Várias condições podem ser definidas. A correspondência do nome da tag diferencia maiúsculas de minúsculas. Há vários operadores disponíveis para cada condição: **Existe** - inclui os nomes de tags especificados; **Igual** - inclui os nomes e valores de tags especificados (diferencia maiúsculas de minúsculas); **Contém** - inclui os nomes de tags especificados onde os valores das tags contêm a string inserida (correspondência de substring, não diferencia maiúsculas de minúsculas); **Não existe** - exclui os nomes de tags especificados; **Diferente** - exclui os nomes e valores de tags especificados (diferencia maiúsculas de minúsculas); **Não contém** - exclui os nomes de tags especificados onde os valores das tags contêm a string inserida (correspondência de substring, não diferencia maiúsculas de minúsculas). Existem dois tipos de cálculo para as condições: **E/Ou** - todas as condições devem ser atendidas, condições com o mesmo nome de tag serão agrupadas pela condição *Ou*; **Ou** - basta que uma condição seja atendida.
Mostrar problemas suprimidos	Marque a caixa de seleção para exibir problemas que, de outra forma, seriam suprimidos (não exibidos) devido à manutenção do host.
Localização dos hosts	Selecione a localização dos hosts - esquerda ou topo. Este parâmetro é rotulado como *Localização do host* ao configurar o widget em um dashboard de template.

4 Status da descoberta

Visão geral

Este widget exibe um resumo do status das regras de descoberta de rede ativas.



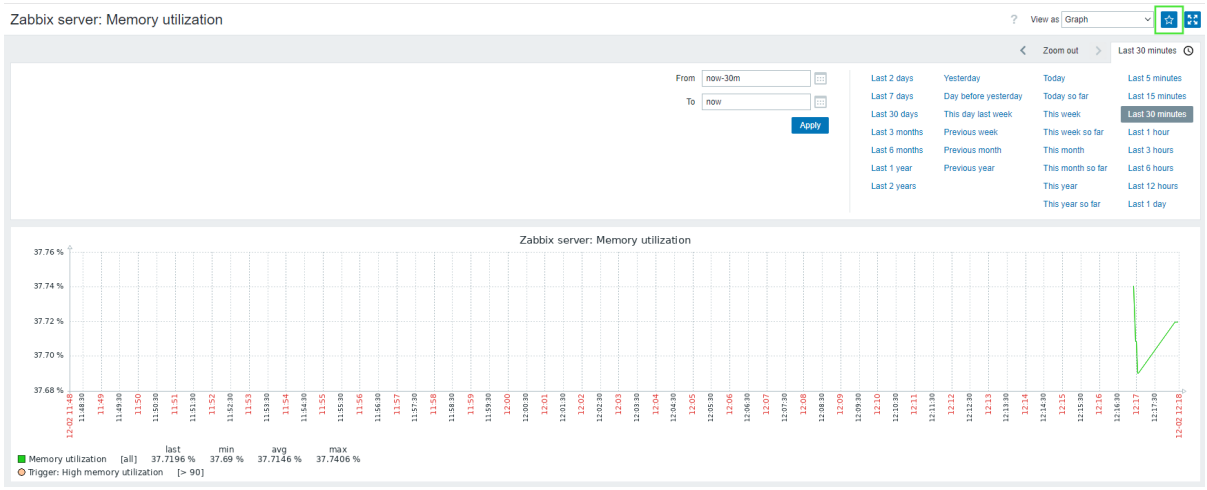
Todos os parâmetros de configuração são comuns para todos os widgets.

5 Gráficos favoritos

Visão geral

Este widget contém atalhos para os gráficos mais necessários, classificados em ordem alfabética.

A lista de atalhos é preenchida quando você visualiza um gráfico em Monitoramento -> Últimos dados -> Gráficos e, em seguida, clica no botão *Adicionar aos favoritos*.



Todos os parâmetros de configuração são comuns para todos os widgets.

6 Mapas favoritos

Visão geral

Este widget contém atalhos para os mapas mais necessários, classificados alfabeticamente.

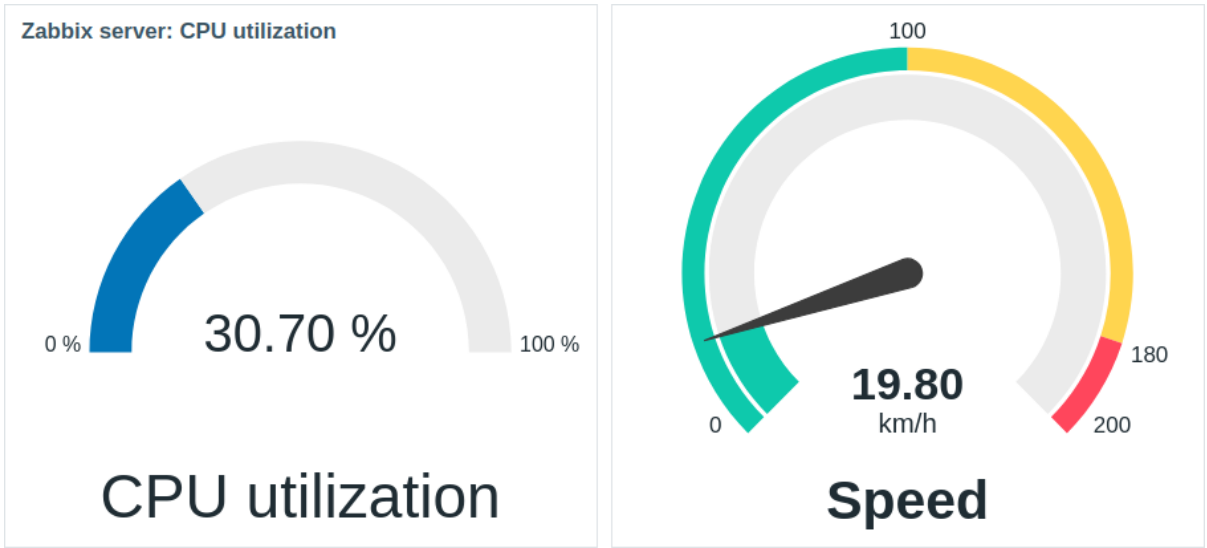
A lista de atalhos é preenchida quando você visualiza um mapa e, em seguida, clica no botão *Adicionar aos favoritos*.

Todos os parâmetros de configuração são comuns para todos os widgets.

7 Gauge

Visão geral

O widget de medidor exibe o valor de um único item como um medidor.

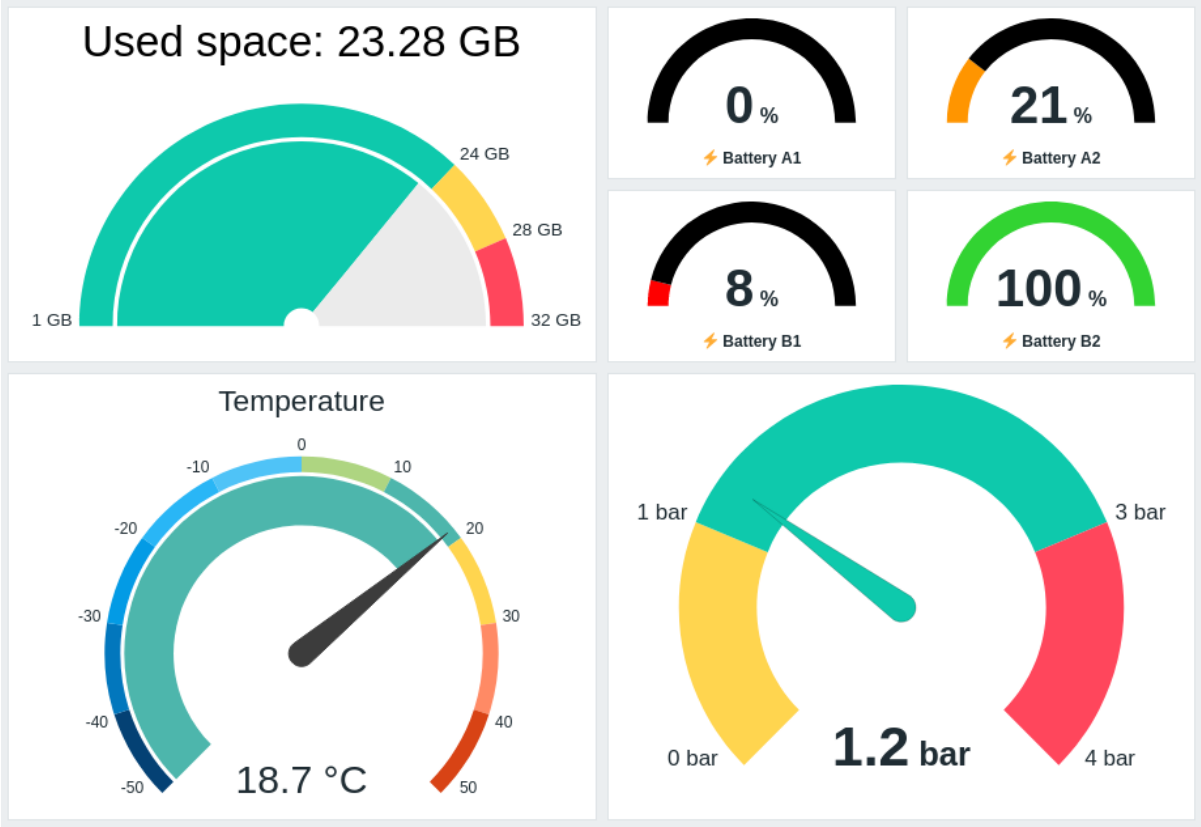


Quando configurado, o widget pode consistir nos seguintes elementos:

- Descrição do item (por exemplo, ”Utilização da CPU”, ”Velocidade”)
- Valor do item (por exemplo, ”30.70”, ”19.80”)
- Unidades do valor do item (por exemplo, ”%”, ”km/h”)
- Escala do medidor (por exemplo, ”0%/100%”, ”0/100/180/200”)
- Arco do medidor (arco do valor do medidor e arco dos limites do medidor)

- Ponteiro do medidor

O widget pode ser ajustado visualmente usando as opções de configuração avançada para criar uma grande variedade de estilos visuais:

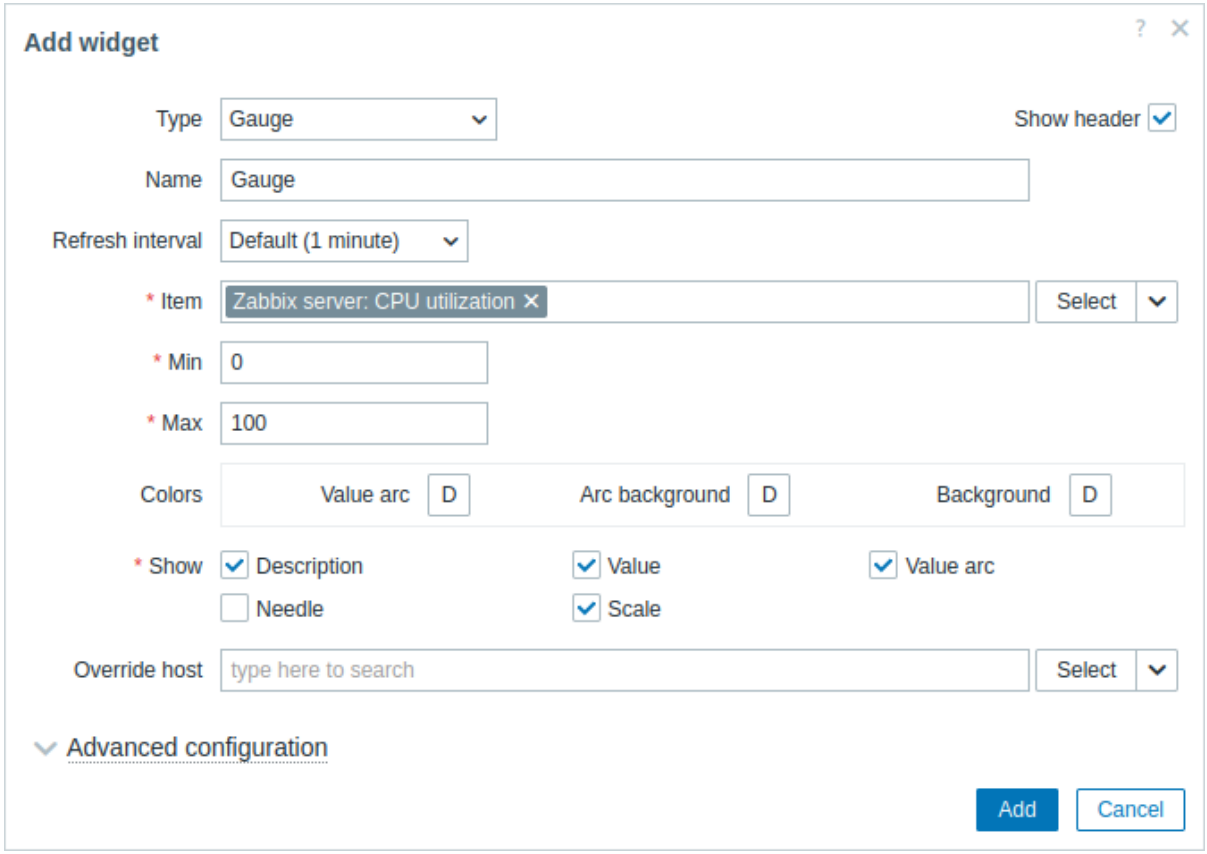


O widget de medidor pode exibir apenas valores numéricos. A exibição de valores binários não é suportada.

Clicar no widget de medidor leva a um gráfico para o item.

Configuração

Para configurar, selecione *Gauge* como tipo:

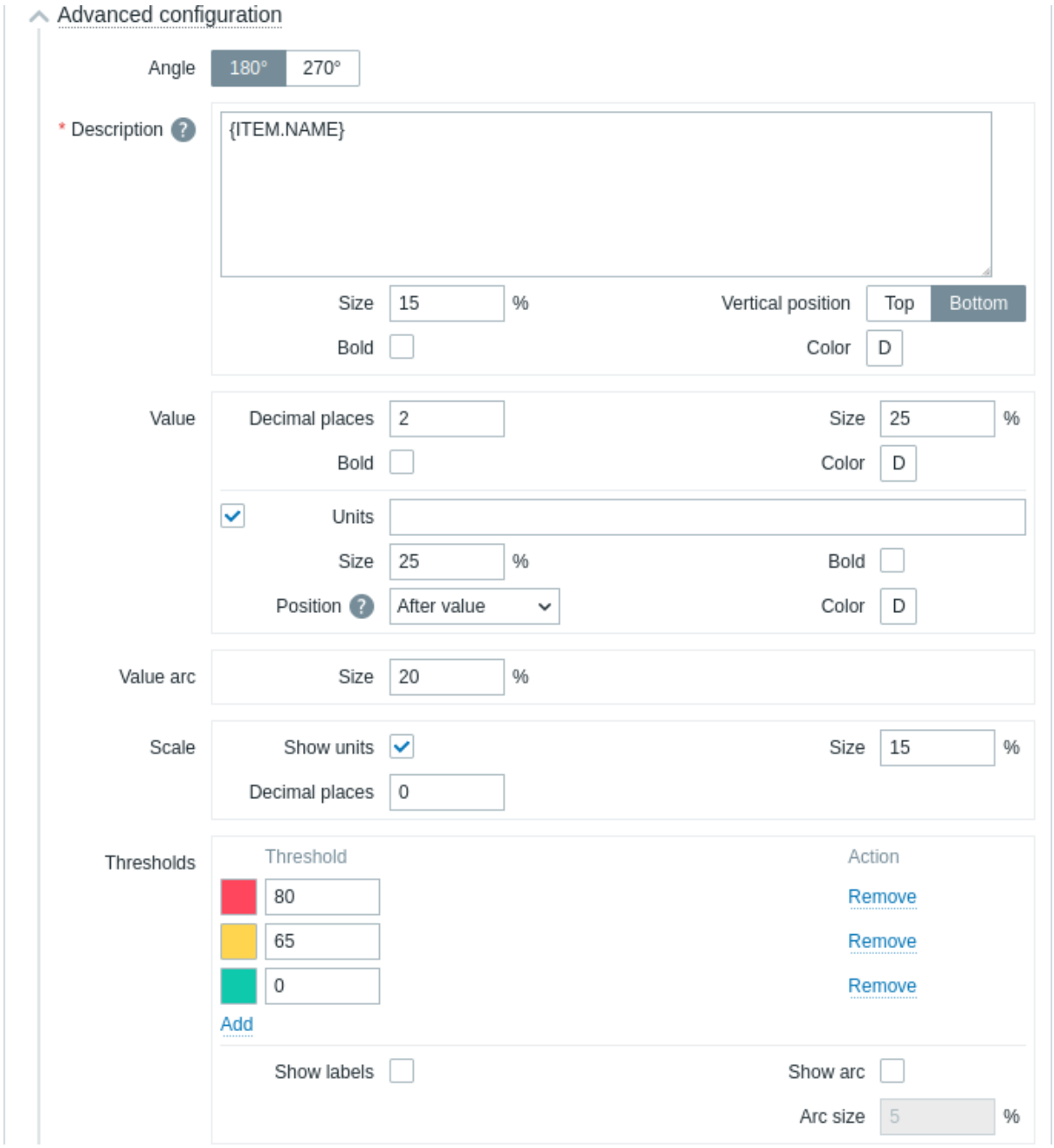


Além dos parâmetros que são comuns para todos os widgets, você pode definir as seguintes opções específicas:

Item	Selecione o item. Como alternativa, selecione um widget compatível como fonte de dados para os itens. Este campo possui preenchimento automático, então ao começar a digitar o nome de um item será exibida uma lista suspensa com os itens correspondentes. Observe que você só pode selecionar itens que retornam dados numéricos.
Min	Insira o valor mínimo do gauge. Sufixos (por exemplo, ”1d”, ”2w”, ”4K”, ”8G”) são suportados. Mapeamentos de valor são suportados.
Max	Insira o valor máximo do gauge. Sufixos (por exemplo, ”1d”, ”2w”, ”4K”, ”8G”) são suportados. Mapeamentos de valor são suportados.
Cores	Selecione a cor no seletor de cores: **Arco de valor** - selecione a cor do arco de valor do gauge; **Fundo do arco** - selecione a cor de fundo do arco de valor do gauge e do arco de limiares do gauge; **Fundo** - selecione a cor de fundo do widget. ”D” significa a cor padrão, que depende do tema do frontend. Se os *Limiares* estiverem definidos, a cor padrão para **Arco de valor** depende da cor do limiar. Para retornar à cor padrão, clique no botão *Usar padrão* no seletor de cores.
Mostrar	Marque a caixa de seleção para exibir o respectivo elemento do gauge - descrição, valor, arco de valor, ponteiro, escala (o valor mínimo e máximo do gauge no início e no final do arco do gauge). Desmarque para ocultar. Pelo menos um elemento deve ser selecionado. Observe que o ponteiro e a escala do gauge podem ser exibidos se o arco de valor do gauge ou o arco de limiares do gauge (veja as opções de configuração avançada) estiverem exibidos. Observe também que, se o ponteiro do gauge estiver exibido, o valor será colocado sob o ponteiro; se o ponteiro estiver oculto, o valor será alinhado com a parte inferior do arco do gauge.
Substituir host	Selecione um widget compatível ou o dashboard como fonte de dados para hosts. Este parâmetro não está disponível ao configurar o widget em um dashboard de template.
Configuração avançada	Clique no rótulo *Configuração avançada* para exibir as opções de configuração avançada. É aqui também que você poderá ajustar os elementos do gauge selecionados no campo *Mostrar*.

Configuração avançada

As opções de configuração avançada estão disponíveis na seção recolhível *Configuração avançada*:



Ângulo	Selecione o ângulo do gauge (180° ou 270°).
Descrição	
Descrição	Insira a descrição do item. Esta descrição pode substituir o nome padrão do item. Descrições multilinha são suportadas. É possível uma combinação de texto e macros suportadas. {HOST.*}, {ITEM.*}, {INVENTORY.*} e macros de usuário são suportadas.
Tamanho	Insira a altura do tamanho da fonte para a descrição do item (em porcentagem, em relação à altura total do widget).
Posição vertical	Selecione a posição vertical da descrição do item (superior ou inferior, em relação ao arco do gauge).
Negrito	Marque a caixa de seleção para exibir a descrição do item em negrito.
Cor	Selecione a cor da descrição do item no seletor de cores. ”D” significa a cor padrão, que depende do tema do frontend. Para retornar à cor padrão, clique no botão *Usar padrão* no seletor de cores.
Valor	
Casas decimais	Insira o número de casas decimais a serem exibidas com o valor. Esta opção afeta apenas itens que retornam dados numéricos (float).
Tamanho	Insira a altura do tamanho da fonte para o valor (em porcentagem, em relação à altura do arco do gauge).
Negrito	Marque a caixa de seleção para exibir o valor em negrito.

Cor	Selecione a cor do valor no seletor de cores. ”D” significa a cor padrão, que depende do tema do frontend. Para retornar à cor padrão, clique no botão *Usar padrão* no seletor de cores.
Unidades	
Unidades	Marque a caixa de seleção para exibir as unidades com o valor do item. Se você inserir um nome de unidade, ele substituirá as unidades definidas na configuração do item.
Tamanho	Insira a altura do tamanho da fonte para as unidades do item (em porcentagem, em relação à altura do arco do gauge).
Negrito	Marque a caixa de seleção para exibir as unidades do item em negrito.
Posição	Selecione a posição das unidades do item (acima, abaixo, antes ou depois, em relação ao valor do item). Esta opção é ignorada para as seguintes unidades relacionadas ao tempo: unixtime, uptime, s.
Cor	Selecione a cor das unidades do item no seletor de cores. ”D” significa a cor padrão, que depende do tema do frontend. Para retornar à cor padrão, clique no botão *Usar padrão* no seletor de cores.
Arco de valor	
Tamanho do arco	Insira a altura do tamanho do arco de valor do gauge (em porcentagem, em relação ao raio do arco do gauge).
Agulha	
Cor	Selecione a cor da agulha do gauge no seletor de cores. ”D” significa a cor padrão, que depende do tema do frontend. Se *Limiares* estiverem definidos, a cor padrão da agulha depende da cor do limiar. Para retornar à cor padrão, clique no botão *Usar padrão* no seletor de cores.
Escala	
Exibir unidades	Marque a caixa de seleção para exibir as unidades com o valor mínimo e máximo do gauge.
Tamanho	Insira a altura do tamanho da fonte para o valor mínimo e máximo do gauge (em porcentagem, em relação à altura do arco do gauge).
Casas decimais	Insira o número de casas decimais a serem exibidas com o valor mínimo e máximo do gauge. Esta opção afeta apenas itens que retornam dados numéricos (float).
Limiares	
Limiares	Clique em *Adicionar* para adicionar um limiar, selecione uma cor de limiar no seletor de cores e especifique um valor numérico. A lista de limiares será classificada em ordem crescente ao salvar. Observe que as cores configuradas como limiares serão exibidas corretamente apenas para itens numéricos. Sufixos (por exemplo, ”1d”, ”2w”, ”4K”, ”8G”) são suportados. Mapeamentos de valores são suportados.
Exibir rótulos	Marque a caixa de seleção para exibir os valores dos limiares como rótulos na escala do gauge.
Exibir arco	Marque a caixa de seleção para exibir o arco de limiares do gauge.
Tamanho do arco	Insira a altura do tamanho do arco de limiares do gauge (em porcentagem, em relação ao raio do arco do gauge).

As informações exibidas pelo widget de medidor podem ser baixadas como uma imagem *.png* usando o menu do widget:

Uma captura de tela do widget será salva na pasta *Downloads*.

8 Geomap

Visão geral

O widget de mapa geográfico exibe hosts como marcadores em um mapa geográfico usando a biblioteca de mapas interativos JavaScript de código aberto Leaflet.

> **Note:**
> O Zabbix oferece vários provedores de serviço de tiles de mapa predefinidos e uma opção para adicionar um provedor de serviço de tiles personalizado ou até mesmo hospedar os tiles (configurável em *Administração > Geral > Mapas geográficos* seção do menu).

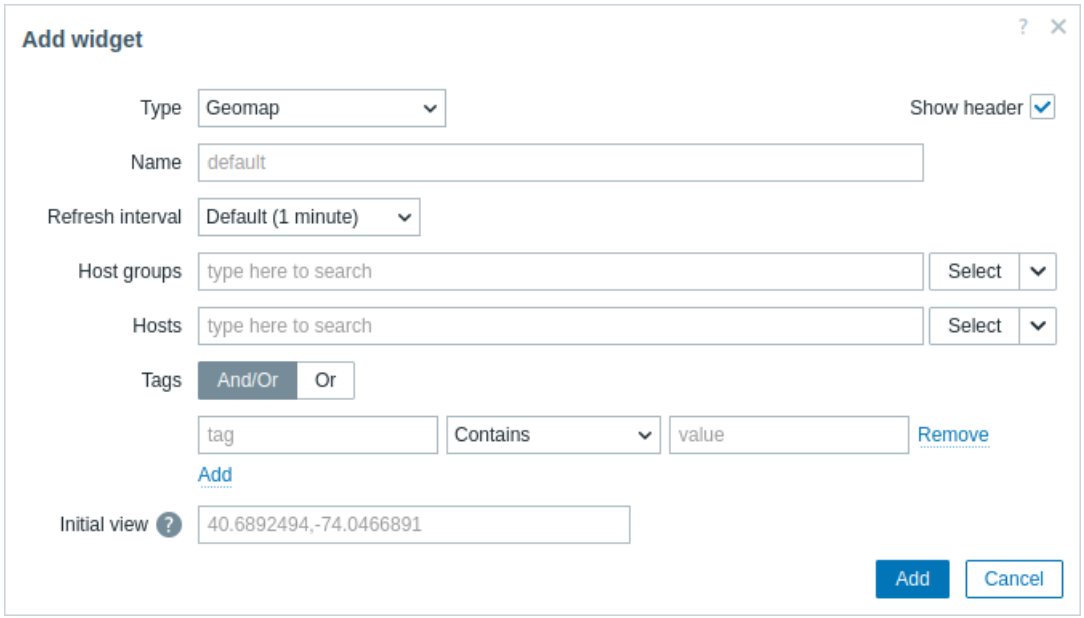
Por padrão, o widget exibe todos os hosts habilitados que possuem coordenadas válidas definidas em seu inventário. É possível configurar o filtro de hosts nos parâmetros do widget.

As coordenadas válidas do host são:

- Latitude: de -90 a 90 (pode ser um número inteiro ou decimal)
- Longitude: de -180 a 180 (pode ser um número inteiro ou decimal)

Configuração

Para adicionar o widget, selecione *Geomap* como tipo.



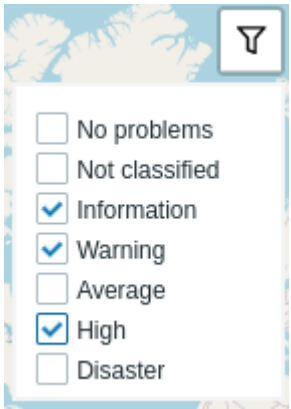
Além dos parâmetros que são comuns para todos os widgets, você pode definir as seguintes opções específicas:

Grupos de hosts	Selecione os grupos de hosts a serem exibidos no mapa. Como alternativa, selecione um widget compatível como fonte de dados para grupos de hosts. Este campo possui preenchimento automático, portanto, ao começar a digitar o nome de um grupo, será exibida uma lista suspensa com os grupos correspondentes. Se nada for selecionado nos campos *Grupos de hosts* e *Hosts*, todos os hosts com coordenadas válidas serão exibidos. Este parâmetro não está disponível ao configurar o widget em um dashboard de template.
Hosts	Selecione os hosts a serem exibidos no mapa. Como alternativa, selecione um widget compatível ou o dashboard como fonte de dados para hosts. Este campo possui preenchimento automático, portanto, ao começar a digitar o nome de um host, será exibida uma lista suspensa com os hosts correspondentes. Se nada for selecionado nos campos *Grupos de hosts* e *Hosts*, todos os hosts com coordenadas válidas serão exibidos. Este parâmetro não está disponível ao configurar o widget em um dashboard de template.

Tags	Especifique tags para limitar o número de hosts exibidos no widget. É possível incluir ou excluir tags e valores de tags específicos. Várias condições podem ser definidas. A correspondência do nome da tag diferencia maiúsculas de minúsculas. Existem vários operadores disponíveis para cada condição: **Existe** - inclui os nomes de tags especificados; **Igual** - inclui os nomes e valores de tags especificados (diferencia maiúsculas de minúsculas); **Contém** - inclui os nomes de tags especificados cujos valores de tag contenham a string inserida (correspondência de substring, não diferencia maiúsculas de minúsculas); **Não existe** - exclui os nomes de tags especificados; **Diferente** - exclui os nomes e valores de tags especificados (diferencia maiúsculas de minúsculas); **Não contém** - exclui os nomes de tags especificados cujos valores de tag contenham a string inserida (correspondência de substring, não diferencia maiúsculas de minúsculas). Existem dois tipos de cálculo para as condições: **E/Ou** - todas as condições devem ser atendidas, condições com o mesmo nome de tag serão agrupadas pela condição *Ou*; **Ou** - basta que uma condição seja atendida.
Visualização inicial	Este parâmetro não está disponível ao configurar o widget em um dashboard de template. Coordenadas centrais separadas por vírgula e um nível de zoom opcional para exibir quando o widget for carregado inicialmente no formato `<latitude>,<longitude>,<zoom>` Se o zoom inicial for especificado, o widget Geomap será carregado no nível de zoom fornecido. Caso contrário, o zoom inicial será calculado como metade do zoom máximo para o provedor de mapas específico. A visualização inicial será ignorada se a visualização padrão estiver definida (veja abaixo). Exemplos: 40.6892494,-74.0466891 40.6892494,-122.0466891

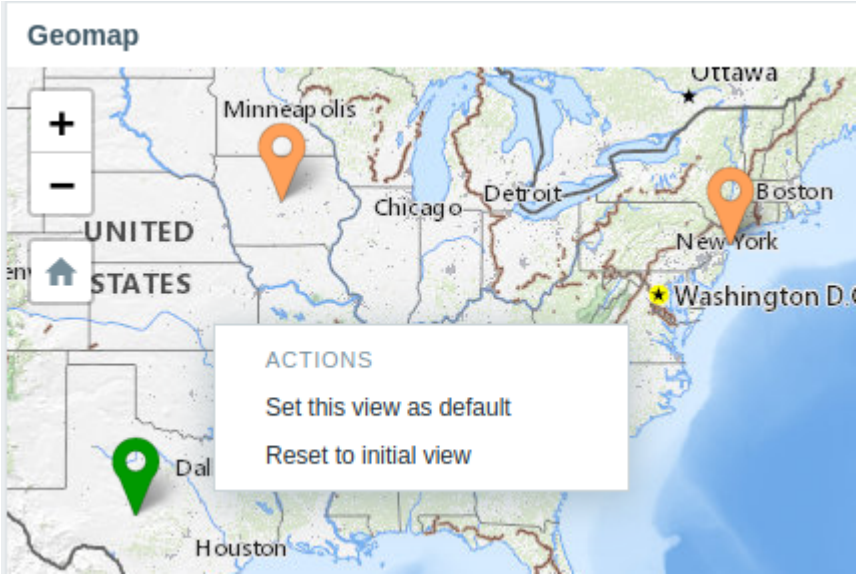
Os marcadores de host exibidos no mapa têm a cor do problema mais grave do host e a cor verde se um host não tiver problemas. Clicar em um marcador de host permite visualizar o nome visível do host e o número de problemas não resolvidos agrupados por gravidade. Clicar no nome visível abrirá o menu do host.

Os hosts exibidos no mapa podem ser filtrados por gravidade do problema. Pressione o ícone de filtro no canto superior direito do widget e marque as gravidades necessárias.



É possível aumentar e diminuir o zoom do mapa usando os botões de mais e menos no canto superior esquerdo do widget ou usando a roda de rolagem do mouse ou o touchpad. Para definir a visualização atual como padrão, clique com o botão direito em qualquer lugar do mapa e selecione *Definir esta visualização como padrão*. Essa configuração substituirá o parâmetro *Visualização inicial* do widget para o usuário atual. Para desfazer essa ação, clique com o botão direito novamente em qualquer lugar do mapa e selecione *Redefinir para visualização inicial*.

Quando *Visualização inicial* ou *Visualização padrão* estiver definida, você pode retornar a essa visualização a qualquer momento pressionando o ícone de casa à esquerda.



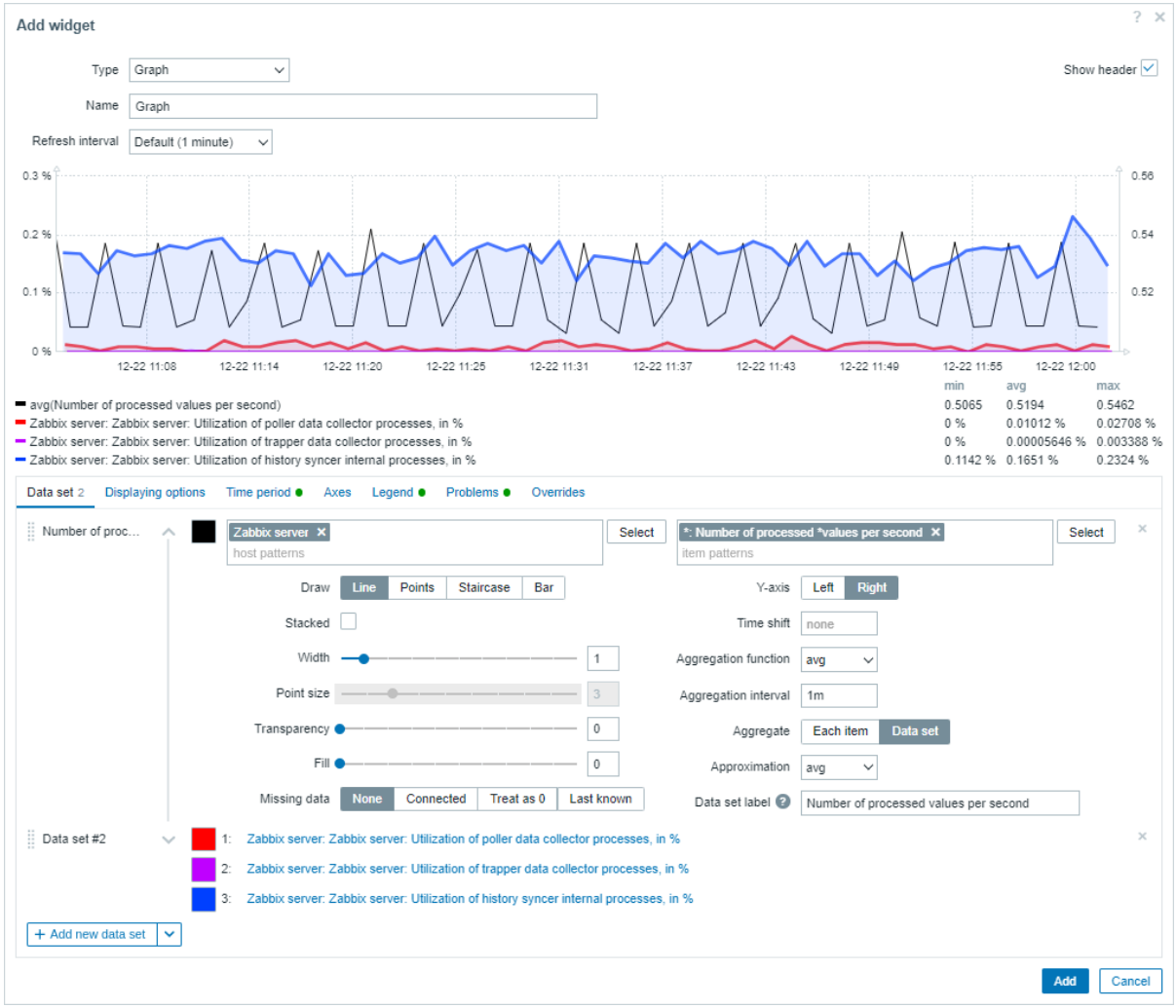
9 Gráfico

Visão geral

O widget de gráfico fornece uma maneira moderna e versátil de visualizar dados coletados pelo Zabbix usando uma técnica de desenho de imagem vetorial. Este widget de gráfico é suportado desde o Zabbix 4.0. Observe que o widget de gráfico suportado antes do Zabbix 4.0 ainda pode ser usado como Gráfico (clássico). Veja também a seção *Adicionando widgets* na página *Dashboards* para mais detalhes.

Configuração

Para configurar, selecione *Gráfico* como tipo:



Conjunto de dados

A aba **Conjunto de dados** permite selecionar dados para o gráfico adicionando conjuntos de dados. Dois tipos de conjuntos de dados podem ser adicionados:

- *Padrões de item* - os dados dos itens correspondentes são exibidos. O gráfico é desenhado usando diferentes tons de uma única cor para cada item.
- *Lista de itens* - os dados dos itens selecionados são exibidos. O gráfico é desenhado usando cores diferentes para cada item.

Por padrão, um conjunto de dados *Padrões de item* é adicionado.

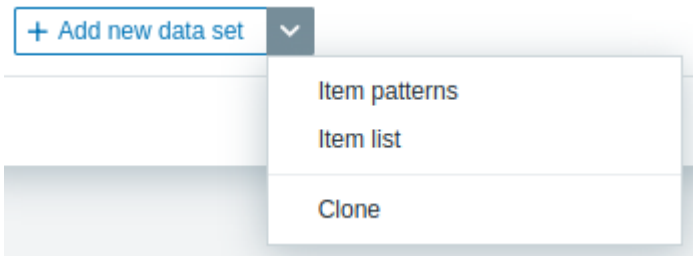
Conjunto de dados	Para o conjunto de dados **Padrões de item**: Selecione ou insira padrões de host e de item; os dados dos itens que correspondem a esses padrões serão exibidos no gráfico; até 50 itens podem ser exibidos. Padrões curinga podem ser usados para seleção (por exemplo, ∗ retornará resultados que correspondam a zero ou mais caracteres). Para especificar um padrão curinga, insira a string manualmente e pressione *Enter*. O símbolo curinga é sempre interpretado, portanto, não é possível adicionar, por exemplo, um item chamado *item** individualmente se houver outros itens correspondentes (por exemplo, *item2*, *item3*). Especificar padrões de host e de item é obrigatório para conjuntos de dados "Padrões de item". Veja também: Detalhes da configuração do conjunto de dados. Para o conjunto de dados **Lista de itens**: Selecione os itens para o gráfico clicando no botão *Adicionar item*. Desde o Zabbix 7.0.1, você também pode selecionar widgets compatíveis como fonte de dados para itens clicando no botão *Adicionar widget*. Especificar itens ou widgets é obrigatório para conjuntos de dados "Lista de itens". Veja também: Detalhes da configuração do conjunto de dados. Observe que apenas tipos de item numéricos são permitidos. Ao configurar o widget em um dashboard de template, o parâmetro para especificar padrões de host não está disponível, e o parâmetro para especificar uma lista de itens permite selecionar apenas os itens configurados no template.
Desenhar	Escolha o tipo de desenho da métrica. Os tipos de desenho possíveis são: *Linha* (definido por padrão), *Pontos*, *Degrau* e *Barra*. Observe que, se houver apenas um ponto de dados no gráfico de linha/degrau, ele será desenhado como um ponto, independentemente do tipo de desenho. O tamanho do ponto é calculado a partir da largura da linha, mas não pode ser menor que 3 pixels, mesmo que a largura da linha seja menor.
Empilhado	Marque a caixa de seleção para exibir os dados como empilhados (áreas preenchidas exibidas). Esta opção é desabilitada quando o tipo de desenho *Pontos* é selecionado.
Largura	Defina a largura da linha. Esta opção está disponível quando o tipo de desenho *Linha* ou *Degrau* é selecionado.
Tamanho do ponto	Defina o tamanho do ponto. Esta opção está disponível quando o tipo de desenho *Pontos* é selecionado.
Transparência	Defina o nível de transparência.
Preenchimento	Defina o nível de preenchimento. Esta opção está disponível quando o tipo de desenho *Linha* ou *Degrau* é selecionado.
Dados ausentes	Selecione a opção para exibir dados ausentes: **Nenhum** - a lacuna é deixada vazia; **Conectado** - dois valores de borda são conectados; **Tratar como 0** - os dados ausentes são exibidos como valores 0; **Último conhecido** - os dados ausentes são exibidos com o mesmo valor que o último valor conhecido; não aplicável para os tipos de desenho *Pontos* e *Barra*.
Eixo Y	Selecione o lado do gráfico onde o eixo Y será exibido.
Deslocamento de tempo	Especifique o deslocamento de tempo, se necessário. Você pode usar sufixos de tempo neste campo. Valores negativos são permitidos.
Função de agregação	Especifique qual função de agregação usar: **min** - exibe o menor valor; **max** - exibe o maior valor; **avg** - exibe o valor médio; **sum** - exibe a soma dos valores; **count** - exibe a contagem dos valores; **first** - exibe o primeiro valor; **last** - exibe o último valor; **none** - exibe todos os valores (sem agregação). A agregação permite exibir um valor agregado para o intervalo escolhido (5 minutos, uma hora, um dia), em vez de todos os valores. Veja também: Agregação em gráficos.

Intervalo de agregação	Especifique o intervalo para agregar valores. Você pode usar sufixos de tempo neste campo. Um valor numérico sem sufixo será considerado como segundos. Observe que, se o widget estiver configurado para exibir dados históricos com base em tendências (*Seleção de dados históricos* estiver definido como *Tendências* ou *Auto*), é recomendável usar um intervalo de agregação que seja múltiplo de 1 hora (por exemplo, 3600, 60m, 1h, 3h, etc.). As tendências armazenam valores agregados por hora, portanto, usar um intervalo de agregação que não seja múltiplo de 1 hora (por exemplo, 100s, 7min, 15min, 90min, etc.) pode levar a resultados difíceis de interpretar.
Agregado	Especifique se deseja agregar: **Cada item** - cada item no conjunto de dados será agregado e exibido separadamente; **Conjunto de dados** - todos os itens do conjunto de dados serão agregados e exibidos como um único valor.
Aproximação	Especifique qual valor exibir quando houver mais de um valor por pixel vertical do gráfico: **all** - exibe o menor, o maior e o valor médio; **min** - exibe o menor valor; **max** - exibe o maior valor; **avg** - exibe o valor médio. Esta configuração é útil ao exibir um gráfico para um grande período de tempo com intervalo de atualização frequente (como um ano de valores coletados a cada 10 minutos).
Rótulo do conjunto de dados	Especifique o rótulo do conjunto de dados que é exibido na configuração do *Conjunto de dados* do gráfico e na *Legenda* do gráfico (para conjuntos de dados agregados). Todos os conjuntos de dados são numerados, incluindo aqueles com um *Rótulo do conjunto de dados* especificado. Se nenhum rótulo for especificado, o conjunto de dados será rotulado automaticamente de acordo com seu número (por exemplo, ”Conjunto de dados #2”, ”Conjunto de dados #3”, etc.). A numeração do conjunto de dados é recalculada após a reordenação/arrastar conjuntos de dados. Rótulos de conjuntos de dados muito longos serão encurtados para caber onde forem exibidos (por exemplo, ”Número de proc...”).

Detalhes da configuração do conjunto de dados

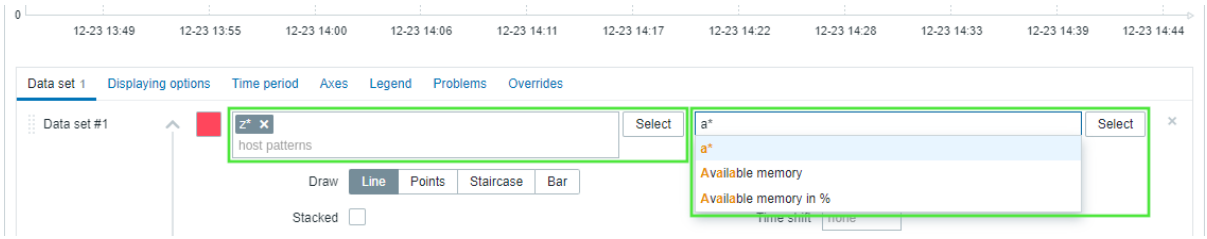
Os conjuntos de dados existentes são exibidos em uma lista. Você pode:

- Clicar no ícone de mover e arrastar um conjunto de dados para um novo local na lista.
- Clicar no ícone de expandir para expandir os detalhes do conjunto de dados. Quando expandido, este ícone se transforma em um ícone de recolher.
- Clicar no ícone de cor para alterar a cor, seja pelo seletor de cores ou manualmente. Para o conjunto de dados *Padrões de item*, a cor é usada para calcular diferentes tons de cor para cada item. Para o conjunto de dados *Lista de itens*, a cor é usada para o item especificado.
- Clicar no botão *Adicionar novo conjunto de dados* para adicionar um conjunto de dados vazio, permitindo selecionar padrões de host e item. Se você clicar no ícone apontando para baixo ao lado do botão *Adicionar novo conjunto de dados*, um menu suspenso aparecerá, permitindo adicionar um novo conjunto de dados *Padrões de item* ou *Lista de itens* ou permitindo *Clonar* o conjunto de dados atualmente aberto. Se todos os conjuntos de dados estiverem recolhidos, a opção *Clonar* não estará disponível.

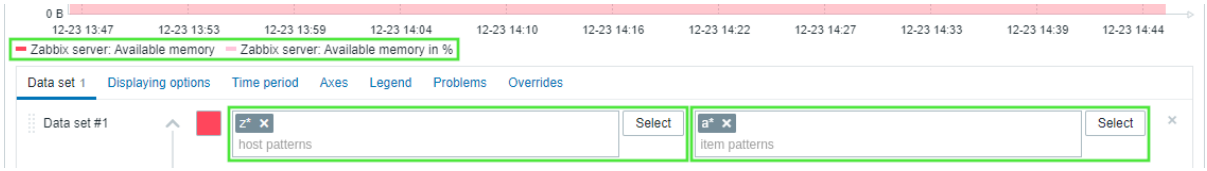


O conjunto de dados **Padrões de item** contém os campos *Padrões de host* e *Padrões de item*, que reconhecem nomes completos ou padrões contendo um caractere curinga (*). Esta funcionalidade permite selecionar todos os nomes de host e nomes de item que contenham o padrão selecionado. Ao digitar o nome do item ou padrão de item no campo *Padrões de item*, apenas os itens pertencentes ao(s) nome(s) de host selecionado(s) são exibidos na lista suspensa.

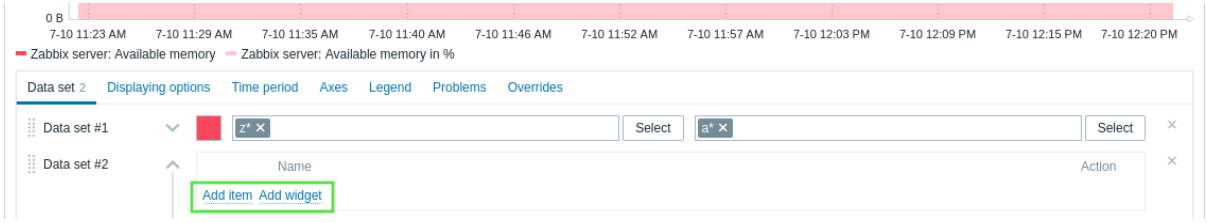
Por exemplo, ao digitar o padrão **z*** no campo *Padrões de host*, a lista suspensa exibe todos os nomes de host que contêm esse padrão: **z*, Zabbix server, Zabbix proxy**. Após pressionar *Enter*, esse padrão é aceito e exibido como **z***. Da mesma forma, ao digitar o padrão **a*** no campo *Padrões de item*, a lista suspensa exibe todos os nomes de item que contêm esse padrão: **a*, Available memory, Available memory in %**.



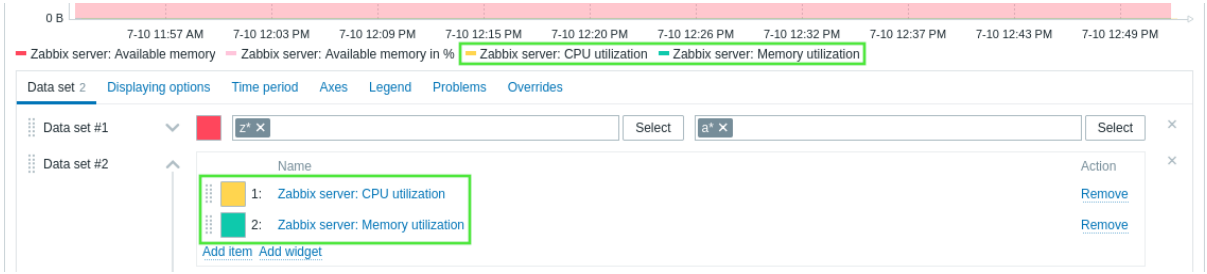
Após pressionar *Enter*, o padrão é aceito e exibido como **a***. O gráfico então exibe todos os itens pertencentes ao(s) nome(s) de host selecionado(s).



O conjunto de dados **Lista de items** contém o botão *Adicionar item* que permite adicionar items para serem exibidos no gráfico. Desde o Zabbix 7.0.1, você também pode adicionar widgets compatíveis como fonte de dados para items clicando no botão *Adicionar widget*.

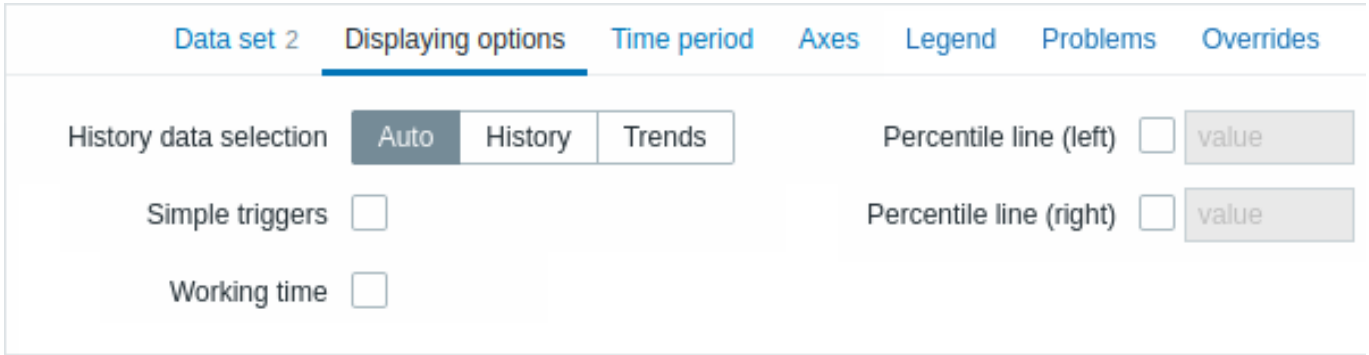


Por exemplo, clicar no botão *Adicionar item* abre uma janela pop-up contendo um parâmetro *Host*. Após selecionar um host, todos os seus items disponíveis para seleção são exibidos em uma lista. Após selecionar um ou mais items, eles serão exibidos na lista de items do conjunto de dados e no gráfico.



Opções de exibição

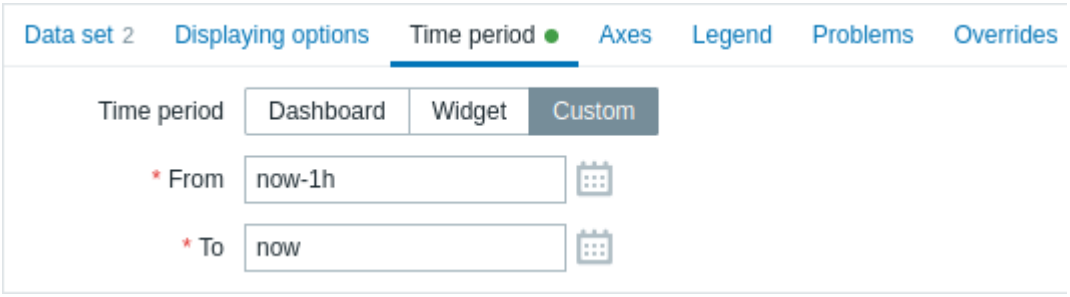
A aba **Opções de exibição** permite definir a seleção de dados do histórico:



Seleção de dados do histórico	Defina a fonte dos dados do gráfico: **Auto** - os dados são obtidos de acordo com o algoritmo de gráfico clássico (padrão); **Histórico** - dados do histórico; **Tendências** - dados das tendências.
Triggers simples	Marque a caixa para mostrar os limiares de trigger para triggers simples. Os limiares serão desenhados como linhas tracejadas usando a cor de severidade do trigger. Um trigger simples é um trigger com uma função (apenas `last`, `max`, `min`, `avg`) para um item na expressão. No máximo três triggers podem ser desenhados. Observe que o trigger deve estar dentro do intervalo desenhado para ser visível.
Tempo de trabalho	Marque a caixa para mostrar o tempo de trabalho no gráfico. O tempo de trabalho (dias úteis) é exibido nos gráficos como um fundo branco, enquanto o tempo não útil é exibido em cinza (com o tema padrão *Original blue* do frontend).
Linha de percentil (esquerda)	Marque a caixa e insira o valor do percentil para mostrar o percentil especificado como uma linha no eixo Y esquerdo do gráfico. Se, por exemplo, um percentil de 95% for definido, então a linha de percentil estará no nível onde 95 por cento dos valores ficam abaixo.
Linha de percentil (direita)	Marque a caixa e insira o valor do percentil para mostrar o percentil especificado como uma linha no eixo Y direito do gráfico. Se, por exemplo, um percentil de 95% for definido, então a linha de percentil estará no nível onde 95 por cento dos valores ficam abaixo.

Período de tempo

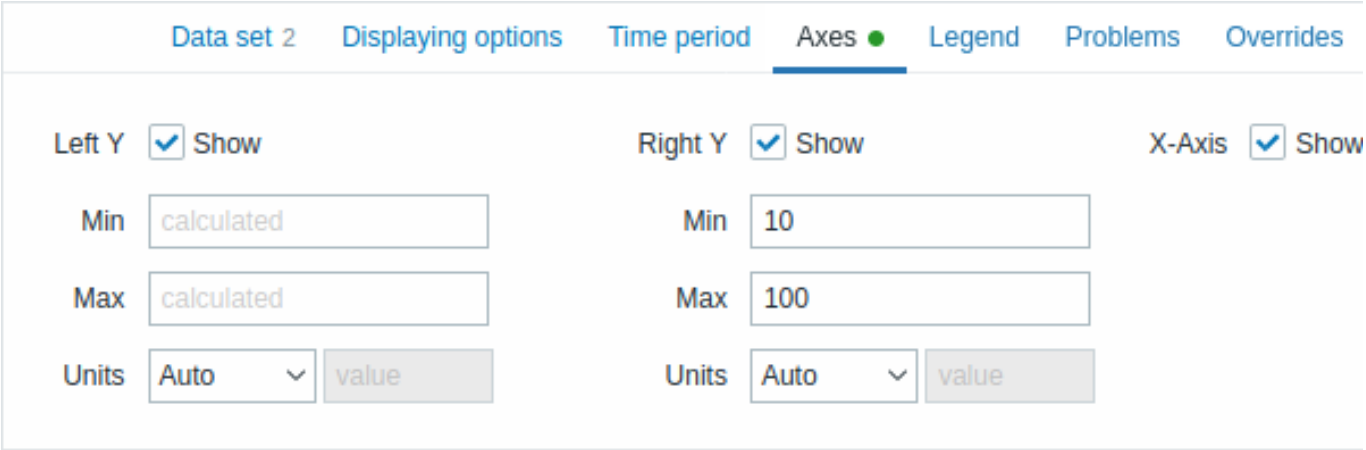
A aba **Período de tempo** permite definir um período de tempo para exibir dados no gráfico:



Período de tempo	Selecione a fonte de dados para o período de tempo: **Dashboard** - define o seletor *Período de tempo* como fonte de dados; **Widget** - define um widget compatível especificado no parâmetro *Widget* como fonte de dados; **Personalizado** - define o período de tempo especificado nos parâmetros *De* e *Até* como fonte de dados; se definido, um ícone de relógio será exibido no canto superior direito do widget, indicando o tempo definido ao passar o mouse.
Widget	Digite ou selecione um widget compatível como fonte de dados para o período de tempo. Este parâmetro está disponível se *Período de tempo* estiver definido como "Widget".
De	Digite ou selecione o início do período de tempo. Sintaxe de tempo relativo (`now`, `now/d`, `now/w-1w`, etc.) é suportada. Este parâmetro está disponível se *Período de tempo* estiver definido como "Personalizado".
Até	Digite ou selecione o final do período de tempo. Sintaxe de tempo relativo (`now`, `now/d`, `now/w-1w`, etc.) é suportada. Este parâmetro está disponível se *Período de tempo* estiver definido como "Personalizado".

Eixos

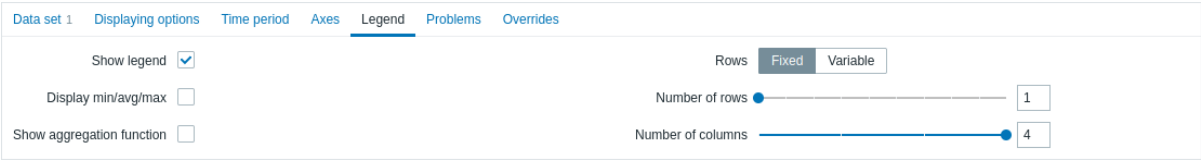
A aba **Eixos** permite personalizar como os eixos são exibidos:



Y à esquerda	Marque esta caixa de seleção para tornar o eixo Y à esquerda visível. A caixa de seleção pode estar desabilitada se não estiver selecionada na aba *Conjunto de dados* ou em *Substituições*.
Y à direita	Marque esta caixa de seleção para tornar o eixo Y à direita visível. A caixa de seleção pode estar desabilitada se não estiver selecionada na aba *Conjunto de dados* ou em *Substituições*.
Eixo X	Desmarque esta caixa de seleção para ocultar o eixo X (marcada por padrão).
Mín	Defina o valor mínimo do eixo correspondente. O valor mínimo do intervalo visível do eixo Y é especificado.
Máx	Defina o valor máximo do eixo correspondente. O valor máximo do intervalo visível do eixo Y é especificado.
Unidades	Escolha a unidade para os valores do eixo do gráfico no menu suspenso. Se a opção *Auto* for escolhida, os valores do eixo serão exibidos usando as unidades do primeiro item do eixo correspondente. A opção *Estático* permite atribuir um nome personalizado ao eixo correspondente. Se a opção *Estático* for escolhida e o campo de entrada *valor* for deixado em branco, o nome do eixo correspondente consistirá apenas em um valor numérico.

Legenda

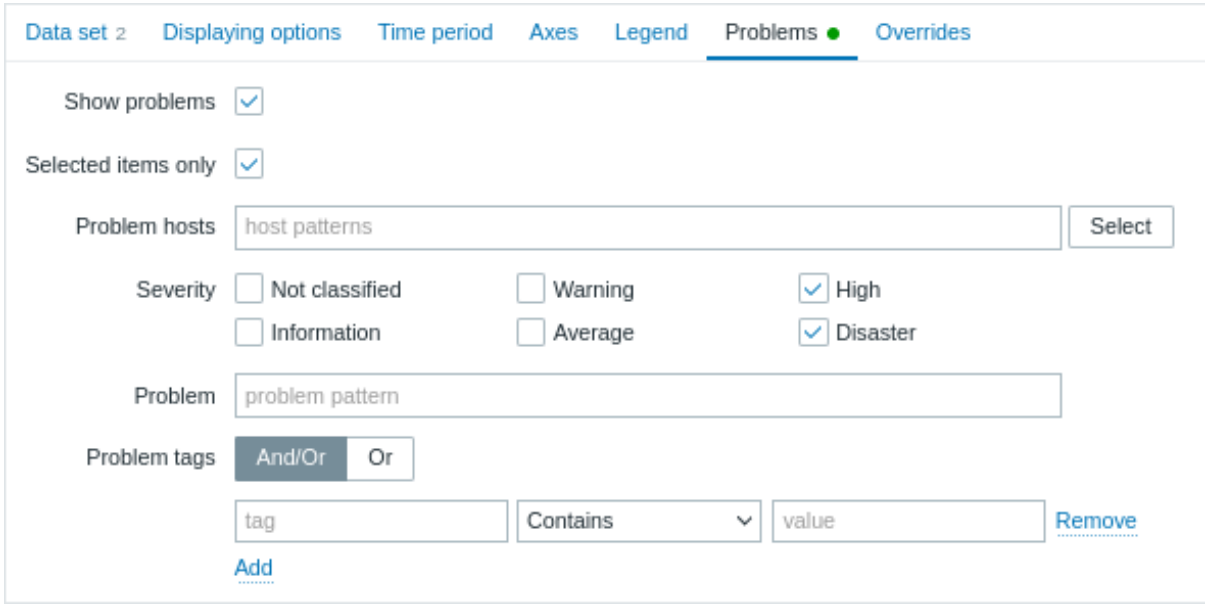
A aba **Legenda** permite personalizar a legenda do gráfico:



Exibir legenda	Desmarque esta caixa para ocultar a legenda no gráfico (marcada por padrão).
Exibir min/méd/máx	Marque esta caixa para exibir os valores mínimo, médio e máximo do item na legenda.
Exibir função de agregação	Marque esta caixa para exibir a função de agregação na legenda.
Linhas	Selecione o modo de exibição das linhas da legenda: **Fixo** - o número de linhas exibidas é determinado pelo valor do parâmetro *Número de linhas*; **Variável** - o número de linhas exibidas é determinado pela quantidade de items configurados, sem exceder o valor do parâmetro *Número máximo de linhas*.
Número de linhas/ Número máximo de linhas	Se *Linhas* estiver definido como ”Fixo”, defina o número de linhas da legenda a serem exibidas (1-10). Se *Linhas* estiver definido como ”Variável”, defina o número máximo de linhas da legenda a serem exibidas (1-10).
Número de colunas	Defina o número de colunas da legenda a serem exibidas (1-4). Este parâmetro está disponível se *Exibir min/méd/máx* estiver desmarcado.

Problemas

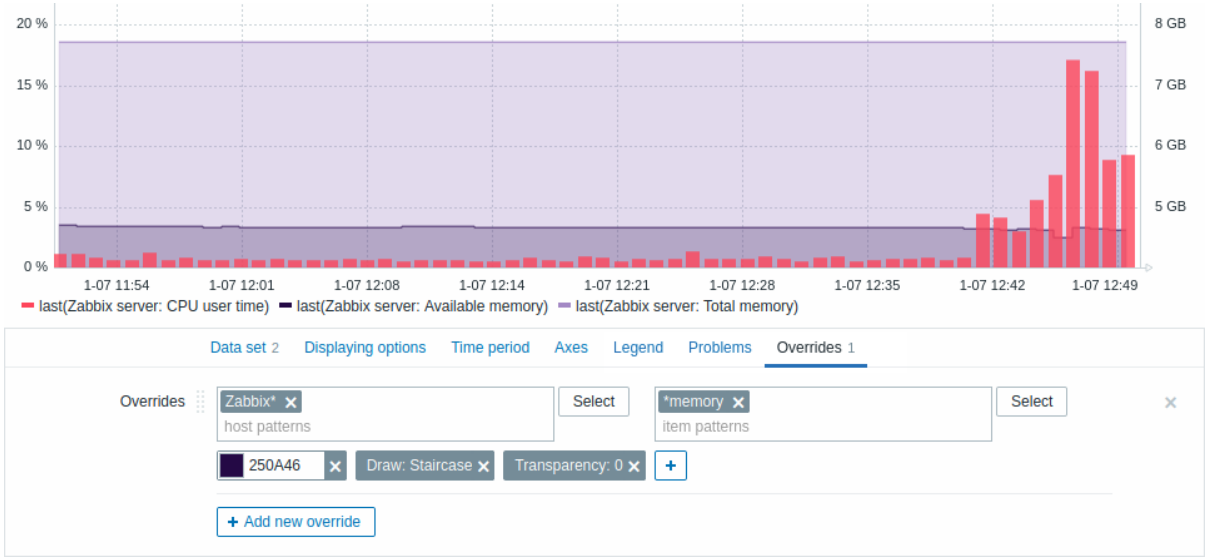
A aba **Problemas** permite personalizar a exibição de problemas:



Exibir problemas	Marque esta caixa de seleção para habilitar a exibição de problemas no gráfico (desmarcada, ou seja, desabilitada por padrão).
Apenas itens selecionados	Marque esta caixa de seleção para incluir apenas problemas dos itens selecionados a serem exibidos no gráfico.
Hosts de problema	Selecione os hosts de problema a serem exibidos no gráfico. Padrões curinga podem ser usados (por exemplo, * retornará resultados que correspondam a zero ou mais caracteres). Para especificar um padrão curinga, basta digitar a string manualmente e pressionar *Enter*. Enquanto você digita, observe como todos os hosts correspondentes são exibidos no menu suspenso. Este parâmetro não está disponível ao configurar o widget em um dashboard de template.
Gravidade	Marque as gravidades dos problemas para filtrar os problemas a serem exibidos no gráfico. Se nenhuma gravidade for marcada, todos os problemas serão exibidos.
Problema	Especifique o nome do problema a ser exibido no gráfico.
Tags de problema	Especifique tags de problema para limitar o número de problemas exibidos no widget. É possível incluir ou excluir tags e valores de tags específicos. Várias condições podem ser definidas. A correspondência do nome da tag diferencia maiúsculas de minúsculas. Existem vários operadores disponíveis para cada condição: **Existe** - inclui os nomes de tags especificados; **Igual** - inclui os nomes e valores de tags especificados (diferencia maiúsculas de minúsculas); **Contém** - inclui os nomes de tags especificados onde os valores das tags contêm a string inserida (correspondência de substring, não diferencia maiúsculas de minúsculas); **Não existe** - exclui os nomes de tags especificados; **Não é igual** - exclui os nomes e valores de tags especificados (diferencia maiúsculas de minúsculas); **Não contém** - exclui os nomes de tags especificados onde os valores das tags contêm a string inserida (correspondência de substring, não diferencia maiúsculas de minúsculas). Existem dois tipos de cálculo para as condições: **E/Ou** - todas as condições devem ser atendidas, condições com o mesmo nome de tag serão agrupadas pela condição *Ou*; **Ou** - basta que uma condição seja atendida.

Substituições

A aba **Substituições** permite adicionar substituições personalizadas para conjuntos de dados:

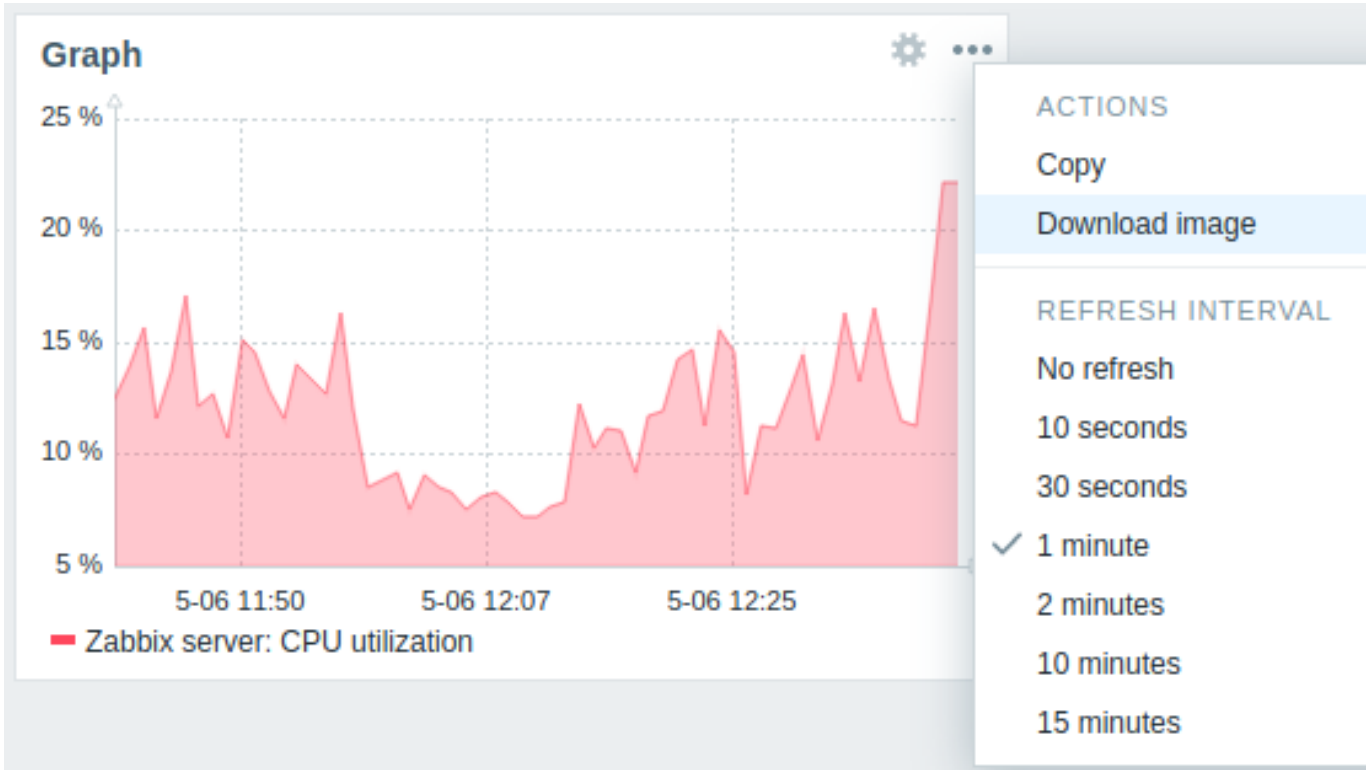


As substituições são úteis quando vários items são selecionados para um conjunto de dados usando o caractere curinga * e você deseja alterar como os items são exibidos por padrão (por exemplo, cor base padrão ou qualquer outra propriedade).

As substituições existentes (se houver) são exibidas em uma lista. Para adicionar uma nova substituição:

- Clique no botão + Add new override
- Selecione hosts e items para a substituição. Alternativamente, você pode inserir padrões de host e item. Padrões com caracteres curinga podem ser usados (por exemplo, * retornará resultados que correspondam a zero ou mais caracteres). Para especificar um padrão com curinga, basta inserir a string manualmente e pressionar *Enter*. Enquanto você digita, observe como todos os hosts correspondentes são exibidos no menu suspenso. O símbolo curinga é sempre interpretado, portanto, não é possível adicionar, por exemplo, um item chamado "item*" individualmente se houver outros items correspondentes (por exemplo, item2, item3). Os parâmetros de padrão de host e padrão de item são obrigatórios. O parâmetro para especificar padrões de host não está disponível ao configurar o widget em um dashboard de template. O parâmetro para especificar uma lista de items permite selecionar apenas items configurados no template ao configurar o widget em um dashboard de template.
- Clique em +, para selecionar os parâmetros de substituição. Pelo menos um parâmetro de substituição deve ser selecionado. Para descrições dos parâmetros, consulte a aba *Conjunto de dados* acima.

As informações exibidas pelo widget de gráfico podem ser baixadas como uma imagem .png usando o menu do widget:



Uma captura de tela do widget será salva na pasta Downloads.

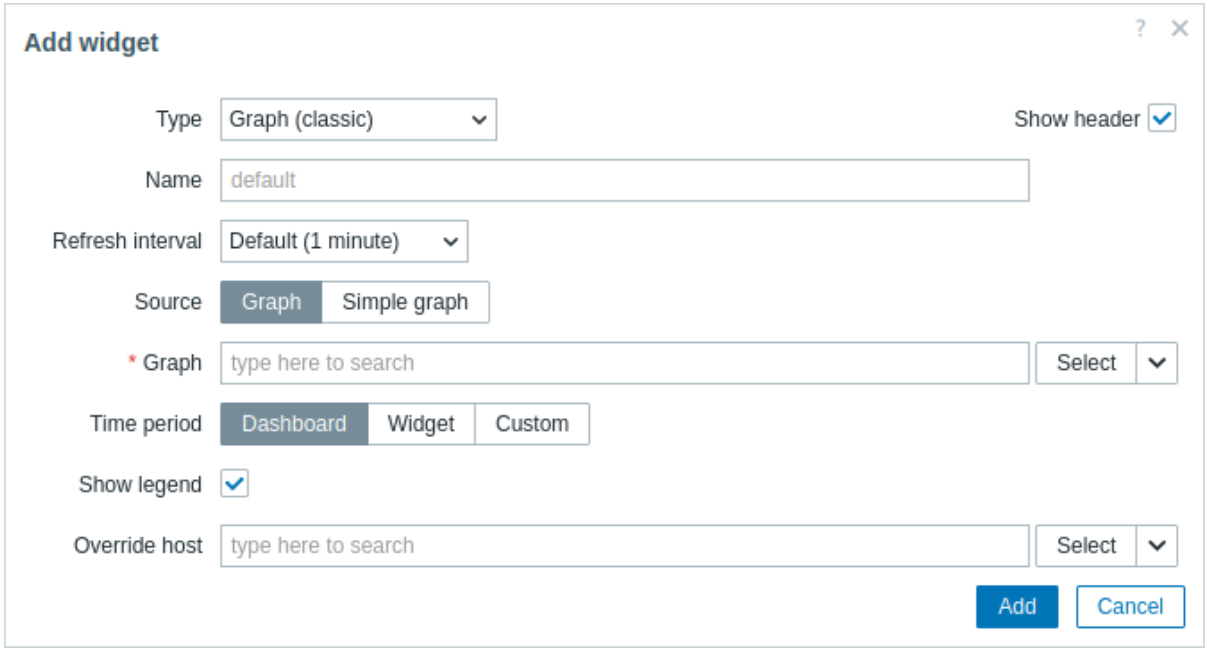
10 Gráfico (clássico)

Visão geral

No widget de gráfico clássico, você pode exibir um único gráfico personalizado ou gráfico simples.

Configuração

Para configurar, selecione *Gráfico (clássico)* como tipo:



Além dos parâmetros que são comuns para todos os widgets, você pode definir as seguintes opções específicas:

Fonte	Selecione o tipo de gráfico: **Gráfico** - gráfico personalizado; **Gráfico simples** - gráfico simples.
Gráfico	Selecione o gráfico personalizado a ser exibido. Como alternativa, selecione um widget compatível como fonte de dados para gráficos. Este parâmetro está disponível se *Fonte* estiver definido como ”Gráfico”.
Item	Selecione o item a ser exibido em um gráfico simples. Como alternativa, selecione um widget compatível como fonte de dados para itens. Este parâmetro está disponível se *Fonte* estiver definido como ”Gráfico simples”.
Período de tempo	Defina um período de tempo para exibir os dados no gráfico. Selecione a fonte de dados para o período de tempo: **Dashboard** - define o seletor *Período de tempo* como fonte de dados; **Widget** - define um widget compatível especificado no parâmetro *Widget* como fonte de dados; **Personalizado** - define o período de tempo especificado nos parâmetros *De* e *Até* como fonte de dados; se definido, um ícone de relógio será exibido no canto superior direito do widget, indicando o tempo definido ao passar o mouse.
Widget	Digite ou selecione um widget compatível como fonte de dados para o período de tempo. Este parâmetro está disponível se *Período de tempo* estiver definido como ”Widget”.
De	Digite ou selecione o início do período de tempo. Sintaxe de tempo relativo (`now`, `now/d`, `now/w-1w`, etc.) é suportada. Este parâmetro está disponível se *Período de tempo* estiver definido como ”Personalizado”.
Até	Digite ou selecione o final do período de tempo. Sintaxe de tempo relativo (`now`, `now/d`, `now/w-1w`, etc.) é suportada. Este parâmetro está disponível se *Período de tempo* estiver definido como ”Personalizado”.
Mostrar legenda	Desmarque esta caixa para ocultar a legenda no gráfico (marcada por padrão).
Substituir host	Selecione um widget compatível ou o dashboard como fonte de dados para hosts. Este parâmetro não está disponível ao configurar o widget em um dashboard de template.

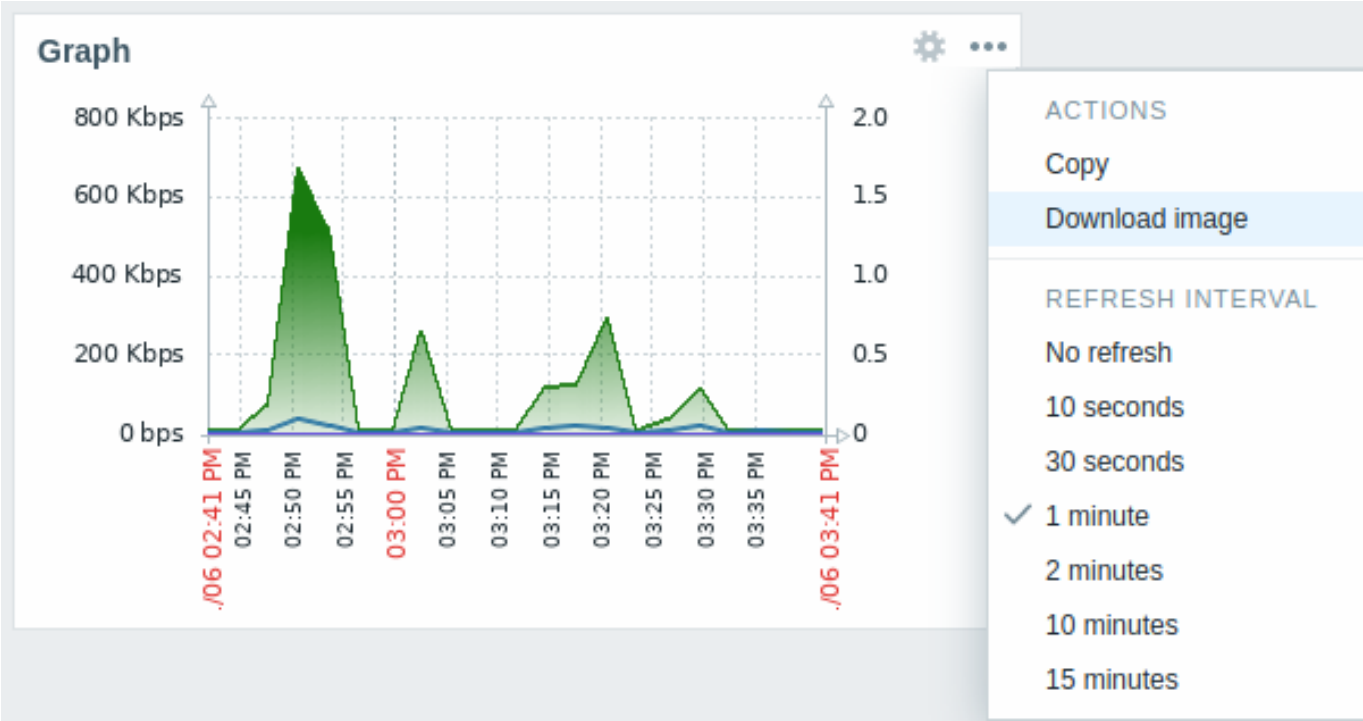
A legenda do gráfico consiste em três seções:

- Itens e seus valores agregados
- Percentis (se configurados)
- Triggers (se houver algum associado aos itens exibidos)

Se a altura do gráfico dentro do widget for insuficiente, a legenda pode não ser exibida ou pode ser exibida apenas parcialmente. Triggers e percentis são ocultados primeiro, seguidos pela legenda do item. Para exibir a legenda completa, aumente o tamanho vertical do widget.

Note:
Não mais que 3 linhas de trigger podem ser exibidas. Se houver mais triggers, as triggers com menor severidade terão prioridade para exibição.

As informações exibidas pelo widget de gráfico clássico podem ser baixadas como imagem *.png* usando o menu do widget:



Uma captura de tela do widget será salva na pasta *Downloads*.

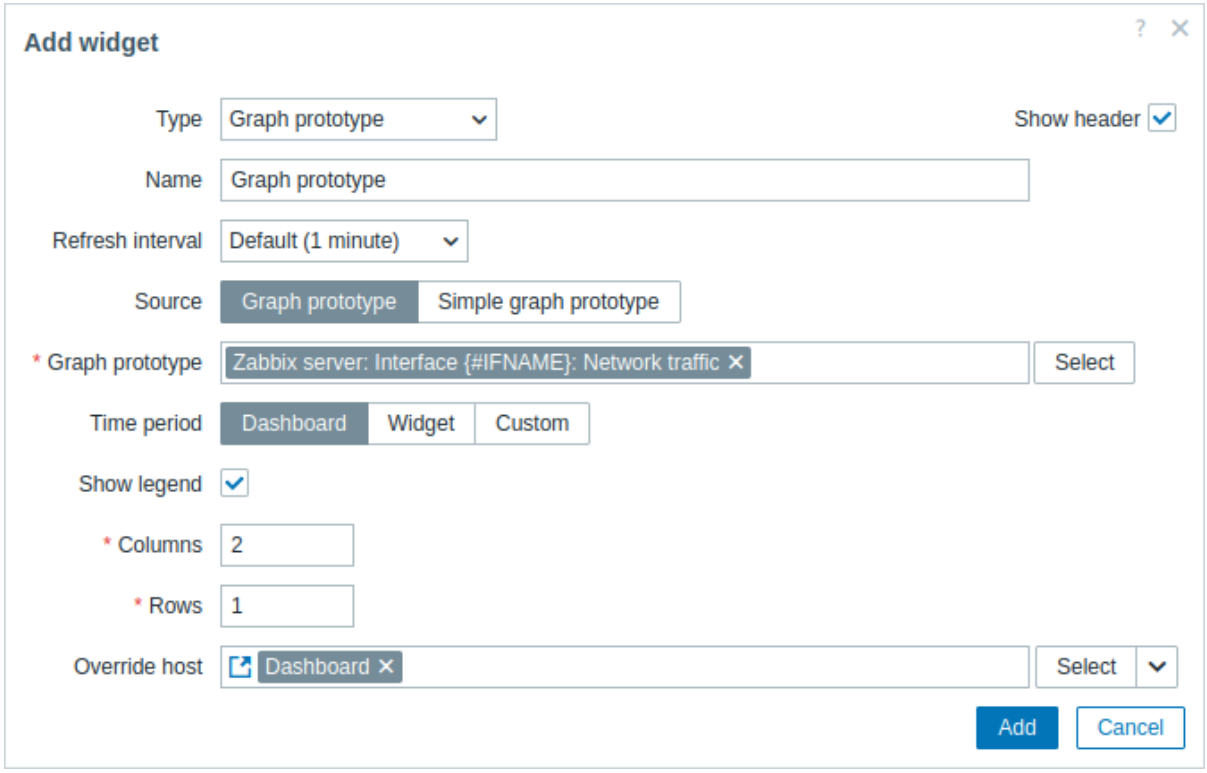
11 Protótipo de gráfico

Visão geral

No widget de protótipo de gráfico, você pode exibir uma grade de gráficos criados a partir de um protótipo de gráfico ou de um protótipo de item por descoberta de baixo nível.

Configuração

Para configurar, selecione *Protótipo de gráfico* como tipo de widget:



Além dos parâmetros que são comuns para todos os widgets, você pode definir as seguintes opções específicas:

Origem	Selecione a origem dos gráficos: **Protótipo de gráfico** ou **Protótipo de gráfico simples**.
Protótipo de gráfico	Selecione um protótipo de gráfico para exibir gráficos descobertos pelo protótipo de gráfico. Este parâmetro está disponível se *Origem* estiver definido como "Protótipo de gráfico".
Protótipo de item	Selecione um protótipo de item para exibir gráficos simples para itens descobertos pelo protótipo de item. Este parâmetro está disponível se *Origem* estiver definido como "Protótipo de gráfico simples".
Período de tempo	Defina um período de tempo para exibir dados nos gráficos. Selecione a fonte de dados para o período de tempo: **Dashboard** - define o seletor *Período de tempo* como fonte de dados; **Widget** - define um widget compatível especificado no parâmetro *Widget* como fonte de dados; **Personalizado** - define o período de tempo especificado nos parâmetros *De* e *Até* como fonte de dados; se definido, um ícone de relógio será exibido no canto superior direito do widget, indicando o tempo definido ao passar o mouse.
Widget	Digite ou selecione um widget compatível como fonte de dados para o período de tempo. Este parâmetro está disponível se *Período de tempo* estiver definido como "Widget".
De	Digite ou selecione o início do período de tempo. Sintaxe de tempo relativo (`now`, `now/d`, `now/w-1w`, etc.) é suportada. Este parâmetro está disponível se *Período de tempo* estiver definido como "Personalizado".
Até	Digite ou selecione o final do período de tempo. Sintaxe de tempo relativo (`now`, `now/d`, `now/w-1w`, etc.) é suportada. Este parâmetro está disponível se *Período de tempo* estiver definido como "Personalizado".
Mostrar legenda	Desmarque esta caixa para ocultar a legenda nos gráficos (marcada por padrão).
Substituir host	Selecione um widget compatível ou o dashboard como fonte de dados para hosts. Este parâmetro não está disponível ao configurar o widget em um dashboard de template.
Colunas	Digite o número de colunas de gráficos a serem exibidas em um widget de protótipo de gráfico.
Linhas	Digite o número de linhas de gráficos a serem exibidas em um widget de protótipo de gráfico.

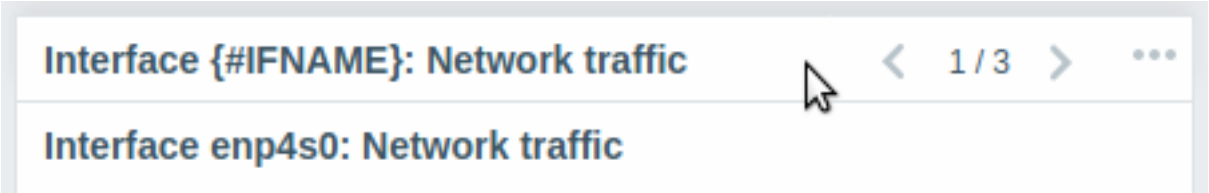
A legenda do gráfico consiste em três seções:

- Itens e seus valores agregados
- Percentis (se configurados)
- Triggers (se houver algum associado aos itens exibidos)

Se a altura do gráfico dentro do widget for insuficiente, a legenda pode não ser exibida ou pode ser exibida apenas parcialmente. Triggers e percentis são ocultados primeiro, seguidos pela legenda do item. Para exibir a legenda completa, aumente o tamanho vertical do widget.

Note:
Não é possível exibir mais de 3 linhas de trigger. Se houver mais triggers, as triggers com menor severidade terão prioridade para exibição.

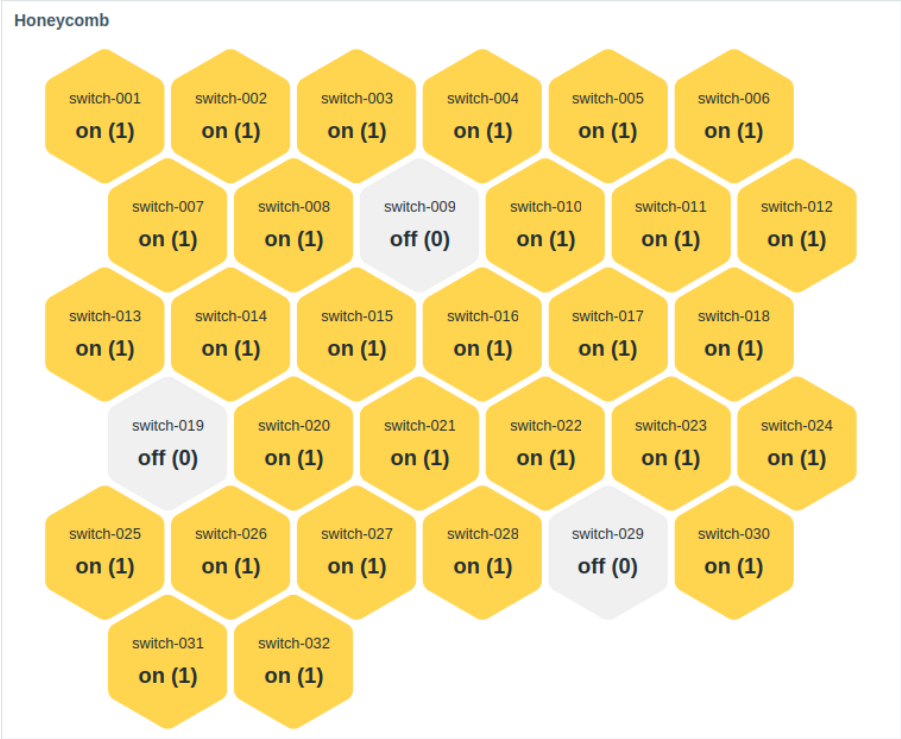
Embora os parâmetros *Colunas* e *Linhas* permitam ajustar mais de um gráfico no widget, ainda pode haver mais gráficos descobertos do que colunas/linhas no widget. Neste caso, a paginação fica disponível no widget, e um cabeçalho deslizante permite alternar entre as páginas usando as setas esquerda e direita:



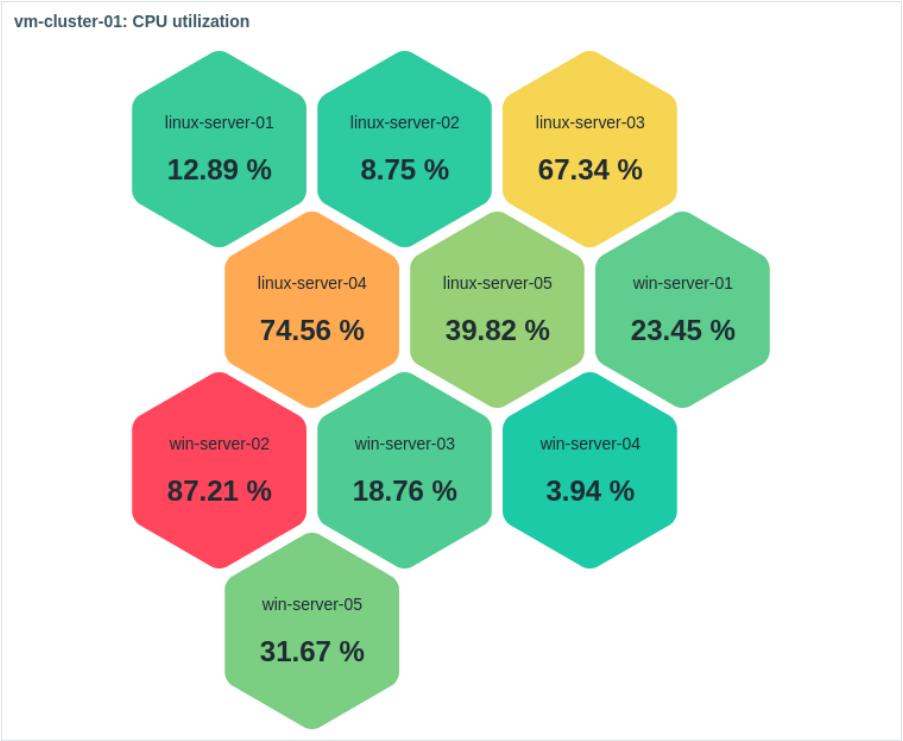
12 Honeycomb

Visão geral

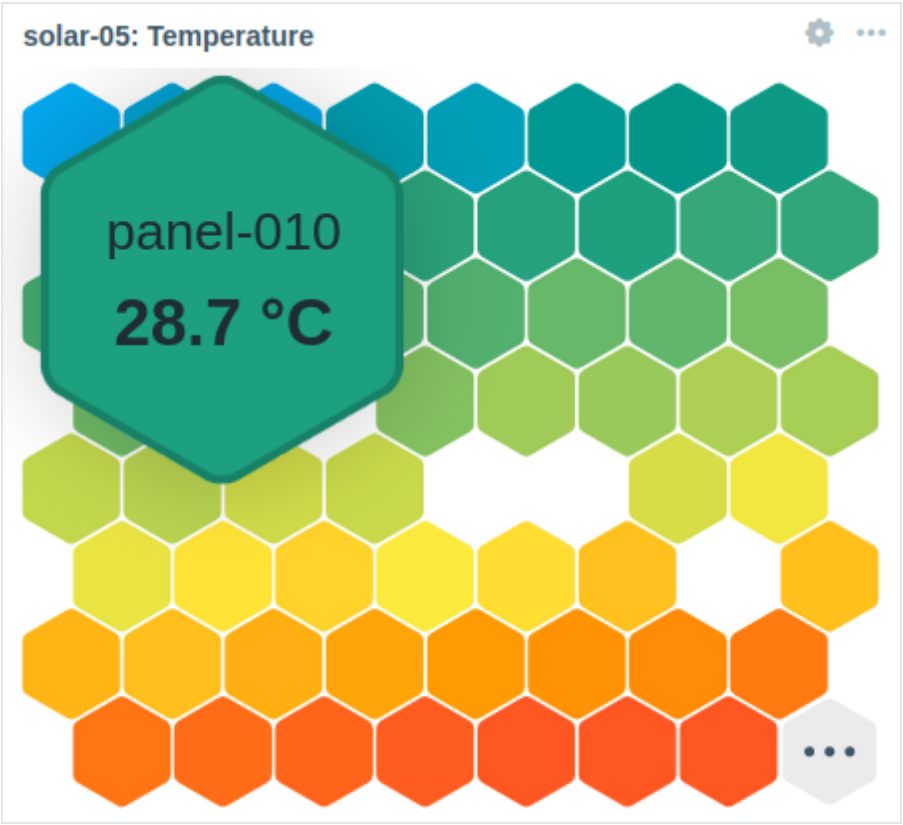
O widget honeycomb oferece uma visão geral dinâmica da infraestrutura de rede e recursos monitorados, onde grupos de hosts, como máquinas virtuais e dispositivos de rede, juntamente com seus respectivos items, são representados visualmente como células hexagonais interativas.



O widget pode ser ajustado visualmente usando as opções de configuração avançada para criar uma grande variedade de estilos visuais.



Ao passar o mouse, a célula honeycomb em foco é ampliada para melhor visibilidade. Clicar em uma célula destaca sua borda até que outra célula seja selecionada.

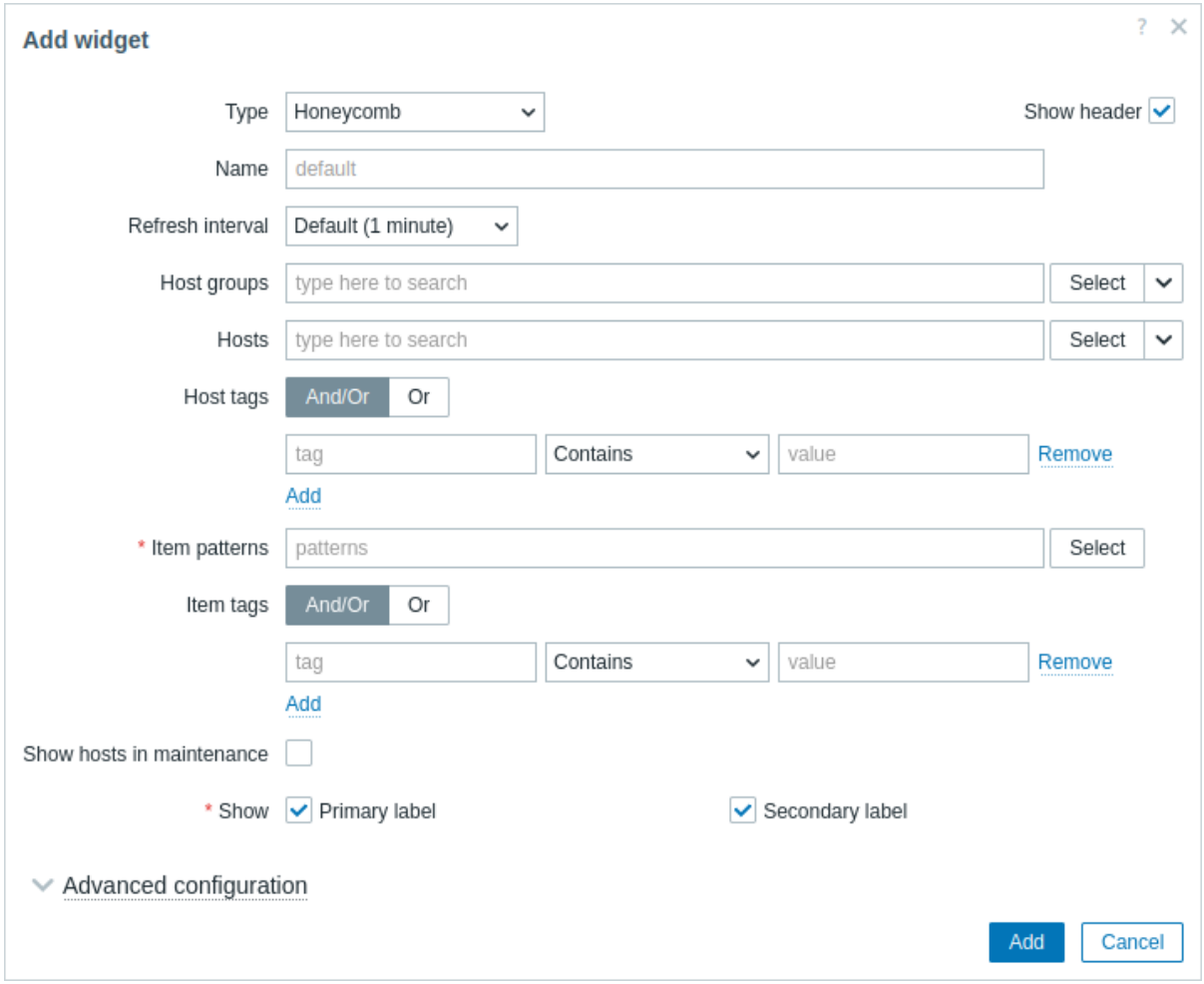


O número de células honeycomb exibidas é limitado pelo tamanho do widget e pelo tamanho mínimo da célula (32 pixels). Se todas as células não couberem no widget, uma elipse é mostrada como a célula final.

O widget pode ser redimensionado para acomodar mais células. Ao redimensionar, o tamanho e o posicionamento das células honeycomb são ajustados dinamicamente. Cada linha no honeycomb manterá uma contagem igual de células, exceto para a última linha se o número total de células não for divisível pela contagem de células da linha.

Configuração

Para configurar, selecione *Honeycomb* como tipo:



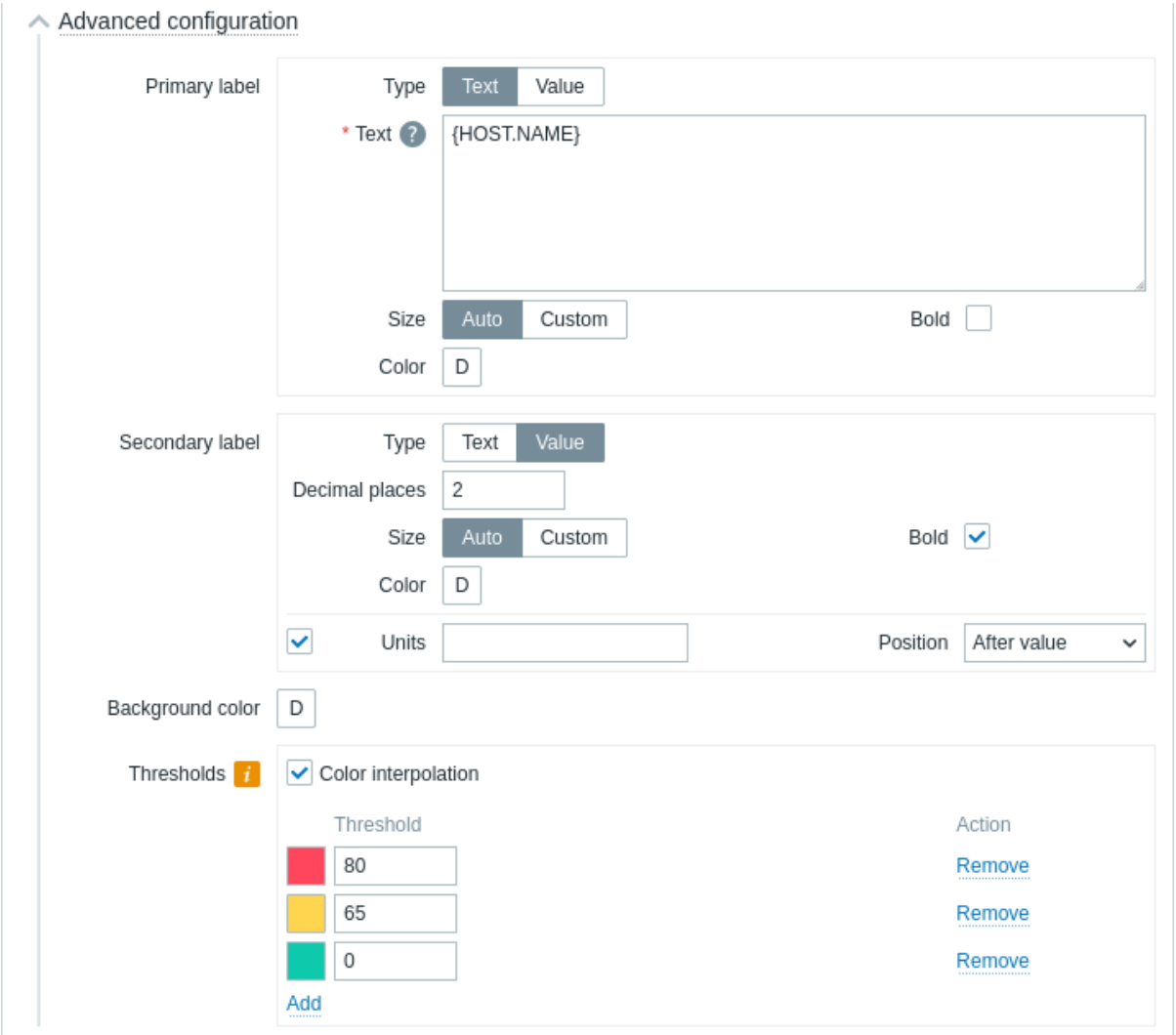
Além dos parâmetros que são comuns para todos os widgets, você pode definir as seguintes opções específicas:

Grupos de hosts	Selecione grupos de hosts. Como alternativa, selecione um widget compatível como fonte de dados para grupos de hosts. Este campo é auto-completável, então ao começar a digitar o nome de um grupo será exibida uma lista suspensa de grupos correspondentes. Selecionar um grupo de hosts pai seleciona implicitamente todos os grupos de hosts aninhados; se nenhum grupo de hosts for selecionado, o widget exibirá todos os grupos de hosts contendo hosts com items que correspondam ao *Padrão de item* selecionado (veja abaixo). Este parâmetro não está disponível ao configurar o widget em um dashboard de template.
Hosts	Selecione hosts. Como alternativa, selecione um widget compatível ou o dashboard como fonte de dados para hosts. Este campo é auto-completável, então ao começar a digitar o nome de um host será exibida uma lista suspensa de hosts correspondentes. Se nenhum host for selecionado, o widget exibirá todos os hosts com items que correspondam ao *Padrão de item* selecionado (veja abaixo). Este parâmetro não está disponível ao configurar o widget em um dashboard de template.

Tags de host	Especifique tags para limitar o número de hosts exibidos no widget. É possível incluir ou excluir tags e valores de tags específicos. Várias condições podem ser definidas. A correspondência do nome da tag diferencia maiúsculas de minúsculas. Existem vários operadores disponíveis para cada condição: **Existe** - inclui os nomes de tags especificados; **Igual** - inclui os nomes e valores de tags especificados (diferencia maiúsculas de minúsculas); **Contém** - inclui os nomes de tags especificados onde os valores das tags contêm a string inserida (correspondência de substring, não diferencia maiúsculas de minúsculas); **Não existe** - exclui os nomes de tags especificados; **Não é igual** - exclui os nomes e valores de tags especificados (diferencia maiúsculas de minúsculas); **Não contém** - exclui os nomes de tags especificados onde os valores das tags contêm a string inserida (correspondência de substring, não diferencia maiúsculas de minúsculas). Existem dois tipos de cálculo para as condições: **E/Ou** - todas as condições devem ser atendidas, condições com o mesmo nome de tag serão agrupadas pela condição Ou; **Ou** - basta que uma condição seja atendida. Este parâmetro não está disponível ao configurar o widget em um dashboard de template.
Padrões de item	Digite padrões de item ou selecione items existentes como padrões de item. Os dados dos items que correspondam aos padrões inseridos ou selecionados serão exibidos no honeycomb. O parâmetro *Padrões de item* é obrigatório. Padrões com curingas podem ser usados para seleção (por exemplo, `*` retornará items que correspondam a zero ou mais caracteres; `Zabbix*` retornará items que comecem com "Zabbix"). Para especificar um padrão curinga, digite a string manualmente e pressione *Enter*. Ao começar a digitar, uma lista suspensa mostrará os items correspondentes limitados aos pertencentes aos *Hosts* selecionados ou hosts dentro dos *Grupos de hosts* selecionados, se houver. O símbolo curinga é sempre interpretado, portanto, não é possível adicionar, por exemplo, um item chamado *item** individualmente, se houver outros items correspondentes (por exemplo, item2, item3). Ao configurar o widget em um dashboard de template, este parâmetro permite selecionar apenas items configurados no template.
Tags de item	Especifique tags para limitar o número de items exibidos no widget. Para mais informações, veja *Tags de host* acima.
Exibir hosts em manutenção	Marque esta caixa para exibir hosts em manutenção (neste caso, o ícone de manutenção será exibido ao lado do nome do host). Este parâmetro é rotulado como *Exibir dados em manutenção* ao configurar o widget em um dashboard de template.
Exibir	Marque esta caixa para exibir o respectivo elemento da célula do honeycomb - rótulo primário, rótulo secundário. Pelo menos um elemento deve ser selecionado.
Configuração avançada	Clique no rótulo *Configuração avançada* para exibir as opções de configuração avançada.

Configuração avançada

As opções de configuração avançada estão disponíveis na seção recolhível *Configuração avançada* e apenas para os elementos selecionados no campo *Mostrar* (veja acima) e para a cor de fundo ou limites para células honeycomb.



Rótulo primário/secundário

Tipo — Selecione o tipo de rótulo:
Texto - o rótulo exibirá o texto especificado no parâmetro *Texto*;
Valor - o rótulo exibirá o valor do item com casas decimais conforme especificado no parâmetro *Casas decimais*.

Texto — Digite o texto do rótulo. Este texto pode substituir o nome padrão do item.
Texto multilinha é suportado. Uma combinação de texto e macros suportadas é possível.
{HOST.*}, {ITEM.*}, {INVENTORY.*} e macros de usuário são suportadas.
As células honeycomb são ordenadas alfabeticamente pelo nome do host e, dentro de cada host, pelo nome do item.
Este parâmetro está disponível se *Tipo* estiver definido como "Texto".

Casas decimais — Digite o número de casas decimais a serem exibidas com o valor.
Este parâmetro está disponível se *Tipo* estiver definido como "Valor" e afeta apenas itens que retornam dados numéricos (float).

Tamanho — Selecione o tamanho do rótulo:
Auto - usar tamanho de rótulo ajustado automaticamente;
Personalizado - insira um tamanho de rótulo personalizado (em porcentagem, relativo ao tamanho da célula honeycomb).
Observe que rótulos que não cabem no tamanho da célula honeycomb são truncados.

Negrito — Marque a caixa de seleção para exibir as unidades do item em negrito.

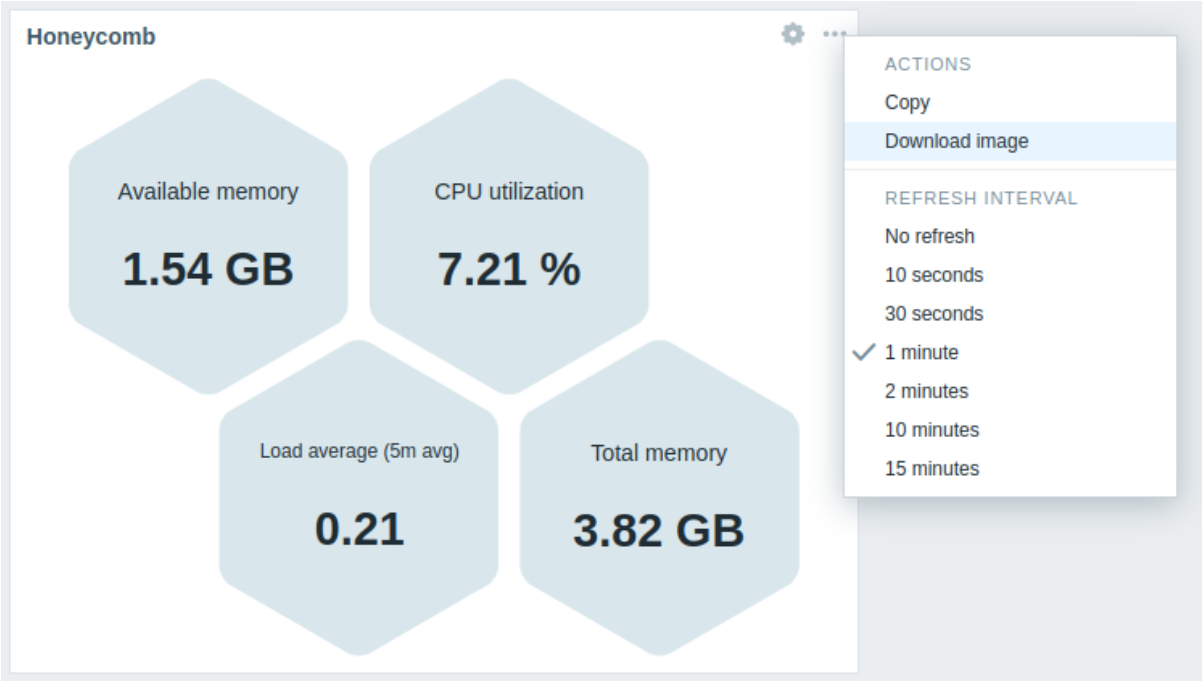
Cor — Selecione a cor das unidades do item no seletor de cores.
"D" significa a cor padrão, que depende do tema do frontend. Para retornar à cor padrão, clique no botão *Usar padrão* no seletor de cores.

Unidades

Unidades — Marque a caixa de seleção para exibir as unidades com o valor do item.
Se você inserir um nome de unidade, ele substituirá as unidades definidas na configuração do item.
Este parâmetro está disponível se *Tipo* estiver definido como "Texto".

Posição	Selecione a posição das unidades do item (antes ou depois do valor do item). Este parâmetro é ignorado para as seguintes unidades relacionadas ao tempo: unixtime, uptime, s. Este parâmetro está disponível se *Tipo* estiver definido como ”Texto”.
Cor de fundo	
Cor de fundo	Selecione a cor de fundo das células honeycomb no seletor de cores. ”D” significa a cor padrão, que depende do tema do frontend. Para retornar à cor padrão, clique no botão *Usar padrão* no seletor de cores.
Limites	
Interpolação de cor	Marque a caixa de seleção para ativar a transição suave entre as cores dos limites para as células honeycomb. Este parâmetro está disponível se dois ou mais limites forem definidos.
Limite	Clique em *Adicionar* para adicionar um limite, selecione uma cor de limite no seletor de cores e especifique um valor numérico. A lista de limites será classificada em ordem crescente ao salvar. Observe que as cores configuradas como limites serão exibidas corretamente apenas para itens numéricos. Sufixos (por exemplo, ”1d”, ”2w”, ”4K”, ”8G”) são suportados. Mapeamentos de valor são suportados.

As informações exibidas pelo widget honeycomb podem ser baixadas como uma imagem *.png* usando o menu do widget:



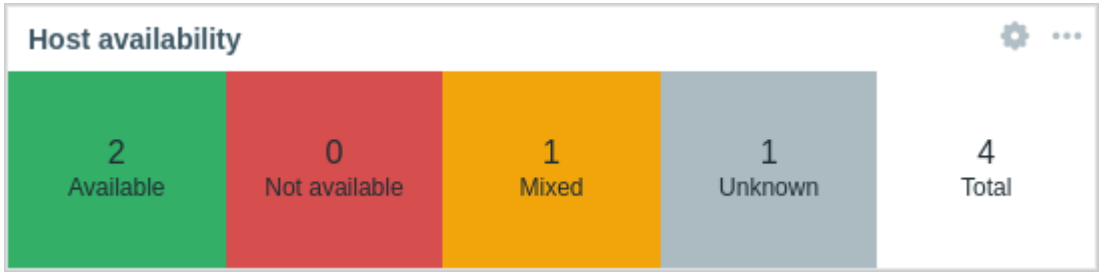
Uma captura de tela do widget será salva na pasta *Downloads*.

13 Disponibilidade do host

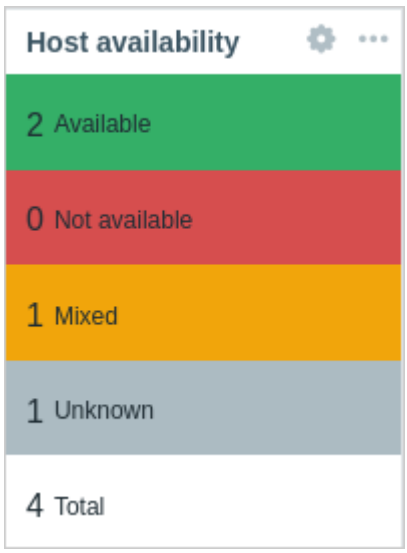
Visão geral

No widget de disponibilidade de host, estatísticas de alto nível sobre a disponibilidade dos hosts são exibidas em colunas/linhas coloridas, dependendo do layout escolhido.

Exibição horizontal (colunas):



Exibição vertical (linhas):



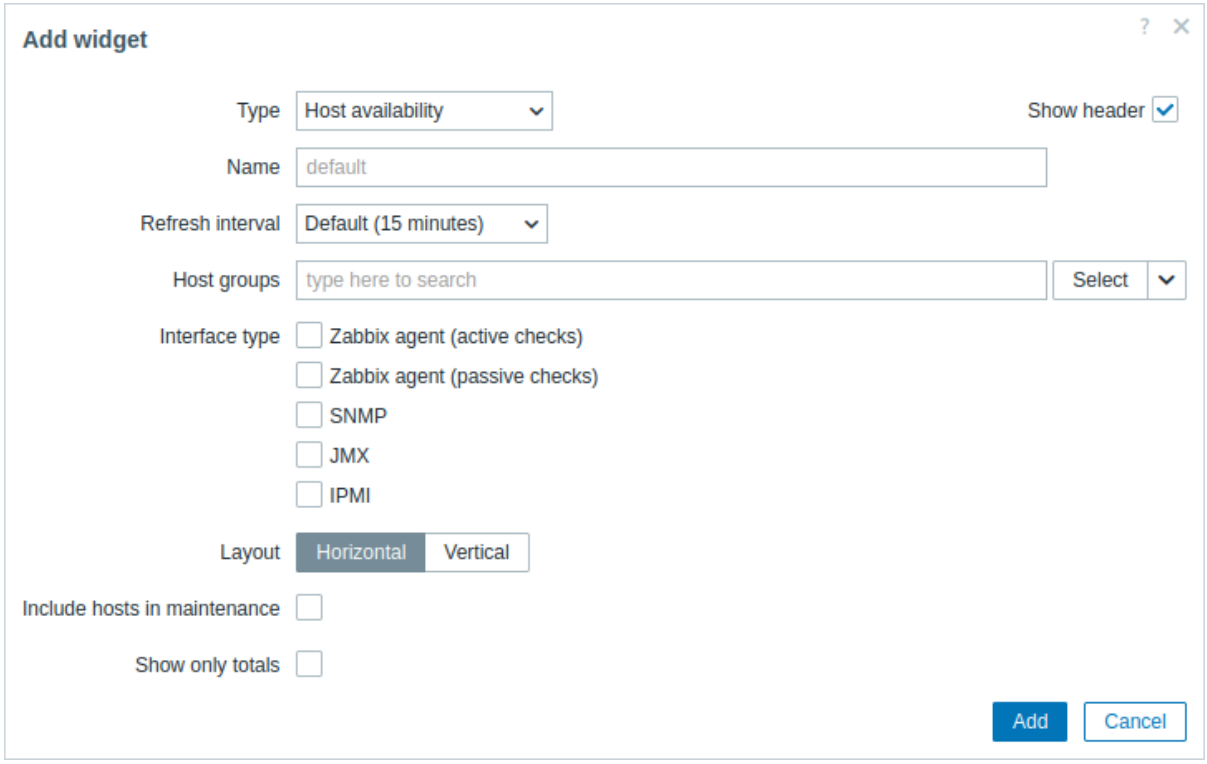
A disponibilidade do host em cada coluna/linha é contada da seguinte forma:

- *Disponível* - hosts com todas as interfaces disponíveis
- *Não disponível* - hosts com todas as interfaces indisponíveis
- *Misto* - hosts com pelo menos uma interface indisponível e pelo menos uma disponível ou desconhecida; as demais podem ter qualquer valor, inclusive ”desconhecido”
- *Desconhecido* - hosts com pelo menos uma interface desconhecida (nenhuma indisponível)
- *Total* - total de todos os hosts

Note:
Para o Zabbix agent (cheques ativos), a célula *Misto* estará sempre vazia, pois esse tipo de item não pode ter múltiplas interfaces.

Configuração

Para configurar, selecione *Disponibilidade do host* como tipo:



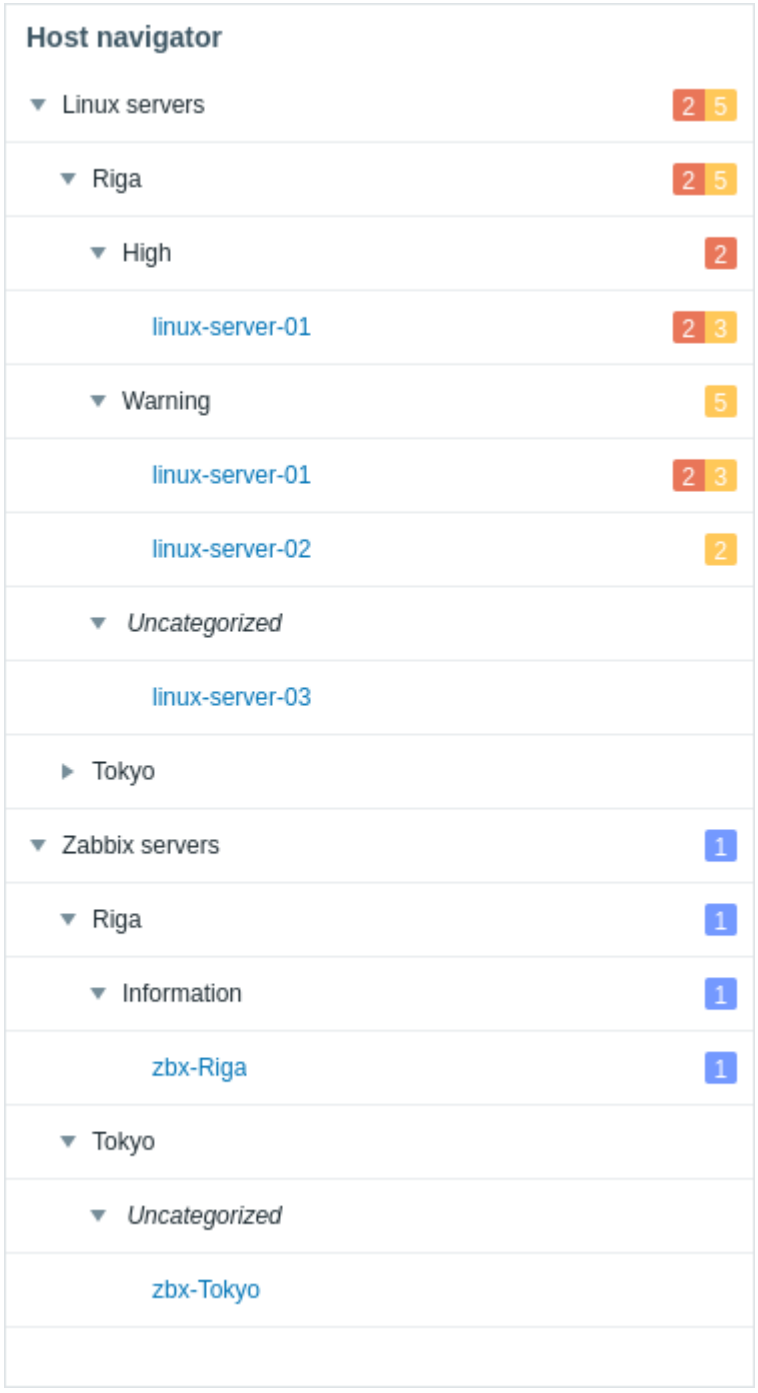
Além dos parâmetros que são comuns para todos os widgets, você pode definir as seguintes opções específicas:

Grupos de hosts	Selecione grupos de hosts. Como alternativa, selecione um widget compatível como fonte de dados para grupos de hosts. Este campo é de preenchimento automático, portanto, ao começar a digitar o nome de um grupo, será exibida uma lista suspensa com os grupos correspondentes. Este parâmetro não está disponível ao configurar o widget em um dashboard de template.
Tipo de interface	Selecione para quais interfaces de host você deseja ver os dados de disponibilidade. A disponibilidade de todas as interfaces é exibida por padrão se nada for selecionado.
Layout	Selecione exibição horizontal (colunas) ou exibição vertical (linhas).
Incluir hosts em manutenção	Inclua hosts que estão em manutenção nas estatísticas. Este parâmetro é rotulado como *Exibir dados em manutenção* ao configurar o widget em um dashboard de template.
Exibir apenas totais	Se marcado, o total de hosts, sem detalhamento por interfaces, é exibido. Esta opção é desabilitada se apenas uma interface for selecionada.

14 Navegador de hosts

Visão geral

O widget de navegação de host exibe hosts com base em várias opções de filtragem e agrupamento.



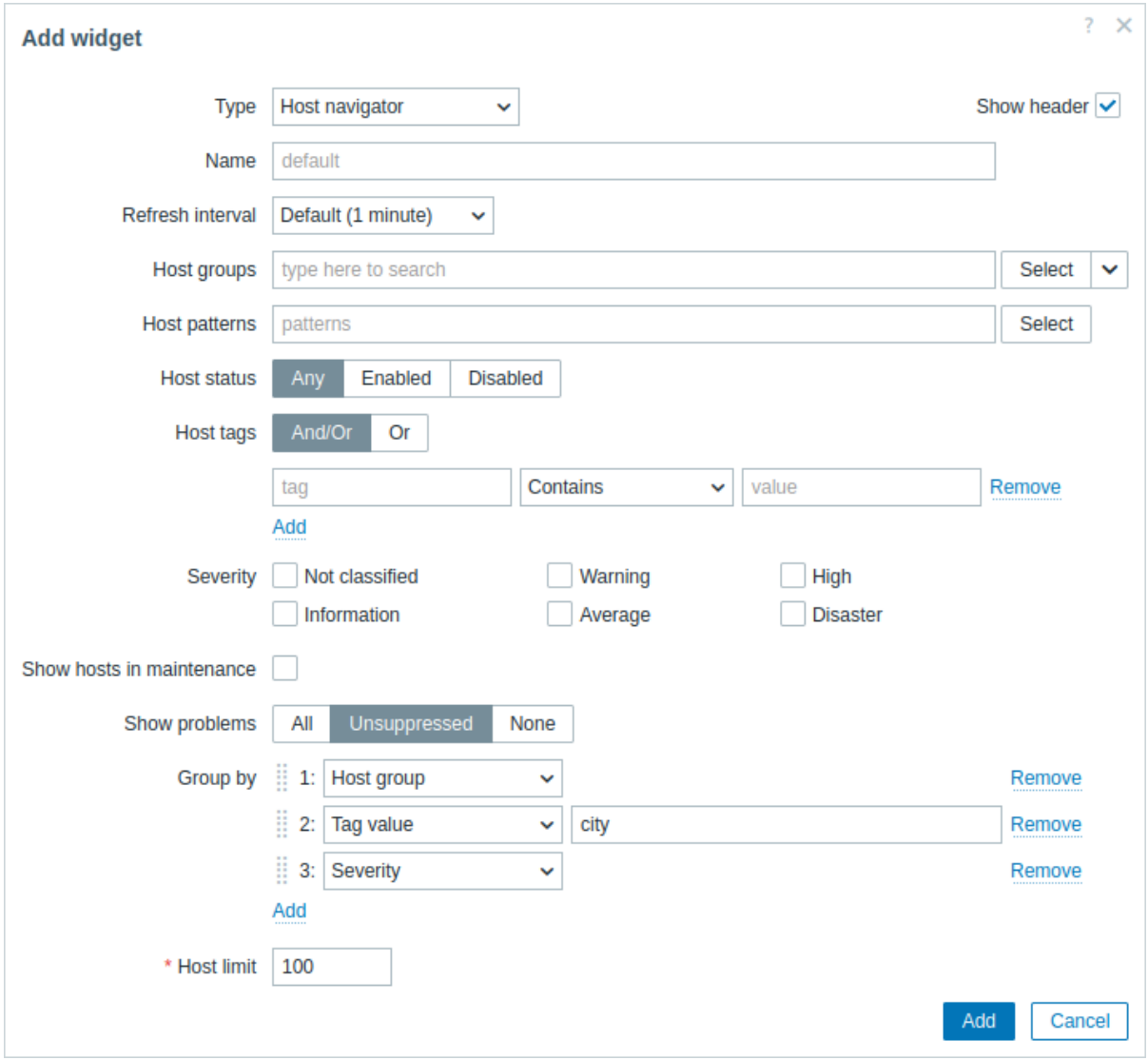
O widget também permite controlar as informações exibidas em outros widgets com base no host selecionado.

Os grupos pelos quais os hosts são organizados podem ser expandidos ou recolhidos.

Para grupos, problemas e hosts em manutenção, detalhes adicionais são acessíveis por dicas ao passar o mouse.

Configuração

Para configurar, selecione *Host navigator* como tipo:



Além dos parâmetros que são comuns para todos os widgets, você pode definir as seguintes opções específicas:

Grupos de hosts	Selecione grupos de hosts. Como alternativa, selecione um widget compatível como fonte de dados para grupos de hosts. Este campo é de preenchimento automático, portanto, ao começar a digitar o nome de um grupo, será exibida uma lista suspensa de grupos correspondentes. Selecionar um grupo de hosts pai seleciona implicitamente todos os grupos de hosts aninhados; se nenhum grupo de hosts for selecionado, o widget exibirá todos os hosts de todos os grupos de hosts.
Padrões de host	Este parâmetro não está disponível ao configurar o widget em um dashboard de template. Digite padrões de host ou selecione hosts existentes como padrões de host. Hosts que correspondam aos padrões especificados serão exibidos no host navigator. Este campo é de preenchimento automático, portanto, ao começar a digitar o nome de um host, será exibida uma lista suspensa de hosts correspondentes. Se nenhum host for selecionado, o widget exibirá todos os hosts. Padrões curinga podem ser usados para seleção (por exemplo, ∗ retornará hosts que correspondam a zero ou mais caracteres; `Zabbix∗` retornará hosts que comecem com "Zabbix"). Para especificar um padrão curinga, digite a string manualmente e pressione *Enter*. Ao começar a digitar, uma lista suspensa mostrará hosts correspondentes limitados àqueles pertencentes a hosts dentro dos *Grupos de hosts* selecionados, se houver.
Status do host	Este parâmetro não está disponível ao configurar o widget em um dashboard de template. Filtre quais hosts exibir com base em seu status (qualquer, habilitado, desabilitado). Este parâmetro não está disponível ao configurar o widget em um dashboard de template.

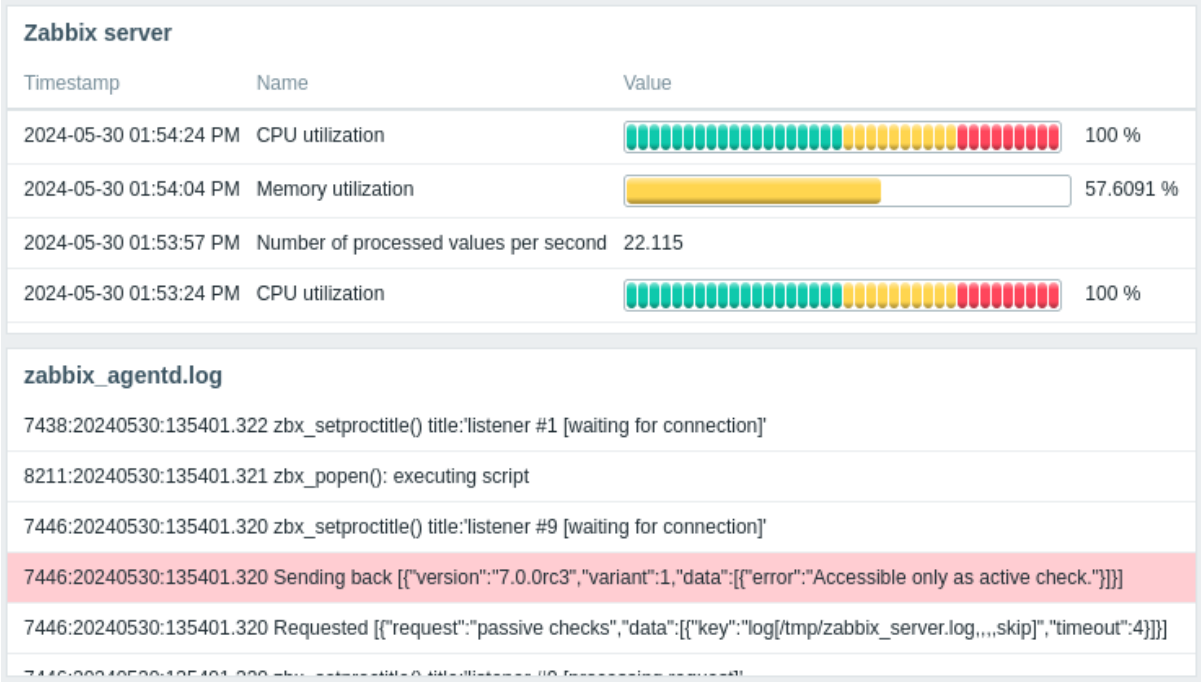
Tags do host	Especifique tags para filtrar os hosts exibidos no widget. É possível incluir e também excluir tags e valores de tags específicos. Várias condições podem ser definidas. A correspondência do nome da tag diferencia maiúsculas de minúsculas. Há vários operadores disponíveis para cada condição: **Existe** - inclui os nomes de tags especificados; **Igual** - inclui os nomes e valores de tags especificados (diferencia maiúsculas de minúsculas); **Contém** - inclui os nomes de tags especificados onde os valores das tags contêm a string inserida (correspondência de substring, não diferencia maiúsculas de minúsculas); **Não existe** - exclui os nomes de tags especificados; **Não é igual** - exclui os nomes e valores de tags especificados (diferencia maiúsculas de minúsculas); **Não contém** - exclui os nomes de tags especificados onde os valores das tags contêm a string inserida (correspondência de substring, não diferencia maiúsculas de minúsculas). Existem dois tipos de cálculo para condições: **E/Ou** - todas as condições devem ser atendidas, condições com o mesmo nome de tag serão agrupadas pela condição Ou; **Ou** - basta que uma condição seja atendida. Este parâmetro não está disponível ao configurar o widget em um dashboard de template.
Severidade	Marque as severidades de problema para filtrar os hosts com problemas a serem exibidos no widget. Se nenhuma severidade for marcada, todos os hosts com todos os problemas serão exibidos.
Exibir hosts em manutenção	Marque esta caixa de seleção para exibir hosts em manutenção (neste caso, o ícone de manutenção será exibido ao lado do nome do host). Este parâmetro é rotulado como *Exibir dados em manutenção* ao configurar o widget em um dashboard de template.
Exibir problemas	Filtre quais problemas exibir com os hosts no widget com base em seu status (todos, não suprimidos, nenhum).
Agrupar por	Adicione um atributo de agrupamento pelo qual agrupar os hosts selecionados: **Grupo de hosts** - agrupa hosts por seu grupo de hosts; **Valor da tag** - insira um nome de tag para agrupar hosts pelos valores dessa tag (por exemplo, insira ”cidade” para agrupar hosts pelos valores ”Riga”, ”Tóquio”, etc.); **Severidade** - agrupa hosts por suas severidades de problema. Se *Exibir problemas* estiver configurado para exibir problemas, eles serão exibidos da seguinte forma: - para cada grupo de severidade, apenas a contagem de problemas correspondente é exibida; - para cada host, todas as suas contagens de problemas são exibidas. Observe que os hosts serão agrupados apenas pelas severidades marcadas no parâmetro *Severidade*; se nenhuma severidade for marcada, todos os hosts serão agrupados por todas as severidades. Os atributos de agrupamento podem ser reordenados arrastando para cima ou para baixo pela alça antes do nome do grupo. Observe que a ordem dos atributos de agrupamento determina a ordem de aninhamento dos grupos. Por exemplo, especificar vários nomes de tags (1: cor, 2: cidade) resultará em hosts agrupados por cor (vermelho, azul, etc.) e depois por cidade (Riga, Tóquio, etc.). Um host pode ser exibido em vários grupos dependendo dos atributos de agrupamento configurados (por exemplo, ao agrupar por grupo de hosts e o host pertencer a vários grupos de hosts). Clicar nesses hosts os seleciona e destaca em todos os grupos. Hosts que não correspondem aos atributos de agrupamento configurados são exibidos no grupo *Sem categoria*. Até 10 atributos de agrupamento podem ser especificados, e todos devem ser exclusivos. Se nenhum atributo de agrupamento for especificado, os hosts não serão agrupados.

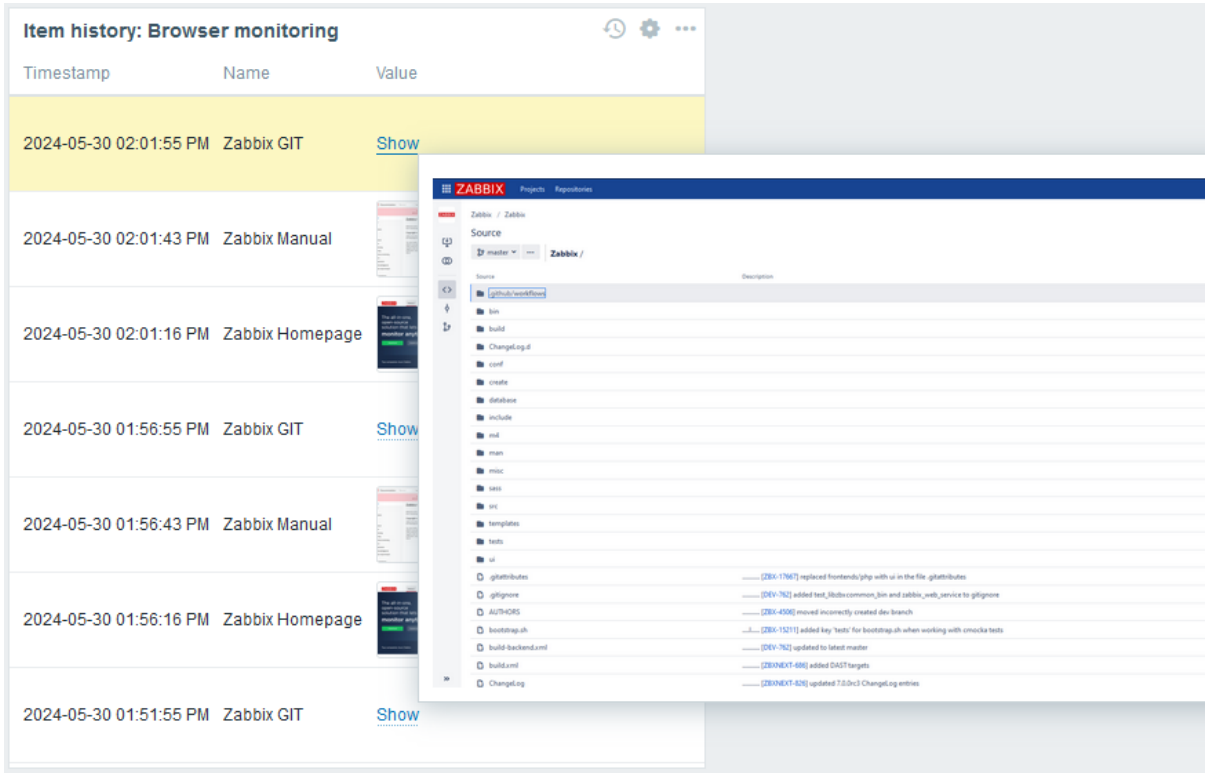
Limite de hosts	Digite o número máximo de hosts a serem exibidos. Os valores possíveis variam de 1 a 9999. Quando houver mais hosts disponíveis para exibição do que o limite definido, uma mensagem correspondente será exibida abaixo dos hosts exibidos (por exemplo, ”100 de 100+ hosts são exibidos”). Observe que o limite de hosts configurado também afeta a exibição dos grupos configurados; por exemplo, se o limite de hosts for definido como 100 e os hosts forem agrupados por valores de tags (mais de 200), apenas os primeiros 100 valores de tags com os hosts correspondentes serão exibidos no widget. Este parâmetro não é afetado pelo parâmetro *Limite para resultados de pesquisa e filtro* em *Administração → Geral → GUI*. Este parâmetro não está disponível ao configurar o widget em um dashboard de template.

15 Histórico de item

Visão geral

O widget de histórico de item exibe os dados mais recentes para vários tipos de item (numérico, caractere, log, texto e binário) em formato de tabela. Ele também pode mostrar barras de progresso, imagens para tipos de dados binários (útil para itens de navegador) e destacar valores (útil para monitoramento de arquivos de log).

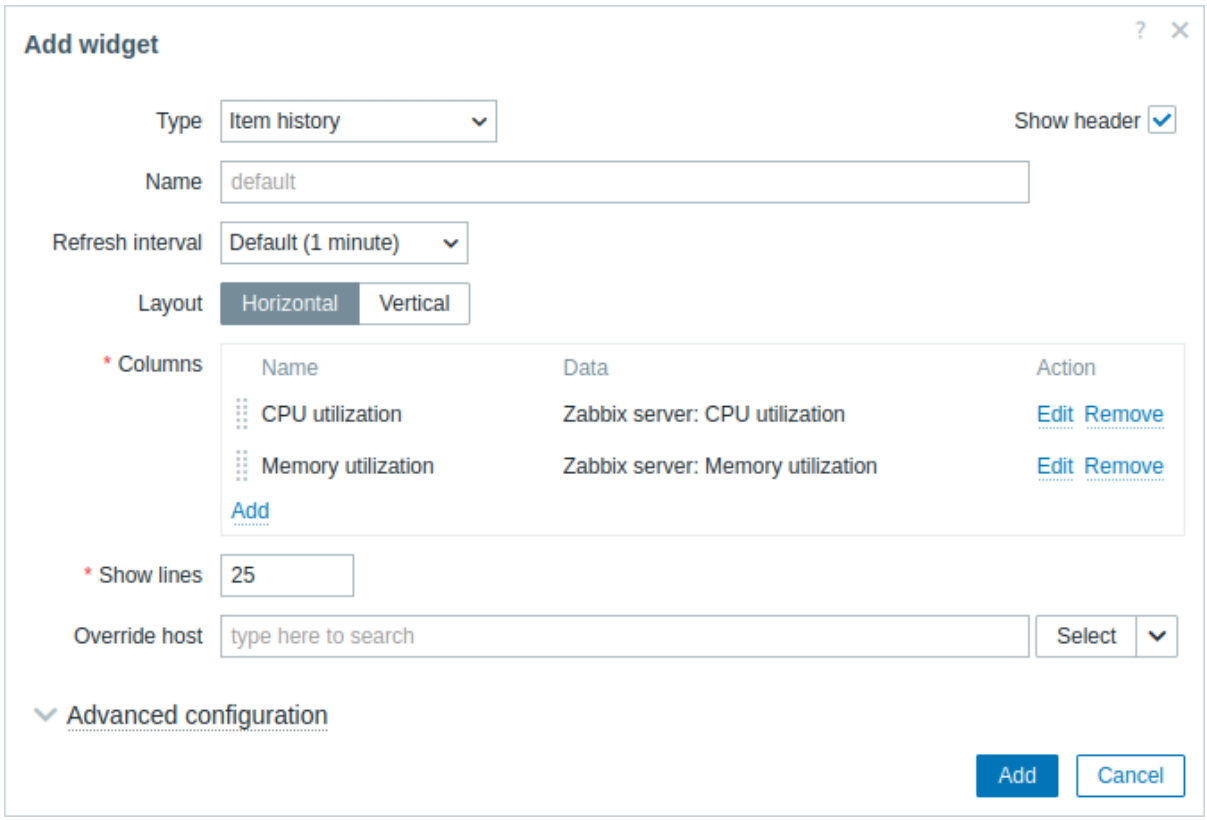




Até 1000 registros podem ser exibidos (100 antes do Zabbix 7.0.6).

Configuração

Para configurar, selecione *Histórico do item* como tipo:



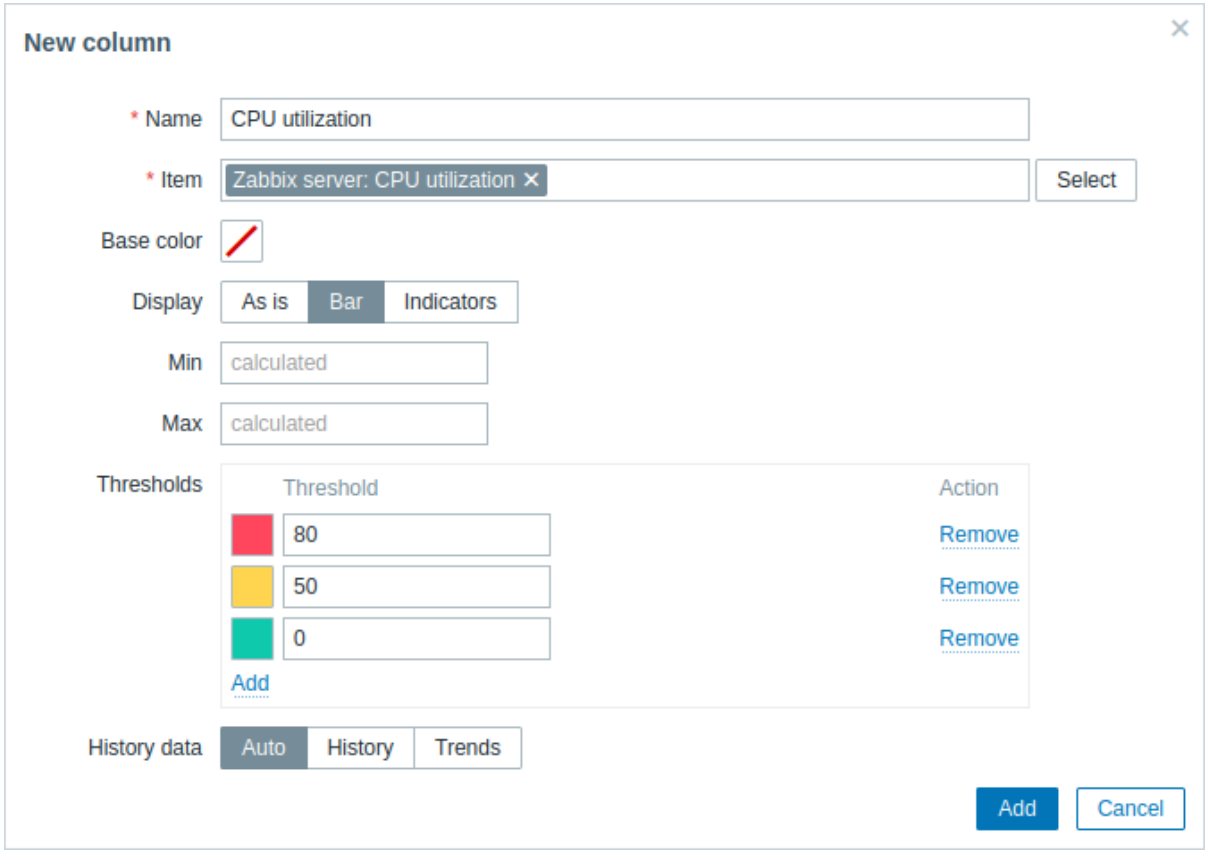
Além dos parâmetros que são comuns para todos os widgets, você pode definir as seguintes opções específicas:

Layout	Selecione a opção de layout para as colunas do item: **Horizontal** - os itens serão exibidos horizontalmente, os valores verticalmente; **Vertical** - os itens serão exibidos verticalmente, os valores horizontalmente.

Colunas	Adicione colunas de dados para exibir. A ordem das colunas determina a ordem de exibição. As colunas podem ser reordenadas arrastando-as para cima ou para baixo pela alça antes do nome da coluna.
Exibir linhas	Especifique o número de linhas de valores do item a serem exibidas.
Substituir host	Selecione um widget compatível ou o dashboard como fonte de dados para hosts. Este parâmetro não está disponível ao configurar o widget em um dashboard de template.
Configuração avançada	Clique no rótulo *Configuração avançada* para exibir as opções de configuração avançada.

Configuração de colunas

Para configurar as colunas, clique em *Adicionar* no parâmetro *Colunas*:



Parâmetros comuns de coluna:

Nome	Insira o nome da coluna. Se deixado em branco, o nome do item do parâmetro *Item* será usado.
Item	Selecione o item. Observe que os parâmetros de configuração da coluna variam com base no tipo de informação do item selecionado; para mais informações, consulte os parâmetros individuais abaixo. Ao configurar o widget em um dashboard de template, apenas itens configurados no template podem ser selecionados.
Cor base	Selecione a cor de fundo da coluna ou a cor de preenchimento se *Exibir* estiver definido como ”Barra” ou ”Indicadores”. Observe que a cor base pode ser substituída por cores de limite ou de destaque.

Parâmetros de coluna específicos para itens do tipo numérico:

Exibir	Selecione como o valor do item deve ser exibido: **Como está** - como texto regular; **Barra** - como barra sólida preenchida com cor; **Indicadores** - como barra segmentada preenchida com cor.

Mín	Insira o valor mínimo para barra/indicadores. Se deixado em branco, o widget usará o valor mínimo do item. Este parâmetro está disponível apenas quando *Exibir* está definido como ”Barra” ou ”Indicadores”.
Máx	Insira o valor máximo para barra/indicadores. Se deixado em branco, o widget usará o valor máximo do item. Este parâmetro está disponível apenas quando *Exibir* está definido como ”Barra” ou ”Indicadores”.
Limiares	Clique em *Adicionar* para adicionar um limiar, selecione uma cor de limiar no seletor de cores e especifique um valor numérico. A lista de limiares será classificada em ordem crescente ao salvar. Sufixos (por exemplo, ”1d”, ”2w”, ”4K”, ”8G”) são suportados. Mapeamentos de valores são suportados.
Dados históricos	Selecione se deseja obter dados do histórico ou das tendências: **Auto** - seleção automática; **Histórico** - obter dados do histórico; **Tendências** - obter dados das tendências.

Parâmetros de coluna específicos para itens do tipo caractere, texto e log:

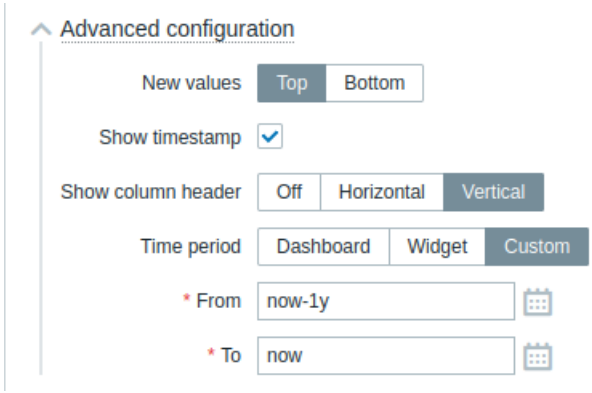
Destaques	Clique em *Adicionar* para adicionar um destaque, selecione uma cor de destaque no seletor de cores e especifique uma expressão regular. A cor selecionada será usada como cor de fundo para os valores do item onde a expressão regular especificada corresponder ao texto.
Exibir	Selecione como o valor do item deve ser exibido: **Como está** - exatamente como recebido. Quebras de linha são preservadas (desde o Zabbix 7.0.5) se o valor contiver retornos de linha. A quebra de linha é aplicada se o *Layout* estiver definido como ”Vertical” e mais de uma coluna estiver configurada; **HTML** - como texto formatado em HTML; **Linha única** - como uma única linha, truncada para um comprimento especificado (1-500 caracteres). Passe o mouse ou clique no valor truncado para abrir um pop-up com o valor completo.
Usar fonte monoespaçada	Marque esta caixa para exibir o valor do item em fonte monoespaçada (desmarcada por padrão).
Exibir hora local	Marque esta caixa para exibir a hora local em vez do timestamp na coluna de timestamp. Observe que a caixa *Exibir timestamp* na configuração avançada também deve ser marcada. Este parâmetro está disponível apenas para itens do tipo log.

Parâmetros de coluna específicos para itens do tipo binário:

Mostrar miniatura	Marque esta caixa de seleção para exibir uma miniatura para binários de imagem ou uma opção ”Mostrar” para binários que não sejam imagens. Desmarque esta caixa de seleção para exibir uma opção ”Mostrar” para todos os valores de itens binários. Passe o mouse ou clique na opção ”Mostrar” para abrir uma janela pop-up com o valor do item (imagem ou string Base64). Se o valor do item for uma string vazia, a opção ”Mostrar” será exibida; passar o mouse ou clicar nela abrirá um pop-up contendo ”String vazia”.

Configuração avançada

As opções de configuração avançada estão disponíveis na seção recolhível *Configuração avançada*:

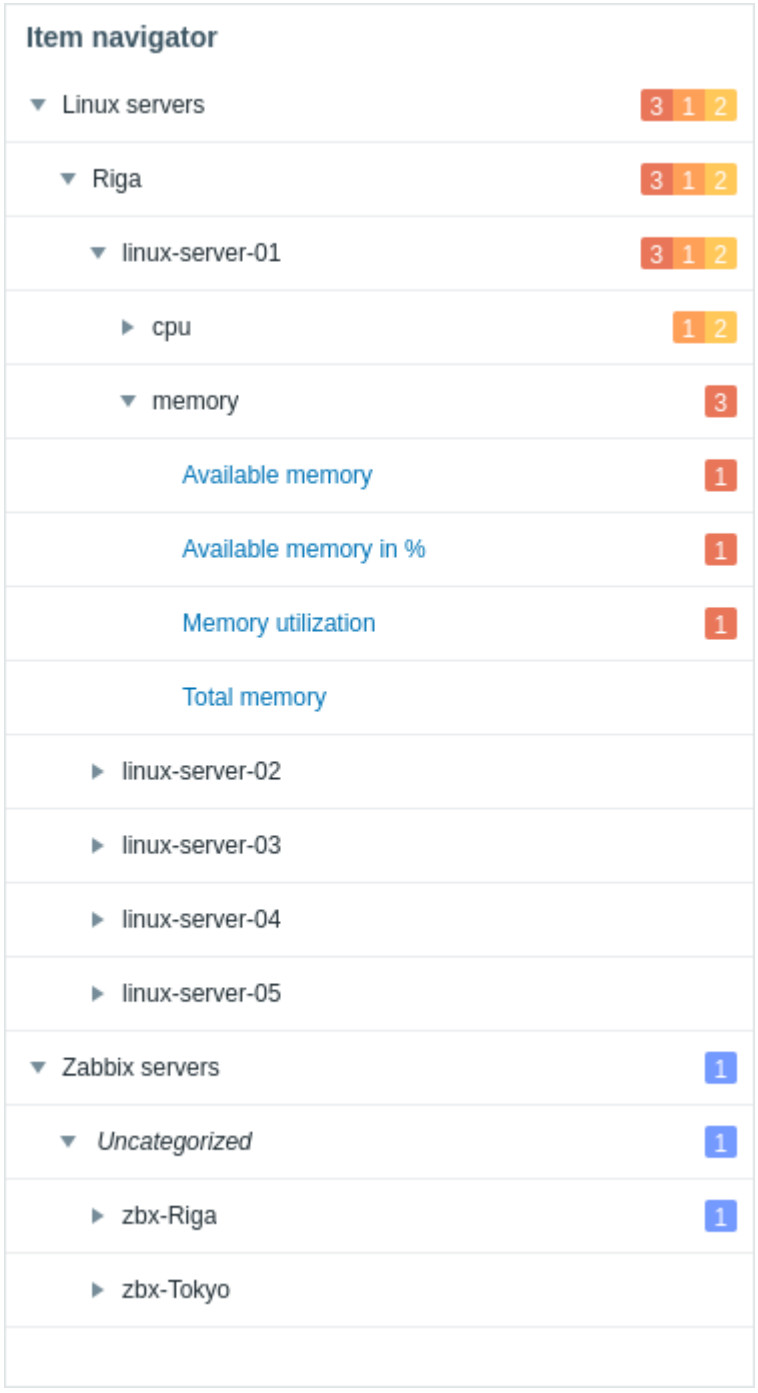


Novos valores	Selecione onde os novos valores do item devem ser adicionados: **Topo** - no topo das colunas; **Fundo** - no final das colunas.
Mostrar data/hora	Marque esta caixa para exibir a coluna de data/hora (desmarcada por padrão).
Mostrar cabeçalho da coluna	Selecione a orientação do cabeçalho da coluna: **Desligado** - ocultar o cabeçalho; **Horizontal** - exibir o cabeçalho horizontalmente; **Vertical** - exibir o cabeçalho verticalmente.
Período de tempo	Selecione a fonte de dados para o período de tempo: **Dashboard** - define o seletor de *Período de tempo* como fonte de dados; **Widget** - define um widget compatível especificado no parâmetro *Widget* como fonte de dados; **Personalizado** - define o período de tempo especificado nos parâmetros *De* e *Até* como fonte de dados; se definido, um ícone de relógio será exibido no canto superior direito do widget, indicando o tempo definido ao passar o mouse.
Widget	Digite ou selecione um widget compatível (*Gráfico*, *Gráfico (clássico)*, *Protótipo de gráfico*) como fonte de dados para o período de tempo. Este parâmetro está disponível se *Período de tempo* estiver definido como "Widget".
De	Digite ou selecione o início do período de tempo. A sintaxe de tempo relativo (`now`, `now/d`, `now/w-1w`, etc.) é suportada. Este parâmetro está disponível se *Período de tempo* estiver definido como "Personalizado".
Até	Digite ou selecione o final do período de tempo. A sintaxe de tempo relativo (`now`, `now/d`, `now/w-1w`, etc.) é suportada. Este parâmetro está disponível se *Período de tempo* estiver definido como "Personalizado".

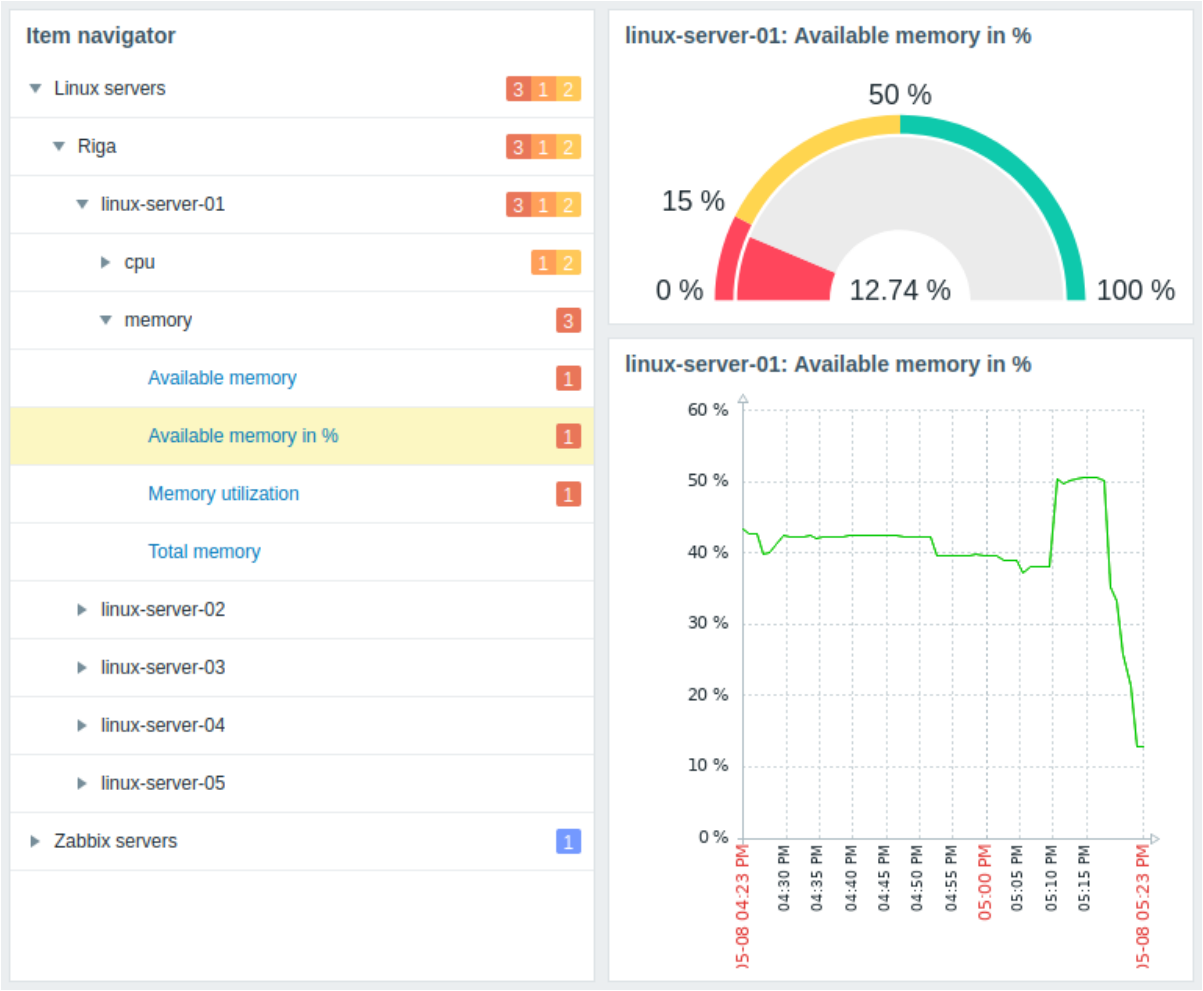
16 Navegador de item

Visão geral

O widget de navegação de item exibe os items com base em várias opções de filtragem e agrupamento.



O widget também permite controlar as informações exibidas em outros widgets com base no item selecionado.

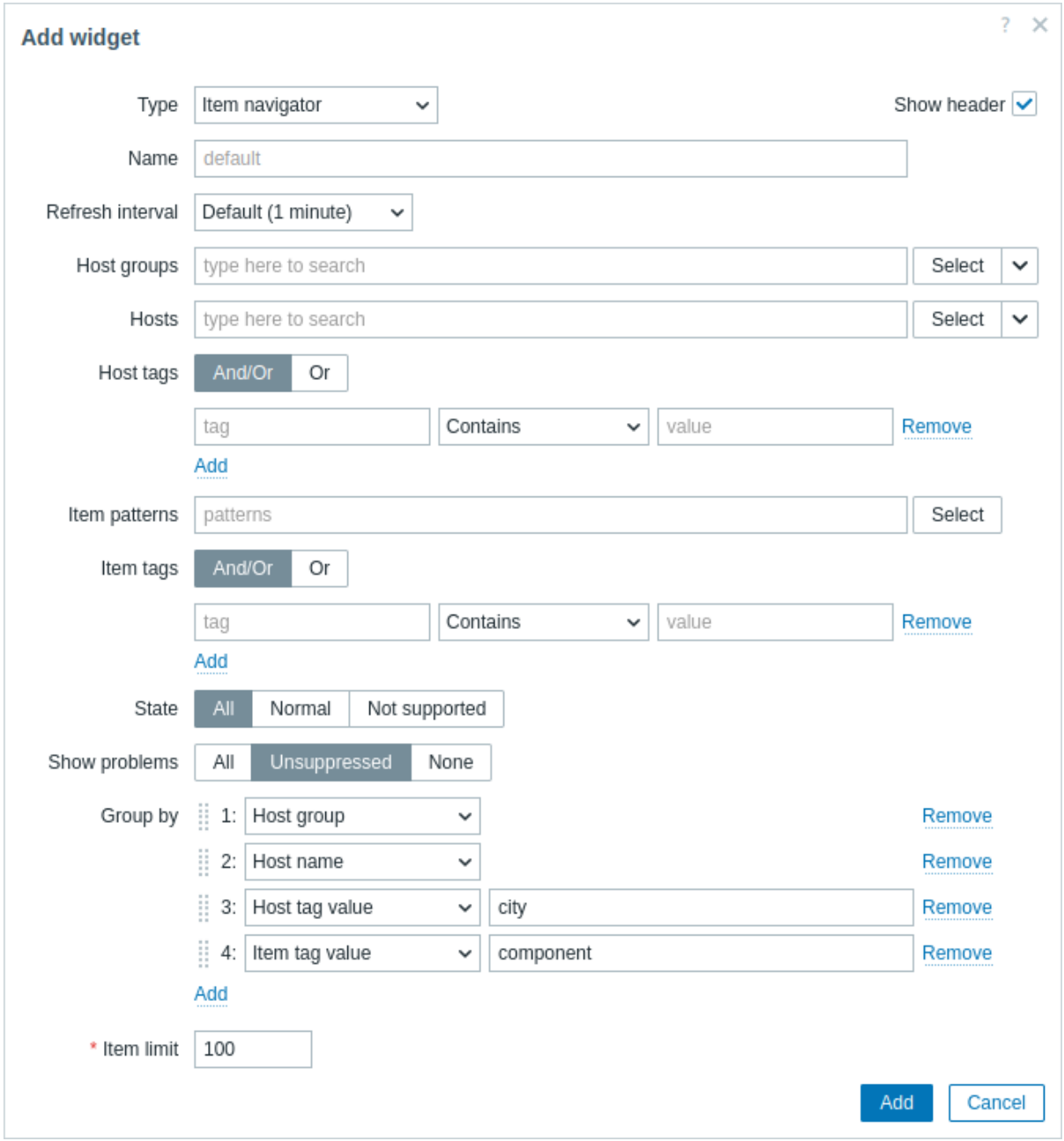


Os grupos pelos quais os items são organizados podem ser expandidos ou recolhidos.

Para grupos e problemas, detalhes adicionais são acessíveis por dicas de mouseover.

Configuração

Para configurar, selecione *Navegador de items* como tipo:



Além dos parâmetros que são comuns para todos os widgets, você pode definir as seguintes opções específicas:

Grupos de hosts	Selecione grupos de hosts. Como alternativa, selecione um widget compatível como fonte de dados para grupos de hosts. Este campo é auto-completável, portanto, ao começar a digitar o nome de um grupo, será exibida uma lista suspensa de grupos correspondentes. Selecionar um grupo de hosts pai seleciona implicitamente todos os grupos de hosts aninhados; se nenhum grupo de hosts for selecionado, o widget exibirá items pertencentes a todos os hosts de todos os grupos de hosts. Este parâmetro não está disponível ao configurar o widget em um dashboard de template.
Hosts	Selecione hosts. Como alternativa, selecione um widget compatível ou o dashboard como fonte de dados para hosts. Este campo é auto-completável, portanto, ao começar a digitar o nome de um host, será exibida uma lista suspensa de hosts correspondentes. Se nenhum host for selecionado, o widget exibirá items pertencentes a todos os hosts. Este parâmetro não está disponível ao configurar o widget em um dashboard de template.

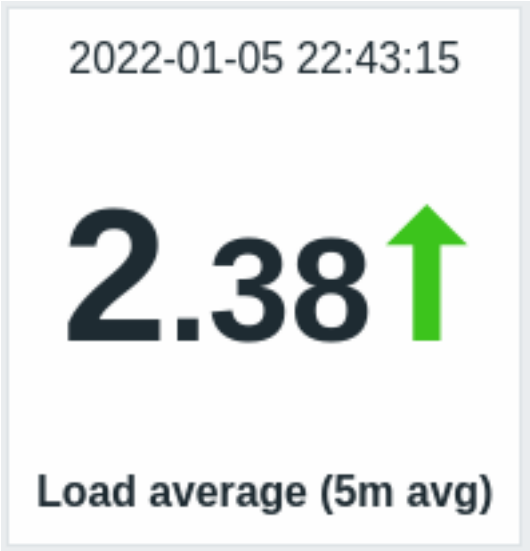
Tags de host	Especifique tags para filtrar os items exibidos no widget. É possível incluir ou excluir tags e valores de tags específicos. Várias condições podem ser definidas. A correspondência do nome da tag diferencia maiúsculas de minúsculas. Existem vários operadores disponíveis para cada condição: **Existe** - inclui os nomes de tags especificados; **Igual** - inclui os nomes e valores de tags especificados (diferencia maiúsculas de minúsculas); **Contém** - inclui os nomes de tags especificados onde os valores das tags contêm a string inserida (correspondência de substring, não diferencia maiúsculas de minúsculas); **Não existe** - exclui os nomes de tags especificados; **Diferente** - exclui os nomes e valores de tags especificados (diferencia maiúsculas de minúsculas); **Não contém** - exclui os nomes de tags especificados onde os valores das tags contêm a string inserida (correspondência de substring, não diferencia maiúsculas de minúsculas). Existem dois tipos de cálculo para as condições: **E/Ou** - todas as condições devem ser atendidas, condições com o mesmo nome de tag serão agrupadas pela condição Ou; **Ou** - basta que uma condição seja atendida.
Padrões de item	Este parâmetro não está disponível ao configurar o widget em um dashboard de template. Insira padrões de item ou selecione items existentes como padrões de item. Os items que corresponderem aos padrões especificados serão exibidos no navegador de items. Padrões curinga podem ser usados para seleção (por exemplo, `*` retornará items que correspondam a zero ou mais caracteres; `Zabbix*` retornará items que começam com "Zabbix"). Para especificar um padrão curinga, insira a string manualmente e pressione *Enter*. Ao começar a digitar, uma lista suspensa mostrará os items correspondentes limitados àqueles pertencentes aos *Hosts* selecionados ou hosts dentro dos *Grupos de hosts* selecionados, se houver. O símbolo curinga é sempre interpretado, portanto, não é possível adicionar, por exemplo, um item chamado *item** individualmente, se houver outros items correspondentes (por exemplo, item2, item3). Ao configurar o widget em um dashboard de template, este parâmetro permite selecionar apenas items configurados no template.
Tags de item	Especifique tags para filtrar os items exibidos no widget. É possível incluir ou excluir tags e valores de tags específicos. Várias condições podem ser definidas. A correspondência do nome da tag diferencia maiúsculas de minúsculas. Existem vários operadores disponíveis para cada condição: **Existe** - inclui os nomes de tags especificados; **Igual** - inclui os nomes e valores de tags especificados (diferencia maiúsculas de minúsculas); **Contém** - inclui os nomes de tags especificados onde os valores das tags contêm a string inserida (correspondência de substring, não diferencia maiúsculas de minúsculas); **Não existe** - exclui os nomes de tags especificados; **Diferente** - exclui os nomes e valores de tags especificados (diferencia maiúsculas de minúsculas); **Não contém** - exclui os nomes de tags especificados onde os valores das tags contêm a string inserida (correspondência de substring, não diferencia maiúsculas de minúsculas). Existem dois tipos de cálculo para as condições: **E/Ou** - todas as condições devem ser atendidas, condições com o mesmo nome de tag serão agrupadas pela condição Ou; **Ou** - basta que uma condição seja atendida.
Estado	Filtra quais items exibir com base em seu estado (todos, normal, não suportado).
Exibir problemas	Filtra quais problemas exibir com os items com base em seu status (todos, não suprimidos, nenhum).

Agrupar por	Adicione um atributo de agrupamento pelo qual agrupar os items: **Grupo de hosts** - agrupa os items pelos grupos de hosts de seus hosts; **Nome do host** - agrupa os items por seus hosts; **Valor da tag do host** - insira um nome de tag para agrupar os items pelos valores dessa tag de host (por exemplo, insira ”cidade” para agrupar os items pelos valores ”Riga”, ”Tóquio”, etc.); **Valor da tag do item** - insira um nome de tag para agrupar os items pelos valores dessa tag de item (por exemplo, insira ”componente” para agrupar os items pelos valores ”cpu”, ”memória”, etc.). Os atributos de agrupamento podem ser reordenados arrastando para cima ou para baixo pela alça antes do nome do grupo. Observe que a ordem dos atributos de agrupamento determina a ordem de aninhamento dos grupos. Por exemplo, especificar vários nomes de tags de host (1: cor, 2: cidade) resultará em items sendo agrupados por cor (vermelho, azul, etc.) e depois por cidade (Riga, Tóquio, etc.). Um item pode ser exibido em vários grupos dependendo dos atributos de agrupamento configurados (por exemplo, ao agrupar por grupo de hosts e o host do item pertencer a vários grupos de hosts). Clicar nesses items os seleciona e destaca em todos os grupos. Items que não correspondem aos atributos de agrupamento configurados são exibidos no grupo *Sem categoria*. Se *Exibir problemas* estiver configurado para exibir problemas, eles serão exibidos da seguinte forma: - para cada grupo, é exibida a contagem de problemas de item subordinados; - para cada item, apenas sua contagem de problemas é exibida. Até 10 atributos de agrupamento podem ser especificados, e todos devem ser exclusivos. Se nenhum atributo de agrupamento for especificado, os items não serão agrupados.
Limite de items	Insira o número máximo de items a serem exibidos. Os valores possíveis variam de 1 a 9999. Quando houver mais items disponíveis para exibição do que o limite definido, uma mensagem correspondente será exibida abaixo dos items exibidos (por exemplo, ”100 de 100+ items são exibidos”). Observe que o limite de items configurado também afeta a exibição dos grupos configurados; por exemplo, se o limite de items for definido como 100 e os items forem agrupados por seus hosts (cada um contendo 200 items), apenas o primeiro host com seus 100 items será exibido no widget. Este parâmetro não é afetado pelo parâmetro *Limite para resultados de pesquisa e filtro* em *Administração → Geral →* GUI.

17 Valor do item

Visão geral

Este widget é útil para exibir o valor de um único item de forma destacada. Pode ser o valor mais recente, bem como um valor agregado para algum período no passado.



Além do valor em si, elementos adicionais podem ser exibidos, se desejado:

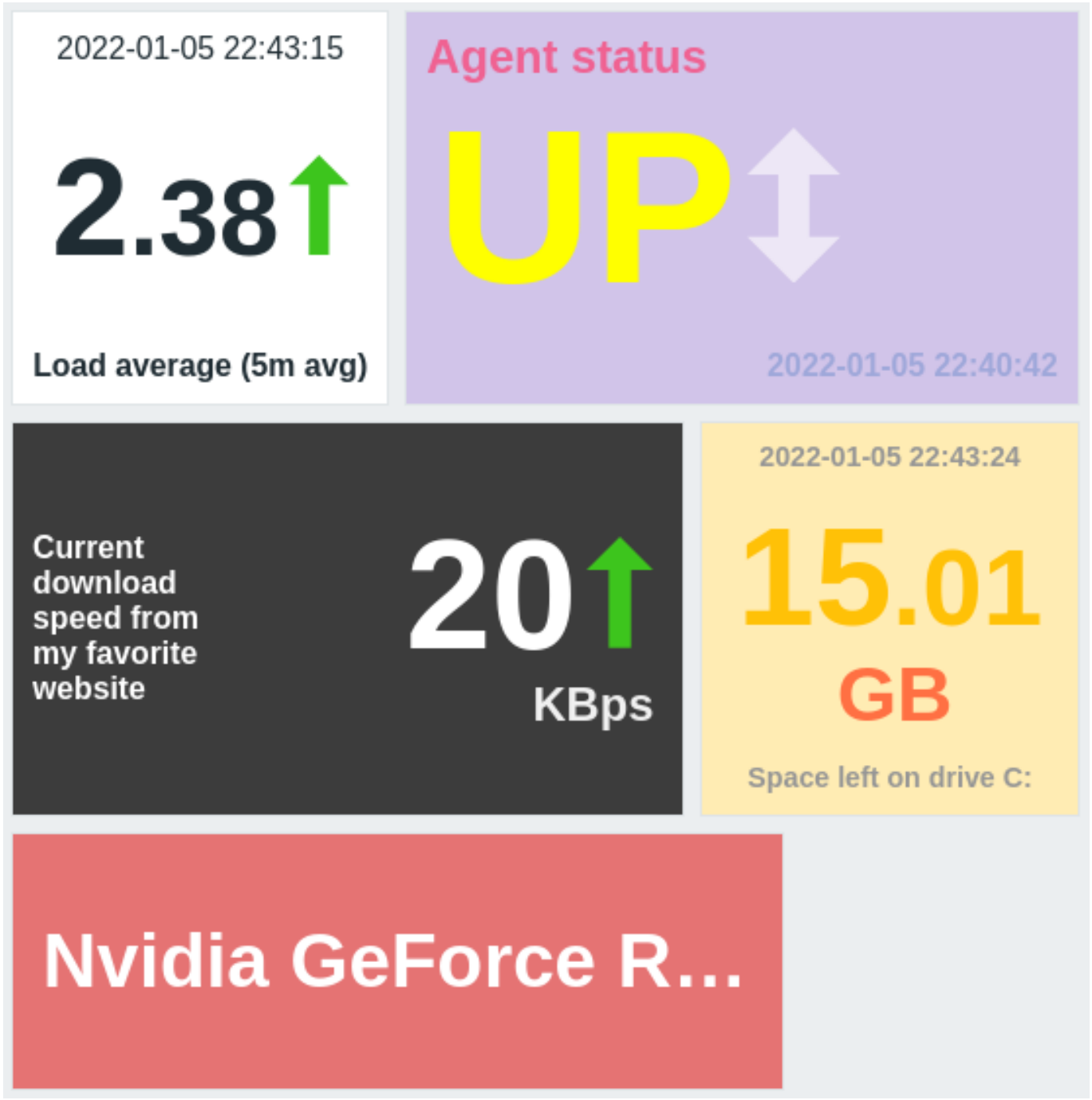
- Hora da métrica
- Descrição do item
- Indicador de alteração para o valor
- Cor de fundo para o valor
- Unidade do item

O widget pode exibir valores numéricos e de string. A exibição de valores binários não é suportada. Valores de string são exibidos em uma única linha e truncados, se necessário. "Sem dados" é exibido se não houver valor para o item.

O indicador de alteração sempre compara com o mesmo período no passado. Assim, por exemplo, o valor mais recente será comparado com o valor anterior, enquanto o último mês será comparado com o mês anterior. Observe que o período anterior para agregações é calculado como um período de tempo do mesmo comprimento que o original, com o horário de término diretamente antes do horário de início do período original.

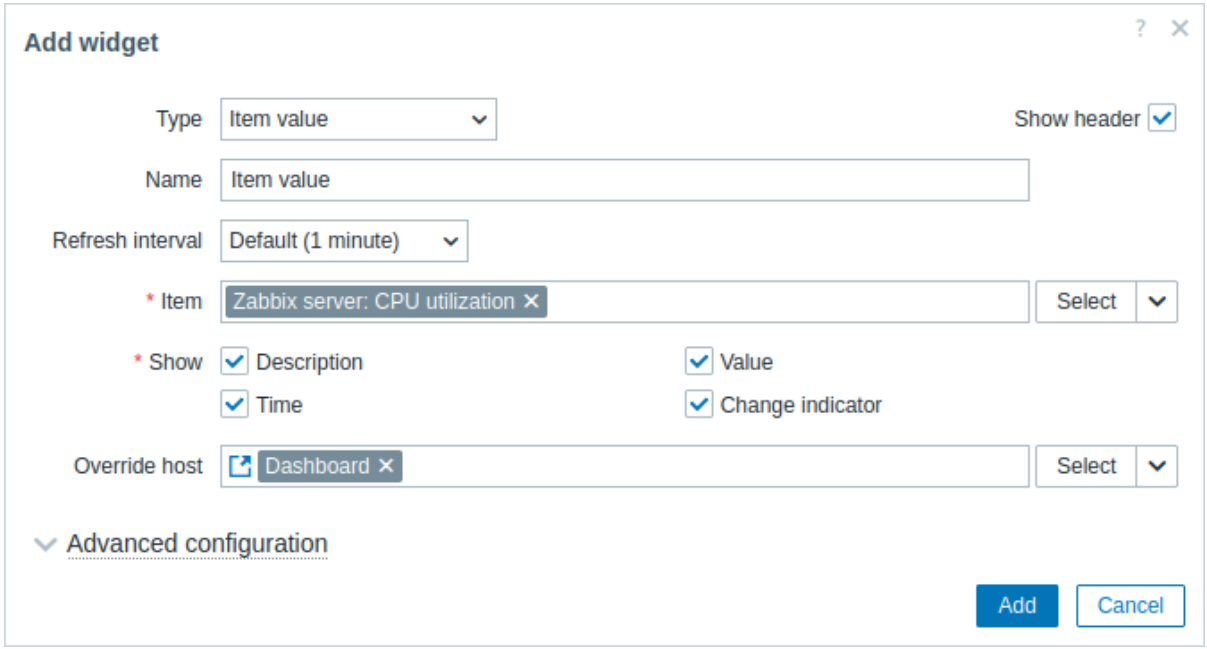
Clicar no valor leva a um gráfico ad-hoc para itens numéricos ou aos dados mais recentes para itens de string.

O widget e todos os elementos nele podem ser ajustados visualmente usando as opções de configuração avançada, permitindo criar uma grande variedade de estilos visuais:



Configuração

Para configurar, selecione *Valor do item* como o tipo de widget:



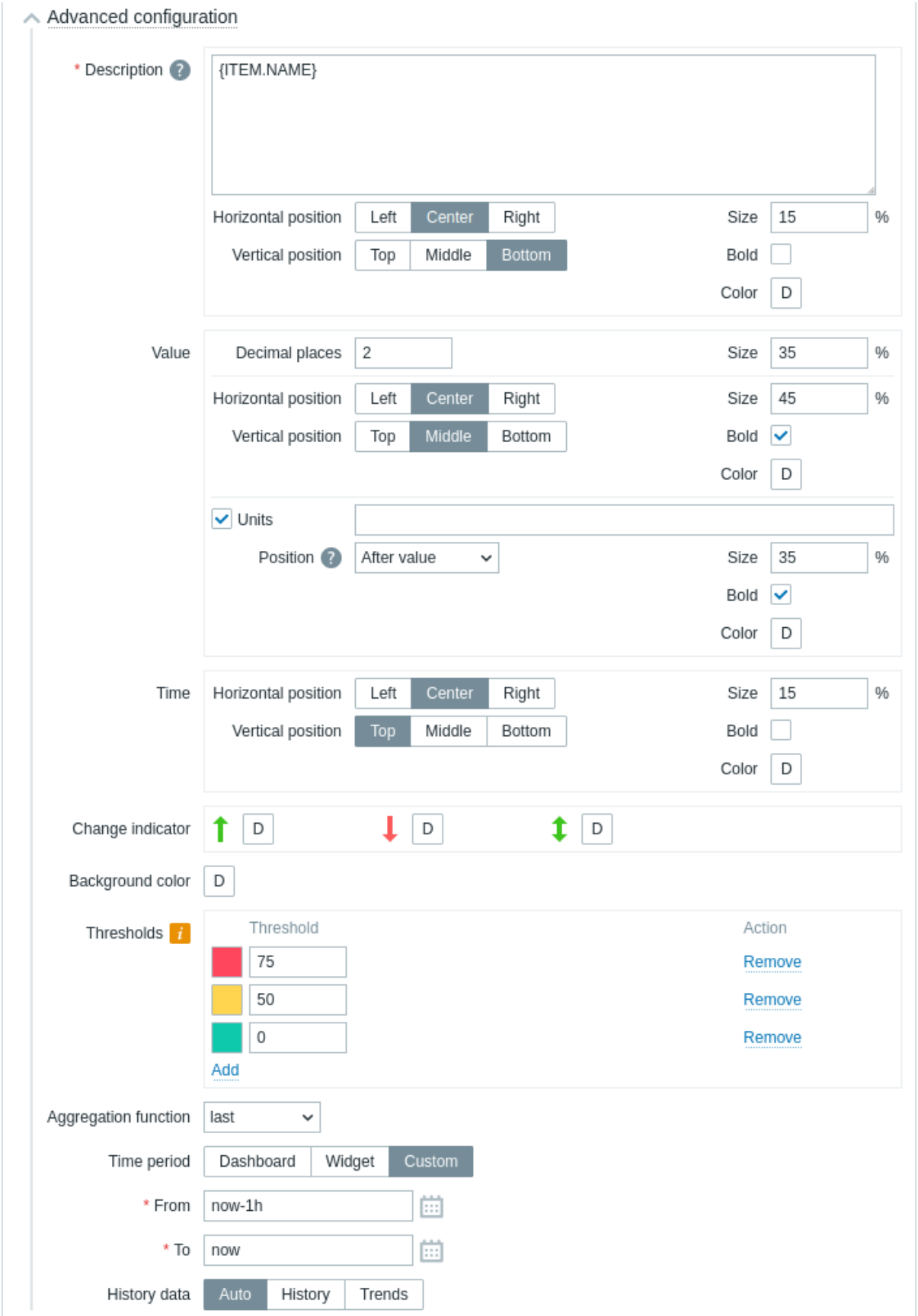
Além dos parâmetros que são comuns para todos os widgets, você pode definir as seguintes opções específicas:

Item	Selecione o item. Como alternativa, selecione um widget compatível como fonte de dados para os itens.
Mostrar	Marque a caixa de seleção para exibir o respectivo elemento (descrição, valor, hora, indicador de alteração). Desmarque para ocultar. Pelo menos um elemento deve ser selecionado.
Substituir host	Selecione um widget compatível ou o dashboard como fonte de dados para hosts. Este parâmetro não está disponível ao configurar o widget em um dashboard de template.
Configuração avançada	Clique no rótulo *Configuração avançada* para exibir as opções de configuração avançada.

Configuração avançada

As opções de configuração avançada estão disponíveis na seção recolhível *Configuração avançada*, e apenas para os elementos que estão selecionados no campo *Exibir* (veja acima).

Além disso, a configuração avançada permite alterar a cor de fundo (estática ou dinâmica) para todo o widget.



Descrição

Descrição	Insira a descrição do item. Esta descrição pode substituir o nome padrão do item. Descrições multilinha são suportadas. É possível uma combinação de texto e macros suportadas. {HOST.*}, {ITEM.*}, {INVENTORY.*} e macros de usuário são suportadas.
Posição horizontal	Selecione a posição horizontal da descrição do item - esquerda, direita ou centralizada.
Posição vertical	Selecione a posição vertical da descrição do item - superior, inferior ou centralizada.
Tamanho	Insira a altura do tamanho da fonte para a descrição do item (em percentual relativo à altura total do widget).

Negrito	Marque a caixa de seleção para exibir a descrição do item em negrito.
Cor	Selecione a cor da descrição do item no seletor de cores. D significa cor padrão (depende do tema do frontend). Para retornar ao valor padrão, clique no botão *Usar padrão* no seletor de cores.
Valor	
Casas decimais	Selecione quantas casas decimais serão exibidas com o valor. Este valor afetará apenas itens do tipo float. Para itens com unidades definidas como ”s”, se este valor for maior que 0 (o padrão é 2), o widget arredonda a unidade de tempo mais significativa e exibe um valor numérico (por exemplo, ”10.43m”) em vez de converter o valor em uma string de tempo completa. Quando definido como 0, os segundos são convertidos em uma string de tempo legível (por exemplo, ”4h 56m 30s”).
Tamanho	Insira a altura do tamanho da fonte para as casas decimais (em percentual relativo à altura total do widget).
Posição horizontal	Selecione a posição horizontal do valor do item - esquerda, direita ou centralizada.
Posição vertical	Selecione a posição vertical do valor do item - superior, inferior ou centralizada.
Tamanho	Insira a altura do tamanho da fonte para o valor do item (em percentual relativo à altura total do widget). Observe que o tamanho do valor do item tem prioridade; outros elementos precisam ceder espaço para o valor. No entanto, com o indicador de alteração, se o valor for muito grande, ele será truncado para mostrar o indicador de alteração.
Negrito	Marque a caixa de seleção para exibir o valor do item em negrito.
Cor	Selecione a cor do valor do item no seletor de cores. D significa cor padrão (depende do tema do frontend). Para retornar ao valor padrão, clique no botão *Usar padrão* no seletor de cores.
Unidades	
Unidades	Marque a caixa de seleção para exibir as unidades com o valor do item. Se você inserir um nome de unidade, ele substituirá a unidade da configuração do item.
Posição	Selecione a posição da unidade do item - acima, abaixo, antes ou depois do valor.
Tamanho	Insira a altura do tamanho da fonte para a unidade do item (em percentual relativo à altura total do widget).
Negrito	Marque a caixa de seleção para exibir a unidade do item em negrito.
Cor	Selecione a cor da unidade do item no seletor de cores. D significa cor padrão (depende do tema do frontend). Para retornar ao valor padrão, clique no botão *Usar padrão* no seletor de cores.
Hora *(valor do relógio do histórico do item)*	
Posição horizontal	Selecione a posição horizontal da hora - esquerda, direita ou centralizada.
Posição vertical	Selecione a posição vertical da hora - superior, inferior ou centralizada.
Tamanho	Insira a altura do tamanho da fonte para a hora (em percentual relativo à altura total do widget).
Negrito	Marque a caixa de seleção para exibir a hora em negrito.
Cor	Selecione a cor da hora no seletor de cores. D significa cor padrão (depende do tema do frontend). Para retornar ao valor padrão, clique no botão *Usar padrão* no seletor de cores.
Indicador de alteração	Selecione a cor dos indicadores de alteração no seletor de cores. Os indicadores de alteração são os seguintes: ↑ - valor do item subiu (para itens numéricos) ↓ - valor do item desceu (para itens numéricos) ↕ - valor do item foi alterado (para itens do tipo string e itens com mapeamento de valor) D significa cor padrão (depende do tema do frontend). Para retornar ao valor padrão, clique no botão *Usar padrão* no seletor de cores. O tamanho vertical do indicador de alteração é igual ao tamanho do valor (parte inteira do valor para itens numéricos). Observe que os indicadores de subida e descida não são exibidos com apenas um valor.
Cor de fundo	Selecione a cor de fundo para todo o widget no seletor de cores. D significa cor padrão (depende do tema do frontend). Para retornar ao valor padrão, clique no botão *Usar padrão* no seletor de cores.
Limiares	Configure a cor de fundo dinâmica para todo o widget. Clique em *Adicionar* para adicionar um limiar, selecione a cor de fundo no seletor de cores e especifique um valor numérico. Assim que o valor do item for igual ou maior que o valor do limiar, a cor de fundo será alterada. A lista será classificada em ordem crescente ao salvar. Observe que a cor de fundo dinâmica será exibida corretamente apenas para itens numéricos.

Função de agregação	Especifique qual função de agregação usar: **min** - exibe o menor valor; **max** - exibe o maior valor; **avg** - exibe o valor médio; **count** - exibe a contagem de valores; **sum** - exibe a soma dos valores; **first** - exibe o primeiro valor; **last** - exibe o último valor; **não usado** - exibe o valor mais recente (sem agregação). A agregação permite exibir um valor agregado para o intervalo escolhido (5 minutos, uma hora, um dia), em vez do valor mais recente. Apenas dados numéricos podem ser exibidos para *min*, *max*, *avg* e *sum*. Para *count*, dados não numéricos serão convertidos em numéricos.
Período de tempo	Especifique o período de tempo a ser usado para agregar valores: **Dashboard** - usa o período de tempo do dashboard; **Widget** - usa o período de tempo do widget especificado; **Personalizado** - usa um período de tempo personalizado. <Este parâmetro não será exibido se a *Função de agregação* estiver definida como "não usado".
Widget	Selecione o widget. Este parâmetro só será exibido se o *Período de tempo* estiver definido como "Widget".
De	Selecione o período de tempo inicial (valor padrão `now-1h`). Veja sintaxe de tempo relativo. Este parâmetro só será exibido se o *Período de tempo* estiver definido como "Personalizado".
Até	Selecione o período de tempo final (valor padrão `now`). Veja sintaxe de tempo relativo. Este parâmetro só será exibido se o *Período de tempo* estiver definido como "Personalizado".
Dados do histórico	Obtenha dados do histórico ou das tendências: **Auto** - seleção automática; **Histórico** - obtém dados do histórico; **Tendências** - obtém dados das tendências. Esta configuração se aplica apenas a dados numéricos. Dados não numéricos sempre serão obtidos do histórico.

Observe que vários elementos não podem ocupar o mesmo espaço; se forem colocados no mesmo espaço, uma mensagem de erro será exibida.

18 Mapa

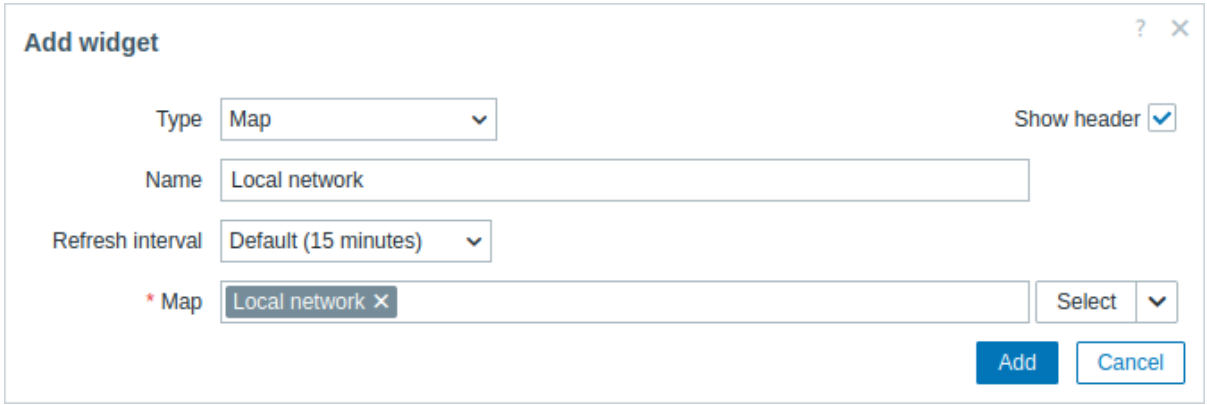
Visão geral

No widget de mapa, você pode exibir:

- um único mapa de rede configurado;
- um dos mapas de rede configurados na árvore de navegação do mapa (ao clicar no nome do mapa na árvore).

Configuração

Para configurar, selecione *Mapa* como tipo:



Além dos parâmetros que são comuns para todos os widgets, você pode definir as seguintes opções específicas:

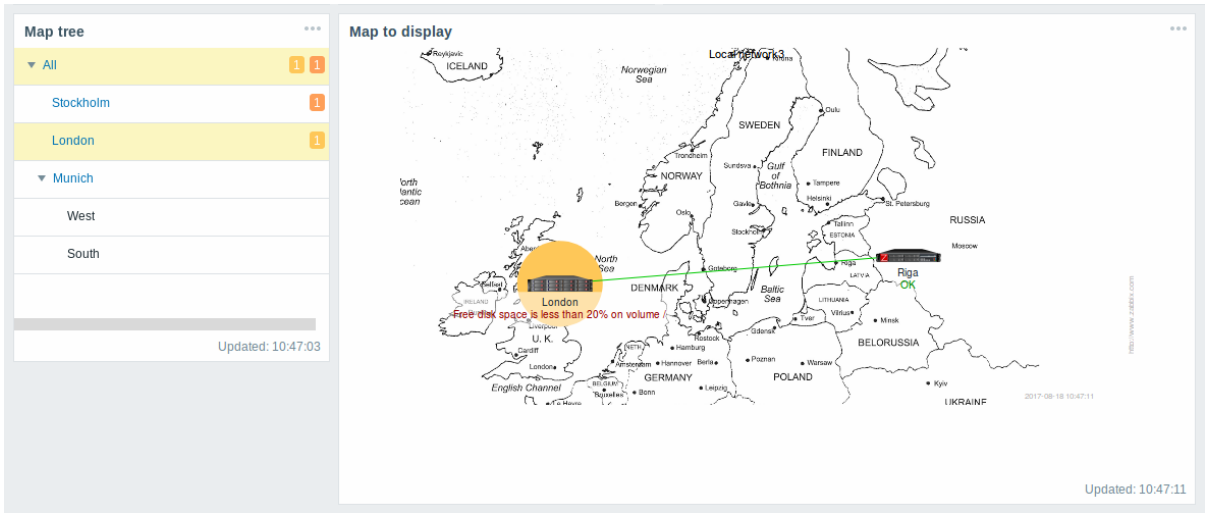
Mapa	Defina um mapa para exibir. Como alternativa, selecione um widget compatível como fonte de dados para o mapa a ser exibido. Este campo é auto-completável, portanto, ao começar a digitar o nome do mapa ou widget, será exibida uma lista suspensa com mapas ou widgets correspondentes.

19 Árvore de navegação do mapa

Visão geral

Este widget permite construir uma hierarquia de mapas existentes enquanto também exibe estatísticas de problemas com cada mapa e grupo de mapas incluídos.

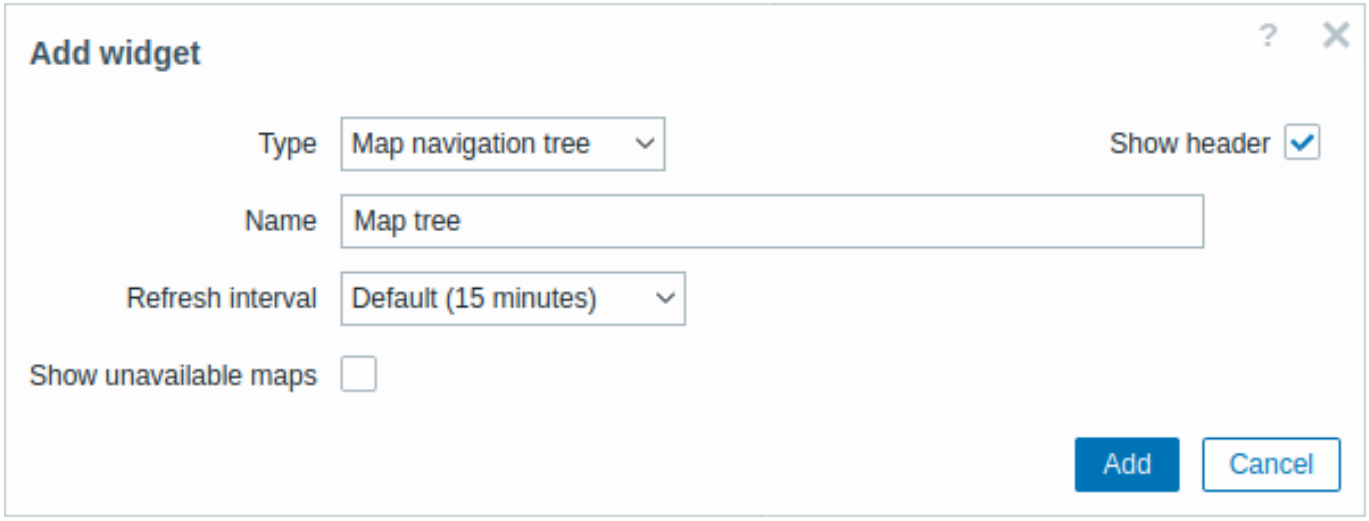
Ele se torna ainda mais poderoso se você vincular o widget *Mapa* à árvore de navegação. Nesse caso, clicar no nome de um mapa na árvore de navegação exibe o mapa completo no widget *Mapa*.



As estatísticas com o mapa de nível superior na hierarquia exibem uma soma dos problemas de todos os submapas e seus próprios problemas.

Configuração

Para configurar, selecione *Árvore de navegação de mapas* como tipo:

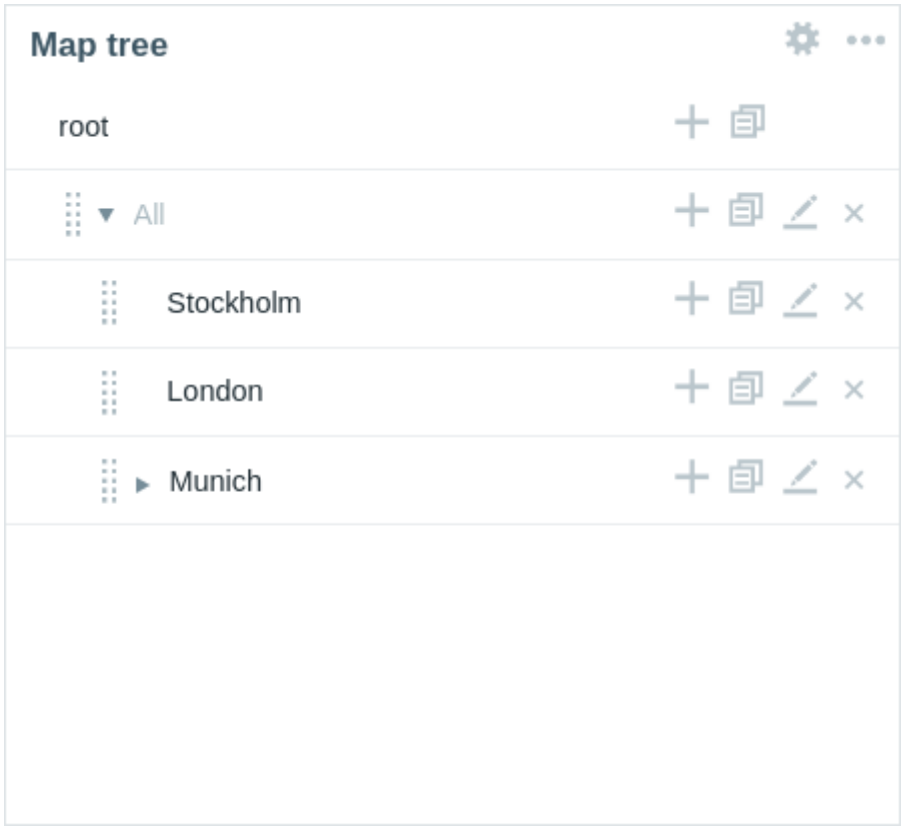


Além dos parâmetros que são comuns para todos os widgets, você pode definir as seguintes opções específicas:

Mostrar mapas indisponíveis	Marque esta caixa de seleção para exibir mapas para os quais o usuário não tem permissão de leitura. Mapas indisponíveis na árvore de navegação serão exibidos com um ícone acinzentado. Observe que, se esta caixa de seleção estiver marcada, submapas disponíveis serão exibidos mesmo que o mapa de nível superior esteja indisponível. Se desmarcada, submapas disponíveis para um mapa pai indisponível não serão exibidos. A contagem de problemas é calculada com base nos mapas disponíveis e nos elementos de mapa disponíveis.

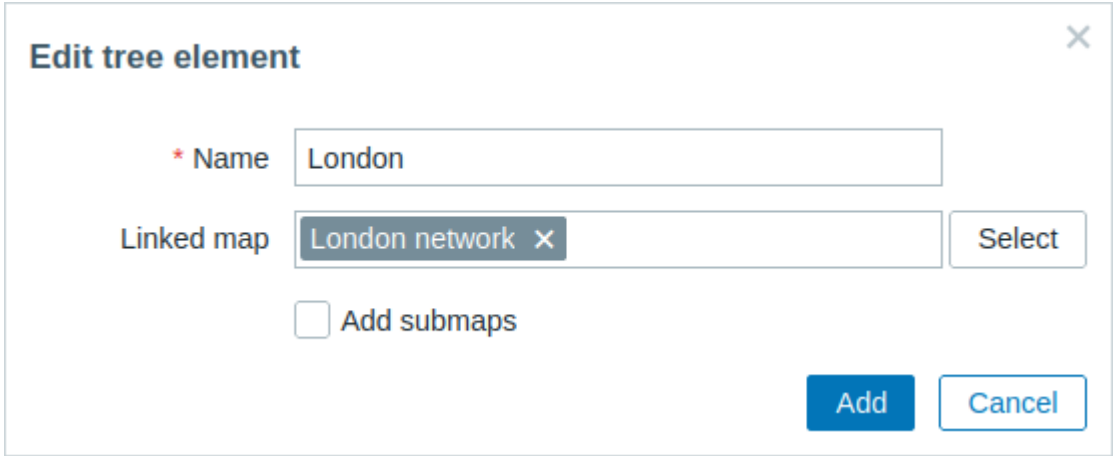
Os elementos da árvore de navegação são exibidos em uma lista. Você pode:

- arrastar um elemento (incluindo seus elementos filhos) para um novo lugar na lista;
- expandir ou recolher um elemento para exibir ou ocultar seus elementos filhos;
- adicionar um elemento filho (com ou sem um mapa vinculado) a um elemento;
- adicionar vários elementos filhos (com mapas vinculados) a um elemento;
- editar um elemento;
- remover um elemento (incluindo seus elementos filhos).



Configuração do elemento

Para configurar um elemento da árvore de navegação, adicione um novo elemento ou edite um elemento existente.



Os seguintes parâmetros de configuração do elemento da árvore de navegação estão disponíveis:

Nome	Insira o nome do elemento da árvore de navegação.
Mapa vinculado	Selecione o mapa para vincular ao elemento da árvore de navegação. Este campo é de preenchimento automático, portanto, ao começar a digitar o nome de um mapa, será exibida uma lista suspensa de mapas correspondentes.
Adicionar submapas	Marque esta caixa de seleção para adicionar os submapas do mapa vinculado como elementos filhos ao elemento da árvore de navegação.

20 Gráfico de pizza

Visão geral

O widget de gráfico de pizza permite exibir valores de itens selecionados como um gráfico de **pizza** ou **rosca**.

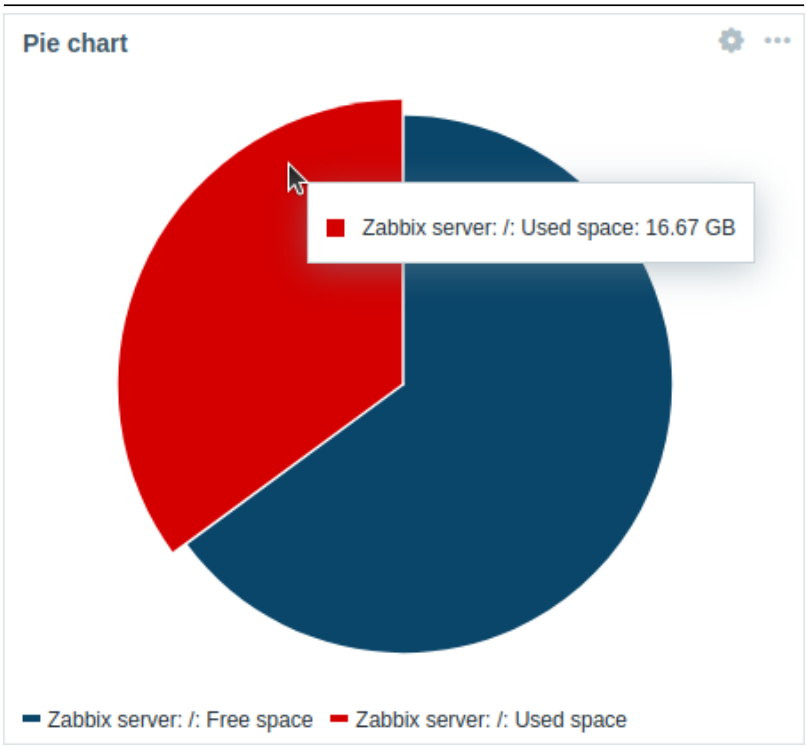


Gráfico de pizza.

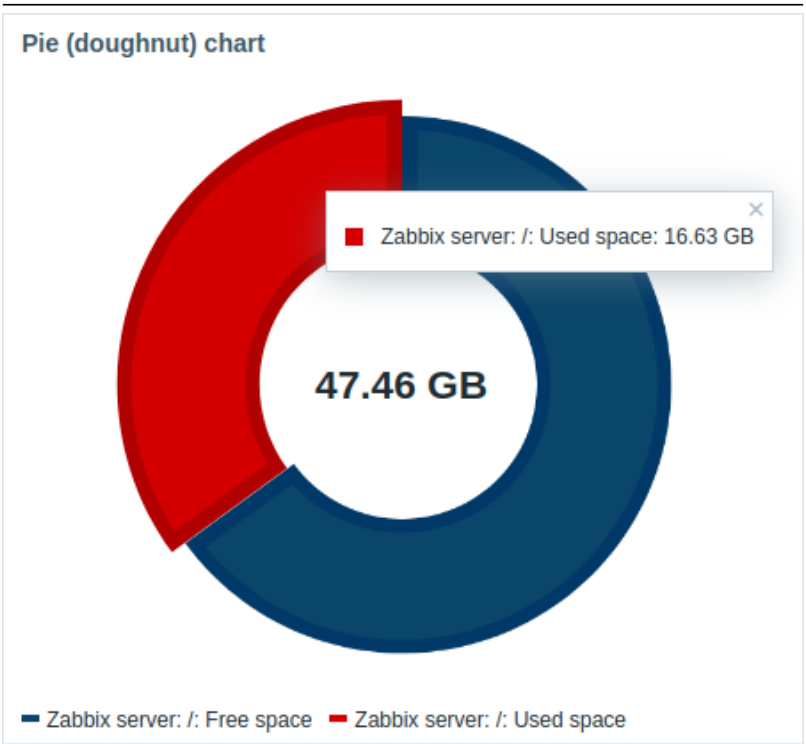


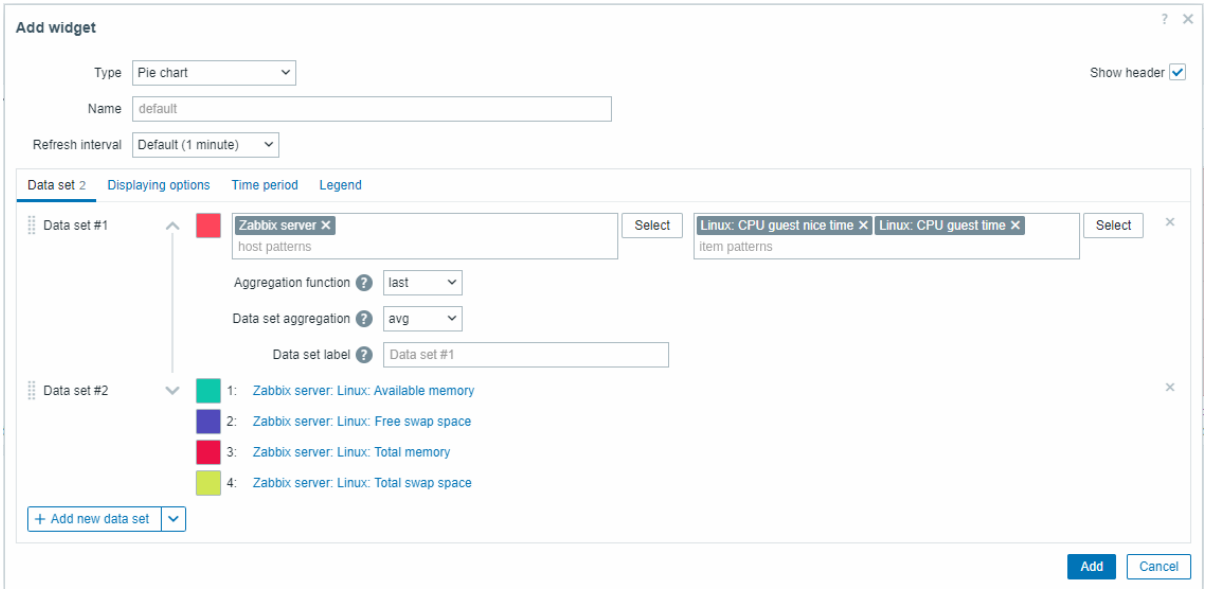
Gráfico de rosca.

Ao passar o mouse, o setor em foco se expande para fora e a legenda desse setor é exibida. Clicar no setor em foco torna o efeito de destaque permanente, até ser fechado com ”x”.

Os gráficos são desenhados usando uma técnica de desenho de imagem vetorial.

Configuração

Para configurar, selecione *Gráfico de pizza* como tipo:



Conjunto de dados

A aba **Conjunto de dados** permite selecionar dados para o gráfico de pizza adicionando conjuntos de dados. Dois tipos de conjuntos de dados podem ser adicionados:

- *Padrões de item* - os dados dos itens correspondentes são exibidos. O gráfico é desenhado usando diferentes tons de uma única cor para cada item.
- *Lista de itens* - os dados dos itens selecionados são exibidos. O gráfico é desenhado usando cores diferentes para cada item.

Por padrão, um conjunto de dados *Padrões de item* é adicionado.

Conjunto de dados	Para o conjunto de dados **Padrões de item**: Selecione ou insira padrões de host e de item; os dados dos itens que correspondem aos padrões inseridos serão exibidos no gráfico de pizza; até 50 itens podem ser exibidos. Padrões curinga podem ser usados para seleção (por exemplo, ∗ retornará resultados que correspondam a zero ou mais caracteres). Para especificar um padrão curinga, insira a string manualmente e pressione *Enter*. O símbolo curinga é sempre interpretado, portanto não é possível adicionar, por exemplo, um item chamado *item** individualmente se houver outros itens correspondentes (por exemplo, *item2*, *item3*). Especificar padrões de host e de item é obrigatório para conjuntos de dados "Padrões de item". Veja também: Detalhes da configuração do conjunto de dados. Para o conjunto de dados **Lista de itens**: Selecione itens para o gráfico de pizza clicando no botão *Adicionar item*. O menu suspenso *Tipo* após o nome do item permite selecionar o tipo de exibição para cada item: **Normal** - o valor do item é representado proporcionalmente no gráfico de pizza (padrão); **Total** - o valor do item ocupa todo o gráfico de pizza. Observe que apenas um item "Total" pode existir por gráfico de pizza, e ele será colocado primeiro na legenda do gráfico de pizza. Se um item for definido como "Total", o parâmetro *Agregação do conjunto de dados* (veja abaixo) será desabilitado e definido como "não utilizado". Desde o Zabbix 7.0.1, você também pode selecionar widgets compatíveis como fonte de dados para itens clicando no botão *Adicionar widget*. As mesmas opções se aplicam aos widgets que aos itens individuais. Especificar itens ou widgets é obrigatório para conjuntos de dados "Lista de itens". Veja também: Detalhes da configuração do conjunto de dados. Observe que apenas tipos de itens numéricos são permitidos. Ao configurar o widget em um dashboard de template, o parâmetro para especificar padrões de host não está disponível, e o parâmetro para especificar uma lista de itens permite selecionar apenas os itens configurados no template.
Função de agregação	Especifique qual função de agregação usar para cada item no conjunto de dados: **min** - exibe o menor valor; **max** - exibe o maior valor; **avg** - exibe o valor médio; **sum** - exibe a soma dos valores; **count** - exibe a contagem dos valores; **first** - exibe o primeiro valor; **last** - exibe o último valor (padrão). A agregação permite exibir um valor agregado para o intervalo (5 minutos, uma hora, um dia) selecionado na aba *Período de tempo* ou usado para todo o dashboard.
Agregação do conjunto de dados	Especifique qual função de agregação usar para todo o conjunto de dados: **não utilizado** - sem agregação, os itens são exibidos separadamente (padrão); **min** - exibe o menor valor; **max** - exibe o maior valor; **avg** - exibe o valor médio; **sum** - exibe a soma dos valores; **count** - exibe a contagem dos valores. A agregação permite exibir um valor agregado para o intervalo (5 minutos, uma hora, um dia) selecionado na aba *Período de tempo* ou usado para todo o dashboard.

Rótulo do conjunto de dados	Especifique um rótulo personalizado para o conjunto de dados. O rótulo é exibido na configuração do conjunto de dados e na legenda do gráfico de pizza (para conjuntos de dados agregados). Todos os conjuntos de dados são numerados, incluindo aqueles com um *Rótulo do conjunto de dados* especificado. Se nenhum rótulo for especificado, o conjunto de dados será rotulado automaticamente de acordo com seu número (por exemplo, ”Conjunto de dados #2”, ”Conjunto de dados #3”, etc.). A numeração do conjunto de dados é recalculada após reordenar/arrastar conjuntos de dados. Rótulos de conjuntos de dados muito longos serão encurtados para caber onde forem exibidos (por exemplo, ”Número de proc...”).

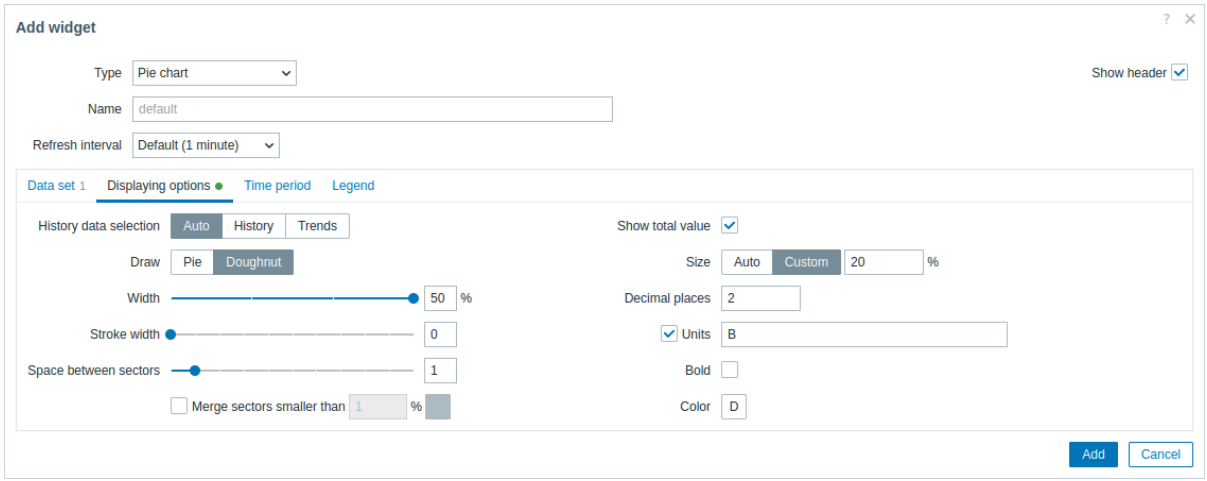
Detalhes da configuração do conjunto de dados

Os conjuntos de dados existentes são exibidos em uma lista. Você pode reorganizar, expandir/recolher, alterar cores e clonar esses conjuntos de dados.

Para mais informações, consulte detalhes da configuração do conjunto de dados no widget *Gráfico*. Esses detalhes também se aplicam ao widget *Gráfico de pizza*.

Opções de exibição

A aba **Opções de exibição** permite definir as opções de seleção e visualização dos dados históricos para o gráfico de pizza:

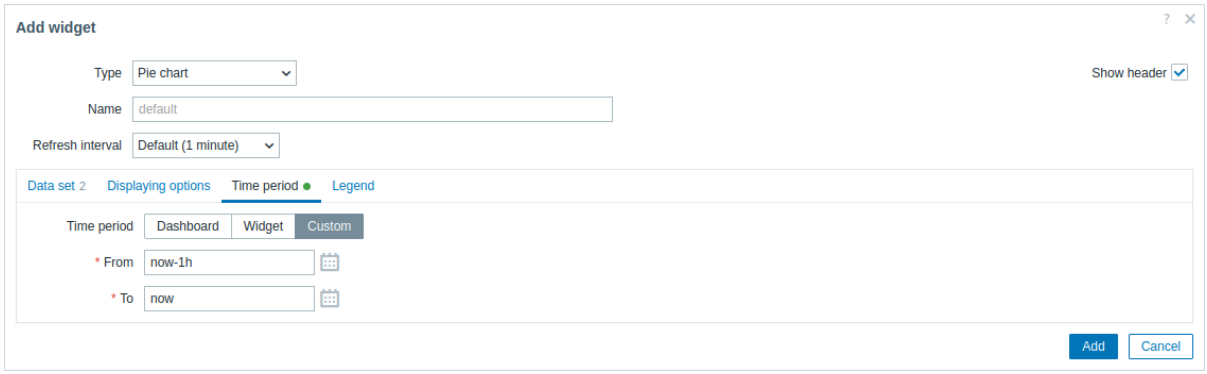


Seleção de dados históricos	Selecione a fonte de dados: **Auto** - os dados são obtidos de acordo com o algoritmo clássico (padrão); **Histórico** - dados do histórico; **Tendências** - dados das tendências.
Desenhar	Selecione o estilo de visualização do gráfico de pizza: **Pizza** - uma pizza completa (os setores ocupam 100% do raio); **Rosquinha** - uma pizza com círculo vazio no meio (os setores usam até 50% do raio).
Espaço entre setores	Selecione o tamanho do espaço (em unidades de 0-10) entre os setores (o padrão é ”1”).
Mesclar setores menores que N%	Marque a caixa de seleção para mesclar setores menores que N%. Se ativado, selecione a cor para os setores mesclados e o limite percentual (N) para mesclar pequenos setores.
Estilo de desenho: ***Rosquinha***	

Largura da borda	Selecione a largura da rosquinha: 20, 30, 40 ou 50% (padrão) do raio. Selecione a largura da borda do setor da rosquinha (0-10). Suportado desde o Zabbix 7.0.1.
Mostrar valor total	Marque a caixa de seleção para mostrar o valor total no meio do gráfico de rosquinha.
Tamanho	Selecione a opção de tamanho para o valor total: **Auto** - o texto é dimensionado automaticamente para caber no meio da rosquinha de forma legível; **Personalizado** - especifique o tamanho do texto como porcentagem da altura total do widget.
Casas decimais	Especifique o número de casas decimais para o valor total (0-6).
Unidades	Especifique as unidades para o valor total.
Negrito	Marque a caixa de seleção para exibir o valor total em negrito.
Cor	Selecione a cor para o valor total.

Período de tempo

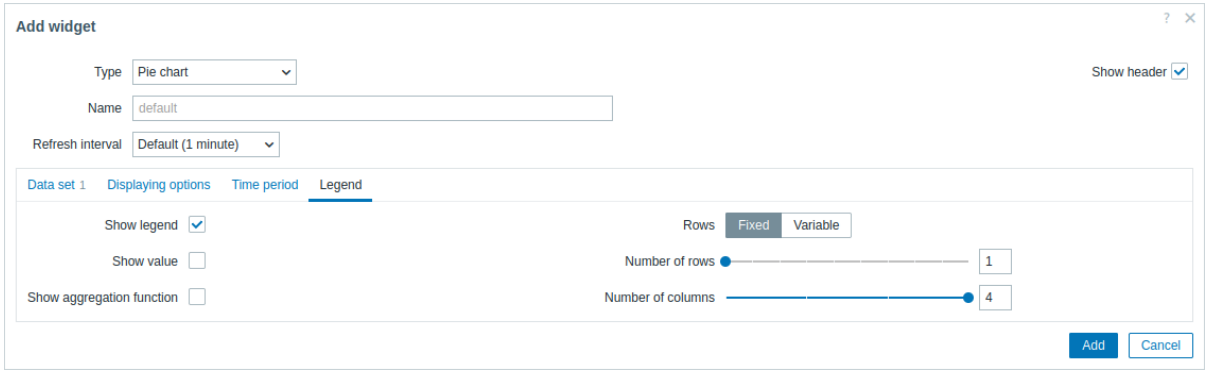
A guia **Período de tempo** permite definir um período de tempo personalizado para as configurações de agregação do gráfico de pizza:



Período de tempo	Selecione a fonte de dados para o período de tempo: **Dashboard** - define o seletor *Período de tempo* como fonte de dados; **Widget** - define um widget compatível especificado no parâmetro *Widget* como fonte de dados; **Personalizado** - define o período de tempo especificado nos parâmetros *De* e *Até* como fonte de dados; se definido, um ícone de relógio será exibido no canto superior direito do widget, indicando o tempo definido ao passar o mouse.
Widget	Digite ou selecione um widget compatível (*Gráfico*, *Gráfico (clássico)*, *Protótipo de gráfico*) como fonte de dados para o período de tempo. Este parâmetro está disponível se *Período de tempo* estiver definido como "Widget".
De	Digite ou selecione o início do período de tempo. Sintaxe de tempo relativo (`now`, `now/d`, `now/w-1w`, etc.) é suportada. Este parâmetro está disponível se *Período de tempo* estiver definido como "Personalizado".
Até	Digite ou selecione o final do período de tempo. Sintaxe de tempo relativo (`now`, `now/d`, `now/w-1w`, etc.) é suportada. Este parâmetro está disponível se *Período de tempo* estiver definido como "Personalizado".

Legenda

A aba **Legenda** permite personalizar a legenda do gráfico de pizza:



Mostrar legenda	Desmarque esta caixa de seleção para ocultar a legenda no gráfico de pizza (marcada por padrão).
Mostrar valor	Marque esta caixa de seleção para mostrar o valor do item na legenda.
Mostrar função de agregação	Marque esta caixa de seleção para mostrar a função de agregação na legenda.
Linhas	Selecione o modo de exibição para as linhas da legenda: **Fixo** - o número de linhas exibidas é determinado pelo valor do parâmetro *Número de linhas*; **Variável** - o número de linhas exibidas é determinado pela quantidade de items configurados, sem exceder o valor do parâmetro *Número máximo de linhas*.
Número de linhas/ Número máximo de linhas	Se *Linhas* estiver definido como ”Fixo”, defina o número de linhas da legenda a serem exibidas (1-10). Se *Linhas* estiver definido como ”Variável”, defina o número máximo de linhas da legenda a serem exibidas (1-10).
Número de colunas	Defina o número de colunas da legenda a serem exibidas (1-4). Este parâmetro está disponível se *Mostrar valor* estiver desmarcado.

As informações exibidas pelo widget de gráfico de pizza podem ser baixadas como uma imagem *.png* usando o menu do widget.

Uma captura de tela do widget será salva na pasta *Downloads*.

21 Hosts com problemas

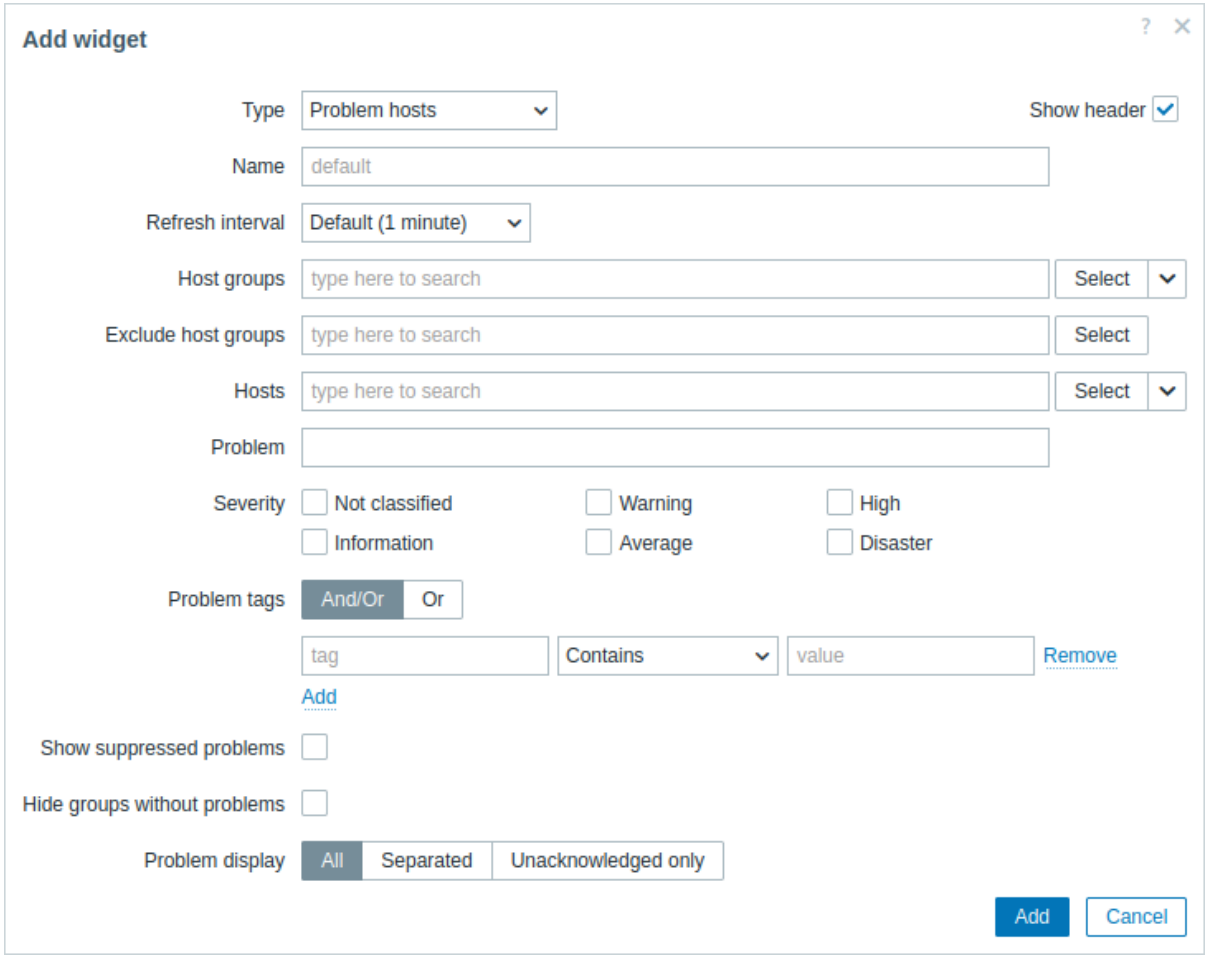
Visão geral

No widget de host com problema, você pode exibir a contagem de problemas por grupo de hosts e a maior gravidade de problema dentro de um grupo.

A contagem de problemas é exibida apenas para problemas de causa.

Configuração

Para configurar, selecione *Hosts com problemas* como tipo:



Além dos parâmetros que são comuns para todos os widgets, você pode definir as seguintes opções específicas:

Grupos de hosts	Selecione os grupos de hosts a serem exibidos no widget. Como alternativa, selecione um widget compatível como fonte de dados para grupos de hosts. Este campo é de preenchimento automático, então ao começar a digitar o nome de um grupo será exibida uma lista suspensa de grupos correspondentes. Especificar um grupo de hosts pai seleciona implicitamente todos os grupos de hosts aninhados. Os dados dos hosts desses grupos de hosts serão exibidos no widget; se nenhum grupo de hosts for inserido, todos os grupos de hosts serão exibidos. Este parâmetro não está disponível ao configurar o widget em um dashboard de template.
Excluir grupos de hosts	Selecione os grupos de hosts a serem ocultados do widget. Este campo é de preenchimento automático, então ao começar a digitar o nome de um grupo será exibida uma lista suspensa de grupos correspondentes. Especificar um grupo de hosts pai seleciona implicitamente todos os grupos de hosts aninhados. Os dados dos hosts desses grupos de hosts não serão exibidos no widget. Por exemplo, os hosts 001, 002, 003 podem estar no Grupo A e os hosts 002, 003 também no Grupo B. Se selecionarmos *exibir* o Grupo A e *excluir* o Grupo B ao mesmo tempo, apenas os dados do host 001 serão exibidos no dashboard. Este parâmetro não está disponível ao configurar o widget em um dashboard de template.
Hosts	Selecione os hosts a serem exibidos no widget. Como alternativa, selecione um widget compatível ou o dashboard como fonte de dados para hosts. Este campo é de preenchimento automático, então ao começar a digitar o nome de um host será exibida uma lista suspensa de hosts correspondentes. Se nenhum host for inserido, todos os hosts serão exibidos. Este parâmetro não está disponível ao configurar o widget em um dashboard de template.
Problema	Você pode limitar o número de hosts com problemas exibidos pelo nome do problema. Se você inserir uma string aqui, apenas aqueles hosts com problemas cujo nome contenha a string inserida serão exibidos. Macros não são expandidas.
Severidade	Marque as severidades dos problemas para filtrar os problemas a serem exibidos no widget. Se nenhuma severidade for marcada, todos os problemas serão exibidos.

Tags do problema	Especifique tags de problema para limitar o número de problemas exibidos no widget. É possível incluir ou excluir tags e valores de tags específicos. Várias condições podem ser definidas. A correspondência do nome da tag diferencia maiúsculas de minúsculas. Há vários operadores disponíveis para cada condição: **Existe** - inclui os nomes de tags especificados; **Igual** - inclui os nomes e valores de tags especificados (diferencia maiúsculas de minúsculas); **Contém** - inclui os nomes de tags especificados onde os valores das tags contêm a string inserida (correspondência de substring, não diferencia maiúsculas de minúsculas); **Não existe** - exclui os nomes de tags especificados; **Diferente** - exclui os nomes e valores de tags especificados (diferencia maiúsculas de minúsculas); **Não contém** - exclui os nomes de tags especificados onde os valores das tags contêm a string inserida (correspondência de substring, não diferencia maiúsculas de minúsculas). Existem dois tipos de cálculo para as condições: **E/Ou** - todas as condições devem ser atendidas, condições com o mesmo nome de tag serão agrupadas pela condição *Ou*; **Ou** - basta que uma condição seja atendida.
Exibir problemas suprimidos	Marque a caixa de seleção para exibir problemas que, de outra forma, seriam suprimidos (não exibidos) devido à manutenção do host.
Ocultar grupos sem problemas	Marque a opção *Ocultar grupos sem problemas* para ocultar dados de grupos de hosts sem problemas no widget. Este parâmetro não está disponível ao configurar o widget em um dashboard de template.
Exibição do problema	Exibir a contagem de problemas como: **Todos** - a contagem total de problemas será exibida; **Separados** - a contagem de problemas não reconhecidos será exibida separadamente como um número da contagem total de problemas; **Apenas não reconhecidos** - apenas a contagem de problemas não reconhecidos será exibida.

22 Problemas

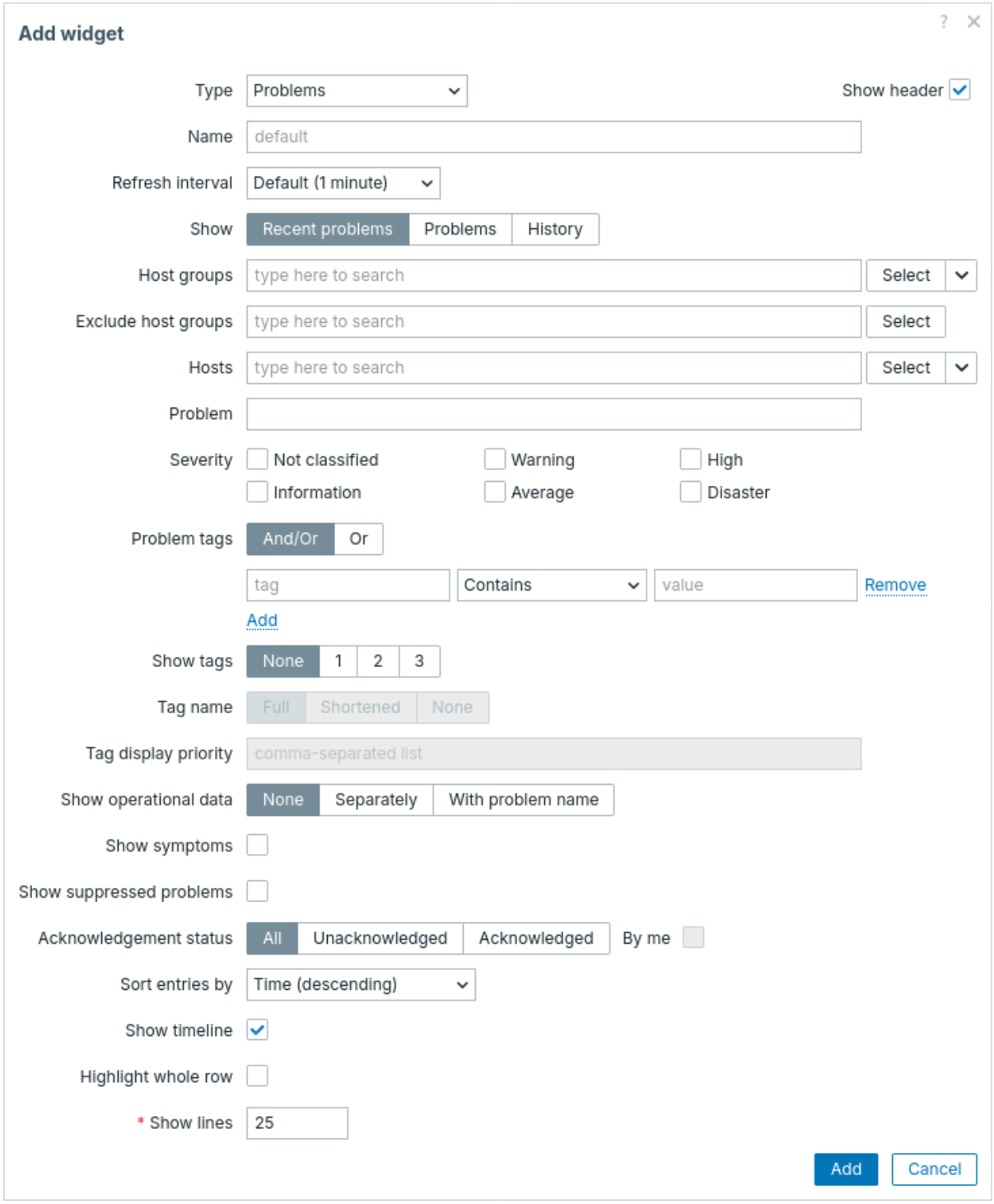
Visão geral

Neste widget, você pode exibir problemas atuais. As informações neste widget são semelhantes a *Monitoramento* > *Problemas*.

Até 1000 registros podem ser exibidos (100 antes do Zabbix 7.0.6).

Configuração

Para configurar, selecione *Problemas* como tipo:



Você pode limitar quantos problemas são exibidos no widget de várias maneiras - por status do problema, nome do problema, severidade, grupo de hosts, host, tag de evento, status de reconhecimento, etc.

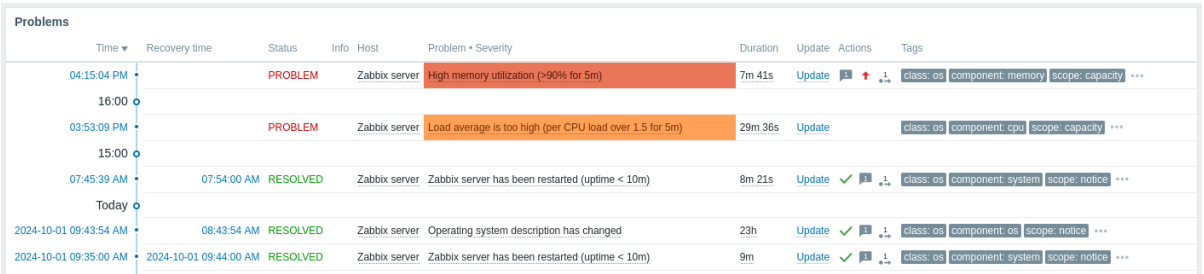
Além dos parâmetros que são comuns para todos os widgets, você pode definir as seguintes opções específicas:

Mostrar	Filtrar por status do problema: **Problemas recentes** - problemas não resolvidos e recentemente resolvidos são exibidos (padrão); **Problemas** - apenas problemas não resolvidos são exibidos; **Histórico** - o histórico de todos os eventos é exibido.

Grupos de hosts	Selecione grupos de hosts para exibir problemas no widget. Alternativamente, selecione um widget compatível como fonte de dados para grupos de hosts. Este campo é auto-completável, então ao começar a digitar o nome de um grupo será exibida uma lista suspensa de grupos correspondentes. Especificar um grupo de hosts pai seleciona implicitamente todos os grupos de hosts aninhados. Problemas desses grupos de hosts serão exibidos no widget; se nenhum grupo de hosts for inserido, problemas de todos os grupos de hosts serão exibidos. Este parâmetro não está disponível ao configurar o widget em um dashboard de template.
Excluir grupos de hosts	Selecione grupos de hosts para ocultar problemas do widget. Este campo é auto-completável, então ao começar a digitar o nome de um grupo será exibida uma lista suspensa de grupos correspondentes. Especificar um grupo de hosts pai seleciona implicitamente todos os grupos de hosts aninhados. Problemas desses grupos de hosts não serão exibidos no widget. Por exemplo, os hosts 001, 002, 003 podem estar no Grupo A e os hosts 002, 003 também no Grupo B. Se selecionarmos para *mostrar* o Grupo A e *excluir* o Grupo B ao mesmo tempo, apenas problemas do host 001 serão exibidos no widget. Este parâmetro não está disponível ao configurar o widget em um dashboard de template.
Hosts	Selecione hosts para exibir problemas no widget. Alternativamente, selecione um widget compatível ou o dashboard como fonte de dados para hosts. Este campo é auto-completável, então ao começar a digitar o nome de um host será exibida uma lista suspensa de hosts correspondentes. Se nenhum host for inserido, problemas de todos os hosts serão exibidos. Este parâmetro não está disponível ao configurar o widget em um dashboard de template.
Problema	Você pode limitar o número de problemas exibidos pelo nome. Se você inserir uma string aqui, apenas os problemas cujo nome contenha a string inserida serão exibidos. Macros não são expandidas.
Severidade	Marque as severidades dos problemas para filtrar os problemas a serem exibidos no widget. Se nenhuma severidade for marcada, todos os problemas serão exibidos.
Tags do problema	Especifique tags de problemas para limitar o número de problemas exibidos no widget. É possível incluir ou excluir tags e valores de tags específicos. Várias condições podem ser definidas. A correspondência do nome da tag diferencia maiúsculas de minúsculas. Existem vários operadores disponíveis para cada condição: **Existe** - inclui os nomes de tags especificados; **Igual** - inclui os nomes e valores de tags especificados (diferencia maiúsculas de minúsculas); **Contém** - inclui os nomes de tags especificados onde os valores das tags contêm a string inserida (correspondência de substring, não diferencia maiúsculas de minúsculas); **Não existe** - exclui os nomes de tags especificados; **Não é igual** - exclui os nomes e valores de tags especificados (diferencia maiúsculas de minúsculas); **Não contém** - exclui os nomes de tags especificados onde os valores das tags contêm a string inserida (correspondência de substring, não diferencia maiúsculas de minúsculas). Existem dois tipos de cálculo para as condições: **E/Ou** - todas as condições devem ser atendidas, condições com o mesmo nome de tag serão agrupadas pela condição *Ou*; **Ou** - basta que uma condição seja atendida. Ao filtrar, as tags especificadas aqui serão exibidas primeiro com o problema, a menos que sejam substituídas pela lista *Prioridade de exibição da tag* (veja abaixo).
Mostrar tags	Selecione o número de tags exibidas: **Nenhuma** - nenhuma coluna *Tags*; **1** - a coluna *Tags* contém uma tag; **2** - a coluna *Tags* contém duas tags; **3** - a coluna *Tags* contém três tags. Para ver todas as tags do problema, passe o mouse sobre o ícone de três pontos.

Nome da tag	Selecione o modo de exibição do nome da tag: **Completo** - nomes e valores das tags são exibidos por completo; **Abreviado** - nomes das tags são abreviados para 3 símbolos, mas os valores das tags são exibidos por completo; **Nenhum** - apenas os valores das tags são exibidos; sem nomes.
Prioridade de exibição da tag	Insira a prioridade de exibição da tag para um problema, como uma lista de tags separadas por vírgula. Devem ser usados apenas nomes de tags, sem valores. Exemplo: `Services,Applications,Application` As tags desta lista sempre serão exibidas primeiro, substituindo a ordenação natural por ordem alfabética.
Mostrar dados operacionais	Selecione o modo de exibição dos dados operacionais: **Nenhum** - nenhum dado operacional é exibido; **Separadamente** - os dados operacionais são exibidos em uma coluna separada; **Com o nome do problema** - adiciona os dados operacionais ao nome do problema, usando parênteses para os dados operacionais.
Mostrar sintomas	Marque a caixa de seleção para exibir em sua própria linha problemas classificados como sintomas.
Mostrar problemas suprimidos	Marque a caixa de seleção para exibir problemas que, de outra forma, seriam suprimidos (não exibidos) devido à manutenção do host ou à supressão de problema individual.
Status de reconhecimento	Filtre para exibir todos os problemas, apenas problemas não reconhecidos ou apenas problemas reconhecidos. Marque a caixa de seleção adicional para filtrar aqueles problemas já reconhecidos por você.
Ordenar entradas por	Ordenar entradas por: **Hora** (decrescente ou crescente); **Severidade** (decrescente ou crescente); **Nome do problema** (decrescente ou crescente); **Host** (decrescente ou crescente). Ordenar entradas por **Host** (decrescente ou crescente) não está disponível ao configurar o widget em um dashboard de template.
Mostrar linha do tempo	Marque a caixa de seleção para exibir uma linha do tempo visual.
Destacar linha inteira	Marque a caixa de seleção para destacar a linha completa para problemas não resolvidos. A cor da severidade do problema é usada para destacar. *Destacar linha inteira* não está disponível nos temas de alto contraste. Esta opção é suportada desde o Zabbix 7.0.19.
Mostrar linhas	Especifique o número de linhas de problemas a serem exibidas.

Usando o widget



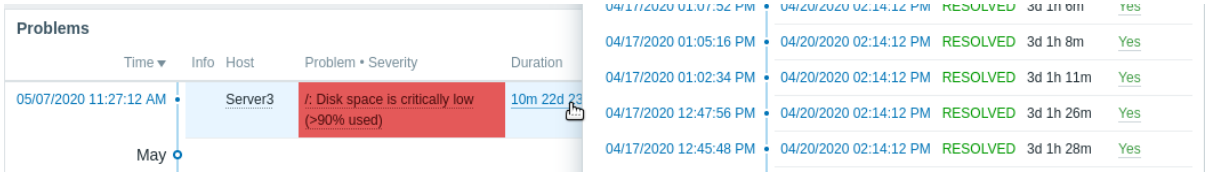
O widget de problemas oferece acesso rápido a informações adicionais:

- Clique na data do problema (na coluna Hora) ou na data de recuperação (na coluna Hora de recuperação) para visualizar os detalhes do evento.
- Se o campo Info não estiver vazio, passe o mouse sobre o ícone exibido para ver detalhes adicionais.
- Clique no nome do host para abrir o menu do host.
- Clique no nome do problema para abrir o menu de evento.
- Passe o mouse ou clique sobre a duração do problema para visualizar o popup de evento de problema.
- Pressionar Atualizar abre uma janela de Atualizar problema.
- Passe o mouse ou pressione o ícone de seta cinza na coluna Ações para ver a lista de ações executadas.

Popup de evento de problema

O popup de evento de problema inclui a lista de eventos de problema para este trigger e, se definido, a descrição do trigger e uma

URL clicável.



Para exibir o popup de evento de problema:

- Passe o mouse sobre a duração do problema na coluna *Duração* do widget *Problemas*. O popup desaparece assim que você remove o mouse da duração.
- Clique na duração na coluna *Duração* do widget *Problemas*. O popup desaparece somente se você clicar novamente na duração.

23 Problemas por gravidade

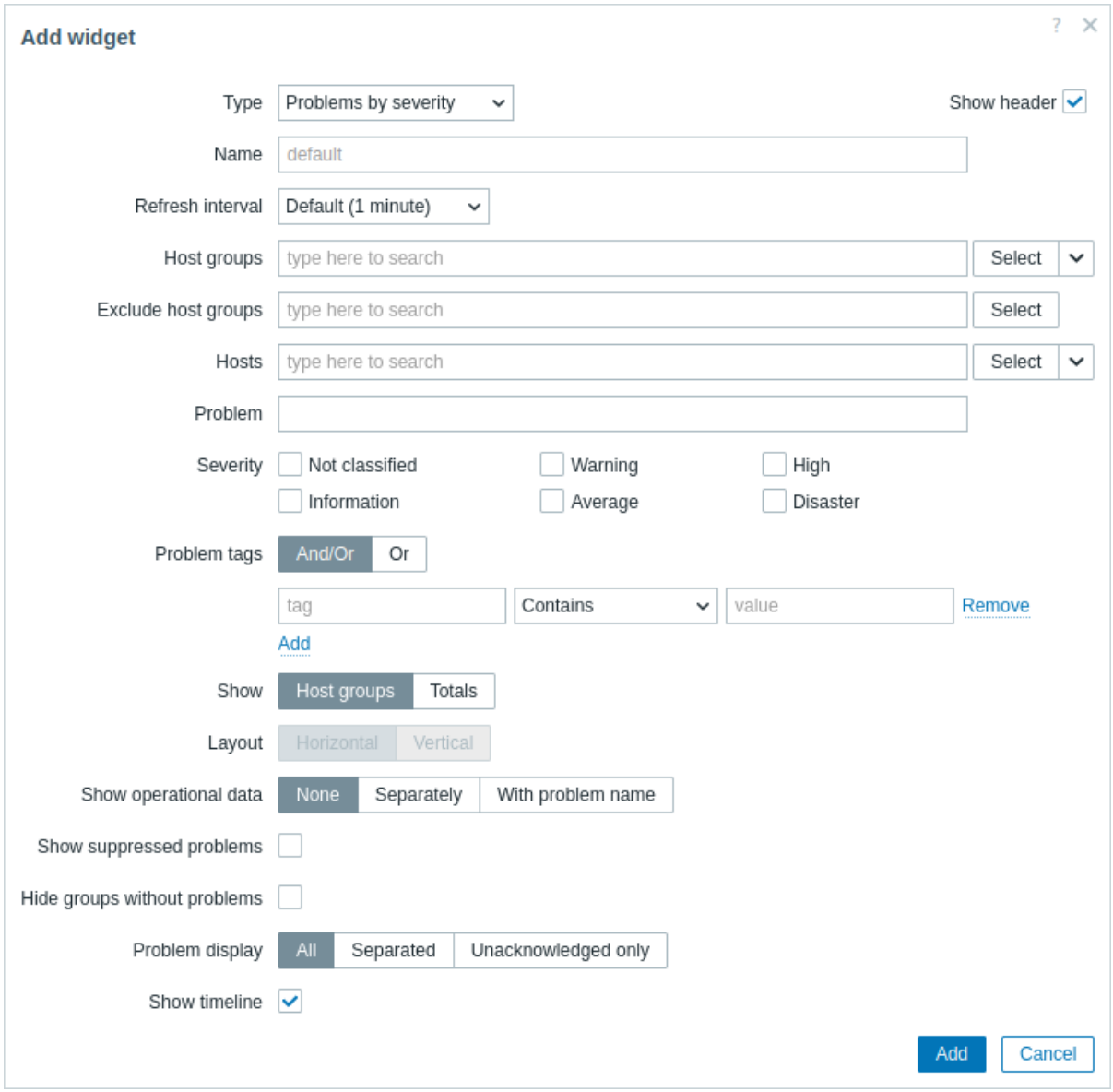
Visão geral

Neste widget, você pode exibir a contagem de problemas por gravidade. Você pode limitar quais hosts e triggers são exibidos no widget e definir como a contagem de problemas é exibida.

A contagem de problemas é exibida apenas para problemas de causa.

Configuração

Para configurar, selecione *Problemas por severidade* como tipo:



Além dos parâmetros que são comuns para todos os widgets, você pode definir as seguintes opções específicas:

Grupos de hosts	Selecione grupos de hosts para exibir no widget. Como alternativa, selecione um widget compatível como fonte de dados para grupos de hosts. Este campo é auto-completável, então ao começar a digitar o nome de um grupo será exibida uma lista suspensa de grupos correspondentes. Especificar um grupo de hosts pai seleciona implicitamente todos os grupos de hosts aninhados. Os dados dos hosts desses grupos de hosts serão exibidos no widget; se nenhum grupo de hosts for inserido, todos os grupos de hosts serão exibidos. Este parâmetro não está disponível ao configurar o widget em um dashboard de template.
Excluir grupos de hosts	Selecione grupos de hosts para ocultar do widget. Este campo é auto-completável, então ao começar a digitar o nome de um grupo será exibida uma lista suspensa de grupos correspondentes. Especificar um grupo de hosts pai seleciona implicitamente todos os grupos de hosts aninhados. Os dados dos hosts desses grupos de hosts não serão exibidos no widget. Por exemplo, os hosts 001, 002, 003 podem estar no Grupo A e os hosts 002, 003 também no Grupo B. Se selecionarmos *exibir* o Grupo A e *excluir* o Grupo B ao mesmo tempo, apenas os dados do host 001 serão exibidos no dashboard. Este parâmetro não está disponível ao configurar o widget em um dashboard de template.
Hosts	Selecione hosts para exibir no widget. Como alternativa, selecione um widget compatível ou o dashboard como fonte de dados para hosts. Este campo é auto-completável, então ao começar a digitar o nome de um host será exibida uma lista suspensa de hosts correspondentes. Se nenhum host for inserido, todos os hosts serão exibidos. Este parâmetro não está disponível ao configurar o widget em um dashboard de template.
Problema	Você pode limitar o número de hosts com problemas exibidos pelo nome do problema. Se você inserir uma string aqui, apenas aqueles hosts com problemas cujo nome contenha a string inserida serão exibidos. Macros não são expandidas.
Severidade	Marque as severidades dos problemas para filtrar os problemas a serem exibidos no widget. Se nenhuma severidade for marcada, todos os problemas serão exibidos.
Tags do problema	Especifique tags de problemas para limitar o número de problemas exibidos no widget. É possível incluir ou excluir tags e valores de tags específicos. Várias condições podem ser definidas. A correspondência do nome da tag é sempre sensível a maiúsculas e minúsculas. Existem vários operadores disponíveis para cada condição: **Existe** - inclui os nomes de tags especificados; **Igual** - inclui os nomes e valores de tags especificados (sensível a maiúsculas e minúsculas); **Contém** - inclui os nomes de tags especificados onde os valores das tags contêm a string inserida (correspondência de substring, não sensível a maiúsculas e minúsculas); **Não existe** - exclui os nomes de tags especificados; **Diferente** - exclui os nomes e valores de tags especificados (sensível a maiúsculas e minúsculas); **Não contém** - exclui os nomes de tags especificados onde os valores das tags contêm a string inserida (correspondência de substring, não sensível a maiúsculas e minúsculas). Existem dois tipos de cálculo para as condições: **E/Ou** - todas as condições devem ser atendidas, condições com o mesmo nome de tag serão agrupadas pela condição *Ou*; **Ou** - basta que uma condição seja atendida.
Exibir	Selecione a opção de exibição: **Grupos de hosts** - exibe problemas por grupo de hosts; **Totais** - exibe um total de problemas para todos os grupos de hosts selecionados em blocos coloridos correspondentes à severidade do problema. Este parâmetro não está disponível ao configurar o widget em um dashboard de template, e apenas um total de problemas será exibido.
Layout	Selecione a opção de layout: **Horizontal** - blocos coloridos de totais serão exibidos horizontalmente; **Vertical** - blocos coloridos de totais serão exibidos verticalmente. Este parâmetro está disponível se *Exibir* estiver definido como ”Totais”.

Exibir dados operacionais	Exibe dados operacionais (veja a descrição de *Dados operacionais* em *Monitoramento > Problemas*) como: **Nenhum** - dados operacionais não serão exibidos **Separadamente** - dados operacionais serão exibidos como uma linha separada **Com o nome do problema** - dados operacionais serão exibidos junto com o nome do problema.
Exibir problemas suprimidos	Marque a caixa de seleção para exibir problemas que, de outra forma, seriam suprimidos (não exibidos) devido à manutenção do host.
Ocultar grupos sem problemas	Marque a opção *Ocultar grupos sem problemas* para ocultar dados de grupos de hosts sem problemas no widget. Este parâmetro não está disponível ao configurar o widget em um dashboard de template.
Exibição de problemas	Exibe a contagem de problemas como: **Todos** - a contagem total de problemas será exibida; **Separados** - a contagem de problemas não reconhecidos será exibida separadamente como um número do total de problemas; **Apenas não reconhecidos** - apenas a contagem de problemas não reconhecidos será exibida.
Exibir linha do tempo	Marque a caixa de seleção para exibir uma linha do tempo visual.

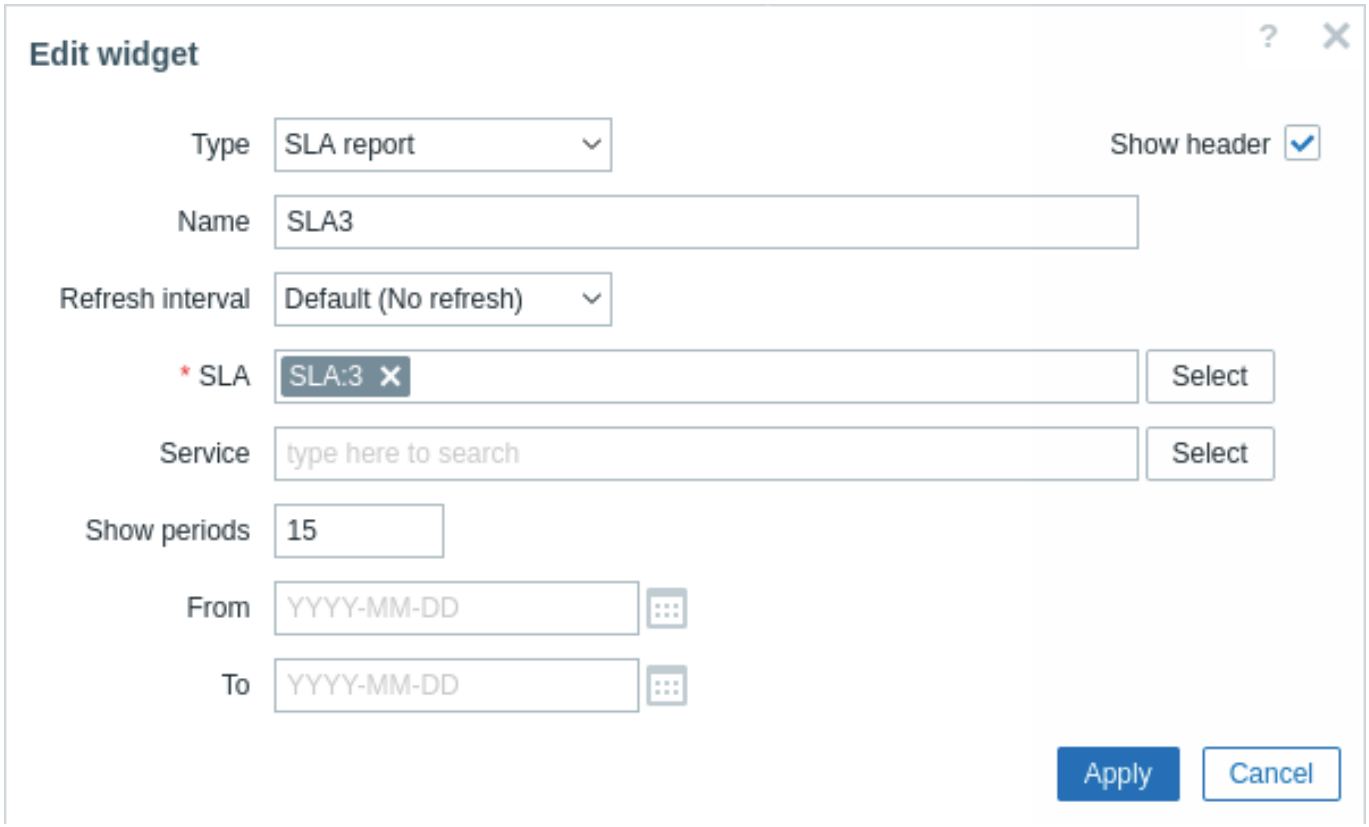
24 Relatório de SLA

Visão geral

Este widget é útil para exibir relatórios de SLA. Funcionalmente, é semelhante à seção *Serviços -> Relatório de SLA*.

Configuração

Para configurar, selecione *Relatório de SLA* como tipo:



Além dos parâmetros que são comuns para todos os widgets, você pode definir as seguintes opções específicas:

SLA	Selecione o SLA para o relatório.
Serviço	Selecione o serviço para o relatório.
Mostrar períodos	Defina quantos períodos serão exibidos no widget (20 por padrão, 100 no máximo).
De	Selecione a data de início para o relatório. Datas relativas são suportadas: `now`, `now/d`, `now/w-1w` etc; modificadores de data suportados: d, w, M, y.

Até	Selecione a data final para o relatório. Datas relativas são suportadas: `now`, `now/d`, `now/w-1w` etc; modificadores de data suportados: d, w, M, y.

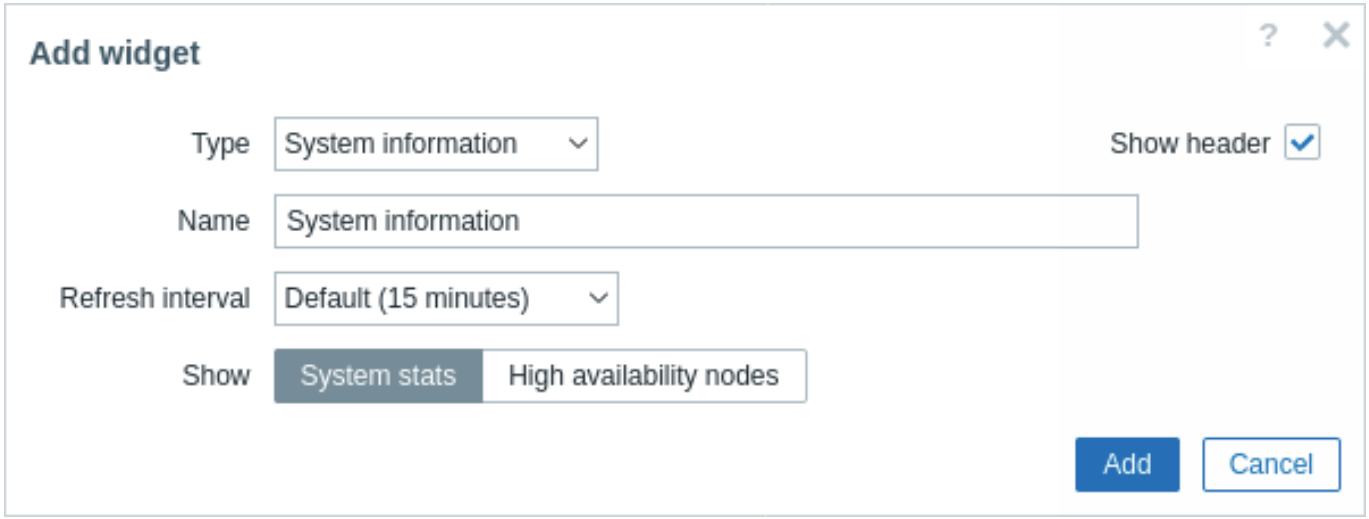
25 Informações do sistema

Visão geral

Este widget exibe as mesmas informações que em *Relatórios* → *Informações do sistema*, no entanto, um único widget de dashboard pode exibir apenas as estatísticas do sistema ou os nós de alta disponibilidade por vez (não ambos).

Configuração

Para configurar, selecione *Informações do sistema* como tipo:



Além dos parâmetros que são comuns para todos os widgets, você pode definir as seguintes opções específicas:

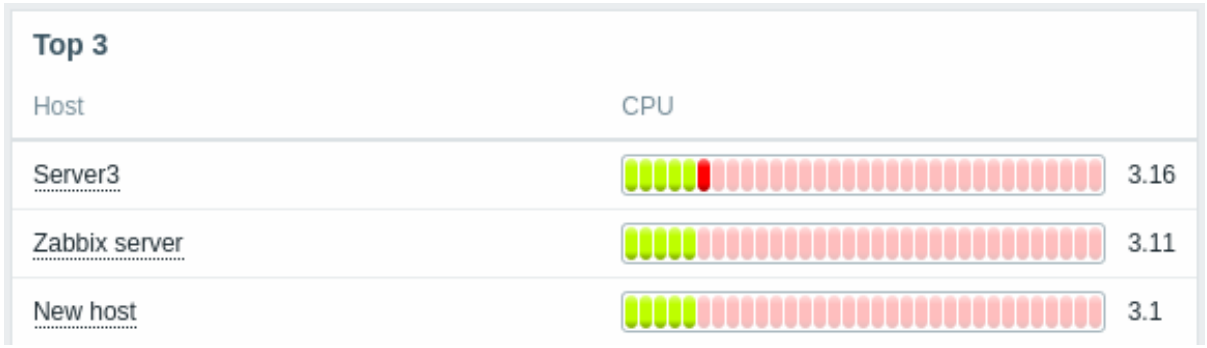
Mostrar	Selecione o que exibir: **Estatísticas do sistema** - exibe um resumo dos principais dados do Zabbix server e do sistema; **Nós de alta disponibilidade** - exibe o status dos nós de alta disponibilidade (se o cluster de alta disponibilidade estiver habilitado).
Mostrar detalhes da verificação de atualização de software	Marque a caixa de seleção para exibir detalhes da verificação de atualização do software Zabbix. Esta opção só está disponível se a verificação de atualização de software estiver habilitada na configuração do Zabbix server e "Estatísticas do sistema" estiver selecionado no campo *Mostrar*.

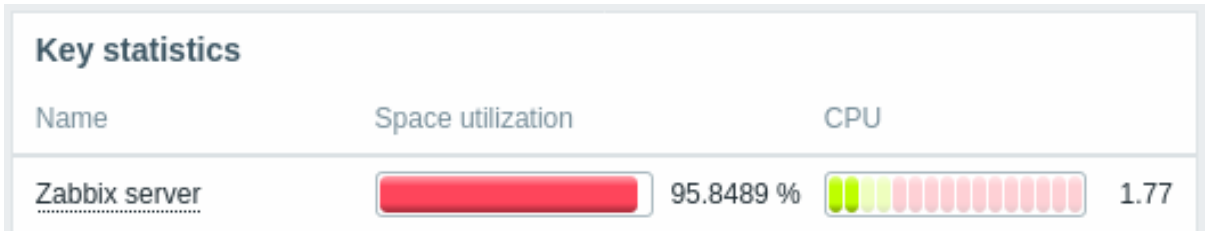
26 Principais hosts

Visão geral

Este widget fornece uma maneira de criar tabelas personalizadas para exibir a situação dos dados, permitindo exibir relatórios do tipo *Top N* e relatórios de barra de progresso úteis para planejamento de capacidade.

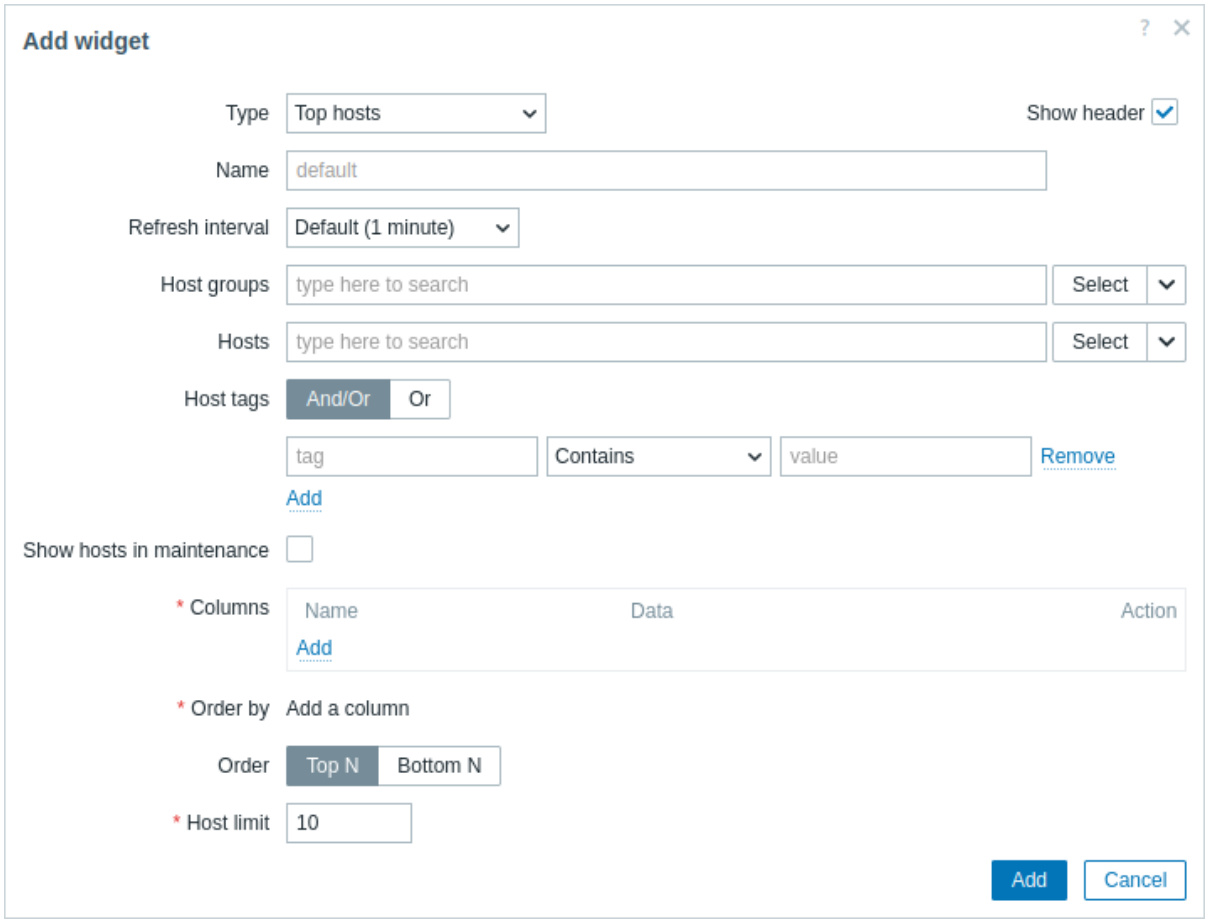
O número máximo de hosts que podem ser exibidos é 1000 (100 antes do Zabbix 7.0.6).





Configuração

Para configurar, selecione *Top hosts* como tipo:

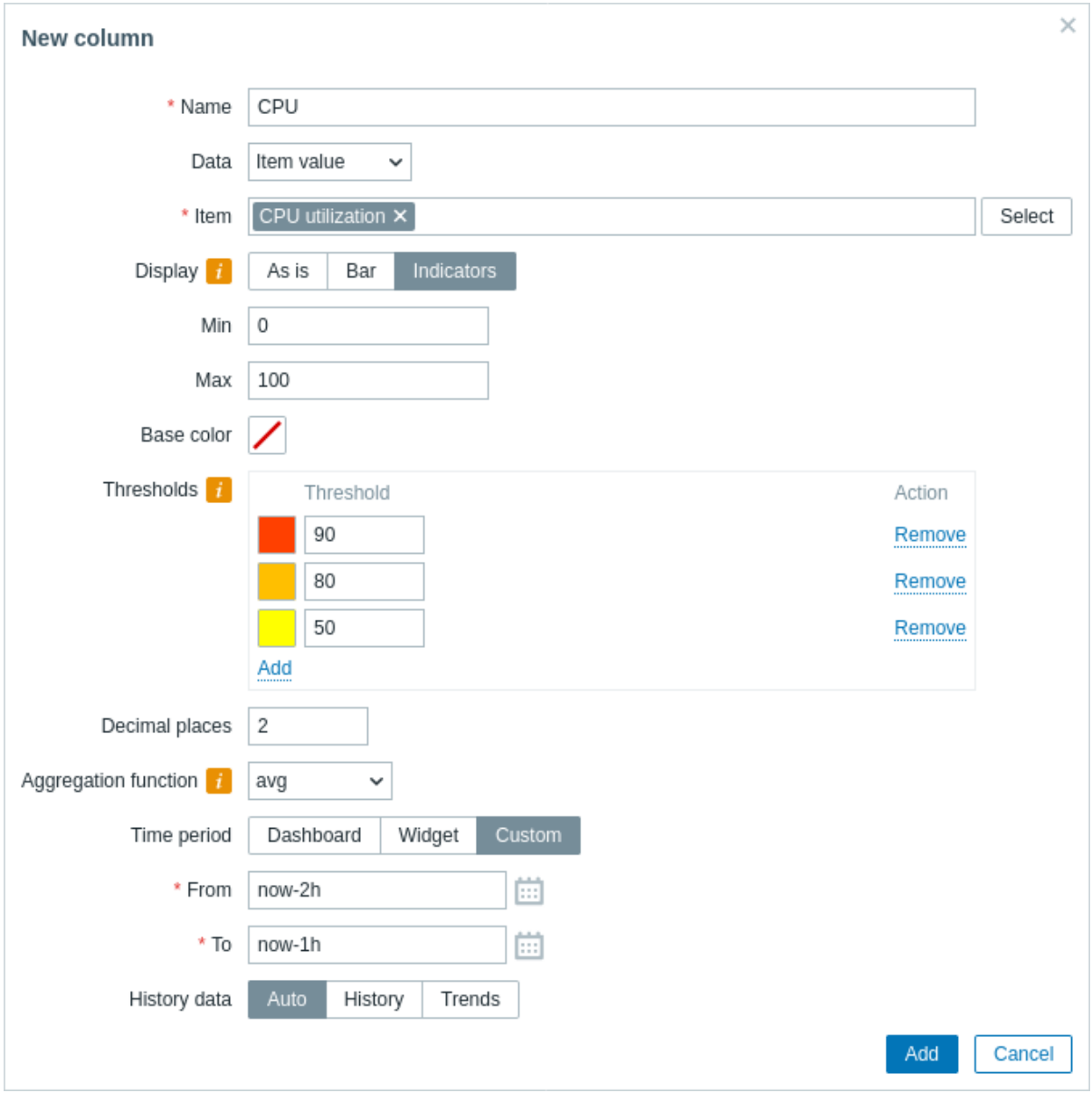


Além dos parâmetros que são comuns para todos os widgets, você pode definir as seguintes opções específicas:

Grupos de hosts	Selecione os grupos de hosts a serem exibidos no widget. Como alternativa, selecione um widget compatível como fonte de dados para grupos de hosts. Este campo é de preenchimento automático, portanto, ao começar a digitar o nome de um grupo, será exibida uma lista suspensa com os grupos correspondentes. Este parâmetro não está disponível ao configurar o widget em um dashboard de template.

Hosts	Selecione os hosts a serem exibidos no widget. Como alternativa, selecione um widget compatível ou o dashboard como fonte de dados para hosts. Este campo é de preenchimento automático, portanto, ao começar a digitar o nome de um host, será exibida uma lista suspensa com os hosts correspondentes. Este parâmetro não está disponível ao configurar o widget em um dashboard de template.
Tags de host	Especifique tags para limitar o número de hosts exibidos no widget. É possível incluir ou excluir tags e valores de tags específicos. Várias condições podem ser definidas. A correspondência do nome da tag diferencia maiúsculas de minúsculas. Há vários operadores disponíveis para cada condição: **Existe** - inclui os nomes de tags especificados; **Igual** - inclui os nomes e valores de tags especificados (diferencia maiúsculas de minúsculas); **Contém** - inclui os nomes de tags especificados onde os valores das tags contêm a string inserida (correspondência de substring, não diferencia maiúsculas de minúsculas); **Não existe** - exclui os nomes de tags especificados; **Não é igual** - exclui os nomes e valores de tags especificados (diferencia maiúsculas de minúsculas); **Não contém** - exclui os nomes de tags especificados onde os valores das tags contêm a string inserida (correspondência de substring, não diferencia maiúsculas de minúsculas). Existem dois tipos de cálculo para as condições: **E/Ou** - todas as condições devem ser atendidas, condições com o mesmo nome de tag serão agrupadas pela condição Ou; **Ou** - basta que uma condição seja atendida. Este parâmetro não está disponível ao configurar o widget em um dashboard de template.
Exibir hosts em manutenção	Marque esta caixa para que os hosts em manutenção também sejam exibidos (neste caso, o ícone de manutenção será mostrado ao lado do nome do host). Desmarcado por padrão.
Colunas	Adicione colunas de dados para exibição. A ordem das colunas determina sua exibição da esquerda para a direita. As colunas podem ser reordenadas arrastando para cima e para baixo pela alça antes do nome da coluna.
Ordenar por	Especifique a coluna da lista de *Colunas* definida para usar na ordenação *Top N* ou *Bottom N*.
Ordem	Especifique a ordenação das linhas: **Top N** - em ordem decrescente de acordo com o valor agregado de *Ordenar por*; **Bottom N** - em ordem crescente de acordo com o valor agregado de *Ordenar por*.
Limite de hosts	Número de linhas de hosts a serem exibidas (1-1000; 1-100 antes do Zabbix 7.0.6). Este parâmetro não está disponível ao configurar o widget em um dashboard de template.

Configuração de colunas



Parâmetros comuns da coluna:

Nome	Nome da coluna.
Dados	Tipo de dado a ser exibido na coluna: **Valor do item** - valor do item especificado; **Nome do host** - nome do host; **Texto** - string de texto estática.
Cor base	Cor de fundo da coluna; cor de preenchimento se os *Valores do item* forem exibidos como barra/indicadores. Para os *Valores do item*, a cor padrão pode ser substituída por uma cor personalizada, se o valor do item ultrapassar um dos limites especificados.

Parâmetros específicos para colunas de valores de item:

Item	Selecione um item; seu nome é usado para corresponder e exibir itens com o mesmo nome em todos os hosts selecionados. Se um host tiver vários desses itens, o widget exibirá o item cuja chave for a primeira em ordem alfabética (por exemplo, *CPU utilization* com a chave `proc.cpu.util` em vez de `system.cpu.util`). Ao configurar o widget em um dashboard de template, apenas itens configurados no template podem ser selecionados. A seleção de itens com valores binários não é suportada.

Exibir	Defina como o valor deve ser exibido: **Como está** - como texto comum; **Barra** - como barra sólida e colorida; **Indicadores** - como barra segmentada e colorida. Observe que apenas itens numéricos podem ser exibidos nesta coluna se esta configuração não for ”como está”.
Mín	Valor mínimo para barra/indicadores.
Máx	Valor máximo para barra/indicadores.
Limiares	Especifique os valores de limiar quando a cor de fundo/preenchimento deve mudar. A lista será classificada em ordem crescente ao salvar. Observe que apenas itens numéricos podem ser exibidos nesta coluna se os limiares forem usados.
Casas decimais	Especifique quantas casas decimais serão exibidas com o valor. Esta configuração se aplica apenas a dados numéricos.
Função de agregação	Especifique qual função de agregação usar: **min** - exibe o menor valor; **max** - exibe o maior valor; **avg** - exibe o valor médio; **count** - exibe a contagem de valores; **sum** - exibe a soma dos valores; **first** - exibe o primeiro valor; **last** - exibe o último valor; **não usado** - exibe o valor mais recente (sem agregação). A agregação permite exibir um valor agregado para o intervalo escolhido (5 minutos, uma hora, um dia), em vez do valor mais recente. Apenas dados numéricos podem ser exibidos para *min*, *max*, *avg* e *sum*. Para *count*, dados não numéricos serão convertidos para numéricos.
Período de tempo	Especifique o período de tempo a ser usado para agregar valores: **Dashboard** - usa o período de tempo do dashboard; **Widget** - usa o período de tempo do widget especificado; **Personalizado** - usa um período de tempo personalizado. Este parâmetro não será exibido se a *Função de agregação* estiver definida como ”não usado”.
Widget	Selecione o widget. Este parâmetro só será exibido se o *Período de tempo* estiver definido como ”Widget”.
De	Selecione o período de tempo inicial (valor padrão `now-1h`). Veja sintaxe de tempo relativo. Este parâmetro só será exibido se o *Período de tempo* estiver definido como ”Personalizado”.
Até	Selecione o período de tempo final (valor padrão `now`). Veja sintaxe de tempo relativo. Este parâmetro só será exibido se o *Período de tempo* estiver definido como ”Personalizado”.
Dados do histórico	Obtenha dados do histórico ou das tendências: **Auto** - seleção automática; **Histórico** - obtém dados do histórico; **Tendências** - obtém dados das tendências. Esta configuração se aplica apenas a dados numéricos. Dados não numéricos sempre serão obtidos do histórico.

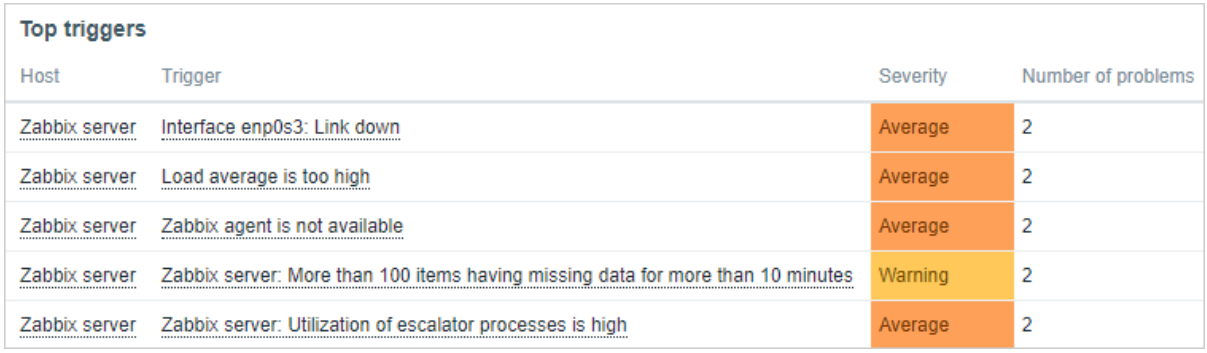
Parâmetros específicos para colunas de texto:

Texto	Insira a string a ser exibida. Pode conter macros de host e inventário macros.

27 Principais triggers

Visão geral

No widget *Top triggers*, você pode ver os triggers com o maior número de problemas.

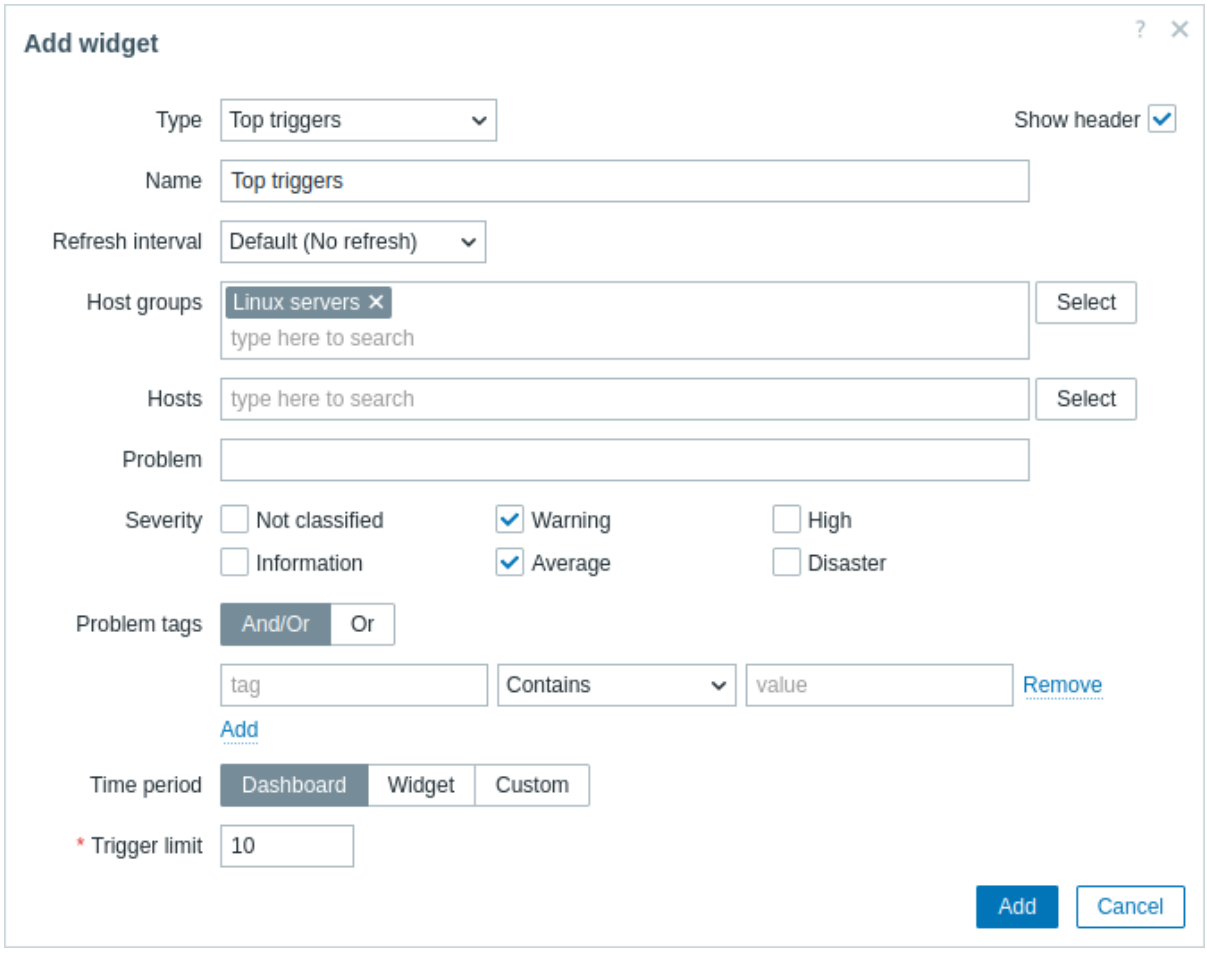


O número máximo de triggers que podem ser exibidos é 1000 (100 antes do Zabbix 7.0.6). Ao visualizar o widget em um dashboard, é possível selecionar o período de tempo para exibir os dados.

As informações sobre os top triggers também estão disponíveis na seção de menu *Relatórios* → *Top 100 triggers*.

Configuração

Para configurar, selecione *Top triggers* como tipo de widget:



Além dos parâmetros que são comuns para todos os widgets, você pode definir as seguintes opções específicas:

Grupos de hosts	Selecione os grupos de hosts para exibir problemas no widget. Este campo é auto-completável, então ao começar a digitar o nome de um grupo será exibida uma lista suspensa de grupos correspondentes. Especificar um grupo de hosts pai seleciona implicitamente todos os grupos de hosts aninhados. Problemas desses grupos de hosts serão exibidos no widget; se nenhum grupo de hosts for inserido, problemas de todos os grupos de hosts serão exibidos. Este parâmetro não está disponível ao configurar o widget em um dashboard de template.
Hosts	Selecione os hosts para exibir problemas no widget. Este campo é auto-completável, então ao começar a digitar o nome de um host será exibida uma lista suspensa de hosts correspondentes. Se nenhum host for inserido, problemas de todos os hosts serão exibidos. Este parâmetro não está disponível ao configurar o widget em um dashboard de template.

Problema	Você pode visualizar os triggers para problemas específicos apenas. Para isso, insira a string a ser correspondida no nome do problema. Macros não são expandidas.
Severidade	Marque as severidades dos triggers para filtrar os triggers a serem exibidos no widget. Se nenhuma severidade for marcada, todos os triggers serão exibidos.
Tags do problema	Especifique as tags dos problemas a serem exibidos no widget. É possível incluir ou excluir tags e valores de tags específicos. Várias condições podem ser definidas. A correspondência do nome da tag diferencia maiúsculas de minúsculas. Existem vários operadores disponíveis para cada condição: **Existe** - inclui os nomes de tags especificados; **Igual** - inclui os nomes e valores de tags especificados (diferencia maiúsculas de minúsculas); **Contém** - inclui os nomes de tags especificados onde os valores das tags contêm a string inserida (correspondência de substring, não diferencia maiúsculas de minúsculas); **Não existe** - exclui os nomes de tags especificados; **Diferente** - exclui os nomes e valores de tags especificados (diferencia maiúsculas de minúsculas); **Não contém** - exclui os nomes de tags especificados onde os valores das tags contêm a string inserida (correspondência de substring, não diferencia maiúsculas de minúsculas). Existem dois tipos de cálculo para as condições: **E/Ou** - todas as condições devem ser atendidas, condições com o mesmo nome de tag serão agrupadas pela condição *Ou*; **Ou** - basta que uma condição seja atendida.
Período de tempo	Selecione a fonte de dados para o período de tempo: **Dashboard** - define o seletor *Período de tempo* como fonte de dados; **Widget** - define um widget compatível especificado no parâmetro *Widget* como fonte de dados; **Personalizado** - define o período de tempo especificado nos parâmetros *De* e *Até* como fonte de dados; se definido, um ícone de relógio será exibido no canto superior direito do widget, indicando o tempo definido ao passar o mouse.
Widget	Digite ou selecione um widget compatível (*Gráfico*, *Gráfico (clássico)*, *Protótipo de gráfico*) como fonte de dados para o período de tempo. Este parâmetro está disponível se *Período de tempo* estiver definido como "Widget".
De	Digite ou selecione o início do período de tempo. A sintaxe de tempo relativo (`now`, `now/d`, `now/w-1w`, etc.) é suportada. Este parâmetro está disponível se *Período de tempo* estiver definido como "Personalizado".
Até	Digite ou selecione o final do período de tempo. A sintaxe de tempo relativo (`now`, `now/d`, `now/w-1w`, etc.) é suportada. Este parâmetro está disponível se *Período de tempo* estiver definido como "Personalizado".
Limite de triggers	Defina o número de triggers a serem exibidos. Faixa de valores possível: 1-1000 (1-100 antes do Zabbix 7.0.6).

28 Visão geral do trigger

Visão geral

No widget de visão geral de triggers, você pode exibir os estados das triggers para um grupo de hosts.

- Os estados das triggers são exibidos como blocos coloridos (a cor dos blocos para triggers de PROBLEMA depende da cor de severidade do problema, que pode ser ajustada na tela de atualização de problema). Observe que alterações recentes no estado da trigger (nos últimos 2 minutos) serão exibidas como blocos piscando.
- Setas cinzas para cima e para baixo indicam triggers que possuem dependências. Ao passar o mouse, os detalhes das dependências são revelados.
- Um ícone de caixa de seleção indica problemas reconhecidos. Todos os problemas ou problemas resolvidos da trigger devem ser reconhecidos para que este ícone seja exibido.

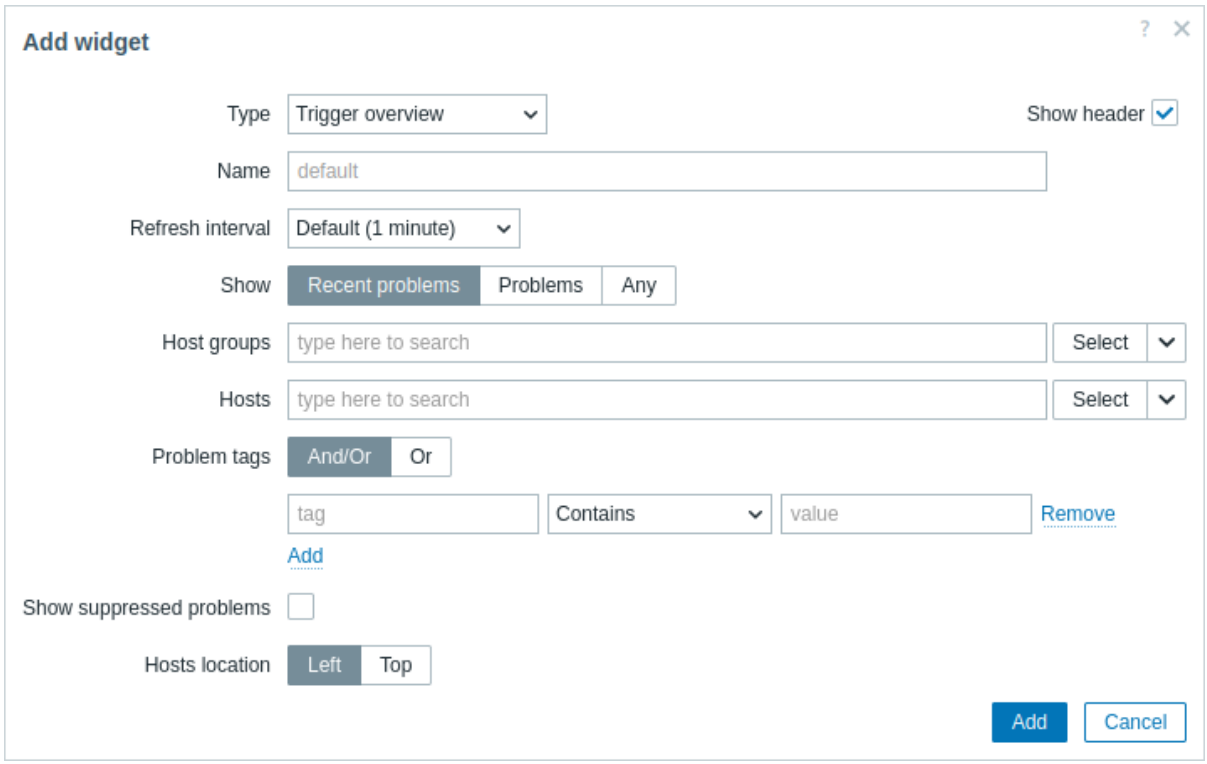
Clicar em um bloco de trigger fornece links dependentes do contexto para eventos de problema da trigger, tela de reconhecimento de problema, configuração da trigger, URL da trigger ou um gráfico simples/lista de valores mais recentes.

Observe que 50 registros são exibidos por padrão (configurável em *Administração* → *Geral* → *GUI*, usando a opção *Número máximo de colunas e linhas nas tabelas de visão geral*). Se existirem mais registros do que o configurado para exibição, uma mensagem

será exibida na parte inferior da tabela, solicitando critérios de filtragem mais específicos. Não há paginação. Observe que esse limite é aplicado primeiro, antes de qualquer filtragem adicional dos dados, por exemplo, por tags.

Configuração

Para configurar, selecione *Visão geral de triggers* como tipo:



Além dos parâmetros que são comuns para todos os widgets, você pode definir as seguintes opções específicas:

Mostrar	Filtrar triggers pelo estado do trigger: **Problemas recentes** - *(padrão)* mostra triggers que recentemente estiveram ou ainda estão em estado de PROBLEMA (resolvidos e não resolvidos); **Problemas** - mostra triggers que estão em estado de PROBLEMA (não resolvidos); **Qualquer** - mostra todos os triggers.
Grupos de hosts	Selecione grupos de hosts. Alternativamente, selecione um widget compatível como fonte de dados para grupos de hosts. Este campo é autocompletável, então ao começar a digitar o nome de um grupo será exibida uma lista suspensa de grupos correspondentes. Este parâmetro não está disponível ao configurar o widget em um dashboard de template.
Hosts	Selecione hosts. Alternativamente, selecione um widget compatível ou o dashboard como fonte de dados para hosts. Este campo é autocompletável, então ao começar a digitar o nome de um host será exibida uma lista suspensa de hosts correspondentes. Este parâmetro não está disponível ao configurar o widget em um dashboard de template.

Tags de problema	Especifique tags para filtrar os triggers exibidos no widget. É possível incluir e também excluir tags e valores de tags específicos. Várias condições podem ser definidas. A correspondência do nome da tag diferencia maiúsculas de minúsculas. **Nota:** Se o parâmetro *Mostrar* estiver definido como 'Qualquer', todos os triggers serão exibidos mesmo que as tags sejam especificadas. No entanto, enquanto as alterações recentes de estado do trigger (exibidas como blocos piscando) serão atualizadas para todos os triggers, os detalhes do estado do trigger (cor de severidade do problema e se o problema foi reconhecido) serão atualizados apenas para triggers que correspondam às tags especificadas. Existem vários operadores disponíveis para cada condição: **Existe** - inclui os nomes de tags especificados; **Igual** - inclui os nomes e valores de tags especificados (diferencia maiúsculas de minúsculas); **Contém** - inclui os nomes de tags especificados onde os valores das tags contêm a string inserida (correspondência de substring, não diferencia maiúsculas de minúsculas); **Não existe** - exclui os nomes de tags especificados; **Diferente** - exclui os nomes e valores de tags especificados (diferencia maiúsculas de minúsculas); **Não contém** - exclui os nomes de tags especificados onde os valores das tags contêm a string inserida (correspondência de substring, não diferencia maiúsculas de minúsculas). Existem dois tipos de cálculo para as condições: **E/Ou** - todas as condições devem ser atendidas, condições com o mesmo nome de tag serão agrupadas pela condição *Ou*; **Ou** - basta que uma condição seja atendida.
Mostrar problemas suprimidos	Marque a caixa de seleção para exibir problemas que, de outra forma, seriam suprimidos (não exibidos) devido à manutenção do host.
Localização dos hosts	Selecione a localização do host - esquerda ou topo. Este parâmetro é rotulado como *Localização do host* ao configurar o widget em um dashboard de template.

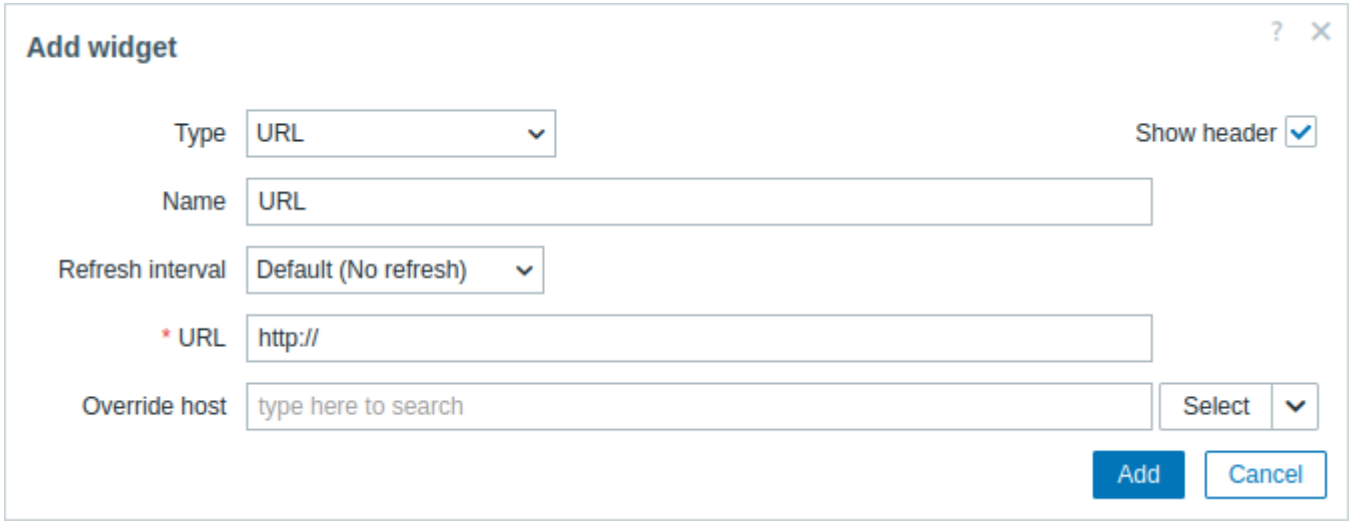
29 URL

Visão geral

Este widget exibe o conteúdo recuperado da URL especificada.

Configuração

Para configurar, selecione *URL* como tipo:



Além dos parâmetros que são comuns para todos os widgets, você pode definir as seguintes opções específicas:

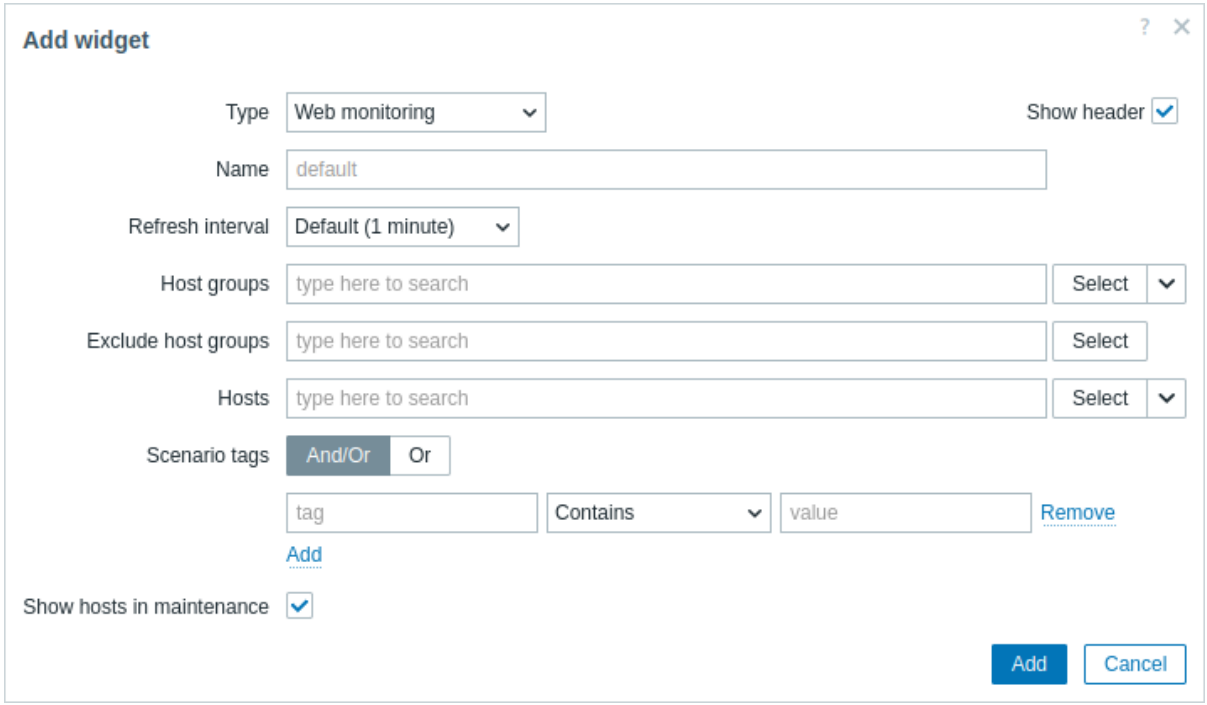
URL	Digite a URL a ser exibida (até 2048 caracteres). URLs externas devem começar com `http://` ou `https://`. URLs internas suportam caminhos relativos (por exemplo, `zabbix.php?action=report.status`). Macros {HOST.*} são suportadas.
Substituir host	Selecione um widget compatível ou o dashboard como fonte de dados para hosts. Este parâmetro não está disponível ao configurar o widget em um dashboard de template.

Attention:
Os navegadores podem não carregar uma página HTTP configurada no widget se o frontend do Zabbix for acessado via HTTPS.

Visão geral

Este widget exibe um resumo do status dos cenários de monitoramento web ativos. Consulte a seção Widget de monitoramento web para obter informações detalhadas.

Configuração



Note:
Nos casos em que um usuário não tem permissão para acessar certos elementos do widget, o nome desse elemento aparecerá como *Inacessível* durante a configuração do widget. Isso resulta em *Item inacessível, Host inacessível, Grupo inacessível, Mapa inacessível e Gráfico inacessível* aparecendo no lugar do nome "real" do elemento.

Além dos parâmetros que são comuns para todos os widgets, você pode definir as seguintes opções específicas:

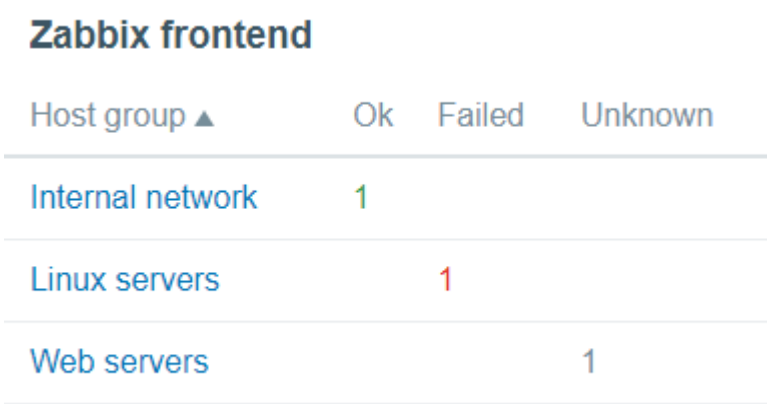
Grupos de hosts	Selecione os grupos de hosts para exibir no widget. Como alternativa, selecione um widget compatível como fonte de dados para grupos de hosts. Este campo é de preenchimento automático, portanto, ao começar a digitar o nome de um grupo, será exibida uma lista suspensa de grupos correspondentes. Especificar um grupo de hosts pai seleciona implicitamente todos os grupos de hosts aninhados. Os dados dos hosts desses grupos de hosts serão exibidos no widget; se nenhum grupo de hosts for inserido, todos os grupos de hosts serão exibidos. Este parâmetro não está disponível ao configurar o widget em um dashboard de template.

Excluir grupos de hosts	Selecione os grupos de hosts para ocultar do widget. Este campo é de preenchimento automático, portanto, ao começar a digitar o nome de um grupo, será exibida uma lista suspensa de grupos correspondentes. Especificar um grupo de hosts pai seleciona implicitamente todos os grupos de hosts aninhados. Os dados dos hosts desses grupos de hosts não serão exibidos no widget. Por exemplo, os hosts 001, 002, 003 podem estar no Grupo A e os hosts 002, 003 também no Grupo B. Se selecionarmos *mostrar* o Grupo A e *excluir* o Grupo B ao mesmo tempo, apenas os dados do host 001 serão exibidos no dashboard. Este parâmetro não está disponível ao configurar o widget em um dashboard de template.
Hosts	Selecione os hosts para exibir no widget. Como alternativa, selecione um widget compatível ou o dashboard como fonte de dados para hosts. Este campo é de preenchimento automático, portanto, ao começar a digitar o nome de um host, será exibida uma lista suspensa de hosts correspondentes. Se nenhum host for inserido, todos os hosts serão exibidos. Este parâmetro não está disponível ao configurar o widget em um dashboard de template.
Tags de cenário	Especifique tags para limitar o número de cenários web exibidos no widget. É possível incluir ou excluir tags e valores de tags específicos. Várias condições podem ser definidas. A correspondência do nome da tag diferencia maiúsculas de minúsculas. Há vários operadores disponíveis para cada condição: **Existe** - inclui os nomes de tags especificados; **Igual** - inclui os nomes e valores de tags especificados (diferencia maiúsculas de minúsculas); **Contém** - inclui os nomes de tags especificados onde os valores das tags contêm a string inserida (correspondência de substring, não diferencia maiúsculas de minúsculas); **Não existe** - exclui os nomes de tags especificados; **Não é igual** - exclui os nomes e valores de tags especificados (diferencia maiúsculas de minúsculas); **Não contém** - exclui os nomes de tags especificados onde os valores das tags contêm a string inserida (correspondência de substring, não diferencia maiúsculas de minúsculas). Existem dois tipos de cálculo para as condições: **E/Ou** - todas as condições devem ser atendidas, condições com o mesmo nome de tag serão agrupadas pela condição *Ou*; **Ou** - basta que uma condição seja atendida.
Exibir hosts em manutenção	Inclui hosts que estão em manutenção nas estatísticas. Este parâmetro é rotulado como *Exibir dados em manutenção* ao configurar o widget em um dashboard de template.

Widget de monitoramento web

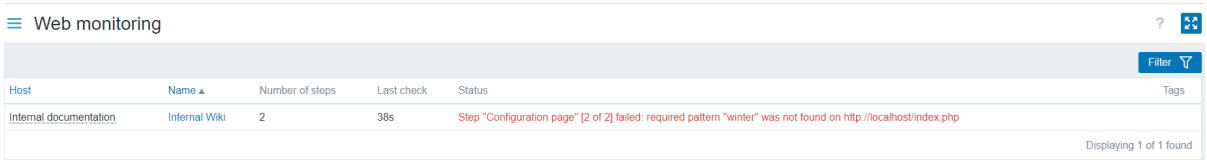
Depois de concluir a configuração, talvez você queira ver o widget com os dados que ele exibe. Para isso, vá em *Dashboards*, clique no nome do dashboard onde você criou o widget.

Neste exemplo, você pode ver o widget chamado ”Zabbix frontend” exibindo o status do monitoramento web para três grupos de hosts: ”Internal network”, ”Linux servers” e ”Web servers”.

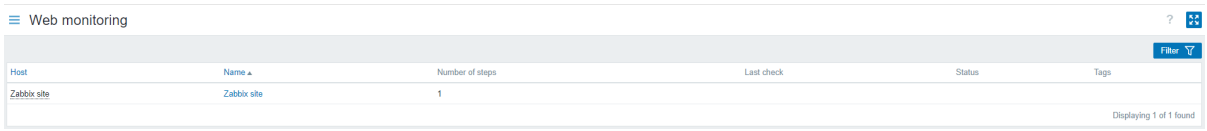


Um widget de monitoramento web exibe as seguintes informações:

- um nome do widget; abaixo dele, há quatro colunas:
  - *Host group* - exibe uma lista de grupos de hosts que contêm hosts com cenários web configurados;
  - *Ok* - exibe o número de cenários web (em verde) quando duas condições são observadas:
    * O Zabbix coletou os dados mais recentes para um ou mais cenários web;
    * todos os passos que foram configurados em um cenário web estão em "Ok" *Status*.
  - *Failed* - exibe o número de cenários web (em vermelho), que possuem alguns passos com falha:
    * clique no nome do host, e será aberta uma nova janela; a coluna *Status* fornece informações detalhadas (em vermelho) sobre o passo em que o Zabbix falhou ao coletar os dados; e também,
    * dá uma dica para o parâmetro que deve ser corrigido no formulário de configuração.



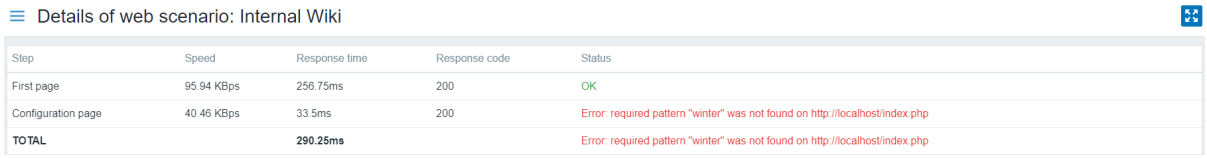
- *Unknown* - exibe o número de cenários web (em cinza) para os quais o Zabbix não coletou dados, nem possui informações sobre os passos com falha.



Visualizando o status e os dados

Links clicáveis no widget permitem navegar facilmente e adquirir rapidamente informações completas sobre cada cenário web. Assim, para visualizar:

- o Status de um cenário web, clique no nome de um grupo de hosts.
- estatísticas mais detalhadas, clique no nome do cenário. Neste exemplo é "Zabbix frontend".
- os detalhes no caso de status *Falhou*, clique no nome de um grupo de hosts; na janela que abrir, clique no nome de um cenário web na coluna *Nome*; isso abrirá informações mais detalhadas sobre as etapas configuradas para as quais o Zabbix não conseguiu coletar os dados.



Agora, você pode retornar ao formulário de configuração do cenário web e corrigir suas configurações.

Para visualizar os detalhes no caso de status *Desconhecido*, você pode repetir os mesmos passos explicados para *Falhou*.

Attention:
Na primeira instância de monitoramento, um cenário web é sempre exibido no estado *Desconhecido*, que é alterado para o estado *Falhou* ou *Ok* logo após a primeira verificação. No caso de um host ser monitorado pelo proxy, a alteração de status ocorre de acordo com a frequência de coleta de dados configurada no proxy.

2 Monitoramento

Visão geral

O menu Monitoramento tem tudo a ver com a exibição de dados. Qualquer informação O Zabbix está configurado para coletar, visualizar e agir, será exibido nas várias seções do menu Monitoramento.

Botões de modo de exibição

Os seguintes botões localizados no canto superior direito são comuns para todas as seções:

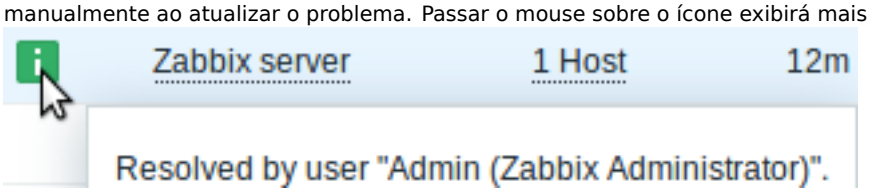
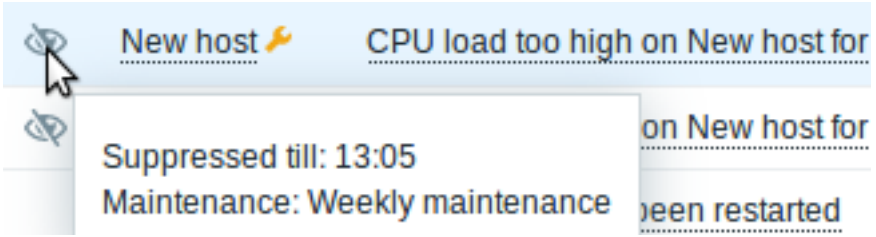
	Exibe a página no modo quiosque. Neste modo, apenas o conteúdo da página é exibido. Para sair do modo quiosque, mova o cursor do mouse até que o botão de saída apareça e clique nele. Você será levado de volta ao modo normal.

1 Problemas

Visão geral

Em *Monitoramento* > *Problemas* você pode ver quais problemas você tem atualmente. Problemas são aqueles triggers que estão no estado "Problema".

Por padrão, todos os novos problemas são classificados como problemas de causa. É possível reclassificar manualmente certos problemas como problemas de sintoma do problema de causa. Para mais detalhes, veja eventos de causa e sintoma.

Coluna	Descrição
Checkbox	Checkboxes para seleção de problemas são exibidos. Ícones, ao lado das checkboxes, têm o seguinte significado: - o número de eventos de sintoma para o problema de causa; - expandir para mostrar eventos de sintoma; - recolher para ocultar eventos de sintoma; - este é um evento de sintoma.
Hora	A hora de início do problema é exibida.
Gravidade	A gravidade do problema é exibida. A gravidade do problema é originalmente baseada na gravidade do trigger subjacente, no entanto, após o evento ter ocorrido, ela pode ser atualizada usando a tela *Atualizar problema* tela. A cor da gravidade do problema é usada como fundo da célula durante o tempo do problema.
Hora de recuperação	A hora de resolução do problema é exibida.
Status	O status do problema é exibido: **Problema** - problema não resolvido **Resolvido** - problema recentemente resolvido. Você pode ocultar problemas recentemente resolvidos usando o filtro. Problemas novos e recentemente resolvidos piscam por 2 minutos. Problemas resolvidos são exibidos por 5 minutos no total. Ambos os valores são configuráveis em *Administração* > *Geral* > *Opções de exibição de triggers*.
Info	Um ícone de informação verde é exibido se um problema for fechado por correlação global ou manualmente ao atualizar o problema. Passar o mouse sobre o ícone exibirá mais detalhes:  O seguinte ícone é exibido se um problema suprimido estiver sendo mostrado (veja a opção *Mostrar problemas suprimidos* no filtro). Passar o mouse sobre o ícone exibirá mais detalhes: 
Host	O host do problema é exibido. Clicar no nome do host abre o menu do host.
Problema	O nome do problema é exibido. O nome do problema é baseado no nome do trigger subjacente. Macros no nome do trigger são resolvidas no momento em que o problema ocorre e os valores resolvidos não são mais atualizados. *Nota* que é possível anexar ao nome do problema dados operacionais mostrando alguns valores recentes de item. Clicar no nome do problema abre o menu de evento. Passe o mouse sobre o ícone após o nome do problema para exibir a descrição do trigger (para aqueles problemas que a possuem).

Coluna	Descrição
Dados operacionais	Dados operacionais são exibidos contendo os valores mais recentes dos itens. Dados operacionais podem ser uma combinação de texto e macros de valor de item se configurados em nível de trigger. Se nenhum dado operacional for configurado em nível de trigger, os valores mais recentes de todos os itens da expressão são exibidos. Esta coluna só é exibida se *Separadamente* for selecionado para *Mostrar dados operacionais* no filtro.
Duração	A duração do problema é exibida. Veja também: Duração negativa do problema
Atualizar	Clique no link *Atualizar* para ir para a tela de atualização do problema, onde várias ações podem ser tomadas sobre o problema, incluindo comentar e reconhecer o problema.
Ações	O histórico de atividades sobre o problema é exibido usando ícones simbólicos: - o problema foi reconhecido. Este ícone é sempre exibido primeiro. - comentários foram feitos. O número de comentários também é exibido. - a gravidade do problema foi aumentada (por exemplo, Informação → Aviso). - a gravidade do problema foi reduzida (por exemplo, Aviso → Informação). - a gravidade do problema foi alterada, mas retornou ao nível original (por exemplo, Aviso → Informação → Aviso). - o problema foi suprimido. - o problema foi dessuprimido. - ações foram tomadas. O número de ações também é exibido. - ações foram tomadas, pelo menos uma está em andamento. O número de ações também é exibido. - ações foram tomadas, pelo menos uma falhou. O número de ações também é exibido. Ao passar o mouse sobre os ícones, pop-ups com detalhes sobre a atividade são exibidos. Veja visualizando detalhes para saber mais sobre os ícones usados no pop-up para ações tomadas.
Tags	Tags são exibidas (se houver). Além disso, tags de um sistema externo de tickets também podem ser exibidas (veja a opção *Processar tags* ao configurar webhooks).

Dados operacionais de problemas

É possível exibir dados operacionais para problemas atuais, ou seja, os valores mais recentes dos items em oposição aos valores dos items no momento do problema.

A exibição dos dados operacionais pode ser configurada no filtro de *Monitoramento* > *Problemas* ou na configuração do respectivo widget do dashboard, selecionando uma das três opções:

- *Nenhum* - nenhum dado operacional é exibido
- *Separadamente* - os dados operacionais são exibidos em uma coluna separada
- *Com o nome do problema* - os dados operacionais são anexados ao nome do problema e entre parênteses. Os dados operacionais são anexados ao nome do problema apenas se o campo *Dados operacionais* não estiver vazio na configuração do trigger.

O conteúdo dos dados operacionais pode ser configurado em cada trigger, no campo *Dados operacionais*. Este campo aceita uma string arbitrária com macros, mais importante, a macro `{ITEM.LASTVALUE<1-9>}`.

`{ITEM.LASTVALUE<1-9>}` neste campo sempre será resolvido para os valores mais recentes dos items na expressão do trigger. `{ITEM.VALUE<1-9>}` neste campo será resolvido para os valores dos items no momento da mudança de status do trigger (ou seja, mudança para problema, mudança para OK, sendo fechado manualmente por um usuário ou sendo fechado por correlação).

Observe que fechar o problema manualmente não produz um novo valor, portanto, o valor resolvido de `{ITEM.LASTVALUE<1-9>}` ou `{ITEM.VALUE<1-9>}` ainda mostrará o valor do momento do problema.

`{ITEM.LASTVALUE<1-9>}` ou `{ITEM.VALUE<1-9>}` será resolvido como *UNKNOWN* se o valor mais recente do histórico tiver sido coletado há mais tempo do que o período definido em *Período máximo de exibição do histórico* (veja *Administração > Geral*).

Duração negativa do problema

Na verdade, é possível em algumas situações comuns ter uma duração negativa do problema, ou seja, quando o tempo de resolução do problema é anterior ao tempo de criação do problema, por exemplo:

- Se algum host for monitorado por proxy e ocorrer um erro de rede, levando a nenhum dado recebido do proxy por um tempo, o trigger nodata(/host/key) será disparado pelo server. Quando a conexão for restaurada, o server receberá dados do item do proxy com um tempo do passado. Então, o problema nodata(/host/key) será resolvido e terá uma duração negativa do problema;
- Quando os dados do item que resolvem o evento do problema são enviados pelo Zabbix sender e contêm um timestamp anterior ao tempo de criação do problema, uma duração negativa do problema também será exibida.

Note:
A duração negativa do problema não afeta o cálculo de SLA ou o Relatório de disponibilidade de um trigger específico de forma alguma; ela não reduz nem expande o tempo do problema.

Opções de edição em massa

Os botões abaixo da lista oferecem algumas opções de edição em massa:

- *Atualização em massa* - atualiza os problemas selecionados navegando até a tela de atualização de problemas

Para usar esta opção, marque as caixas de seleção antes dos respectivos problemas e clique no botão *Atualização em massa*.

Botões

O botão à direita oferece a seguinte opção:

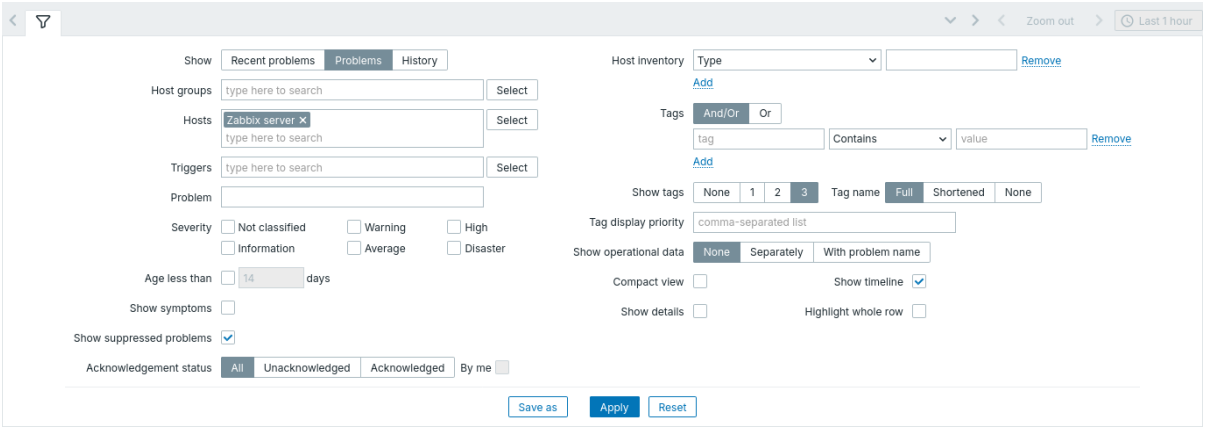
Export to CSV	Exportar o conteúdo de todas as páginas para um arquivo CSV.

Os botões do modo de visualização, sendo comuns a todas as seções, estão descritos na página Monitoramento.

Usando o filtro

Você pode usar o filtro para exibir apenas os problemas do seu interesse. Para melhor desempenho na pesquisa, os dados são pesquisados com macros não resolvidas.

O filtro está localizado acima da tabela. As configurações de filtro favoritas podem ser salvas como abas e, em seguida, acessadas rapidamente clicando nas abas acima do filtro.



Parâmetro	Descrição
Exibir	Filtrar por status do problema: **Problemas recentes** - problemas não resolvidos e recentemente resolvidos são exibidos (padrão) **Problemas** - apenas problemas não resolvidos são exibidos **Histórico** - o histórico de todos os eventos é exibido
Grupos de hosts	Filtrar por um ou mais grupos de hosts. Ao especificar um grupo de hosts pai, todos os grupos de hosts aninhados são implicitamente selecionados.
Hosts	Filtrar por um ou mais hosts.
Triggers	Filtrar por um ou mais triggers.

Parâmetro	Descrição
Problema	Filtrar por nome do problema.
Gravidade	Filtrar por gravidade do trigger (problema).
Idade menor que	Filtrar por quanto tempo o problema existe.
Exibir sintomas	Marque a caixa de seleção para exibir em sua própria linha os problemas classificados como sintomas.
Exibir problemas suprimidos	Marque a caixa de seleção para exibir problemas que, de outra forma, seriam suprimidos (não exibidos) devido à manutenção do host ou à supressão de problema individual.
Status de reconhecimento	Filtrar para exibir todos os problemas, apenas problemas não reconhecidos ou apenas problemas reconhecidos. Marque a caixa de seleção adicional para filtrar apenas os problemas já reconhecidos por você.
Inventário do host	Filtrar por tipo e valor de inventário.
Tags	Filtrar por nome e valor da tag de evento. É possível incluir ou excluir tags e valores de tags específicos. Várias condições podem ser definidas. A correspondência do nome da tag diferencia maiúsculas de minúsculas. Há vários operadores disponíveis para cada condição: **Existe** - inclui os nomes de tags especificados **Igual** - inclui os nomes e valores de tags especificados (diferencia maiúsculas de minúsculas) **Contém** - inclui os nomes de tags especificados cujos valores de tag contenham a string inserida (correspondência de substring, não diferencia maiúsculas de minúsculas) **Não existe** - exclui os nomes de tags especificados **Diferente** - exclui os nomes e valores de tags especificados (diferencia maiúsculas de minúsculas) **Não contém** - exclui os nomes de tags especificados cujos valores de tag contenham a string inserida (correspondência de substring, não diferencia maiúsculas de minúsculas) Existem dois tipos de cálculo para as condições: **E/Ou** - todas as condições devem ser atendidas, condições com o mesmo nome de tag serão agrupadas pela condição Ou **Ou** - basta que uma condição seja atendida Quando filtradas, as tags especificadas aqui serão exibidas primeiro com o problema, a menos que sejam substituídas pela lista *Prioridade de exibição da tag* (veja abaixo).
Exibir tags	Selecione o número de tags exibidas: **Nenhuma** - nenhuma coluna *Tags* em *Monitoramento* > *Problemas* **1** - a coluna *Tags* contém uma tag **2** - a coluna *Tags* contém duas tags **3** - a coluna *Tags* contém três tags Para ver todas as tags do problema, passe o mouse sobre o ícone de três pontos.
Nome da tag	Selecione o modo de exibição do nome da tag: **Completo** - nomes e valores das tags são exibidos por completo **Abreviado** - nomes das tags são abreviados para 3 símbolos; valores das tags são exibidos por completo **Nenhum** - apenas os valores das tags são exibidos; sem nomes
Prioridade de exibição da tag	Informe a prioridade de exibição da tag para um problema, como uma lista de tags separadas por vírgula (por exemplo: `cliente,escopo,componente`). Devem ser usados apenas os nomes das tags, sem valores. As tags desta lista sempre serão exibidas primeiro, substituindo a ordenação natural por ordem alfabética.
Exibir dados operacionais	Selecione o modo de exibição dos dados operacionais: **Nenhum** - nenhum dado operacional é exibido **Separadamente** - os dados operacionais são exibidos em uma coluna separada **Com o nome do problema** - adiciona os dados operacionais ao nome do problema, usando parênteses para os dados operacionais
Visualização compacta	Marque a caixa de seleção para habilitar a visualização compacta.
Exibir detalhes	Marque a caixa de seleção para exibir as expressões de trigger subjacentes dos problemas. Desabilitado se a caixa de seleção *Visualização compacta* estiver marcada.
Exibir linha do tempo	Marque a caixa de seleção para exibir a linha do tempo visual e o agrupamento. Desabilitado se a caixa de seleção *Visualização compacta* estiver marcada.
Destacar linha inteira	Marque a caixa de seleção para destacar a linha completa para problemas não resolvidos. A cor da gravidade do problema é usada para destacar. *Destacar linha inteira* não está disponível nos temas de alto contraste. Para versões do Zabbix anteriores à 7.0.19, esta opção é suportada apenas quando a caixa de seleção *Visualização compacta* está marcada nos temas padrão azul e escuro.

Abas para filtros favoritos

Conjuntos de parâmetros de filtro usados com frequência podem ser salvos em abas.

Para salvar um novo conjunto de parâmetros de filtro, abra a aba principal, configure as configurações do filtro e pressione o botão *Salvar como*. Em uma nova janela pop-up, defina as *Propriedades do filtro*.

problem_filter0.png

Parâmetro	Descrição
Nome	O nome do filtro a ser exibido na lista de abas.
Mostrar número de registros	Marque se deseja que o número de problemas seja exibido ao lado do nome da aba.
Substituir seletor de período de tempo	Marque para definir um período de tempo padrão específico para este conjunto de filtros. Se definido, você só poderá alterar o período de tempo para esta aba atualizando as configurações do filtro. Para abas sem um período de tempo personalizado, o intervalo de tempo pode ser alterado pressionando o botão seletor de tempo no canto superior direito (o nome do botão depende do intervalo de tempo selecionado: Esta semana, Últimos 30 minutos, Ontem, etc.). Esta opção está disponível apenas para filtros em *Monitoramento* > *Problemas*.
De/Até	Início e fim do período de tempo na sintaxe absoluta (`Y-m-d H:i:s`) ou relativa (`now-1d`). Disponível se *Definir período de tempo personalizado* estiver marcado.

Quando salvo, o filtro é criado como uma aba de filtro nomeada e imediatamente ativada.

Para editar as propriedades do filtro de um filtro existente, pressione o símbolo de engrenagem ao lado do nome da aba ativa.

problem_filter2.png

Notas:

- Para ocultar a área do filtro, clique no nome da aba atual. Clique novamente no nome da aba ativa para abrir a área do filtro novamente.
- A navegação por teclado é suportada: use as setas para alternar entre as abas, pressione *Enter* para abrir.
- Os botões esquerdo/direito acima do filtro podem ser usados para alternar entre os filtros salvos. Alternativamente, o botão apontando para baixo abre um menu suspenso com todos os filtros salvos e você pode clicar no que precisa.
- As abas de filtro podem ser reorganizadas arrastando e soltando.
- Se as configurações de um filtro salvo foram alteradas (mas não salvas), um ponto verde é exibido após o nome do filtro. Para atualizar o filtro de acordo com as novas configurações, clique no botão *Atualizar*, que é exibido no lugar do botão *Salvar como*.
- As configurações atuais do filtro são lembradas no perfil do usuário. Quando o usuário abrir a página novamente, as configurações do filtro permanecerão as mesmas.

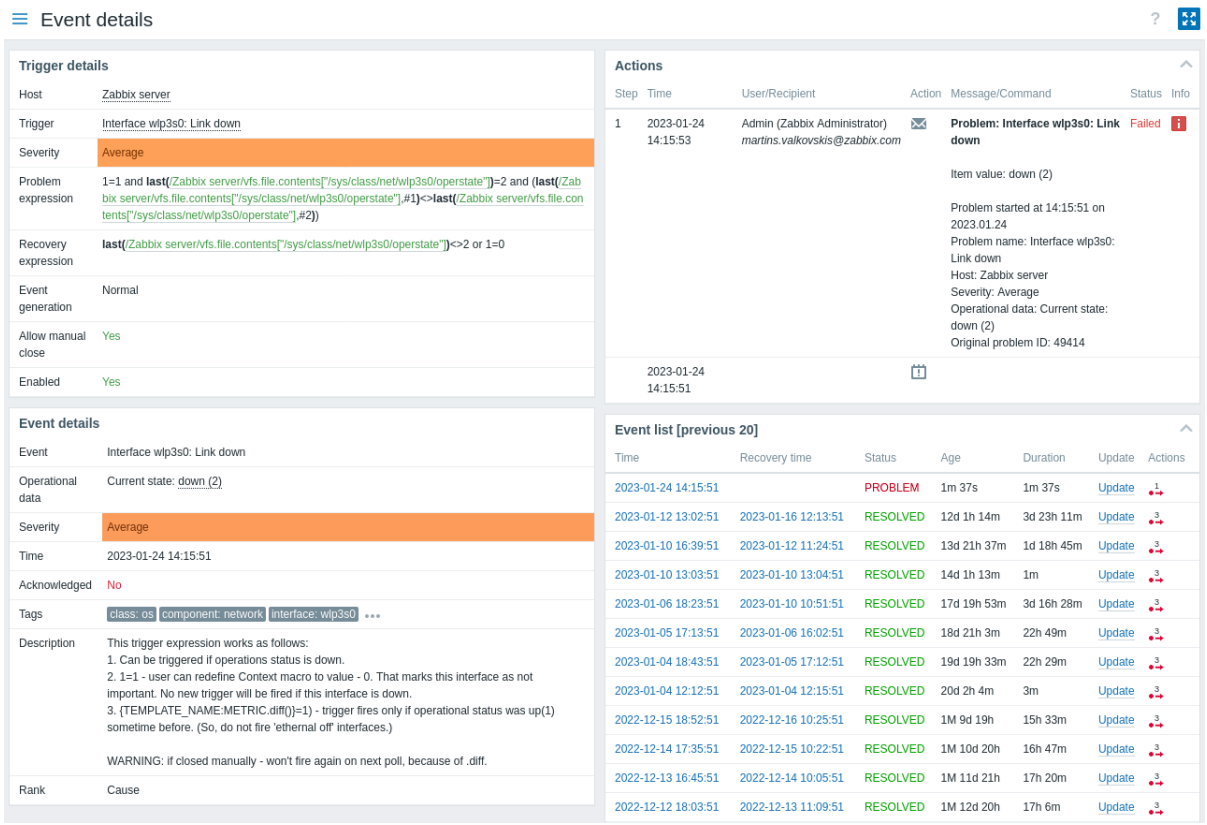
Note:
Para compartilhar filtros, copie e envie para outros a URL de um filtro ativo. Após abrir esta URL, outros usuários poderão salvar este conjunto de parâmetros como um filtro permanente em sua conta Zabbix.
 Veja também: Parâmetros de página.

Botões de filtro

Apply	Aplicar os critérios de filtragem especificados (sem salvar).
Reset	Redefinir o filtro atual e retornar aos parâmetros salvos da aba atual. Na aba principal, isso limpará o filtro.
Save as	Salvar os parâmetros atuais do filtro em uma nova aba. Disponível apenas na aba principal.
Update	Substituir os parâmetros da aba pelos parâmetros especificados atualmente. Não disponível na aba principal.

Visualizando detalhes

Os horários de início e recuperação do problema em *Monitoramento* > *Problemas* são links. Clicando neles, mais detalhes do evento são exibidos.



Observe que a gravidade do problema pode ser diferente para o trigger e para o evento do problema - se ela tiver sido atualizada para o evento do problema usando a tela *Atualizar problema*.

Na lista de ações, os seguintes ícones são usados para indicar o tipo de atividade:

- - evento de problema gerado
- - mensagem enviada
- - evento de problema reconhecido
- - evento de problema não reconhecido
- - um comentário foi adicionado
- - a gravidade do problema foi aumentada (por exemplo, Informação → Aviso)
- - a gravidade do problema foi diminuída (por exemplo, Aviso → Informação)
- - a gravidade do problema foi alterada, mas retornou ao nível original (por exemplo, Aviso → Informação → Aviso)
- - um comando remoto foi executado
- - o evento de problema foi recuperado
- - o problema foi fechado manualmente
- - o problema foi suprimido
- - o problema foi dessuprimido
- - o problema foi convertido em um problema sintoma
- - o problema foi convertido em um problema causa

1 Problemas de causa e sintoma

Visão geral

Por padrão, todos os novos problemas são classificados como problemas de causa. É possível reclassificar manualmente certos problemas como problemas de sintoma do problema de causa.

Por exemplo, uma queda de energia pode ser a verdadeira causa raiz de um host estar inacessível ou de algum serviço estar fora do ar. Nesse caso, os problemas "host está inacessível" e "serviço está fora do ar" devem ser classificados como problemas de sintoma de "queda de energia" - o problema de causa.

A hierarquia causa-sintoma suporta apenas dois níveis. Um problema que já é um sintoma não pode ter problemas de sintoma "subordinados" atribuídos a ele; quaisquer problemas atribuídos como sintomas a um problema de sintoma se tornarão sintomas do mesmo problema de causa.

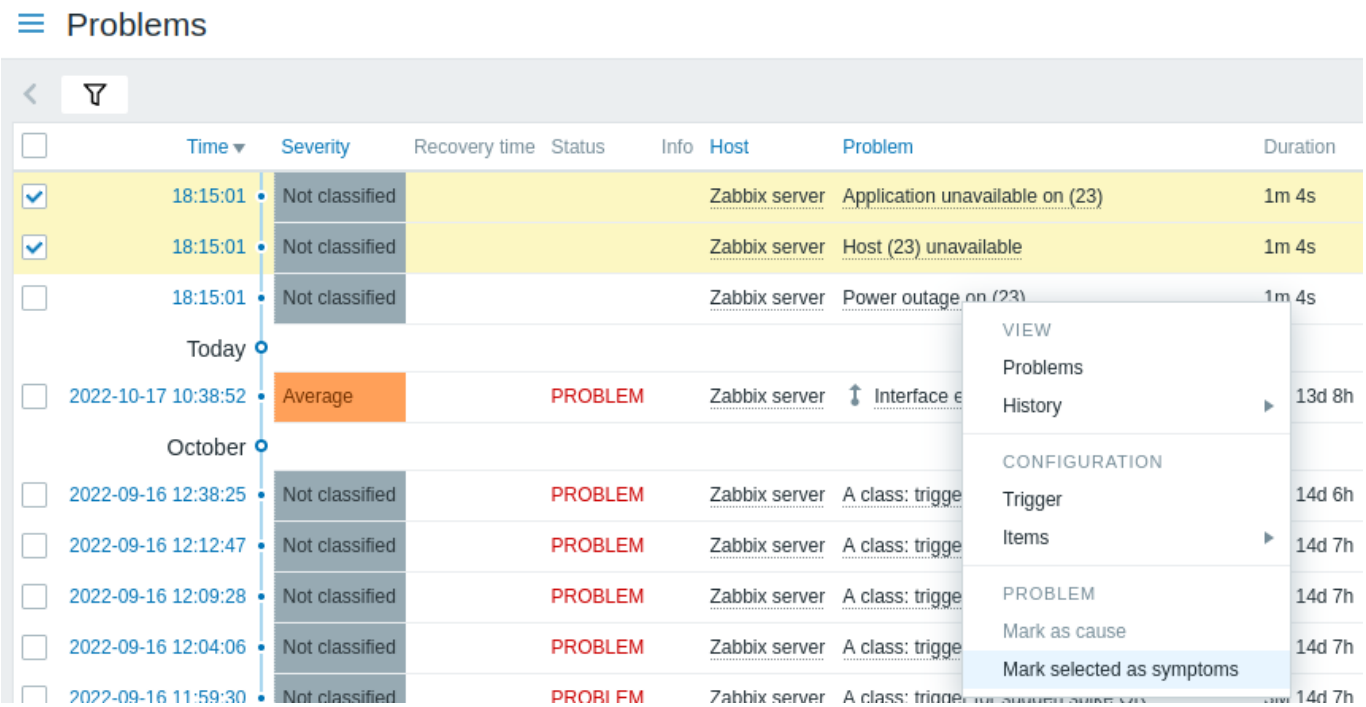
Apenas problemas de causa são contabilizados nos totais de problemas em mapas, widgets do dashboard como *Problemas por severidade* ou *Hosts com problemas*, etc. No entanto, a classificação de problemas não afeta os serviços.

Um problema de sintoma pode ser vinculado a apenas um problema de causa. Problemas de sintoma não são resolvidos automaticamente se o problema de causa for resolvido ou fechado.

Configuração

Para reclassificar um problema como um problema sintoma, primeiro selecione-o na lista de problemas. Um ou vários problemas podem ser selecionados.

Em seguida, vá para o problema causa e, em seu menu de contexto, clique na opção *Marcar selecionados como sintomas*.



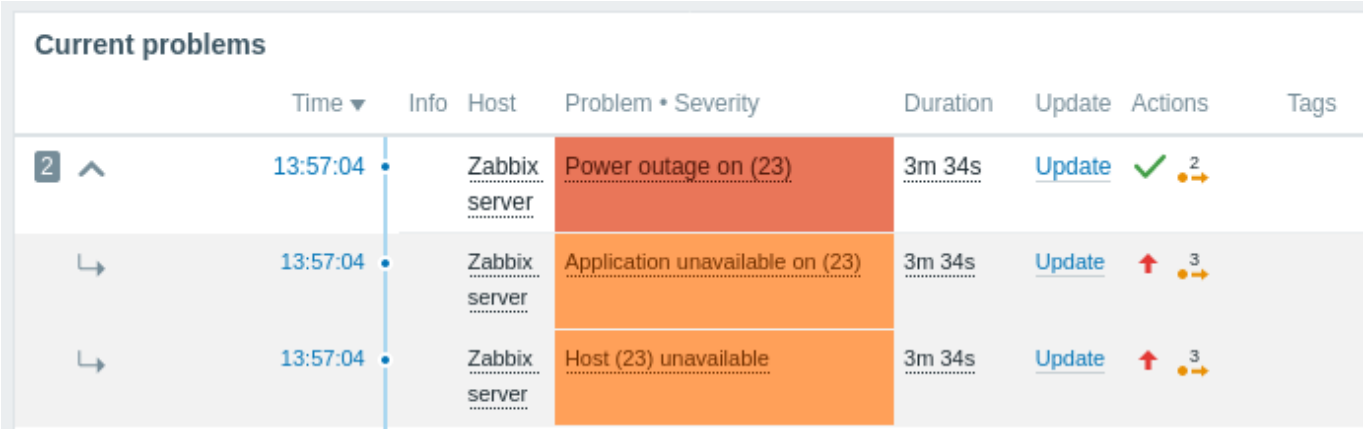
Depois disso, os problemas selecionados serão atualizados pelo server para problemas sintoma do problema causa.

Enquanto o status do problema está sendo atualizado, ele é exibido de uma das duas maneiras:

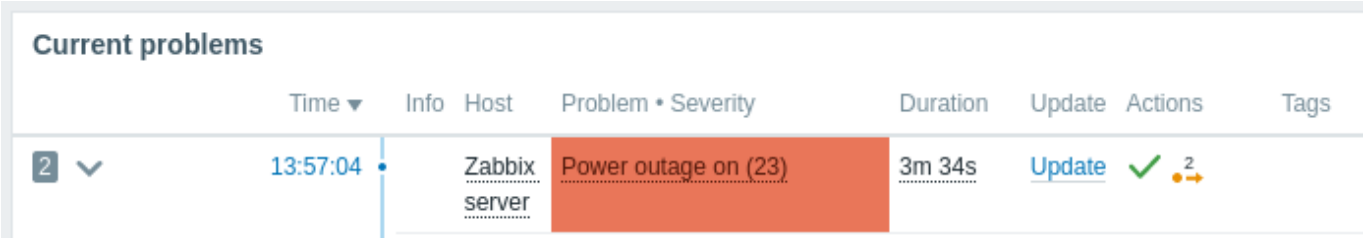
- Um status "ATUALIZANDO" piscando é exibido na coluna Status;
- Um ícone piscando ou na coluna Info (isto ocorre se apenas *Problemas* estiverem selecionados no filtro e, portanto, a coluna Status não for exibida).

Exibição

Os problemas de sintoma são exibidos abaixo do problema de causa e marcados de acordo em *Monitoramento -> Problemas* (e no widget de painel *Problemas*) - com um ícone, fonte menor e fundo diferente.



Na visualização recolhida, apenas o problema de causa é visto; a existência de problemas de sintoma é indicada pelo número no início da linha e pelo ícone para expandir a visualização.



Também é possível exibir adicionalmente os problemas de sintoma em fonte normal e em sua própria linha. Para isso, selecione *Mostrar sintomas* nas configurações do filtro ou na configuração do widget.

Reverter para problema de causa

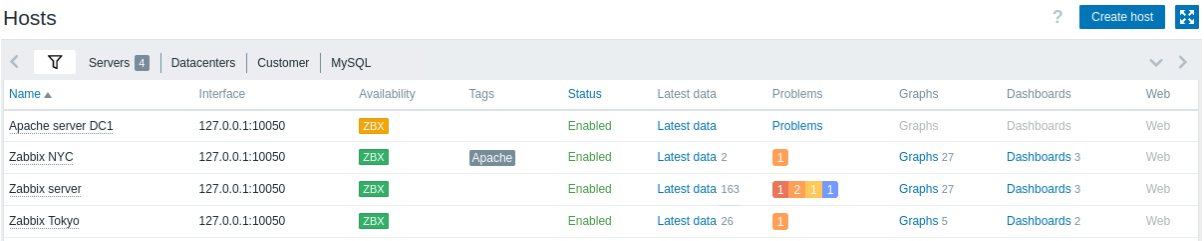
Um problema de sintoma pode ser revertido para um problema de causa. Para isso, faça o seguinte:

- clique na opção *Marcar como causa* no menu de contexto do problema de sintoma;
- marque a opção *Converter para causa* na tela de atualização de problema e clique em *Atualizar* (esta opção também funcionará se vários problemas forem selecionados).

2 Hosts

Visão geral

A seção *Monitoramento* → *Hosts* exibe uma lista completa de hosts monitorados com informações detalhadas sobre a interface do host, disponibilidade, tags, problemas atuais, status (habilitado/desabilitado) e links para navegar facilmente até os últimos dados do host, histórico de problemas, gráficos, dashboards e cenários web.



Coluna	Descrição
Nome	O nome visível do host. Clicar no nome exibe o menu do host. Um ícone de chave inglesa laranja após o nome indica que este host está em manutenção. Clique no cabeçalho da coluna para classificar os hosts por nome em ordem crescente ou decrescente.
Interface	A interface principal do host é exibida.

Coluna	Descrição
Disponibilidade	A disponibilidade do host por interface configurada é exibida. Os ícones de disponibilidade representam o status atual da interface do host no Zabbix server. Portanto, se você desabilitar um host no frontend, sua disponibilidade será atualizada após o Zabbix server sincronizar as alterações de configuração. Da mesma forma, se você habilitar um host, sua disponibilidade será atualizada após o Zabbix server sincronizar as alterações de configuração e coletar dados do host. Os ícones de disponibilidade representam apenas os tipos de interface (Agent, SNMP, IPMI, JMX) que estão configurados. Ao passar o mouse sobre o ícone, é exibido um pop-up com uma lista de todas as interfaces do mesmo tipo com detalhes, status e erros. Para a interface Agent, o pop-up exibe interfaces (passivas) e verificações ativas. Se um host tiver apenas verificações ativas, o ícone da interface Agent será exibido mesmo que o host não tenha uma interface Agent configurada. A coluna fica vazia para hosts sem interfaces. O status de uma única interface de host é determinado pela conexão entre um item habilitado que usa a interface e o host. O status pode ser: **Disponível** - a conexão com o host foi bem-sucedida; **Não disponível** - a conexão com o host não foi bem-sucedida (timeout, problemas de firewall, etc.); **Desconhecido** - a conexão com o host não foi tentada ou o resultado é desconhecido. Para detalhes adicionais sobre como o Zabbix server determina o status da interface, consulte Status de interface desconhecido e Configurações de interface de host inalcançável/indisponível. O status de todas as interfaces de host de um único tipo (Agent, SNMP, IPMI, JMX) é determinado por aquelas interfaces que são usadas por pelo menos um item habilitado. O status é indicado pela cor do ícone: **Verde** - todas as interfaces estão disponíveis; **Amarelo** - pelo menos uma interface não está disponível e pelo menos uma está disponível ou desconhecida; **Vermelho** - todas as interfaces não estão disponíveis; **Cinza** - pelo menos uma interface está desconhecida, mas nenhuma está indisponível. Antes do Zabbix 7.0.10, esse status é determinado por todas as interfaces do mesmo tipo, independentemente de algum item usá-las. **Disponibilidade de verificação ativa.** Se pelo menos uma verificação ativa estiver habilitada no host, as verificações ativas também afetam a disponibilidade total da interface Agent conforme descrito acima. Para determinar a disponibilidade da verificação ativa, mensagens de heartbeat são enviadas na thread de verificação ativa do agent. A frequência das mensagens de heartbeat é controlada pelo parâmetro `HeartbeatFrequency` na configuração do Zabbix agent ou agent 2 (padrão 60 segundos, intervalo 0-3600). As verificações ativas são consideradas indisponíveis quando o heartbeat da verificação ativa é mais antigo que 2 x HeartbeatFrequency segundos. **Nota:** Zabbix agents anteriores à versão 6.2.x não enviam heartbeats de verificação ativa, portanto, a disponibilidade de seus hosts permanece desconhecida.
Tags	Tags do host e de todos os templates vinculados, com macros não resolvidas.
Status	Status do host - *Habilitado* ou *Desabilitado*. Clique no cabeçalho da coluna para classificar os hosts por status em ordem crescente ou decrescente.
Últimos dados	Clicar no link abrirá a página *Monitoramento* **-** *Últimos dados* com todos os últimos dados coletados do host. O número de items com os últimos dados é exibido em cinza.
Problemas	O número de problemas abertos do host classificados por severidade. A cor do quadrado indica a severidade do problema. O número no quadrado significa o número de problemas para a severidade dada. Clicar no ícone abrirá a página *Monitoramento* **-** *Problemas* para o host atual. Se um host não tiver problemas, um link para a seção *Problemas* para este host é exibido como texto. Use o filtro para selecionar se problemas suprimidos devem ser incluídos (não incluídos por padrão).

Coluna	Descrição
Gráficos	Clicar no link exibirá os gráficos configurados para o host. O número de gráficos é exibido em cinza. Se um host não tiver gráficos, o link é desabilitado (texto em cinza) e nenhum número é exibido.
Dashboards	Clicar no link exibirá os dashboards configurados para o host. O número de dashboards é exibido em cinza. Se um host não tiver dashboards, o link é desabilitado (texto em cinza) e nenhum número é exibido.
Web	Clicar no link exibirá os cenários web configurados para o host. O número de cenários web é exibido em cinza. Se um host não tiver cenários web, o link é desabilitado (texto em cinza) e nenhum número é exibido.

Botões

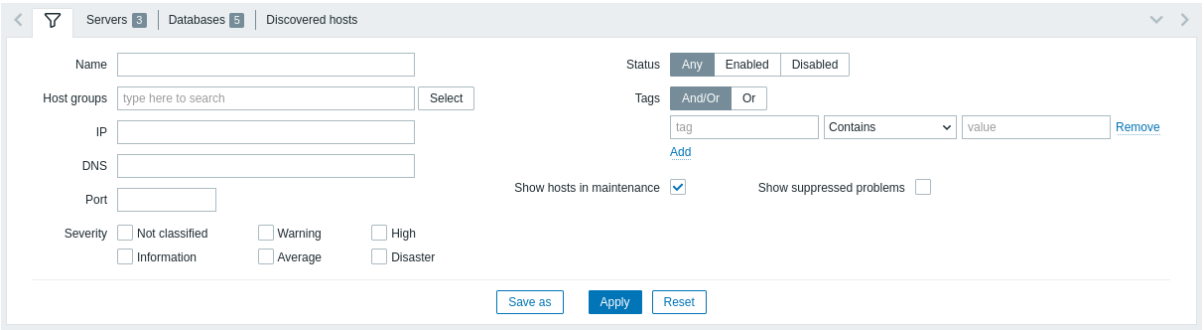
Criar host permite criar um novo host. Este botão está disponível apenas para usuários Admin e Super Admin.

Os botões do modo de visualização comuns a todas as seções estão descritos na Monitoramento página.

Usando o filtro

Você pode usar o filtro para exibir apenas os hosts do seu interesse. Para melhor desempenho na pesquisa, os dados são pesquisados com macros não resolvidas.

O filtro está localizado acima da tabela. É possível filtrar hosts por nome, grupo de hosts, IP ou DNS, porta da interface, tags, severidade do problema, status (habilitado/desabilitado/qualquer); você também pode selecionar se deseja exibir problemas suprimidos e hosts que estão atualmente em manutenção.



Parâmetro	Descrição
Nome	Filtra pelo nome visível do host.
Grupos de hosts	Filtra por um ou mais grupos de hosts. Especificar um grupo de hosts pai seleciona implicitamente todos os grupos de hosts aninhados.
IP	Filtra pelo endereço IP.
DNS	Filtra pelo nome DNS.
Porta	Filtra pelo número da porta.
Severidade	Filtra pela severidade do problema. Por padrão, problemas de todas as severidades são exibidos. Os problemas são exibidos se não estiverem suprimidos.
Status	Filtra pelo status do host.

Parâmetro	Descrição
Tags	Filtra pelo nome e valor da tag do host. Os hosts podem ser filtrados por tags no nível do host, bem como por tags de todos os templates vinculados, incluindo templates aninhados. É possível incluir ou excluir tags e valores de tags específicos. Várias condições podem ser definidas. A correspondência do nome da tag diferencia maiúsculas de minúsculas. Há vários operadores disponíveis para cada condição: **Existe** - inclui os nomes de tags especificados **Igual** - inclui os nomes e valores de tags especificados (diferencia maiúsculas de minúsculas) **Contém** - inclui os nomes de tags especificados onde os valores das tags contêm a string inserida (correspondência de substring, não diferencia maiúsculas de minúsculas) **Não existe** - exclui os nomes de tags especificados **Não é igual** - exclui os nomes e valores de tags especificados (diferencia maiúsculas de minúsculas) **Não contém** - exclui os nomes de tags especificados onde os valores das tags contêm a string inserida (correspondência de substring, não diferencia maiúsculas de minúsculas) Existem dois tipos de cálculo para as condições: **E/Ou** - todas as condições devem ser atendidas, condições com o mesmo nome de tag serão agrupadas pela condição Ou **Ou** - basta que uma condição seja atendida
Mostrar hosts em manutenção	Marque a caixa de seleção para exibir hosts que estão em manutenção (exibido por padrão).
Mostrar problemas suprimidos	Marque a caixa de seleção para exibir problemas que, de outra forma, seriam suprimidos (não exibidos) devido à manutenção do host ou à supressão de problemas individual.

Salvando o filtro

As configurações de filtro favoritas podem ser salvas como abas e, em seguida, acessadas rapidamente clicando na respectiva aba acima do filtro.

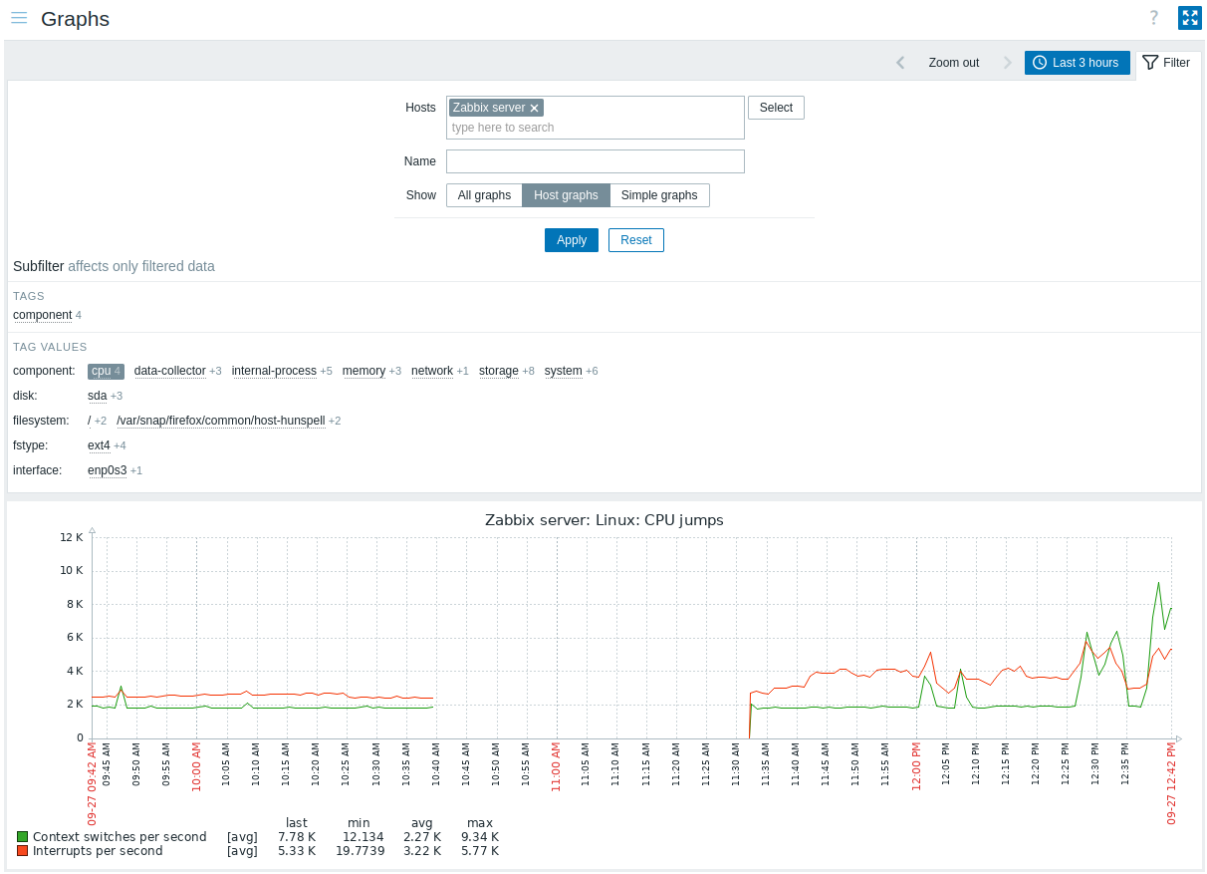
Veja mais detalhes sobre salvar filtros.

1 Gráficos

Visão geral

Os gráficos do host podem ser acessados em *Monitoramento* → *Hosts* clicando em Gráficos para o respectivo host.

Qualquer gráfico personalizado que tenha sido configurado para o host pode ser exibido, assim como qualquer gráfico simples.



Os gráficos são classificados por:

- nome do gráfico (gráficos personalizados)
- nome do item (gráficos simples)

Os gráficos de hosts desabilitados também são acessíveis.

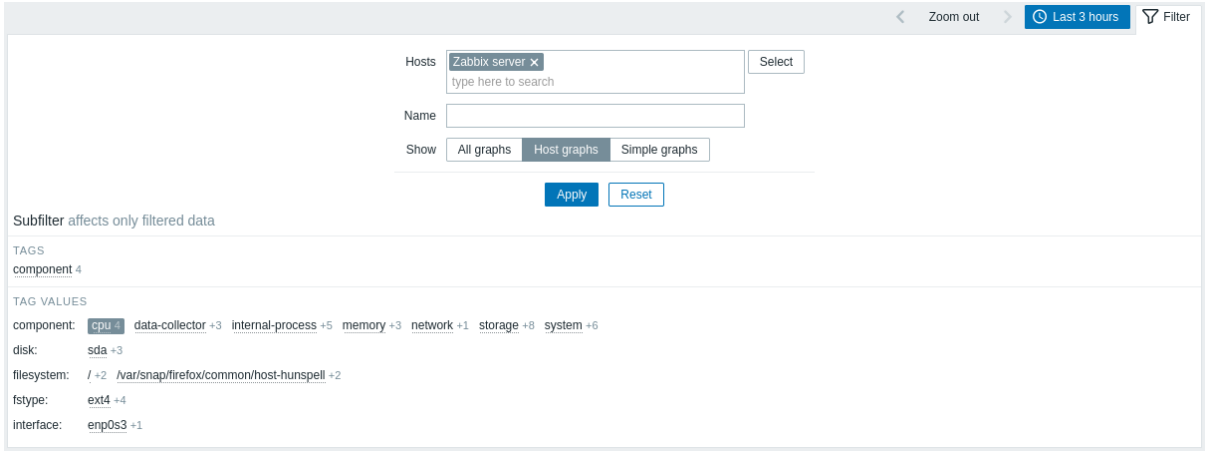
Seletor de período de tempo

Observe o seletor de período de tempo acima do gráfico. Ele permite selecionar períodos frequentemente necessários com um clique do mouse.

Veja também: *Seletor de período de tempo*

Usando o filtro

Para visualizar um gráfico específico, selecione-o no filtro. O filtro permite especificar o host, o nome do gráfico e a opção *Mostrar* (todos/gráficos do host/gráficos simples).



Se nenhum host for selecionado no filtro, nenhum gráfico será exibido.

Usando subfiltro

O subfiltro permite modificar ainda mais o filtro a partir do filtro principal.

Ele contém links clicáveis para acesso rápido a gráficos relacionados. Os gráficos são relacionados por uma entidade comum - nome ou valor da tag. Quando um nome/valor de tag é clicado, ele é destacado com um fundo cinza, e os gráficos são imediatamente filtrados (não é necessário clicar em *Aplicar* no filtro principal). Clicar em outro nome/valor de tag o adiciona aos resultados filtrados. Clicar novamente no nome/valor da tag remove o filtro.

Os subfiltros são gerados com base nos dados filtrados, que são limitados a 1000 registros. Se você quiser ver mais registros no subfiltro, é necessário aumentar o valor do parâmetro *Limite para resultados de pesquisa e filtro* (em *Administração -> Geral -> GUI*).

Ao contrário do filtro principal, o subfiltro é atualizado junto com cada solicitação de atualização da tabela para sempre obter informações atualizadas das opções de filtragem disponíveis e seus números de contagem.

O número de entidades exibidas é limitado a 100 horizontalmente. Se houver mais, um ícone de reticências é exibido no final; ele não é clicável. Listas verticais (como tags com seus valores) são limitadas a 20 entradas. Se houver mais, um ícone de reticências é exibido; ele não é clicável.

Um número ao lado de cada entidade clicável indica o número de gráficos que ela possui nos resultados do filtro principal.

Assim que uma entidade é selecionada, os números com outras entidades disponíveis são exibidos com um sinal de mais indicando quantos gráficos podem ser adicionados à seleção atual.

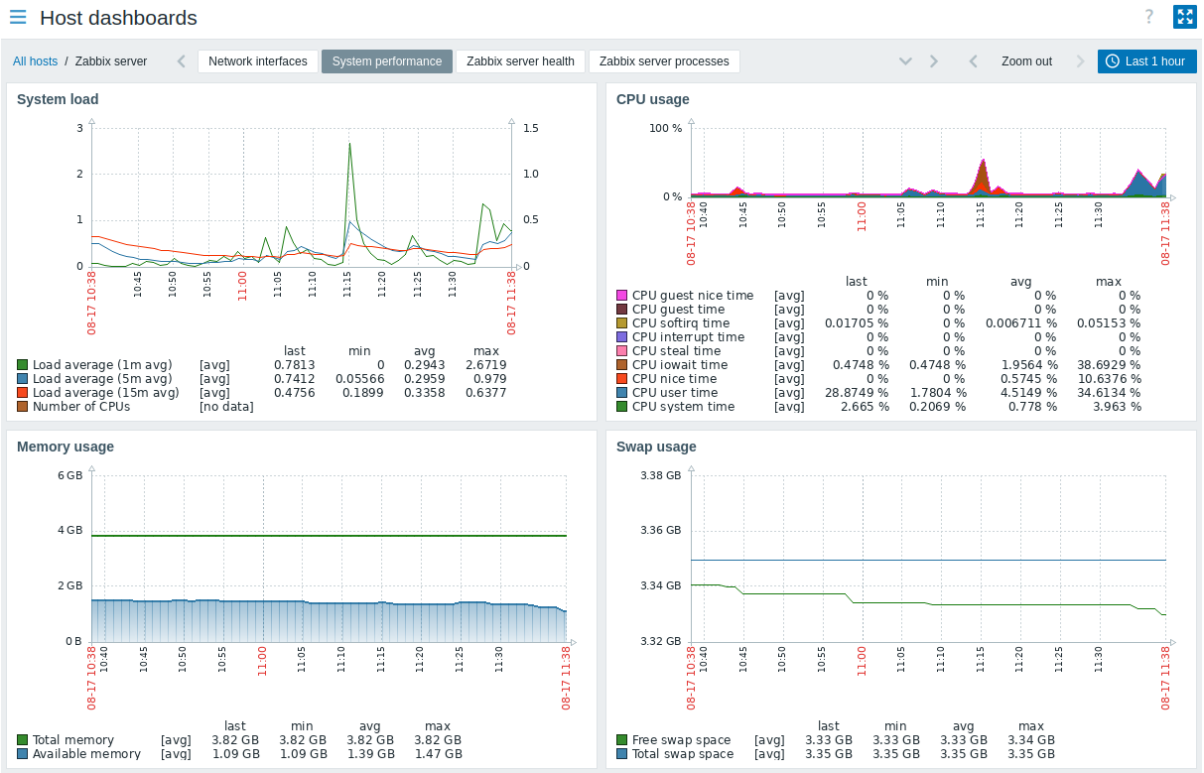
Botões

Os botões do modo de visualização, comuns a todas as seções, estão descritos na página Monitoramento.

2 Dashboards de host

Visão geral

Os dashboards de host são semelhantes aos dashboards globais; no entanto, os dashboards de host não possuem um proprietário e exibem dados apenas para o host selecionado.



Ao visualizar dashboards de host, você pode alternar entre os dashboards configurados clicando em:

- nas abas do dashboard;
- nos botões de seta abaixo do cabeçalho;
- no botão de seta abaixo do cabeçalho, que exibirá a lista completa de dashboards de host disponíveis.

Para alternar para a seção *Monitoramento → Hosts*, clique no link de navegação *Todos os hosts* sob o cabeçalho no canto superior esquerdo.

Configuração

Os dashboards de host são configurados no nível de template. Uma vez que um template é vinculado a um host, os dashboards de host são gerados para esse host. Observe que os dashboards de host *não podem* ser configurados na seção *Dashboards*, que é reservada para dashboards globais.

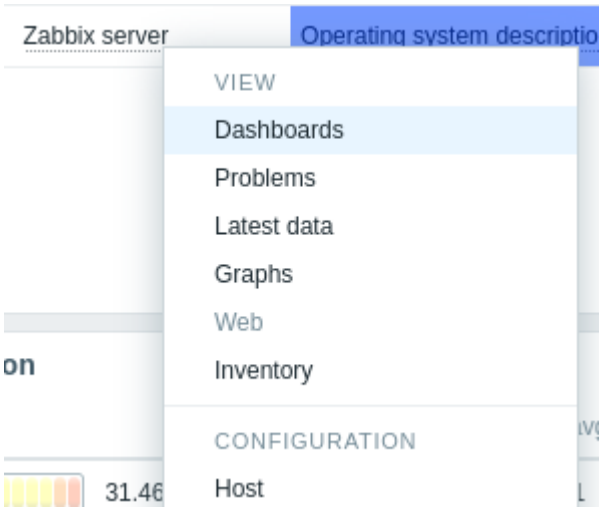
Os widgets dos dashboards de host também só podem ser configurados no nível de template, exceto para alteração do intervalo de atualização. Além disso, os widgets dos dashboards de host só podem ser copiados para outros dashboards de host dentro do mesmo template. Observe que widgets de dashboards globais *não podem* ser copiados para dashboards de host.

Note:
Os dashboards de host eram telas de host antes do Zabbix 5.2. Ao importar um template antigo que contenha telas, a importação da tela será ignorada.

Acesso

Os dashboards de host podem ser acessados:

- após pesquisar por um nome de host na pesquisa global (clique no link *Dashboards* fornecido nos resultados da pesquisa);
- após clicar em um nome de host em *Inventário* → *Hosts* (clique no link *Dashboards* fornecido na visão geral do host);
- no menu do host clicando em *Dashboards*.



Observe que os dashboards de host *não podem* ser acessados diretamente na seção *Dashboards*, que é reservada para dashboards globais.

3 Cenários web

Visão geral

As informações do cenário web do host podem ser acessadas em *Monitoramento* → *Hosts* clicando em Web para o respectivo host.

Clicar no nome do host exibe o menu do host. Os dados de hosts desabilitados também são acessíveis. O nome de um host desabilitado é listado em vermelho.

O número máximo de cenários exibidos por página depende da configuração *Linhas por página* no perfil do usuário.

Por padrão, apenas os valores que estão dentro das últimas 24 horas são exibidos. Esse limite foi introduzido com o objetivo de melhorar o tempo de carregamento inicial para páginas grandes de dados mais recentes. Você pode estender esse período de tempo alterando o valor do parâmetro *Período máximo de exibição do histórico* na seção de menu *Administração* → *Geral* → *GUI*.

O nome do cenário é um link para estatísticas mais detalhadas sobre ele:

Usando filtro

A página mostra uma lista de todos os cenários da web do host selecionado. Ver cenários da web para outro host ou grupo de hosts sem retornar ao *Monitoramento* → *Hosts*, selecione esse host ou grupo no filtro. Vocês também pode filtrar cenários com base em tags.

Botões

Os botões do modo de visualização comuns a todas as seções estão descritos na Monitoramento página.

3 Últimos dados

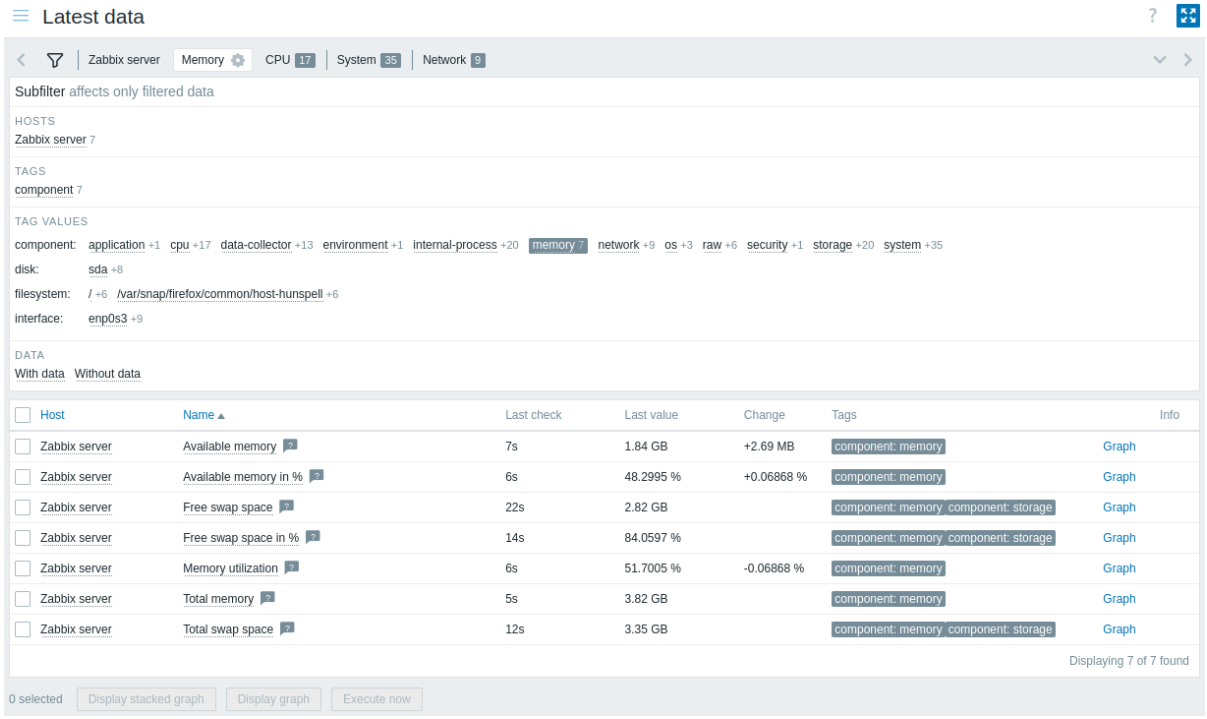
Visão geral

A seção *Monitoramento* → *Últimos dados* exibe os valores mais recentes coletados pelos items.

Esta seção contém os seguintes elementos:

- Filtro
- Subfiltro
- Lista de items

Note:
O subfiltro e a lista de items são exibidos apenas se o filtro estiver definido e houver resultados para exibir.



Coluna	Descrição
Host	Nome do host ao qual o item pertence. Clicar no nome exibe o menu de contexto do host. Se um host estiver em manutenção, um ícone de chave inglesa laranja será exibido após o nome do host. Se um host estiver desabilitado, o nome do host será exibido em vermelho. Observe que os dados de hosts desabilitados (incluindo gráficos e listas de valores de items) são acessíveis na seção *Últimos dados*.
Nome	Nome do item. Clicar no nome exibe o menu do item. Um ícone de ponto de interrogação é exibido ao lado do nome do item para todos os items que possuem uma descrição. Passe o mouse sobre o ícone para exibir uma dica de ferramenta com a descrição do item.
Última verificação	Tempo desde a última verificação do item.
Último valor	Valor mais recente do item. Os valores são exibidos com conversão de unidade e mapeamento de valor aplicados. Passe o mouse sobre o valor para exibir os dados brutos (truncados em 8192 caracteres). Valores binários são exibidos como um marcador (*valor binário*) em vez do valor real. Por padrão, apenas os valores recebidos nas últimas 24 horas são exibidos. Esse limite melhora o tempo de carregamento inicial para grandes páginas de últimos dados; para estendê-lo, atualize o valor do parâmetro *Período máximo de exibição do histórico* em *Administração → Geral → GUI*.
Alteração	Diferença entre o valor anterior e o valor mais recente. Para items com uma frequência de atualização de 1 dia ou mais, o valor da alteração nunca será exibido (com a configuração padrão). Nesse caso, o último valor não será exibido se tiver sido recebido há mais de 24 horas.
Tags	Tags associadas ao item. As tags na lista de items são clicáveis. Clicar em uma tag a habilita no subfiltro, fazendo com que a lista de items exiba apenas os items que contêm essa tag (e quaisquer outras tags previamente selecionadas no subfiltro). Observe que, uma vez que os items tenham sido filtrados dessa forma, as tags na lista de items não são mais clicáveis. Outras modificações baseadas em tags (por exemplo, para remover tags ou especificar outros filtros) devem ser feitas no subfiltro.
Gráfico/Histórico	Link para gráfico simples/histórico dos valores do item.

Coluna	Descrição
Info	Informações adicionais sobre o item. Se um item tiver erros (por exemplo, se tornou não suportado), um ícone de informação será exibido. Passe o mouse sobre o ícone para obter detalhes.

Botões

Os botões do modo de visualização, comuns a todas as seções, estão descritos na página Monitoramento.

Ações em massa

Os botões abaixo da lista oferecem ações em massa com um ou vários items selecionados:

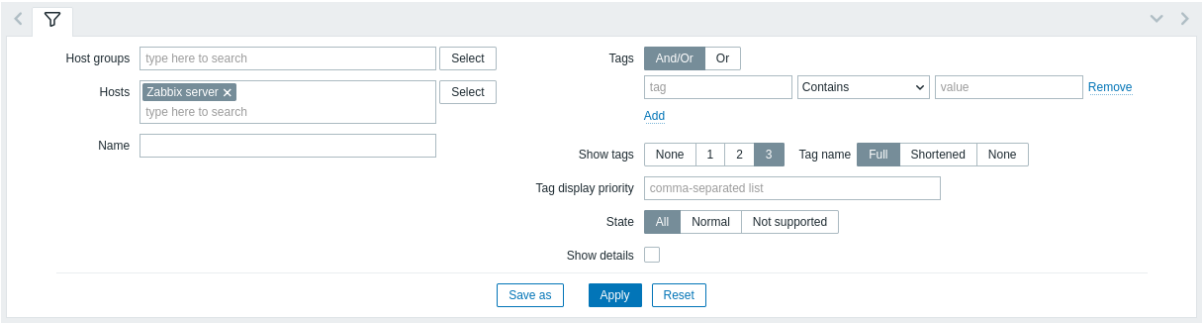
- *Exibir gráfico empilhado* - exibe um gráfico ad-hoc empilhado.
- *Exibir gráfico* - exibe um gráfico ad-hoc simples.
- *Executar agora* - executa uma verificação para novos valores de item imediatamente. Suportado apenas para verificações **passivas** (veja mais detalhes). Esta opção está disponível apenas para hosts com acesso de leitura e gravação. O acesso a esta opção para hosts com permissões somente leitura depende da opção função de usuário chamada *Invocar "Executar agora" em hosts somente leitura*.

Para usar essas opções, marque as caixas de seleção antes dos respectivos items e clique no botão desejado.

Usando o filtro

Você pode usar o filtro para exibir apenas os items do seu interesse. Para melhor desempenho na pesquisa, os dados são pesquisados com macros não resolvidas.

O ícone do filtro está localizado acima da lista de items e do subfiltro. Clique nele para expandir o filtro.



O filtro permite restringir a lista de items por grupo de hosts, host, nome do item, tag, estado e outras configurações. Ao especificar um grupo de hosts pai no filtro, todos os grupos de hosts aninhados são implicitamente selecionados. Veja *Monitoramento → Problemas* para detalhes sobre filtragem por tags.

A opção de filtro *Mostrar detalhes* permite ampliar as informações exibidas para os items. Marque-a para exibir detalhes como o intervalo de atualização do item, configurações de histórico e tendências, tipo de item e erros do item (ok/não suportado).

Salvando filtro

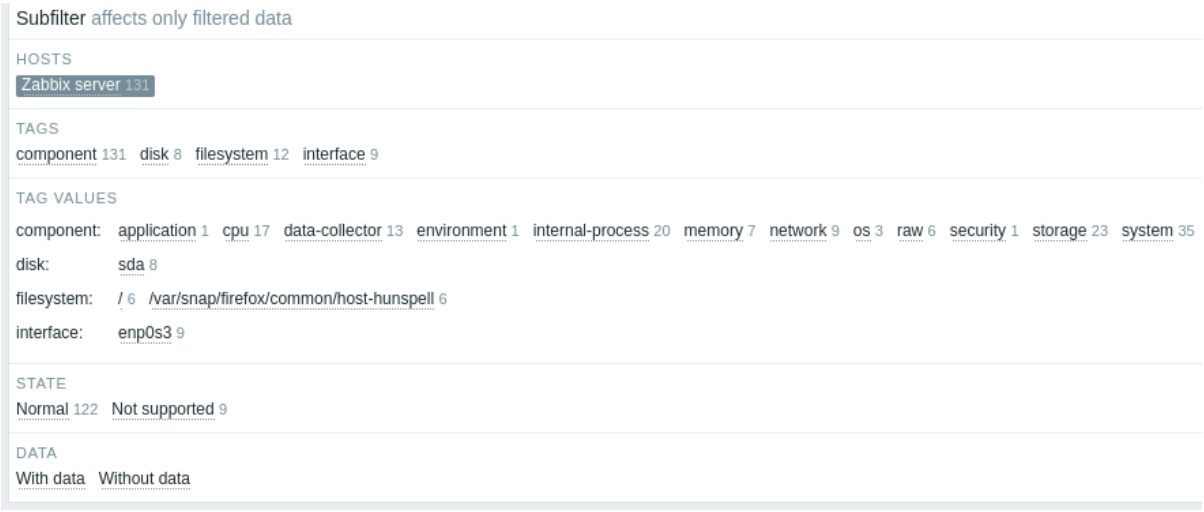
As configurações de filtro favoritas podem ser salvas como abas e, em seguida, acessadas rapidamente clicando na respectiva aba acima do filtro.

Veja mais detalhes sobre salvar filtros.

Usando o subfiltro

O subfiltro permite modificar ainda mais o filtro principal.

Ele contém links clicáveis para acesso rápido aos itens relacionados. Os itens são relacionados por uma entidade comum - host, nome ou valor da tag, estado do item ou status dos dados. Quando uma entidade é clicada, ela é destacada com um fundo cinza e os itens são filtrados imediatamente (não é necessário clicar em *Aplicar* no filtro principal). Clicar em outra entidade a adiciona aos resultados filtrados. Clicar novamente na entidade remove o filtro.



Os subfiltros são gerados com base nos dados filtrados, que são limitados a 1000 registros. Se houver 20 hosts, cada um com 100 itens (totalizando 2000 registros), apenas metade dos hosts será visível no subfiltro. Se você quiser ver mais registros no subfiltro, precisa aumentar o valor do parâmetro *Limite para resultados de pesquisa e filtro* (em *Administração -> Geral -> GUI*).

Ao contrário do filtro principal, o subfiltro é atualizado a cada atualização da tabela para sempre ter informações atualizadas sobre as opções de filtragem disponíveis e seus números de contagem.

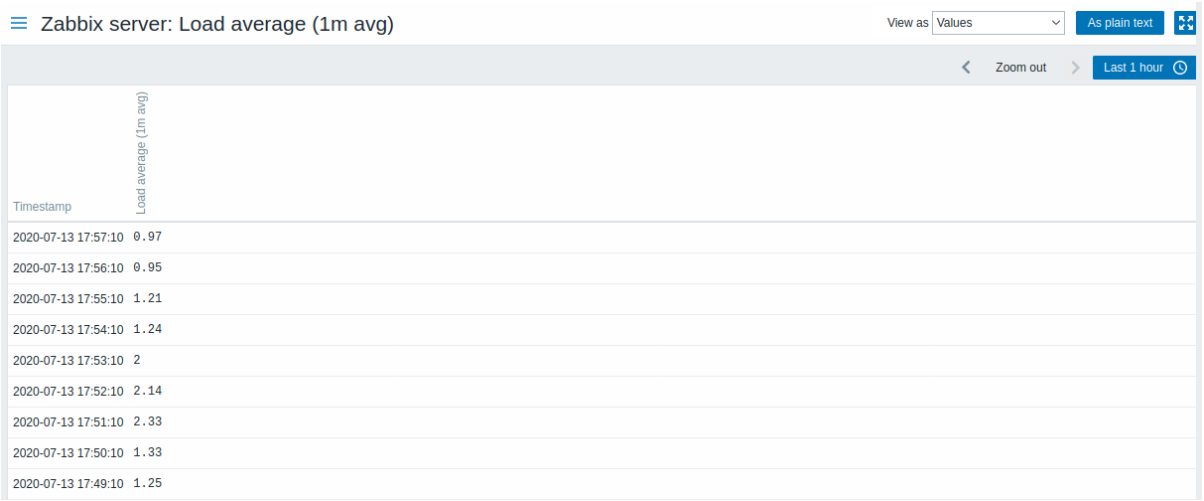
Para cada grupo de entidades (hosts, tags, valores de tags, etc.), até 10 linhas de entidades são exibidas. Se houver mais entidades, essa lista pode ser expandida para exibir no máximo 1000 entradas (o valor de SUBFILTER_VALUES_PER_GROUP em definições do frontend) clicando no ícone de três pontos ••• no final da lista. Para *Valores de tags*, a lista pode ser expandida para exibir no máximo 200 nomes de tags com seus respectivos valores. Observe que, uma vez totalmente expandida, a lista não pode ser recolhida.

Um número ao lado de cada entidade clicável indica o número de itens agrupados nela (com base nos resultados do filtro principal). Quando uma entidade é clicada, os números das outras entidades disponíveis são exibidos com um sinal de mais, indicando quantos itens podem ser adicionados à seleção atual. Entidades sem itens não são exibidas, a menos que tenham sido selecionadas anteriormente no subfiltro.

Gráficos e histórico

A coluna *Gráfico/Histórico* na lista de items oferece os seguintes links:

- **Histórico** - para todos os items textuais, levando a listagens (*Valores/500 últimos valores*) exibindo o histórico dos valores anteriores do item.
- **Gráfico** - para todos os items numéricos, levando a um gráfico simples. Observe que, quando o gráfico é exibido, um menu suspenso no canto superior direito oferece a possibilidade de alternar para *Valores/500 últimos valores* também.



Os valores exibidos nesta lista são brutos, ou seja, nenhum pós-processamento é aplicado.

Note:
A quantidade total de valores exibidos é definida pelo valor do parâmetro *Limite para resultados de pesquisa e filtro*, definido em *Administração → Geral → GUI*.

4 Mapas

Visão geral

Na seção *Monitoramento* → *Mapas* você pode configurar, gerenciar e visualizar mapas de rede.

Ao abrir esta seção, você verá o último mapa acessado ou uma lista de todos os mapas aos quais você tem acesso.

Todos os mapas podem ser públicos ou privados. Mapas públicos estão disponíveis para todos os usuários, enquanto mapas privados são acessíveis apenas ao seu proprietário e aos usuários com quem o mapa foi compartilhado.

Listagem de mapas

Dados exibidos:

Coluna	Descrição
Nome	Nome do mapa. Clique no nome para visualizar o mapa.
Largura	A largura do mapa é exibida.
Altura	A altura do mapa é exibida.
Ações	Duas ações estão disponíveis: **Propriedades** - define as propriedades gerais do mapa **Editar** - acessa a grade para adicionar elementos ao mapa

Para configurar um novo mapa, clique no botão *Criar mapa* no canto superior direito. Para importar um mapa de um arquivo YAML, XML ou JSON, clique no botão *Importar* no canto superior direito. O usuário que importar o mapa será definido como seu proprietário.

Dois botões abaixo da lista oferecem algumas opções de edição em massa:

- *Exportar* - exporta os mapas para um arquivo YAML, XML ou JSON
- *Excluir* - exclui os mapas

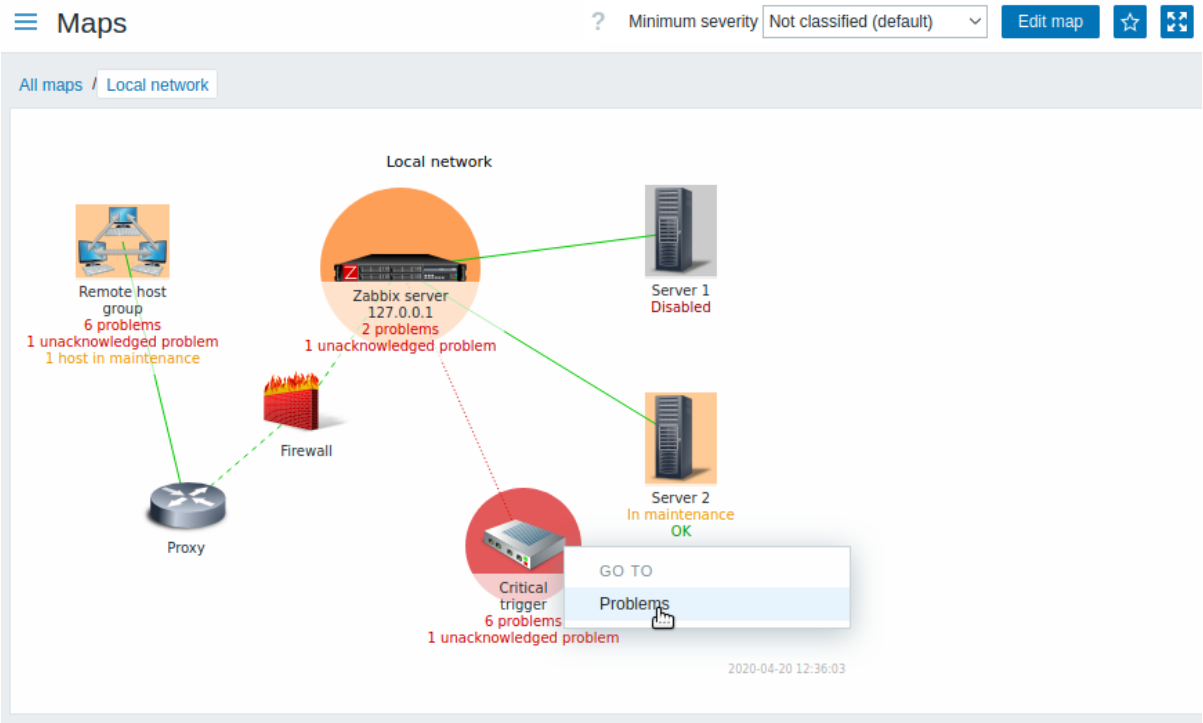
Para usar essas opções, marque as caixas de seleção antes dos respectivos mapas e clique no botão necessário.

Usando o filtro

Você pode usar o filtro para exibir apenas os mapas de seu interesse. Para melhor desempenho de pesquisa, os dados são pesquisados com macros não resolvidas.

Visualizando mapas

Para visualizar um mapa, clique em seu nome na lista de todos os mapas.



Você pode usar o menu suspenso na barra de título do mapa para selecionar o menor nível de gravidade dos triggers de problema a serem exibidos. A gravidade marcada como *padrão* é o nível definido na configuração do mapa. Se o mapa contiver um submapa, navegar até o submapa manterá a gravidade do mapa de nível superior (exceto se for *Não classificado*, nesse caso, não será passada para o submapa).

Realce de ícone

Se um elemento do mapa estiver em status de problema, ele será destacado com um círculo redondo. A cor de preenchimento do círculo corresponde à cor de gravidade do problema. Somente problemas no nível de gravidade selecionado ou acima dele serão exibidos com o elemento. Se todos os problemas forem reconhecidos, uma borda verde grossa ao redor do círculo será exibida.

Adicionalmente:

- um host em manutenção é destacado com um quadrado laranja preenchido. Observe que o destaque de manutenção tem prioridade sobre o destaque de gravidade do problema, se o elemento do mapa for um host.
- um host desabilitado (não monitorado) é destacado com um quadrado cinza preenchido.

O destaque é exibido se a caixa de seleção *Realce de ícone* estiver marcada na configuração do mapa.

Marcadores de alteração recente

Triângulos vermelhos apontando para dentro ao redor de um elemento indicam uma alteração recente no status do trigger - uma alteração que ocorreu nos últimos 30 minutos. Esses triângulos são exibidos se a caixa de seleção *Marcar elementos na alteração do status do trigger* estiver marcada na configuração do mapa.

Links

Clicar em um elemento do mapa abre um menu com alguns links disponíveis. Clicar no nome do host exibe o menu do host.

Botões

Os botões à direita oferecem as seguintes opções:

Edit map	Ir para a edição do conteúdo do mapa.
	Adicionar o mapa ao widget de favoritos em Dashboards.
	O mapa está no widget de favoritos em Dashboards. Clique para remover o mapa do widget de favoritos.

Os botões do modo de visualização, comuns a todas as seções, estão descritos na página Monitoramento.

Resumo legível em mapas

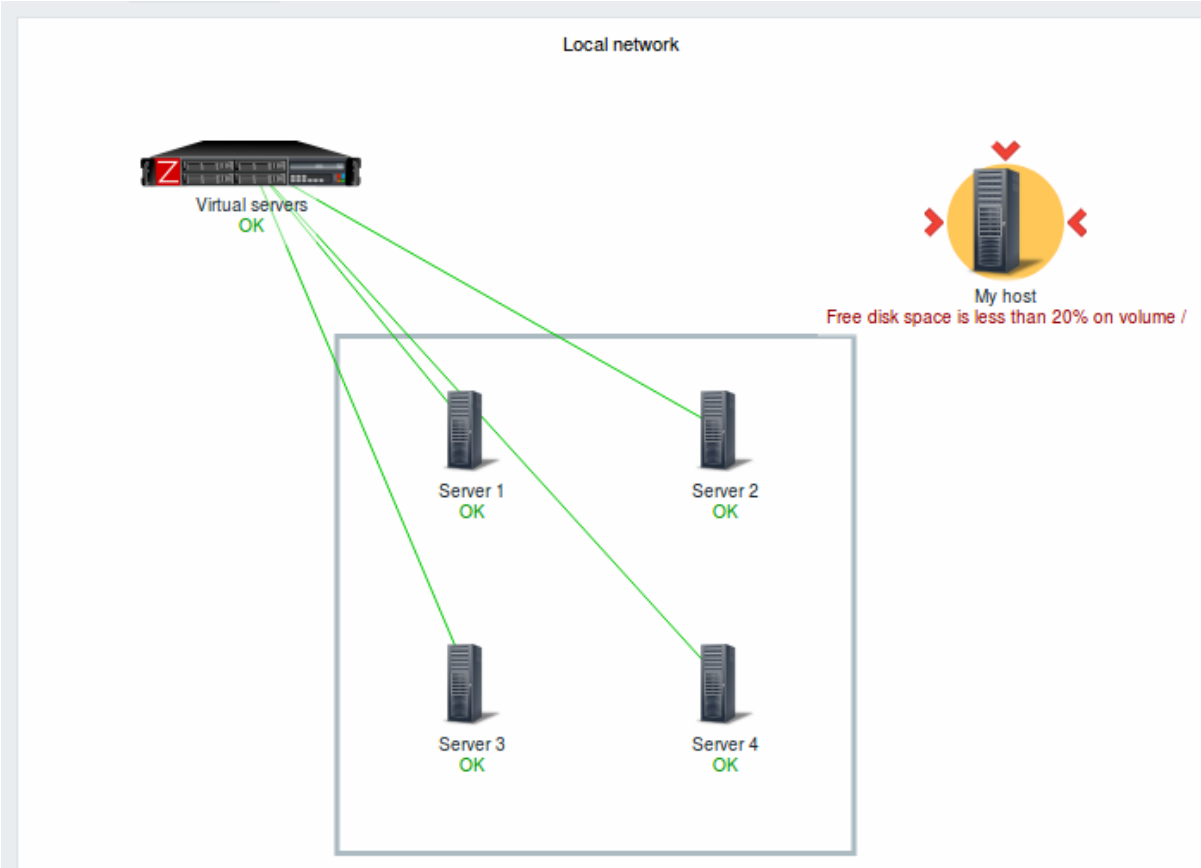
Uma propriedade oculta "aria-label" está disponível permitindo que as informações do mapa sejam lidas por um leitor de tela. Tanto a descrição geral do mapa quanto a descrição individual dos elementos estão disponíveis, no seguinte formato:

- para descrição do mapa: `<Nome do mapa>, <* de * itens em estado de problema>, <* problemas no total>.`
- para descrever um elemento com um problema: `<Tipo de elemento>, Status <Status do elemento>, <Nome do elemento>, <Descrição do problema>.`
- para descrever um elemento com múltiplos problemas: `<Tipo de elemento>, Status <Status do elemento>, <* problemas>.`
- para descrever um elemento sem problemas: `<Tipo de elemento>, Status <Status do elemento>, <Nome do elemento>.`

Por exemplo, esta descrição está disponível:

```
'Rede local, 1 de 6 elementos em estado de problema, 1 problema no total. Host, Status problema, Meu host,
```

para o seguinte mapa:



Referenciando um mapa de rede

Os mapas de rede podem ser referenciados tanto pelos parâmetros GET `sysmapid` quanto `mapname`. Por exemplo,

```
http://zabbix/zabbix/zabbix.php?action=map.view&mapname=Local%20network
```

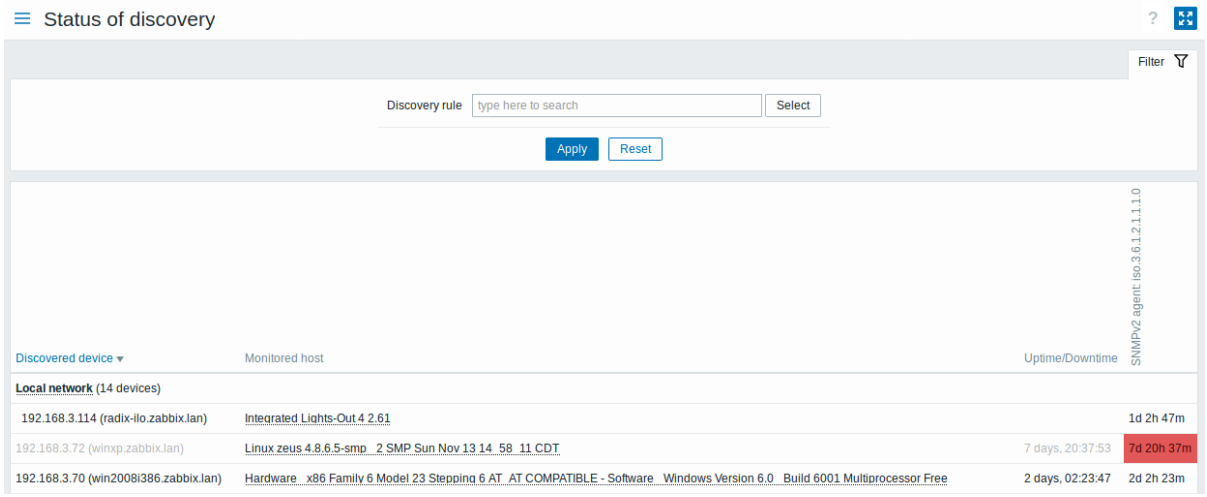
abrirá o mapa com esse nome (Local network).

Se ambos `sysmapid` (ID do mapa) e `mapname` (nome do mapa) forem especificados, `mapname` terá prioridade.

5 Descoberta

Visão geral

Na seção *Monitoramento* → *Descoberta*, são exibidos os resultados da descoberta de rede. Os dispositivos descobertos são classificados pela regra de descoberta.



Dados exibidos:

Coluna	Descrição
Dispositivo descoberto	Os dispositivos descobertos são listados, agrupados pela regra de descoberta. Clicar na regra de descoberta exibe o menu da regra contendo o link para o formulário de configuração da regra de descoberta.
Host monitorado	Se um dispositivo já estiver sendo monitorado, o nome do host será listado nesta coluna. Clicar no nome do host exibe o menu do host.
Tempo de atividade/inatividade	A duração do dispositivo sendo descoberto ou perdido após a descoberta anterior é exibida nesta coluna.
Verificação de descoberta	O estado do serviço individual (verificação de descoberta) para cada dispositivo descoberto é exibido. Uma célula vermelha indica que o serviço está inativo. O tempo de atividade ou inatividade do serviço é incluído dentro da célula. Esta coluna é exibida apenas se o serviço foi encontrado em pelo menos um dispositivo descoberto.

Botões

Os botões do modo de visualização, comuns a todas as seções, estão descritos na Monitoramento página.

Usando filtro

Você pode usar o filtro para exibir apenas as regras de descoberta que você está interesse. Para um melhor desempenho de pesquisa, os dados são pesquisados com macros não resolvidas.

Sem nada selecionado no filtro, todas as regras de descoberta habilitadas são exibido. Para selecionar uma regra de descoberta específica para exibição, comece a digitar seu nome no filtro. Todas as regras de descoberta habilitadas correspondentes serão listados para seleção. Mais de uma regra de descoberta pode ser selecionada.

3 Serviços

Visão Geral¶

¶ O menu Serviços é usado para as funções de service monitoring do Zabbix.

1 Serviços

Visão geral

Nesta seção, você pode ver um status de alto nível de todos os serviços que foram configurados no Zabbix, com base na sua infraestrutura.

Um serviço pode ser uma hierarquia composta por vários níveis de outros serviços, chamados de serviços ”filhos”, que são atributos para o status geral do serviço (veja também uma visão geral da funcionalidade de monitoramento de serviços).

As principais categorias de status do serviço são *OK* ou *Problema*, onde o status *Problema* é expresso pelo nome e cor de gravidade do problema correspondente.

Enquanto o modo de visualização permite monitorar os serviços com seu status e outros detalhes, você também pode configurar a hierarquia de serviços nesta seção (adicionar/editar serviços, serviços filhos) alternando para o modo de edição.

Para alternar do modo de visualização para o modo de edição (e vice-versa), clique no respectivo botão no canto superior direito:

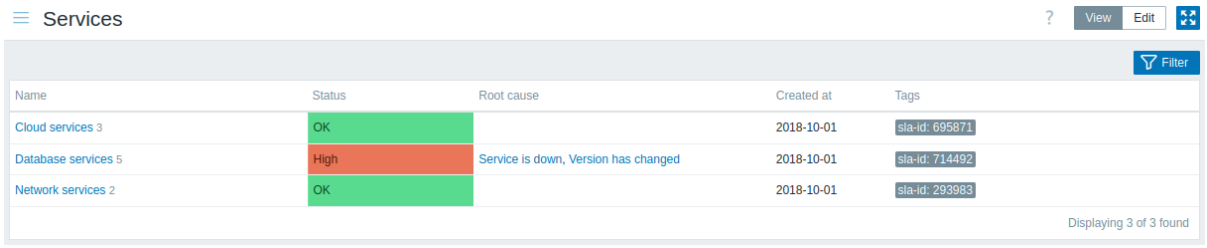
-

- visualizar serviços
-

- adicionar/editar serviços e serviços filhos

Observe que o acesso à edição depende das configurações de função do usuário.

Visualizando serviços



Uma lista dos serviços existentes é exibida.

Dados exibidos:

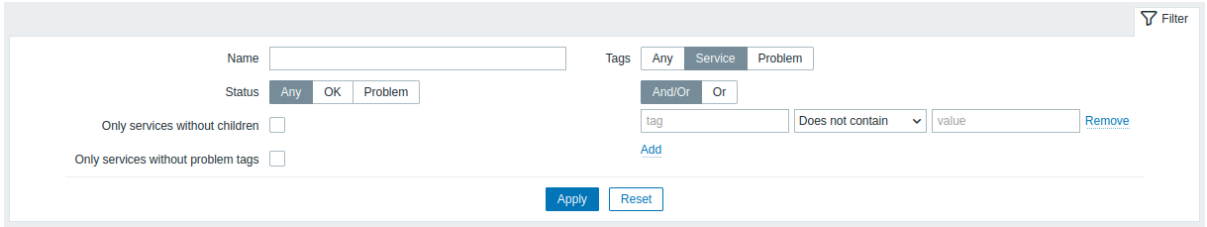
Parâmetro	Descrição
Nome	Nome do serviço. O nome do serviço é um link para os detalhes do serviço. O número após o nome indica quantos serviços filhos o serviço possui.
Status	Status do serviço: **OK** - sem problemas **<cor e severidade do problema>** - indica um problema e sua severidade. Em caso de múltiplos problemas, a cor e a severidade do problema mais crítico são exibidas.
Causa raiz	Problemas subjacentes que afetam direta ou indiretamente o status do serviço são listados. Os mesmos problemas são listados conforme retornado pela macro {SERVICE.ROOTCAUSE} macro. Clique no nome do problema para ver mais detalhes sobre ele em *Monitoramento* → *Problemas*. Problemas que não afetam o status do serviço não estão na lista.
Criado em	O horário em que o serviço foi criado é exibido.
Tags	As tags do serviço são exibidas. As tags são usadas para identificar um serviço em ações de serviço e SLAs.

Botões

Os botões do modo de visualização, comuns a todas as seções, estão descritos na página Monitoramento.

Usando o filtro

Você pode usar o filtro para exibir apenas os serviços do seu interesse.



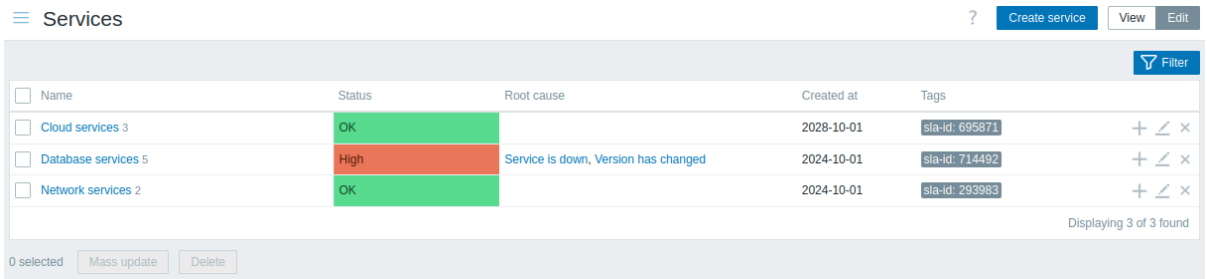
Parâmetro	Descrição
Nome	Filtrar pelo nome do serviço.
Status	Filtrar pelo status do serviço.
Apenas serviços sem filhos	Marque a caixa de seleção para exibir apenas serviços sem serviços filhos. Este parâmetro está disponível apenas no modo de edição de serviços.

Parâmetro	Descrição
Apenas serviços sem tags de problema	Marque a caixa de seleção para exibir apenas serviços sem tags de problema. Este parâmetro está disponível apenas no modo de edição de serviços.
Tags	Filtrar pelo nome e valor da tag do serviço, ou pelo nome e valor da tag de problema do serviço (no modo de edição de serviços). É possível incluir ou excluir tags e valores de tags específicos. Várias condições podem ser definidas. A correspondência do nome da tag diferencia maiúsculas de minúsculas. Existem vários operadores disponíveis para cada condição: **Existe** - inclui os nomes de tags especificados; **Igual** - inclui os nomes e valores de tags especificados (diferencia maiúsculas de minúsculas); **Contém** - inclui os nomes de tags especificados onde os valores das tags contêm a string inserida (correspondência de substring, não diferencia maiúsculas de minúsculas); **Não existe** - exclui os nomes de tags especificados; **Diferente** - exclui os nomes e valores de tags especificados (diferencia maiúsculas de minúsculas); **Não contém** - exclui os nomes de tags especificados onde os valores das tags contêm a string inserida (correspondência de substring, não diferencia maiúsculas de minúsculas). Existem dois tipos de cálculo para as condições: **E/Ou** - todas as condições devem ser atendidas, condições com o mesmo nome de tag serão agrupadas pela condição Ou; **Ou** - basta que uma condição seja atendida.

Editando serviços

Clique no botão *Editar* para acessar o modo de edição. Quando estiver no modo de edição, a listagem é complementada com caixas de seleção antes das entradas e também estas opções adicionais:

- - adicionar um serviço filho a este serviço
- - editar este serviço
- - excluir este serviço

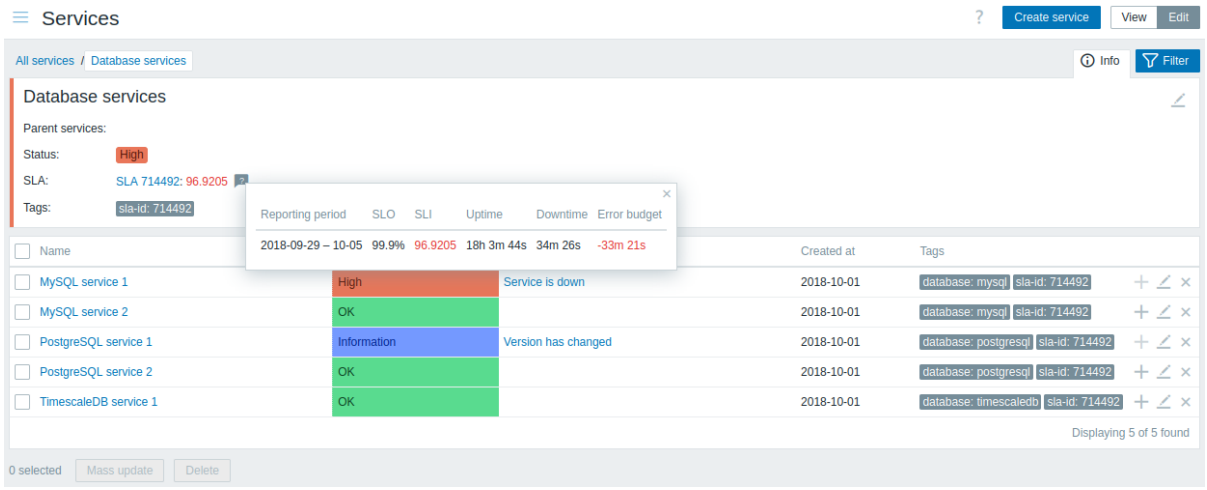


Para configurar um novo serviço, clique no botão *Criar serviço* no canto superior direito.

Detalhes do serviço

Para acessar os detalhes do serviço, clique no nome do serviço. Para retornar à lista de todos os serviços, clique em *Todos os serviços*.

Os detalhes do serviço incluem a caixa de informações e a lista de serviços filhos.



Para acessar a caixa de informações, clique na guia *Info*. A caixa de informações contém as seguintes entradas:

- Nomes dos serviços pais (se houver)
- Status atual deste serviço
- SLA(s) atual(is) deste serviço, no formato `Nome do SLA:indicador de nível de serviço`. 'Nome do SLA' também é um link para o relatório de SLA deste serviço. Se você posicionar o mouse na caixa de informações ao lado do indicador de nível de serviço (SLI), uma lista de informações pop-up será exibida com detalhes do SLI. O indicador de nível de serviço exibe o nível de serviço atual, em porcentagem.
- Tags do serviço

A caixa de informações também contém um link para a configuração do serviço.

Para usar o filtro para serviços filhos, clique na guia *Filtro*.

Quando estiver no modo de edição, a listagem de serviços filhos é complementada com opções adicionais de edição:

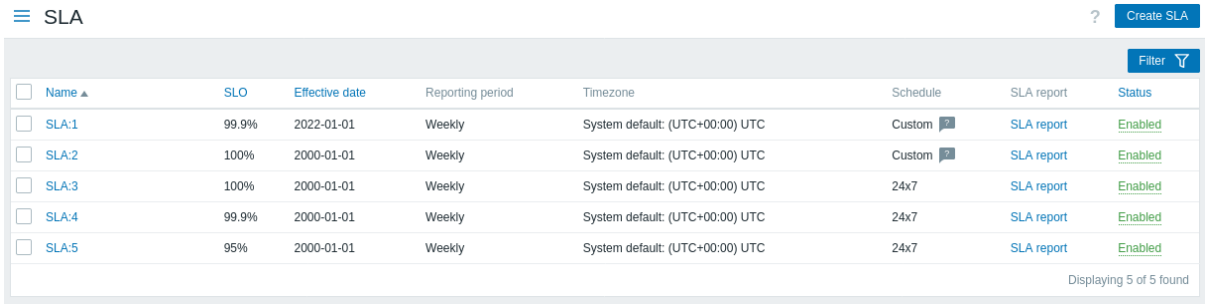
- - adicionar um serviço filho a este serviço
- - editar este serviço
- - excluir este serviço

2 SLA

Visão geral

Esta seção permite visualizar e configure SLAs.

SLAs



Uma lista dos SLAs configurados é exibida. *Observe* que apenas os SLAs relacionados aos serviços acessíveis ao usuário serão exibidos (apenas para leitura, a menos que *Gerenciar SLA* esteja habilitado para a função do usuário).

Dados exibidos:

Parâmetro	Descrição
Nome	O nome do SLA é exibido. O nome é um link para a configuração do SLA.
SLO	O objetivo do nível de serviço (SLO) é exibido.
Data efetiva	A data de início do cálculo do SLA é exibida.

Parâmetro	Descrição
Período de relatório	O período usado no relatório do SLA é exibido - *diário*, *semanal*, *mensal*, *trimestral* ou *anual*.
Fuso horário	O fuso horário do SLA é exibido.
Agenda	A agenda do SLA é exibida - 24x7 ou personalizada.
Relatório de SLA	Clique no link para ver o relatório de SLA para este SLA.
Status	O status do SLA é exibido - habilitado ou desabilitado.

3 Relatório de SLA

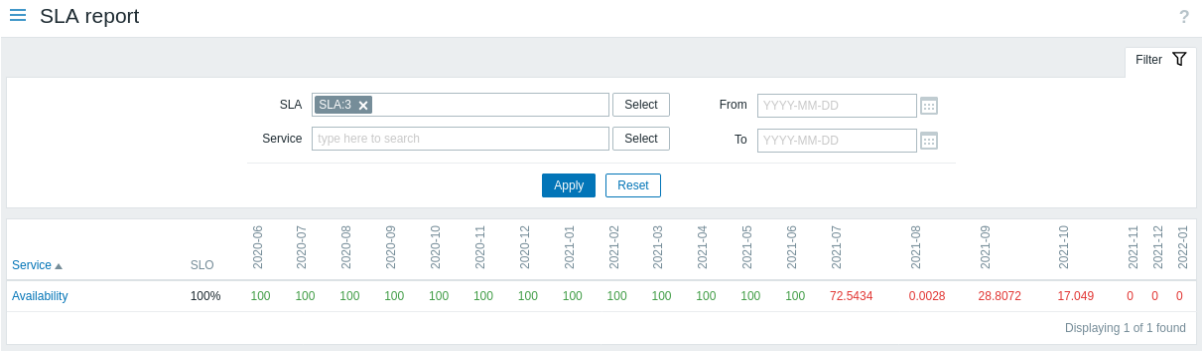
Visão geral

Esta seção permite visualizar relatórios de SLA, com base nos critérios selecionados no filtro.

Os relatórios de SLA também podem ser exibidos como um widget de dashboard.

Relatório

O filtro permite selecionar o relatório com base no nome do SLA, bem como no Nome do Serviço. Também é possível limitar o período exibido.

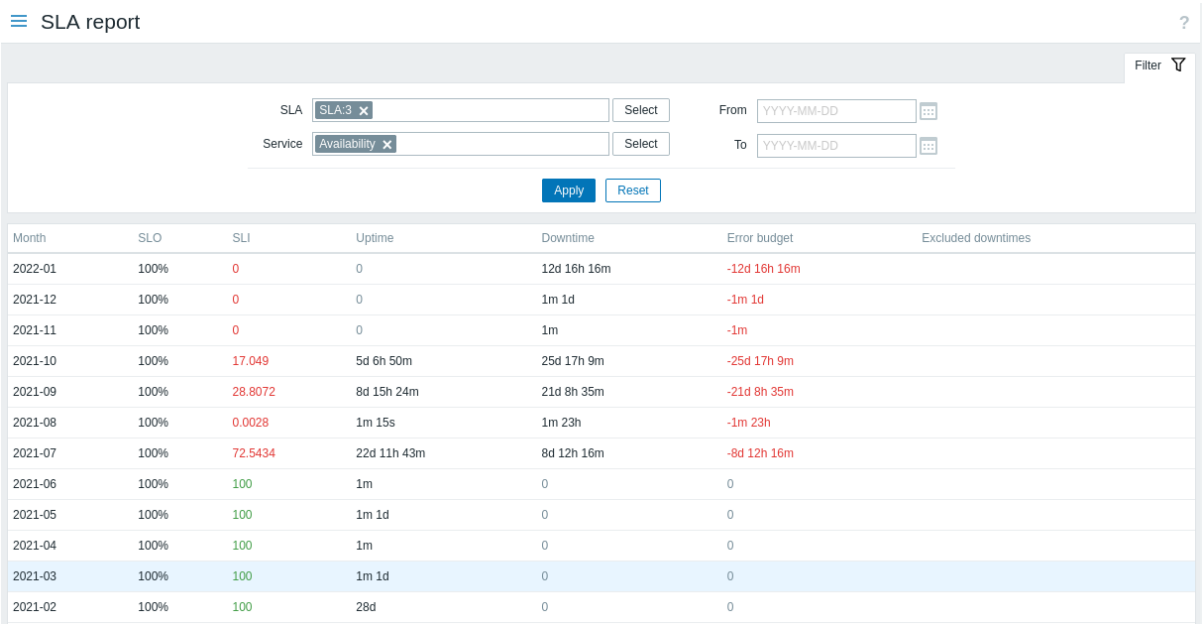


Cada coluna (período) exibe o SLI para esse período. SLIs que estão em violação do conjunto SLO são destacados em vermelho.

20 períodos são exibidos no relatório. Um máximo de 100 períodos podem ser exibidos, se tanto as opções de data *De* e *Até* forem especificadas.

Detalhes do relatório

Se você clicar no nome do serviço no relatório, poderá acessar outro relatório que exibe uma visão mais detalhada.



Observe que a duração negativa do problema não afeta o cálculo ou o relatório de SLA.

4 Inventário

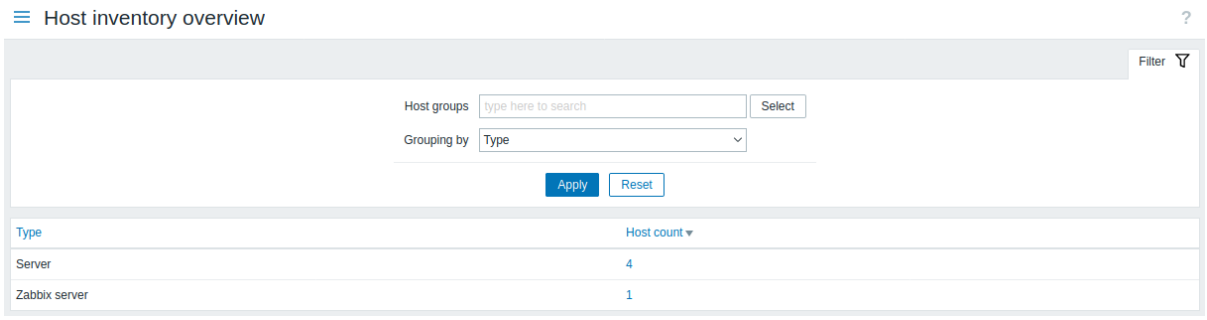
Visão geral

O menu Inventário apresenta seções que fornecem uma visão geral dos dados de inventário do host por um parâmetro escolhido, bem como a capacidade de visualizar detalhes do inventário do host.

1 Visão geral

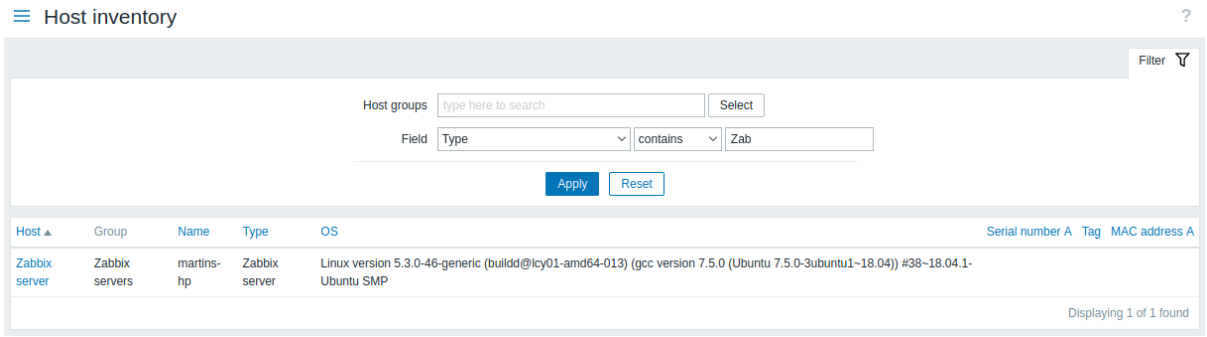
Através do módulo *Inventário -> Visão geral* você pode visualizar uma visão geral do inventário de hosts.

Para que a visão geral seja apresentada, selecione um grupo de host (ou todos) e o campo do inventário que irá agrupar os dados. Será apresentada a quantidade de hosts correspondentes a valor localizado naquele campo do inventário.



A completude de uma visão geral depende de quanta informação estiver disponível no inventário dos hosts.

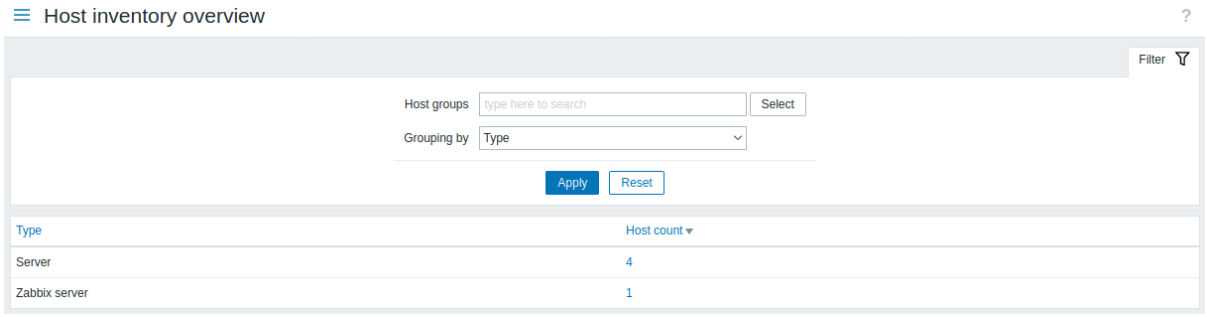
Os números na coluna *Contagem de hosts* possuem links; eles levam à um relatório detalhado com os hosts que foram contabilizados na visão geral.



Visão geral

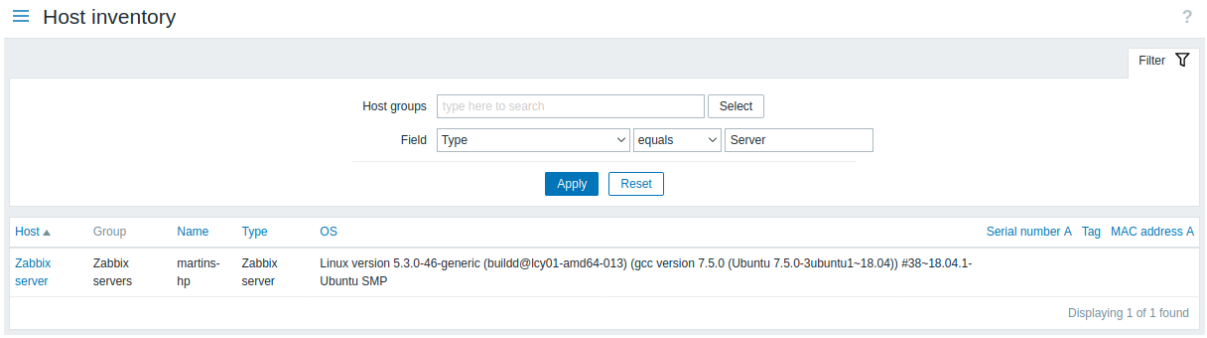
A seção *Inventário → Visão geral* fornece maneiras de ter uma visão geral de dados de inventário de host.

Para que uma visão geral seja exibida, selecione grupos de hosts (ou nenhum) e o campo de inventário pelo qual exibir dados. O número de hospedeiros correspondente a cada entrada do campo escolhido será exibido.



A integridade de uma visão geral depende de quanto estoque informações são mantidas com os anfitriões.

Os números na coluna *Host count* são links; eles levam a esses hospedeiros sendo filtrado na tabela *Host Inventories*.

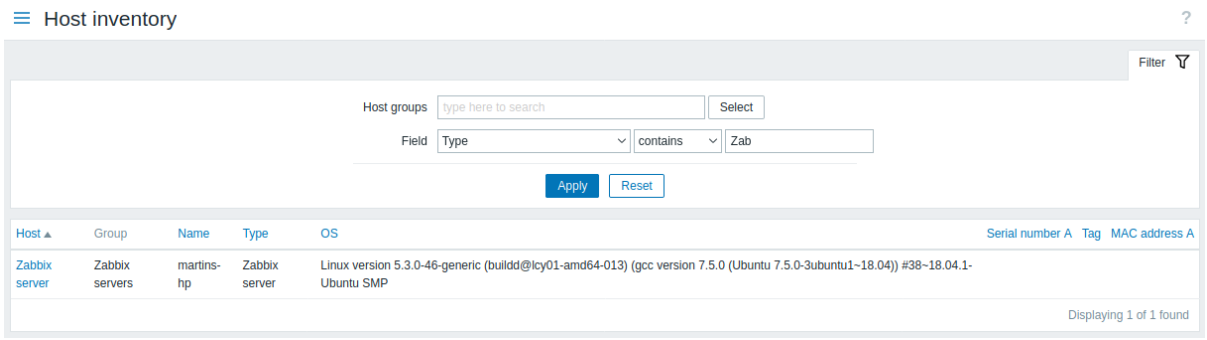


2 Hosts

Visão geral

Na seção *Inventário* → *Hosts*, os dados de inventário dos hosts são exibidos.

Você pode filtrar os hosts por grupo(s) de hosts e por qualquer campo de inventário para exibir apenas os hosts do seu interesse.

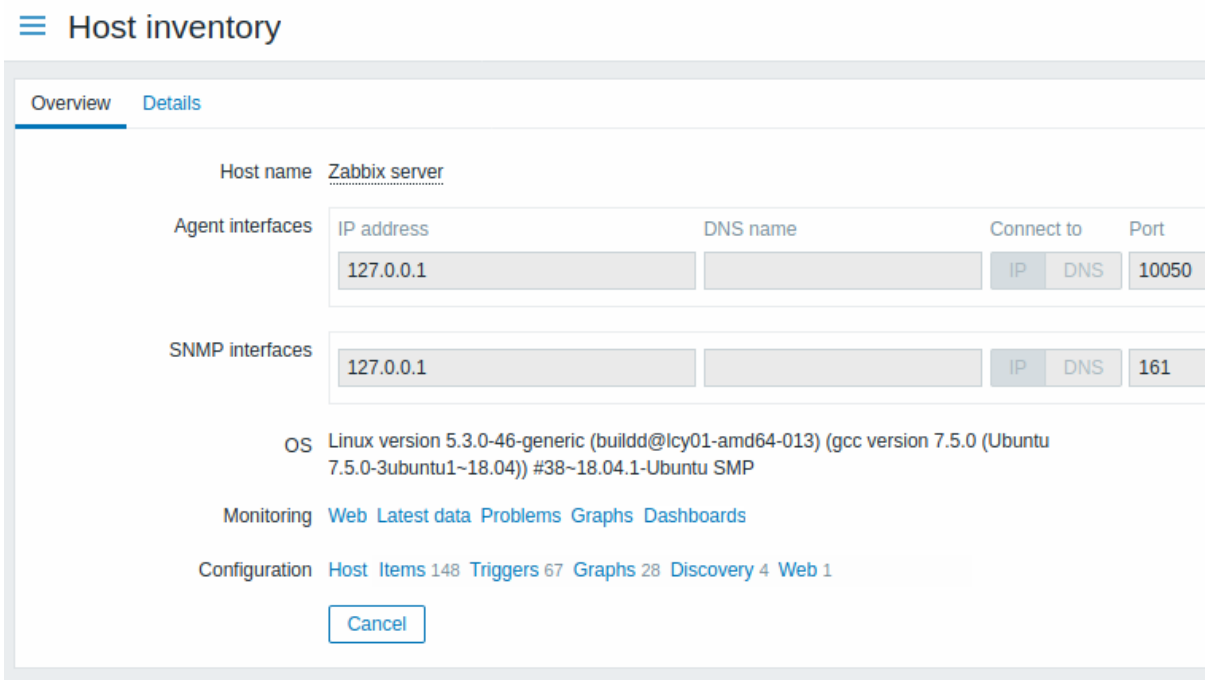


Para exibir todos os inventários de hosts, pressione o botão ”Redefinir”.

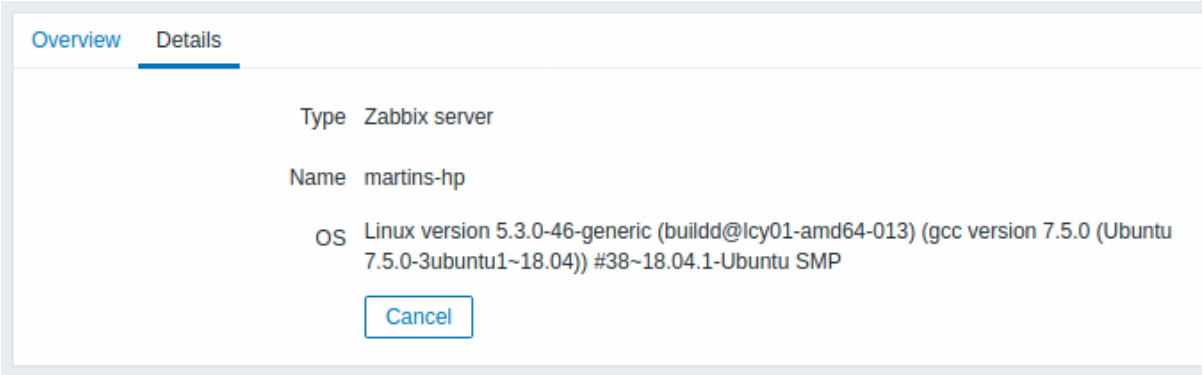
Embora apenas alguns campos-chave de inventário sejam exibidos na tabela, você também pode visualizar todas as informações de inventário disponíveis para esse host. Para isso, clique no nome do host na primeira coluna.

Detalhes do inventário

A aba **Visão geral** contém algumas informações gerais sobre o host, os dados de monitoramento mais recentes e as opções de configuração do host:



A aba **Detalhes** contém todos os detalhes de inventário disponíveis para o host:



A completude dos dados de inventário depende de quanta informação de inventário é mantida com o host. Se nenhuma informação for mantida, a aba *Detalhes* é desabilitada.

5 Relatórios

Visão geral

O menu Relatórios apresenta várias seções que contêm uma variedade de relatórios predefinidos e personalizáveis pelo usuário, focados em exibir uma visão geral de parâmetros como informações do sistema, gatilhos (triggers) e dados coletados.

1 Informações do sistema

Visão geral

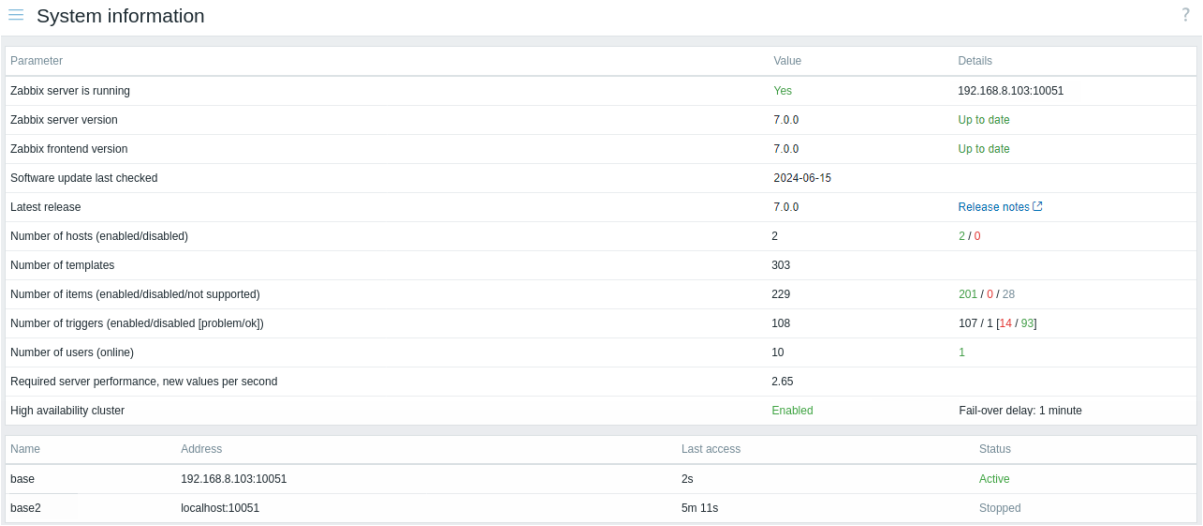
Em *Relatórios* → *Informações do sistema*, é exibido um resumo dos principais dados do server e do sistema Zabbix. Os dados do sistema são coletados usando items internos.

Observe que, em uma configuração de alta disponibilidade, é possível redirecionar a fonte das informações do sistema (instância do server). Para isso, edite o arquivo *zabbix.conf.php* - descomente e defina `$ZBX_SERVER` ou ambos `$ZBX_SERVER` e `$ZBX_SERVER_PORT` para um server diferente daquele exibido como ativo. Observe que, ao definir apenas `$ZBX_SERVER`, um valor padrão (10051) para `$ZBX_SERVER_PORT` será usado.

Com a configuração de alta disponibilidade ativada, um bloco separado é exibido abaixo das estatísticas do sistema com detalhes dos nós de alta disponibilidade. Este bloco é visível apenas para usuários *Super Admin* do Zabbix.

Informações do sistema também estão disponíveis como um widget do dashboard.

Estatísticas do sistema



Dados exibidos:

Parâmetro	Valor	Detalhes
Zabbix server is running	Status do Zabbix server: **Yes** - server está em execução **No** - server não está em execução *Nota:* Para exibir o restante das informações, o frontend precisa que o server esteja em execução e que pelo menos um processo trapper esteja iniciado no server (parâmetro StartTrappers no arquivo zabbix_server.conf > 0).	Localização e porta do Zabbix server.
Zabbix server version	O número da versão atual do server é exibido. *Nota:* Só é exibido quando o Zabbix server está em execução.	O status da versão do server é exibido: **Up to date** - usando a versão mais recente; **New update available** - uma versão mais atual está disponível; **Outdated** - o período de suporte completo para esta versão expirou. Esta informação só está disponível se a verificação de atualização de software estiver habilitada na configuração do Zabbix server. Nada é exibido se a última verificação de atualização de software foi realizada há mais de uma semana ou não existem dados sobre a versão atual.
Zabbix frontend version	O número da versão do Zabbix frontend é exibido.	O status da versão do Zabbix frontend é exibido: **Up to date** - usando a versão mais recente; **New update available** - uma versão mais atual está disponível; **Outdated** - o período de suporte completo para esta versão expirou. Esta informação só está disponível se a verificação de atualização de software estiver habilitada na configuração do Zabbix server. Nada é exibido se a última verificação de atualização de software foi realizada há mais de uma semana ou não existem dados sobre a versão atual.
Software update last checked	A data da última verificação de atualização de software do Zabbix é exibida. Esta informação só está disponível se a verificação de atualização de software estiver habilitada na configuração do Zabbix server.	
Latest release	O número de uma versão mais recente (se disponível) para a versão atual do Zabbix é exibido. Esta informação só está disponível se a verificação de atualização de software estiver habilitada na configuração do Zabbix server. Nada é exibido se a última verificação de atualização de software foi realizada há mais de uma semana ou não existem dados sobre a versão atual.	Um link para as notas de lançamento da versão mais recente do Zabbix disponível é exibido.
Number of hosts	O número total de hosts configurados é exibido.	Número de hosts monitorados/hosts não monitorados.
Number of templates	O número total de templates é exibido.	
Number of items	O número total de items é exibido.	Número de items de host monitorados/desabilitados/não suportados. Items em hosts desabilitados são contados como desabilitados.

Parâmetro	Valor	Detalhes
Number of triggers	O número total de triggers é exibido.	Número de triggers de host habilitadas/desabilitadas; divisão das triggers habilitadas de acordo com os estados ”Problem”/”OK”. Triggers listadas sob o estado ”OK” incluem triggers com o status ”Unknown”. Triggers dependentes de items desabilitados ou atribuídas a hosts desabilitados são contadas como desabilitadas.
Number of users	O número total de usuários configurados é exibido.	Número de usuários online.
Required server performance, new values per second	O número esperado de novos valores processados pelo Zabbix server por segundo é exibido.	*Required server performance* é uma estimativa e pode ser útil como diretriz. Para números precisos de valores processados, use o item interno `zabbix[wcache,values,all]`. Items habilitados de hosts monitorados são incluídos no cálculo. Items de log são contados como um valor por intervalo de atualização do item. Valores de intervalo regular são contados; valores de intervalo flexível e de agendamento não são. O cálculo não é ajustado durante um período de manutenção ”nodata”. Items trapper não são contados.
Global scripts on Zabbix server	**Disabled** será exibido neste campo se os scripts globais estiverem desabilitados no Zabbix server, configurando EnableGlobalScripts=0 na configuração do server.	
High availability cluster	Status do cluster de alta disponibilidade para o Zabbix server: **Disabled** - server standalone **Enabled** - pelo menos um nó de alta disponibilidade existe	Se habilitado, o atraso de failover é exibido.

System information também exibirá uma mensagem de erro nas seguintes condições:

- O banco de dados utilizado não possui o conjunto de caracteres ou collation necessários (UTF-8).
- A versão do banco de dados está abaixo ou acima do intervalo suportado (disponível apenas para usuários com o tipo *Super admin role*).
- O housekeeping para o TimescaleDB está configurado incorretamente (as tabelas de histórico ou tendência contêm chunks comprimidos, mas as opções *Override item history period* ou *Override item trend period* estão desabilitadas).

Nós de alta disponibilidade

Se o cluster de alta disponibilidade estiver habilitado, outro bloco de dados será exibido com o status de cada nó de alta disponibilidade.

Name	Address	Last access	Status
node-active	192.168.1.13:10051	12s	Active
node6	192.168.1.10:10053	1h 2m 40s	Unavailable
node7	192.168.1.11:10053	3m 40s	Unavailable
node4	192.168.1.8:10052	1h 34m 29s	Stopped
node5	192.168.1.9:10053	1h 9m 51s	Stopped
node8	192.168.1.12:10051	21m 16s	Stopped
node1	192.168.1.5:10051	17s	Standby
node2	192.168.1.6:10051	16s	Standby
node3	192.168.1.7:10052	16 2021-10-20 17:58:47	Standby

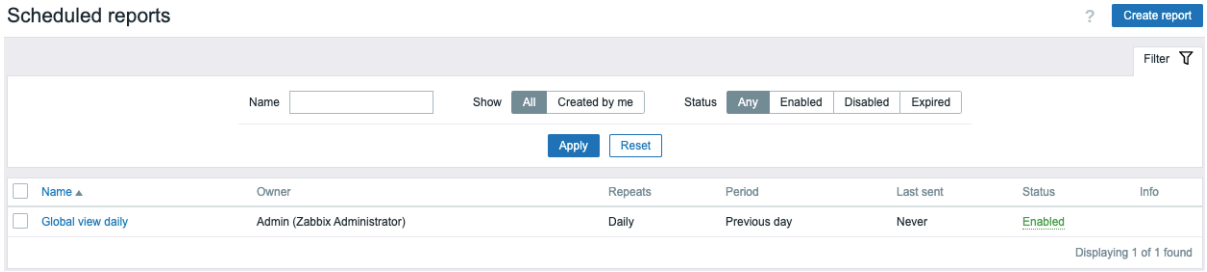
Dados exibidos:

Coluna	Descrição
Nome	Nome do nó, conforme definido na configuração do server.
Endereço	Endereço IP e porta do nó.
Último acesso	Hora do último acesso do nó. Ao passar o mouse sobre a célula, é exibido o timestamp do último acesso em formato longo.
Status	Status do nó. As linhas da tabela são classificadas por esses status nesta ordem de prioridade: **Ativo** - o nó está ativo e funcionando **Indisponível** - o nó não foi visto por mais tempo do que o atraso de failover (você pode querer descobrir o motivo) **Parado** - o nó foi parado ou não pôde ser iniciado (você pode querer iniciá-lo ou excluí-lo) **Standby** - o nó está ativo e aguardando

2 Relatórios agendados

Visão geral

Em *Relatórios* → *Relatórios agendados*, usuários com permissões suficientes podem configurar a geração agendada de versões em PDF dos painéis, que serão enviadas por e-mail para os destinatários especificados.



A tela de abertura exibe informações sobre os relatórios agendados, que podem ser filtradas para facilitar a navegação - veja a seção Usando o filtro abaixo.

Dados exibidos:

Coluna	Descrição
Nome	Nome do relatório. Clicar nele abre o formulário de configuração do relatório.
Proprietário	Usuário que criou o relatório.
Repetições	Frequência de geração do relatório (diária/semanal/mensal/anual).
Período	Período de tempo para o qual o relatório é preparado (dia, semana, mês ou ano anterior).
Último envio	A data e hora em que o relatório mais recente foi enviado.
Status	Status atual do relatório (habilitado/desabilitado/expirado). Usuários com permissões suficientes podem alterar o status clicando nele - de ”Habilitado” para ”Desabilitado” (e vice-versa); de ”Expirado” para ”Desabilitado” (e vice-versa). Para usuários com direitos insuficientes, o status não é clicável.
Info	Exibe ícones informativos: Um ícone vermelho indica que a geração do relatório falhou; ao passar o mouse sobre ele, será exibida uma dica de ferramenta com as informações do erro. Um ícone amarelo indica que um relatório foi gerado, mas o envio para alguns (ou todos) os destinatários falhou ou que um relatório está expirado; ao passar o mouse sobre ele, será exibida uma dica de ferramenta com informações adicionais.

Usando o filtro

Você pode usar o filtro para restringir a lista de relatórios. Para melhor desempenho na pesquisa, os dados são pesquisados com macros não resolvidas.

As seguintes opções de filtragem estão disponíveis:

- *Nome* - é permitido correspondência parcial do nome
- *Mostrar* - criados pelo usuário atual ou todos os relatórios
- *Status* - selecione entre ”Qualquer” (mostrar todos os relatórios), ”Habilitado”, ”Desabilitado” ou ”Expirado”

O filtro está localizado abaixo do nome da seção *Relatórios agendados*. Ele pode ser aberto e recolhido clicando na guia *Filtro* no canto superior direito.

Atualização em massa

Às vezes, você pode querer excluir ou alterar o status de vários relatórios de uma só vez. Em vez de abrir cada relatório individualmente para edição, você pode usar a função de atualização em massa para isso.

Para atualizar vários relatórios de uma vez, faça o seguinte:

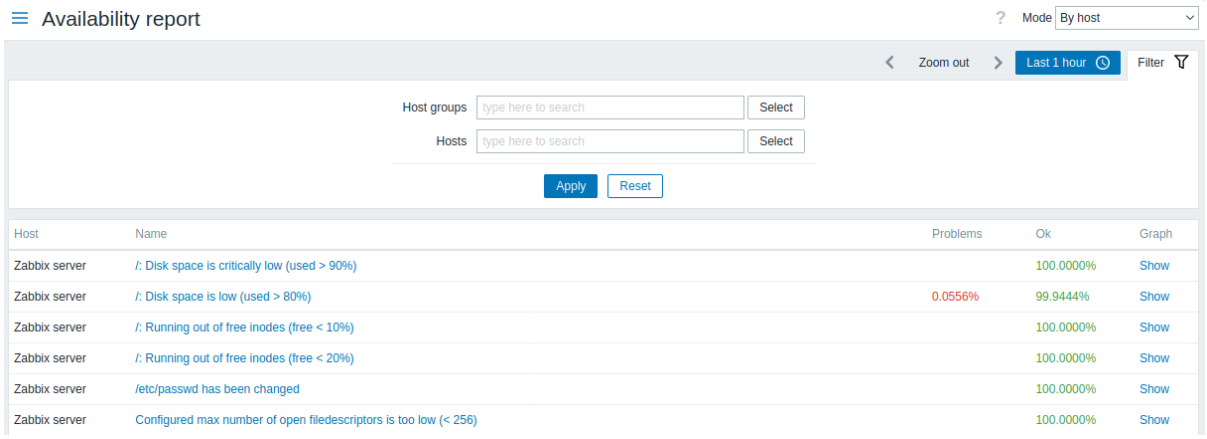
- Marque as caixas de seleção dos relatórios a serem atualizados na lista
- Clique no botão necessário abaixo da lista para fazer as alterações (*Ativar*, *Desativar* ou *Excluir*)

3 Relatório de disponibilidade

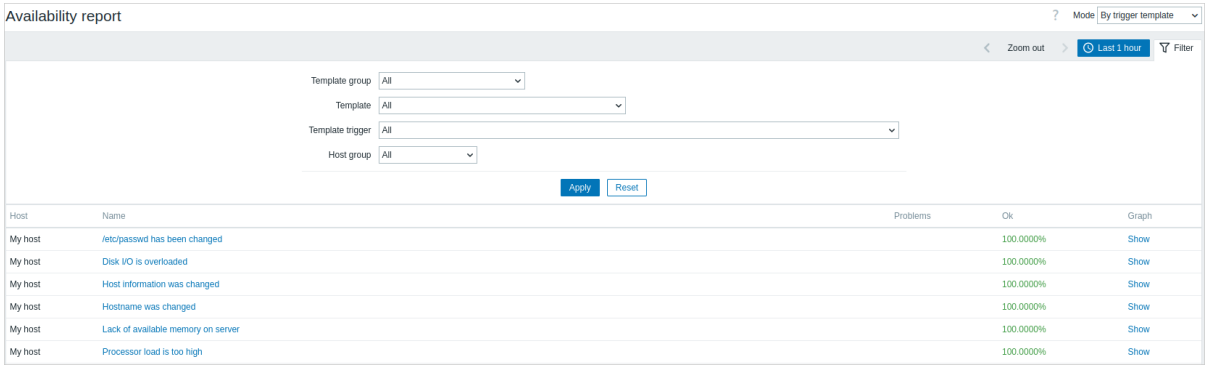
Visão geral

Em *Relatórios* > *Relatório de disponibilidade* você pode ver qual proporção de tempo cada trigger esteve em estado de problema/ok. A porcentagem de tempo para cada estado é exibida.

Assim, é fácil determinar a situação de disponibilidade de vários elementos em seu sistema.



No menu suspenso no canto superior direito, você pode escolher o modo de seleção - se deseja exibir triggers por hosts ou por triggers pertencentes a um template.



O nome do trigger é um link para os eventos mais recentes desse trigger.

Usando o filtro

O filtro pode ajudar a reduzir o número de hosts e/ou triggers exibidos. Para melhor desempenho na pesquisa, os dados são pesquisados com macros não resolvidas.

O filtro está localizado abaixo do nome da seção *Relatório de disponibilidade*. Ele pode ser aberto e recolhido clicando na guia *Filtro* à direita.

Filtrando por template de trigger

No modo *Por template de trigger*, os resultados podem ser filtrados por um ou vários parâmetros listados abaixo.

Parâmetro	Descrição
Grupo de template	Filtra hosts por triggers que são herdados de templates pertencentes ao grupo de template selecionado.
Template	Filtra hosts por triggers que são herdados do template selecionado, incluindo templates aninhados. Se um template aninhado tiver seus próprios triggers, esses triggers não serão exibidos.

Parâmetro	Descrição
Trigger de template	Filtra hosts pelo trigger selecionado. Outros triggers dos hosts filtrados não serão exibidos.
Grupo de hosts	Filtra hosts pertencentes ao grupo de hosts selecionado.

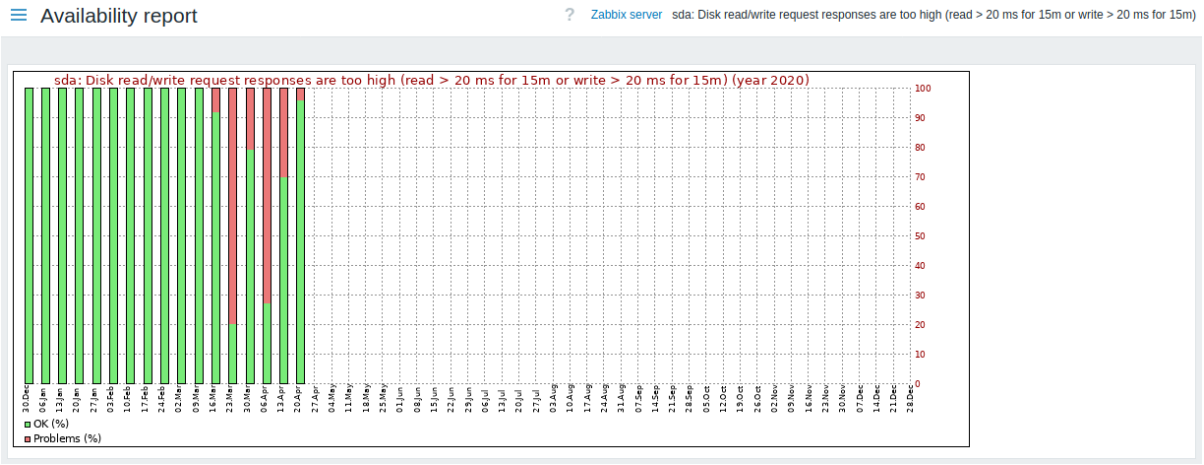
Filtrando por host

No modo *Por host*, os resultados podem ser filtrados por host ou por grupo de hosts. Especificar um grupo de hosts pai seleciona implicitamente todos os grupos de hosts aninhados.

Seletor de período de tempo

O seletor de *Período de tempo* permite selecionar períodos frequentemente necessários com um clique do mouse. O seletor de *Período de tempo* pode ser expandido e recolhido clicando na guia *Período de tempo* ao lado do filtro.

Clicar em *Mostrar* na coluna Gráfico exibe um gráfico de barras onde as informações de disponibilidade são exibidas no formato de barras, cada barra representando uma semana passada do ano atual.



A parte verde de uma barra representa o tempo OK e a vermelha o tempo de problema.

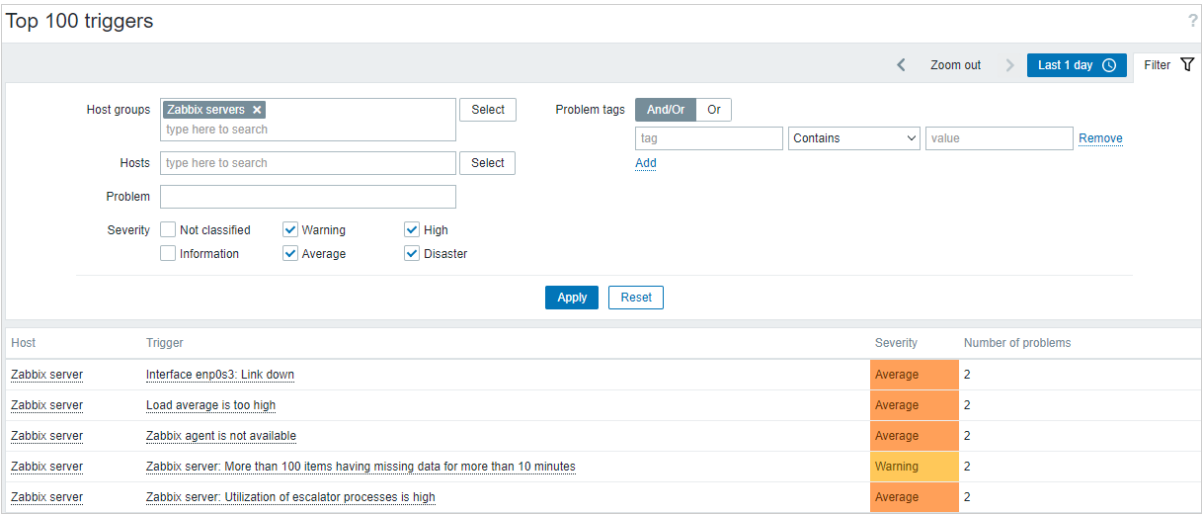
Efeito dos períodos de manutenção

A manutenção não exclui automaticamente o tempo do relatório de disponibilidade. Somente a manutenção configurada como *Sem coleta de dados* interrompe a coleta (portanto, nenhum problema é gerado) e, portanto, deixa a porcentagem de disponibilidade inalterada para os triggers afetados.

4 Top 100 triggers

Visão geral

Em *Relatórios* → *Top 100 triggers*, você pode ver os triggers com o maior número de problemas detectados durante o período selecionado.



Tanto as entradas das colunas de host quanto de trigger são links que oferecem algumas opções úteis:

- para host - clicar no nome do host exibe o menu do host
- para trigger - clicar no nome do trigger exibe links para os eventos mais recentes, gráfico simples para cada item do trigger e os formulários de configuração do próprio trigger e de cada item do trigger

Usando o filtro

Você pode usar o filtro para exibir triggers por grupo de hosts, host, nome do problema, tags ou severidade do trigger. Especificar um grupo de hosts pai seleciona implicitamente todos os grupos de hosts aninhados. Para melhor desempenho de pesquisa, os dados são pesquisados com macros não resolvidas.

O filtro está localizado abaixo do nome da seção *Top 100 triggers*. Ele pode ser aberto e recolhido clicando na guia *Filtro* à direita.

Seletor de período de tempo

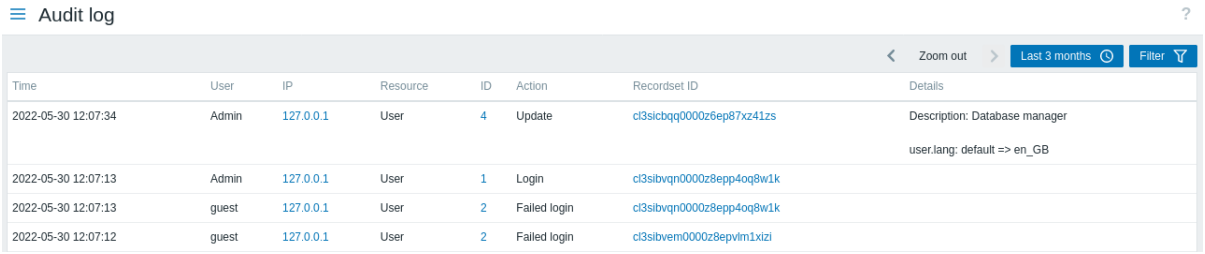
O seletor de *Período de tempo* permite selecionar períodos frequentemente necessários com um clique do mouse. O seletor de *Período de tempo* pode ser expandido e recolhido clicando na guia *Período de tempo* ao lado do filtro.

5 Log de auditoria

Visão geral

Na seção *Relatórios* → *Log de auditoria*, é possível visualizar os registros de atividade do usuário e do sistema.

Note:
Para que os registros de auditoria sejam coletados e exibidos, a caixa de seleção *Habilitar registro de auditoria* deve estar marcada na seção *Administração* → *Log de auditoria*. Sem essa configuração habilitada, o histórico de atividades não será registrado no banco de dados e não será exibido no log de auditoria.



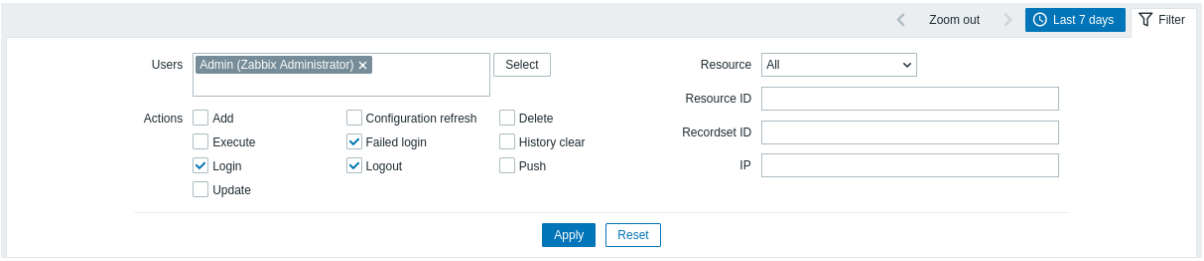
O log de auditoria exibe os seguintes dados:

Coluna	Descrição
Hora	Data e hora do registro de auditoria.
Usuário	Usuário que realizou a atividade.
IP	IP de onde a atividade foi iniciada. Ao clicar no hiperlink, os registros do log de auditoria serão filtrados por esse IP (suportado desde o Zabbix 7.0.4).
Recurso	Tipo do recurso afetado (*Token de API*, *Ação*, *Autenticação*, *Autoregistro*, etc.).
ID	ID do recurso afetado. Ao clicar no hiperlink, os registros do log de auditoria serão filtrados por esse ID de recurso.
Ação	Tipo da atividade (*Adicionar*, *Atualização de configuração*, *Excluir*, *Executar*, *Falha de login*, *Limpar histórico*, *Login*, *Logout*, *Push*, *Atualizar*).
ID do conjunto de registros	ID compartilhado para todos os registros do log de auditoria criados como resultado da mesma operação. Por exemplo, ao vincular um template a um host, um registro de auditoria separado é criado para cada entidade herdada do template (item, trigger, etc.) - todos esses registros terão o mesmo *ID do conjunto de registros*. Ao clicar no hiperlink, os registros do log de auditoria serão filtrados por esse *ID do conjunto de registros*.
Detalhes	Descrição do recurso e informações detalhadas sobre a atividade realizada. Se um registro contiver mais de duas linhas, um link adicional *Detalhes* será exibido. Clique neste link para visualizar a lista completa de alterações.

Note:
Quando um item trapper ou um item HTTP agent (com trapping habilitado) recebe algum dado, uma entrada no log de auditoria será adicionada apenas se os dados forem enviados usando o método de API `history.push`, e não pela ferramenta Zabbix sender.

Usando o filtro

O filtro está localizado abaixo da barra *Log de auditoria*. Ele pode ser aberto e recolhido clicando na guia *Filtro* no canto superior direito.



Você pode usar o filtro para restringir os registros por usuário, recurso afetado, ID do recurso, operação realizada (*ID do conjunto de registros*) e IP (desde o Zabbix 7.0.4). Dependendo do recurso, uma ou mais ações específicas podem ser selecionadas no filtro.

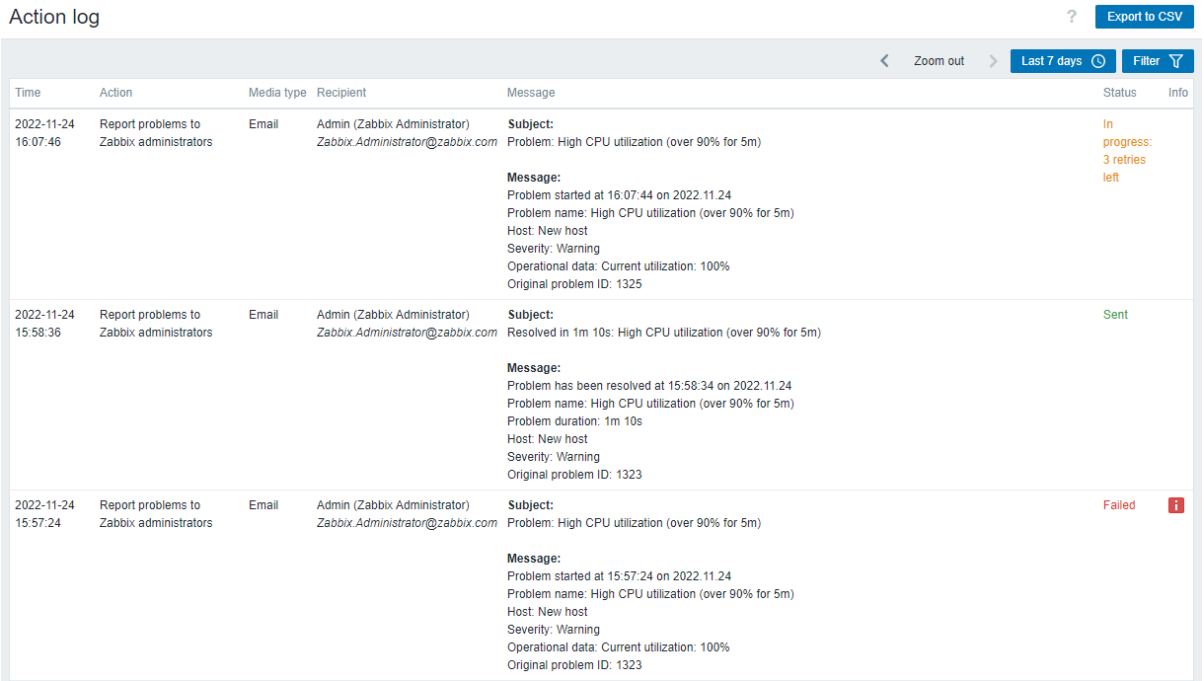
Para melhor desempenho de pesquisa, todos os dados são pesquisados com macros não resolvidas.

Seletor de período de tempo

O seletor de *Período de tempo* permite selecionar períodos frequentemente necessários com um clique do mouse. O seletor de *Período de tempo* pode ser expandido e recolhido clicando na guia *Período de tempo* no canto superior direito.

Visão geral

Na seção *Relatórios* → *Log de ações*, os usuários podem visualizar detalhes das operações (notificações, comandos remotos) executadas dentro de uma ação.



Dados exibidos:

Coluna	Descrição
Hora	Carimbo de data/hora da operação.
Ação	Nome da ação que causou as operações.

Coluna	Descrição
Tipo de mídia	Tipo de mídia (por exemplo, Email, Jira, etc.) usado para enviar uma notificação. Para operações que executaram comandos remotos, esta coluna ficará vazia.
Destinatário	Informações sobre o destinatário da notificação - nome de usuário, nome e sobrenome (entre parênteses) e informações adicionais dependendo do tipo de mídia (e-mail, nome de usuário, etc.). Para operações que executaram comandos remotos, esta coluna ficará vazia.
Mensagem	O conteúdo da mensagem, comando remoto ou nome do script global. Um comando remoto é separado do host de destino com dois pontos: `<host>:<comando>`. Por exemplo, se o comando remoto foi executado no Zabbix server, então a informação terá o seguinte formato: `Zabbix server:<comando>`.
Status	Status da operação: *Em andamento* - operação de envio de notificação está em andamento (o número restante de vezes que o server tentará enviar a notificação também é exibido) *Enviado* - notificação foi enviada *Executado* - comando remoto foi executado *Falhou* - operação não foi concluída
Info	Informações de erro (se houver) relacionadas à execução da operação.

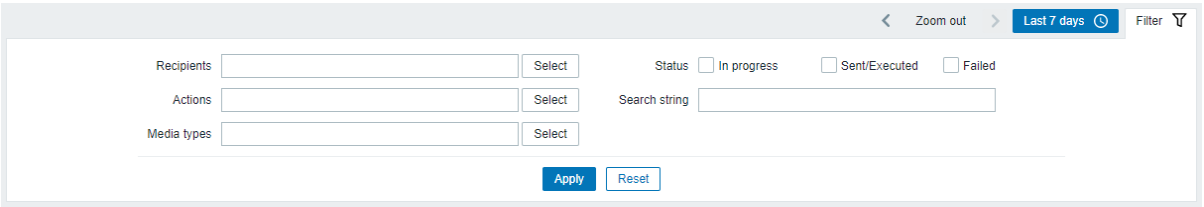
Botões

O botão no canto superior direito da página oferece a seguinte opção:

Export to CSV	Exporta os registros do log de ações da página atual para um arquivo CSV. Se um filtro for aplicado, apenas os registros filtrados serão exportados. No arquivo CSV exportado, as colunas ”Destinatário” e ”Mensagem” são divididas em várias colunas - ”Nome de usuário Zabbix do destinatário”, ”Nome do destinatário”, ”Sobrenome do destinatário”, ”Destinatário” e ”Assunto”, ”Mensagem”, ”Comando”.

Usando o filtro

O filtro está localizado abaixo da barra *Log de ações*. Ele pode ser aberto e recolhido clicando na guia *Filtro* no canto superior direito da página.



Você pode usar o filtro para restringir os registros por destinatários de notificações, ações, tipos de mídia, status ou pelo conteúdo da mensagem/comando remoto (*String de pesquisa*). Para melhor desempenho na pesquisa, os dados são pesquisados com macros não resolvidas.

Seletor de período de tempo

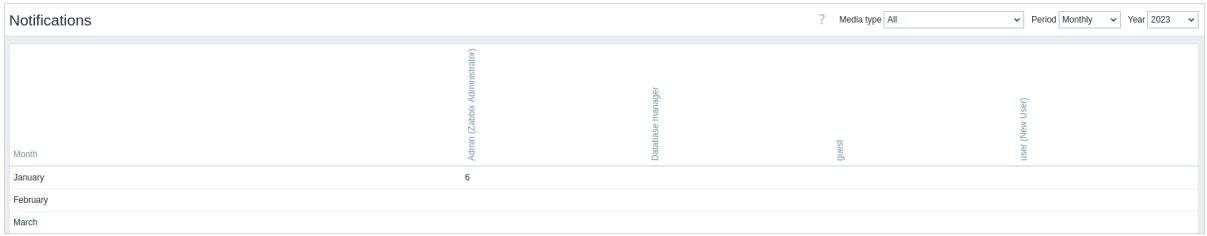
O seletor de *Período de tempo* permite selecionar períodos frequentemente necessários com um clique do mouse. O seletor de *Período de tempo* pode ser expandido e recolhido clicando na guia *Período de tempo* ao lado do filtro.

7 Notificações

Visão geral

Na seção *Relatórios* → *Notificações* um relatório sobre o número de notificações enviadas a cada usuário são exibidas.

Nas listas suspensas no canto superior direito, você pode escolher a mídia tipo (ou todos), período (dados para cada dia/semana/mês/ano) e ano para as notificações enviadas.



Cada coluna exibe os totais por um usuário do sistema.

6 Coleta de dados

Visão geral

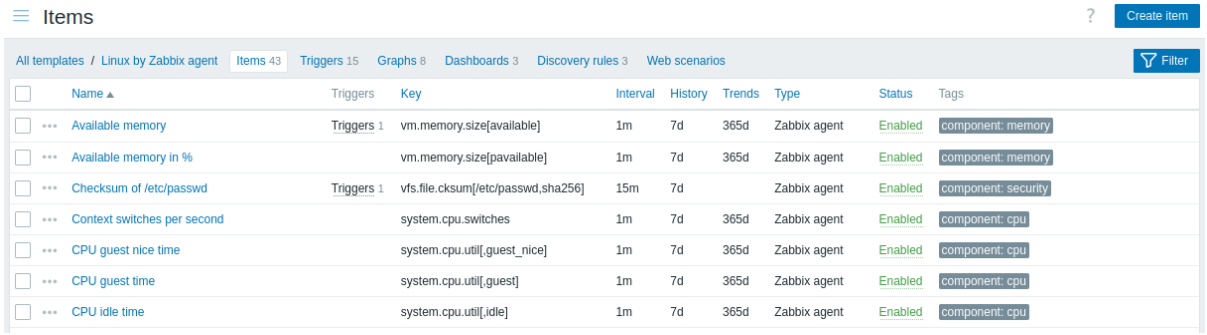
Este menu apresenta seções relacionadas à configuração da coleta de dados.

1 Items

Visão geral

A lista de items de um template pode ser acessada em *Coleta de dados* → *Templates* clicando em *Items* para o respectivo template.

Uma lista de items existentes é exibida.



Dados exibidos:

Coluna	Descrição
Menu do item	Clique no ícone de três pontos para abrir o menu do item.
Template	Template ao qual o item pertence. Desde o Zabbix 7.0.4, clicar no nome do template abre o formulário de configuração do template. Esta coluna é exibida apenas se vários templates ou nenhum template estiverem selecionados no filtro.
Nome	Nome do item exibido como um link azul para os detalhes do item. Clicar no link do nome do item abre o formulário de configuração do item. Se o item for herdado de outro template, o nome do template é exibido antes do nome do item como um link cinza. Clicar no link do template abre a lista de items naquele nível de template.
Triggers	Ao passar o mouse sobre Triggers, será exibida uma infobox mostrando os triggers associados ao item. O número de triggers é exibido em cinza.
Key	A chave do item é exibida.
Intervalo	A frequência da checagem é exibida.
Histórico	Quantos dias o histórico dos dados do item será mantido é exibido.
Tendências	Quantos dias o histórico de tendências do item será mantido é exibido.
Tipo	O tipo do item é exibido (Zabbix agent, SNMP agent, checagem simples, etc).
Status	O status do item é exibido - *Habilitado* ou *Desabilitado*. Clicando no status você pode alterá-lo - de Habilitado para Desabilitado (e vice-versa).
Tags	As tags do item são exibidas. Até três tags (pares nome:valor) podem ser exibidas. Se houver mais tags, um link "..." é exibido, permitindo ver todas as tags ao passar o mouse.

Para configurar um novo item, clique no botão *Criar item* no canto superior direito.

Opções de edição em massa

Os botões abaixo da lista oferecem algumas opções de edição em massa:

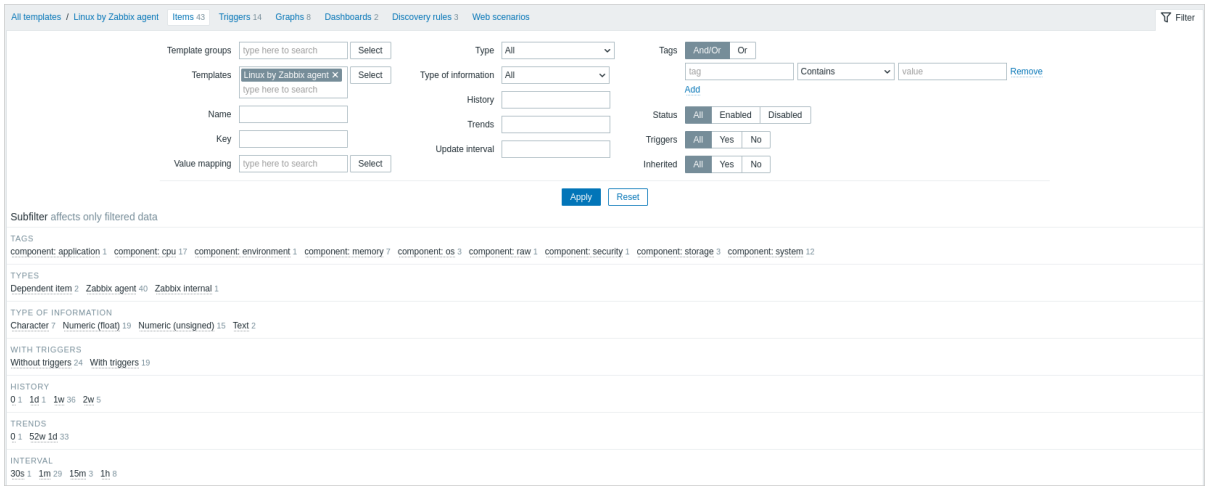
- *Ativar* - altera o status do item para *Ativado*.
- *Desativar* - altera o status do item para *Desativado*.
- *Copiar* - copia os items para outros hosts ou templates.
- *Atualização em massa* - atualiza várias propriedades de vários items de uma só vez.
- *Excluir* - exclui os items.

Para usar essas opções, marque as caixas de seleção antes dos respectivos items e clique no botão desejado.

Usando o filtro

A lista de items pode conter muitos items. Usando o filtro, você pode filtrar alguns deles para localizar rapidamente os items que está procurando. Para melhor desempenho de pesquisa, os dados são pesquisados com macros não resolvidas.

O ícone *Filtro* está disponível no canto superior direito. Clicando nele será aberto um filtro onde você pode especificar os critérios de filtragem desejados.



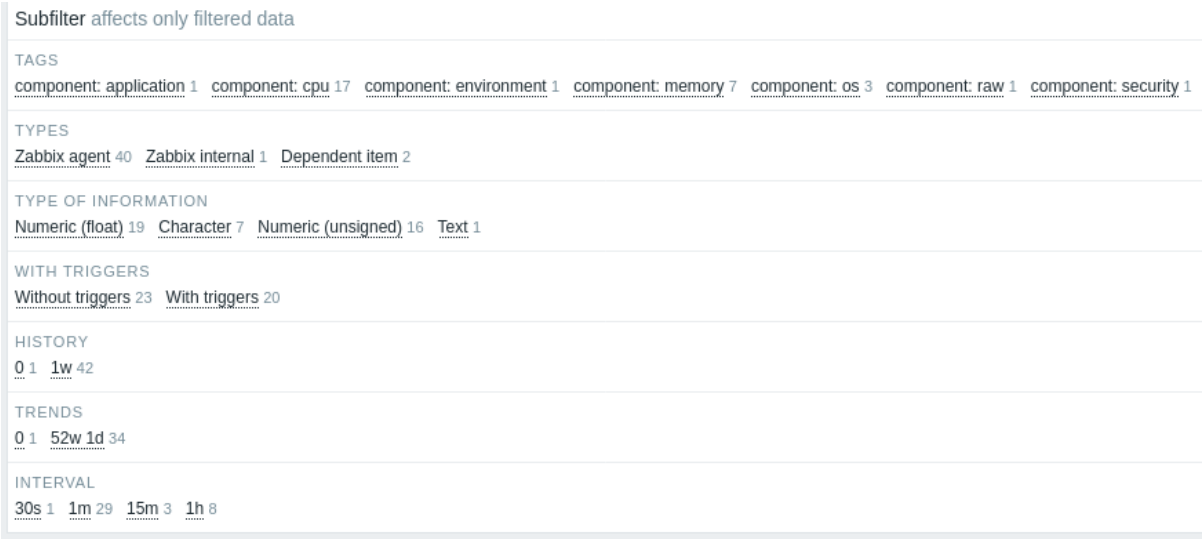
Parâmetro	Descrição
Grupos de templates	Filtrar por um ou mais grupos de templates. Especificar um grupo de templates pai seleciona implicitamente todos os grupos aninhados.
Templates	Filtrar por um ou mais templates.
Nome	Filtrar por nome do item.
Chave	Filtrar por chave do item.
Mapeamento de valores	Filtrar pelo mapa de valores utilizado. Este parâmetro não é exibido se a opção *Templates* estiver vazia.
Tipo	Filtrar por tipo de item (Zabbix agent, SNMP agent, etc.).
Tipo de informação	Filtrar por tipo de informação (Numérico sem sinal, float, etc.).
Histórico	Filtrar por quanto tempo o histórico do item é mantido.
Tendências	Filtrar por quanto tempo as tendências do item são mantidas.
Intervalo de atualização	Filtrar por intervalo de atualização do item.
Tags	Especifique tags para limitar o número de items exibidos. É possível incluir ou excluir tags e valores de tags específicos. Várias condições podem ser definidas. A correspondência do nome da tag é sempre sensível a maiúsculas e minúsculas. Existem vários operadores disponíveis para cada condição: **Existe** - inclui os nomes de tags especificados **Igual** - inclui os nomes e valores de tags especificados (sensível a maiúsculas e minúsculas) **Contém** - inclui os nomes de tags especificados onde os valores das tags contêm a string inserida (correspondência de substring, não sensível a maiúsculas e minúsculas) **Não existe** - exclui os nomes de tags especificados **Não é igual** - exclui os nomes e valores de tags especificados (sensível a maiúsculas e minúsculas) **Não contém** - exclui os nomes de tags especificados onde os valores das tags contêm a string inserida (correspondência de substring, não sensível a maiúsculas e minúsculas) Existem dois tipos de cálculo para as condições: **E/Ou** - todas as condições devem ser atendidas, condições com o mesmo nome de tag serão agrupadas pela condição Ou **Ou** - basta que uma condição seja atendida

Parâmetro	Descrição
Status	Filtrar por status do item - *Habilitado* ou *Desabilitado*.
Triggers	Filtrar items com (ou sem) triggers.
Herdado	Filtrar items herdados (ou não herdados) de templates vinculados.

Usando subfiltro

O subfiltro permite modificar ainda mais o filtro do filtro principal.

Ele contém links clicáveis para acesso rápido aos itens relacionados. Os itens são relacionados por uma entidade comum - tag, tipo de item, intervalo de atualização do item, etc. Quando uma entidade é clicada, ela é destacada com um fundo cinza, e os itens são imediatamente filtrados (não é necessário clicar em *Aplicar* no filtro principal). Clicar em outra entidade a adiciona aos resultados filtrados. Clicar novamente na entidade remove o filtro.



Os subfiltros são gerados com base nos dados filtrados, que são limitados a 1000 registros. Se você quiser ver mais registros no subfiltro, é necessário aumentar o valor do parâmetro *Limite para resultados de pesquisa e filtro* (em *Administração -> Geral -> GUI*).

Ao contrário do filtro principal, o subfiltro é atualizado a cada solicitação de atualização da tabela para sempre ter informações atualizadas sobre as opções de filtragem disponíveis e seus números de contagem.

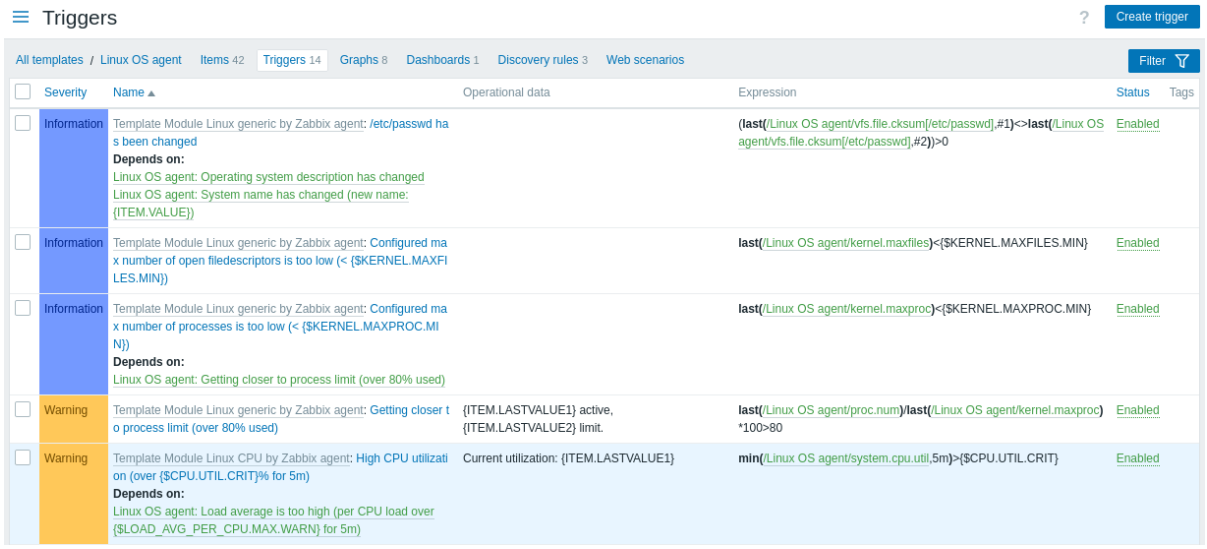
O número de entidades exibidas é limitado a 100 horizontalmente. Se houver mais, um ícone de três pontos é exibido no final; ele não é clicável.

Um número ao lado de cada entidade clicável indica o número de itens agrupados nela (com base nos resultados do filtro principal). Quando uma entidade é clicada, os números com outras entidades disponíveis são exibidos com um sinal de mais indicando quantos itens podem ser adicionados à seleção atual. Entidades sem itens não são exibidas, a menos que tenham sido selecionadas anteriormente no subfiltro.

2 Triggers

Visão geral

A lista de triggers de um template pode ser acessada em *Coleta de dados* → *Templates* clicando em *Triggers* para o respectivo template.



Dados exibidos:

Coluna	Descrição
Severidade	A severidade do trigger é exibida tanto pelo nome quanto pela cor de fundo da célula.
Template	Template ao qual o trigger pertence. Desde o Zabbix 7.0.4, clicar no nome do template abre o formulário de configuração do template. Esta coluna é exibida apenas se vários templates ou nenhum template estiverem selecionados no filtro.
Nome	Nome do trigger exibido como um link azul para os detalhes do trigger. Clicar no nome do trigger abre o formulário de configuração do trigger. Se o trigger for herdado de outro template, o nome do template é exibido antes do nome do trigger como um link cinza. Clicar no link do template abre a lista de triggers no nível daquele template.
Dados operacionais	Definição dos dados operacionais do trigger, contendo strings arbitrárias e macros que serão resolvidas dinamicamente em *Monitoramento* → *Problemas*.
Expressão	A expressão do trigger é exibida. A parte template-item da expressão é exibida como um link, levando ao formulário de configuração do item.
Status	O status do trigger é exibido - *Habilitado* ou *Desabilitado*. Clicando no status você pode alterá-lo - de Habilitado para Desabilitado (e vice-versa).
Tags	Se um trigger contiver tags, o nome e o valor da tag serão exibidos nesta coluna.

Para configurar um novo trigger, clique no botão *Criar trigger* no canto superior direito.

Opções de edição em massa

Os botões abaixo da lista oferecem algumas opções de edição em massa:

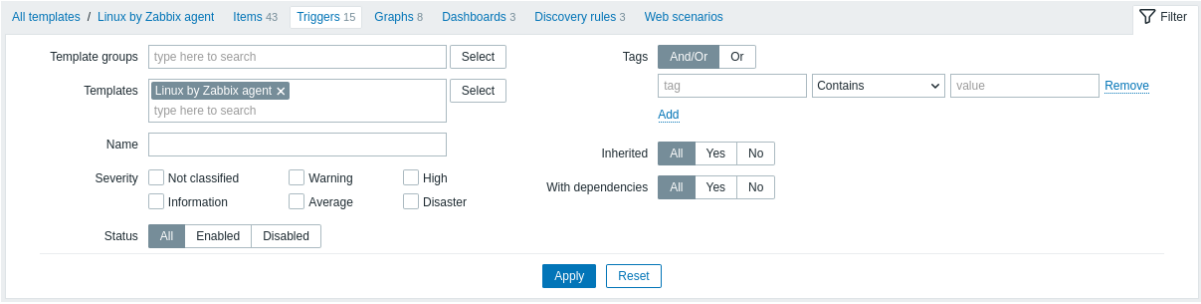
- *Ativar* - altera o status do gatilho para *Ativado*
- *Desativar* - altera o status do gatilho para *Desativado*
- *Copiar* - copia os gatilhos para outros hosts ou modelos
- *Atualização em massa* - atualiza várias propriedades para vários gatilhos de uma vez só
- *Excluir* - exclui os gatilhos

Para usar essas opções, marque as caixas de seleção antes do respectivo acionadores e, em seguida, clique no botão necessário.

Usando o filtro

Você pode usar o filtro para exibir apenas os triggers do seu interesse. Para melhor desempenho na pesquisa, os dados são pesquisados com macros não resolvidas.

O ícone *Filtro* está disponível no canto superior direito. Clicando nele, será aberto um filtro onde você pode especificar os critérios de filtragem desejados.



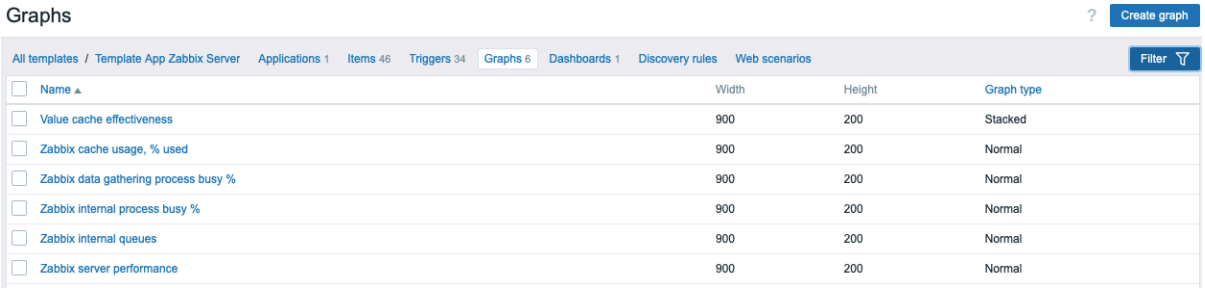
Parâmetro	Descrição
Grupos de template	Filtrar por um ou mais grupos de template. Ao especificar um grupo de template pai, todos os grupos aninhados são implicitamente selecionados.
Templates	Filtrar por um ou mais templates. Se grupos de template já estiverem selecionados acima, a seleção de template será limitada a esses grupos.
Nome	Filtrar por nome do trigger.
Severidade	Selecione para filtrar por uma ou várias severidades de trigger.
Status	Filtrar por status do trigger.
Tags	Filtrar por nome e valor da tag do trigger. É possível incluir ou excluir tags e valores de tags específicos. Várias condições podem ser definidas. A correspondência do nome da tag diferencia maiúsculas de minúsculas. Existem vários operadores disponíveis para cada condição: **Existe** - inclui os nomes de tag especificados **Igual** - inclui os nomes e valores de tag especificados (diferencia maiúsculas de minúsculas) **Contém** - inclui os nomes de tag especificados onde os valores de tag contêm a string inserida (correspondência de substring, não diferencia maiúsculas de minúsculas) **Não existe** - exclui os nomes de tag especificados **Diferente** - exclui os nomes e valores de tag especificados (diferencia maiúsculas de minúsculas) **Não contém** - exclui os nomes de tag especificados onde os valores de tag contêm a string inserida (correspondência de substring, não diferencia maiúsculas de minúsculas) Existem dois tipos de cálculo para as condições: **E/Ou** - todas as condições devem ser atendidas, condições com o mesmo nome de tag serão agrupadas pela condição Ou **Ou** - basta que uma condição seja atendida Macros e funções de macro são suportadas nos campos de nome e valor da tag.
Herdado	Filtrar triggers herdados (ou não herdados) de templates vinculados.
Com dependências	Filtrar triggers com (ou sem) dependências.

3 Gráficos

Visão geral

A lista de gráficos personalizados para um template pode ser acessada em *Coleta de dados* → *Templates* clicando em *Gráficos* para o respectivo template.

Uma lista de gráficos existentes é exibida.



Dados exibidos:

Coluna	Descrição
Template	Template ao qual o gráfico pertence. Desde o Zabbix 7.0.4, clicar no nome do template abre o formulário de configuração do template. Esta coluna é exibida apenas se vários templates ou nenhum template estiverem selecionados no filtro.
Nome	Nome do gráfico personalizado, exibido como um link azul para os detalhes do gráfico. Clicar no link do nome do gráfico abre o formulário de configuração do gráfico. Se o gráfico for herdado de outro template, o nome do template é exibido antes do nome do gráfico como um link cinza. Clicar no link do template abre a lista de gráficos naquele nível de template.
Largura	A largura do gráfico é exibida.
Altura	A altura do gráfico é exibida.
Tipo de gráfico	O tipo de gráfico é exibido - *Normal*, *Empilhado*, *Pizza* ou *Explodido*.

Para configurar um novo gráfico, clique no botão *Criar gráfico* no canto superior direito.

Opções de edição em massa

Os botões abaixo da lista oferecem algumas opções de edição em massa:

- *Copiar* - copie os gráficos para outros hosts ou modelos
- *Excluir* - excluir os gráficos

Para usar essas opções, marque as caixas de seleção antes dos respectivos gráficos, em seguida, clique no botão desejado.

Usando filtro

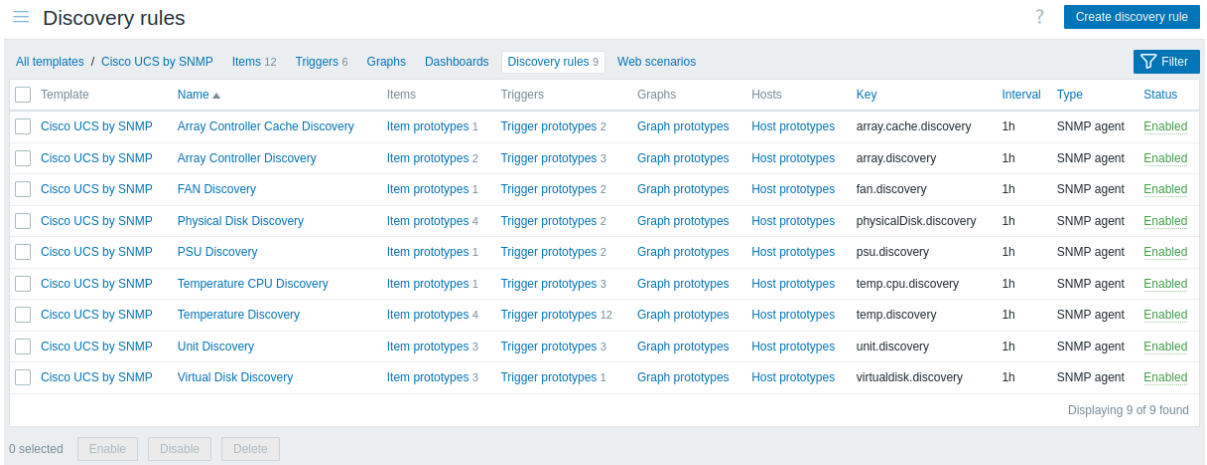
Você pode filtrar gráficos por grupo de template e template. Para melhor desempenho de pesquisa, os dados são pesquisados com macros não resolvidas.

4 Regras de descoberta

Visão geral

A lista de regras de descoberta de baixo nível para um template pode ser acessada em *Coleta de dados* → *Templates* clicando em *Descoberta* para o respectivo template.

Uma lista das regras de descoberta de baixo nível existentes é exibida. Também é possível ver todas as regras de descoberta independentemente do template, ou todas as regras de descoberta de um grupo de templates específico, alterando as configurações do filtro.



Dados exibidos:

Coluna	Descrição
Template	O template ao qual a regra de descoberta pertence. Desde o Zabbix 7.0.4, clicar no nome do template abre o formulário de configuração do template.

Coluna	Descrição
Nome	Nome da regra, exibido como um link azul. Clicar no nome da regra abre o formulário de configuração da regra de descoberta de baixo nível. Se a regra de descoberta for herdada de outro template, o nome do template é exibido antes do nome da regra como um link cinza. Clicar no link do template abre a lista de regras de descoberta no nível daquele template.
Itens	Um link para a lista de protótipos de item é exibido. O número de protótipos de item existentes é exibido em cinza.
Triggers	Um link para a lista de protótipos de trigger é exibido. O número de protótipos de trigger existentes é exibido em cinza.
Gráficos	Um link para a lista de protótipos de gráfico é exibido. O número de protótipos de gráfico existentes é exibido em cinza.
Hosts	Um link para a lista de protótipos de host é exibido. O número de protótipos de host existentes é exibido em cinza.
Chave	A chave do item usada para descoberta é exibida.
Intervalo	A frequência de execução da descoberta é exibida.
Tipo	O tipo de item usado para descoberta é exibido (agent Zabbix, agent SNMP, etc).
Status	O status da regra de descoberta é exibido - *Habilitado* ou *Desabilitado*. Clicando no status você pode alterá-lo - de Habilitado para Desabilitado (e vice-versa).

Para configurar uma nova regra de descoberta de baixo nível, clique no botão *Criar regra de descoberta* no canto superior direito.

Opções de edição em massa

Os botões abaixo da lista oferecem algumas opções de edição em massa:

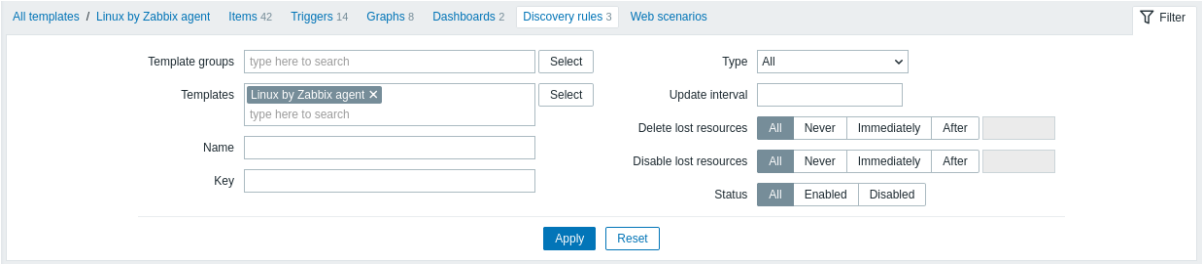
- *Ativar* - altere o status da regra de descoberta de baixo nível para *Ativado*
- *Desativar* - altere o status da regra de descoberta de baixo nível para *Desativado*
- *Excluir* - exclui as regras de descoberta de baixo nível

Para usar essas opções, marque as caixas de seleção antes do respectivo regras de descoberta e, em seguida, clique no botão necessário.

Usando o filtro

Você pode usar o filtro para exibir apenas as regras de descoberta que são do seu interesse. Para melhor desempenho na busca, os dados são pesquisados com macros não resolvidas.

O ícone *Filtro* está disponível no canto superior direito. Clicando nele, será aberto um filtro onde você pode especificar os critérios de filtragem desejados, como template, nome da regra de descoberta, chave do item, tipo de item, etc.



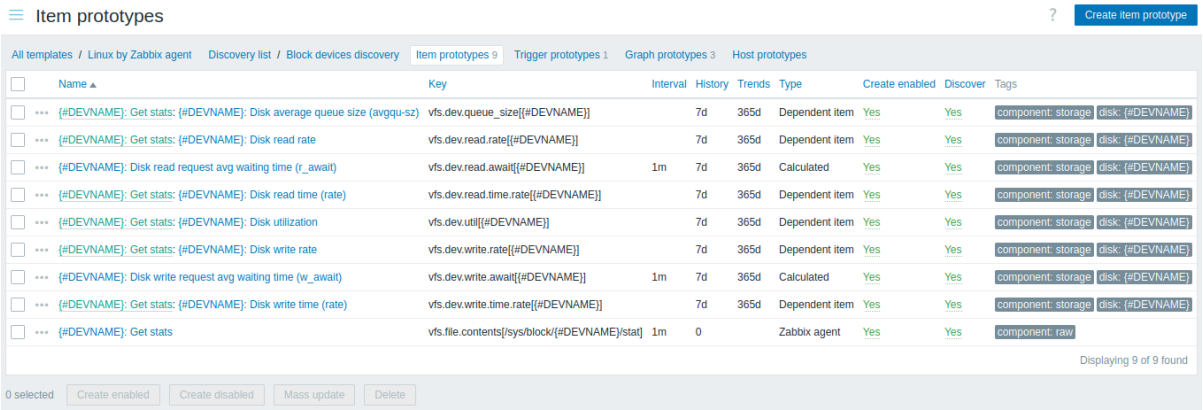
Parâmetro	Descrição
Grupos de templates	Filtrar por um ou mais grupos de templates. Especificar um grupo de templates pai seleciona implicitamente todos os grupos aninhados.
Templates	Filtrar por um ou mais templates.
Nome	Filtrar por nome da regra de descoberta.
Chave	Filtrar por chave do item de descoberta.
Tipo	Filtrar por tipo do item de descoberta.
Intervalo de atualização	Filtrar por intervalo de atualização. Não disponível para Zabbix trapper e itens dependentes.
Excluir recursos perdidos	Filtrar por período de *Excluir recursos perdidos*.
Desabilitar recursos perdidos	Filtrar por período de *Desabilitar recursos perdidos*.
Status	Filtrar por status da regra de descoberta (Todos/Habilitado/Desabilitado).

1 Prototipagem de item

Visão geral

Nesta seção, os protótipos de item configurados de uma regra de descoberta de baixo nível no template são exibidos.

Se o template estiver vinculado ao host, os protótipos de item se tornarão a base para a criação de items reais do host durante a descoberta de baixo nível.



Dados exibidos:

Coluna	Descrição
Nome	Nome do protótipo de item, exibido como um link azul. Ao clicar no nome, o formulário de configuração do protótipo de item é aberto. Se o protótipo de item pertencer a um template vinculado, o nome do template será exibido antes do nome do protótipo de item como um link cinza. Ao clicar no link do template, a lista de protótipos de item no nível do template vinculado será aberta. Se o protótipo de item for do tipo ”Item dependente”, o nome do item mestre será exibido antes do nome do protótipo de item em verde (como no caso dos protótipos de item no nível do host).
Chave	A chave do protótipo de item é exibida.
Intervalo	A frequência da verificação é exibida.
Histórico	Quantos dias manter o histórico de dados do item é exibido.
Tendências	Quantos dias manter o histórico de tendências do item é exibido.
Tipo	O tipo do protótipo de item é exibido (Zabbix agent, SNMP agent, verificação simples, etc).
Criar habilitado	Criar o item com base neste protótipo como: **Sim** - habilitado **Não** - desabilitado. Você pode alternar entre ’Sim’ e ’Não’ clicando neles.
Descobrir	Descobrir o item com base neste protótipo: **Sim** - descobrir **Não** - não descobrir. Você pode alternar entre ’Sim’ e ’Não’ clicando neles.
Tags	As tags do protótipo de item são exibidas.

Para configurar um novo protótipo de item, clique no botão *Criar protótipo de item* no canto superior direito.

Opções de edição em massa

Os botões abaixo da lista oferecem algumas opções de edição em massa:

- *Criar habilitado* - cria esses items como *Habilitado*
- *Criar desabilitado* - cria esses items como *Desabilitado*
- *Atualização em massa* - atualiza em massa esses protótipos de item
- *Excluir* - exclui esses protótipos de item

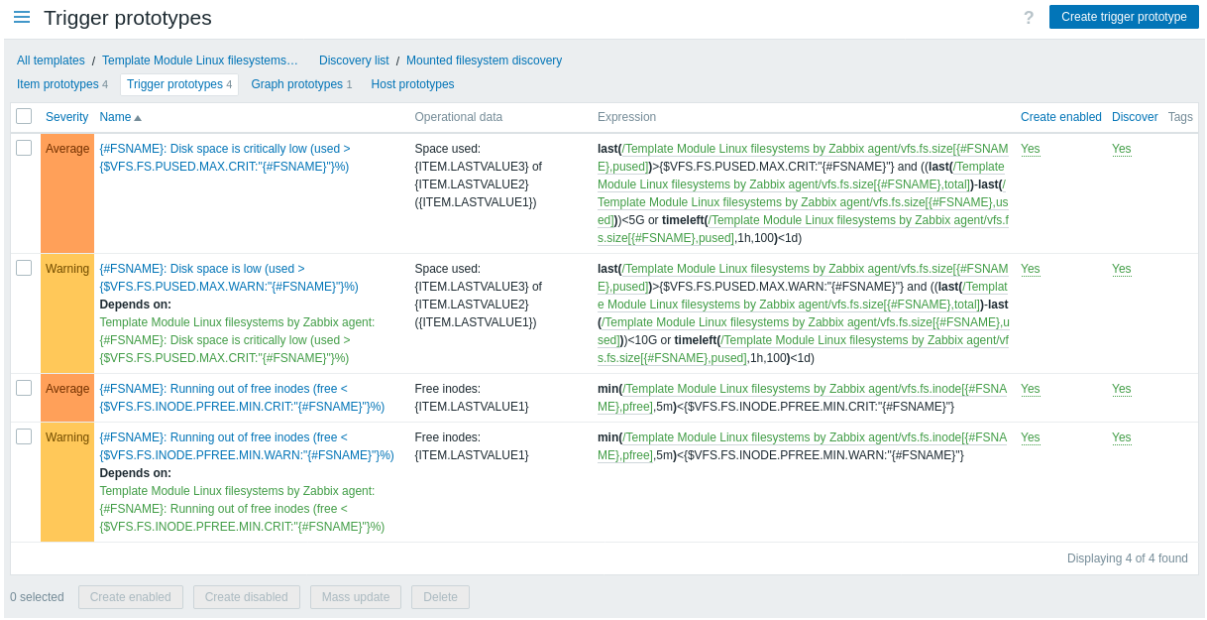
Para usar essas opções, marque as caixas de seleção antes dos respectivos protótipos de item e clique no botão desejado.

2 Protótipos de trigger

Visão geral

Nesta seção, os protótipos de trigger configurados de uma regra de descoberta de baixo nível no template são exibidos.

Se o template estiver vinculado ao host, os protótipos de trigger servirão de base para a criação de triggers reais do host durante a descoberta de baixo nível.



Dados exibidos:

Coluna	Descrição
Severidade	A severidade do trigger é exibida tanto pelo nome quanto pela cor de fundo da célula.
Nome	Nome do protótipo de trigger, exibido como um link azul. Clicar no nome abre o formulário de configuração do protótipo de trigger. Se o protótipo de trigger pertencer a um template vinculado, o nome do template será exibido antes do nome do trigger como um link cinza. Clicar no link do template abre a lista de protótipos de trigger no nível do template vinculado.
Expressão	A expressão do trigger é exibida. A parte template-item da expressão é exibida como um link, levando ao formulário de configuração do item.
Dados operacionais	O formato dos dados operacionais do trigger é exibido, contendo strings arbitrárias e macros que serão resolvidas dinamicamente em *Monitoramento* → *Problemas*.
Criar habilitado	Cria o trigger com base neste protótipo como: **Sim** - habilitado **Não** - desabilitado. Você pode alternar entre 'Sim' e 'Não' clicando neles.
Descobrir	Descobre o trigger com base neste protótipo: **Sim** - descobrir **Não** - não descobrir. Você pode alternar entre 'Sim' e 'Não' clicando neles.
Tags	As tags do protótipo de trigger são exibidas.

Para configurar um novo protótipo de trigger, clique no botão *Criar protótipo de trigger* no canto superior direito.

Opções de edição em massa

Os botões abaixo da lista oferecem algumas opções de edição em massa:

- *Criar habilitado* - cria esses triggers como *Habilitado*
- *Criar desabilitado* - cria esses triggers como *Desabilitado*
- *Atualização em massa* - atualiza em massa esses protótipos de trigger
- *Excluir* - exclui esses protótipos de trigger

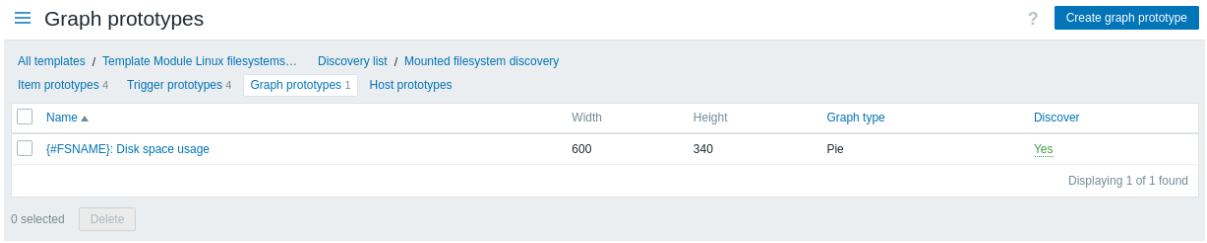
Para usar essas opções, marque as caixas de seleção antes dos respectivos protótipos de trigger e clique no botão desejado.

3 Protótipos de gráficos

Visão geral

Nesta seção, os protótipos de gráficos configurados de uma regra de descoberta de baixo nível no template são exibidos.

Se o template estiver vinculado ao host, os protótipos de gráficos servirão de base para a criação de gráficos reais do host durante a descoberta de baixo nível.



Dados exibidos:

Coluna	Descrição
Nome	Nome do protótipo de gráfico, exibido como um link azul. Clicar no nome abre o formulário de configuração do protótipo de gráfico. Se o protótipo de gráfico pertencer a um template vinculado, o nome do template será exibido antes do nome do gráfico como um link cinza. Clicar no link do template abre a lista de protótipos de gráficos no nível do template vinculado.
Largura	A largura do protótipo de gráfico é exibida.
Altura	A altura do protótipo de gráfico é exibida.
Tipo	O tipo do protótipo de gráfico é exibido - *Normal*, *Empilhado*, *Pizza* ou *Explodido*.
Descobrir	Descobrir o gráfico com base neste protótipo: **Sim** - descobrir **Não** - não descobrir. Você pode alternar entre 'Sim' e 'Não' clicando neles.

Para configurar um novo protótipo de gráfico, clique no botão *Criar protótipo de gráfico* no canto superior direito.

Opções de edição em massa

Os botões abaixo da lista oferecem algumas opções de edição em massa:

- *Excluir* - exclui esses protótipos de gráficos

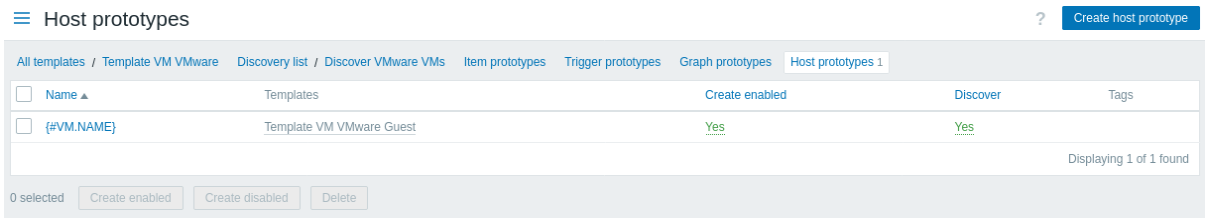
Para usar essas opções, marque as caixas de seleção antes dos respectivos protótipos de gráficos e clique no botão desejado.

4 Protótipos de host

Visão geral

Nesta seção, os protótipos de host configurados de uma regra de descoberta de baixo nível no template são exibidos.

Se o template estiver vinculado ao host, os protótipos de host se tornarão a base para a criação de hosts reais durante a descoberta de baixo nível.



Dados exibidos:

Coluna	Descrição
Nome	Nome do protótipo de host, exibido como um link azul. Clicar no nome abre o formulário de configuração do protótipo de host. Se o protótipo de host pertencer a um template vinculado, o nome do template será exibido antes do nome do host como um link cinza. Clicar no link do template abre a lista de protótipos de host no nível do template vinculado.
Templates	Os templates do protótipo de host são exibidos.

Coluna	Descrição
Criar habilitado	Criar o host com base neste protótipo como: **Sim** - habilitado **Não** - desabilitado. Você pode alternar entre 'Sim' e 'Não' clicando neles.
Descobrir	Descobrir o host com base neste protótipo: **Sim** - descobrir **Não** - não descobrir. Você pode alternar entre 'Sim' e 'Não' clicando neles.
Tags	As tags do protótipo de host são exibidas.

Para configurar um novo protótipo de host, clique no botão *Criar protótipo de host* no canto superior direito.

Opções de edição em massa

Os botões abaixo da lista oferecem algumas opções de edição em massa:

- *Criar habilitado* - cria esses hosts como *Habilitado*
- *Criar desabilitado* - cria esses hosts como *Desabilitado*
- *Excluir* - exclui esses protótipos de host

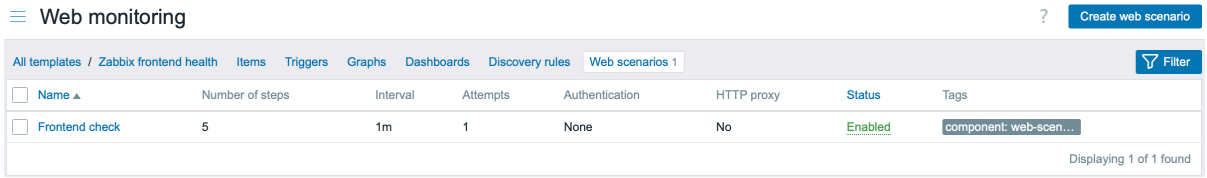
Para usar essas opções, marque as caixas de seleção antes dos respectivos protótipos de host e clique no botão desejado.

5 Cenários Web

Visão geral

A lista de cenários web para um template pode ser acessada em *Coleta de dados* → *Templates* clicando em *Web* para o respectivo template.

Uma lista de cenários web existentes é exibida.



Dados exibidos:

Coluna	Descrição
Nome	Nome do cenário web. Clicar no nome do cenário web abre o formulário de configuração do cenário web. Se o cenário web for herdado de outro template, o nome do template é exibido antes do nome do cenário web como um link cinza. Clicar no link do template abre a lista de cenários web no nível daquele template.
Número de etapas	O número de etapas que o cenário contém.
Intervalo de atualização	Com que frequência o cenário é executado.
Tentativas	Quantas tentativas de execução das etapas do cenário web são realizadas.
Autenticação	O método de autenticação é exibido - Basic, NTLM, Kerberos, Digest ou Nenhum.
Proxy HTTP	Exibe o proxy HTTP ou 'Não' se não for usado.
Status	O status do cenário web é exibido - *Habilitado* ou *Desabilitado*. Clicando no status você pode alterá-lo.
Tags	As tags do cenário web são exibidas. Até três tags (pares nome:valor) podem ser exibidas. Se houver mais tags, um link "..." é exibido que permite ver todas as tags ao passar o mouse.

Para configurar um novo cenário web, clique no botão *Criar cenário web* no canto superior direito.

Opções de edição em massa

Os botões abaixo da lista oferecem algumas opções de edição em massa:

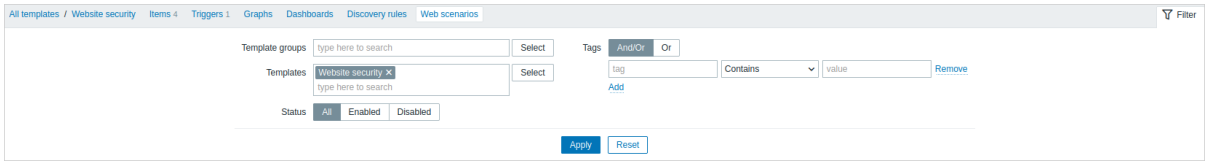
- *Ativar* - altera o status do cenário para *Ativado*
- *Desativar* - altera o status do cenário para *Desativado*
- *Excluir* - exclui os cenários da web

Para usar essas opções, marque as caixas de seleção antes do respectivo web cenários e, em seguida, clique no botão necessário.

Usando o filtro

Você pode usar o filtro para exibir apenas os cenários nos quais está interessado. Para melhor desempenho de pesquisa, os dados são pesquisados com macros não resolvidas.

O link *Filtro* está disponível acima da lista de cenários web. Se você clicar nele, um filtro ficará disponível onde você pode filtrar cenários por grupo de template, template, status e tags.

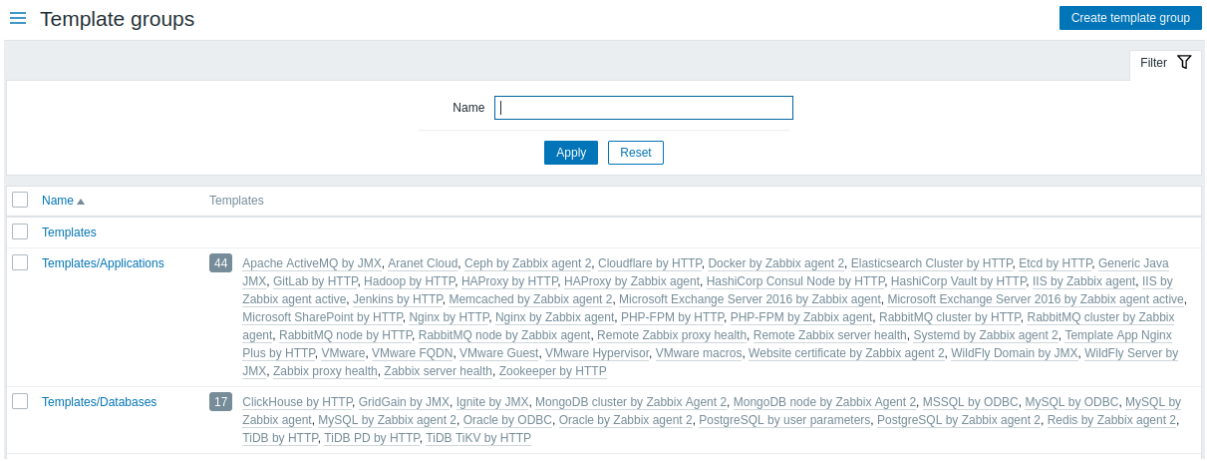


1 Grupos de templates

Visão geral

Na seção *Coleta de dados* → *Grupos de templates*, os usuários podem configurar e manter grupos de templates.

Uma lista dos grupos de templates existentes com seus detalhes é exibida. Você pode pesquisar e filtrar grupos de templates pelo nome.



Dados exibidos:

Coluna	Descrição
Nome	Nome do grupo de templates. Clicar no nome do grupo abre o formulário de configuração do grupo.
Templates	Número de templates no grupo (exibido em cinza) seguido pela lista de membros do grupo. Clicar no nome de um template abrirá o formulário de configuração do template. Clicar no número abre a lista de templates neste grupo.

Opções de edição em massa

Para excluir vários grupos de template de uma vez, marque as caixas de seleção antes dos respectivos grupos e clique no botão Excluir abaixo da lista.

Usando o filtro

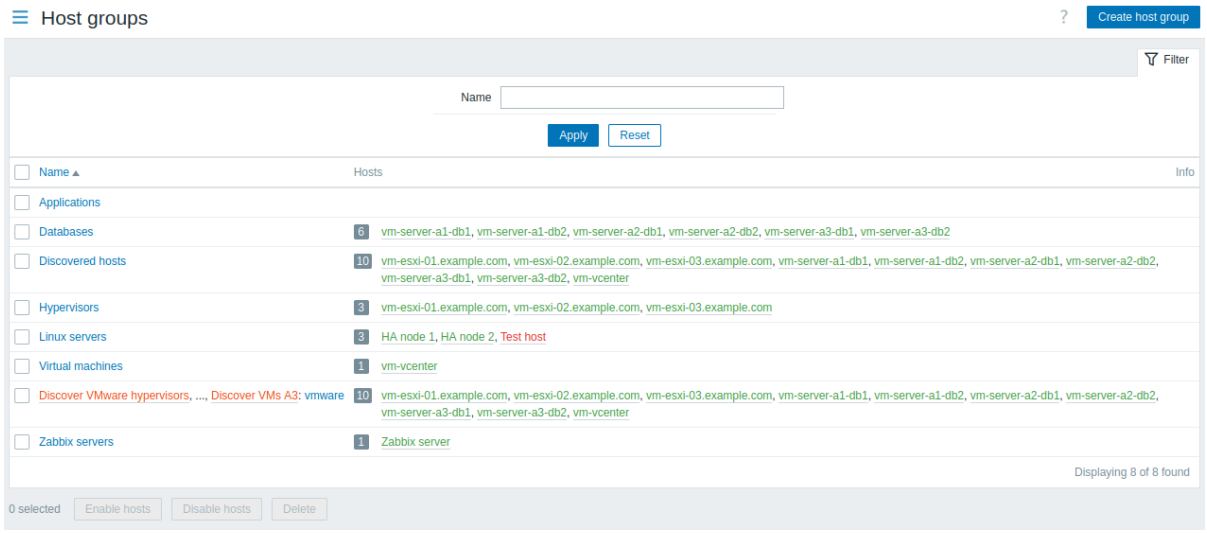
Você pode usar o filtro para exibir apenas os grupos de template que são do seu interesse. Para melhor desempenho de pesquisa, os dados são pesquisados com macros não resolvidas.

2 Grupos de hosts

Visão geral

Na seção *Coleta de dados* → *Grupos de hosts*, os usuários podem configurar e manter grupos de hosts.

Uma listagem dos grupos de hosts existentes com seus detalhes é exibida. Você pode pesquisar e filtrar grupos de hosts pelo nome.



Dados exibidos:

Coluna	Descrição
Nome	Nome do grupo de hosts. Clicar no nome do grupo abre o formulário de configuração do grupo. Grupos de hosts descobertos são exibidos com os nomes das regras de descoberta de baixo nível como prefixo. Clicar no nome da regra LLD abre o formulário de configuração do protótipo de host. Observe que grupos de hosts descobertos são excluídos quando todas as regras LLD que os descobriram são excluídas.
Hosts	Número de hosts no grupo (exibido em cinza) seguido pela lista de membros do grupo. Clicar no nome do host abrirá o formulário de configuração do host. Clicar no número, na listagem completa de hosts, filtrará apenas aqueles que pertencem ao grupo.
Info	Informações de erro (se houver) relacionadas ao grupo de hosts são exibidas.

Opções de edição em massa

Os botões abaixo da lista oferecem algumas opções de edição em massa:

- *Habilitar hosts* - altera o status de todos os hosts do grupo para "Monitorado".
- *Desabilitar hosts* - altera o status de todos os hosts do grupo para "Não monitorado".
- *Excluir* - exclui os grupos de hosts. Observe que excluir um grupo de hosts exclui apenas o grupo lógico, não os hosts do grupo. Não é possível excluir um grupo de hosts que seja o único grupo de alguns hosts.

Para usar essas opções, marque as caixas de seleção antes dos respectivos grupos de hosts e clique no botão desejado.

Usando filtro

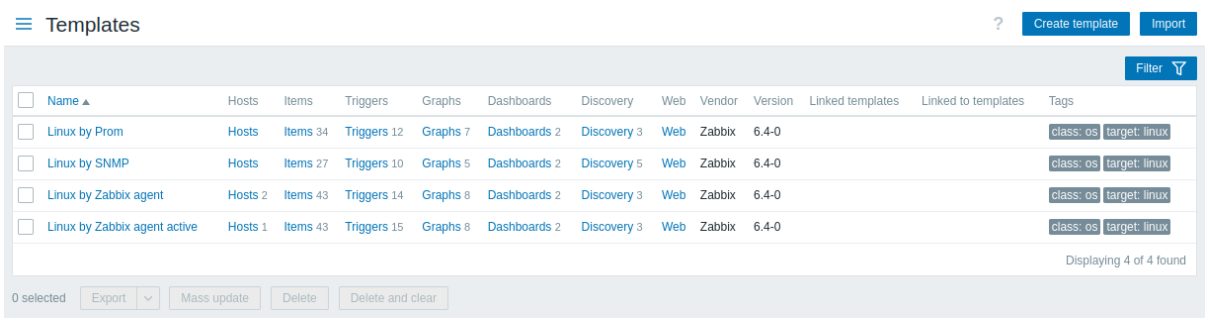
Você pode usar o filtro para exibir apenas os grupos de hosts que você está interesse. Para um melhor desempenho de pesquisa, os dados são pesquisados com macros não resolvidas.

3 Templates

Visão geral

Na seção *Coleta de dados* → *Templates*, os usuários podem configurar e manter templates.

Uma lista dos templates existentes com seus detalhes é exibida.



Dados exibidos:

Coluna	Descrição
Nome	Nome do template. Clicar no nome do template abre o formulário de configuração do template.
Hosts	Número de hosts editáveis aos quais o template está vinculado; hosts somente leitura não estão incluídos. Clicar em *Hosts* abrirá a lista de hosts filtrando apenas aqueles que estão vinculados ao template.
Entidades (Itens, Triggers, Gráficos, Dashboards, Descoberta, Web)	Número das respectivas entidades no template (exibido em cinza). Clicar no nome da entidade irá, na listagem completa dessa entidade, filtrar apenas aquelas que pertencem ao template.
Templates vinculados	Templates que estão vinculados ao template.
Vinculado a templates	Templates aos quais o template está vinculado.
Fornecedor, *Versão*	Fornecedor e versão do template; exibido se a configuração do template contiver essas informações, e apenas para templates prontos para uso, templates importados ou templates modificados através da API de Template. Para templates prontos para uso, a versão é exibida da seguinte forma: versão principal do Zabbix, delimitador (”-”), número de revisão (aumentado a cada nova versão do template e reiniciado a cada versão principal do Zabbix). Por exemplo, 6.4-0, 6.4-3, 7.0-0, 7.0-3.
Tags	Tags do template, com macros não resolvidas.

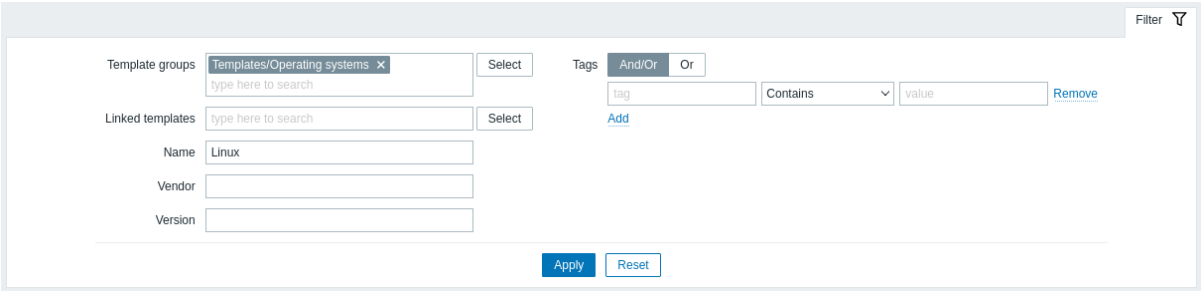
Para configurar um novo template, clique no botão *Criar template* no canto superior direito.

Para importar um template de um arquivo YAML, XML ou JSON, clique no botão *Importar* no canto superior direito.

Usando o filtro

Você pode usar o filtro para exibir apenas os templates de seu interesse. Para melhor desempenho na pesquisa, os dados são pesquisados com macros não resolvidas.

O link *Filtro* está disponível abaixo dos botões *Criar template* e *Importar*. Se você clicar nele, um filtro ficará disponível onde você pode filtrar os templates por grupo de templates, templates vinculados, nome e tags.



Parâmetro	Descrição
Grupos de templates	Filtra por um ou mais grupos de templates. Especificar um grupo de templates pai seleciona implicitamente todos os grupos aninhados.
Templates vinculados	Filtra por templates vinculados diretamente.
Nome	Filtra pelo nome do template.
Fornecedor	Filtra pelo fornecedor do template.
Versão	Filtra pela versão do template.

Parâmetro	Descrição
Tags	Filtra pelo nome e valor da tag do template. A filtragem é possível apenas por tags no nível do template (não herdadas). É possível incluir ou excluir tags e valores de tags específicos. Várias condições podem ser definidas. A correspondência do nome da tag diferencia maiúsculas de minúsculas. Existem vários operadores disponíveis para cada condição: **Existe** - inclui os nomes de tags especificados; **Igual** - inclui os nomes e valores de tags especificados (diferencia maiúsculas de minúsculas); **Contém** - inclui os nomes de tags especificados onde os valores das tags contêm a string inserida (correspondência de substring, não diferencia maiúsculas de minúsculas); **Não existe** - exclui os nomes de tags especificados; **Não é igual** - exclui os nomes e valores de tags especificados (diferencia maiúsculas de minúsculas); **Não contém** - exclui os nomes de tags especificados onde os valores das tags contêm a string inserida (correspondência de substring, não diferencia maiúsculas de minúsculas). Existem dois tipos de cálculo para as condições: **E/Ou** - todas as condições devem ser atendidas, condições com o mesmo nome de tag serão agrupadas pela condição Ou; **Ou** - basta que uma condição seja atendida.

Opções de edição em massa

Os botões abaixo da lista oferecem algumas opções de edição em massa:

- *Exportar* - exporta o template para um arquivo YAML, XML ou JSON;
- *Atualização em massa* - atualiza várias propriedades para vários templates de uma vez;
- *Excluir* - exclui o template, mas mantém suas entidades vinculadas (items, triggers, etc.) nos hosts;
- *Excluir e limpar* - exclui o template e suas entidades vinculadas dos hosts.

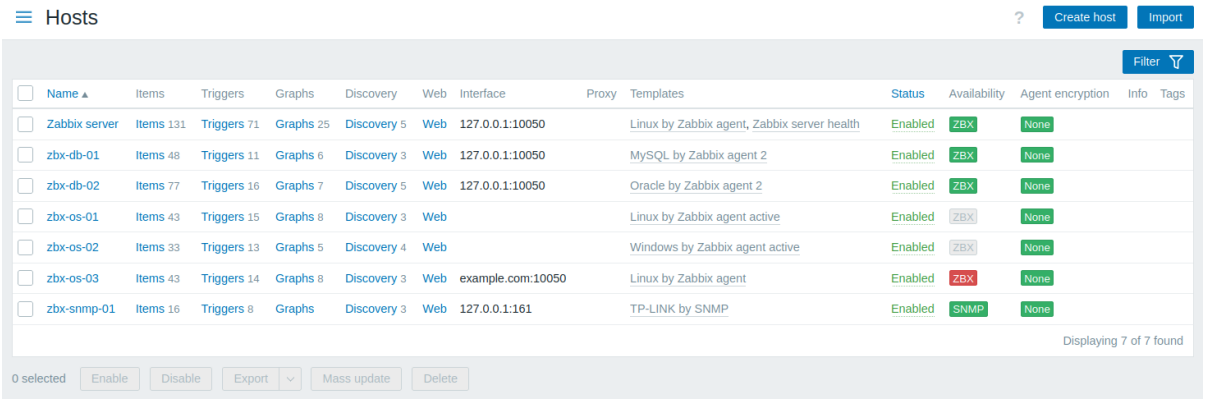
Para usar essas opções, marque as caixas de seleção antes dos respectivos templates e clique no botão desejado.

4 Hosts

Visão geral

Na seção *Coleta de dados* → *Hosts*, os usuários podem configurar e manter hosts.

Uma lista dos hosts existentes com seus detalhes é exibida.



Dados exibidos:

Coluna	Descrição
Nome	Nome do host. Clicar no nome do host abre o formulário de configuração do host. Um ícone de chave inglesa laranja após o nome do host indica que este host está em manutenção. Os detalhes da manutenção são exibidos quando o ponteiro do mouse é posicionado sobre o ícone.

Coluna	Descrição
Entidades (Itens, Triggers, Gráficos, Descoberta, Web)	Clicar no nome da entidade exibirá os items, triggers, etc., do host. O número das respectivas entidades é exibido em cinza.
Interface	A interface principal do host é exibida.
Proxy	Os proxies atribuídos são exibidos nesta coluna: <Nome do proxy> - o host é monitorado por um proxy independente (mesmo que o proxy faça parte de um grupo de proxies); <Nome do grupo de proxies: nome do proxy> - o host é monitorado por um grupo de proxies, e o Zabbix server atribuiu um proxy para monitorar o host; <Nome do grupo de proxies> - o host é monitorado por um grupo de proxies sem nenhum proxy, ou se o Zabbix server não atribuiu um proxy para monitorar o host; Nada - o host não é monitorado por proxy nem por grupo de proxies. Esta coluna só é exibida se a opção de filtro *Monitorado por* estiver definida como 'Qualquer', 'Proxy' ou 'Grupo de proxies'.
Templates	Os templates vinculados ao host são exibidos. Se outros templates estiverem contidos no template vinculado, eles serão exibidos entre parênteses, separados por vírgula. Clicar no nome de um template abrirá seu formulário de configuração.
Status	O status do host é exibido - *Habilitado* ou *Desabilitado*. Clicando no status você pode alterá-lo manualmente. Hosts descobertos que foram perdidos são marcados com um ícone de informação. O texto da dica fornece detalhes sobre seu status.

Coluna	Descrição
Disponibilidade	A disponibilidade do host por interface configurada é exibida. Os ícones de disponibilidade representam o status atual da interface do host no Zabbix server. Portanto, se você desabilitar um host no frontend, sua disponibilidade será atualizada após o Zabbix server sincronizar as alterações de configuração. Da mesma forma, se você habilitar um host, sua disponibilidade será atualizada após o Zabbix server sincronizar as alterações de configuração e consultar o host. Os ícones de disponibilidade representam apenas os tipos de interface (Agent, SNMP, IPMI, JMX) que estão configurados. Ao passar o mouse sobre o ícone, é exibido um pop-up com uma lista de todas as interfaces do mesmo tipo com detalhes, status e erros. Para a interface Agent, o pop-up exibe interfaces (passivas) e verificações ativas. Se um host tiver apenas verificações ativas, o ícone da interface Agent será exibido mesmo que o host não tenha uma interface Agent configurada. A coluna fica vazia para hosts sem interfaces. O status de uma única interface de host é determinado pela conexão entre um item habilitado usando a interface e o host. O status pode ser: **Disponível** - a conexão com o host foi bem-sucedida; **Não disponível** - a conexão com o host não foi bem-sucedida (timeout, problemas de firewall, etc.); **Desconhecido** - a conexão com o host não foi tentada ou o resultado é desconhecido. Para detalhes adicionais sobre como o Zabbix server determina o status da interface, consulte Status de interface desconhecido e Configurações de interface de host inalcançável/indisponível. O status de todas as interfaces de host de um único tipo (Agent, SNMP, IPMI, JMX) é determinado por aquelas interfaces que são usadas por pelo menos um item habilitado. O status é indicado pela cor do ícone: **Verde** - todas as interfaces estão disponíveis; **Amarelo** - pelo menos uma interface não está disponível e pelo menos uma está disponível ou desconhecida; **Vermelho** - todas as interfaces não estão disponíveis; **Cinza** - pelo menos uma interface está desconhecida, mas nenhuma está indisponível. Antes do Zabbix 7.0.10, esse status é determinado por todas as interfaces do mesmo tipo, independentemente de algum item usá-las. **Disponibilidade de verificação ativa.** Se pelo menos uma verificação ativa estiver habilitada no host, as verificações ativas também afetam a disponibilidade total da interface Agent conforme descrito acima. Para determinar a disponibilidade da verificação ativa, mensagens de heartbeat são enviadas na thread de verificação ativa do agent. A frequência das mensagens de heartbeat é controlada pelo parâmetro `HeartbeatFrequency` na configuração do Zabbix agent ou agent 2 (padrão 60 segundos, intervalo 0-3600). As verificações ativas são consideradas indisponíveis quando o heartbeat da verificação ativa é mais antigo que 2 x HeartbeatFrequency segundos. **Nota:** Zabbix agents mais antigos que a versão 6.2.x não enviam heartbeats de verificação ativa, portanto, a disponibilidade de seus hosts permanece desconhecida.
Criptografia do agent	O status da criptografia para conexões com o host é exibido: **Nenhuma** - sem criptografia; **PSK** - usando chave pré-compartilhada; **Cert** - usando certificado.
Info	Informações de erro (se houver) relacionadas ao host são exibidas.
Tags	Tags do host com macros não resolvidas.

Para configurar um novo host, clique no botão *Criar host* no canto superior direito. Para importar um host de um arquivo YAML, XML ou JSON, clique no botão *Importar* no canto superior direito.

Opções de edição em massa

Os botões abaixo da lista oferecem algumas opções de edição em massa:

- *Habilitar* - altera o status do host para *Monitorado*;
- *Desabilitar* - altera o status do host para *Não monitorado*;

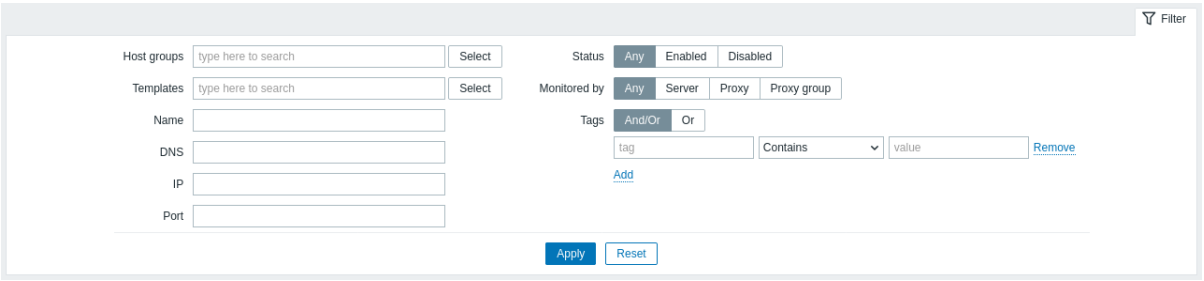
- *Exportar* - exporta os hosts para um arquivo YAML, XML ou JSON;
- *Atualização em massa* - atualiza várias propriedades para vários hosts de uma só vez;
- *Excluir* - exclui os hosts.

Para usar essas opções, marque as caixas de seleção antes dos respectivos hosts e clique no botão desejado.

Usando o filtro

Você pode usar o filtro para exibir apenas os hosts do seu interesse. Para melhor desempenho na pesquisa, os dados são pesquisados com macros não resolvidas.

O ícone *Filtro* está disponível no canto superior direito. Clicando nele, será aberto um filtro onde você pode especificar os critérios de filtragem desejados.



Parâmetro	Descrição
Grupos de hosts	Filtra por um ou mais grupos de hosts. Ao especificar um grupo de hosts pai, todos os grupos de hosts aninhados são implicitamente selecionados.
Templates	Filtra por templates vinculados.
Nome	Filtra pelo nome visível do host.
DNS	Filtra pelo nome DNS.
IP	Filtra pelo endereço IP.
Porta	Filtra pelo número da porta.
Status	Filtra pelo status do host.
Monitorado por	Filtra hosts que são monitorados pelo Zabbix server, proxy ou grupo de proxies.
Proxies	Filtra hosts que são monitorados pelos proxies especificados aqui. Este campo só está disponível se ”Proxy” for selecionado no campo *Monitorado por*.
Grupos de proxies	Filtra hosts que são monitorados pelos grupos de proxies especificados aqui. Este campo só está disponível se ”Grupo de proxies” for selecionado no campo *Monitorado por*.
Tags	Filtra pelo nome e valor da tag do host. É possível incluir ou excluir tags e valores de tags específicos. Várias condições podem ser definidas. A correspondência do nome da tag diferencia maiúsculas de minúsculas. Existem vários operadores disponíveis para cada condição: **Existe** - inclui os nomes de tags especificados; **Igual** - inclui os nomes e valores de tags especificados (diferencia maiúsculas de minúsculas); **Contém** - inclui os nomes de tags especificados onde os valores das tags contêm a string inserida (correspondência de substring, não diferencia maiúsculas de minúsculas); **Não existe** - exclui os nomes de tags especificados; **Diferente** - exclui os nomes e valores de tags especificados (diferencia maiúsculas de minúsculas); **Não contém** - exclui os nomes de tags especificados onde os valores das tags contêm a string inserida (correspondência de substring, não diferencia maiúsculas de minúsculas). Existem dois tipos de cálculo para as condições: **E/Ou** - todas as condições devem ser atendidas, condições com o mesmo nome de tag serão agrupadas pela condição Ou; **Ou** - basta que uma condição seja atendida.

Status de interface desconhecido

O Zabbix server exibe o status ”Desconhecido” para uma interface de host (Agent, SNMP, IPMI, JMX) nos seguintes casos:

- O host está desabilitado.
- O host está configurado para ser monitorado por um proxy, por um proxy diferente ou pelo server se anteriormente era monitorado por um proxy.

- O host é monitorado por um proxy que parece estar offline (nenhuma atualização recebida do proxy durante o intervalo máximo de heartbeat - 1 hora).
- Todos os items do host com esse tipo de interface estão desabilitados.
- Nenhum poller para esse tipo de interface foi configurado (por exemplo, o parâmetro de configuração do server `StartAgentPollers` ou `StartSNMPPollers` está definido como 0).

A disponibilidade da interface é definida como ”Desconhecido” após a sincronização do cache de configuração do Zabbix server.

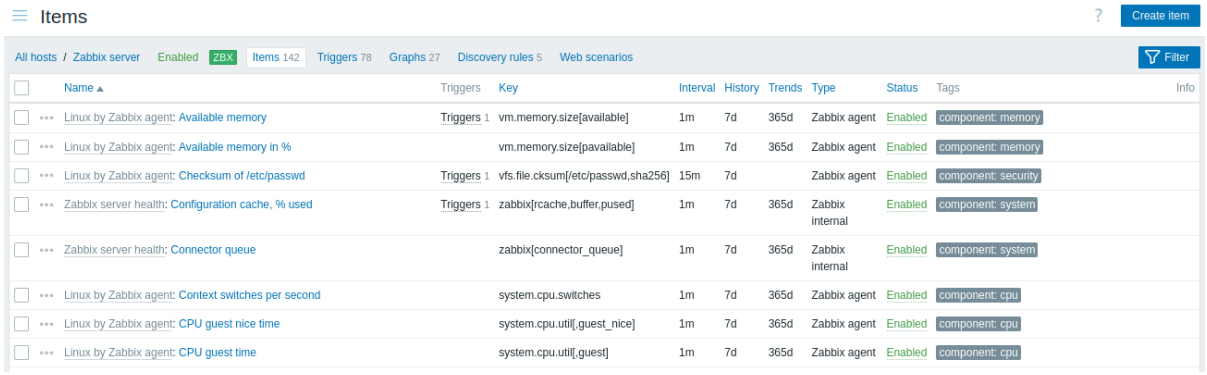
A disponibilidade da interface (Disponível/Não disponível) em hosts monitorados por proxies é restaurada após a sincronização do cache de configuração do proxy.

1 Items

Visão geral

A lista de items de um host pode ser acessada em *Coleta de dados* → *Hosts* clicando em *Items* para o respectivo host.

Uma lista de items existentes é exibida.



Dados exibidos:

Coluna	Descrição
Menu do item	Clique no ícone de três pontos para abrir o menu do item.
Host	Host do item. Desde o Zabbix 7.0.4, clicar no nome do host abre o formulário de configuração do host. Esta coluna é exibida apenas se vários hosts ou nenhum host estiverem selecionados no filtro.
Nome	Nome do item exibido como um link azul para os detalhes do item. Clicar no link do nome do item abre o formulário de configuração do item. Se o item do host pertencer a um template, o nome do template é exibido antes do nome do item como um link cinza. Clicar no link do template abrirá a lista de items no nível do template. Se o item foi criado a partir de um protótipo de item, seu nome é precedido pelo nome da regra de descoberta de baixo nível, em laranja. Clicar no nome da regra de descoberta abrirá a lista de protótipos de item.
Triggers	Ao passar o mouse sobre Triggers, será exibida uma infobox mostrando os triggers associados ao item. O número de triggers é exibido em cinza.
Chave	A chave do item é exibida.
Intervalo	A frequência da checagem é exibida. *Nota* que items passivos também podem ser verificados imediatamente pressionando o botão Executar agora.
Histórico	Quantos dias o histórico de dados do item será mantido é exibido.
Tendências	Quantos dias o histórico de tendências do item será mantido é exibido.
Tipo	O tipo do item é exibido (Zabbix agent, SNMP agent, checagem simples, etc).
Status	O status do item é exibido - *Habilitado*, *Desabilitado* ou *Não suportado*. Você pode alterar o status manualmente clicando nele - de Habilitado para Desabilitado (e vice-versa); de Não suportado para Desabilitado (e vice-versa). Items descobertos que foram perdidos são marcados com um ícone de informação. O texto da dica fornece detalhes sobre seu status.
Tags	As tags do item são exibidas. Até três tags (pares nome:valor) podem ser exibidas. Se houver mais tags, um link ”...” é exibido que permite ver todas as tags ao passar o mouse.

Coluna	Descrição
Info	Se o item estiver funcionando corretamente, nenhum ícone é exibido nesta coluna. Em caso de erros, um ícone quadrado com a letra "i" é exibido. Passe o mouse sobre o ícone para ver uma dica com a descrição do erro.

Para configurar um novo item, clique no botão *Criar item* no canto superior direito.

Opções de edição em massa

Os botões abaixo da lista oferecem algumas opções de edição em massa:

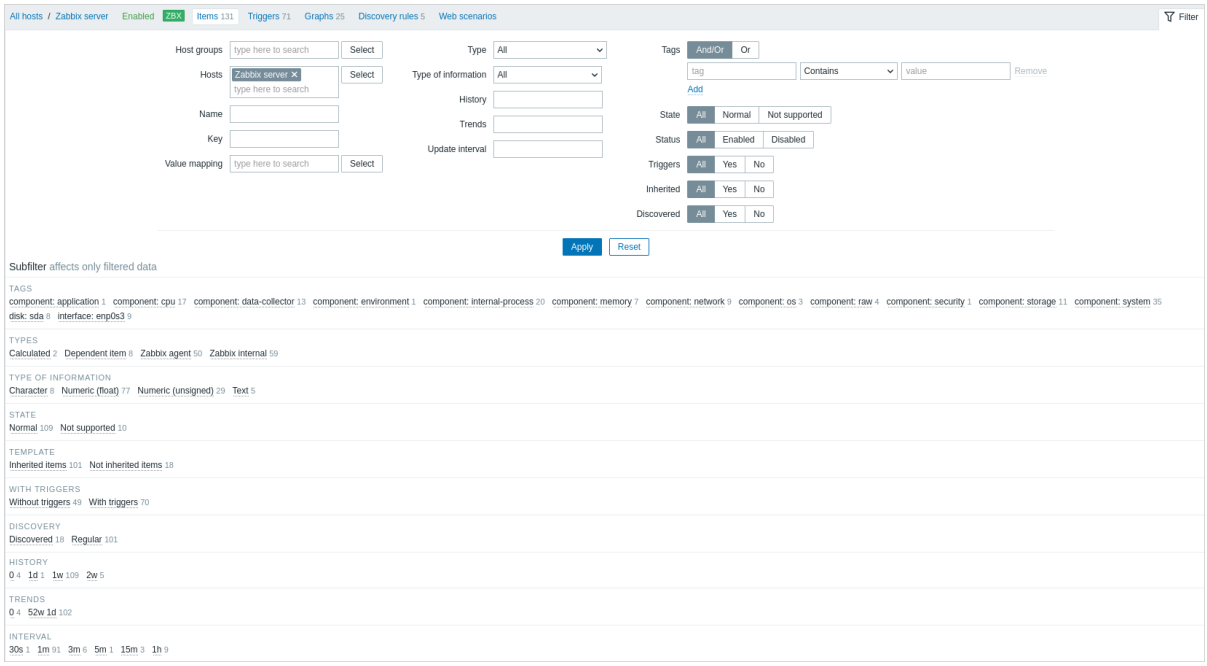
- *Habilitar* - altera o status do item para *Habilitado*
- *Desabilitar* - altera o status do item para *Desabilitado*
- *Executar agora* - executa uma checagem para novos valores do item imediatamente. Suportado apenas para checagens **passivas** (veja mais detalhes). Observe que, ao verificar os valores imediatamente, o cache de configuração não é atualizado, portanto os valores não refletirão alterações muito recentes na configuração do item.
- *Limpar histórico e tendências* - exclui dados de histórico e tendências dos itens.
- *Copiar* - copia os itens para outros hosts ou templates.
- *Atualização em massa* - atualiza várias propriedades de vários itens de uma só vez.
- *Excluir* - exclui os itens.

Para usar essas opções, marque as caixas de seleção antes dos respectivos itens e clique no botão desejado.

Usando o filtro

Você pode usar o filtro para exibir apenas os items de seu interesse. Para melhor desempenho na pesquisa, os dados são pesquisados com macros não resolvidas.

O ícone *Filtro* está disponível no canto superior direito. Clicando nele, será aberto um filtro onde você pode especificar os critérios de filtragem desejados.



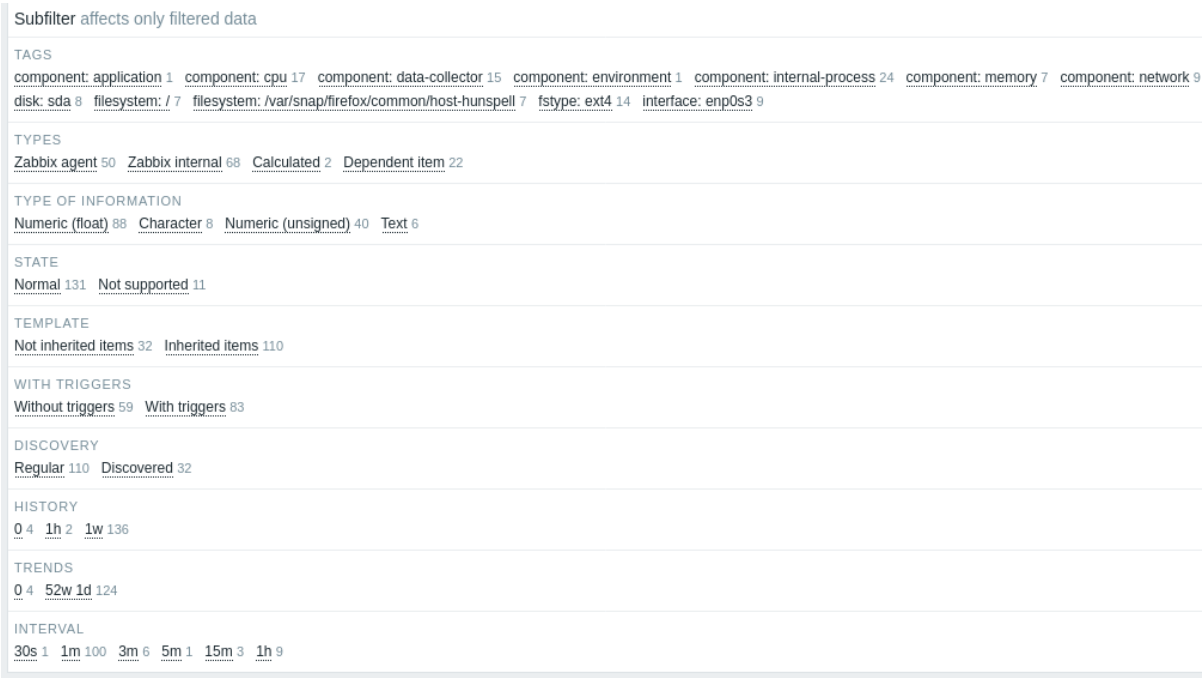
Parâmetro	Descrição
Grupos de hosts	Filtrar por um ou mais grupos de hosts. Especificar um grupo de hosts pai seleciona implicitamente todos os grupos de hosts aninhados. Grupos de hosts contendo apenas templates não podem ser selecionados.
Hosts	Filtrar por um ou mais hosts.
Nome	Filtrar por nome do item.
Chave	Filtrar por chave do item.
Mapeamento de valores	Filtrar pelo mapa de valores utilizado. Este parâmetro não é exibido se a opção *Hosts* estiver vazia.
Tipo	Filtrar por tipo de item (Zabbix agent, SNMP agent, etc.).
Tipo de informação	Filtrar por tipo de informação (Numérico sem sinal, ponto flutuante, etc.).
Histórico	Filtrar por quanto tempo o histórico do item é mantido.

Parâmetro	Descrição
Tendências	Filtrar por quanto tempo as tendências do item são mantidas.
Intervalo de atualização	Filtrar por intervalo de atualização do item.
Tags	Especifique tags para limitar o número de items exibidos. É possível incluir ou excluir tags e valores de tags específicos. Várias condições podem ser definidas. A correspondência do nome da tag diferencia maiúsculas de minúsculas. Existem vários operadores disponíveis para cada condição: **Existe** - inclui os nomes de tags especificados **Igual** - inclui os nomes e valores de tags especificados (diferencia maiúsculas de minúsculas) **Contém** - inclui os nomes de tags especificados onde os valores das tags contêm a string inserida (correspondência de substring, não diferencia maiúsculas de minúsculas) **Não existe** - exclui os nomes de tags especificados **Diferente** - exclui os nomes e valores de tags especificados (diferencia maiúsculas de minúsculas) **Não contém** - exclui os nomes de tags especificados onde os valores das tags contêm a string inserida (correspondência de substring, não diferencia maiúsculas de minúsculas) Existem dois tipos de cálculo para as condições: **E/Ou** - todas as condições devem ser atendidas, condições com o mesmo nome de tag serão agrupadas pela condição Ou **Ou** - basta que uma condição seja atendida
Estado	Filtrar por estado do item - *Normal* ou *Não suportado*.
Status	Filtrar por status do item - *Habilitado* ou *Desabilitado*.
Triggers	Filtrar items com (ou sem) triggers.
Herdado	Filtrar items herdados (ou não herdados) de um template.
Descoberta	Filtrar items descobertos (ou não descobertos) por descoberta de baixo nível.

Usando subfiltro

O subfiltro permite modificar ainda mais o filtro do filtro principal.

Ele contém links clicáveis para acesso rápido aos itens relacionados. Os itens são relacionados por uma entidade comum - tag, tipo de item, estado do item, intervalo de atualização do item, etc. Quando uma entidade é clicada, ela é destacada com um fundo cinza e os itens são imediatamente filtrados (não é necessário clicar em *Aplicar* no filtro principal). Clicar em outra entidade a adiciona aos resultados filtrados. Clicar novamente na entidade remove o filtro.



Os subfiltros são gerados com base nos dados filtrados, que são limitados a 1000 registros. Se você quiser ver mais registros no subfiltro, precisa aumentar o valor do parâmetro *Limite para resultados de pesquisa e filtro* (em *Administração -> Geral -> GUI*).

Ao contrário do filtro principal, o subfiltro é atualizado a cada solicitação de atualização da tabela para sempre ter informações atualizadas sobre as opções de filtragem disponíveis e seus números de contador.

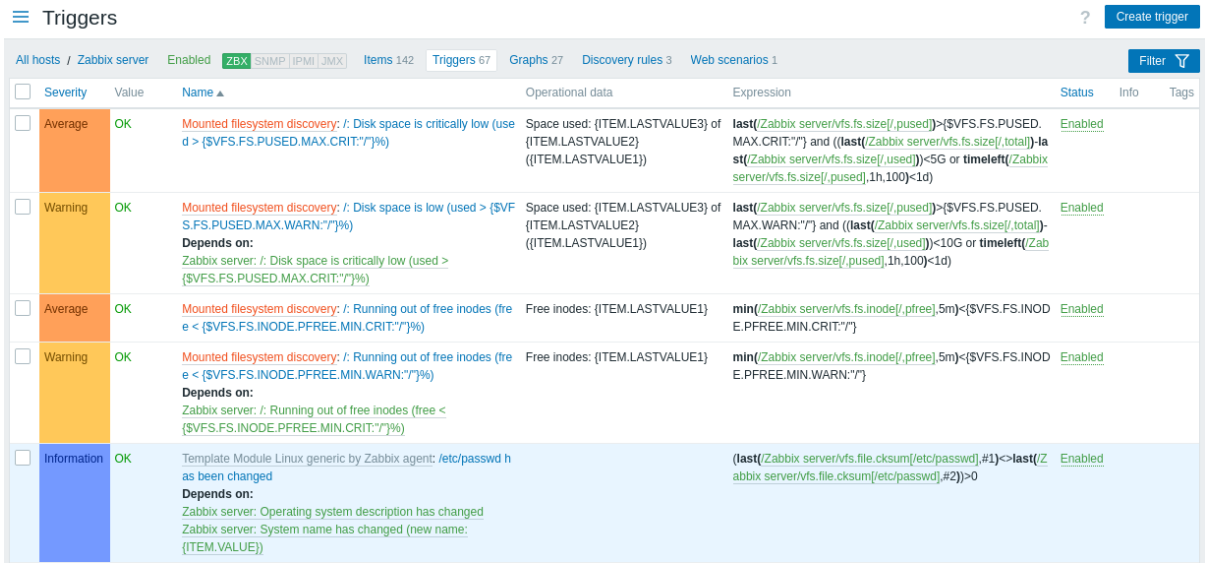
O número de entidades exibidas é limitado a 100 horizontalmente. Se houver mais, um ícone de reticências é exibido no final; ele não é clicável.

Um número ao lado de cada entidade clicável indica o número de itens agrupados nela (com base nos resultados do filtro principal). Quando uma entidade é clicada, os números com outras entidades disponíveis são exibidos com um sinal de mais indicando quantos itens podem ser adicionados à seleção atual. Entidades sem itens não são exibidas, a menos que tenham sido selecionadas anteriormente no subfiltro.

2 Triggers

Visão geral

A lista de triggers de um host pode ser acessada em *Coleta de dados* → *Hosts* clicando em *Triggers* para o respectivo host.



Dados exibidos:

Coluna	Descrição
Severidade	A severidade do trigger é exibida tanto pelo nome quanto pela cor de fundo da célula.
Valor	O valor do trigger é exibido: **OK** - o trigger está no estado OK **PROBLEMA** - o trigger está no estado de Problema
Host	Host do trigger. Desde o Zabbix 7.0.4, clicar no nome do host abre o formulário de configuração do host. Esta coluna é exibida apenas se vários hosts ou nenhum host estiverem selecionados no filtro.
Nome	Nome do trigger, exibido como um link azul para os detalhes do trigger. Clicar no link do nome do trigger abre o formulário de configuração do trigger. Se o trigger do host pertencer a um template, o nome do template é exibido antes do nome do trigger como um link cinza. Clicar no link do template abre a lista de triggers no nível do template. Se o trigger foi criado a partir de um protótipo de trigger, seu nome é precedido pelo nome da regra de descoberta de baixo nível, em laranja. Clicar no nome da regra de descoberta abrirá a lista de protótipos de trigger.
Dados operacionais	Definição de dados operacionais do trigger, contendo strings arbitrárias e macros que serão resolvidas dinamicamente em *Monitoramento* → *Problemas*.
Expressão	A expressão do trigger é exibida. A parte host-item da expressão é exibida como um link, levando ao formulário de configuração do item.
Status	O status do trigger é exibido - *Habilitado*, *Desabilitado* ou *Desconhecido*. Ao clicar no status, você pode alterá-lo manualmente - de Habilitado para Desabilitado (e vice-versa); de Desconhecido para Desabilitado (e vice-versa). Problemas de um trigger desabilitado não são mais exibidos no frontend, mas não são excluídos. Triggers descobertos que foram perdidos são marcados com um ícone de informação. O texto da dica fornece detalhes sobre seu status.
Info	Se tudo estiver funcionando corretamente, nenhum ícone é exibido nesta coluna. Em caso de erros, um ícone quadrado com a letra ”i” é exibido. Passe o mouse sobre o ícone para ver uma dica com a descrição do erro.
Tags	Se um trigger contiver tags, o nome e o valor da tag são exibidos nesta coluna.

Para configurar um novo trigger, clique no botão *Criar trigger* no canto superior direito.

Opções de edição em massa

Os botões abaixo da lista oferecem algumas opções de edição em massa:

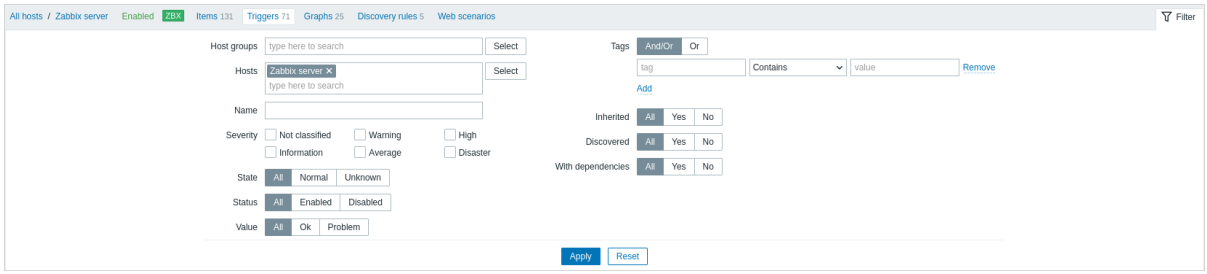
- *Ativar* - altera o status do gatilho para *Ativado*
- *Desativar* - altera o status do gatilho para *Desativado*
- *Copiar* - copia os gatilhos para outros hosts ou modelos
- *Atualização em massa* - atualiza várias propriedades para vários gatilhos de uma vez só
- *Excluir* - exclui os gatilhos

Para usar essas opções, marque as caixas de seleção antes do respectivo acionadores e, em seguida, clique no botão necessário.

Usando o filtro

Você pode usar o filtro para exibir apenas os triggers que lhe interessam. Para melhor desempenho na pesquisa, os dados são pesquisados com macros não resolvidas.

O ícone *Filtro* está disponível no canto superior direito. Clicando nele será aberto um filtro onde você pode especificar os critérios de filtragem desejados.



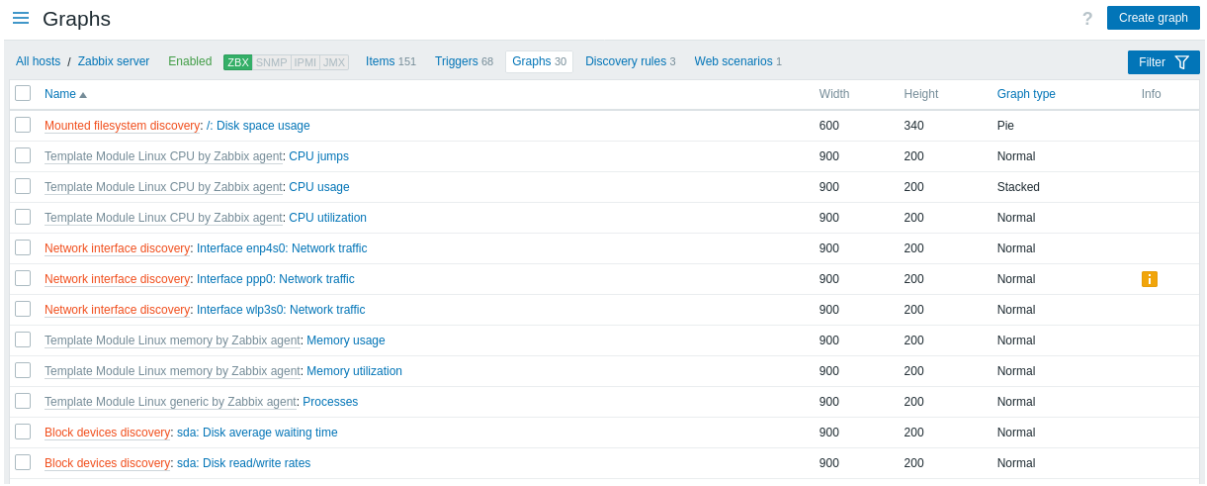
Parâmetro	Descrição
Grupos de hosts	Filtra por um ou mais grupos de hosts. Ao especificar um grupo de hosts pai, todos os grupos de hosts aninhados são implicitamente selecionados. Grupos de hosts contendo apenas templates não podem ser selecionados.
Hosts	Filtra por um ou mais hosts. Se grupos de hosts já estiverem selecionados acima, a seleção de hosts será limitada a esses grupos.
Nome	Filtra pelo nome do trigger.
Severidade	Selecione para filtrar por uma ou várias severidades de trigger.
Estado	Filtra pelo estado do trigger.
Status	Filtra pelo status do trigger.
Valor	Filtra pelo valor do trigger.
Tags	Filtra pelo nome e valor da tag do trigger. É possível incluir ou excluir tags e valores de tags específicos. Várias condições podem ser definidas. A correspondência do nome da tag diferencia maiúsculas de minúsculas. Há vários operadores disponíveis para cada condição: **Existe** - inclui os nomes de tags especificados **Igual** - inclui os nomes e valores de tags especificados (diferencia maiúsculas de minúsculas) **Contém** - inclui os nomes de tags especificados onde os valores das tags contêm a string inserida (correspondência de substring, não diferencia maiúsculas de minúsculas) **Não existe** - exclui os nomes de tags especificados **Não é igual** - exclui os nomes e valores de tags especificados (diferencia maiúsculas de minúsculas) **Não contém** - exclui os nomes de tags especificados onde os valores das tags contêm a string inserida (correspondência de substring, não diferencia maiúsculas de minúsculas) Existem dois tipos de cálculo para as condições: **E/Ou** - todas as condições devem ser atendidas, condições com o mesmo nome de tag serão agrupadas pela condição Ou **Ou** - basta que uma condição seja atendida Macros e funções de macro são suportadas tanto nos campos de nome quanto de valor da tag.
Herdado	Filtra triggers herdados (ou não herdados) de um template.
Descoberto	Filtra triggers descobertos (ou não descobertos) por descoberta de baixo nível.
Com dependências	Filtra triggers com (ou sem) dependências.

3 Gráficos

Visão geral

A lista de gráficos personalizados para um host pode ser acessada em *Coleta de dados* → *Hosts* clicando em *Gráficos* para o respectivo host.

Uma lista de gráficos existentes é exibida.



Dados exibidos:

Coluna	Descrição
Host	Host do gráfico. Desde o Zabbix 7.0.4, clicar no nome do host abre o formulário de configuração do host. Esta coluna é exibida apenas se vários hosts ou nenhum host estiverem selecionados no filtro.
Nome	Nome do gráfico personalizado, exibido como um link azul para os detalhes do gráfico. Clicar no link do nome do gráfico abre o formulário de configuração do gráfico. Se o gráfico do host pertencer a um template, o nome do template será exibido antes do nome do gráfico como um link cinza. Clicar no link do template abre a lista de gráficos no nível do template. Se o gráfico foi criado a partir de um protótipo de gráfico, seu nome é precedido pelo nome da regra de descoberta de baixo nível, em laranja. Clicar no nome da regra de descoberta abrirá a lista de protótipos de gráficos.
Largura	A largura do gráfico é exibida.
Altura	A altura do gráfico é exibida.
Tipo de gráfico	O tipo de gráfico é exibido - *Normal*, *Empilhado*, *Pizza* ou *Explodido*.
Info	Se o gráfico estiver funcionando corretamente, nenhum ícone será exibido nesta coluna. Em caso de erros, um ícone quadrado com a letra ”i” é exibido. Passe o mouse sobre o ícone para ver uma dica de ferramenta com a descrição do erro.

Para configurar um novo gráfico, clique no botão *Criar gráfico* no canto superior direito.

Opções de edição em massa

Os botões abaixo da lista oferecem algumas opções de edição em massa:

- *Copiar* - copie os gráficos para outros hosts ou modelos
- *Excluir* - excluir os gráficos

Para usar essas opções, marque as caixas de seleção antes dos respectivos gráficos, em seguida, clique no botão desejado.

Usando filtro

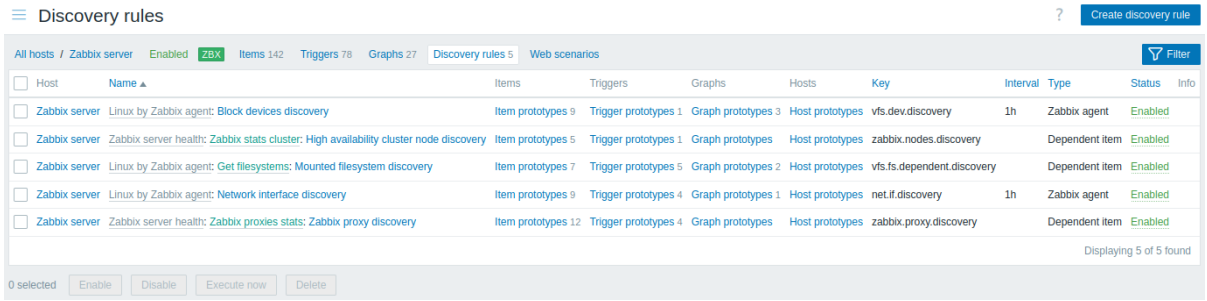
Você pode filtrar gráficos por grupo de hosts e host. Para melhor desempenho de pesquisa, os dados são pesquisados com macros não resolvidas.

4 Regras de descoberta

Visão geral

A lista de regras de descoberta de baixo nível para um host pode ser acessada em *Coleta de dados* → *Hosts* clicando em *Descoberta* para o respectivo host.

Uma lista das regras de descoberta de baixo nível existentes é exibida. Também é possível ver todas as regras de descoberta independentemente do host, ou todas as regras de descoberta de um grupo de hosts específico alterando as configurações do filtro.



Dados exibidos:

Coluna	Descrição
Host	O nome visível do host é exibido. Desde o Zabbix 7.0.4, clicar no nome do host abre o formulário de configuração do host. Na ausência de um nome visível do host, o nome técnico do host é exibido.
Nome	Nome da regra, exibido como um link azul. Clicar no nome da regra abre o formulário de configuração da regra de descoberta de baixo nível. Se a regra de descoberta pertence a um template, o nome do template é exibido antes do nome da regra como um link cinza. Clicar no link do template abre a lista de regras no nível do template.
Itens	Um link para a lista de protótipos de item é exibido. O número de protótipos de item existentes é exibido em cinza.
Triggers	Um link para a lista de protótipos de trigger é exibido. O número de protótipos de trigger existentes é exibido em cinza.
Gráficos	Um link para a lista de protótipos de gráfico é exibido. O número de protótipos de gráfico existentes é exibido em cinza.
Hosts	Um link para a lista de protótipos de host é exibido. O número de protótipos de host existentes é exibido em cinza.
Chave	A chave do item usada para descoberta é exibida.
Intervalo	A frequência de execução da descoberta é exibida. *Nota* que a descoberta também pode ser executada imediatamente pressionando o botão *Executar agora* abaixo da lista.
Tipo	O tipo de item usado para descoberta é exibido (Zabbix agent, SNMP agent, etc).
Status	O status da regra de descoberta é exibido - *Habilitado*, *Desabilitado* ou *Não suportado*. Clicando no status você pode alterá-lo - de Habilitado para Desabilitado (e vice-versa); de Não suportado para Desabilitado (e vice-versa).
Info	Se tudo estiver correto, nenhum ícone é exibido nesta coluna. Em caso de erros, um ícone quadrado com a letra "i" é exibido. Passe o mouse sobre o ícone para ver uma dica de ferramenta com a descrição do erro.

Para configurar uma nova regra de descoberta de baixo nível, clique no botão *Criar regra de descoberta* no canto superior direito.

Opções de edição em massa

Os botões abaixo da lista oferecem algumas opções de edição em massa:

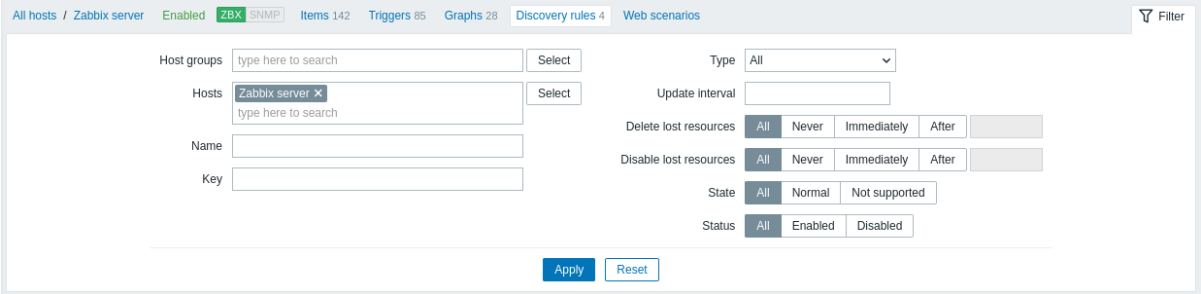
- *Habilitar* - altera o status da regra de descoberta de baixo nível para *Habilitado*.
- *Desabilitar* - altera o status da regra de descoberta de baixo nível para *Desabilitado*.
- *Executar agora* - executa a descoberta com base nas regras de descoberta imediatamente. Veja mais detalhes. Observe que, ao executar a descoberta imediatamente, o cache de configuração não é atualizado, portanto, o resultado não refletirá alterações muito recentes na configuração da regra de descoberta.
- *Excluir* - exclui as regras de descoberta de baixo nível.

Para usar essas opções, marque as caixas de seleção antes das respectivas regras de descoberta e clique no botão desejado.

Usando o filtro

Você pode usar o filtro para exibir apenas as regras de descoberta que são do seu interesse. Para melhor desempenho na pesquisa, os dados são pesquisados com macros não resolvidas.

O link *Filtro* está disponível acima da lista de regras de descoberta. Se você clicar nele, um filtro ficará disponível onde você pode filtrar as regras de descoberta por grupo de hosts, host, nome, chave do item, tipo de item e outros parâmetros.

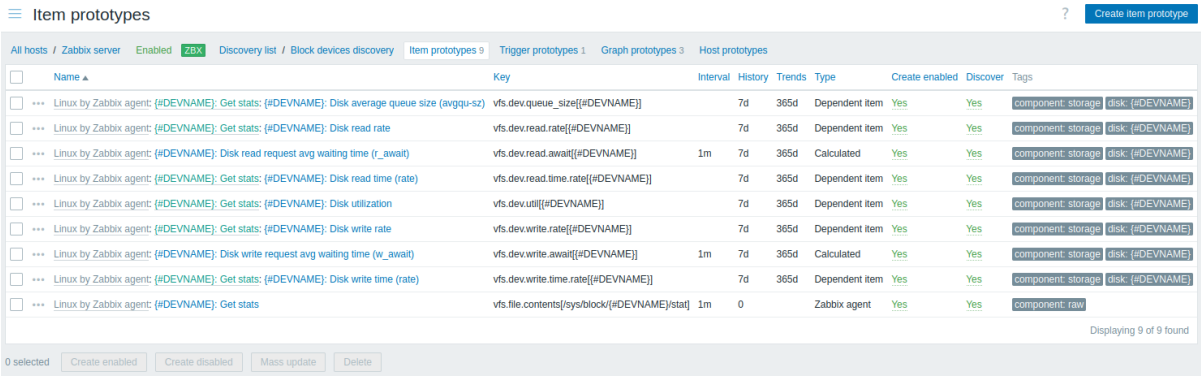


Parâmetro	Descrição
Grupos de hosts	Filtrar por um ou mais grupos de hosts. Especificar um grupo de hosts pai seleciona implicitamente todos os grupos de hosts aninhados.
Hosts	Filtrar por um ou mais hosts.
Nome	Filtrar por nome da regra de descoberta.
Chave	Filtrar por chave do item de descoberta.
Tipo	Filtrar por tipo de item de descoberta.
Intervalo de atualização	Filtrar por intervalo de atualização. Não disponível para Zabbix trapper e itens dependentes.
Excluir recursos perdidos	Filtrar por período de *Excluir recursos perdidos*.
Desabilitar recursos perdidos	Filtrar por período de *Desabilitar recursos perdidos*.
SNMP OID	Filtrar por SNMP OID. Disponível apenas se *SNMP agent* for selecionado como tipo.
Estado	Filtrar por estado da regra de descoberta (Todos/Normal/Não suportado).
Status	Filtrar por status da regra de descoberta (Todos/Habilitado/Desabilitado).

1 Protótipos de item

Visão geral

Nesta seção, os protótipos de item de uma regra de descoberta de baixo nível no host são exibidos. Os protótipos de item são a base dos itens reais do host que são criados durante a descoberta de baixo nível.



Dados exibidos:

Coluna	Descrição
Nome	Nome do protótipo de item, exibido como um link azul. Clicar no nome abre o formulário de configuração do protótipo de item. Se o protótipo de item pertence a um template, o nome do template é exibido antes do nome da regra como um link cinza. Clicar no link do template abre a lista de protótipos de item no nível do template. Se o protótipo de item for do tipo ”Item dependente”, o nome do item mestre é exibido antes do nome do protótipo de item em verde.
Chave	A chave do protótipo de item é exibida.
Intervalo	A frequência da verificação é exibida.
Histórico	Quantos dias manter o histórico de dados do item é exibido.
Tendências	Quantos dias manter o histórico de tendências do item é exibido.
Tipo	O tipo do protótipo de item é exibido (Zabbix agent, SNMP agent, verificação simples, etc).
Criar habilitado	Criar o item com base neste protótipo como: **Sim** - habilitado **Não** - desabilitado. Você pode alternar entre ’Sim’ e ’Não’ clicando neles.
Descobrir	Descobrir o item com base neste protótipo: **Sim** - descobrir **Não** - não descobrir. Você pode alternar entre ’Sim’ e ’Não’ clicando neles.
Tags	As tags do protótipo de item são exibidas.

Para configurar um novo protótipo de item, clique no botão *Criar protótipo de item* no canto superior direito.

Opções de edição em massa

Os botões abaixo da lista oferecem algumas opções de edição em massa:

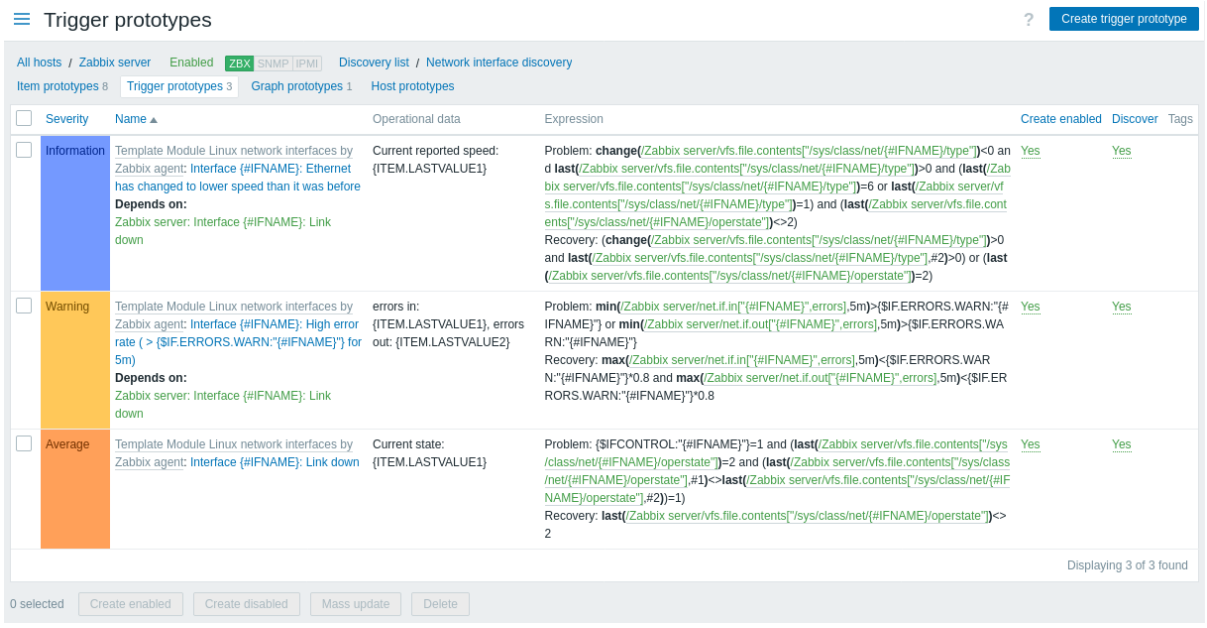
- *Criar habilitado* - cria esses items como *Habilitado*
- *Criar desabilitado* - cria esses items como *Desabilitado*
- *Atualização em massa* - atualiza em massa esses protótipos de item
- *Excluir* - exclui esses protótipos de item

Para usar essas opções, marque as caixas de seleção antes dos respectivos protótipos de item e clique no botão desejado.

2 Protótipos de trigger

Visão geral

Nesta seção, os protótipos de trigger de uma regra de descoberta de baixo nível no host são exibidos. Os protótipos de trigger são a base dos triggers reais do host que são criados durante a descoberta de baixo nível.



Dados exibidos:

Coluna	Descrição
Nome	Nome do protótipo de trigger, exibido como um link azul. Clicar no nome abre o formulário de configuração do protótipo de trigger. Se o protótipo de trigger pertencer a um template vinculado, o nome do template será exibido antes do nome do trigger como um link cinza. Clicar no link do template abre a lista de protótipos de trigger no nível do template vinculado.
Dados operacionais	O formato dos dados operacionais do trigger é exibido, contendo strings arbitrárias e macros que serão resolvidas dinamicamente em *Monitoramento* → *Problemas*.
Criar habilitado	Criar o trigger com base neste protótipo como: **Sim** - habilitado **Não** - desabilitado. Você pode alternar entre 'Sim' e 'Não' clicando neles.
Descobrir	Descobrir o trigger com base neste protótipo: **Sim** - descobrir **Não** - não descobrir. Você pode alternar entre 'Sim' e 'Não' clicando neles.
Tags	As tags do protótipo de trigger são exibidas.

Para configurar um novo protótipo de trigger, clique no botão *Criar protótipo de trigger* no canto superior direito.

Opções de edição em massa

Os botões abaixo da lista oferecem algumas opções de edição em massa:

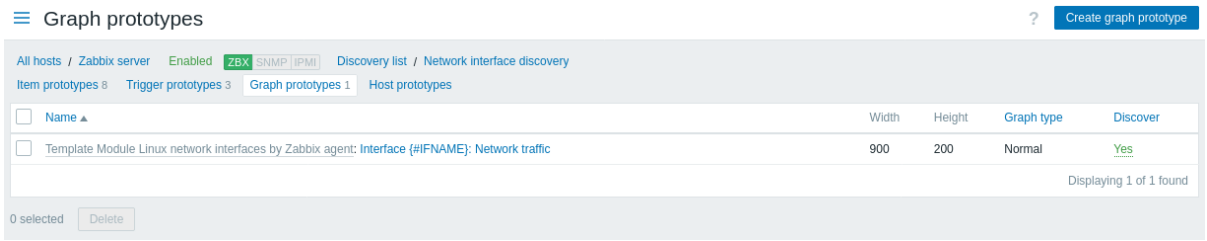
- *Criar habilitado* - criar esses triggers como *Habilitado*
- *Criar desabilitado* - criar esses triggers como *Desabilitado*
- *Atualização em massa* - atualizar em massa esses protótipos de trigger
- *Excluir* - excluir esses protótipos de trigger

Para usar essas opções, marque as caixas de seleção antes dos respectivos protótipos de trigger e clique no botão desejado.

3 Protótipos de gráficos

Visão geral

Nesta seção, os protótipos de gráficos de uma regra de descoberta de baixo nível no host são exibidos. Os protótipos de gráficos são a base dos gráficos reais do host que são criados durante a descoberta de baixo nível.



Dados exibidos:

Coluna	Descrição
Nome	Nome do protótipo de gráfico, exibido como um link azul. Clicar no nome abre o formulário de configuração do protótipo de gráfico. Se o protótipo de gráfico pertencer a um template vinculado, o nome do template será exibido antes do nome do gráfico como um link cinza. Clicar no link do template abre a lista de protótipos de gráficos no nível do template vinculado.
Largura	A largura do protótipo de gráfico é exibida.
Altura	A altura do protótipo de gráfico é exibida.
Tipo	O tipo do protótipo de gráfico é exibido - *Normal*, *Empilhado*, *Pizza* ou *Explodido*.
Descobrir	Descobrir o gráfico com base neste protótipo: **Sim** - descobrir **Não** - não descobrir. Você pode alternar entre 'Sim' e 'Não' clicando neles.

Para configurar um novo protótipo de gráfico, clique no botão *Criar protótipo de gráfico* no canto superior direito.

Opções de edição em massa

Os botões abaixo da lista oferecem algumas opções de edição em massa:

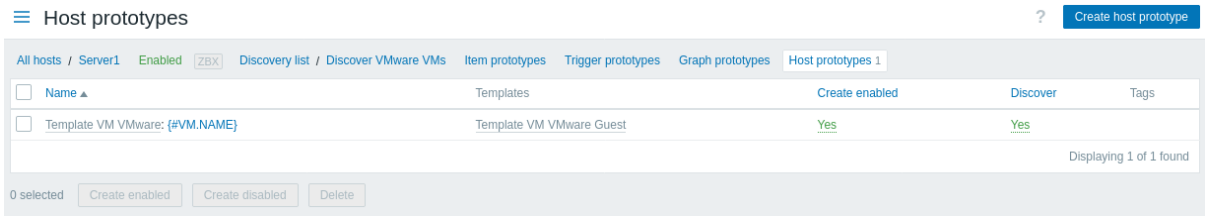
- *Excluir* - exclui esses protótipos de gráficos

Para usar essas opções, marque as caixas de seleção antes dos respectivos protótipos de gráficos e clique no botão desejado.

4 Protótipos de host

Visão geral

Nesta seção, os protótipos de host de uma regra de descoberta de baixo nível no host são exibidos. Os protótipos de host são a base dos hosts reais que são criados durante a descoberta de baixo nível.



Dados exibidos:

Coluna	Descrição
Nome	Nome do protótipo de host, exibido como um link azul. Clicar no nome abre o formulário de configuração do protótipo de host. Se o protótipo de host pertencer a um template vinculado, o nome do template será exibido antes do nome do host como um link cinza. Clicar no link do template abre a lista de protótipos de host no nível do template vinculado.
Templates	Os templates do protótipo de host são exibidos.
Criação ativada	Crie o host com base neste protótipo como: **Sim** - ativado **Não** - desativado. Você pode alternar entre 'Sim' e 'Não' clicando neles.
Descobrir	Descubra o host com base neste protótipo: **Sim** - descobrir **Não** - não descobrir. Você pode alternar entre 'Sim' e 'Não' clicando neles.
Tags	As tags do protótipo de host são exibidas.

Para configurar um novo protótipo de host, clique no botão *Criar protótipo de host* no canto superior direito.

Opções de edição em massa

Os botões abaixo da lista oferecem algumas opções de edição em massa:

- *Criar habilitado* - cria esses hosts como *Habilitado*
- *Criar desabilitado* - cria esses hosts como *Desabilitado*
- *Excluir* - exclui esses protótipos de host

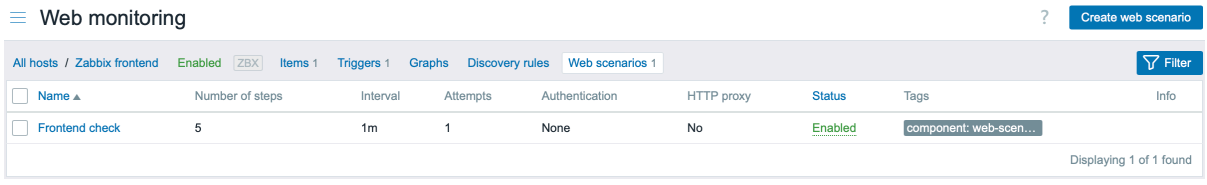
Para usar essas opções, marque as caixas de seleção antes dos respectivos protótipos de host e clique no botão desejado.

5 Cenários Web

Visão geral

A lista de cenários web de um host pode ser acessada em *Coleta de dados* → *Hosts* clicando em *Web* para o respectivo host.

Uma lista de cenários web existentes é exibida.



Dados exibidos:

Coluna	Descrição
Nome	Nome do cenário web. Clicar no nome do cenário web abre o formulário de configuração do cenário web. Se o cenário web do host pertence a um template, o nome do template é exibido antes do nome do cenário web como um link cinza. Clicar no link do template abrirá a lista de cenários web no nível do template.
Número de etapas	O número de etapas que o cenário contém.
Intervalo de atualização	Com que frequência o cenário é executado.
Tentativas	Quantas tentativas de execução das etapas do cenário web são realizadas.
Autenticação	O método de autenticação é exibido - Basic, NTLM, Kerberos, Digest ou Nenhum.
Proxy HTTP	Exibe o proxy HTTP ou 'Não' se não for usado.
Status	O status do cenário web é exibido - *Habilitado* ou *Desabilitado*. Clicando no status você pode alterá-lo.
Tags	As tags do cenário web são exibidas. Até três tags (pares nome:valor) podem ser exibidas. Se houver mais tags, um link "..." é exibido, permitindo ver todas as tags ao passar o mouse.
Info	Se tudo estiver funcionando corretamente, nenhum ícone é exibido nesta coluna. Em caso de erros, um ícone quadrado com a letra "i" é exibido. Passe o mouse sobre o ícone para ver uma dica de ferramenta com a descrição do erro.

Para configurar um novo cenário web, clique no botão *Criar cenário web* no canto superior direito.

Opções de edição em massa

Os botões abaixo da lista oferecem algumas opções de edição em massa:

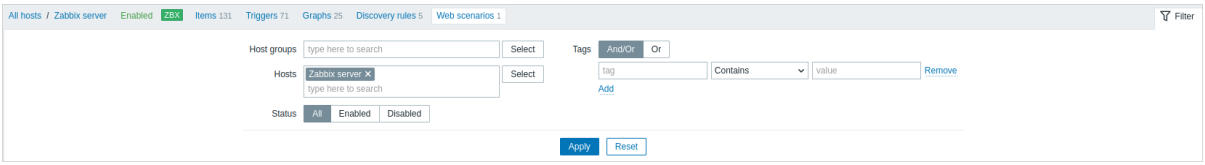
- *Habilitar* - altera o status do cenário para *Habilitado*
- *Desabilitar* - altera o status do cenário para *Desabilitado*
- *Limpar histórico e tendências* - limpa os dados de histórico e tendências dos cenários
- *Excluir* - exclui os cenários web

Para usar essas opções, marque as caixas de seleção antes dos respectivos cenários web e clique no botão desejado.

Usando o filtro

Você pode usar o filtro para exibir apenas os cenários nos quais está interessado. Para melhor desempenho na pesquisa, os dados são pesquisados com macros não resolvidas.

O link *Filtro* está disponível acima da lista de cenários web. Clicando nele, será aberto um filtro onde você pode especificar os critérios de filtragem desejados.



Parâmetro	Descrição
Grupos de hosts	Filtrar por um ou mais grupos de hosts. Ao especificar um grupo de hosts pai, todos os grupos de hosts aninhados são implicitamente selecionados.
Hosts	Filtrar por um ou mais hosts. Se grupos de hosts já estiverem selecionados acima, a seleção de hosts será limitada a esses grupos.
Status	Filtrar pelo status do cenário web.

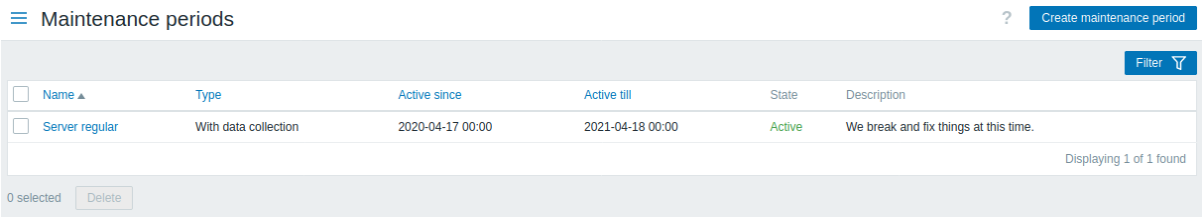
Parâmetro	Descrição
Tags	Filtrar por nome e valor da tag do cenário web. É possível incluir ou excluir tags e valores de tags específicos. Várias condições podem ser definidas. A correspondência do nome da tag diferencia maiúsculas de minúsculas. Há vários operadores disponíveis para cada condição: **Existe** - inclui os nomes de tags especificados; **Igual** - inclui os nomes e valores de tags especificados (diferencia maiúsculas de minúsculas); **Contém** - inclui os nomes de tags especificados onde os valores das tags contêm a string inserida (correspondência de substring, não diferencia maiúsculas de minúsculas); **Não existe** - exclui os nomes de tags especificados; **Diferente** - exclui os nomes e valores de tags especificados (diferencia maiúsculas de minúsculas); **Não contém** - exclui os nomes de tags especificados onde os valores das tags contêm a string inserida (correspondência de substring, não diferencia maiúsculas de minúsculas). Existem dois tipos de cálculo para as condições: **E/Ou** - todas as condições devem ser atendidas, condições com o mesmo nome de tag serão agrupadas pela condição Ou; **Ou** - basta que uma condição seja atendida.

5 Manutenção

Visão geral

Na seção *Coleta de dados* → *Manutenção*, os usuários podem configurar e manter períodos de manutenção para hosts.

Uma listagem dos períodos de manutenção existentes com seus detalhes é exibida.



Dados exibidos:

Coluna	Descrição
Nome	Nome do período de manutenção. Clicar no nome do período de manutenção abre o formulário de configuração do período de manutenção.
Tipo	O tipo de manutenção é exibido: *Com coleta de dados* ou *Sem coleta de dados*
Ativo desde	A data e hora em que a execução dos períodos de manutenção se torna ativa. Nota: Este horário não ativa um período de manutenção; os períodos de manutenção precisam ser definidos separadamente.
Ativo até	A data e hora em que a execução dos períodos de manutenção deixa de estar ativa.
Estado	O estado do período de manutenção: **Aproximando** - ficará ativo em breve **Ativo** - está ativo **Expirado** - não está mais ativo
Descrição	Descrição do período de manutenção é exibida.

Para configurar um novo período de manutenção, clique no botão *Criar período de manutenção* no canto superior direito.

Opções de edição em massa

Um botão abaixo da lista oferece uma opção de edição em massa:

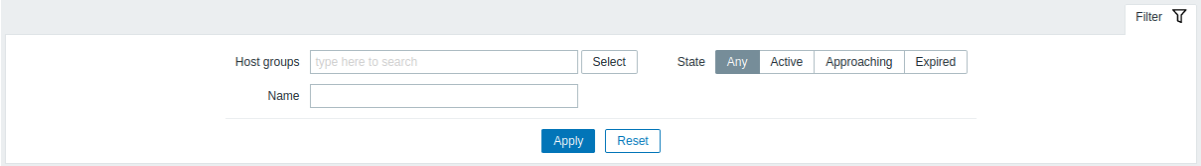
- *Excluir* - exclui os períodos de manutenção

Para usar esta opção, marque as caixas de seleção antes dos respectivos períodos de manutenção e clique em *Excluir*.

Usando o filtro

Você pode usar o filtro para exibir apenas os períodos de manutenção que lhe interessam. Para melhor desempenho na pesquisa, os dados são pesquisados com macros não resolvidas.

O link *Filtro* está disponível acima da lista de períodos de manutenção. Se você clicar nele, um filtro ficará disponível onde você pode filtrar os períodos de manutenção por grupo de hosts, nome e estado.



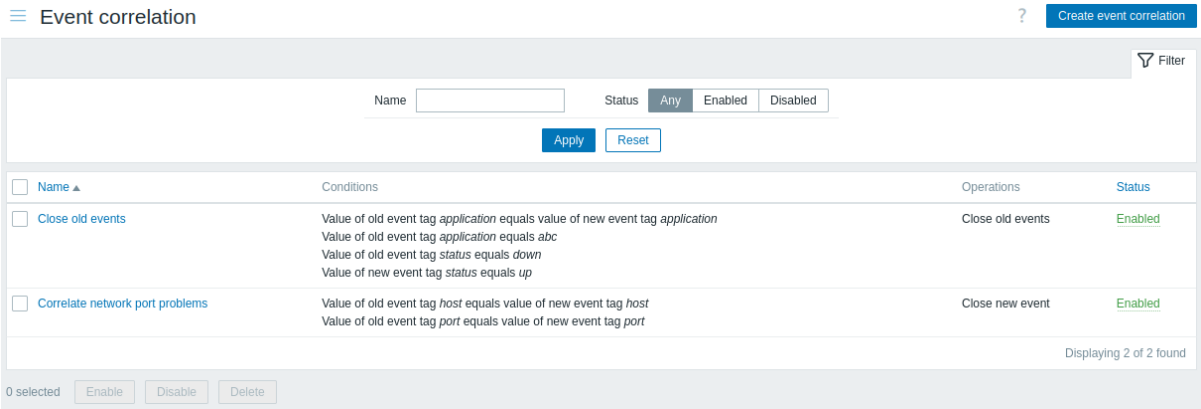
Cálculo de filas durante a manutenção

Attention:
O proxy Zabbix não tem conhecimento dos períodos de manutenção; consulte Cálculo de filas durante a manutenção para obter detalhes.

6 Correlação de eventos

Visão geral

Na seção *Coleta de dados* → *Correlação de eventos*, os usuários podem configurar e manter regras de correlação global para eventos do Zabbix.



Dados exibidos:

Coluna	Descrição
Nome	Nome da regra de correlação. Clicar no nome da regra de correlação abre o formulário de configuração da regra.
Condições	As condições da regra de correlação são exibidas.
Operações	As operações da regra de correlação são exibidas.
Status	O status da regra de correlação é exibido - *Ativado* ou *Desativado*. Ao clicar no status, você pode alterá-lo.

Para configurar uma nova regra de correlação, clique no botão *Criar correlação de eventos* no canto superior direito.

Opções de edição em massa

Os botões abaixo da lista oferecem algumas opções de edição em massa:

- *Ativar* - altera o status da regra de correlação para *Ativado*
- *Desativar* - altera o status da regra de correlação para *Desativado*
- *Excluir* - exclui as regras de correlação

Para usar essas opções, marque as caixas de seleção antes das respectivas regras de correlação e clique no botão desejado.

Usando o filtro

Você pode usar o filtro para exibir apenas as regras de correlação que são do seu interesse. Para melhor desempenho na pesquisa, os dados são pesquisados com macros não resolvidas.

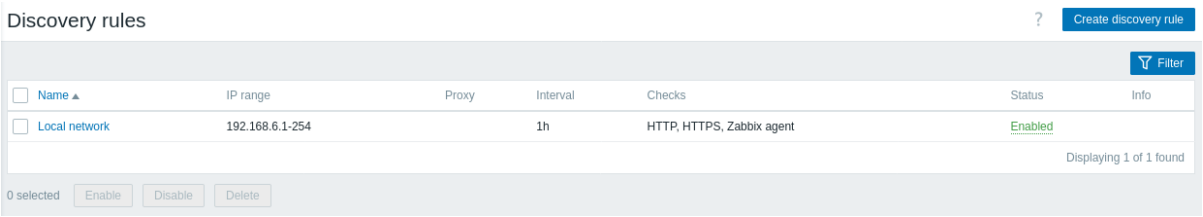
O link *Filtro* está disponível acima da lista de regras de correlação. Se você clicar nele, um filtro ficará disponível onde você pode filtrar as regras de correlação por nome e status.

7 Descoberta

Visão geral

Na seção *Coleta de dados* → *Descoberta*, os usuários podem configurar e manter regras de descoberta.

Uma listagem das regras de descoberta existentes com seus detalhes é exibida.



Dados exibidos:

Coluna	Descrição
Nome	Nome da regra de descoberta. Clicar no nome da regra de descoberta abre o formulário de configuração da regra de descoberta.
Intervalo de IP	O intervalo de endereços IP a ser usado para a varredura de rede é exibido.
Proxy	O nome do proxy é exibido, se a descoberta for realizada pelo proxy.
Intervalo	A frequência de execução da descoberta é exibida.
Verificações	Os tipos de verificações usadas para descoberta são exibidos.
Status	O status da regra de descoberta é exibido - *Habilitado* ou *Desabilitado*. Clicando no status você pode alterá-lo.
Info	Se tudo estiver funcionando corretamente, nada é exibido nesta coluna. Em caso de erros, um ícone de informação vermelho com a letra "i" é exibido. Passe o mouse sobre o ícone para ver uma dica de ferramenta com a descrição do erro.

Para configurar uma nova regra de descoberta, clique no botão *Criar regra de descoberta* no canto superior direito.

Opções de edição em massa

Os botões abaixo da lista oferecem algumas opções de edição em massa:

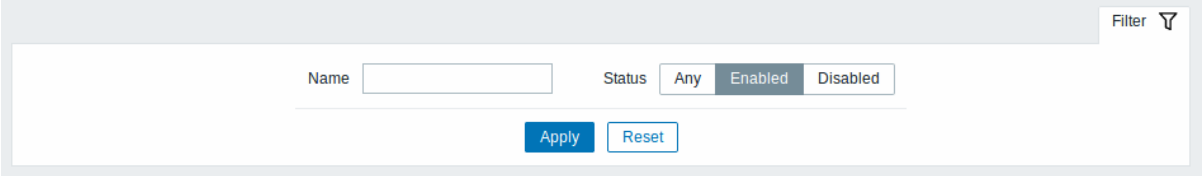
- *Ativar* - altera o status da regra de descoberta para *Ativado*
- *Desativar* - altera o status da regra de descoberta para *Desativado*
- *Excluir* - exclui as regras de descoberta

Para usar essas opções, marque as caixas de seleção antes das respectivas regras de descoberta e clique no botão desejado.

Usando o filtro

Você pode usar o filtro para exibir apenas as regras de descoberta que são do seu interesse. Para melhor desempenho na pesquisa, os dados são pesquisados com macros não resolvidas.

O link *Filtro* está disponível acima da lista de regras de descoberta. Se você clicar nele, um filtro ficará disponível onde você pode filtrar as regras de descoberta por nome e status.



7 Alertas

Visão geral

Este menu apresenta seções relacionadas à configuração de alertas no Zabbix.

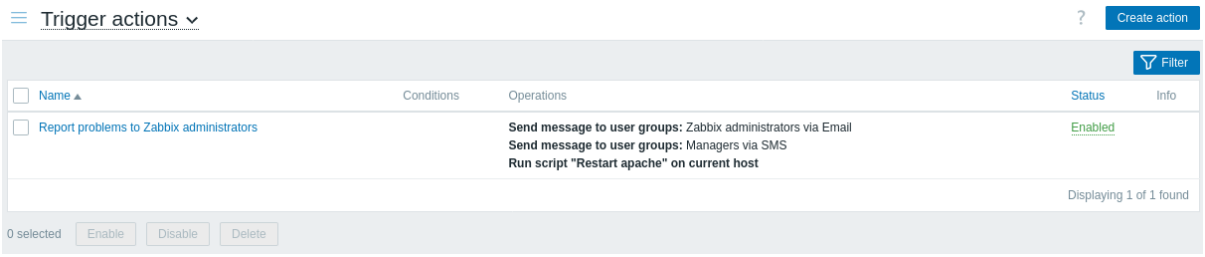
1 Ações

Visão geral

Na seção *Alertas* → *Ações*, os usuários podem configurar e manter ações.

As ações exibidas são ações atribuídas à fonte de evento selecionada (trigger, serviços, descoberta, autoregistro, ações internas).

Para alternar para uma fonte de evento diferente, clique em *Ações* na seção de menu *Alertas*. Também é possível alternar entre as fontes usando o menu suspenso do título no canto superior esquerdo.



Dados exibidos:

Coluna	Descrição
Nome	Nome da ação. Clicar no nome da ação abre o formulário de configuração da ação.
Condições	As condições da ação são exibidas.
Operações	As operações da ação são exibidas. A lista de operações também exibe o tipo de mídia (e-mail, SMS ou script) usado para notificação, bem como o nome e sobrenome (entre parênteses após o nome de usuário) de um destinatário da notificação. A operação da ação pode ser tanto uma notificação quanto um comando remoto, dependendo do tipo de operação selecionado.
Status	O status da ação é exibido - *Habilitado* ou *Desabilitado*. Clicando no status você pode alterá-lo. Veja a seção Escalonamentos para mais detalhes sobre o que acontece se uma ação for desabilitada durante um escalonamento em andamento.
Info	Se tudo estiver funcionando corretamente, nenhum ícone é exibido nesta coluna. Em caso de erros, como operações ou condições de ação ausentes após upgrade/descoberta de baixo nível, um ícone de aviso é exibido. Passe o mouse sobre o ícone para ver uma dica de ferramenta com a descrição do erro. Esta coluna é exibida desde o Zabbix 7.0.10.

Para configurar uma nova ação, clique no botão *Criar ação* no canto superior direito.

Para usuários sem direitos de Super admin, as ações são exibidas de acordo com as configurações de permissão. Isso significa que, em alguns casos, um usuário sem direitos de Super admin não consegue visualizar a lista completa de ações devido a certas restrições de permissão. Uma ação é exibida para o usuário sem direitos de Super admin se as seguintes condições forem atendidas:

- O usuário tem acesso de leitura e gravação a grupos de hosts, hosts, templates e triggers nas condições da ação
- O usuário tem acesso de leitura e gravação a grupos de hosts, hosts e templates nas operações da ação, operações de recuperação e operações de atualização
- O usuário tem acesso de leitura a grupos de usuários e usuários nas operações da ação, operações de recuperação e operações de atualização

Opções de edição em massa

Os botões abaixo da lista oferecem algumas opções de edição em massa:

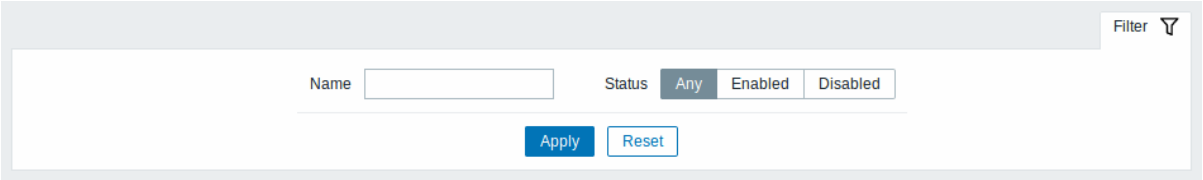
- *Ativar* - altere o status da ação para *Ativado*
- *Desativar* - altere o status da ação para *Desativado*
- *Excluir* - exclui as ações

Para usar essas opções, marque as caixas de seleção antes das respectivas ações, em seguida, clique no botão desejado.

Usando filtro

Você pode usar o filtro para exibir apenas as ações de seu interesse para um melhor desempenho de pesquisa, os dados são pesquisados com macros não resolvido.

O link *Filtro* está disponível acima da lista de ações. Se você clicar nele, fica disponível um filtro onde você pode filtrar as ações por nome e status.



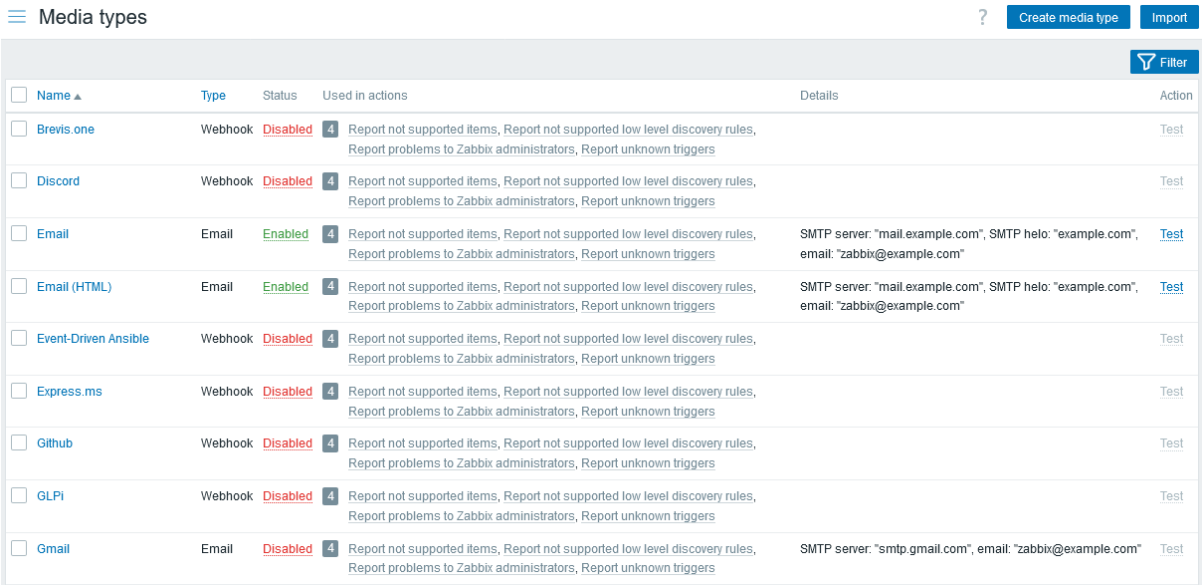
2 Tipos de mídia

Visão geral

Na seção *Alertas → Tipos de mídia*, os usuários podem configurar e manter informações sobre tipos de mídia.

As informações do tipo de mídia contêm instruções gerais para usar um meio como canal de entrega para notificações. Detalhes específicos, como os endereços de e-mail individuais para os quais enviar uma notificação, são mantidos com usuários individuais.

Uma lista dos tipos de mídia existentes com seus detalhes é exibida.



Dados exibidos:

Coluna	Descrição
Nome	Nome do tipo de mídia. Clicar no nome abre o formulário de configuração do tipo de mídia.
Tipo	O tipo de mídia (e-mail, SMS, etc) é exibido.
Status	O status do tipo de mídia é exibido - *Habilitado* ou *Desabilitado*. Clicando no status você pode alterá-lo.
Usado em ações	As ações onde o tipo de mídia é usado são exibidas, (desde o Zabbix 7.0.4) precedidas pelo número total dessas ações. Clicar no nome da ação abre o formulário de configuração da ação. Se o usuário não tiver permissões para a ação, o nome não será clicável.
Detalhes	Informações detalhadas do tipo de mídia são exibidas.
Ações	A seguinte ação está disponível: **Testar** - clique para abrir um formulário de teste onde você pode inserir parâmetros do tipo de mídia (por exemplo, um endereço de destinatário com assunto e corpo de teste) e enviar uma mensagem de teste para verificar se o tipo de mídia configurado funciona. Veja também: Teste de tipo de mídia para E-mail, Webhook ou Script.

Para configurar um novo tipo de mídia, clique no botão *Criar tipo de mídia* no canto superior direito.

Para importar um tipo de mídia, clique no botão *Importar* no canto superior direito.

Opções de edição em massa

Os botões abaixo da lista oferecem algumas opções de edição em massa:

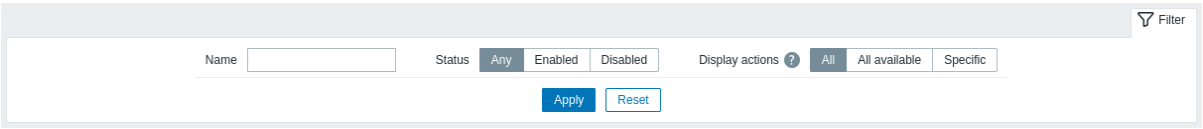
- *Ativar* - altere o status do tipo de mídia para *Ativado*
- *Desativar* - altere o status do tipo de mídia para *Desativado*
- *Exportar* - exporta os tipos de mídia para um arquivo YAML, XML ou JSON
- *Excluir* - exclua os tipos de mídia

Para usar essas opções, marque as caixas de seleção antes da respectiva mídia tipos e, em seguida, clique no botão necessário.

Usando o filtro

Você pode usar o filtro para exibir apenas os tipos de mídia de seu interesse. Para melhor desempenho na pesquisa, os dados são pesquisados com macros não resolvidas.

O link *Filtro* está disponível acima da lista de tipos de mídia. Se você clicar nele, um filtro ficará disponível onde você pode filtrar os tipos de mídia por nome e status. Além disso, desde o Zabbix 7.0.7, você pode usar o filtro para exibir ações na coluna *Usado em ações* com base no escopo de uso do tipo de mídia (definido pelo parâmetro *Enviar para tipo de mídia* nos detalhes da operação da ação).



3 Scripts

Visão geral

Na seção *Alertas* > *Scripts*, scripts globais definidos pelo usuário podem ser configurados e mantidos.

Os scripts globais, dependendo do escopo configurado e também das permissões do usuário, estão disponíveis para execução:

- a partir do menu do host em vários locais do frontend (*Dashboard*, *Problemas*, *Últimos dados*, *Mapas*, etc.)
- a partir do menu de eventos
- podem ser executados como uma operação de ação

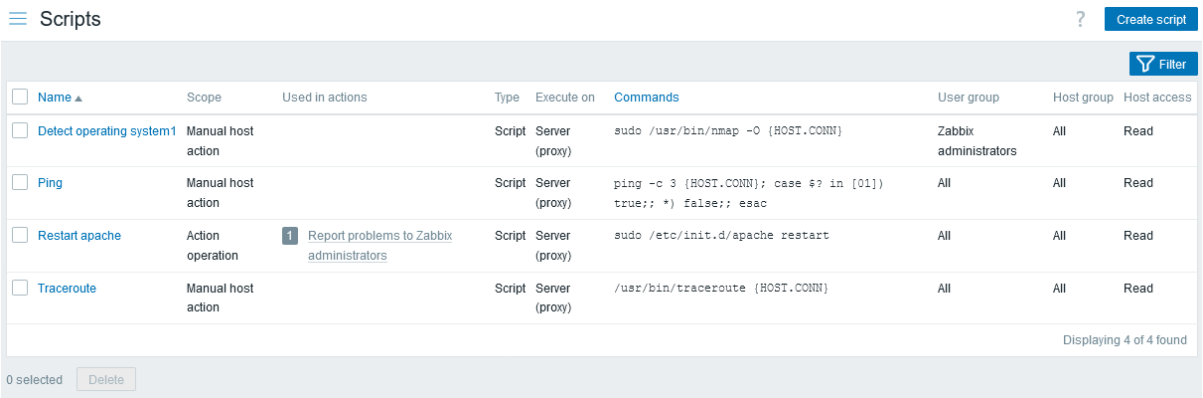
Os scripts são executados no Zabbix agent, Zabbix server (proxy) ou apenas no Zabbix server. Veja também Execução de comandos.

Tanto no Zabbix agent quanto no Zabbix proxy, os scripts remotos estão desabilitados por padrão. Eles podem ser habilitados por:

- Para comandos remotos executados no Zabbix agent:
  - adicionando um parâmetro AllowKey=system.run[<comando>,*] para cada comando permitido na configuração do agent, * representa o modo wait e nowait;
- Para comandos remotos executados no Zabbix proxy:
  - **Atenção: Não é necessário habilitar comandos remotos no Zabbix proxy se os comandos remotos forem executados no Zabbix agent que é monitorado pelo Zabbix proxy.** Se, no entanto, for necessário executar comandos remotos no Zabbix proxy, defina o parâmetro *EnableRemoteCommands* como '1' na configuração do proxy.

A execução de scripts globais no Zabbix server pode ser desabilitada definindo `EnableGlobalScripts=0` na configuração do server. Para novas instalações, desde o Zabbix 7.0, a execução de scripts globais no Zabbix server está desabilitada por padrão.

Uma listagem dos scripts existentes com seus detalhes é exibida.



Dados exibidos:

Coluna	Descrição
Nome	Nome do script. Clicar no nome do script abre o formulário de configuração do script.
Escopo	Escopo do script - operação de ação, ação manual de host ou ação manual de evento. Esta configuração determina onde o script está disponível.
Usado em ações	Todas as ações onde o script é usado são exibidas, (desde o Zabbix 7.0.4) precedidas pelo número total dessas ações. Clicar no nome da ação abre o formulário de configuração da ação. Se o usuário não tiver permissões para a ação, o nome não será clicável.
Tipo	O tipo de script é exibido - comando *URL*, *Webhook*, *Script*, *SSH*, *Telnet* ou *IPMI*.
Executar em	É exibido se o script será executado no Zabbix agent, Zabbix proxy ou server, ou apenas no Zabbix server.
Comandos	Todos os comandos a serem executados dentro do script são exibidos. Nada é exibido aqui para webhooks.
Grupo de usuários	O grupo de usuários para o qual o script está disponível é exibido (ou *Todos* para todos os grupos de usuários).
Grupo de hosts	O grupo de hosts para o qual o script está disponível é exibido (ou *Todos* para todos os grupos de hosts).
Acesso ao host	O nível de permissão para o grupo de hosts é exibido - *Leitura* ou *Escrita*. Apenas usuários com o nível de permissão necessário terão acesso à execução do script.

Para configurar um novo script, clique no botão *Criar script* no canto superior direito.

Opções de edição em massa

Um botão abaixo da lista oferece uma opção de edição em massa:

- *Excluir* - exclui os scripts

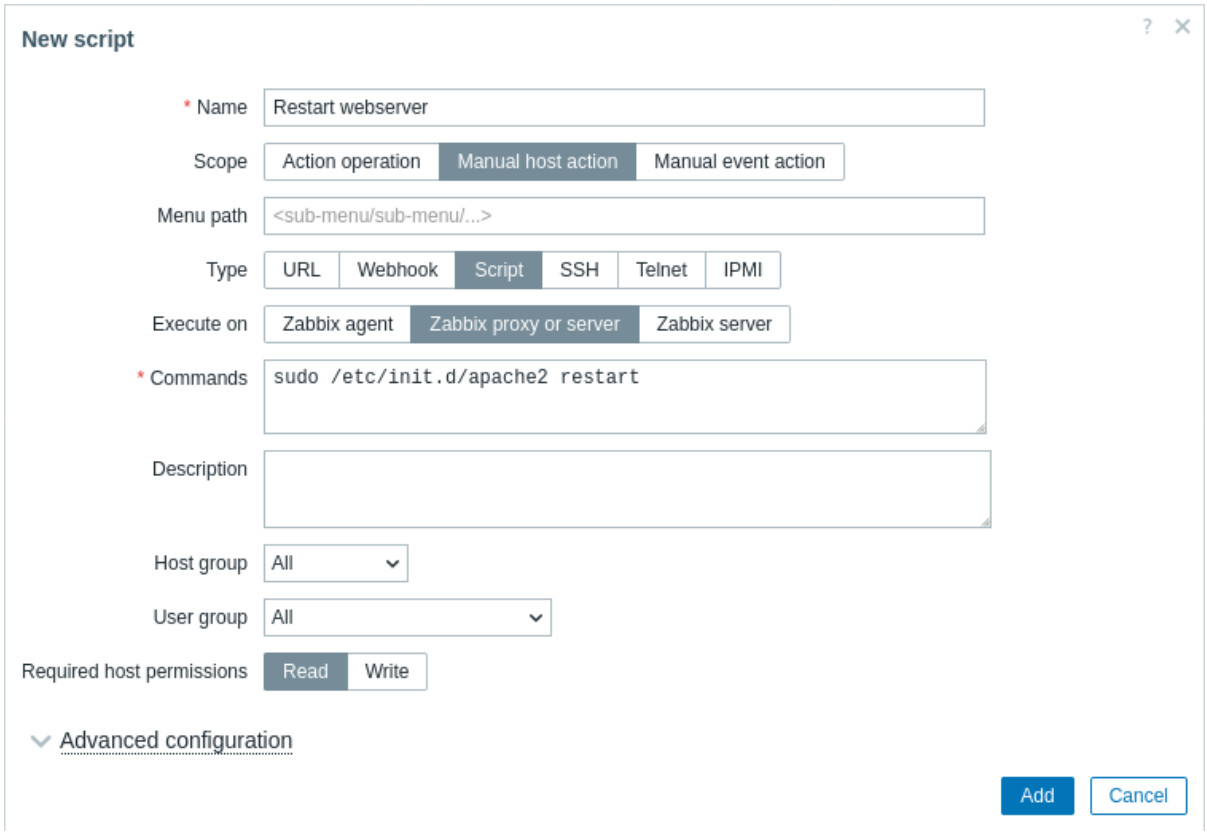
Para usar esta opção, marque as caixas de seleção antes dos scripts respectivos e clique em *Excluir*.

Usando o filtro

Você pode usar o filtro para exibir apenas os scripts do seu interesse. Para melhor desempenho de pesquisa, os dados são pesquisados com macros não resolvidas.

O link *Filtro* está disponível acima da lista de scripts. Se você clicar nele, um filtro ficará disponível onde você pode filtrar scripts por nome e escopo.

Configurando um script global



Note:
Recomenda-se usar funções de macro em vez de macros simples em scripts globais para aumentar a segurança, pois as macros não são automaticamente escapadas.

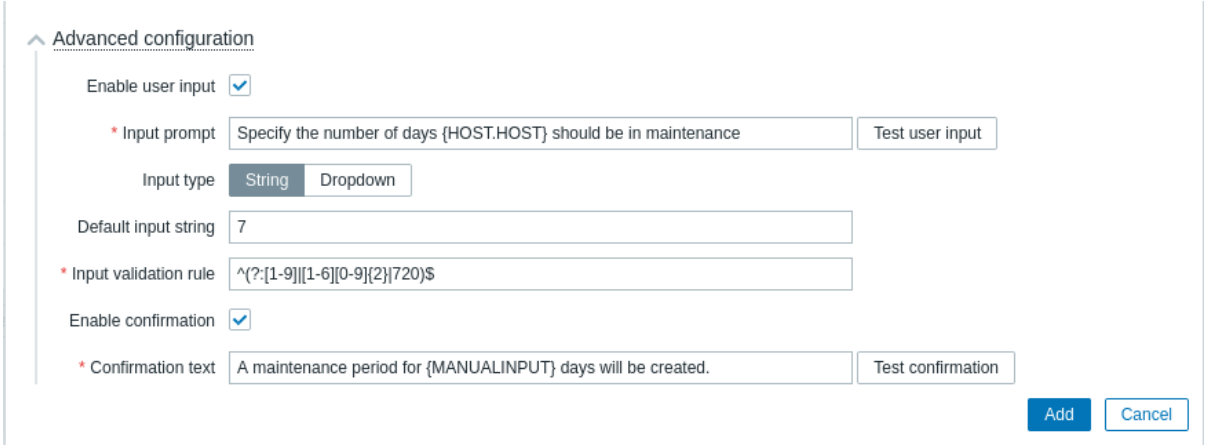
Atributos do script:

Parâmetro	Descrição
Nome	Nome exclusivo do script. Ex.: `Limpar sistema de arquivos /tmp`
Escopo	Escopo do script - operação de ação, ação manual de host ou ação manual de evento. Esta configuração determina onde o script pode ser usado - em comandos remotos de operações de ação, a partir do menu de host ou do menu de evento, respectivamente. Definir o escopo como 'Operação de ação' torna o script disponível para todos os usuários com acesso a *Alertas* > *Ações*. Se um script estiver realmente sendo usado em uma ação, seu escopo não pode ser alterado de 'operação de ação'. **Suporte a macros** O escopo afeta o alcance das macros disponíveis. Por exemplo, macros relacionadas ao usuário ({USER.*}) são suportadas em scripts para permitir o envio de informações sobre o usuário que iniciou o script. No entanto, elas não são suportadas se o escopo do script for operação de ação, pois as operações de ação são executadas automaticamente. A macro {MANUALINPUT} permite especificar uma entrada manual no momento da execução do script. Ela é suportada para scripts de ação manual de host e de ação manual de evento. Para saber quais outras macros são suportadas, faça uma busca por 'Notificações e comandos baseados em trigger/Comandos baseados em trigger', 'Scripts de ação manual de host' e 'Scripts de ação manual de evento' na tabela de macros suportadas. Observe que, se uma macro puder ser resolvida para um valor com espaços (por exemplo, nome do host), não se esqueça de colocar aspas conforme necessário.
Caminho do menu	O caminho de menu desejado para o script. Por exemplo, `Padrão` ou `Padrão/`, exibirá o script no respectivo diretório. Os menus podem ser aninhados, por exemplo, `Menu principal/Submenu1/Submenu2`. Ao acessar scripts pelo menu de host/evento nas seções de monitoramento, eles serão organizados de acordo com os diretórios fornecidos. Este campo é exibido apenas se 'Ação manual de host' ou 'Ação manual de evento' for selecionado como *Escopo*.
Tipo	Clique no botão respectivo para selecionar o tipo de script: **URL**, **Webhook**, **Script**, **SSH**, **Telnet** ou comando **IPMI**. O tipo **URL** está disponível apenas quando 'Ação manual de host' ou 'Ação manual de evento' é selecionado como *Escopo*.
Tipo de script: **URL**	
URL	Especifique a URL para acesso rápido a partir do menu de host ou menu de evento. Macros e macros de usuário personalizadas são suportadas. O suporte a macros depende do escopo do script (veja *Escopo* acima). Use a macro {MANUALINPUT} neste campo para poder especificar uma entrada manual no momento da execução do script, por exemplo: `http://{MANUALINPUT}/zabbix/zabbix.php?action=dashboard.view` Os valores das macros não devem ser codificados em URL.
Abrir em nova janela	Determina se a URL deve ser aberta em uma nova guia do navegador ou na mesma.
Tipo de script: **Webhook**	
Parâmetros	Especifique as variáveis do webhook como pares atributo-valor. Veja também: configuração de mídia Webhook. Macros e macros de usuário personalizadas são suportadas nos valores dos parâmetros. O suporte a macros depende do escopo do script (veja *Escopo* acima).
Script	Digite o código JavaScript no editor modal que é aberto ao clicar no campo de parâmetro ou no ícone de lápis ao lado dele. O suporte a macros depende do escopo do script (veja *Escopo* acima). Veja também: configuração de mídia Webhook, Objetos Javascript adicionais.
Timeout	Tempo limite de execução do JavaScript (1-60s, padrão 30s). Sufixos de tempo são suportados, por exemplo, 30s, 1m.
Tipo de script: **Script**	

Parâmetro	Descrição
Executar em	Clique no botão respectivo para executar o script shell em: **Zabbix agent** - o script será executado pelo Zabbix agent (se o item system.run estiver permitido) no host **Zabbix proxy ou server** - o script será executado pelo Zabbix proxy ou server - dependendo se o host é monitorado por proxy ou server. Será executado no proxy se ativado por EnableRemoteCommands. Será executado no server se scripts globais estiverem ativados pelo parâmetro EnableGlobalScripts do server. **Zabbix server** - o script será executado apenas pelo Zabbix server. Esta opção não estará disponível se scripts globais estiverem desativados pelo parâmetro EnableGlobalScripts do server.
Comandos	Digite o caminho completo para os comandos a serem executados dentro do script. O suporte a macros depende do escopo do script (veja *Escopo* acima). Macros de usuário personalizadas são suportadas.
Tipo de script: **SSH**	
Método de autenticação	Selecione o método de autenticação - senha ou chave pública.
Nome de usuário	Digite o nome de usuário.
Senha	Digite a senha. Este campo está disponível se 'Senha' for selecionado como método de autenticação.
Arquivo de chave pública	Digite o caminho para o arquivo de chave pública. Este campo está disponível se 'Chave pública' for selecionado como método de autenticação.
Arquivo de chave privada	Digite o caminho para o arquivo de chave privada. Este campo está disponível se 'Chave pública' for selecionado como método de autenticação.
Frase secreta	Digite a frase secreta. Este campo está disponível se 'Chave pública' for selecionado como método de autenticação.
Porta	Digite a porta do serviço SSH remoto no host de destino ao qual o Zabbix irá se conectar.
Comandos	Digite os comandos. O suporte a macros depende do escopo do script (veja *Escopo* acima). Macros de usuário personalizadas são suportadas.
Tipo de script: **Telnet**	
Nome de usuário	Digite o nome de usuário.
Senha	Digite a senha.
Porta	Digite a porta do serviço Telnet remoto no host de destino ao qual o Zabbix irá se conectar.
Comandos	Digite os comandos. O suporte a macros depende do escopo do script (veja *Escopo* acima). Macros de usuário personalizadas são suportadas.
Tipo de script: **IPMI**	
Comando	Digite o comando IPMI. O suporte a macros depende do escopo do script (veja *Escopo* acima). Macros de usuário personalizadas são suportadas.
Descrição	Digite uma descrição para o script.
Grupo de hosts	Selecione o grupo de hosts para o qual o script estará disponível (ou *Todos* para todos os grupos de hosts).
Grupo de usuários	Selecione o grupo de usuários para o qual o script estará disponível (ou *Todos* para todos os grupos de usuários). Este campo é exibido apenas se 'Ação manual de host' ou 'Ação manual de evento' for selecionado como *Escopo*.
Permissões de host necessárias	Selecione o nível de permissão para o grupo de hosts - *Leitura* ou *Escrita*. Apenas usuários com o nível de permissão necessário terão acesso à execução do script. Este campo é exibido apenas se 'Ação manual de host' ou 'Ação manual de evento' for selecionado como *Escopo*.
Configuração avançada	Clique no rótulo *Configuração avançada* para exibir as opções de configuração avançada. Este campo é exibido apenas se 'Ação manual de host' ou 'Ação manual de evento' for selecionado como *Escopo*.

Configuração avançada

Opções de configuração avançada estão disponíveis em uma seção recolhível *Configuração avançada*:



Parâmetro	Descrição
Habilitar entrada do usuário	Marque a caixa de seleção para habilitar a entrada manual do usuário antes de executar o script. A entrada manual do usuário substituirá o valor da macro {MANUALINPUT} no script. Veja também: Entrada manual do usuário.
Mensagem de entrada	Insira um texto personalizado solicitando a entrada personalizada do usuário. Este texto será exibido acima do campo de entrada no popup *Entrada manual*. Para ver uma prévia do popup *Entrada manual*, clique em *Testar entrada do usuário*. A prévia também permite testar se a string de entrada está em conformidade com a regra de validação de entrada (veja os parâmetros abaixo). O suporte a macro e macro de usuário depende do escopo do script (veja *Escopo* nos parâmetros gerais de configuração do script).
Tipo de entrada	Selecione o tipo de entrada manual: **String** - string única; **Dropdown** - o valor é selecionado entre várias opções do menu suspenso.
Opções do menu suspenso	Insira valores exclusivos para o menu suspenso de entrada do usuário em uma lista separada por vírgulas. Para incluir uma opção vazia no menu suspenso, adicione uma vírgula extra no início, meio ou final da lista. Este campo é exibido apenas se 'Dropdown' for selecionado como *Tipo de entrada*.
String de entrada padrão	Insira a string padrão para a entrada do usuário (ou nenhuma). Este campo será validado em relação à expressão regular fornecida no campo *Regra de validação de entrada*. O valor inserido aqui será exibido por padrão no popup *Entrada manual*. Este campo é exibido apenas se 'String' for selecionado como *Tipo de entrada*.
Regra de validação de entrada	Insira uma expressão regular para validar a string de entrada do usuário. Expressões regulares globais são suportadas. Este campo é exibido apenas se 'String' for selecionado como *Tipo de entrada*.
Habilitar confirmação	Marque a caixa de seleção para exibir uma mensagem de confirmação antes de executar o script. Este recurso pode ser especialmente útil com operações potencialmente perigosas (como um script de reinicialização) ou aquelas que podem levar muito tempo.
Texto de confirmação	Insira um texto de confirmação personalizado para o popup de confirmação habilitado com a caixa de seleção acima (por exemplo, *O sistema remoto será reiniciado. Tem certeza?*). Para ver como o texto ficará, clique em *Testar confirmação* ao lado do campo. Macros e macros de usuário personalizadas são suportadas. *Nota:* as macros não serão expandidas ao testar a mensagem de confirmação.

Se tanto a entrada manual do usuário quanto uma mensagem de confirmação forem configuradas, elas serão exibidas em janelas popup consecutivas.

Entrada manual do usuário

A entrada manual do usuário permite fornecer um parâmetro personalizado em cada execução do script. Isso elimina a necessidade de criar vários scripts de usuário semelhantes com apenas uma diferença de parâmetro.

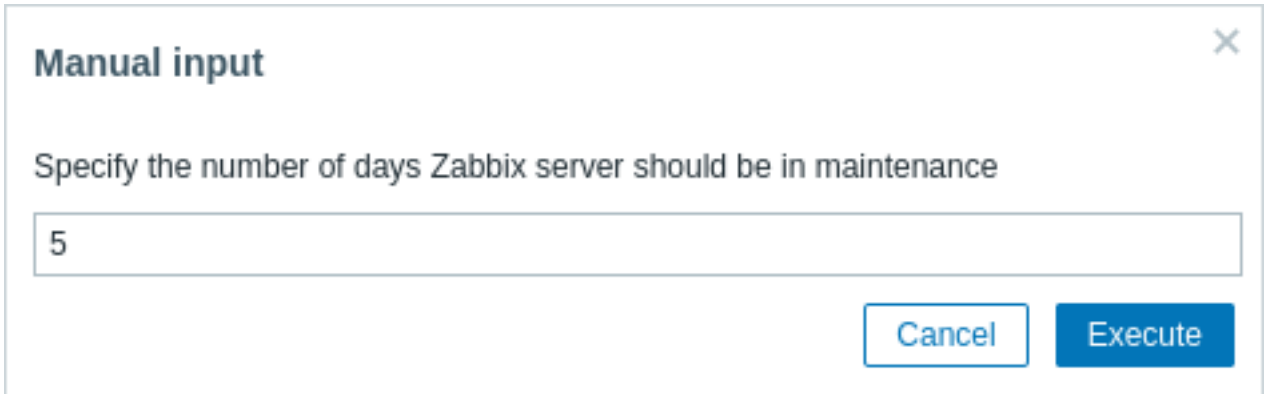
Por exemplo, você pode querer fornecer um número inteiro diferente ou um endereço de URL diferente para o script durante a execução.

Para habilitar a entrada manual do usuário:

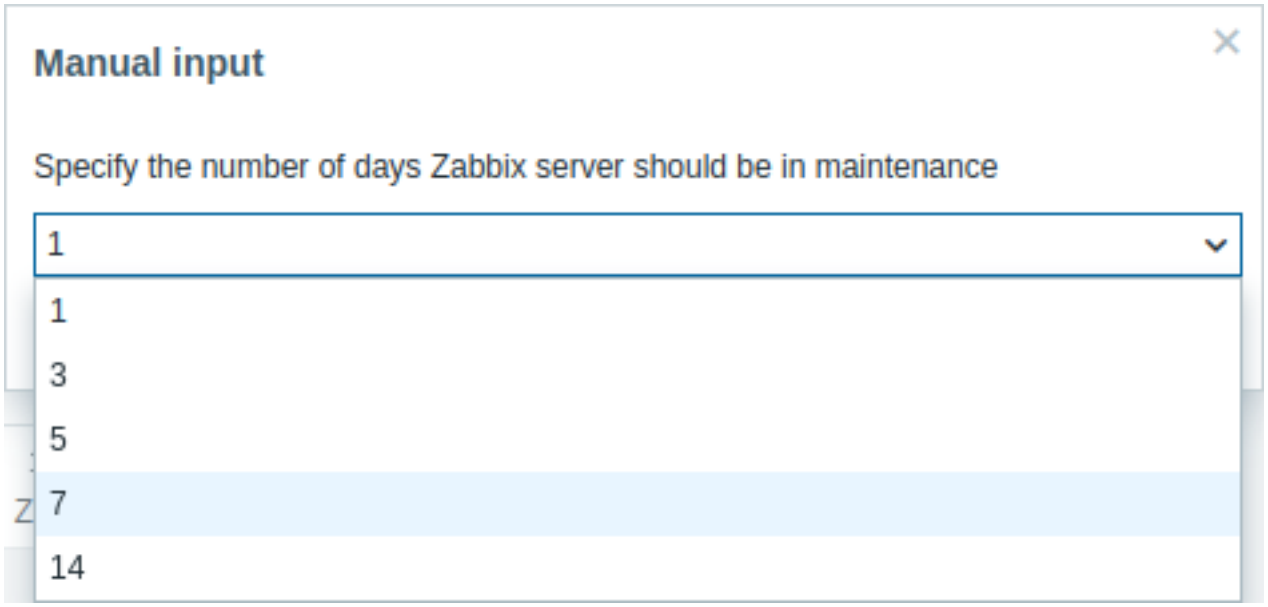
- use a macro {MANUALINPUT} no script (comandos, script, parâmetro do script) onde necessário; ou no campo URL de scripts de URL;
- na configuração avançada do script, habilite a entrada manual do usuário e configure as opções de entrada.

Com a entrada do usuário habilitada, antes da execução do script, um popup *Entrada manual* aparecerá para o usuário solicitando o fornecimento de um valor personalizado. O valor fornecido substituirá {MANUALINPUT} no script.

Dependendo da configuração, o usuário será solicitado a inserir um valor de string:



Ou selecionar o valor em uma lista suspensa de opções predefinidas:



A entrada manual do usuário está disponível apenas para scripts cujo escopo seja 'Ação manual do host' ou 'Ação manual do evento'.

Execução do script e resultado

Os scripts executados pelo Zabbix server são executados na ordem descrita na página de execução de comandos.

O resultado do script é exibido em uma janela pop-up que aparece após a execução do script. O valor de retorno do script é uma saída padrão:

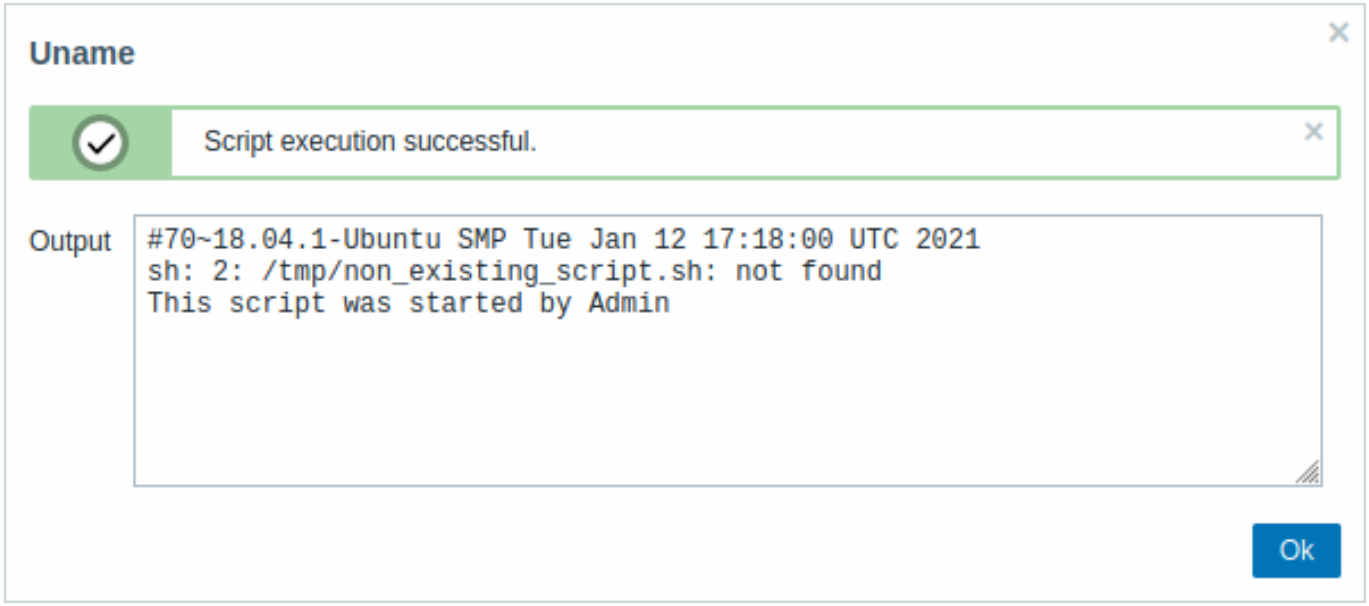
- Se o script for finalizado com sucesso (código de saída 0), o valor de retorno é limitado a 16MB (incluindo espaços em branco à direita, que são truncados).
- Se o script for finalizado com erro (código de saída diferente de zero), o valor de retorno é um erro padrão limitado a 2KB.

O Zabbix não armazena a saída estendida do script por padrão. Para preservar todos os detalhes da saída, você pode implementar o registro de logs dentro do próprio script (por exemplo, redirecionando a saída para um arquivo de log local).

Observe que, para scripts executados no Zabbix server ou Zabbix proxy, os limites do banco de dados também se aplicam.

Abaixo está um exemplo de script e a janela de resultado:

```
uname -v
/tmp/non_existing_script.sh
echo "This script was started by {USER.USERNAME}"
```



O resultado do script não exibe o próprio script.

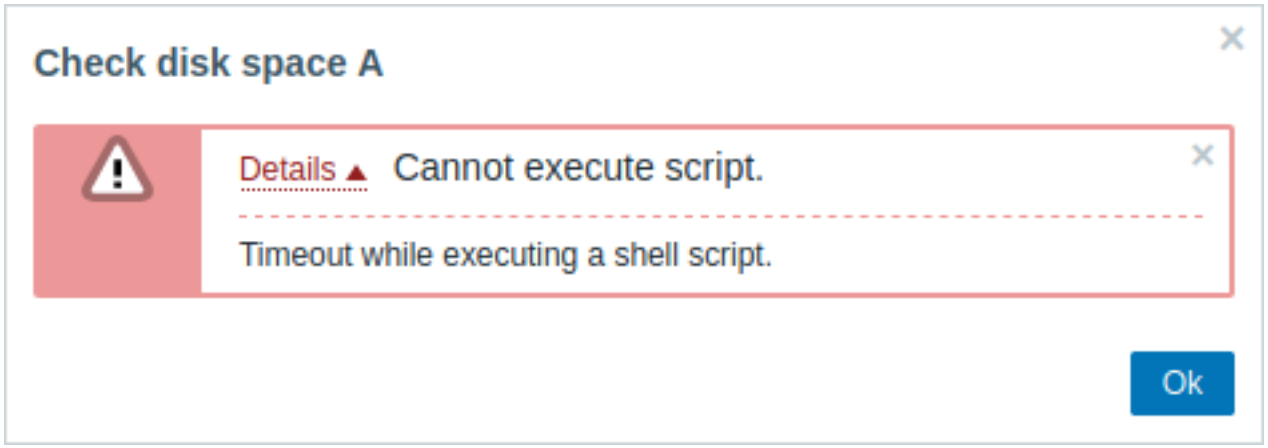
Tempo limite do script

Zabbix agent

Você pode se deparar com uma situação em que ocorre um timeout ao executar um script.

Veja um exemplo de um script rodando no Zabbix agent e a janela de resultado abaixo:

```
sleep 5
df -h
```



A mensagem de erro, neste caso, é a seguinte:

```
Timeout while executing a shell script.
```

Para evitar tais situações, é aconselhável otimizar o próprio script (no exemplo acima, "5") em vez de ajustar o parâmetro `Timeout` na configuração do Zabbix agent e na configuração do Zabbix server. No entanto, para o Zabbix agent em modo ativo, o parâmetro `Timeout` na configuração do Zabbix server deve ser pelo menos alguns segundos maior do que o parâmetro `RefreshActiveChecks` na configuração do Zabbix agent. Isso garante que o server tenha tempo suficiente para receber os resultados das checagens ativas do agent. Observe que a execução de scripts em um agent ativo é suportada desde o Zabbix agent 7.0.

Caso o parâmetro `Timeout` tenha sido alterado na configuração do Zabbix agent, a seguinte mensagem de erro aparecerá:

```
Get value from agent failed: ZBX_TCP_READ() timed out.
```

Isso significa que a modificação foi feita na configuração do Zabbix agent, mas é necessário modificar também o parâmetro `Timeout` na configuração do Zabbix server.

Zabbix server/proxy

Veja um exemplo de um script rodando no Zabbix server e a janela de resultado abaixo:

```
sleep 11
df -h
```

Também é aconselhável otimizar o próprio script (em vez de ajustar o parâmetro `TrapperTimeout` para um valor correspondente (no nosso caso, > 11) modificando a configuração do Zabbix server).

8 Usuários

Visão geral

Este menu apresenta seções relacionadas à configuração de usuários no Zabbix. Este menu está disponível apenas para usuários do tipo SuperAdmin.

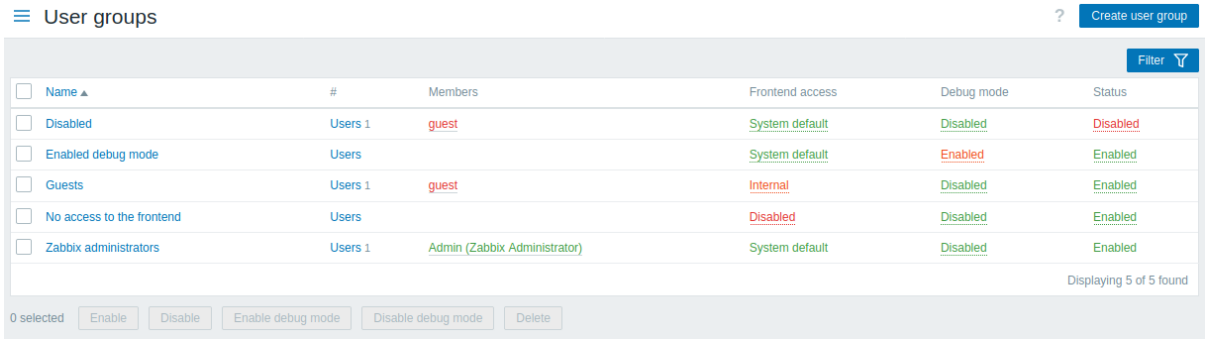
1 Grupos de usuários

Visão geral

Na seção *Usuários* → *Grupos de usuários*, os grupos de usuários do sistema são mantidos.

Grupos de usuários

Uma lista dos grupos de usuários existentes com seus detalhes é exibida.



Dados exibidos:

Coluna	Descrição
Nome	Nome do grupo de usuários. Clicar no nome do grupo de usuários abre o formulário de configuração do grupo.
#	O número de usuários no grupo. Clicar em *Usuários* exibirá os respectivos usuários filtrados na lista de usuários.
Membros	Nomes de usuário dos usuários individuais no grupo (com nome e sobrenome entre parênteses). Clicar no nome de usuário abrirá o formulário de configuração do usuário. Usuários de grupos desabilitados são exibidos em vermelho.
Acesso ao frontend	O nível de acesso ao frontend é exibido: **Padrão do sistema** - os usuários são autenticados pelo Zabbix, LDAP ou HTTP (dependendo do método de autenticação definido globalmente); **Interno** - os usuários são autenticados pelo Zabbix; ignorado se a autenticação HTTP for o padrão global; **LDAP** - os usuários são autenticados pelo LDAP; ignorado se a autenticação HTTP for o padrão global; **Desabilitado** - o acesso ao frontend do Zabbix é proibido para este grupo. Clicando no nível atual, você pode alterá-lo.
Modo de depuração	O status do modo de depuração é exibido - *Habilitado* ou *Desabilitado*. Clicando no status, você pode alterá-lo.
Status	O status do grupo de usuários é exibido - *Habilitado* ou *Desabilitado*. Clicando no status, você pode alterá-lo.

Para configurar um novo grupo de usuários, clique no botão *Criar grupo de usuários* no canto superior direito.

Opções de edição em massa

Os botões abaixo da lista oferecem algumas opções de edição em massa:

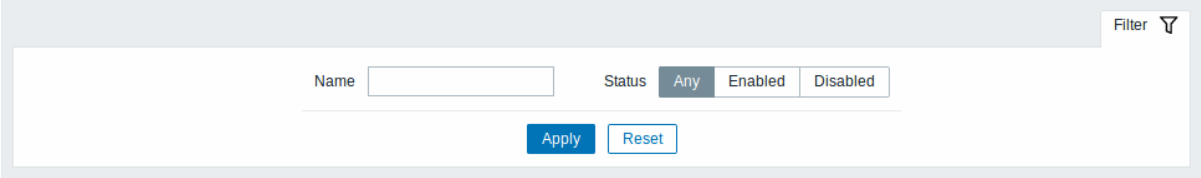
- *Ativar* - altera o status do grupo de usuários para *Ativado*
- *Desativar* - altera o status do grupo de usuários para *Desativado*
- *Ativar modo de depuração* - ativar o modo de depuração para os grupos de usuários
- *Desabilitar o modo de depuração* - desabilite o modo de depuração para os grupos de usuários
- *Excluir* - exclui os grupos de usuários

Para usar essas opções, marque as caixas de seleção antes do respectivo usuário grupos e, em seguida, clique no botão necessário.

Usando filtro

Você pode usar o filtro para exibir apenas os grupos de usuários que você está interesse. Para um melhor desempenho de pesquisa, os dados são pesquisados com macros não resolvidas.

O link *Filtro* está disponível acima da lista de grupos de usuários. Se você clique nele, um filtro fica disponível onde você pode filtrar grupos de usuários por nome e status.



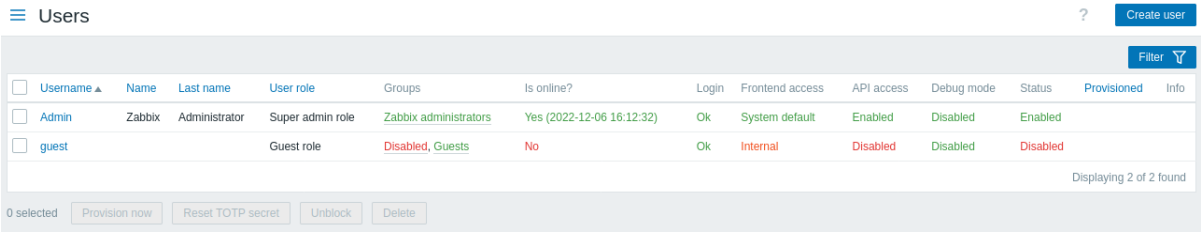
3 Usuários

Visão geral

Na seção *Usuários* → *Usuários* são mantidos os usuários do sistema.

Usuários

Uma lista de usuários existentes com seus detalhes é exibida.



Dados exibidos:

Coluna	Descrição
Nome de usuário	Nome de usuário para login no Zabbix. Clicar no nome de usuário abre o formulário de configuração do usuário.
Nome	Primeiro nome do usuário.
Sobrenome	Segundo nome do usuário.
Função do usuário	A função do usuário é exibida.
Grupos	Os grupos dos quais o usuário é membro são listados. Clicar no nome do grupo de usuários abre o formulário de configuração do grupo. Grupos desabilitados são exibidos em vermelho.
Está online?	O status online do usuário é exibido - *Sim* ou *Não*. O horário da última atividade do usuário é exibido entre parênteses.
Login	O status de login do usuário é exibido - *Ok* ou *Bloqueado*. Um usuário pode ser temporariamente bloqueado ao exceder o número de tentativas de login malsucedidas definido na seção *Administração → Geral → Outros* (cinco por padrão). Clicando em *Bloqueado* você pode desbloquear o usuário.
Acesso ao frontend	O nível de acesso ao frontend é exibido - *Padrão do sistema*, *Interno*, *LDAP* ou *Desabilitado*, dependendo do que está definido para todo o grupo de usuários.
Acesso à API	O status de acesso à API é exibido - *Habilitado* ou *Desabilitado*, dependendo do que está definido para a função do usuário.
Modo de depuração	O status do modo de depuração é exibido - *Habilitado* ou *Desabilitado*, dependendo do que está definido para todo o grupo de usuários.
Status	O status do usuário é exibido - *Habilitado* ou *Desabilitado*, dependendo do que está definido para todo o grupo de usuários.
Provisionado	A data em que o usuário foi provisionado pela última vez é exibida. Usado para usuários criados por provisionamento JIT de LDAP/SAML.

Coluna	Descrição
Info	Informações sobre erros são exibidas. Um aviso amarelo é exibido para usuários sem grupos de usuários. Um aviso vermelho é exibido para usuários sem funções e para usuários sem funções e grupos de usuários.

Para configurar um novo usuário, clique no botão *Criar usuário* no canto superior direito.

Opções de edição em massa

Os botões abaixo da lista oferecem algumas opções de edição em massa:

- *Provisionar agora* - atualiza as informações do usuário a partir do LDAP (esta opção só é habilitada se um usuário LDAP for selecionado)
- *Redefinir segredo TOTP* - redefine os segredos TOTP do usuário para todos os métodos TOTP e exclui a sessão do usuário (esta opção só é habilitada se o MFA estiver habilitado; para usuários sem segredos TOTP, sua sessão não será excluída)
- *Desbloquear* - reabilita o acesso ao sistema para usuários bloqueados
- *Excluir* - exclui os usuários

Para usar essas opções, marque as caixas de seleção antes dos respectivos usuários e clique no botão desejado.

Usando o filtro

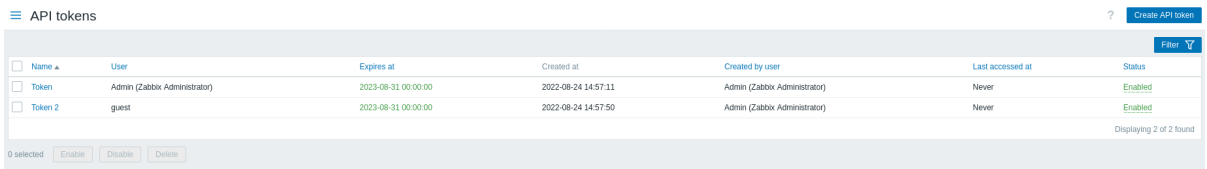
Você pode usar o filtro para exibir apenas os usuários que lhe interessam. Para melhor desempenho na pesquisa, os dados são pesquisados com macros não resolvidas.

O link *Filtro* está disponível acima da lista de usuários. Se você clicar nele, um filtro ficará disponível onde você pode filtrar usuários por nome de usuário, nome, sobrenome, função do usuário e grupo de usuários.

4 Tokens API

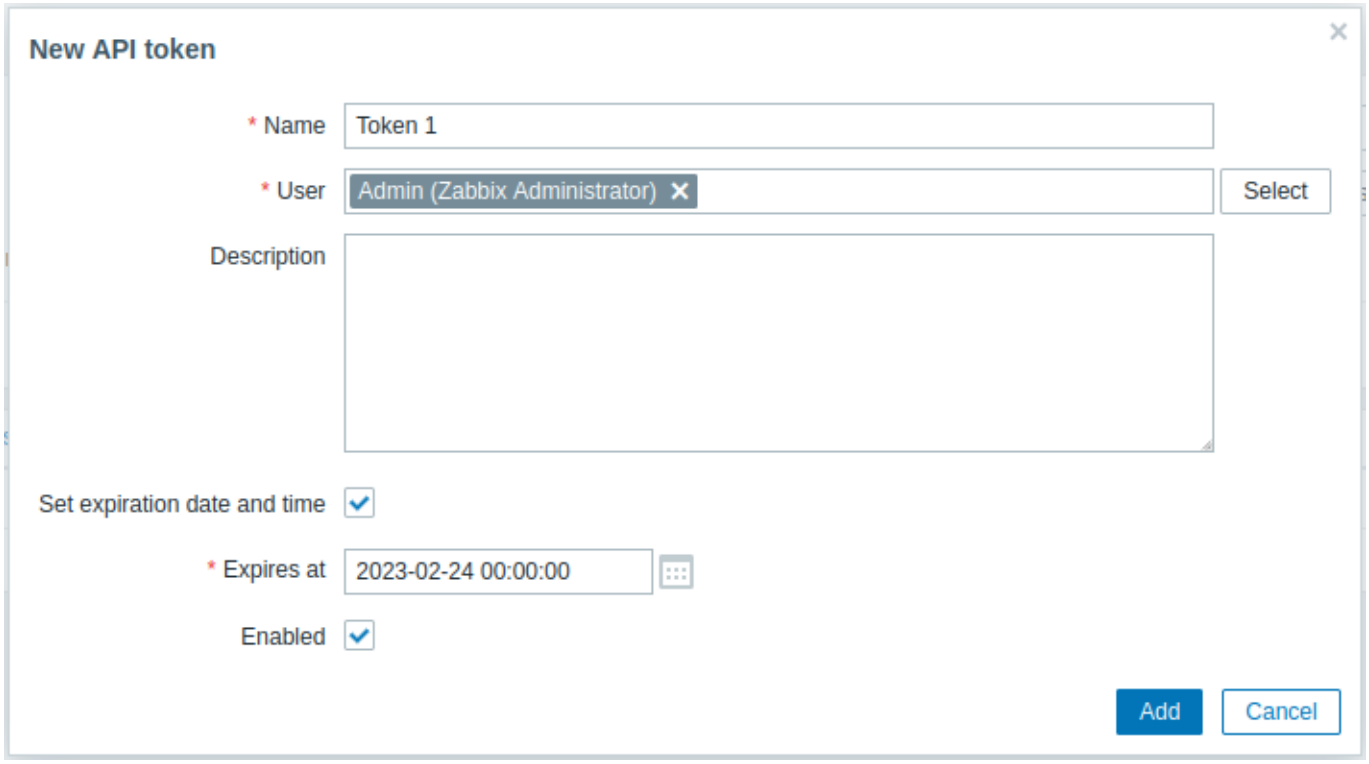
Visão geral

Esta seção permite criar e gerenciar tokens de API.



Você pode filtrar tokens de API por nome, usuários aos quais os tokens estão atribuídos, data de expiração, usuários que criaram tokens ou status (habilitado/desabilitado). Clique no status do token na lista para habilitar/desabilitar rapidamente um token. Você também pode habilitar/desabilitar vários tokens selecionando-os na lista e, em seguida, clicando nos botões *Habilitar/Desabilitar* abaixo da lista.

Para criar um novo token, pressione o botão *Criar token de API* no canto superior direito e preencha os campos obrigatórios na tela de configuração do token:



Parâmetro	Descrição
Nome	Nome visível do token.
Usuário	Usuário ao qual o token deve ser atribuído. Para selecionar rapidamente um usuário, comece a digitar o nome de usuário, nome ou sobrenome e, em seguida, selecione o usuário desejado na lista de preenchimento automático. Alternativamente, você pode pressionar o botão *Selecionar* e selecionar um usuário da lista completa de usuários. Um token pode ser atribuído apenas a um usuário.
Descrição	Descrição opcional do token.
Definir data e hora de expiração	Desmarque esta caixa se um token não deve ter uma data de expiração.
Data de expiração	Clique no ícone do calendário para selecionar a data de expiração do token ou insira a data manualmente no formato AAAA-MM-DD hh:mm:ss
Habilitado	Desmarque esta caixa se você precisar criar um token em estado desabilitado.

Pressione Adicionar para criar um token. Na próxima tela, copie e salve em um local seguro o valor do *Token de autenticação* **antes de fechar a página** e, em seguida, pressione Fechar. O token aparecerá na lista.

Warning:
O valor do *Token de autenticação* não pode ser visualizado novamente posteriormente. Ele só está disponível imediatamente após a criação de um token. Se você perder um token salvo, terá que regenerá-lo e, ao fazer isso, será criada uma nova string de autorização.

Clique no nome do token para editar o nome, descrição, configurações de data de expiração ou status do token. Observe que não é possível alterar para qual usuário o token está atribuído. Pressione o botão *Atualizar* para salvar as alterações. Se um token foi perdido ou exposto, você pode pressionar o botão *Regenerar* para gerar um novo valor de token. Uma caixa de diálogo de confirmação aparecerá, pedindo para confirmar esta operação, pois, após prosseguir, o token gerado anteriormente se tornará inválido.

Usuários sem acesso à seção de menu *Administração* podem ver e modificar detalhes dos tokens atribuídos a eles em *Perfil do usuário* → *Tokens de API* seção somente se *Gerenciar tokens de API* estiver permitido em suas permissões de função de usuário.

5 Autenticação

Visão geral

A seção *Usuários* → *Autenticação* permite especificar o método de autenticação do usuário para o Zabbix e os requisitos internos de senha.

Os métodos de autenticação disponíveis são autenticação interna, HTTP, LDAP, SAML e MFA.

Autenticação padrão

Por padrão, o Zabbix usa a autenticação **interna** do Zabbix para todos os usuários.

É possível alterar o método de autenticação padrão para LDAP em todo o sistema. Para isso, navegue até a guia *LDAP* e configure os parâmetros do LDAP, depois retorne à guia *Autenticação* e altere o seletor *Autenticação padrão* para LDAP.

Observe que o método de autenticação pode ser ajustado no nível do grupo de usuários. Mesmo que a autenticação LDAP esteja definida globalmente, alguns grupos de usuários ainda podem ser autenticados pelo Zabbix. Esses grupos devem ter o acesso ao frontend definido como Interno.

Também é possível habilitar a autenticação LDAP apenas para grupos de usuários específicos, se a autenticação interna for usada globalmente. Nesse caso, os detalhes de autenticação LDAP podem ser especificados e usados para grupos de usuários específicos cujo acesso ao frontend deve ser definido como LDAP. Se um usuário estiver incluído em pelo menos um grupo de usuários com autenticação LDAP, esse usuário não poderá usar o método de autenticação interna.

Os métodos de autenticação HTTP, SAML 2.0 e MFA podem ser usados além do método de autenticação padrão.

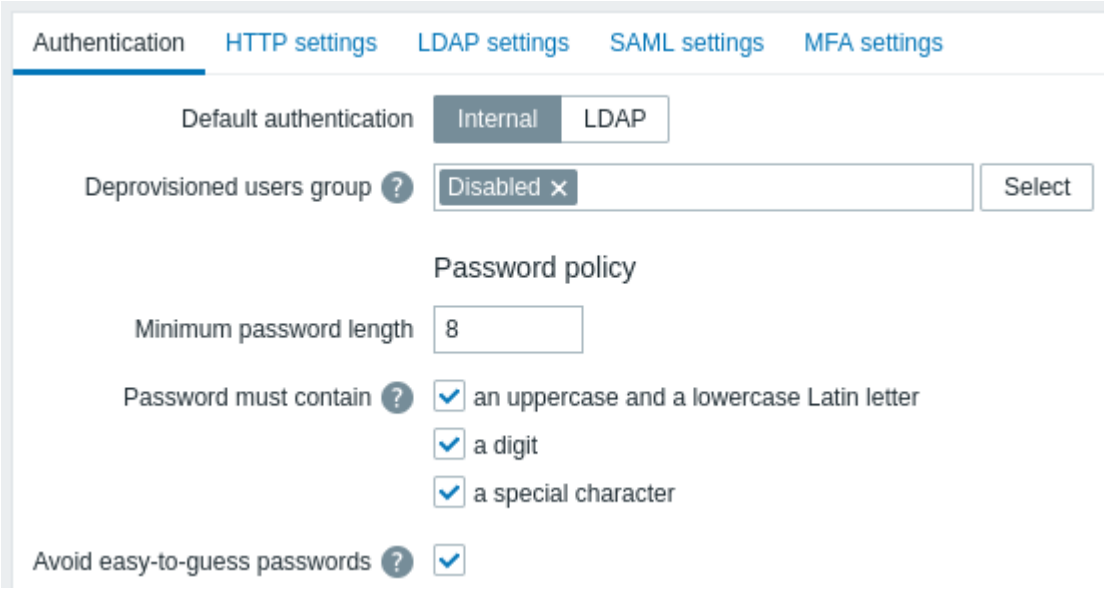
O Zabbix suporta o provisionamento just-in-time (JIT), que permite criar contas de usuário no Zabbix na primeira vez que um usuário externo se autentica e provisionar essas contas de usuário. O provisionamento JIT é suportado para LDAP e SAML.

Veja também:

- Autenticação HTTP
- Autenticação LDAP
- Autenticação SAML
- Autenticação MFA

Configuração

A aba *Autenticação* permite definir o método de autenticação padrão, especificar um grupo para usuários desprovisionados e definir requisitos de complexidade de senha para usuários do Zabbix.



Parâmetros de configuração:

Parâmetro	Descrição
Autenticação padrão	Selecione o método de autenticação padrão para o Zabbix - *Interno* ou *LDAP*.
Grupo de usuários desprovisionados	Especifique um grupo de usuários para usuários desprovisionados. Esta configuração é necessária apenas para provisionamento JIT, referente a usuários que foram criados no Zabbix a partir de sistemas LDAP ou SAML, mas que não precisam mais ser provisionados. Um grupo de usuários desabilitado deve ser especificado.
Comprimento mínimo da senha	Por padrão, o comprimento mínimo da senha é definido como 8. Faixa suportada: 1-70. Observe que senhas com mais de 72 caracteres serão truncadas.

Parâmetro	Descrição
A senha deve conter	Marque uma ou várias caixas de seleção para exigir o uso de caracteres especificados em uma senha: - uma letra latina maiúscula e minúscula - um dígito - um caractere especial Passe o mouse sobre o ponto de interrogação para ver uma dica com a lista de caracteres para cada opção.
Evitar senhas fáceis de adivinhar	Se marcado, uma senha será verificada em relação aos seguintes requisitos: - não deve conter o nome, sobrenome ou nome de usuário do usuário - não deve ser uma das senhas comuns ou específicas do contexto. A lista de senhas comuns e específicas do contexto é gerada automaticamente a partir da lista das "Top 100k passwords" da NCSC, da lista das "Top 1M passwords" da SecLists e da lista de senhas específicas do contexto do Zabbix. Usuários internos não poderão definir senhas incluídas nesta lista, pois tais senhas são consideradas fracas devido ao seu uso comum.

Alterações nos requisitos de complexidade de senha não afetarão as senhas de usuários existentes, mas se um usuário existente optar por alterar uma senha, a nova senha deverá atender aos requisitos atuais. Uma dica com a lista de requisitos será exibida ao lado do campo *Senha* no perfil do usuário e no formulário de configuração do usuário acessível no menu *Usuários* → *Usuários*.

1 HTTP

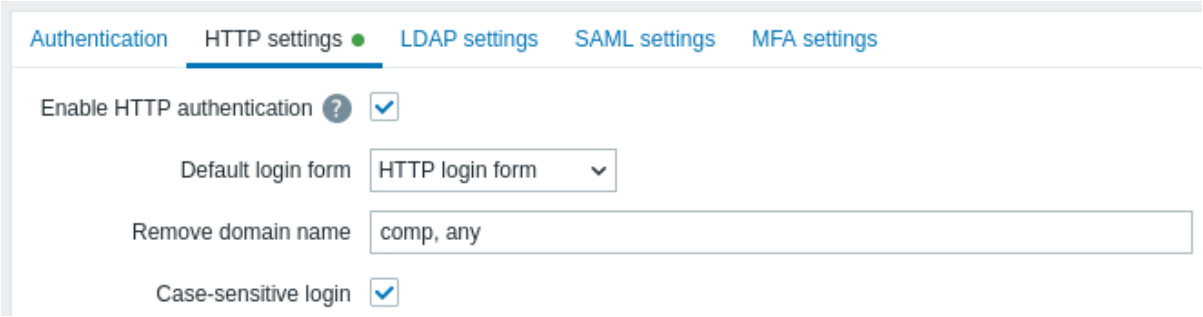
Visão geral

A autenticação baseada em HTTP ou servidor web (por exemplo: BasicAuthentication, NTLM/Kerberos) pode ser usada para verificar nomes de usuário e senhas. Observe que um usuário também deve existir no Zabbix, porém sua senha do Zabbix não será usada.

> **Attention:**
> Tenha cuidado! Certifique-se de que a autenticação do servidor web está configurada e funcionando corretamente antes de ativá-la.

A autenticação HTTP pode ser desabilitada no frontend configurando a opção correspondente na guia *Configurações HTTP* na seção *Usuários > Autenticação*. Quando a autenticação HTTP estiver desabilitada, a guia com as opções de autenticação HTTP não será exibida no frontend. Observe que reinstalar o frontend (executando o setup.php) redefinirá as configurações de autenticação, incluindo a configuração de autenticação HTTP.

Configuração



Parâmetros de configuração:

Parâmetro	Descrição
Habilitar autenticação HTTP	Marque a caixa de seleção para habilitar a autenticação HTTP. Ao passar o mouse sobre , será exibida uma caixa de dica alertando que, no caso de autenticação do servidor web, todos os usuários (mesmo com acesso ao frontend definido como LDAP/Interno) serão autenticados pelo servidor web, não pelo Zabbix.

Parâmetro	Descrição
Formulário de login padrão	Especifique se deseja direcionar usuários não autenticados para: **Formulário de login do Zabbix** - página padrão de login do Zabbix. **Formulário de login HTTP** - página de login HTTP. É recomendável habilitar a autenticação baseada no servidor web apenas para a página `index_http.php`. Se o *Formulário de login padrão* estiver definido como 'Página de login HTTP', o usuário será autenticado automaticamente se o módulo de autenticação do servidor web definir um login de usuário válido na variável `$_SERVER`. As chaves `$_SERVER` suportadas são `PHP_AUTH_USER`, `REMOTE_USER`, `AUTH_USER`.
Remover nome de domínio	Uma lista separada por vírgulas de nomes de domínio que devem ser removidos do nome de usuário. Exemplo: `comp,any` - se o nome de usuário for 'Admin@any', 'comp\Admin', o usuário será autenticado como 'Admin'; se o nome de usuário for 'notacompany\Admin', o login será negado.
Login sensível a maiúsculas e minúsculas	Desmarque a caixa de seleção para desabilitar o login sensível a maiúsculas e minúsculas para nomes de usuário (habilitado por padrão). Desabilitar o login sensível a maiúsculas e minúsculas permite, por exemplo, fazer login como "admin" mesmo que o usuário do Zabbix seja "Admin" ou "ADMIN". Observe que, se o login sensível a maiúsculas e minúsculas estiver desabilitado e houver vários usuários do Zabbix com nomes de usuário semelhantes (por exemplo, Admin e admin), o login para esses usuários sempre será negado com a seguinte mensagem de erro: "Falha na autenticação: as credenciais fornecidas não são exclusivas."

Note:
Para usuários internos que não conseguem fazer login usando credenciais HTTP (com o formulário de login HTTP definido como padrão), resultando no erro 401, você pode adicionar uma linha `ErrorDocument 401 /index.php?form=default` às diretivas de autenticação básica, o que redirecionará para o formulário de login padrão do Zabbix.

2 LDAP

Visão geral

A autenticação LDAP externa pode ser usada para verificar nomes de usuário e senhas.

A autenticação LDAP do Zabbix funciona pelo menos com Microsoft Active Directory e OpenLDAP.

Se apenas o login LDAP estiver configurado, o usuário também deve existir no Zabbix, no entanto, sua senha do Zabbix não será usada. Se a autenticação for bem-sucedida, o Zabbix corresponderá um nome de usuário local com o atributo de nome de usuário retornado pelo LDAP.

Provisionamento de usuários

É possível configurar o **provisionamento de usuários** JIT (just-in-time) para usuários LDAP. Nesse caso, não é necessário que um usuário já exista no Zabbix. A conta do usuário pode ser criada quando o usuário faz login no Zabbix pela primeira vez.

Quando um usuário LDAP insere seu login e senha LDAP, o Zabbix verifica o servidor LDAP *padrão* para saber se esse usuário existe. Se o usuário existir e ainda não tiver uma conta no Zabbix, um novo usuário é criado no Zabbix e o usuário pode fazer login.

Um usuário criado via provisionamento JIT é associado ao servidor LDAP (diretório) que está definido como padrão no momento da criação. Alterar o servidor LDAP padrão posteriormente não altera ou atualiza o servidor LDAP vinculado aos usuários que já foram provisionados.

Attention:
Se o provisionamento JIT estiver habilitado, um grupo de usuários para usuários desprovisionados deve ser especificado na aba *Autenticação*.

O provisionamento JIT também permite atualizar contas de usuários provisionadas com base em alterações no LDAP. Por exemplo, se um usuário for movido de um grupo LDAP para outro, o usuário também será movido de um grupo para outro no Zabbix; se um usuário for removido de um grupo LDAP, o usuário também será removido do grupo no Zabbix e, se não pertencer a nenhum outro grupo, adicionado ao grupo de usuários para usuários desprovisionados. As contas de usuários provisionadas são atualizadas com base no período de provisionamento configurado ou quando o usuário faz login no Zabbix.

Observe que o provisionamento em segundo plano é feito pelo frontend do Zabbix quando o usuário está interagindo com ele ou pelo menos tem uma página do frontend aberta no navegador. Não há processos de segundo plano dedicados para provisionar usuários.

O LDAP suporta três maneiras de se conectar ao diretório para autenticação e pesquisas:

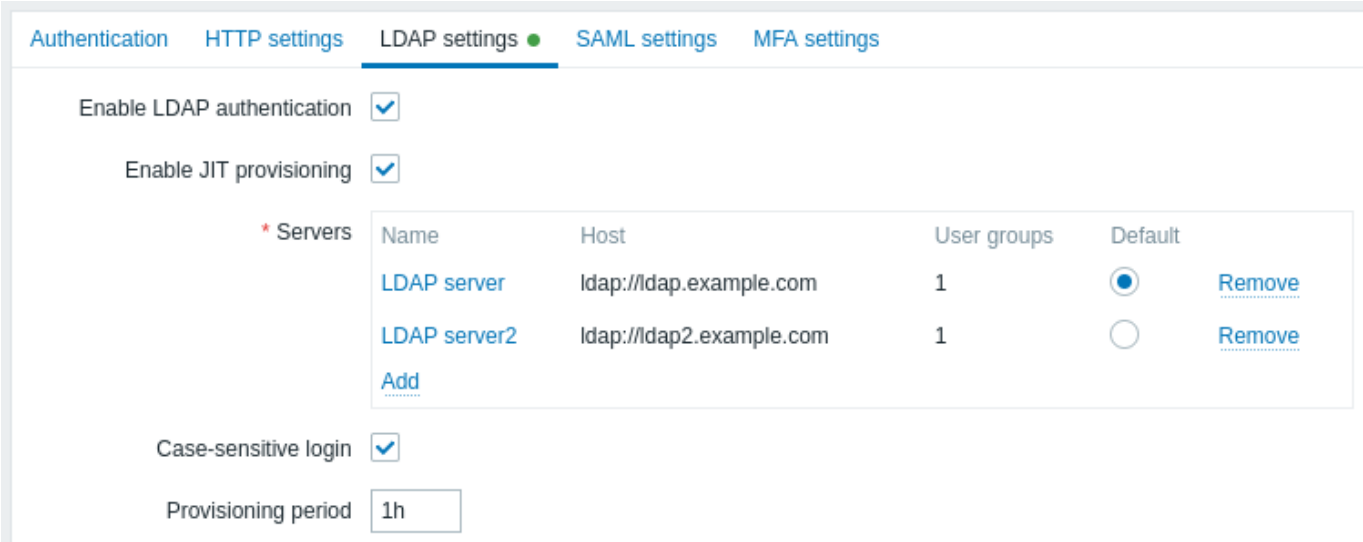
- Conexão anônima — nenhum *Bind DN* / *Bind password* é fornecido e o servidor LDAP permite consultas anônimas.
- Usuário de conexão dedicado (conta de serviço) — uma conta LDAP específica é definida em *Bind DN* / *Bind password* e é usada pelo Zabbix para pesquisa e provisionamento. Esta é a opção recomendada e mais flexível porque o Zabbix pode realizar pesquisas e provisionamento em segundo plano sem as credenciais do usuário final.
- Conexão direta do usuário — o Zabbix realiza a conexão usando as credenciais que o usuário insere no login (nenhum *Bind DN* / *Bind password* configurado); isso é configurado incluindo um placeholder como `uid=%{user}` no Base DN. Neste modo, o Zabbix só tem acesso à senha do usuário durante o login interativo. Como resultado, ações de provisionamento que exigem autenticação no LDAP fora da sessão de login do usuário (por exemplo, usando o botão *Provisionar agora* ou executando o provisionamento em segundo plano quando o usuário não está logado ativamente) não podem autenticar e, portanto, não funcionarão. O provisionamento e as atualizações para a conexão direta do usuário acontecem apenas no momento em que o usuário faz login.

Vários servidores

Vários servidores LDAP podem ser definidos, se necessário. Por exemplo, um servidor diferente pode ser usado para autenticar um grupo de usuários diferente. Uma vez que os servidores LDAP estejam configurados, na configuração do grupo de usuários torna-se possível selecionar o servidor LDAP necessário para o respectivo grupo de usuários.

Se um usuário estiver em vários grupos de usuários e vários servidores LDAP, o primeiro servidor na lista de servidores LDAP ordenados por nome em ordem crescente será usado para autenticação.

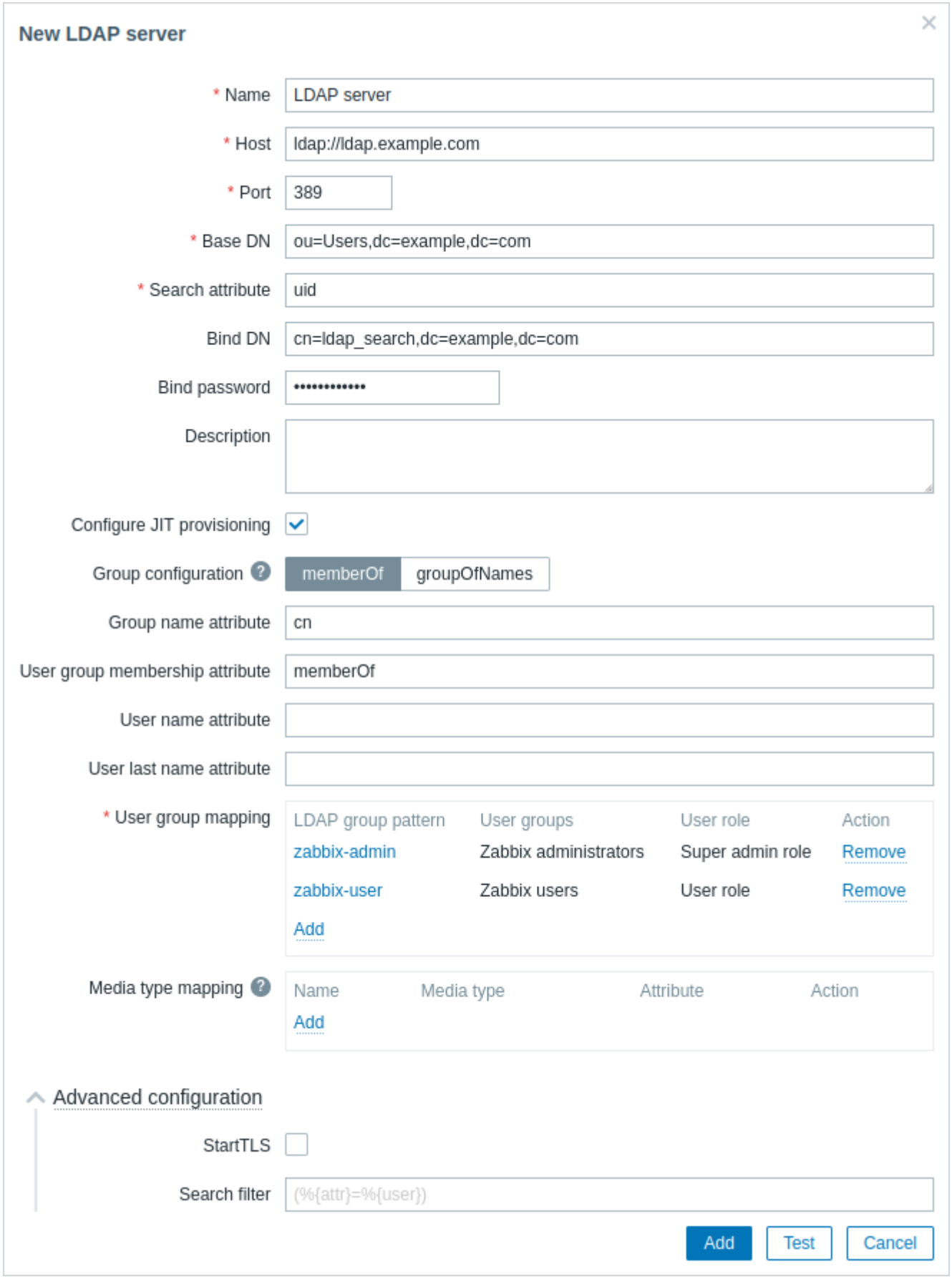
Configuração



Parâmetros de configuração:

Parâmetro	Descrição
Habilitar autenticação LDAP	Marque a caixa de seleção para habilitar a autenticação LDAP.
Habilitar provisionamento JIT	Marque a caixa de seleção para habilitar o provisionamento JIT.
Servidores	Clique em *Adicionar* para configurar um servidor LDAP (veja Configuração do servidor LDAP abaixo).
Login com diferenciação de maiúsculas/minúsculas	Desmarque a caixa de seleção para desabilitar o login com diferenciação de maiúsculas/minúsculas para nomes de usuário (habilitado por padrão). Desabilitar o login com diferenciação de maiúsculas/minúsculas permite, por exemplo, fazer login como ”admin” mesmo que o usuário do Zabbix seja ”Admin” ou ”ADMIN”. Observe que, se o login com diferenciação de maiúsculas/minúsculas estiver desabilitado e houver vários usuários do Zabbix com nomes de usuário semelhantes (por exemplo, Admin e admin), o login para esses usuários sempre será negado com a seguinte mensagem de erro: ”Falha na autenticação: as credenciais fornecidas não são exclusivas.”
Período de provisionamento	Defina o período de provisionamento, ou seja, com que frequência o usuário logado será provisionado enquanto estiver trabalhando com o frontend.

Configuração do servidor LDAP



Parâmetros de configuração do servidor LDAP:

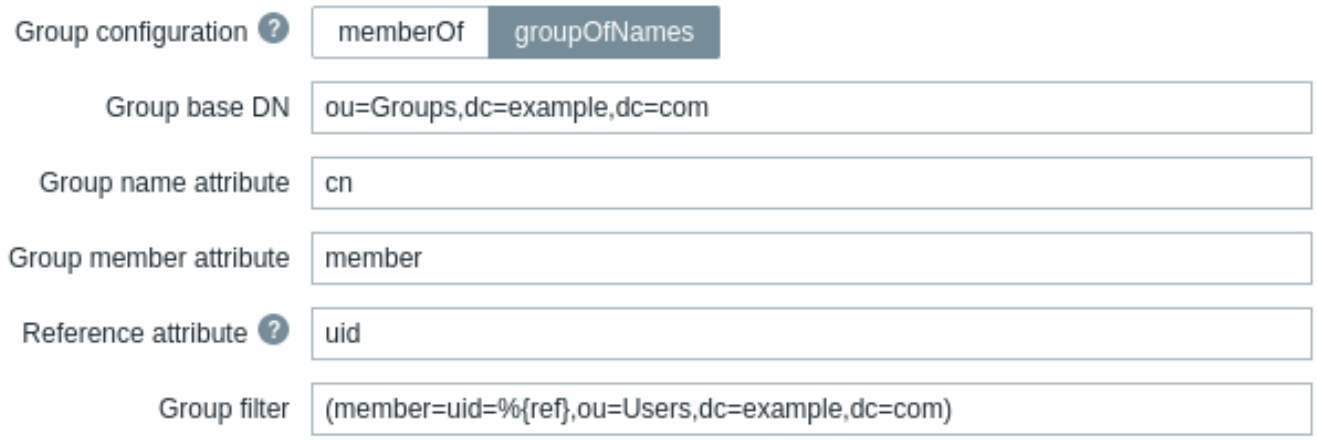
Parâmetro	Descrição
Nome	Nome do servidor LDAP na configuração do Zabbix.

Parâmetro	Descrição
Host	Nome do host, IP ou URI do servidor LDAP. Exemplos: ldap.example.com, 127.0.0.1, ldap://ldap.example.com Para servidor LDAP seguro, use o protocolo *ldaps* e o nome do host. Exemplo: ldaps://ldap.example.com Com OpenLDAP 2.x.x e posterior, pode-se usar um URI LDAP completo no formato ldap://hostname:port ou ldaps://hostname:port.
Porta	Porta do servidor LDAP. O padrão é 389. Para conexão LDAP segura, normalmente o número da porta é 636. Não é usada ao utilizar URIs LDAP completas.
Base DN	Caminho base para contas de usuário no servidor LDAP: ou=Users,ou=system (para OpenLDAP), DC=company,DC=com (para Microsoft Active Directory) uid=%{user},dc=example,dc=com (para binding direto do usuário, veja a observação abaixo)
Atributo de pesquisa	Atributo da conta LDAP usado para pesquisa: uid (para OpenLDAP), sAMAccountName (para Microsoft Active Directory)
Bind DN	Conta LDAP para binding e pesquisa no servidor LDAP, exemplos: uid=ldap_search,ou=system (para OpenLDAP), CN=ldap_search,OU=user_group,DC=company,DC=com (para Microsoft Active Directory) O binding anônimo também é suportado. Observe que o binding anônimo potencialmente expõe a configuração do domínio para usuários não autorizados (informações sobre usuários, computadores, servidores, grupos, serviços, etc.). Por motivos de segurança, desabilite binds anônimos nos hosts LDAP e use acesso autenticado.
Senha de binding	Senha LDAP da conta para binding e pesquisa no servidor LDAP.
Descrição	Descrição do servidor LDAP.
Configurar provisionamento JIT	Marque esta caixa para exibir opções relacionadas ao provisionamento JIT.
Configuração de grupo	Selecione o método de configuração de grupo: **memberOf** - pesquisando usuários e seu atributo de associação a grupo **groupOfNames** - pesquisando grupos através do atributo member Observe que memberOf é preferível por ser mais rápido; use groupOfNames se seu servidor LDAP não suportar `memberOf` ou se for necessário filtrar grupos.
Atributo de nome do grupo	Especifique o atributo para obter o nome do grupo de todos os objetos no atributo `memberOf` (veja o campo *Atributo de associação a grupo do usuário*) O nome do grupo é necessário para o mapeamento de grupos de usuários.
Atributo de associação a grupo do usuário	Especifique o atributo que contém informações sobre os grupos aos quais o usuário pertence (por exemplo, `memberOf`). Por exemplo, o atributo memberOf pode conter informações como: `memberOf=cn=zabbix-admin,ou=Groups,dc=example,dc=com` Este campo está disponível apenas para o método memberOf.
Atributo de nome do usuário	Especifique o atributo que contém o primeiro nome do usuário.
Atributo de sobrenome do usuário	Especifique o atributo que contém o sobrenome do usuário.
Mapeamento de grupo de usuários	Mapeie um padrão de grupo de usuários LDAP para grupo de usuários e função de usuário do Zabbix. Isto é necessário para determinar qual grupo/função de usuário o usuário provisionado receberá no Zabbix. Clique em *Adicionar* para adicionar um mapeamento. O campo *Padrão de grupo LDAP* suporta curingas. O nome do grupo deve corresponder a um grupo existente. Se um usuário LDAP corresponder a vários grupos de usuários do Zabbix, o usuário se tornará membro de todos eles. Se um usuário corresponder a várias funções de usuário do Zabbix, o usuário receberá aquela com o maior nível de permissão entre elas.
Mapeamento de tipo de mídia	Mapeie os atributos de mídia LDAP do usuário (por exemplo, e-mail) para mídia de usuário do Zabbix para envio de notificações (o valor do atributo é usado como o campo *Enviar para* da mídia).
Configuração avançada	Clique no rótulo *Configuração avançada* para exibir opções de configuração avançada (veja abaixo).

Parâmetro	Descrição
StartTLS	Marque a caixa para usar a operação StartTLS ao conectar-se ao servidor LDAP. A conexão falhará se o servidor não suportar StartTLS. StartTLS não pode ser usado com servidores que usam o protocolo *ldaps*.
Filtro de pesquisa	Defina uma string personalizada ao autenticar um usuário no LDAP. Os seguintes placeholders são suportados: `%{attr}` - nome do atributo de pesquisa (uid, sAMAccountName) `%{user}` - valor do nome de usuário a ser autenticado Por exemplo, para realizar uma pesquisa case-sensitive em um ambiente LDAP ou Microsoft Active Directory case-insensitive, a string pode ser definida da seguinte forma: `(%{attr}:caseExactMatch:=%{user})`. Se o filtro não for personalizado, o LDAP usará o padrão: `(%{attr}=%{user})`.

Note:
Para configurar um servidor LDAP para **binding direto do usuário**, adicione um atributo uid=%{user} ao parâmetro *Base DN* (por exemplo, *uid=%{user},dc=example,dc=com*) e deixe os parâmetros *BindDN* e *Senha de binding* em branco. Ao autenticar, o placeholder %{user} será substituído pelo nome de usuário inserido durante o login.
 Com o binding direto do usuário, o Zabbix só tem acesso às credenciais do usuário durante o login interativo. Portanto, tarefas de provisionamento que operam sem login interativo (por exemplo, o botão *Provisionar agora* ou o método de API `user.provision`) ignorarão usuários que se autenticam usando binding direto, pois o Zabbix não pode fazer o binding em nome deles. Use binding anônimo ou um usuário dedicado (conta de serviço) se precisar que o provisionamento e as pesquisas LDAP funcionem a partir do frontend sem exigir que o usuário final esteja logado.

Os seguintes campos são específicos para "groupOfNames" como método de *Configuração de grupo*:



Parâmetro	Descrição
Base DN do grupo	Caminho base para os grupos no servidor LDAP.
Atributo de nome do grupo	Especifique o atributo para obter o nome do grupo no caminho base especificado para grupos. O nome do grupo é necessário para o mapeamento de grupos de usuários.
Atributo de membro do grupo	Especifique o atributo que contém informações sobre os membros do grupo no LDAP (por exemplo, `member`).
Atributo de referência	Especifique o atributo de referência para o filtro de grupo (veja o campo *Filtro de grupo*). Depois, use `%{ref}` no filtro de grupo para obter valores para o atributo especificado aqui.
Filtro de grupo	Especifique o filtro para recuperar o grupo do qual o usuário é membro. Por exemplo, `(member=uid=%{ref},ou=Users,dc=example,dc=com)` corresponderá ao "User1" se o atributo member do grupo for `uid=User1,ou=Users,dc=example,dc=com` e retornará o grupo do qual "User1" é membro.

Warning:
Em caso de problemas com certificados, para fazer uma conexão LDAP segura (ldaps) funcionar, pode ser necessário adicionar uma linha `TLS_REQCERT allow` ao arquivo de configuração /etc/openldap/ldap.conf. Isso pode diminuir a segurança da conexão com o catálogo LDAP.

Note:
Recomenda-se criar uma conta LDAP separada (*Bind DN*) para realizar binding e pesquisa no servidor LDAP com privilégios mínimos no LDAP, em vez de usar contas de usuários reais (usadas para login no frontend do Zabbix).
 Essa abordagem oferece mais segurança e não exige alteração da *Senha de binding* quando o usuário altera sua própria senha no servidor LDAP.
 Na tabela acima, é o nome da conta *ldap_search*.

Testando o acesso

O botão *Testar* permite testar o acesso do usuário:

Parâmetro	Descrição
Login	Nome de usuário LDAP a ser testado (preenchido com o nome de usuário atual do frontend do Zabbix). Este nome de usuário deve existir no servidor LDAP. O Zabbix não ativará a autenticação LDAP se não for possível autenticar o usuário de teste.
Senha do usuário	Senha do usuário LDAP a ser testada.

3 SAML (Linguagem de Marcação de Asserção de Segurança)

Visão geral

A autenticação SAML 2.0 pode ser usada para fazer login no Zabbix.

Se apenas o login SAML estiver configurado, o usuário também deverá existir no Zabbix, porém, sua senha do Zabbix não será utilizada. Se a autenticação for bem-sucedida, o Zabbix irá comparar um nome de usuário local com o atributo de nome de usuário retornado pelo SAML.

Provisionamento de usuários

É possível configurar o **provisionamento de usuários** JIT (just-in-time) para usuários SAML. Neste caso, não é necessário que um usuário já exista no Zabbix. A conta do usuário pode ser criada quando o usuário fizer login no Zabbix pela primeira vez.

Attention:
Se o provisionamento JIT estiver habilitado, um grupo de usuários para usuários desprovisionados deve ser especificado na aba *Autenticação*.

Além do provisionamento JIT, também é possível habilitar e configurar o provisionamento SCIM (System for Cross-domain Identity Management) - gerenciamento *contínuo* de contas de usuários para aqueles usuários que foram criados pelo provisionamento de usuários. O provisionamento SCIM requer um token de API do Zabbix (com permissões de Super admin) para autenticação no Zabbix.

Por exemplo, se um usuário for movido de um grupo SAML para outro, o usuário também será movido de um grupo para outro no Zabbix; se um usuário for removido de um grupo SAML, o usuário também será removido do grupo no Zabbix e, se não pertencer a nenhum outro grupo, adicionado ao grupo de usuários para usuários desprovisionados.

Se o SCIM estiver habilitado e configurado, um usuário SAML será provisionado no momento em que fizer login no Zabbix e continuamente atualizado com base nas alterações no SAML. Usuários SAML já existentes não serão provisionados, e apenas usuários provisionados serão atualizados. Observe que apenas mídias válidas serão adicionadas a um usuário quando o usuário for provisionado ou atualizado.

Se o SCIM não estiver habilitado, um usuário SAML será provisionado (e posteriormente atualizado) no momento em que fizer login no Zabbix.

Note:
Se a autenticação SAML estiver habilitada, os usuários poderão escolher entre fazer login localmente ou via SAML single sign-on. Se o provisionamento JIT for utilizado, então apenas o single sign-on será possível.

Configurando o provedor de identidade

Para funcionar com o Zabbix, um provedor de identidade SAML (onelogin.com, auth0.com, okta.com, etc.) precisa ser configurado da seguinte forma:

- *Assertion Consumer URL* deve ser definido como `<path_to_zabbix_ui>/index_sso.php?acs`
- *Single Logout URL* deve ser definido como `<path_to_zabbix_ui>/index_sso.php?sls`

Exemplos de <path_to_zabbix_ui>: https://example.com/zabbix/ui, http://another.example.com/zabbix, http://<any_public_ip_address>/zabbix

Configurando o Zabbix

Attention:
É necessário instalar o php-openssl se você quiser usar autenticação SAML no frontend.

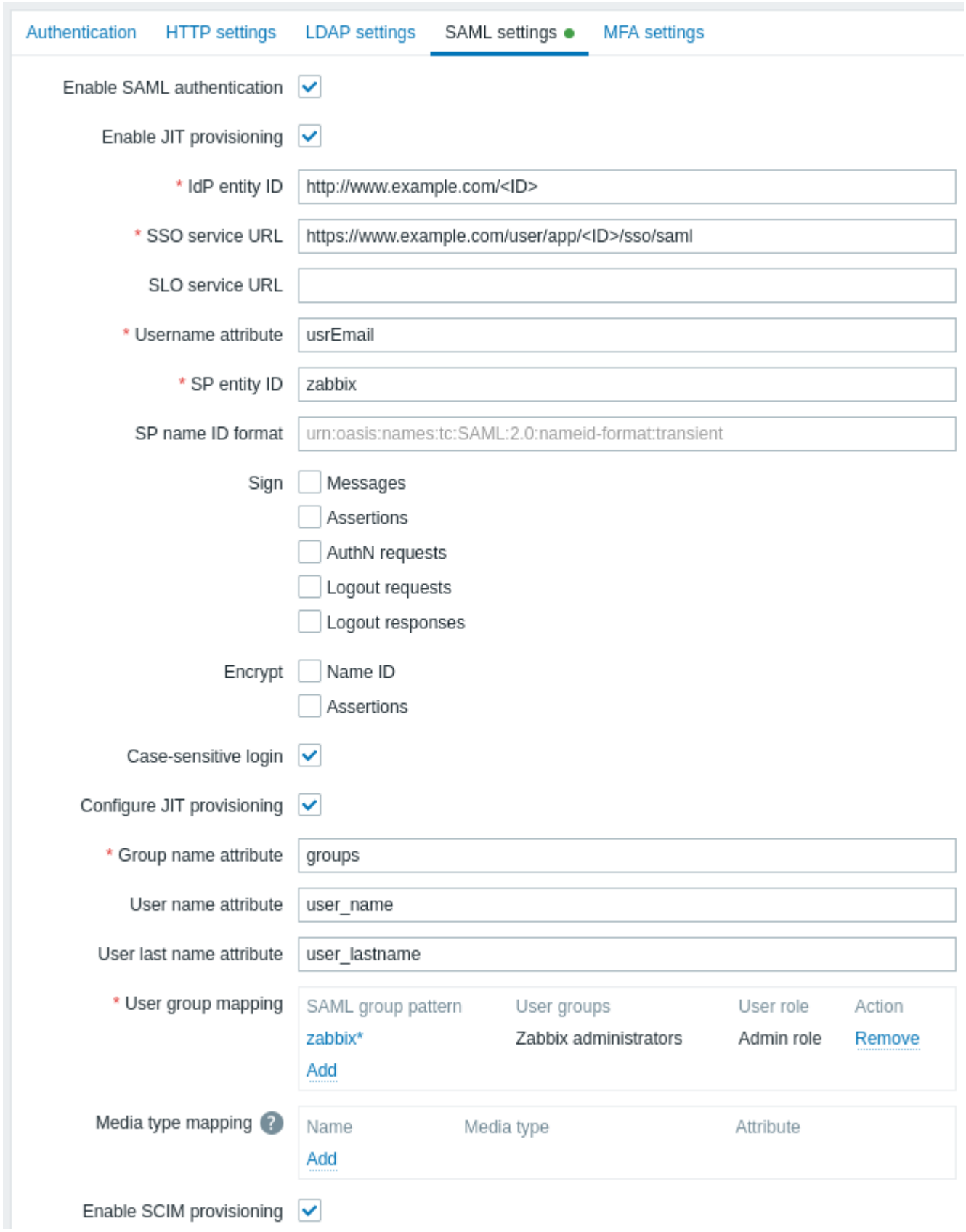
Para usar a autenticação SAML, o Zabbix deve ser configurado da seguinte forma:

1. A chave privada e o certificado devem ser armazenados em *ui/conf/certs/*, a menos que caminhos personalizados sejam fornecidos em zabbix.conf.php.

Por padrão, o Zabbix procurará nos seguintes locais:

- ui/conf/certs/sp.key - arquivo de chave privada do SP
- ui/conf/certs/sp.crt - arquivo de certificado do SP
- ui/conf/certs/idp.crt - arquivo de certificado do IDP

2. Todas as configurações mais importantes podem ser feitas no frontend do Zabbix. No entanto, é possível especificar configurações adicionais no arquivo de configuração.



Parâmetros de configuração disponíveis no frontend do Zabbix:

Parâmetro	Descrição
Habilitar autenticação SAML	Marque a caixa de seleção para habilitar a autenticação SAML.
Habilitar provisionamento JIT	Marque a caixa de seleção para habilitar o provisionamento de usuário JIT.
IDP entity ID	O identificador exclusivo da entidade dentro do provedor de identidade SAML.
SSO service URL	A URL para a qual os usuários serão redirecionados ao fazer login.

Parâmetro	Descrição
SLO service URL	A URL para a qual os usuários serão redirecionados ao fazer logout. Se deixado em branco, o serviço SLO não será usado.
Username attribute	Atributo SAML a ser usado como nome de usuário ao fazer login no Zabbix. A lista de valores suportados é determinada pelo provedor de identidade. Exemplos: uid userprincipalname samaccountname username userusername urn:oid:0.9.2342.19200300.100.1.1 urn:oid:1.3.6.1.4.1.5923.1.1.1.13 urn:oid:0.9.2342.19200300.100.1.44
SP entity ID	O identificador exclusivo do provedor de serviço (se não corresponder, a operação será rejeitada). É possível especificar uma URL ou qualquer string de dados.
SP name ID format	Solicite um formato específico de name ID na resposta. Exemplos: 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
Sign	Marque as caixas de seleção para selecionar as entidades para as quais a assinatura SAML deve ser habilitada: *Mensagens* *Declarações* *Solicitações AuthN* *Solicitações de logout* *Respostas de logout*
Encrypt	Marque as caixas de seleção para selecionar as entidades para as quais a criptografia SAML deve ser habilitada: *Name ID* *Declarações*
Login com diferenciação de maiúsculas/minúsculas	Desmarque a caixa de seleção para desabilitar o login com diferenciação de maiúsculas/minúsculas para nomes de usuário (habilitado por padrão). Desabilitar o login com diferenciação de maiúsculas/minúsculas permite, por exemplo, fazer login como ”admin” mesmo que o usuário do Zabbix seja ”Admin” ou ”ADMIN”. Observe que, se o login com diferenciação de maiúsculas/minúsculas estiver desabilitado e houver vários usuários do Zabbix com nomes de usuário semelhantes (por exemplo, Admin e admin), o login para esses usuários será sempre negado com a seguinte mensagem de erro: ”Falha na autenticação: as credenciais fornecidas não são exclusivas.”
Configurar provisionamento JIT	Marque esta caixa de seleção para exibir opções relacionadas ao provisionamento de usuário JIT.
Atributo de nome do grupo	Especifique o atributo de nome do grupo para o provisionamento de usuário JIT.
Atributo de nome do usuário	Especifique o atributo de nome do usuário para o provisionamento de usuário JIT.
Atributo de sobrenome do usuário	Especifique o atributo de sobrenome do usuário para o provisionamento de usuário JIT.
Mapeamento de grupo de usuário	Mapeie um padrão de grupo de usuário SAML para grupo de usuário e função de usuário do Zabbix. Isto é necessário para determinar qual grupo/função de usuário o usuário provisionado terá no Zabbix. Clique em *Adicionar* para adicionar um mapeamento. O campo *Padrão de grupo SAML* suporta curingas. O nome do grupo deve corresponder a um grupo existente. Se um usuário SAML corresponder a vários grupos de usuários do Zabbix, o usuário se tornará membro de todos eles. Se um usuário corresponder a várias funções de usuário do Zabbix, o usuário terá o nível de permissão mais alto entre elas.

Parâmetro	Descrição
Mapeamento de tipo de mídia	Mapeie os atributos de mídia SAML do usuário (por exemplo, e-mail) para a mídia de usuário do Zabbix para envio de notificações.
Habilitar provisionamento SCIM	Marque esta caixa de seleção para habilitar o provisionamento SCIM 2.0.

Veja exemplos de configuração de provedores de identidade SAML para login e provisionamento de usuários no Zabbix com:

- Microsoft Azure AD
- Okta
- Onelogin

Observações sobre o provisionamento SCIM

Para o provisionamento SCIM, especifique o caminho para o frontend do Zabbix e adicione api_scim.php a ele, no lado do provedor de identidade, ou seja:

`https://<caminho-para-o-zabbix-ui>/api_scim.php`

Os atributos de usuário que são usados no Zabbix (nome de usuário, nome, sobrenome e atributos de mídia) precisam ser adicionados como atributos personalizados e, se necessário, o namespace externo deve ser o mesmo do esquema de usuário: `urn:ietf:params:scim:schemas:core:2.0:User`.

Configurações avançadas

Parâmetros SAML adicionais podem ser configurados no arquivo de configuração do frontend do Zabbix (*zabbix.conf.php*):

- $SSO['SP_KEY'] = '<caminho para o arquivo de chave privada do SP>';
- $SSO['SP_CERT'] = '<caminho para o arquivo de certificado do SP>';
- $SSO['IDP_CERT'] = '<caminho para o arquivo de certificado do IDP>';
- $SSO['SETTINGS']

Note:
O array `$SSO['SETTINGS']` deve seguir a mesma estrutura esperada pela biblioteca *SAML PHP Toolkit* (fornecida com o Zabbix). Para uma descrição completa das opções de configuração disponíveis, consulte a documentação oficial da biblioteca.

Apenas as seguintes opções podem ser definidas como parte de $SSO['SETTINGS']:

- *strict*
- *baseurl*
- *compress*
- *contactPerson*
- *organization*
- *sp* (apenas opções especificadas nesta lista)
  - *attributeConsumingService*
  - *x509certNew*
- *idp* (apenas opções especificadas nesta lista)
  - *singleLogoutService* (apenas uma opção)
    - *responseUrl*
  - *certFingerprint*
  - *certFingerprintAlgorithm*
  - *x509certMulti*
- *security* (apenas opções especificadas nesta lista)
  - *signMetadata*
  - *wantNameId*
  - *requestedAuthnContext*
  - *requestedAuthnContextComparison*
  - *wantXMLValidation*
  - *relaxDestinationValidation*
  - *destinationStrictlyMatches*
  - *rejectUnsolicitedResponsesWithInResponseTo*
  - *signatureAlgorithm*
  - *digestAlgorithm*
  - *lowercaseUrlencoding*

Todas as outras opções serão obtidas do banco de dados e não podem ser sobrescritas. A opção *debug* será ignorada.

Além disso, se a interface do Zabbix estiver atrás de um proxy ou balanceador de carga, a opção personalizada *use_proxy_headers* pode ser usada:

- *false* (padrão) - ignora a opção;
- *true* - usa os cabeçalhos HTTP X-Forwarded-* para construir a URL base.

Se estiver usando um balanceador de carga para conectar à instância do Zabbix, onde o balanceador de carga usa TLS/SSL e o Zabbix não, você deve indicar os parâmetros 'baseurl', 'strict' e 'use_proxy_headers' da seguinte forma:

```
$SSO['SETTINGS'] = [
    'strict' => false,
    'baseurl' => 'https://zabbix.example.com/zabbix/',
    'use_proxy_headers' => true
];
```

Exemplo de configuração:

```
$SSO['SETTINGS'] = [
    'security' => [
        'signatureAlgorithm' => 'http://www.w3.org/2001/04/xmldsig-more#rsa-sha384'
        'digestAlgorithm' => 'http://www.w3.org/2001/04/xmldsig-more#sha384',
        // ...
    ],
    // ...
];
```

Configuração do frontend com Kerberos/ADFS

O arquivo de configuração do frontend do Zabbix (*zabbix.conf.php*) pode ser usado para configurar o SSO com autenticação Kerberos e ADFS:

```
$SSO['SETTINGS'] = [
    'security' => [
        'requestedAuthnContext' => [
            'urn:oasis:names:tc:SAML:2.0:ac:classes:Kerberos',
        ],
        'requestedAuthnContextComparison' => 'exact'
    ]
];
```

Neste caso, no campo de configuração SAML *SP name ID* defina:

`urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified`

4 MFA

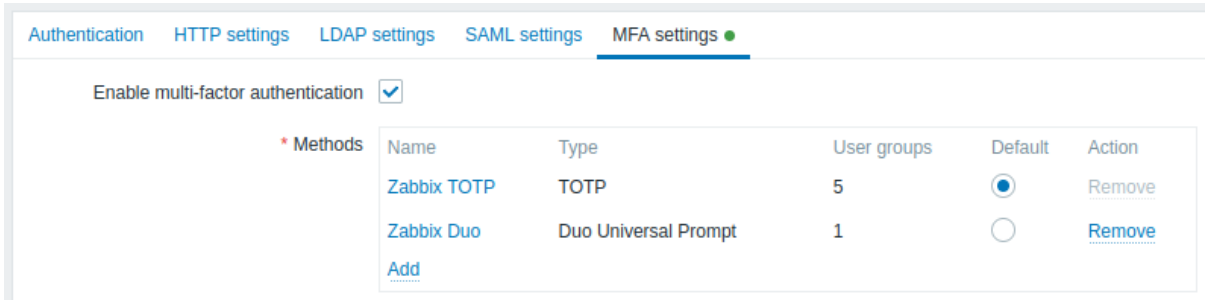
Visão geral

A autenticação multifator (MFA) pode ser usada para acessar o Zabbix, fornecendo uma camada adicional de segurança além de apenas um nome de usuário e senha.

Com o MFA, o usuário deve existir no Zabbix, deve fornecer as credenciais do Zabbix ao fazer login e também deve provar sua identidade por outros meios, geralmente um código gerado por um aplicativo autenticador no telefone do usuário.

Vários métodos de MFA estão disponíveis, permitindo que os usuários escolham a opção que melhor se adapta aos seus requisitos de segurança e preferências. Esses métodos são Senha Única Baseada em Tempo (TOTP) e Duo Universal Prompt.

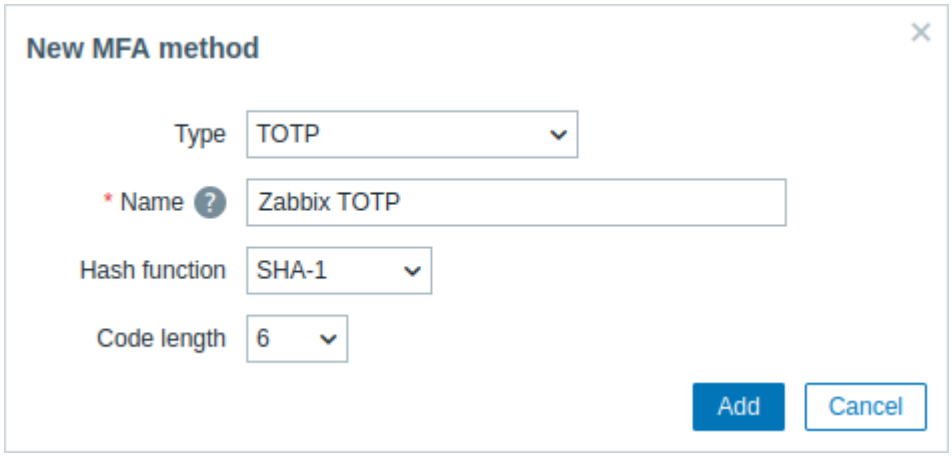
Configuração



Parâmetros de configuração:

Parâmetro	Descrição
Habilitar autenticação multifator	Marque a caixa de seleção para habilitar a autenticação multifator.
Métodos	Clique em *Adicionar* para configurar um método MFA (veja configuração do método abaixo).

Configuração do método



Parâmetros de configuração do método:

Parâmetro	Descrição
Tipo	Selecione o tipo de método MFA: **TOTP** - use um aplicativo autenticador para gerar senhas únicas baseadas em tempo; **Duo Universal Prompt** - use o serviço de autenticação Duo para fornecer autenticação multifator.
Nome	Digite um nome que será exibido como o nome da conta para todos os usuários MFA nos aplicativos autenticadores (por exemplo, ”Zabbix”).
Função hash	Selecione a função hash (SHA-1, SHA-256 ou SHA-512) para gerar códigos TOTP. Este parâmetro está disponível se o tipo de método MFA estiver definido como ”TOTP”. Observe que escolher SHA-256 ou SHA-512 pode limitar significativamente a compatibilidade, pois muitos aplicativos atualmente não suportam essas funções.
Comprimento do código	Selecione o comprimento do código de verificação (6 ou 8). Este parâmetro está disponível se o tipo de método MFA estiver definido como ”TOTP”.
Hostname da API	Digite o hostname da API fornecido pelo serviço de autenticação Duo. Este parâmetro está disponível se o tipo de método MFA estiver definido como ”Duo Universal Prompt”.
ID do cliente	Digite o ID do cliente fornecido pelo serviço de autenticação Duo. Este parâmetro está disponível se o tipo de método MFA estiver definido como ”Duo Universal Prompt”.
Segredo do cliente	Digite o segredo do cliente fornecido pelo serviço de autenticação Duo. Este parâmetro está disponível se o tipo de método MFA estiver definido como ”Duo Universal Prompt”.

Exemplos de configuração

Esta seção fornece exemplos de configuração de MFA usando Senha Única Baseada em Tempo (TOTP) e Duo Universal Prompt.

TOTP

Para TOTP, os usuários devem verificar sua identidade usando um aplicativo autenticador (por exemplo, o aplicativo Google Authenticator).

1. Vá para as configurações de MFA no Zabbix em *Usuários* → *Autenticação* e habilite a autenticação multifator.

2. Adicione um novo método de MFA com a seguinte configuração:

   - Tipo: TOTP
   - Nome: Zabbix TOTP
   - Função de hash: SHA-1
   - Comprimento do código: 6

3. Clique no botão *Adicionar* e depois no botão *Atualizar*.

4. Vá para *Usuários* → *Grupos de usuários* e crie um novo grupo de usuários com a seguinte configuração:

   - Nome do grupo: TOTP group
   - Usuários: Admin
   - Autenticação multifator: Padrão (ou ”Zabbix TOTP” se não estiver definido como padrão)

5. Saia do Zabbix e faça login novamente usando suas credenciais. Após o login bem-sucedido, você será solicitado a se inscrever na MFA, exibindo um código QR e uma chave secreta.

6. Escaneie o código QR ou insira a chave secreta no aplicativo Google Authenticator. O aplicativo irá gerar um código de verificação que você deve inserir para concluir o processo de login.

7. Para logins subsequentes, recupere o código de verificação no aplicativo Google Authenticator e insira-o durante o login.

Duo Universal Prompt

Para o Duo Universal Prompt, os usuários devem verificar sua identidade usando o aplicativo autenticador Duo Mobile.

Attention:
O método de MFA Duo Universal Prompt requer a instalação da extensão php-curl, acesso ao Zabbix via HTTPS e permissão para conexões de saída para os servidores Duo. Além disso, se você habilitou o Content Security Policy (CSP) no servidor web, certifique-se de adicionar "duo.com" à diretiva CSP no arquivo de configuração do seu host virtual.

1. Cadastre-se para uma conta de administrador Duo gratuita em Duo Signup.

2. Abra o Painel de Administração Duo, vá para *Applications* → *Protect an Application*, pesquise o aplicativo *Web SDK* e clique em *Protect*.

3. Anote as credenciais (Client ID, Client secret, API hostname) necessárias para configurar o método de MFA no Zabbix.

4. Vá para as configurações de MFA no Zabbix em *Usuários* → *Autenticação* e habilite a autenticação multifator.

5. Adicione um novo método de MFA com a seguinte configuração:

- Tipo: Duo Universal Prompt
- Nome: Zabbix Duo
- API hostname: (use o API hostname do Duo)
- Client ID: (use o Client ID do Duo)
- Client secret: (use o Client secret do Duo)

6. Clique no botão *Adicionar* e depois no botão *Atualizar*.

7. Vá para *Usuários* → *Grupos de usuários* e crie um novo grupo de usuários com a seguinte configuração:

- Nome do grupo: Duo group
- Usuários: Admin
- Autenticação multifator: Padrão (ou "Zabbix Duo" se não estiver definido como padrão)

8. Saia do Zabbix e faça login novamente usando suas credenciais. Após o login bem-sucedido, você será solicitado a se inscrever no MFA e será redirecionado para o Duo. Conclua a configuração do Duo e verifique seu usuário com o aplicativo Duo do seu telefone para fazer login.

9. Para logins subsequentes, use o método de MFA apropriado fornecido pelo aplicativo Duo (como recuperar um código de verificação, responder a notificações push ou usar chaves físicas) e insira as informações necessárias durante o login.

5 Funções do usuário

Visão geral

Na seção *Usuários* > *Funções de usuário* você pode criar funções de usuário.

As funções de usuário permitem criar permissões detalhadas com base no tipo de usuário selecionado inicialmente (*Usuário*, *Admin*, *Super admin*).

Ao selecionar um tipo de usuário, todas as permissões disponíveis para esse tipo de usuário são concedidas (marcadas por padrão).

As permissões só podem ser revogadas do subconjunto que está disponível para o tipo de usuário; elas não podem ser estendidas além do que está disponível para o tipo de usuário.

As caixas de seleção para permissões indisponíveis ficam acinzentadas; os usuários não poderão acessar o elemento mesmo inserindo uma URL direta para esse elemento no navegador.

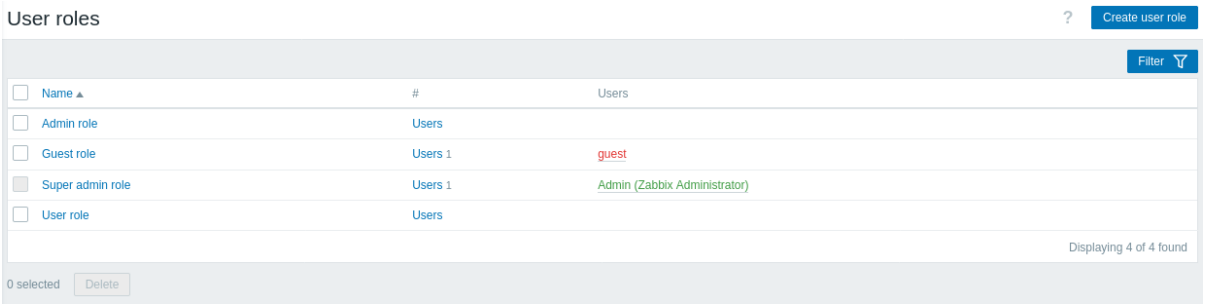
Note:
Restringir o acesso a alguns elementos da interface do usuário apenas impede a abertura dessa página — não remove a capacidade de acessar os dados subjacentes em outras partes da interface.

As funções de usuário podem ser atribuídas a usuários do sistema. Cada usuário pode ter apenas uma função atribuída.

Funções de usuário padrão

Por padrão, o Zabbix é configurado com quatro funções de usuário, que possuem um conjunto predefinido de permissões:

- Função de convidado
- Função de usuário
- Função de administrador
- Função de superadministrador

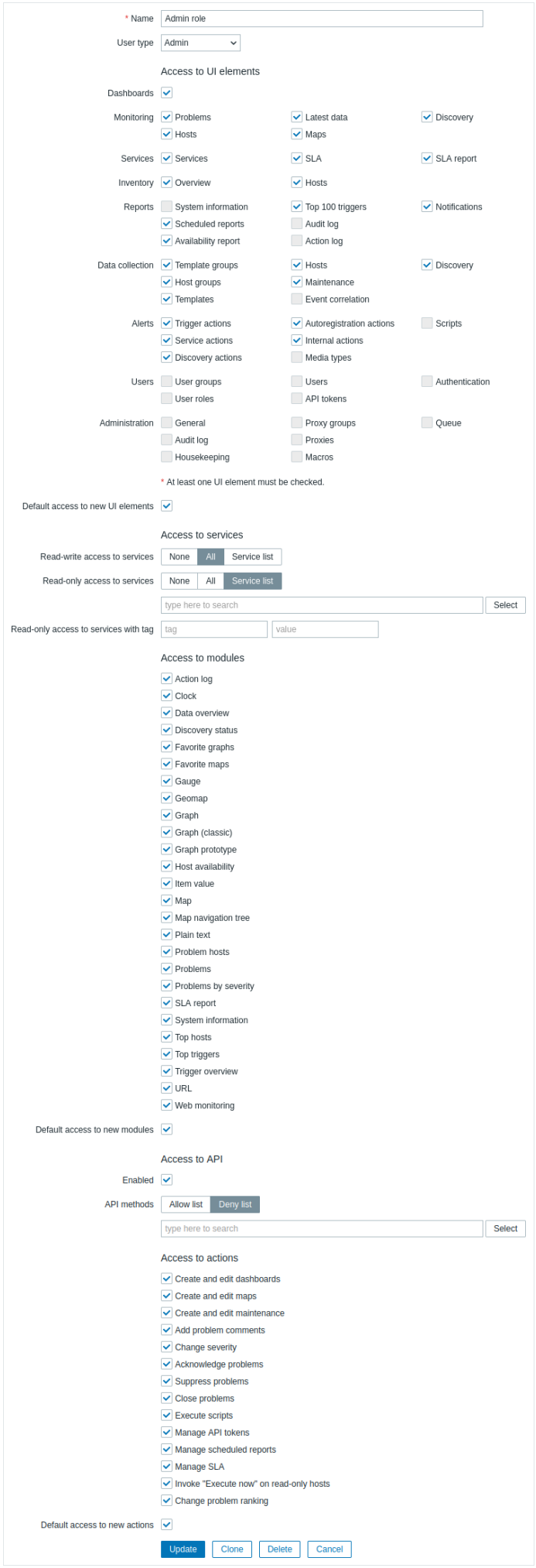


Estas são baseadas nos principais tipos de usuário do Zabbix. A lista de todos os usuários atribuídos à respectiva função é exibida. Os usuários incluídos em grupos desabilitados são indicados em vermelho. A *Função de convidado* é uma função do tipo usuário com a única permissão de visualizar algumas seções do frontend.

Note:
A *Função de superadministrador* padrão não pode ser modificada ou excluída, porque deve existir pelo menos um usuário superadministrador com privilégios ilimitados no Zabbix. Usuários do tipo *superadministrador* podem modificar as configurações de sua própria função, mas não o tipo de usuário.

Configuração

Para criar uma nova função, clique no botão *Criar função de usuário* no canto superior direito. Para atualizar uma função existente, clique no nome da função para abrir o formulário de configuração.


* Name
Admin role
User type
Admin
Access to UI elements
Dashboards
Monitoring
Problems
Latest data
Discovery
Hosts
Maps
Services
Services
SLA
SLA report
Inventory
Overview
Hosts
Reports
System information
Top 100 triggers
Notifications
Scheduled reports
Audit log
Availability report
Action log
Data collection
Template groups
Hosts
Discovery
Host groups
Maintenance
Templates
Event correlation
Alerts
Trigger actions
Autoregistration actions
Scripts
Service actions
Internal actions
Discovery actions
Media types
Users
User groups
Users
Authentication
User roles
API tokens
Administration
General
Proxy groups
Queue
Audit log
Proxies
Housekeeping
Macros
* At least one UI element must be checked.
Default access to new UI elements
Access to services
Read-write access to services
None
All
Service list
Read-only access to services
None
All
Service list
type here to search
Select
Read-only access to services with tag
tag
value
Access to modules
Action log
Clock
Data overview
Discovery status
Favorite graphs
Favorite maps
Gauge
Geomap
Graph
Graph (classic)
Graph prototype
Host availability
Item value
Map
Map navigation tree
Plain text
Problem hosts
Problems
Problems by severity
SLA report
System information
Top hosts
Top triggers
Trigger overview
URL
Web monitoring
Default access to new modules
Access to API
Enabled
API methods
Allow list
Deny list
type here to search
Select
Access to actions
Create and edit dashboards
Create and edit maps
Create and edit maintenance
Add problem comments
Change severity
Acknowledge problems
Suppress problems
Close problems
Execute scripts
Manage API tokens
Manage scheduled reports
Manage SLA
Invoke "Execute now" on read-only hosts
Change problem ranking
Default access to new actions
Update
Clone
Delete
Cancel

As permissões disponíveis são exibidas. Para revogar uma determinada permissão, desmarque sua caixa de seleção.

As permissões disponíveis, juntamente com os padrões para cada função de usuário pré-configurada no Zabbix, são descritas abaixo.

Permissões padrão

Acesso aos elementos da interface

O acesso padrão às seções do menu depende do tipo de usuário. Veja a página de Permissões para detalhes.

Acesso a outras opções

Parâmetro	**Descrição**	**Funções de usuário padrão**			
		Função de super-administrador	*Função de administrador*	*Função de usuário*	*Função de convidado*
Acesso padrão a novos elementos da interface	Esta opção especifica como as novas seções do menu estarão acessíveis após uma atualização do Zabbix. As seções de menu existentes dos módulos permanecem inalteradas.	Sim	Sim	Sim	Sim
Acesso a serviços					
Acesso de leitura e gravação a serviços	Selecione o acesso de leitura e gravação a serviços: **Nenhum** - sem acesso **Todos** - acesso de leitura e gravação a todos os serviços **Lista de serviços** - selecione os serviços para acesso de leitura e gravação O acesso de leitura e gravação, se concedido, tem precedência sobre as configurações de acesso somente leitura e é herdado dinamicamente pelos serviços filhos.	Todos	Todos	Nenhum	Nenhum
Acesso de leitura e gravação a serviços com tag	Especifique o nome da tag e, opcionalmente, o valor para conceder acesso adicional de leitura e gravação a serviços que correspondam à tag. Esta opção está disponível se 'Lista de serviços' for selecionada no parâmetro *Acesso de leitura e gravação a serviços*. O acesso de leitura e gravação, se concedido, tem precedência sobre as configurações de acesso somente leitura e é herdado dinamicamente pelos serviços filhos.				

Acesso somente leitura a serviços	Selecione o acesso somente leitura a serviços: **Nenhum** - sem acesso **Todos** - acesso somente leitura a todos os serviços **Lista de serviços** - selecione os serviços para acesso somente leitura O acesso somente leitura não tem precedência sobre o acesso de leitura e gravação e é herdado dinamicamente pelos serviços filhos.			Todos	Todos
Acesso somente leitura a serviços com tag	Especifique o nome da tag e, opcionalmente, o valor para conceder acesso adicional somente leitura a serviços que correspondam à tag. Esta opção está disponível se 'Lista de serviços' for selecionada no parâmetro *Acesso somente leitura a serviços*. O acesso somente leitura não tem precedência sobre o acesso de leitura e gravação e é herdado dinamicamente pelos serviços filhos.				
Acesso a módulos					
<Nome do módulo>	Permitir/recusar acesso a um módulo específico. Apenas módulos habilitados são exibidos nesta seção. Não é possível conceder ou restringir acesso a um módulo que está atualmente desabilitado.	Sim	Sim	Sim	Sim
Acesso padrão a novos módulos	Esta opção especifica como novos módulos e widgets estarão acessíveis após uma atualização do Zabbix. Também se aplica a módulos e widgets adicionados em *Administração > Geral > Módulos*.				
Acesso à API					
Habilitado	Habilitar/desabilitar o acesso à API.	Sim	Sim	Sim	Não
Métodos da API	Selecione *Lista de permissão* para permitir, ou *Lista de negação* para negar os métodos da API especificados no campo de pesquisa. Observe que não é possível permitir alguns métodos da API e negar outros. No campo de pesquisa, comece a digitar o nome do método e selecione o método na lista de preenchimento automático. Você também pode pressionar o botão Selecionar e selecionar métodos da lista completa disponível para este tipo de usuário. Observe que, se determinada ação do bloco Acesso a ações estiver desmarcada, os usuários não poderão usar métodos da API relacionados a essa ação. Curingas são suportados. Exemplos: `dashboard.*` (todos os métodos do serviço de API 'dashboard.') `*` (qualquer método), `*.export` (métodos com o nome '.export' de todos os serviços de API). Se nenhum método tiver sido especificado, a regra *Lista de permissão/negação* será ignorada.				
Acesso a ações					

Criar e editar painéis Criar e editar mapas	Desmarcar esta caixa também revogará os direitos de usar os métodos da API `.create`, `.update` e `.delete` para os elementos correspondentes.	Sim	Sim	Sim	Não
Criar e editar manutenção				Não	
Adicionar comentários ao problema Alterar severidade Reconhecer problemas Suprimir problemas Fechar problemas	Desmarcar esta caixa também revogará os direitos de executar a ação correspondente via método da API `event.acknowledge`.			Sim	
Executar scripts	Desmarcar esta caixa também revogará os direitos de usar o método da API `script.execute`.				
Gerenciar tokens de API	Desmarcar esta caixa também revogará os direitos de usar todos os métodos da API `token.*`.				
Gerenciar relatórios agendados	Desmarcar esta caixa também revogará os direitos de usar todos os métodos da API `report.*`.			Não	
Gerenciar SLA	Habilitar/desabilitar os direitos de gerenciar SLA.				
Invocar "Executar agora" em hosts somente leitura	Permitir o uso da opção "Executar agora" nos últimos dados para items de hosts somente leitura.			Sim	
Alterar classificação do problema	Permitir alterar a classificação do problema de causa para sintoma e vice-versa.				
Acesso padrão a novas ações	Esta opção especifica como novas ações estarão acessíveis após uma atualização do Zabbix.				

Veja também:

- Configurando um usuário

9 Administração

Visão geral

O menu Administração é destinado às funções administrativas do Zabbix. Este menu está disponível apenas para usuários do tipo SuperAdmin.

1 Geral

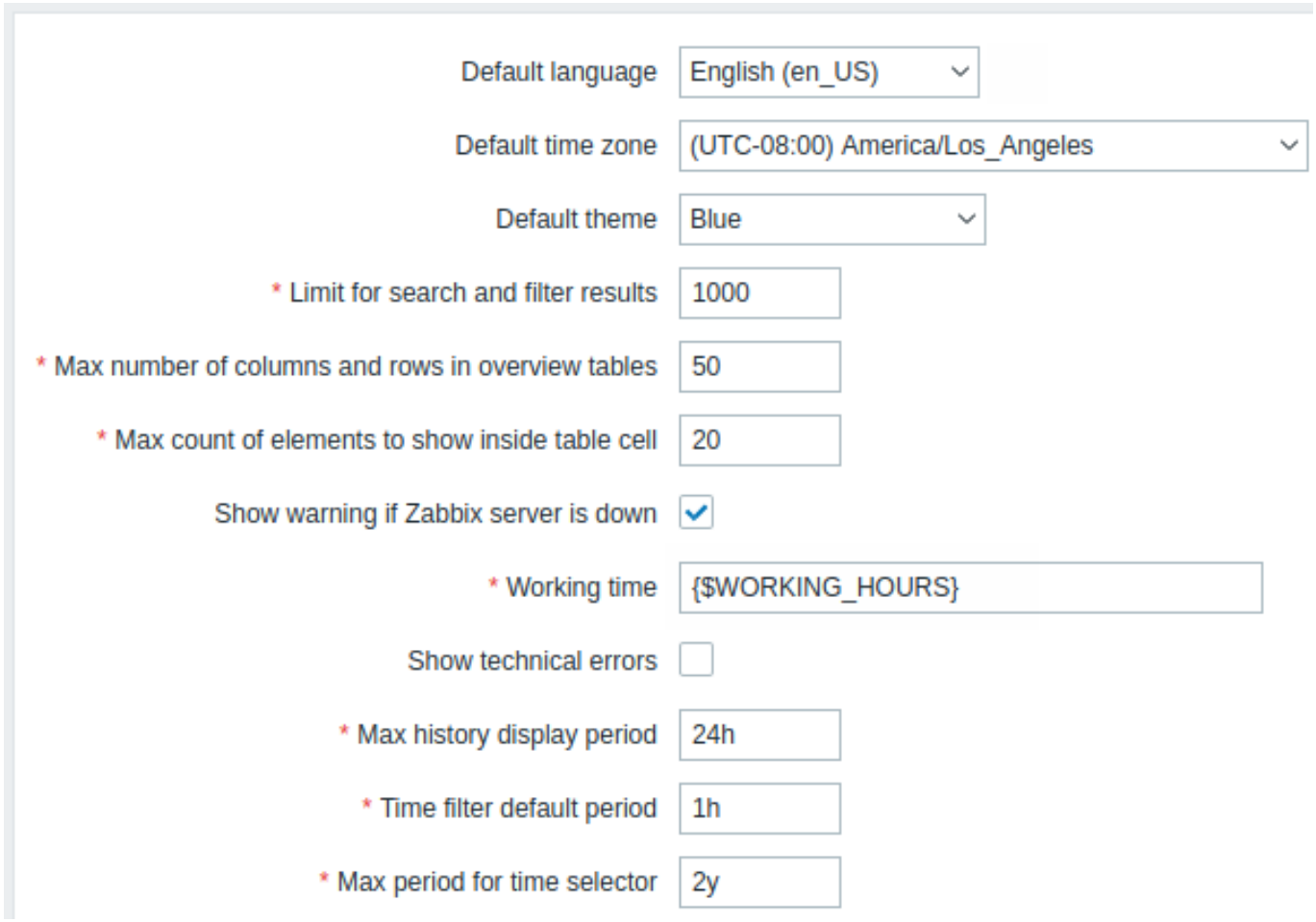
Visão geral

A seção *Administração* → *Geral* contém várias subseções para definir padrões relacionados ao frontend e personalizar o Zabbix.

A lista de subseções disponíveis aparece ao pressionar *Geral* na seção de menu *Administração*. Também é possível alternar entre as subseções usando o menu suspenso de título no canto superior esquerdo.

GUI

Esta seção fornece a personalização de vários padrões relacionados ao frontend.



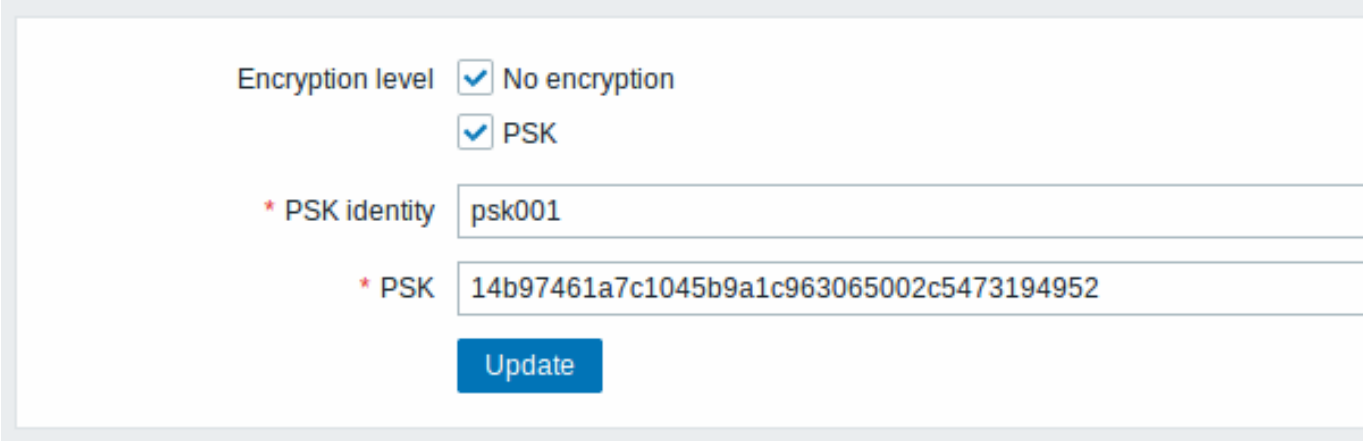
Parâmetros de configuração:

Parâmetro	Descrição
Idioma padrão	Idioma padrão para usuários que não especificaram um idioma em seus perfis e usuários convidados. Para mais informações, consulte Instalação de idiomas adicionais do frontend.
Fuso horário padrão	Fuso horário padrão para usuários que não especificaram um fuso horário em seus perfis e usuários convidados.
Tema padrão	Tema padrão para usuários que não especificaram um tema em seus perfis e usuários convidados.
Limite para resultados de pesquisa e filtro	Quantidade máxima de elementos (linhas) que serão exibidos em uma lista da interface web, por exemplo, em *Coleta de dados* > *Hosts*. *Nota*: Se definido, por exemplo, como '50', apenas os primeiros 50 elementos serão exibidos em todas as listas afetadas do frontend. Se alguma lista contiver mais de cinquenta elementos, a indicação disso será o sinal '+' em *"Exibindo de 1 a 50 de **50+** encontrados"*. Além disso, se o filtro for usado e ainda houver mais de 50 correspondências, apenas as primeiras 50 serão exibidas. Observe que aumentar o valor deste parâmetro pode levar à diminuição do desempenho e ao aumento do consumo de memória no lado do servidor web.
Número máximo de colunas e linhas nas tabelas de visão geral	Número máximo de colunas e linhas a serem exibidas nos widgets de painel Visão geral de dados e Visão geral de triggers. O mesmo limite se aplica tanto às colunas quanto às linhas. Se existirem mais linhas e/ou colunas do que as exibidas, o sistema exibirá um aviso na parte inferior da tabela: "Nem todos os resultados são exibidos. Por favor, forneça critérios de pesquisa mais específicos."
Quantidade máxima de elementos para mostrar dentro da célula da tabela	Para entradas que são exibidas em uma única célula da tabela, não serão mostradas mais do que o configurado aqui.
Exibir aviso se o Zabbix server estiver inativo	Este parâmetro permite que uma mensagem de aviso seja exibida em uma janela do navegador se o Zabbix server não puder ser alcançado (possivelmente inativo). A mensagem permanece visível mesmo se o usuário rolar a página para baixo. Ao passar o mouse sobre ela, a mensagem é temporariamente ocultada para revelar o conteúdo abaixo dela.

Parâmetro	Descrição
Horário de trabalho	Este parâmetro do sistema define o horário de trabalho. Nos gráficos, o horário de trabalho é exibido como fundo branco e o horário fora do expediente é exibido como cinza. Consulte a página Especificação do período de tempo para descrição do formato de tempo. Macros de usuário são suportadas.
Exibir erros técnicos	Se marcado, todos os usuários registrados poderão ver erros técnicos (PHP/SQL). Se desmarcado, as informações estarão disponíveis apenas para Zabbix Super Admins e usuários pertencentes a grupos de usuários com modo de depuração ativado.
Período máximo de exibição do histórico	Período máximo de tempo para exibir dados históricos em *Monitoramento* > *Últimos dados*, informações do cenário web do host em *Monitoramento* > *Hosts* e no widget de painel *Visão geral de dados*. Intervalo permitido: 24 horas (padrão) - 1 semana. Sufixos de tempo, por exemplo, 1w (uma semana), 36h (36 horas), são suportados.
Período padrão do filtro de tempo	Período de tempo a ser usado em gráficos e painéis por padrão. Intervalo permitido: 1 minuto - 10 anos (padrão: 1 hora). Sufixos de tempo, por exemplo, 10m (dez minutos), 5w (cinco semanas), são suportados. Nota: quando um usuário altera o período de tempo ao visualizar um gráfico, esse período de tempo é armazenado como preferência do usuário, substituindo o padrão global ou uma seleção anterior do usuário.
Período máximo para o seletor de tempo	Período máximo disponível para gráficos e painéis. Os usuários não poderão visualizar dados mais antigos. Intervalo permitido: 1 ano - 10 anos (padrão: 2 anos). Sufixos de tempo, por exemplo, 1y (um ano), 365w (365 semanas), são suportados.

Autoregistro

Nesta seção, você pode configurar o nível de criptografia para o autoregistro de agent ativo.



Os parâmetros marcados com um asterisco são obrigatórios.

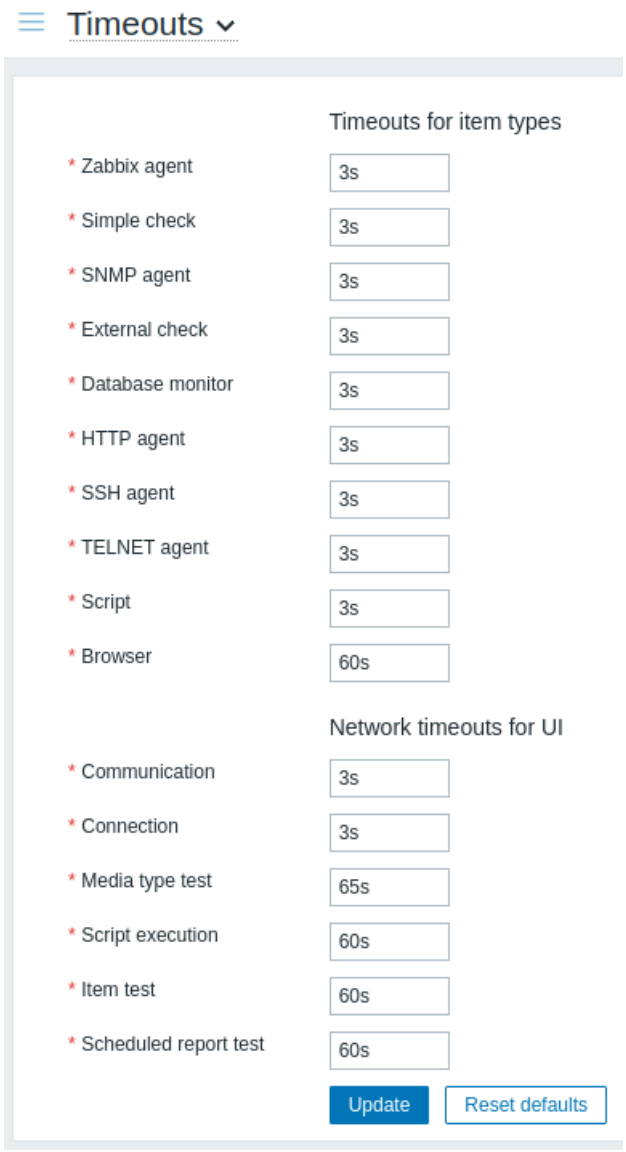
Parâmetros de configuração:

Parâmetro	Descrição
Nível de criptografia	Selecione uma ou ambas as opções para o nível de criptografia: **Sem criptografia** - conexões não criptografadas são permitidas **PSK** - conexões criptografadas TLS com uma chave pré-compartilhada são permitidas
Identidade PSK	Insira a string de identidade da chave pré-compartilhada. Este campo só está disponível se 'PSK' for selecionado como *Nível de criptografia*. Não coloque informações confidenciais na identidade PSK, pois ela é transmitida sem criptografia pela rede para informar ao receptor qual PSK usar.
PSK	Insira a chave pré-compartilhada (um número par de caracteres hexadecimais). Comprimento máximo: 512 dígitos hexadecimais (PSK de 256 bytes) se o Zabbix usar a biblioteca GnuTLS ou OpenSSL, 64 dígitos hexadecimais (PSK de 32 bytes) se o Zabbix usar a biblioteca mbed TLS (PolarSSL). Exemplo: 1f87b595725ac58dd977beef14b97461a7c1045b9a1c963065002c5473194952 Este campo só está disponível se 'PSK' for selecionado como *Nível de criptografia*.

Veja também: Autoregistro seguro

Timeouts

Nesta seção, é possível definir timeouts globais por tipo de item e timeouts de rede. Todos os campos deste formulário são obrigatórios.

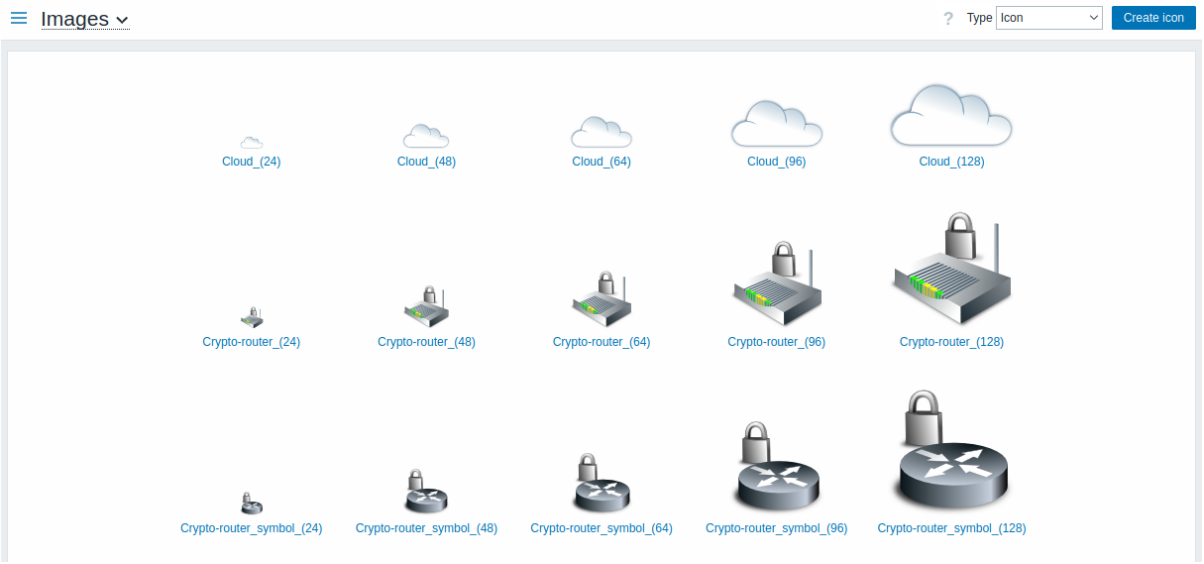


Parâmetro	Descrição
Timeouts para tipos de item	Quantos segundos esperar por uma resposta de um item monitorado (com base em seu tipo). Intervalo permitido: 1 - 600s (padrão: 3s; padrão para o tipo de item Browser: 60s). Sufixos de tempo, por exemplo, 30s, 1m, e macros de usuário são suportados. Tipos de item suportados: - Zabbix agent (checagens passivas e ativas) - Simple check (exceto itens `icmpping*`, `vmware.*`) - SNMP agent (apenas para itens SNMP `walk[OID]` e `get[OID]`) - External check - Database monitor - HTTP agent - SSH agent - TELNET agent - Script - Browser Observe que, se um proxy for usado e tiver timeouts configurados, as configurações de timeout do proxy substituirão as globais. Se houver timeouts definidos para itens específicos, eles substituirão as configurações do proxy e globais.

Parâmetro		Descrição
Timeouts de rede para a UI		
	Comunicação	Quantos segundos esperar antes de fechar um socket ocioso (se uma conexão com o Zabbix server foi estabelecida anteriormente, mas o frontend não consegue concluir a operação de leitura/envio de dados durante esse tempo, a conexão será encerrada). Intervalo permitido: 1 - 300s (padrão: 3s).
	Conexão	Quantos segundos esperar antes de interromper uma tentativa de conexão com o Zabbix server. Intervalo permitido: 1 - 30s (padrão: 3s).
	Teste de tipo de mídia	Quantos segundos esperar por uma resposta ao testar um tipo de mídia. Intervalo permitido: 1 - 300s (padrão: 65s).
	Execução de script	Quantos segundos esperar por uma resposta ao executar um script. Intervalo permitido: 1 - 300s (padrão: 60s). Esse timeout é para toda a cadeia de execução do script, que pode ter vários comprimentos. Por exemplo, se o script for executado no agent, é uma viagem de ida e volta pelo server (possivelmente também proxy) até o agent e de volta. Para scripts executados no agent, o timeout do server ou do proxy se aplica. Para scripts executados apenas em agent ativo, é provável que o timeout padrão do server/proxy precise ser aumentado. O timeout do server/proxy deve ser maior que a frequência de atualização da checagem ativa, caso contrário, o timeout do server/proxy pode ser excedido antes mesmo que o agent ativo receba o script. Veja também: Timeout de script
	Teste de item	Quantos segundos esperar pelos dados retornados ao testar um item. Intervalo permitido: 1 - 600s (padrão: 60s).
	Teste de relatório agendado	Quantos segundos esperar pelos dados retornados ao testar um relatório agendado. Intervalo permitido: 1 - 300s (padrão: 60s).

Imagens

A seção Imagens exibe todas as imagens disponíveis no Zabbix. As imagens são armazenadas no banco de dados.

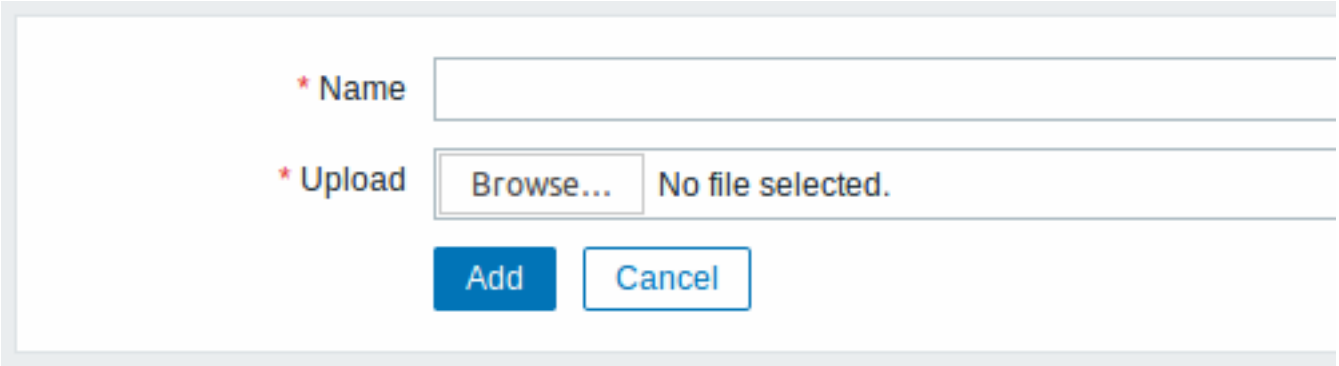


O menu suspenso *Tipo* permite alternar entre ícones e imagens de fundo:

- Ícones são usados para exibir elementos de mapas de rede
- Fundos são usados como imagens de fundo dos mapas de rede

Adicionando imagem

Você pode adicionar sua própria imagem clicando no botão *Criar ícone* ou *Criar fundo* no canto superior direito.



Atributos da imagem:

Parâmetro	Descrição
Nome	Nome exclusivo da imagem.
Upload	Selecione o arquivo (PNG, JPEG, GIF ou WebP) do sistema local para ser enviado ao Zabbix. *Observação*: pode ser possível enviar outros formatos que serão convertidos para PNG durante o upload. A biblioteca GD é usada para o processamento de imagens, portanto, os formatos suportados dependem da versão da biblioteca utilizada (o Zabbix requer a versão 2.0.28 ou superior).

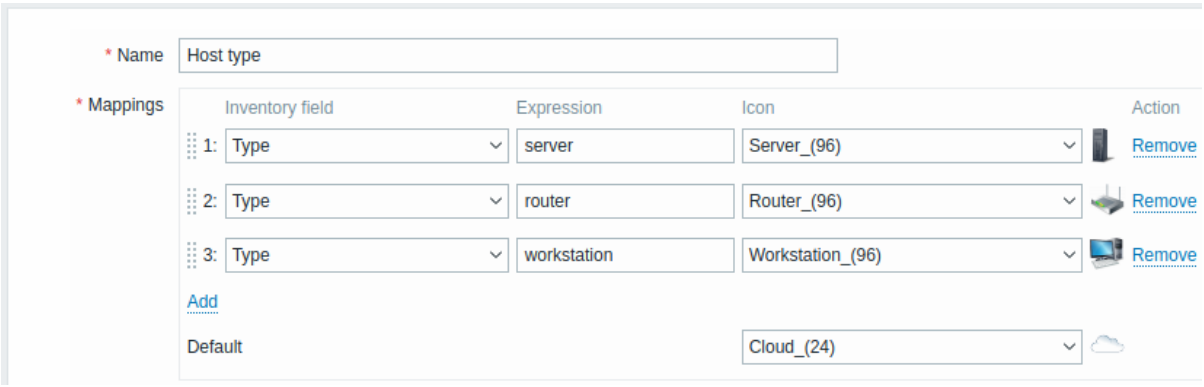
Note:
O tamanho máximo do arquivo de upload é limitado pelo valor de ZBX_MAX_IMAGE_SIZE, que é de 1024x1024 bytes ou 1 MB.
O upload de uma imagem pode falhar se o tamanho da imagem estiver próximo de 1 MB e o parâmetro de configuração do MySQL `max_allowed_packet` estiver no padrão de 1 MB. Nesse caso, aumente o parâmetro max_allowed_packet.

Mapeamento de ícones

Esta seção permite criar o mapeamento de determinados hosts com determinados ícones. As informações do campo de inventário do host são usadas para criar o mapeamento.

Os mapeamentos podem então ser usados na configuração do mapa de rede para atribuir automaticamente ícones apropriados aos hosts correspondentes.

Para criar um novo mapa de ícones, clique em *Criar mapa de ícones* no canto superior direito.



Parâmetros de configuração:

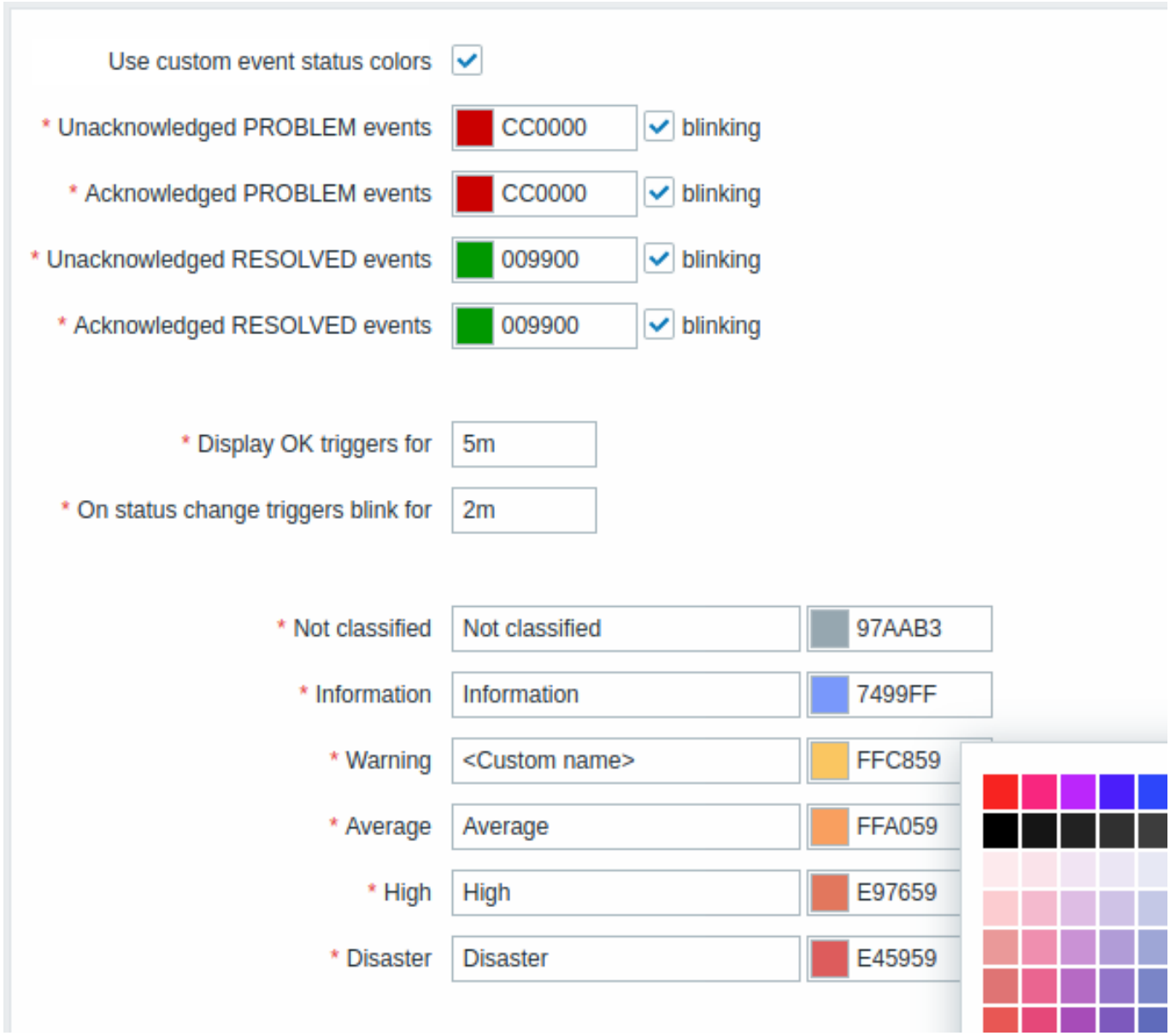
Parâmetro	Descrição
Nome	Nome exclusivo do mapa de ícones.
Mapeamentos	Uma lista de mapeamentos. A ordem dos mapeamentos determina qual terá prioridade. Você pode mover os mapeamentos para cima e para baixo na lista com arrastar e soltar.
Campo de inventário	Campo de inventário do host que será analisado para buscar uma correspondência.
Expressão	Expressão regular que descreve a correspondência.
Ícone	Ícone a ser usado se uma correspondência para a expressão for encontrada.
Padrão	Ícone padrão a ser usado.

Expressões regulares

Esta seção permite criar expressões regulares personalizadas que podem ser usadas em vários lugares no frontend. Veja a seção Expressões regulares para mais detalhes.

Opções de exibição de trigger

Esta seção permite personalizar como o status do trigger é exibido no frontend e os nomes e cores da severidade do trigger.

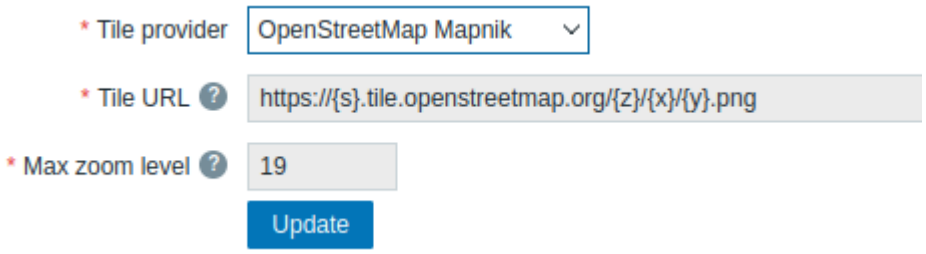


Parâmetro	Descrição
Usar cores personalizadas para status de evento	Ao marcar este parâmetro, ativa-se a personalização das cores para problemas reconhecidos/não reconhecidos.
Eventos de PROBLEMA não reconhecidos, *Eventos de PROBLEMA reconhecidos*, *Eventos de RESOLVIDO não reconhecidos*, *Eventos de RESOLVIDO reconhecidos*	Digite um novo código de cor ou clique na cor para selecionar uma nova na paleta fornecida. Se a caixa de seleção *piscando* estiver marcada, os triggers piscarão por algum tempo após a mudança de status para se tornarem mais visíveis.
Exibir triggers OK por	Período de tempo para exibir triggers OK. Faixa permitida: 0 - 24 horas. Sufixos de tempo, por exemplo, 5m, 2h, 1d, são suportados.
Ao mudar o status, triggers piscam por	Duração do piscar do trigger. Faixa permitida: 0 - 24 horas. Sufixos de tempo, por exemplo, 5m, 2h, 1d, são suportados.

Parâmetro	Descrição
Não classificado, *Informação*, *Aviso*, *Média*, *Alta*, *Desastre*	Nomes e/ou cores de severidade personalizados para exibir em vez do padrão do sistema. Digite um novo código de cor ou clique na cor para selecionar uma nova na paleta fornecida. Observe que os nomes de severidade personalizados inseridos aqui serão usados em todos os idiomas. Se você precisar traduzi-los para outros idiomas para determinados usuários, consulte a página Personalizando severidades de trigger.

Mapas geográficos

Esta seção permite selecionar o provedor de serviço de mapas geográficos e configurar as configurações do provedor de serviço para o widget de mapa geográfico do dashboard. Para fornecer visualização usando mapas geográficos, o Zabbix utiliza a biblioteca de mapas interativos JavaScript de código aberto Leaflet. Observe que o Zabbix não tem controle sobre a qualidade das imagens fornecidas por provedores de tiles de terceiros, incluindo os provedores de tiles predefinidos.



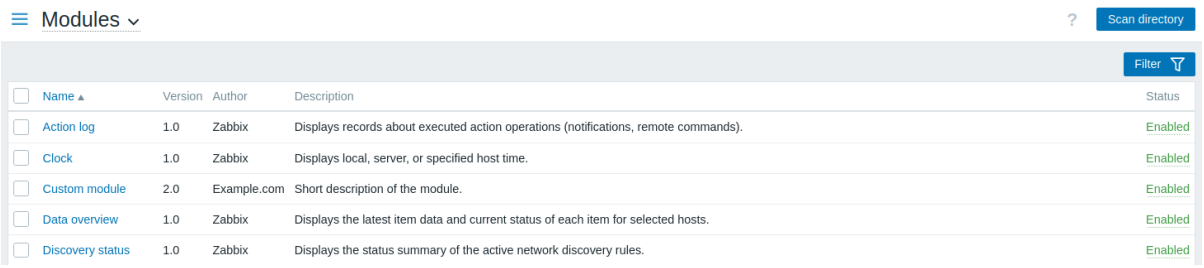
Parâmetro	Descrição
Tile provider	Selecione um dos provedores de serviço de tiles disponíveis ou selecione *Outro* para adicionar outro provedor de tiles ou tiles hospedados localmente (veja Usando um provedor de serviço de tiles personalizado).
Tile URL	O template de URL (até 2048 caracteres) para carregar e exibir a camada de tiles nos mapas geográficos. Este campo é editável apenas se o *Tile provider* estiver definido como *Outro*. Os seguintes placeholders são suportados: *{s}* representa um dos subdomínios disponíveis; *{z}* representa o parâmetro de nível de zoom na URL; *{x}* e *{y}* representam as coordenadas do tile; *{r}* pode ser usado para adicionar "@2x" à URL para carregar tiles retina. Exemplo: `https://{s}.example.com/{z}/{x}/{y}{r}.png`
Attribution text	Texto de atribuição do provedor de tiles a ser exibido em uma pequena caixa de texto no mapa. Este campo é visível apenas se o *Tile provider* estiver definido como *Outro*.
Max zoom level	Nível máximo de zoom do mapa. Este campo é editável apenas se o *Tile provider* estiver definido como *Outro*.

Usando um provedor de serviço de tile personalizado

O widget Geomap é capaz de carregar imagens de tile raster de um serviço de provedor de tile personalizado, auto-hospedado ou de terceiros. Para usar um serviço de provedor de tile de terceiros personalizado ou uma pasta ou servidor de tile auto-hospedado, selecione *Outro* no campo *Provedor de tile* e especifique a URL personalizada no campo *URL do tile* usando os placeholders apropriados.

Módulos

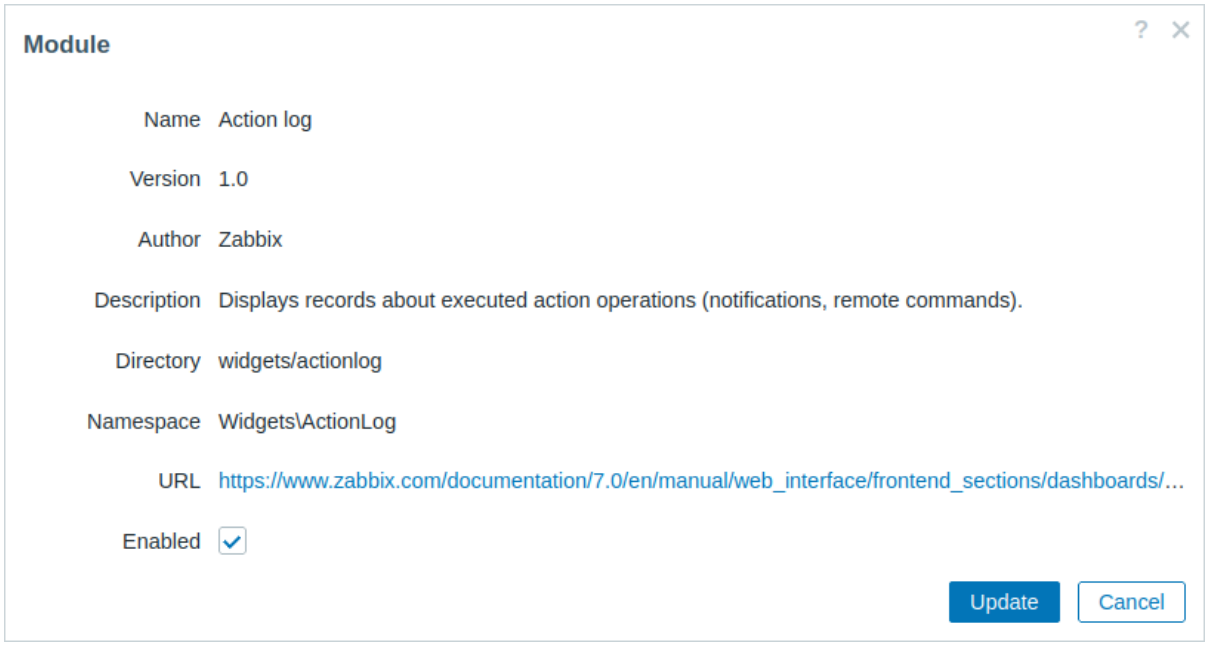
Esta seção permite administrar módulos frontend personalizados, bem como os integrados.



Clique em *Procurar diretório* para registrar/remover o registro de quaisquer módulos personalizados. Os módulos registrados aparecerão na lista; os módulos não registrados serão removidos da lista.

Clique no status do módulo na lista para ativar/desativar um módulo. Você também pode ativar/desativar vários módulos de uma vez, selecionando-os na lista e, em seguida, clicando nos botões *Ativar/Desativar* abaixo da lista.

Clique no nome do módulo na lista para visualizar seus detalhes em uma janela pop-up.

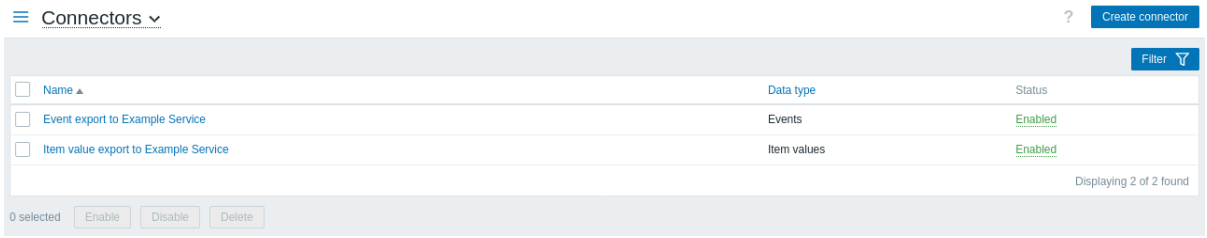


O status do módulo também pode ser atualizado na janela pop-up de detalhes do módulo; para isso, marque/desmarque a caixa de seleção *Ativado* e clique em *Atualizar*.

Você pode filtrar os módulos por nome ou status (ativado/desativado).

Conectores

Esta seção permite configurar conectores para transmissão de dados do Zabbix para sistemas externos via HTTP.

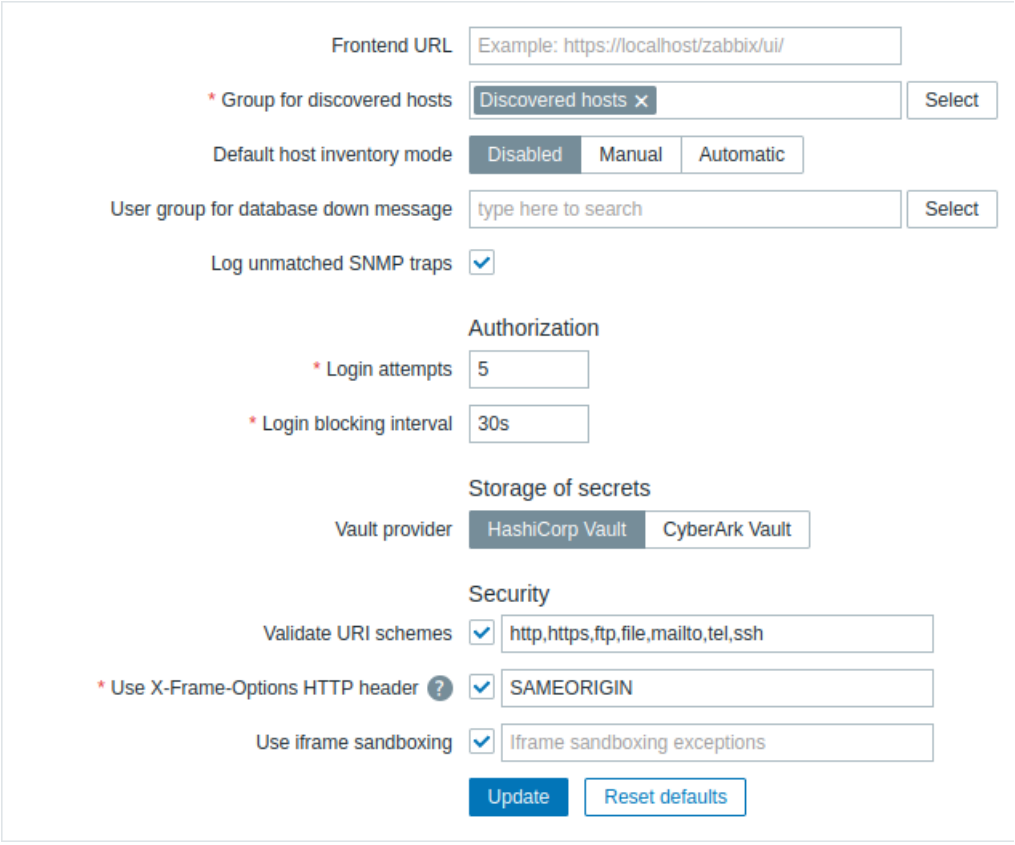


Clique em *Criar conector* para configurar um novo conector.

Você pode filtrar conectores por nome ou status (habilitado/desabilitado). Clique no status do conector na lista para habilitar/desabilitar um conector. Você também pode habilitar/desabilitar conectores em massa selecionando-os na lista e, em seguida, clicando nos botões *Habilitar/Desabilitar* abaixo da lista.

Outros

Esta seção permite configurar outros parâmetros diversos do frontend.



Parâmetro	Descrição
URL do frontend	URL (até 2048 caracteres) para a interface web do Zabbix. Este parâmetro é usado pelo serviço web do Zabbix para comunicação com o frontend e deve ser especificado para habilitar relatórios agendados.
Grupo para hosts descobertos	Hosts descobertos por descoberta de rede e autoregistro de agent serão automaticamente colocados no grupo de hosts selecionado aqui.
Modo padrão de inventário de host	Modo padrão para inventário de host. Ele será seguido sempre que um novo host ou protótipo de host for criado pelo server ou frontend, a menos que seja substituído durante a descoberta/autoregistro do host pela operação *Definir modo de inventário de host*.
Grupo de usuários para mensagem de banco de dados fora do ar	Grupo de usuários para envio de mensagem de alarme ou 'Nenhum'. O server Zabbix depende da disponibilidade do banco de dados backend. Ele não pode funcionar sem um banco de dados. Se o banco de dados estiver fora do ar, usuários selecionados podem ser notificados pelo Zabbix. As notificações serão enviadas para o grupo de usuários definido aqui usando as entradas de mídia de usuário habilitadas. As notificações são transmitidas usando os seguintes tipos de mídia (quando habilitados): e-mail; SMS; scripts de alerta personalizados. Mesmo que uma entrada de mídia webhook seja configurada e habilitada, ela não acionará notificações. O server Zabbix não irá parar; ele aguardará até que o banco de dados volte para continuar o processamento. A notificação consiste no seguinte conteúdo: `[MySQL|PostgreSQL|Oracle] database <DB Name> [on <DB Host>:<DB Port>] is not available: <mensagem de erro dependendo do tipo de SGBD (banco de dados)>` <DB Host> não é adicionado à mensagem se for definido como um valor vazio e <DB Port> não é adicionado se for o valor padrão (”0”). O gerenciador de alertas (um processo especial do server Zabbix) tenta estabelecer uma nova conexão com o banco de dados a cada 10 segundos. Se o banco de dados ainda estiver fora do ar, o gerenciador de alertas repete o envio de alertas, mas não mais frequentemente do que a cada 15 minutos.
Registrar traps SNMP não correspondentes	Registra trap SNMP se nenhuma interface SNMP correspondente for encontrada.

Autorização

Parâmetro	Descrição
Tentativas de login	Número de tentativas de login sem sucesso antes que a possibilidade de login seja bloqueada.
Intervalo de bloqueio de login	Período de tempo durante o qual o login será proibido quando o limite de *Tentativas de login* for excedido. Faixa permitida: 30 - 3600 segundos. Sufixos de tempo, por exemplo, 90s, 5m, 1h, são suportados.

Armazenamento de segredos

O parâmetro *Vault provider* permite selecionar o software de gerenciamento de segredos para armazenar os valores das macros de usuário. Opções suportadas:

- *HashiCorp Vault* (padrão)
- *CyberArk Vault*

Veja também: Armazenamento de segredos.

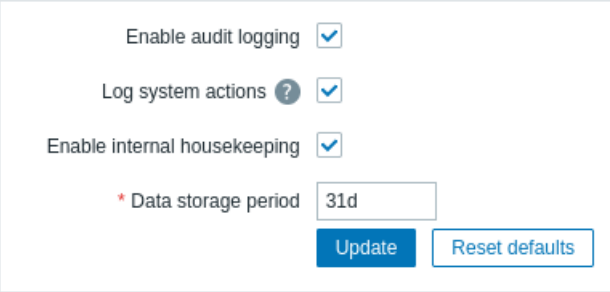
Segurança

Parâmetro	Descrição
Validar esquemas de URI	Desmarque esta caixa para desabilitar a validação de esquema de URI (habilitado por padrão). Se marcada, você pode especificar uma lista separada por vírgulas de esquemas de URI permitidos (padrão: http,https,ftp,file,mailto,tel,ssh). Aplica-se a todos os campos no frontend onde URIs são usados (por exemplo, URLs de elementos de mapas).
Usar cabeçalho HTTP X-Frame-Options	Desmarque esta caixa para desabilitar o cabeçalho HTTP X-Frame-options (não recomendado). Se marcada, você pode especificar o valor do cabeçalho HTTP X-Frame-options. Valores suportados: **SAMEORIGIN** (padrão) ou **'self'** (deve estar entre aspas simples) - a página só pode ser exibida em um frame na mesma origem que a própria página; **DENY** ou **'none'** (deve estar entre aspas simples) - a página não pode ser exibida em um frame, independentemente do site que tente fazê-lo; **uma string de nomes de host separados por espaço**; adicionar **'self'** (deve estar entre aspas simples) à lista permite que a página seja exibida em um frame na mesma origem que a própria página. Observe que usar **'self'** ou **'none'** sem aspas simples fará com que sejam considerados como nomes de host.
Usar sandboxing em iframe	Desmarque esta caixa para desabilitar a colocação do conteúdo da URL recuperada em sandbox (não recomendado). Se marcada, você pode especificar as exceções de sandboxing do iframe; restrições não especificadas ainda serão aplicadas. Se este campo estiver vazio, todas as restrições do atributo sandbox serão aplicadas. Para mais informações, veja a descrição do atributo `sandbox`.

2 Log de auditoria

Visão geral

Esta seção permite configurar as configurações do log de auditoria.



Os seguintes parâmetros estão disponíveis:

Parâmetro	Descrição
Habilitar log de auditoria	Habilita (padrão) ou desabilita o log de auditoria.

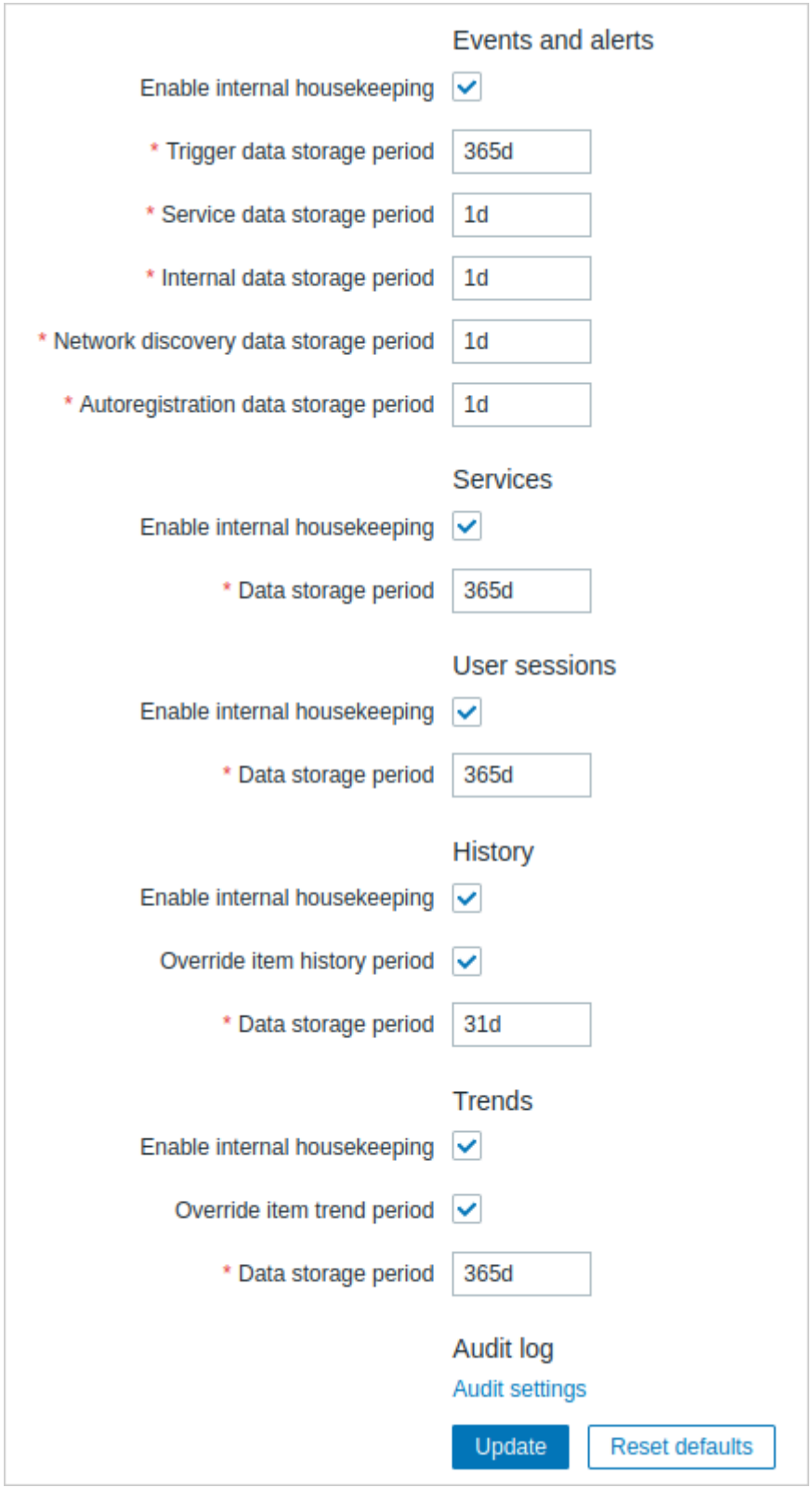
Parâmetro	Descrição
Registrar ações do sistema	Habilita (padrão) ou desabilita o log de auditoria de atividades de descoberta de baixo nível, descoberta de rede e autorregistro realizadas pelo server (usuário do sistema).
Habilitar limpeza interna	Habilita (padrão) ou desabilita a limpeza interna para registros de log de auditoria.
Período de armazenamento de dados	Quantidade de dias que os registros de log de auditoria devem ser mantidos antes de serem removidos pelo housekeeper. Obrigatório se a limpeza estiver habilitada. Padrão: 31 dias.

Os parâmetros de compactação do log de auditoria estão disponíveis na seção *Administração > Limpeza*, dentro do bloco *Compactação de histórico, tendências e log de auditoria*, que se torna visível quando o TimescaleDB é usado.

3 Housekeeping

Visão geral

O housekeeper é um processo periódico, executado pelo Zabbix server. O processo remove informações desatualizadas e informações excluídas pelo usuário.



Nesta seção, as tarefas de housekeeping podem ser ativadas ou desativadas individualmente para: eventos e alertas/serviços de TI/sessões de usuário/histórico/tendências. As configurações de housekeeping de auditoria estão disponíveis em uma seção de menu separada.

Se o housekeeping estiver ativado, é possível definir por quantos dias os registros de dados serão mantidos antes de serem removidos pelo housekeeper.

Excluir um item/trigger também excluirá os problemas gerados por esse item/trigger.

Além disso, um evento só será excluído pelo housekeeper se não estiver associado a um problema de nenhuma forma. Isso significa que, se um evento for um problema ou um evento de recuperação, ele não será excluído até que o registro do problema relacionado seja removido. O housekeeper excluirá primeiro os problemas e depois os eventos, para evitar possíveis problemas com eventos ou registros de problemas obsoletos.

Para histórico e tendências, uma opção adicional está disponível: *Substituir período de histórico do item* e *Substituir período de tendência do item*. Esta opção permite definir globalmente por quantos dias o histórico/tendências dos itens serão armazenados (1 hora a 25 anos; ou ”0”), substituindo os valores respectivos de *Armazenar até* definidos para itens individuais no formulário de configuração do item. Observe que o período de armazenamento não será substituído para itens que tenham a opção de configuração *Não armazenar* ativada.

É possível substituir o período de armazenamento de histórico/tendência mesmo que o housekeeping interno esteja desativado. Assim, ao usar um housekeeper externo, o período de armazenamento do histórico pode ser definido usando o campo *Período de armazenamento de dados* do histórico.

O bloco *Compressão de histórico, tendências e log de auditoria* aparece se estiver usando o TimescaleDB.

Attention:
Se estiver usando o TimescaleDB, para aproveitar ao máximo o particionamento automático das tabelas de histórico e tendências do TimescaleDB, as opções *Substituir período de histórico do item* e *Substituir período de tendência do item* devem ser ativadas, bem como a opção *Ativar housekeeping interno* para histórico e tendências. Caso contrário, os dados mantidos nessas tabelas ainda serão armazenados em partições, no entanto, o housekeeper não irá descartar partições desatualizadas, e avisos sobre configuração incorreta serão exibidos. Quando a exclusão de partições desatualizadas estiver ativada, o Zabbix server e o frontend não rastrearão mais os itens excluídos, e o histórico dos itens excluídos será limpo quando uma partição desatualizada for excluída.

Sufixos de tempo são suportados nos campos de período, por exemplo, 1d (um dia), 1w (uma semana). O mínimo é 1 dia (1 hora para histórico), o máximo - 25 anos.

O botão *Restaurar padrões* permite reverter quaisquer alterações feitas.

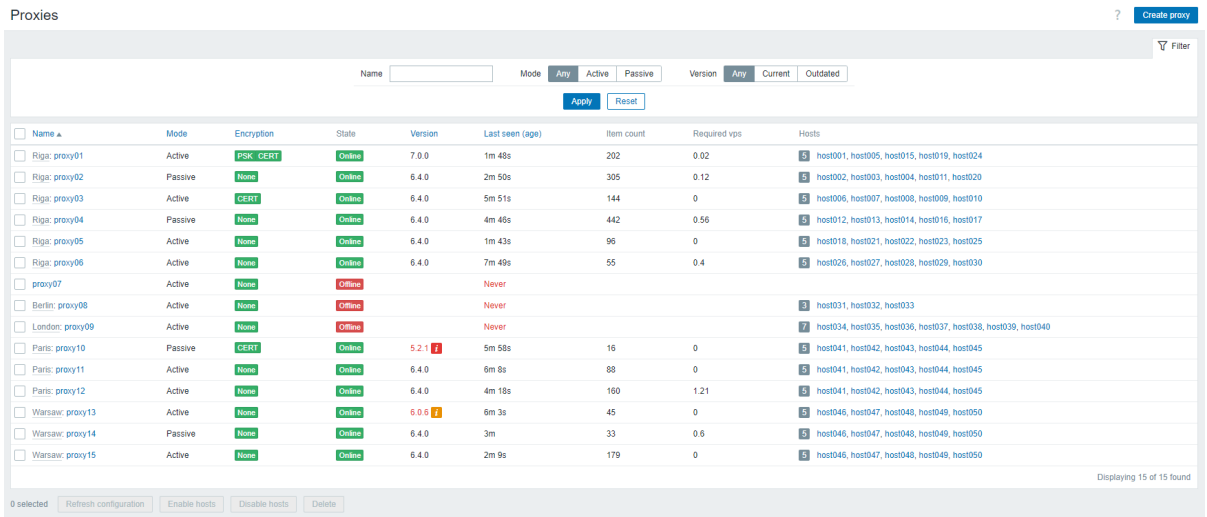
4 Proxies

Visão geral

Na seção *Administração* → *Proxies*, proxies para monitoramento distribuído podem ser configurados no frontend do Zabbix.

Proxies

Uma lista dos proxies existentes com seus detalhes é exibida.



Dados exibidos:

Coluna	Descrição
Nome	Nome do proxy. Clicar no nome do proxy abre o formulário de configuração do proxy. Se o proxy pertence a um grupo de proxies, o nome do grupo é exibido antes do nome do proxy como um link cinza. Clicar no nome do grupo abre o formulário de configuração do grupo de proxies.
Modo	Modo do proxy - *Ativo* ou *Passivo*.

Coluna	Descrição
Criptografia	Status da criptografia para conexões do proxy: **Nenhuma** - sem criptografia; **PSK** - usando chave pré-compartilhada; **Cert** - usando certificado.
Estado	Estado do proxy: **Desconhecido** - o proxy foi criado enquanto o Zabbix server estava desligado, ou o server ainda não atualizou o estado; **Online** - o proxy se comunicou com o Zabbix server (proxy passivo respondeu à solicitação do server; proxy ativo enviou solicitação) dentro do período de failover; **Offline** - o proxy não se comunicou com o Zabbix server dentro do período de failover.
Versão	Versão do proxy (número de versão com três dígitos). Se o proxy estiver desatualizado ou não for suportado, o número da versão será destacado (vermelho) e um ícone de status de informação (amarelo ou vermelho) será exibido. Passe o mouse sobre o ícone para detalhes.
Última vez visto (idade)	O horário em que o proxy foi visto pela última vez pelo server.
Contagem de items	O número de items habilitados em hosts habilitados atribuídos ao proxy.
vps necessário	Desempenho necessário do proxy (o número de valores que precisam ser coletados por segundo).
Hosts	Contagem de hosts habilitados atribuídos ao proxy e uma lista de hosts monitorados pelo proxy. Clicar no nome do host abre o formulário de configuração do host.

Para configurar um novo proxy, clique no botão *Criar proxy* no canto superior direito.

Opções de edição em massa

Os botões abaixo da lista oferecem algumas opções de edição em massa:

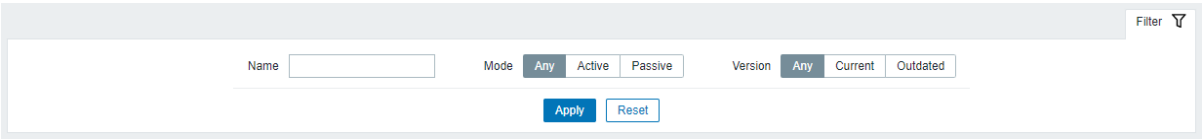
- *Atualizar configuração* - atualiza a configuração dos proxies;
- *Habilitar hosts* - altera o status dos hosts monitorados pelo proxy para *Monitorado*;
- *Desabilitar hosts* - altera o status dos hosts monitorados pelo proxy para *Não monitorado*;
- *Excluir* - exclui os proxies.

Para usar essas opções, marque as caixas de seleção antes dos respectivos proxies e clique no botão desejado.

Usando o filtro

Você pode usar o filtro para exibir apenas os proxies nos quais está interessado. Para melhor desempenho de pesquisa, os dados são pesquisados com macros não resolvidas.

O link *Filtro* está disponível acima da lista de proxies. Se você clicar nele, um filtro ficará disponível onde você pode filtrar proxies por nome, modo e versão. Observe que a opção de filtro *Desatualizado* exibe proxies desatualizados (parcialmente suportados) e não suportados.



5 Grupos de proxy

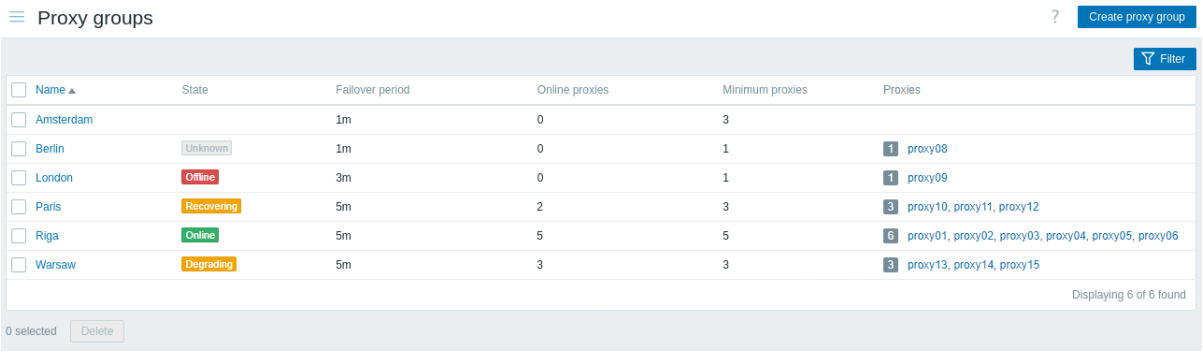
Visão geral

Em *Administração* → *Grupos de proxy* é possível configurar grupos de proxy.

Os grupos de proxy são usados no balanceamento de carga de proxy com distribuição automatizada de hosts entre proxies e alta disponibilidade entre proxies.

Grupos de proxy

Uma lista dos grupos de proxy existentes com seus detalhes é exibida.



Dados exibidos:

Coluna	Descrição
Nome	Nome do grupo de proxy. Clicar no nome do grupo de proxy abre o formulário de configuração do grupo de proxy.
Estado	Estado do grupo de proxy (exibido se o grupo contiver pelo menos um proxy): **Desconhecido** - o grupo foi criado enquanto o Zabbix server estava desligado, ou o server ainda não atualizou o estado; **Online** - o número mínimo de proxies comunicou-se com o Zabbix server dentro do período de failover; **Degradando** - o grupo ficará offline em breve devido ao número de proxies online estar abaixo do limite *Mínimo de proxies*; **Offline** - menos do que o número mínimo de proxies comunicou-se com o Zabbix server dentro do período de failover; **Recuperando** - o grupo ficará online em breve devido ao número de proxies online atingir o limite *Mínimo de proxies*. O estado do grupo de proxy pode ser monitorado com os itens zabbix[proxy group,<name>,state] e zabbix[proxy group,discovery].
Período de failover	Período durante o qual um proxy no grupo de proxy deve se comunicar com o Zabbix server para ser considerado online.
Proxies online	Número de proxies online (exibido em vermelho se estiver abaixo do mínimo do grupo).
Mínimo de proxies	Número mínimo de proxies online necessários para manter o grupo de proxy online.
Proxies	Contagem de proxies no grupo e uma lista de proxies no grupo, com links para o formulário de configuração do proxy. O número máximo de proxies listados é limitado pelo valor *Contagem máxima de elementos a serem exibidos dentro da célula da tabela*.

Para configurar um novo grupo de proxy, clique no botão *Criar grupos de proxy* no canto superior direito.

Opções de edição em massa

Os botões abaixo da lista oferecem algumas opções de edição em massa:

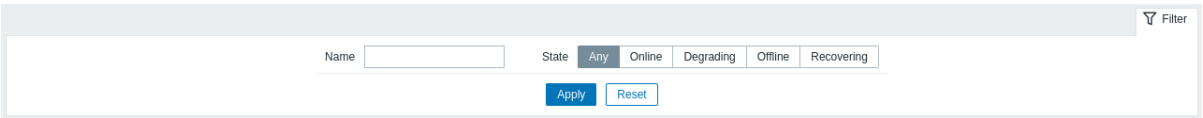
- *Excluir* - exclui os grupos de proxy.

Para usar essas opções, marque as caixas de seleção antes dos respectivos grupos de proxy e clique no botão necessário.

Usando o filtro

Você pode usar o filtro para exibir apenas os grupos de proxy que lhe interessam. Para melhor desempenho na pesquisa, os dados são pesquisados com macros não resolvidas.

O link *Filtro* está disponível acima da lista de grupos de proxy. Se você clicar nele, um filtro ficará disponível onde você pode filtrar grupos de proxy por nome e status.



6 Macros

Visão geral

Esta seção permite definir macros de usuário em todo o sistema como pares nome-valor. Observe que os valores das macros podem ser mantidos como texto simples, texto secreto ou segredo do Vault. Também é possível adicionar uma descrição.

7 Fila

Visão geral

Na seção *Administração* → *Fila*, são exibidos os items que estão aguardando para serem atualizados.

Idealmente, ao abrir esta seção, tudo deve estar ”verde”, o que significa que não há items na fila. Se todos os items forem atualizados sem atraso, não haverá nenhum aguardando. No entanto, devido à falta de desempenho do server, alguns items podem ser atrasados e as informações são exibidas nesta seção. Para mais detalhes, consulte a seção Fila.

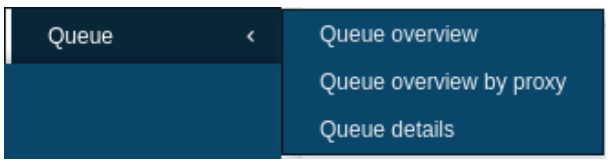
Note:
O Zabbix proxy não tem conhecimento dos períodos de manutenção; consulte Cálculo de filas durante a manutenção para obter detalhes.

Note:
A fila está disponível apenas se o Zabbix server estiver em execução. Os items não são contabilizados na fila se a interface do item ficar indisponível devido a problemas de conexão ou se o agent não estiver funcionando corretamente.

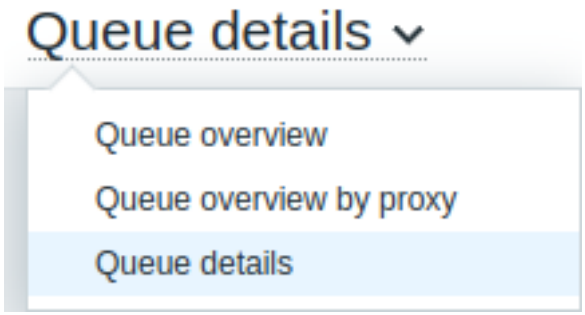
A seção *Administração* → *Fila* contém as seguintes páginas:

- Visão geral da fila — exibe a fila por tipo de item;
- Visão geral da fila por proxy — exibe a fila por proxy;
- Detalhes da fila — exibe uma lista de items atrasados.

A lista de páginas disponíveis aparece ao pressionar *Fila* na seção de menu *Administração*. Também é possível alternar entre as páginas usando um menu suspenso de título no canto superior esquerdo.



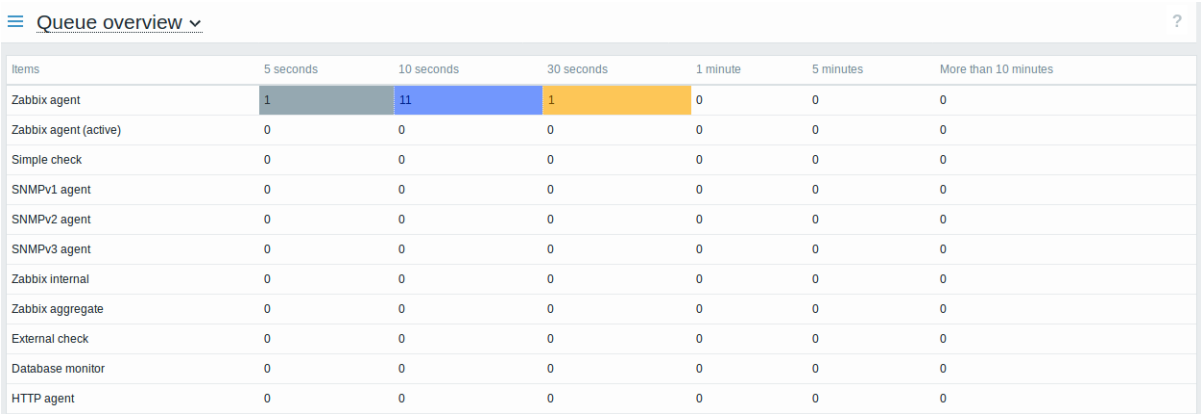
Menu de terceiro nível.



Menu suspenso de título.

Visão geral por tipo de item

Nesta tela é fácil localizar se o problema está relacionado a um ou vários tipos de itens.

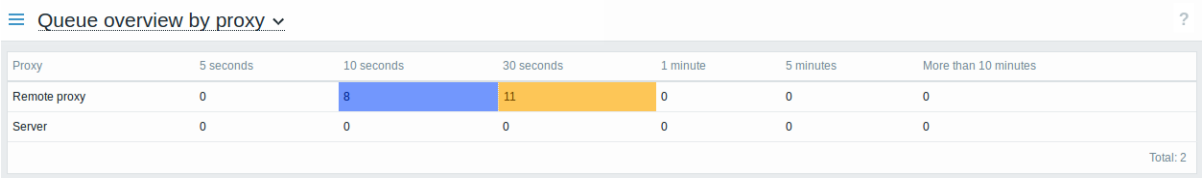

Queue overview

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

Cada linha contém um tipo de item. Cada coluna mostra o número de espera itens - aguardando 5-10 segundos/10-30 segundos/30-60 segundos/1-5 minutos/5-10 minutos ou mais de 10 minutos, respectivamente.

Visão geral por proxy

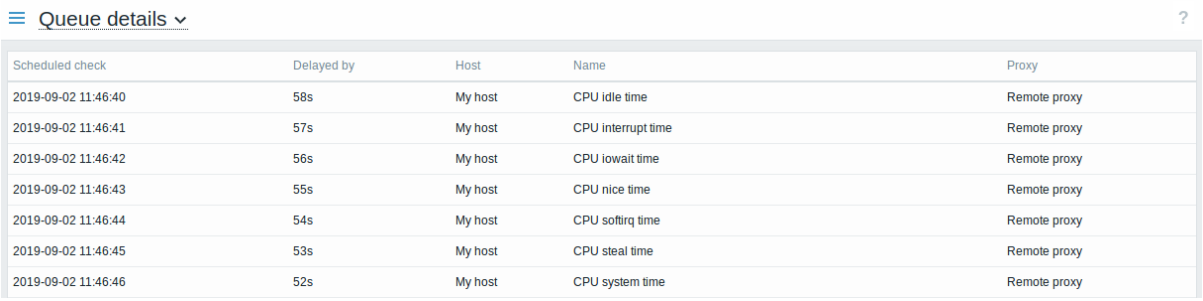
Nesta tela é fácil localizar se o problema está relacionado a um dos os proxies ou o servidor.



Cada linha contém um proxy, com o servidor por último na lista. Cada coluna mostra o número de itens em espera - aguardando 5-10 segundos/10-30 segundos/30-60 segundos/1-5 minutos/5-10 minutos ou mais de 10 minutos respectivamente.

Lista de items pendentes

Nesta tela, cada item pendente é listado.

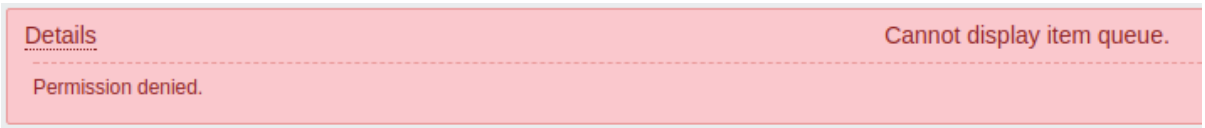


Dados exibidos:

Coluna	Descrição
Verificação agendada	O horário em que a verificação estava programada é exibido.
Atrasado por	O tempo de atraso é exibido.
Host	O host do item é exibido.
Nome	O nome do item pendente é exibido.
Proxy	O nome do proxy é exibido, se o host for monitorado por proxy.

Possíveis mensagens de erro

Você pode se deparar com uma situação em que nenhum dado é exibido e a seguinte mensagem de erro aparece:



A mensagem de erro neste caso é a seguinte:

```
Cannot display item queue. Permission denied
```

Isso acontece quando os parâmetros de configuração do PHP no arquivo *zabbix.conf.php* - `$ZBX_SERVER` ou ambos `$ZBX_SERVER` e `$ZBX_SERVER_PORT` - apontam para um servidor Zabbix existente que usa um banco de dados diferente.

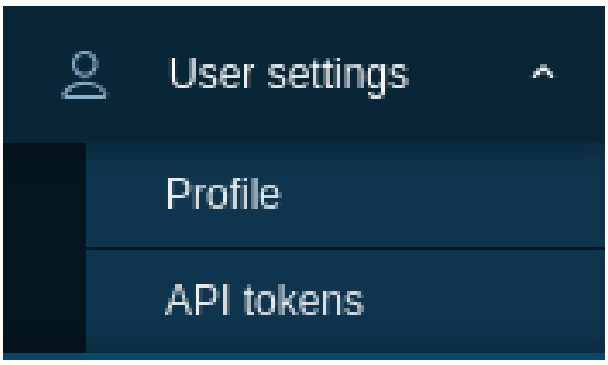
3 Configurações do usuário

Visão geral

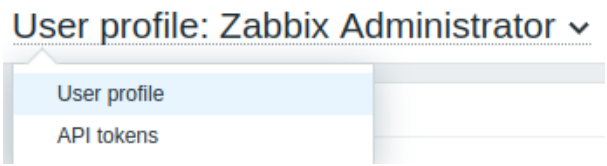
Dependendo das permissões do papel do usuário, a seção *Configurações do usuário* pode conter as seguintes páginas:

- *Perfil* ou *Perfil do usuário* - para personalizar certos recursos do frontend do Zabbix.
- *Tokens de API* - para gerenciar tokens de API atribuídos ao usuário atual.

A lista de páginas disponíveis aparece ao clicar no ícone de usuário próximo à parte inferior do menu do Zabbix (não disponível para o usuário *guest*). Também é possível alternar entre as páginas usando o menu suspenso de título no canto superior esquerdo.



Menu de terceiro nível.

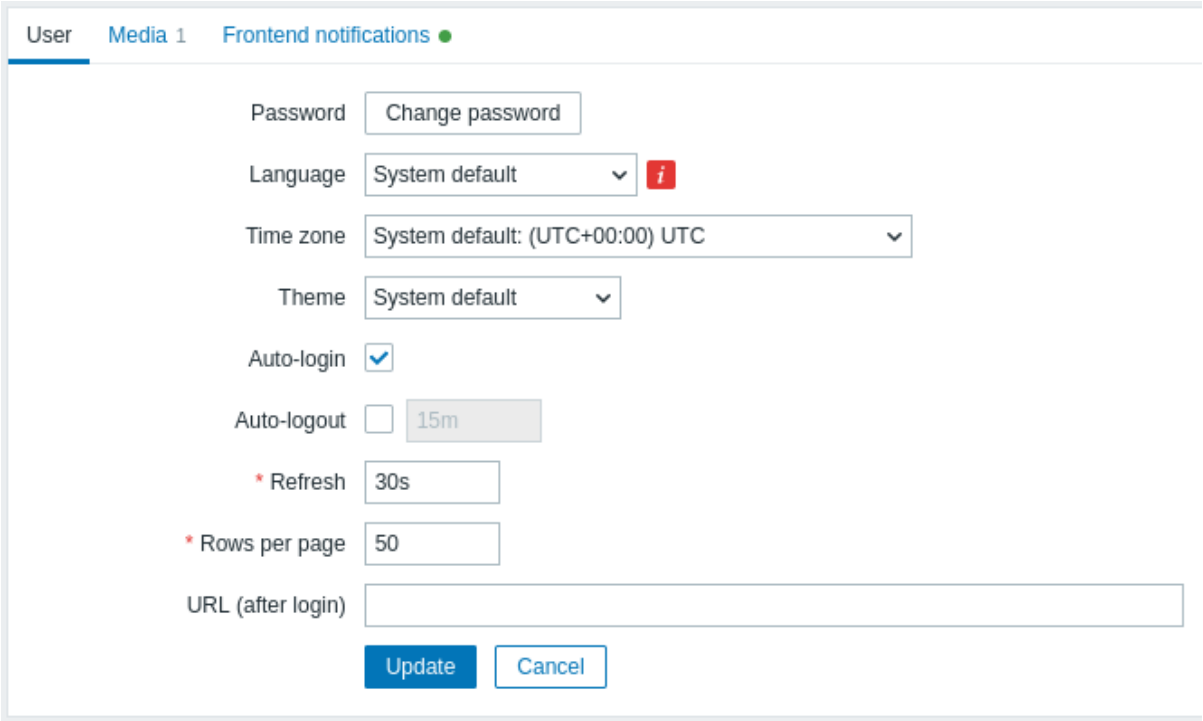


Menu suspenso de título.

Perfil do usuário

A seção *Perfil do usuário* fornece opções para definir um idioma de interface personalizado, tema de cor, número de linhas exibidas nas listas, etc. As alterações feitas aqui serão aplicadas apenas ao usuário atual.

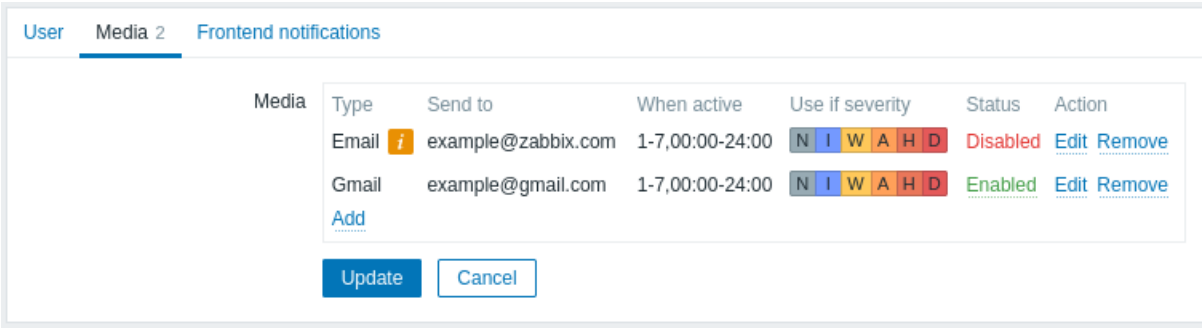
A aba *Usuário* permite definir várias preferências do usuário.



Parâmetro	Descrição
Senha	Clique no botão *Alterar senha* para abrir três campos: *Senha antiga*, *Nova senha*, *Nova senha (novamente)*. Ao alterar a senha com sucesso, o usuário será desconectado de todas as sessões ativas. Observe que a senha só pode ser alterada para usuários que utilizam a autenticação interna do Zabbix.
Idioma	Selecione o idioma da interface de sua preferência ou selecione **Padrão do sistema** para usar as configurações padrão do sistema. Selecionar *English (en_US)* também ativará o formato de data/hora dos EUA no frontend. Para mais informações, consulte Instalação de idiomas adicionais para o frontend.
Fuso horário	Selecione o fuso horário para substituir o fuso horário global no nível do usuário ou selecione **Padrão do sistema** para usar as configurações globais de fuso horário.
Tema	Selecione um tema de cor especificamente para o seu perfil: **Padrão do sistema** - usar as configurações padrão do sistema; **Azul** - tema azul padrão; **Escuro** - tema escuro alternativo; **Alto contraste claro** - tema claro com alto contraste; **Alto contraste escuro** - tema escuro com alto contraste.

Parâmetro	Descrição
Login automático	Marque esta caixa para que o Zabbix se lembre de você e faça o login automaticamente por 30 dias. Ao fazer login com *Lembrar por 30 dias* ativado: - O temporizador de logout automático é redefinido (a sessão persiste até o final do período de 30 dias). - O login automático é ativado nas visitas subsequentes dentro de 30 dias. Ao fazer login sem *Lembrar por 30 dias*: - O parâmetro de login automático é limpo. - O logout automático segue o tempo limite configurado normalmente. Cookies do navegador são usados para isso.
Logout automático	Com esta caixa marcada, você será desconectado automaticamente após a quantidade definida de segundos (mínimo de 90 segundos, máximo de 1 dia). Observe que, se fizer login com *Lembrar por 30 dias* ativado, o temporizador de logout automático é substituído, mantendo sua sessão ativa por todo o período. Sufixos de tempo são suportados, por exemplo: 90s, 5m, 2h, 1d. Observe que esta opção não funcionará nos seguintes casos: * Quando as páginas do menu *Monitoramento* realizam atualizações de informações em segundo plano. Caso páginas que estejam atualizando dados em um intervalo de tempo específico (dashboards, gráficos, últimos dados, etc.) sejam deixadas abertas, o tempo de vida da sessão é estendido, desabilitando, respectivamente, o recurso de logout automático. * Se fizer login com a opção *Lembrar por 30 dias* marcada. * Quando a autenticação é realizada via um Provedor de Identidade SSO (IdP), o logout automático encerra apenas a sessão do Zabbix. A sessão SSO com o IdP permanece ativa, portanto, o usuário pode conseguir fazer login novamente sem digitar uma senha. *Logout automático* também pode aceitar "0", o que significa que o recurso de logout automático será desabilitado após a atualização das configurações do perfil.
Atualizar	Defina com que frequência as informações nas páginas do menu *Monitoramento* serão atualizadas (mínimo de 0 segundos, máximo de 1 hora). Sufixos de tempo são suportados, por exemplo: 30s, 90s, 1m, 1h.
Linhas por página	Defina quantas linhas serão exibidas por página nas listas. Menos linhas (e menos registros para exibir) resultam em tempos de carregamento mais rápidos.
URL (após login)	Defina uma URL específica a ser exibida após o login. Em vez do *Dashboard* padrão, pode ser, por exemplo, a URL de *Monitoramento* > *Triggers*.

A aba *Mídia* permite especificar detalhes de mídia para o usuário, como tipos de mídia e endereços a serem usados e quando usá-los para entregar notificações.



Se o tipo de mídia foi desabilitado:

- um ícone de informação amarelo é exibido após o nome;
- "Desabilitado" é exibido na coluna *Status*.

Note:
Apenas usuários do tipo *Admin* e *Super admin* usuários podem alterar seus próprios detalhes de mídia.

Observe que para usuários provisionados:

- a mídia do usuário provisionado não pode ser excluída;
- a mídia do usuário provisionado pode ser desabilitada/habilitada;
- campos de mídia do usuário provisionado, como *Quando ativo*, *Usar se gravidade* e *Habilitado* podem ser editados manualmente;

- mídias de usuário adicionais podem ser adicionadas (por exemplo, endereços de e-mail adicionais) para usuários provisionados manualmente;
- mídias de usuário adicionadas manualmente podem ser excluídas.

A aba *Notificações no frontend* permite definir notificações globais.

Tokens de API

A seção *Tokens de API* permite visualizar os tokens atribuídos ao usuário, editar os detalhes do token e criar novos tokens. Esta seção está disponível para um usuário apenas se a ação *Gerenciar tokens de API* estiver permitida nas configurações da função de usuário.

Você pode filtrar os tokens de API por nome, data de expiração ou status (*Habilitado/Desabilitado*). Clique no status do token na lista para habilitar/desabilitar rapidamente um token. Você também pode habilitar/desabilitar vários tokens de uma vez, selecionando-os na lista e, em seguida, clicando nos botões *Habilitar/Desabilitar* abaixo da lista.

Attention:
Os usuários não podem visualizar o valor do *Token de autenticação* dos tokens atribuídos a eles no Zabbix. O valor do *Token de autenticação* é exibido apenas uma vez - imediatamente após a criação de um token. Se o token for perdido, ele deverá ser gerado novamente.

1 Notificações globais

Visão geral

As notificações globais fornecem uma maneira de exibir problemas em tempo real diretamente na tela atual dentro do frontend do Zabbix.

Sem as notificações globais, ao trabalhar fora das seções *Problemas* ou *Dashboard*, você não receberia nenhuma informação sobre os problemas atuais. As notificações globais garantem que essas informações sejam exibidas, independentemente da sua localização atual dentro do frontend do Zabbix.

As notificações globais incluem tanto a exibição de uma mensagem quanto a reprodução de um som.

Attention:
A reprodução automática de sons pode estar desabilitada (por padrão) em versões recentes dos navegadores. Nesses casos, você precisa habilitar essa configuração manualmente.

Configuração

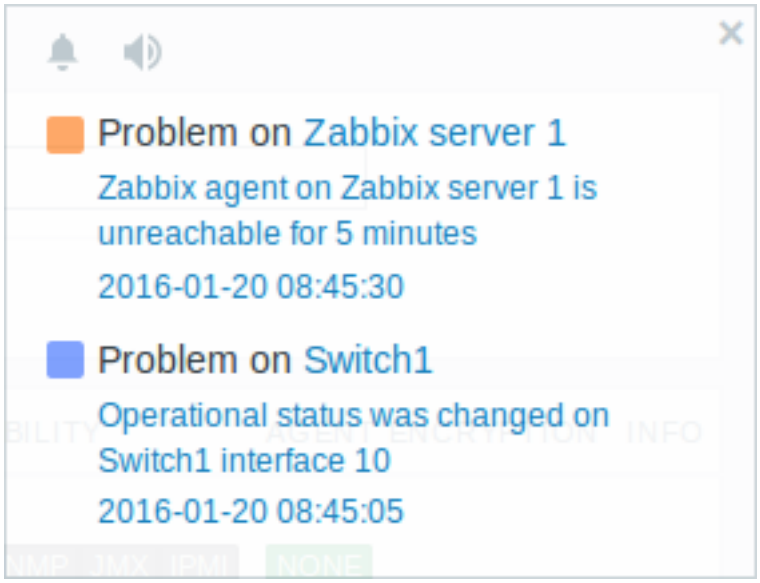
As notificações globais podem ser ativadas por usuário na aba **Notificações do frontend** da configuração do perfil.

profile_c.png

Parâmetro	Descrição
Notificações do frontend	Marque a caixa de seleção para ativar as notificações globais.
Tempo limite da mensagem	Defina a duração pela qual a mensagem será exibida. Por padrão, as mensagens permanecem na tela por 60 segundos. Sufixos de tempo são suportados, por exemplo: 30s, 5m, 2h, 1d.
Reproduzir som	Defina a duração pela qual o som será reproduzido. **Uma vez** - o som é reproduzido uma vez e completamente; **10 segundos** - o som é repetido por 10 segundos; **Tempo limite da mensagem** - o som é repetido enquanto a mensagem estiver visível.
Gravidade do trigger	Defina as gravidades dos triggers para as quais as notificações globais e os sons serão ativados. Você também pode selecionar sons apropriados para várias gravidades. Se nenhuma gravidade for marcada, nenhuma mensagem será exibida. Além disso, mensagens de recuperação só serão exibidas para gravidades marcadas. Por exemplo, se *Recuperação* e *Desastre* estiverem marcadas, notificações globais serão exibidas para problemas e recuperações de triggers de gravidade *Desastre*.
Mostrar problemas suprimidos	Marque a caixa de seleção para exibir notificações para problemas que, de outra forma, seriam suprimidos (não exibidos) devido à manutenção do host.

Mensagens globais exibidas

À medida que as mensagens chegam, elas são exibidas em uma seção flutuante no lado direito. Você pode reposicionar livremente esta seção arrastando o cabeçalho da seção.



Para esta seção, vários controles estão disponíveis:

- O botão **Adiar** silencia o som de alarme atualmente ativo;
- O botão **Silenciar/Ativar som** alterna entre reproduzir e não reproduzir os sons de alarme.

2 Som nos navegadores

Visão geral

O som é usado em notificações globais.

Para que os sons sejam reproduzidos no frontend do Zabbix, as *Notificações do frontend* devem estar habilitadas na guia *Notificações do frontend* do perfil do usuário, com todas as gravidades de trigger marcadas. Além disso, os sons devem ser ativados na janela pop-up de notificação global.

Se, por algum motivo, o áudio não puder ser reproduzido no dispositivo, o botão na janela pop-up de notificação global permanecerá permanentemente no estado ”mudo”, acompanhado da mensagem ”Não é possível suportar áudio de notificação para este dispositivo” ao passar o mouse sobre o botão .

Os sons, incluindo os clipes de áudio padrão, são suportados apenas no formato MP3.

Os sons do frontend do Zabbix foram testados com sucesso nos navegadores Firefox e Opera mais recentes no Linux, e nos navegadores Chrome, Firefox, Microsoft Edge e Opera no Windows.

Attention:
A reprodução automática de sons pode estar desativada (por padrão) nas versões mais recentes dos navegadores. Nesses casos, você precisa ativar essa configuração manualmente.

4 Pesquisa global

É possível pesquisar o frontend do Zabbix por hosts, grupos de hosts e modelos.

A caixa de entrada de pesquisa está localizada abaixo do logotipo do Zabbix no menu. O a pesquisa pode ser iniciada pressionando *Enter* ou clicando no botão pesquisa ícone.

Se houver um host que contenha a string inserida em qualquer parte do nome, uma lista suspensa aparecerá, listando todos esses hosts (com a correspondência parte destacada em laranja). A lista suspensa também listará um host se esse o nome visível do host corresponde ao nome técnico inserido como pesquisa corda; o host correspondente será listado, mas sem nenhum destaque.

Atributos pesquisáveis

Os hosts podem ser pesquisados pelas seguintes propriedades:

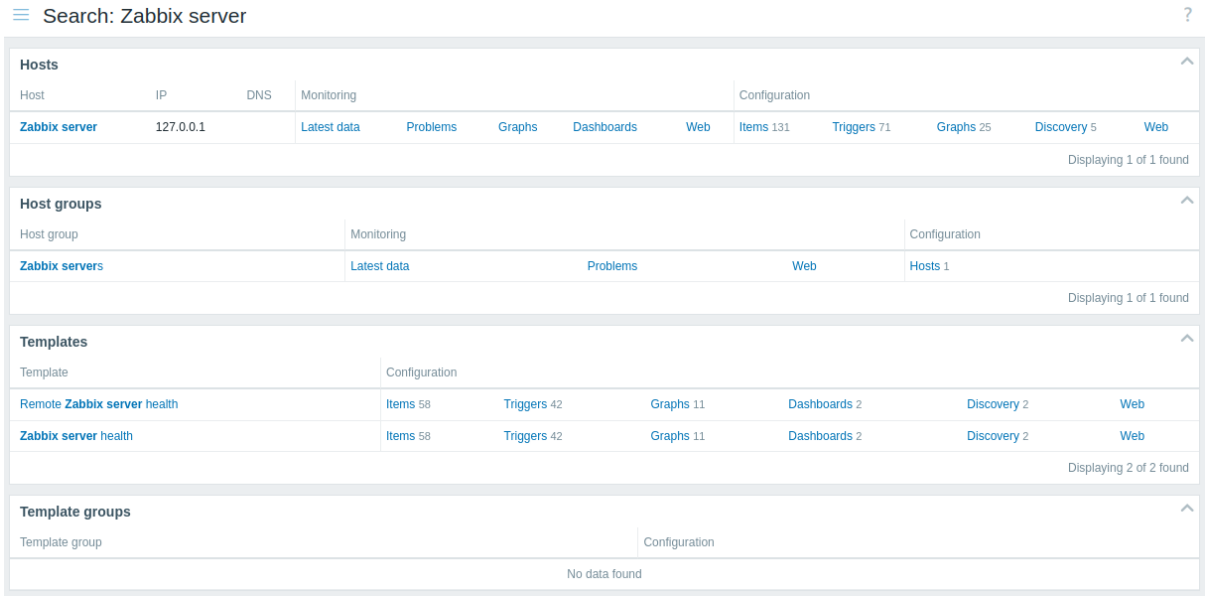
- Nome do host
- Nome visível
- Endereço IP
- Nome DNS

Os templates podem ser pesquisados por nome ou nome visível. Se você pesquisar por um nome que seja diferente do nome visível (de um template/host), nos resultados da pesquisa ele será exibido abaixo do nome visível entre parênteses.

Os grupos de host e de template podem ser pesquisados por nome. Especificar um grupo pai seleciona implicitamente todos os grupos aninhados.

Resultados da pesquisa

Os resultados da pesquisa consistem em quatro blocos separados para hosts, grupos de hosts, templates e grupos de templates.



É possível recolher/expandir cada bloco individualmente. A contagem de entradas é exibida na parte inferior de cada bloco, por exemplo, *Exibindo 13 de 13 encontrados*. Se não houver entradas, a contagem de entradas não é exibida. O total de entradas exibidas em um bloco é limitado a 100.

Cada entrada fornece links para dados de monitoramento e configuração. Veja a lista completa de links.

Para todos os dados de configuração (como items, triggers, gráficos), a quantidade de entidades encontradas é exibida por um número ao lado do nome da entidade, em cinza. **Observe** que, se houver zero entidades, nenhum número será exibido.

Hosts habilitados são exibidos em azul, hosts desabilitados em vermelho.

Links disponíveis

Para cada entrada, os seguintes links estão disponíveis:

- Hosts
  - Monitoramento
    - Últimos dados
    - Problemas
    - Gráficos
    - Dashboards do host
    - Cenários web
  - Configuração
    - Items
    - Triggers
    - Gráficos
    - Regras de descoberta
    - Cenários web
- Grupos de hosts
  - Monitoramento
    - Últimos dados
    - Problemas
    - Cenários web
  - Configuração
    - Hosts
- Templates
  - Configuração
    - Items
    - Triggers
    - Gráficos
    - Dashboards do template
    - Regras de descoberta
    - Cenários web
- Grupos de templates
  - Configuração
    - Templates

5 Modo de manutenção de front-end

Visão geral

É possível desabilitar temporariamente o frontend do Zabbix para restringir o acesso. Isso é útil para proteger o banco de dados do Zabbix contra alterações iniciadas por usuários, preservando sua integridade.

Enquanto o frontend do Zabbix estiver em modo de manutenção, você pode parar o banco de dados com segurança e realizar tarefas de manutenção.

Usuários de endereços IP definidos poderão interagir normalmente com o frontend durante o modo de manutenção.

Configuração

Para habilitar o modo de manutenção, abra o arquivo `maintenance.inc.php` (localizado em `/conf` do diretório de documentos HTML do Zabbix no servidor web) e descomente as seguintes linhas:

```
// Modo de manutenção.
define('ZBX_DENY_GUI_ACCESS', 1);

// Array de endereços IP, que têm permissão para se conectar ao frontend (opcional).
$ZBX_GUI_ACCESS_IP_RANGE = array('127.0.0.1');

// Mensagem exibida na tela de aviso (opcional).
$ZBX_GUI_ACCESS_MESSAGE = 'Estamos atualizando o banco de dados MySQL até às 15:00. Aguarde...';
```

Note:
Na maioria dos casos, o arquivo `maintenance.inc.php` está localizado em `/conf` do diretório de documentos HTML do Zabbix no servidor web. No entanto, alguns sistemas operacionais e servidores web podem usar um local diferente.
Por exemplo, o local para:

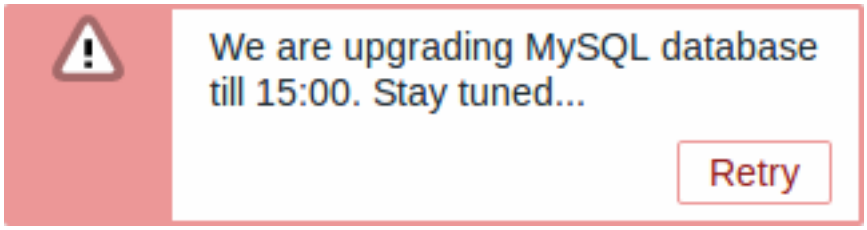
- SUSE e RedHat é `/etc/zabbix/web/maintenance.inc.php`.
- Sistemas baseados em Debian é `/usr/share/zabbix/conf/`.

Veja também Copiando arquivos PHP.

Parâmetro	Detalhes
ZBX_DENY_GUI_ACCESS	Se definido com qualquer valor, o modo de manutenção será ativado. Para desativar o modo de manutenção, comente ou exclua.
ZBX_GUI_ACCESS_IP_RANGE	Array de endereços IP que têm permissão para se conectar ao frontend (opcional). Por exemplo: `array('192.168.1.1', '192.168.1.2')`
ZBX_GUI_ACCESS_MESSAGE	Uma mensagem para informar os usuários sobre a manutenção (opcional). Se não for definida, a mensagem padrão *'Zabbix está em manutenção'* será usada.

Exibição

Os usuários verão a seguinte tela ao tentar acessar o frontend do Zabbix enquanto estiver no modo de manutenção. A tela é atualizada a cada 30 segundos para retornar ao estado normal sem intervenção do usuário quando a manutenção terminar.



Os endereços IP definidos em *ZBX_GUI_ACCESS_IP_RANGE* poderão acessar o frontend normalmente.

6 Parâmetros da página

Visão geral

A maioria das páginas da interface web do Zabbix suporta vários parâmetros HTTP GET que controlam o que será exibido. Eles podem ser passados especificando pares de parâmetro=valor após a URL, separados da URL por um ponto de interrogação (?) e entre si por e comercial (&).

Monitoramento > Problemas

Os seguintes parâmetros são suportados:

- `show` - opção de filtro *Mostrar*: 1 - problemas recentes, 2 - todos, 3 - em estado de problema
- `name` - opção de filtro *Problema*: string livre
- `severities` - opção de filtro *Gravidade*: array de gravidades selecionadas no formato 'severities[*]=*' (substitua * pelo nível de gravidade): 0 - não classificado, 1 - informação, 2 - aviso, 3 - média, 4 - alta, 5 - desastre
- `inventory` - opção de filtro *Inventário do host*: array de campos de inventário: [campo], [valor]
- `evaltype` - opção de filtro *Tags*, estratégia de filtragem de tags: 0 - E/Ou, 2 - Ou
- `tags` - opção de filtro *Tags*: array de tags definidas: [tag], [operador], [valor]
- `show_tags` - opção de filtro *Mostrar tags*: 0 - nenhuma, 1 - uma, 2 - duas, 3 - três
- `tag_name_format` - opção de filtro *Nome da tag*: 0 - nome completo, 1 - abreviado, 2 - nenhum
- `tag_priority` - opção de filtro *Prioridade de exibição da tag*: string separada por vírgulas da prioridade de exibição da tag
- `show_suppressed` - opção de filtro *Mostrar problemas suprimidos*: deve ser 'show_suppressed=1' para mostrar
- `acknowledgement_status` - opção de filtro *Status de reconhecimento*: 0 - todos, 1 - Não reconhecidos, 2 - Reconhecidos
- `acknowledged_by_me` - opção de propriedades de filtro - *Por mim*: 0 - Desativado, 1 - Ativado; isso é suportado apenas quando 'acknowledgement_status=2'
- `compact_view` - opção de filtro *Visualização compacta*: deve ser 'compact_view=1' para mostrar
- `highlight_row` - opção de filtro *Destacar linha inteira* (usar a cor do problema como cor de fundo para cada linha de problema): deve ser '1' para destacar; para versões do Zabbix anteriores à 7.0.19, esta opção é suportada apenas quando `compact_view=1`.

- `filter_name` - opção de propriedades de filtro *Nome*: string livre
- `filter_show_counter` - opção de propriedades de filtro *Mostrar número de registros*: 1 - mostrar, 0 - não mostrar
- `filter_custom_time` - opção de propriedades de filtro *Definir período de tempo personalizado*: 1 - definir, 0 - não definir
- `sort` - coluna de ordenação: clock, host, severity, name
- `sortorder` - ordem de classificação dos resultados: DESC - decrescente, ASC - crescente
- `age_state` - opção de filtro *Idade menor que*: deve ser 'age_state=1' para habilitar 'age'. Usado apenas quando 'show' é igual a 3.
- `age` - opção de filtro *Idade menor que*: dias
- `groupids` - opção de filtro *Grupos de hosts*: array de IDs de grupos de hosts
- `hostids` - opção de filtro *Hosts*: array de IDs de hosts
- `triggerids` - opção de filtro *Triggers*: array de IDs de triggers
- `show_timeline` - opção de filtro *Mostrar linha do tempo*: deve ser 'show_timeline=1' para mostrar
- `details` - opção de filtro *Mostrar detalhes*: deve ser 'details=1' para mostrar
- `from` - início do intervalo de datas, pode ser 'relativo' (ex.: now-1m). Usado apenas quando 'filter_custom_time' é igual a 1.
- `to` - fim do intervalo de datas, pode ser 'relativo' (ex.: now-1m). Usado apenas quando 'filter_custom_time' é igual a 1.

Modo quiosque

O modo quiosque nas páginas do frontend suportadas pode ser ativado usando parâmetros de URL. Por exemplo, em painéis:

- `/zabbix.php?action=dashboard.view&kiosk=1` - ativa o modo quiosque
- `/zabbix.php?action=dashboard.view&kiosk=0` - ativa o modo normal

Apresentação de slides

É possível ativar uma apresentação de slides no painel:

- `/zabbix.php?action=dashboard.view&slideshow=1` - ativar apresentação de slides

7 Definições

Visão geral

Embora muitas coisas no frontend possam ser configuradas usando o próprio frontend, algumas personalizações atualmente só são possíveis editando um arquivo de definições.

Este arquivo é o `defines.inc.php` localizado em /include do diretório de documentos HTML do Zabbix.

Parâmetros

Parâmetros neste arquivo que podem ser de interesse dos usuários:

- ZBX_MIN_PERIOD

Período mínimo do gráfico, em segundos. Um minuto por padrão.

- GRAPH_YAXIS_SIDE_DEFAULT

Localização padrão do eixo Y em gráficos simples e valor padrão para a caixa suspensa ao adicionar items a gráficos personalizados. Valores possíveis: 0 - esquerda, 1 - direita.

Padrão: 0

- ZBX_SESSION_NAME

String usada como o nome do cookie de sessão do frontend do Zabbix.

Padrão: zbx_sessionid

- ZBX_DATA_CACHE_TTL

Tempo limite de TTL em segundos usado para invalidar o cache de dados da resposta do Vault. Defina 0 para desabilitar o cache de resposta do Vault.

Padrão: 60

- SUBFILTER_VALUES_PER_GROUP

Número de valores de subfiltro por grupo (por exemplo, no subfiltro de dados mais recentes).

Padrão: 1000

- ZBX_MAX_WIDGET_LINES

Número máximo de linhas de widget a serem exibidas.

Padrão: 1000

8 Criando seu próprio tema

Visão geral

Por padrão, o Zabbix fornece vários temas predefinidos. Você pode siga o procedimento passo a passo fornecido aqui para criar seu ter. Sinta-se à vontade para compartilhar o resultado do seu trabalho com a comunidade Zabbix se você criou algo legal.

Passo 1

Para definir seu próprio tema, você precisará criar um arquivo CSS e salvá-lo em a pasta `assets/styles/` (por exemplo, *custom-theme.css*). Você pode ou copie os arquivos de um tema diferente e crie seu tema baseado nele ou comece do zero.

Passo 2

Adicione seu tema à lista de temas retornados pelo APP::getThemes() método. Você pode fazer isso substituindo o método ZBase::getThemes() em a aula de APP. Isso pode ser feito adicionando o seguinte código antes do chave de fechamento em *include/classes/core/APP.php*:

```
função estática pública getThemes() {
    return array_merge(parent::getThemes(), [
        'tema personalizado' => _('tema personalizado')
    ]);
}
```

::: não importante Observe que o nome especificado no primeiro par de aspas deve corresponder ao nome do arquivo do tema sem extensão. :::

Para adicionar vários temas, basta listá-los sob o primeiro tema, por exemplo:

```
função estática pública getThemes() {
    return array_merge(parent::getThemes(), [
        'tema personalizado' => _('tema personalizado'),
        'outro tema' => _('Outro tema'),
        'onemoretheme' => _('Mais um tema')
    ]);
}
```

Observe que todo tema, exceto o último, deve ter uma vírgula à direita.

Para alterar as cores do gráfico, a entrada deve ser adicionada no *graph_theme* tabela de banco de dados.

Etapa 3

Ative o novo tema.

No frontend do Zabbix, você pode definir este tema para ser o padrão ou altere seu tema no perfil do usuário.

Aproveite o novo visual e toque!

9 Modo de depuração

Visão geral

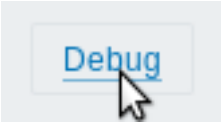
O modo de depuração pode ser usado para diagnosticar problemas de desempenho com páginas frontend.

Configuração

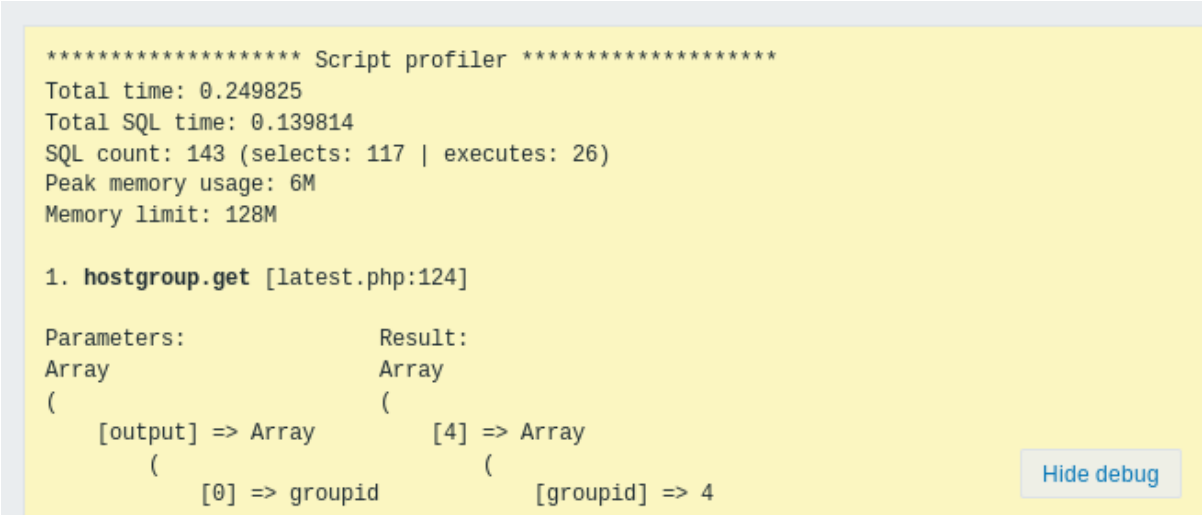
O modo de depuração pode ser ativado para usuários individuais que pertencem a um grupo de usuários:

- ao configurar um grupo de usuários;
- ao visualizar grupos de usuários configurados.

Quando o *Modo de depuração* está ativado para um grupo de usuários, seus usuários verão um botão *Depurar* no canto inferior direito da janela do navegador:



Clicar no botão *Depurar* abre uma nova janela abaixo do conteúdo da página, que contém as estatísticas SQL da página, juntamente com uma lista de chamadas de API e instruções SQL individuais:



Em caso de problemas de desempenho com a página, esta janela pode ser usada para pesquisar a causa raiz do problema.

Warning:
O *Modo de depuração* ativado afeta negativamente o desempenho do frontend.

10 Cookies usados pelo Zabbix

Visão geral

Esta página fornece uma lista de cookies usados pelo Zabbix.

Nome	Descrição	Valores	Expira/Max-Age	HttpOnly[1]	Secure[2]
ZBX_SESSIONID	Dados de sessão do frontend do Zabbix, armazenados como JSON codificado em Base64		Sessão (expira quando a sessão de navegação termina)	+	+ (somente se HTTPS estiver habilitado no servidor web)
tab	Número da aba ativa; este cookie é usado apenas em páginas com várias abas (por exemplo, página de configuração de *Host*, *Trigger* ou *Ação*) e é criado quando um usuário navega de uma aba primária para outra aba (como a aba *Tags* ou *Dependências*). 0 é usado para a aba primária.	Exemplo: 1	Sessão (expira quando a sessão de navegação termina)	-	-
browserwarning_ignore	Indica se um aviso sobre o uso de um navegador desatualizado deve ser ignorado.	yes	Sessão (expira quando a sessão de navegação termina)	-	-

Nome	Descrição	Valores	Expira/Max-Age	HttpOnly[1]	Secure[2]
system-message-ok	Uma mensagem para mostrar assim que a página for recarregada.	Mensagem em texto simples	Sessão (expira quando a sessão de navegação termina) ou assim que a página for recarregada	+	-
system-message-error	Uma mensagem de erro para mostrar assim que a página for recarregada.	Mensagem em texto simples	Sessão (expira quando a sessão de navegação termina) ou assim que a página for recarregada	+	-

Note:
Forçar o flag 'HttpOnly' nos cookies do Zabbix por uma diretiva do servidor web não é suportado.

Notas de rodapé

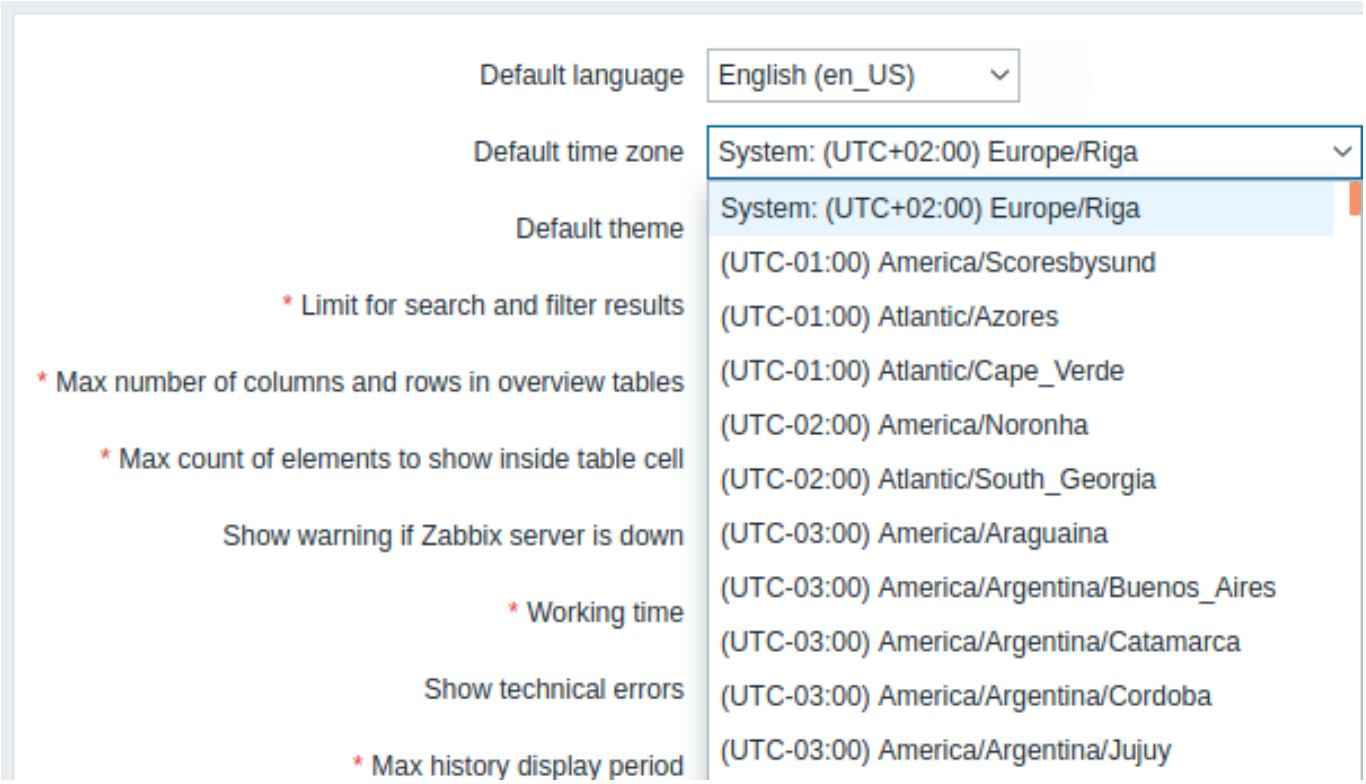
[1] Quando `HttpOnly` é 'true', o cookie será acessível apenas pelo protocolo HTTP. Isso significa que o cookie não estará acessível por linguagens de script, como JavaScript. Esta configuração pode ajudar efetivamente a reduzir o roubo de identidade por meio de ataques XSS (embora não seja suportada por todos os navegadores).

[2] `Secure` indica que o cookie deve ser transmitido apenas por uma conexão HTTPS segura a partir do cliente. Quando definido como 'true', o cookie só será definido se existir uma conexão segura.

11 Fuso horário

Visão geral

O fuso horário do frontend pode ser definido globalmente no frontend e ajustado para usuários individuais.



Se *System* for selecionado, o fuso horário do servidor web será usado para o frontend (incluindo o valor de 'date.timezone' do php.ini, se definido), enquanto o Zabbix server usará o fuso horário da máquina em que está sendo executado.

Note:
O Zabbix server só usará o fuso horário global/usuário especificado ao expandir macros em notificações (por exemplo, {EVENT.TIME} pode expandir para um fuso horário diferente por usuário) e para o limite de tempo quando as notificações são enviadas (veja a configuração "Quando ativo" na configuração de mídia do usuário).

A escolha do fuso horário não afeta o formato de data/hora do frontend. Em vez disso, você pode ajustar o idioma da interface (durante a instalação ou em configurações do usuário) - selecionar *English (en_US)* também habilitará o formato de data/hora dos EUA no frontend.

Configuração

O fuso horário global:

- pode ser definido manualmente ao instalar o frontend
- pode ser modificado em *Administração → Geral → GUI*

Fuso horário no nível do usuário:

- pode ser definido ao configurar/atualizar um usuário
- pode ser definido por cada usuário em seu perfil de usuário

Veja também: Alinhando fusos horários ao usar intervalos de agendamento.

12 Redefinindo senha

Visão geral Esta seção descreve os passos para redefinir senhas de usuários no Zabbix.

Etapas Procure o administrador do Zabbix se você esqueceu sua senha do Zabbix e não consegue fazer login.

Um usuário Superadministrador pode alterar as senhas de todos os usuários no formulário de configuração de usuários.

Se um Superadministrador esqueceu sua senha e não consegue fazer login, a seguinte consulta SQL deve ser executada para aplicar a senha padrão ao usuário Superadministrador (substitua 'Admin' pelo nome de usuário apropriado do Superadministrador):

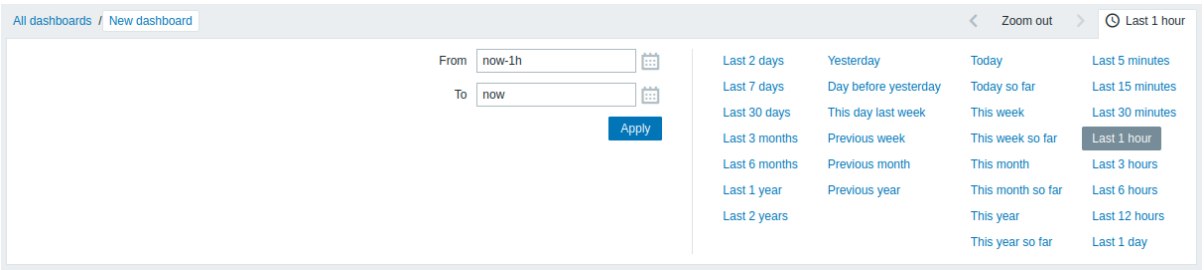
```
UPDATE users SET passwd = '$2a$10$ZXIvHAEP2ZM.dLXTm6uPHOMVlARXX7cqjbhM6Fn0cANzkCQBWpMrS' WHERE username =
```

Após executar esta consulta, a senha do usuário será definida como *zabbix*. Certifique-se de alterar a senha padrão no primeiro login.

13 Seletor de período de tempo

Visão geral

O seletor de *Período de tempo* permite selecionar períodos frequentemente necessários com um clique do mouse. Ele pode ser expandido ou recolhido clicando na guia *Período de tempo* no canto superior direito.



Opções como *Hoje*, *Esta semana*, etc., exibem todo o período, incluindo as horas/dias no futuro. Opções como *Hoje até agora*, *Esta semana até agora*, etc., exibem apenas as horas já passadas.

Uma vez selecionado um período, ele pode ser movido para frente e para trás no tempo clicando nos botões de seta . O botão *Reduzir zoom* permite reduzir o zoom do período em 50% em cada direção.

> **Note:**
> Para gráficos (exceto aqueles com o período de tempo *Personalizado*), selecionar o período de tempo exibido também é possível destacando uma área no gráfico com o botão esquerdo do mouse. Assim que você soltar o botão esquerdo do mouse, o gráfico dará zoom na área destacada. Reduzir o zoom também é possível clicando duas vezes no gráfico.

Os campos *De/Até* exibem o período selecionado em sintaxe de tempo absoluto (no formato `Y-m-d H:i:s`) ou sintaxe de tempo relativo. Um período de tempo relativo pode conter uma ou várias operações matemáticas (- ou +), por exemplo, `now-1d` ou `now-1d-2h+5m`.

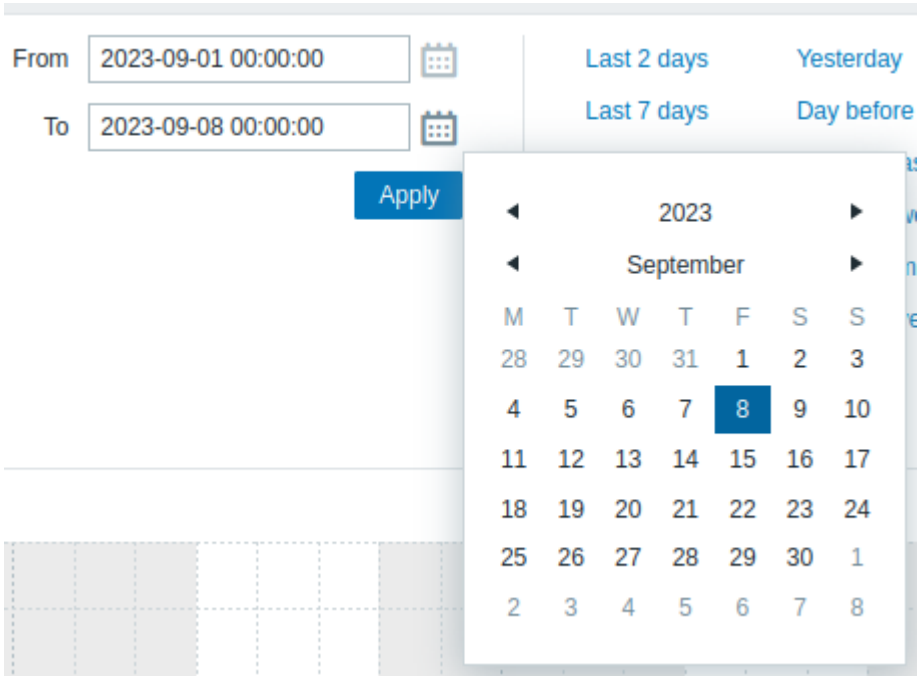
As seguintes abreviações de tempo relativo são suportadas:

- `now`
- `s` (segundos)
- `m` (minutos)
- `h` (horas)
- `d` (dias)
- `w` (semanas)
- `M` (meses)
- `y` (anos)

A precisão é suportada no seletor de *Período de tempo* (por exemplo, `/M` em `now-1d/M`). Detalhes da precisão:

Precisão	De	Até
`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`	Segunda-feira da semana 00:00:00	Domingo da semana 23:59:59
`M`	Primeiro dia do mês 00:00:00	Último dia do mês 23:59:59
`y`	1º de janeiro do ano 00:00:00	31 de dezembro do ano 23:59:59

Também é possível selecionar um período de tempo usando o *Seletor de datas*. Para abri-lo, clique no ícone de calendário ao lado dos campos *De/Até*.



Note:
Dentro do seletor de datas, você pode navegar entre ano/mês/data usando `Tab`, `Shift+Tab` e as setas do teclado. Pressionar `Enter` confirma a seleção.

Exemplos

De	Para	Período selecionado
`now/d`	`now/d`	00:00 - 23:59 hoje
`now/d`	`now/d+1d`	00:00 hoje - 23:59 amanhã
`now/w`	`now/w`	Segunda-feira 00:00:00 - Domingo 23:59:59 desta semana
`now-1y/w`	`now-1y/w`	A semana de Segunda-feira 00:00:00 - Domingo 23:59:59 há um ano

Attention:
Usar ”now/M+1M” para o parâmetro *Para* pode adicionar 31 dias, o que pode resultar no deslocamento da data em 1-3 dias, dependendo do número de dias no mês. Por exemplo, se usado em janeiro, o resultado pode ser 02 de março em vez de 28 de fevereiro, como esperado. Para evitar esse problema, use ”now/M-3d+1M/M”, que ajusta corretamente para o comprimento do mês. Da mesma forma, se configurar *De* para retroceder, use ”now/M+3d-1M/M”.

19 Melhores práticas

Visão geral Esta seção descreve as melhores práticas para configurar o Zabbix.

Embora essas práticas não sejam obrigatórias para o funcionamento do Zabbix, sua implementação é altamente recomendada para uma experiência de uso ideal e segura.

1 Melhores práticas de segurança

Visão geral

Esta seção contém as melhores práticas para configurar o Zabbix de forma segura.

As práticas nesta seção não são obrigatórias para o funcionamento do Zabbix, mas são recomendadas para uma melhor segurança do sistema.

Codificação UTF-8

UTF-8 é a única codificação suportada pelo Zabbix. Sabe-se que funciona sem falhas de segurança. Os usuários devem estar cientes de que existem problemas de segurança conhecidos ao usar algumas das outras codificações.

Caminhos do instalador do Windows

Ao usar instaladores do Windows, é recomendável usar os caminhos padrão fornecidos pelo instalador. O uso de caminhos personalizados sem as permissões adequadas pode comprometer a segurança da instalação.

Macros em scripts globais definidos pelo usuário

Para aumentar a segurança, recomenda-se usar funções de macro em vez de macros simples em scripts globais definidos pelo usuário, pois as macros não são automaticamente escapadas.

Avisos de Segurança do Zabbix e banco de dados CVE

Consulte *Avisos de Segurança do Zabbix e banco de dados CVE*.

Modelos de e-mail HTML

Ao criar ou editar modelos de mensagem usados para e-mails HTML, sempre envolva cada macro com a função de macro `htmlencode()`. Por exemplo:

```
<b>Problema iniciado</b> em {{EVENT.TIME}.htmlencode()} em {{EVENT.DATE}.htmlencode()}<br><b>Nome do probl
{{EVENT.NAME}.htmlencode()}<br><b>Host:</b> {{HOST.NAME}.htmlencode()}
<br><b>Gravidade:</b>
{{EVENT.SEVERITY}.htmlencode()}<br><b>Dados operacionais:</b> {{EVENT.OPDATA}.htmlencode()}<br><b>ID do pr
```

O uso de `htmlencode()` garante que quaisquer caracteres HTML nos valores das macros sejam escapados e evita a injeção de HTML nas notificações (por exemplo, um invasor inserindo um link malicioso/de phishing em uma notificação).

Nota: as mensagens de e-mail HTML padrão fornecidas pelo Zabbix já aplicam `htmlencode()` às macros. Esta orientação se aplica ao editar modelos existentes ou criar novos — verifique se as macros estão codificadas antes de usar um modelo para enviar e-mails HTML.

1 Controle de acesso

Visão geral

Esta seção contém as melhores práticas para configurar o controle de acesso de forma segura.

Princípio do menor privilégio

As contas de usuário, em todos os momentos, devem operar com o menor número de privilégios possível. Isso significa que as contas de usuário no frontend do Zabbix, usuários do banco de dados ou o usuário para os processos do Zabbix server/proxy/agent devem ter apenas os privilégios essenciais para executar as funções pretendidas.

Attention:
Conceder privilégios extras ao usuário 'zabbix' permitirá que ele acesse arquivos de configuração e execute operações que podem comprometer a segurança da infraestrutura.

Ao configurar privilégios de contas de usuário, os tipos de usuário do frontend do Zabbix devem ser considerados. Observe que, embora o tipo de usuário *Admin* tenha menos privilégios do que o tipo de usuário *Super Admin*, ele ainda pode gerenciar configurações e executar scripts personalizados.

Note:
Algumas informações estão disponíveis mesmo para usuários sem privilégios. Por exemplo, enquanto *Alertas* → *Scripts* está disponível apenas para usuários *Super Admin*, os scripts também podem ser recuperados por meio da API do Zabbix. Nesse caso, limitar as permissões de script e excluir informações confidenciais dos scripts (por exemplo, credenciais de acesso) pode ajudar a evitar a exposição de informações confidenciais disponíveis em scripts globais.

Usuário seguro para o Zabbix agent

Por padrão, os processos do Zabbix server, proxy e agent (ou agent 2) compartilham um usuário `zabbix`. Para evitar que o Zabbix agent/agent 2 (executando na mesma máquina que o server/proxy) acesse detalhes confidenciais na configuração do server/proxy (por exemplo, credenciais do banco de dados), o agent deve ser executado sob um usuário diferente:

Para o Zabbix agent:

1. Crie um grupo e usuário seguro (por exemplo, `zabbix-agent`).
2. Defina este usuário no parâmetro User do arquivo de configuração do agent.
3. Reinicie o agent para que as permissões sejam atribuídas ao novo usuário.

Para o Zabbix agent 2, a configuração deve ser aplicada no nível do serviço, pois o arquivo de configuração do agent 2 não suporta o parâmetro `User`. Para um exemplo, consulte ZBX-26442.

Revogue o acesso de gravação à configuração SSL (Windows)

Se você compilou o agent do Zabbix no Windows, com o OpenSSL localizado em um diretório não protegido (por exemplo, `C:\zabbix`, `c:\openssl-64bit`, `C:\OpenSSL-Win64-111-static` ou `C:\dev\openssl`), certifique-se de revogar o acesso de gravação de usuários não administradores a esse diretório. Caso contrário, o agent carrega as configurações SSL de um caminho que pode ser modificado por usuários sem privilégios, resultando em uma potencial vulnerabilidade de segurança.

Reforçando a segurança dos componentes do Zabbix

Algumas funcionalidades podem ser desativadas para reforçar a segurança dos componentes do Zabbix:

- a execução de scripts globais no Zabbix server pode ser desativada definindo EnableGlobalScripts=0 na configuração do server;
- a execução de scripts globais no Zabbix proxy está desativada por padrão (pode ser ativada definindo EnableRemoteCommands=1 na configuração do proxy);
- a execução de scripts globais nos agentes Zabbix está desativada por padrão (pode ser ativada adicionando um parâmetro AllowKey=system.run[<command>,*] para cada comando permitido na configuração do agent);
- a autenticação HTTP do usuário pode ser desativada definindo `$ALLOW_HTTP_AUTH=false` no arquivo de configuração do frontend (zabbix.conf.php). Observe que reinstalar o frontend (executando setup.php) removerá este parâmetro.

Acesso ao caminho UNC no Windows pelo agent do Zabbix

Agents do Zabbix no Windows seguem caminhos UNC (compartilhamentos SMB como `\\server\share\file.txt`) em items como `vfs.file.*`, `vfs.dir.*`, `modbus.get` e `perf_counter*`. Isso pode ser um risco de segurança em alguns contextos.

Quando o Windows é solicitado a acessar um caminho UNC, ele tenta se autenticar nesse servidor. Isso significa que uma solicitação maliciosa ao agent do Zabbix pode expor o hash NTLM ao servidor do solicitante. Os usuários podem mitigar isso com os parâmetros de configuração AllowKey e DenyKey, se necessário.

1 Protegendo o MySQL/MariaDB

Visão geral

Esta seção contém as melhores práticas para proteger um banco de dados MySQL/MariaDB.

Note:
Para uma configuração básica, consulte as instruções padrão de criação do banco de dados MySQL/MariaDB, que incluem a criação do usuário 'zabbix' com privilégios totais no banco de dados do Zabbix. Este usuário é o proprietário do banco de dados e também possui os privilégios necessários para modificar a estrutura do banco de dados ao atualizar o Zabbix.

Para melhorar a segurança, recomenda-se criar funções e usuários adicionais no banco de dados com privilégios mínimos. Essas funções e usuários devem ser configurados com base no princípio do menor privilégio, ou seja, eles devem ter apenas os privilégios essenciais para executar as funções pretendidas.

Criando funções de usuário

Crie as seguintes funções com os privilégios correspondentes:

- **zbx_srv** - função para executar o Zabbix server e proxy:

```
CREATE ROLE 'zbx_srv';
GRANT DELETE, INSERT, SELECT, UPDATE ON zabbix.* TO 'zbx_srv';
FLUSH PRIVILEGES;
```

- **zbx_web** - função para executar o Zabbix frontend e API:

```
CREATE ROLE 'zbx_web';
GRANT DELETE, INSERT, SELECT, UPDATE ON zabbix.* TO 'zbx_web';
FLUSH PRIVILEGES;
```

- **zbx_bckp** - função para backup de tabelas:

```
CREATE ROLE 'zbx_bckp';
GRANT LOCK TABLES, TRIGGER, SELECT ON zabbix.* TO 'zbx_bckp';
GRANT process ON *.* TO 'zbx_bckp';
FLUSH PRIVILEGES;
```

Note:
A restauração e atualização de tabelas devem ser realizadas pelo proprietário do banco de dados.

- **zbx_part** - função com um conjunto reduzido de privilégios para particionamento do banco de dados; observe que esta função só pode ser criada após o banco de dados ter sido criado, pois concede privilégios em tabelas específicas do banco de dados:

```
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';
-- Para MariaDB: pule a próxima linha (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;
```

Após criar as funções, elas podem ser atribuídas aos usuários.

Atribuindo funções de usuário

Para atribuir as funções de usuário criadas, crie usuários e atribua as funções relevantes a eles. Substitua <user>, <host>, <role> e <password> conforme necessário.

```
CREATE USER '<user>'@'<host>' IDENTIFIED BY '<password>';
GRANT '<role>' TO '<user>'@'<host>';
SET DEFAULT ROLE '<role>' TO '<user>'@'<host>';
-- Para MariaDB: SET DEFAULT ROLE '<role>' FOR '<user>'@'<host>'
FLUSH PRIVILEGES;
```

Por exemplo, para criar e atribuir a função para executar o Zabbix server e proxy:

```
CREATE USER 'usr_srv'@'localhost' IDENTIFIED BY 'password';
GRANT 'zbx_srv' TO 'usr_srv'@'localhost';
SET DEFAULT ROLE ALL TO 'usr_srv'@'localhost';
FLUSH PRIVILEGES;
```

2 Protegendo o PostgreSQL/TimescaleDB

Visão geral

Esta seção contém as melhores práticas para proteger um banco de dados PostgreSQL.

> **Note:**
> Para uma configuração básica, consulte as instruções padrão de criação de banco de dados PostgreSQL, que incluem a criação do usuário 'zabbix' com privilégios totais no banco de dados Zabbix. Este usuário é o proprietário do banco de dados e também possui os privilégios necessários para modificar a estrutura do banco de dados ao atualizar o Zabbix.

Para melhorar a segurança, recomenda-se configurar um padrão de uso de schema seguro, bem como criar funções e usuários de banco de dados adicionais com privilégios mínimos. Essas funções e usuários devem ser configurados com base no princípio do menor privilégio, ou seja, eles devem ter apenas os privilégios essenciais para executar as funções pretendidas.

Configuração do banco de dados

Crie o usuário que será o proprietário do banco de dados e crie o banco de dados Zabbix; o proprietário do banco de dados é o usuário especificado na criação do banco de dados:

```
createuser -U postgres -h localhost --pwprompt usr_owner
createdb -U postgres -h localhost -O usr_owner -E Unicode -T template0 zabbix
```

> **Note:**
> Uma instalação limpa ou atualização do banco de dados deve ser realizada pelo proprietário do banco de dados. Isso ocorre porque o direito de remover um objeto do banco de dados ou alterar sua definição é um privilégio inerente ao proprietário do banco de dados e não pode ser concedido ou revogado.

> **Attention:**
> Os comandos a seguir nesta página devem ser executados enquanto a conexão com o PostgreSQL for feita especificamente para o banco de dados `zabbix`.

Crie o schema `zabbix` e defina o proprietário do banco de dados (`usr_owner`) como proprietário deste schema:

```
CREATE SCHEMA zabbix AUTHORIZATION usr_owner;
```

Configure um padrão de uso de schema seguro:

```
REVOKE CREATE ON SCHEMA public FROM PUBLIC;
REVOKE ALL ON DATABASE zabbix FROM PUBLIC;
-- Nota: search_path deve apontar para o schema "zabbix":
ALTER ROLE ALL IN DATABASE zabbix SET search_path = "zabbix";
```

Após configurar o banco de dados, prossiga para a criação de funções de usuário.

Criando funções de usuário

Crie as seguintes funções com os privilégios correspondentes:

- **zbx_srv** - função para executar o Zabbix server e proxy:

```
CREATE ROLE zbx_srv;
GRANT CONNECT ON DATABASE zabbix TO zbx_srv;
GRANT USAGE ON SCHEMA zabbix TO zbx_srv;
ALTER DEFAULT PRIVILEGES FOR ROLE usr_owner IN SCHEMA zabbix GRANT DELETE, INSERT, SELECT, UPDATE ON TABLE
ALTER DEFAULT PRIVILEGES FOR ROLE usr_owner IN SCHEMA zabbix GRANT SELECT, UPDATE, USAGE ON sequences TO z
```

- **zbx_web** - função para executar o Zabbix frontend e API:

```
CREATE ROLE zbx_web;
GRANT CONNECT ON DATABASE zabbix TO zbx_web;
GRANT USAGE ON SCHEMA zabbix TO zbx_web;
```

```
ALTER DEFAULT PRIVILEGES FOR ROLE usr_owner IN SCHEMA zabbix GRANT DELETE, INSERT, SELECT, UPDATE ON TABLE
ALTER DEFAULT PRIVILEGES FOR ROLE usr_owner IN SCHEMA zabbix GRANT SELECT, UPDATE, USAGE ON sequences TO z
```

- **zbx_bckp** - função para backup de tabelas:

```
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:**
> A restauração de tabelas só é possível pelo proprietário do banco de dados.

Após criar as funções, elas podem ser atribuídas aos usuários.

Atribuindo funções de usuário

Para atribuir as funções de usuário criadas, crie usuários e atribua as funções relevantes a eles. Substitua <user>, <role> e <password> conforme necessário.

```
CREATE USER <user> WITH ENCRYPTED password '<password>';
GRANT <role> TO <user>;
```

Por exemplo, para criar e atribuir a função para executar o Zabbix server e proxy:

```
CREATE USER usr_srv WITH ENCRYPTED password 'password';
GRANT zbx_srv TO usr_srv;
```

Particionamento de banco de dados com TimescaleDB

O particionamento de banco de dados é facilitado pelo TimescaleDB. Para utilizar o TimescaleDB, o Zabbix server requer privilégios de proprietário do banco de dados.

Se o schema zabbix do PostgreSQL já foi criado no banco de dados zabbix, você pode habilitar o TimescaleDB com o seguinte comando:

```
echo "CREATE EXTENSION IF NOT EXISTS timescaledb WITH SCHEMA zabbix CASCADE;" | sudo -u postgres psql zabb
```

3 Protegendo o Oracle

Visão geral

Esta seção contém as melhores práticas para proteger um banco de dados Oracle.

> **Note:**
> Para uma configuração básica, consulte as instruções padrão de criação do banco de dados Oracle, que incluem a criação do usuário 'zabbix' com privilégios totais no banco de dados Zabbix. Este usuário é o proprietário do banco de dados e também possui os privilégios necessários para modificar a estrutura do banco de dados ao atualizar o Zabbix.

Para melhorar a segurança, recomenda-se criar usuários adicionais do banco de dados com privilégios mínimos. Esses usuários devem ser configurados com base no princípio do menor privilégio, ou seja, eles devem ter apenas os privilégios essenciais para executar as funções pretendidas.

> **Attention:**
> O suporte ao Oracle DB está obsoleto desde o Zabbix 7.0.

Criando usuários

Assumindo que o proprietário do banco de dados plugável (PDB) é usr_owner, recomenda-se criar dois usuários adicionais com os privilégios correspondentes (para operações diárias):

- **usr_srv** - usuário para executar o Zabbix server;
- **usr_web** - usuário para executar o Zabbix frontend e API.

Esses usuários devem ser criados pelo proprietário do PDB (usr_owner) usando os seguintes comandos:

```
CREATE USER usr_srv IDENTIFIED BY "usr_srv" DEFAULT TABLESPACE "usr_owner" TEMPORARY TABLESPACE temp;
CREATE USER usr_web IDENTIFIED BY "usr_web" DEFAULT TABLESPACE "usr_owner" TEMPORARY TABLESPACE temp;

GRANT CREATE SESSION, DELETE ANY TABLE, INSERT ANY TABLE, SELECT ANY TABLE, UPDATE ANY TABLE, SELECT ANY S
GRANT CREATE SESSION, DELETE ANY TABLE, INSERT ANY TABLE, SELECT ANY TABLE, UPDATE ANY TABLE, SELECT ANY S
```

Note:
A restauração e atualização da tabela devem ser realizadas pelo proprietário do banco de dados.

Após criar os usuários, prossiga para a criação de sinônimos.

Gerando sinônimos

O script abaixo cria sinônimos, para que `usr_srv` e `usr_web` possam acessar tabelas no schema `usr_owner` sem especificar explicitamente o schema.

```
BEGIN
FOR x IN (SELECT owner,table_name FROM all_tables WHERE owner ='usr_owner')
LOOP
  EXECUTE IMMEDIATE 'CREATE OR REPLACE SYNONYM usr_srv.'|| x.table_name ||' FOR '||x.owner||'.'|| x.table_
  EXECUTE IMMEDIATE 'CREATE OR REPLACE SYNONYM usr_web.'|| x.table_name ||' FOR '||x.owner||'.'|| x.table_
END LOOP;
END;
/
```

Attention:
Este script deve ser executado toda vez após a estrutura do banco de dados do Zabbix ser criada ou alterada (por exemplo, após atualizar o Zabbix, se algumas tabelas foram criadas ou renomeadas).

2 Criptografia

Visão geral

Esta seção contém as melhores práticas para configurar a criptografia de forma segura.

Configurando SSL para o frontend do Zabbix

Em sistemas baseados em RHEL, instale o pacote `mod_ssl`:

```
dnf install mod_ssl
```

Crie um diretório para as chaves SSL:

```
mkdir -p /etc/httpd/ssl/private
chmod 700 /etc/httpd/ssl/private
```

Crie o certificado SSL:

```
openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/httpd/ssl/private/apache-selfsigned.key -
```

Preencha os prompts adequadamente. A linha mais importante é a que solicita o `Common Name`. Você deve inserir o nome de domínio que deseja associar ao seu servidor. Você pode inserir o endereço IP público se não tiver um nome de domínio.

```
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 []:
```

Edite o arquivo de configuração SSL do Apache (`/etc/httpd/conf.d/ssl.conf`):

```
DocumentRoot "/usr/share/zabbix"
ServerName example.com:443
SSLCertificateFile /etc/httpd/ssl/apache-selfsigned.crt
SSLCertificateKeyFile /etc/httpd/ssl/private/apache-selfsigned.key
```

Reinicie o serviço Apache para aplicar as alterações:

```
systemctl restart httpd.service
```

3 Servidor web

Visão geral

Esta seção contém as melhores práticas para configurar o servidor web de forma segura.

Forçando o redirecionamento da URL raiz para o Zabbix SSL

Em sistemas baseados em RHEL, adicione um host virtual à configuração do Apache (`/etc/httpd/conf/httpd.conf`) e defina um redirecionamento permanente da raiz do documento para a URL SSL do Zabbix. Observe que `example.com` deve ser substituído pelo nome real do servidor.

```
#### Adicione as linhas:

<VirtualHost *:*>
    ServerName example.com
    Redirect permanent / https://example.com
</VirtualHost>
```

Reinicie o serviço Apache para aplicar as alterações:

```
systemctl restart httpd.service
```

Habilitando HTTP Strict Transport Security (HSTS) no servidor web

Para proteger o frontend do Zabbix contra ataques de downgrade de protocolo, recomendamos habilitar a política HSTS no servidor web.

Para habilitar a política HSTS para o seu frontend do Zabbix na configuração do Apache, siga estas etapas:

1. Localize o arquivo de configuração do seu host virtual:

- `/etc/httpd/conf/httpd.conf` em sistemas baseados em RHEL
- `/etc/apache2/sites-available/000-default.conf` no Debian/Ubuntu

2. Adicione a seguinte diretiva ao arquivo de configuração do seu host virtual:

```
<VirtualHost *:*>
    Header set Strict-Transport-Security "max-age=31536000"
</VirtualHost>
```

3. Reinicie o serviço Apache para aplicar as alterações:

```
#### Em sistemas baseados em RHEL:
systemctl restart httpd.service

#### No Debian/Ubuntu
systemctl restart apache2.service
```

Aplicando cookies de sessão Secure e SameSite no Zabbix

Ao configurar o Zabbix, é essencial aplicar os atributos Secure e SameSite para cookies de sessão, a fim de aumentar a segurança e evitar ataques de falsificação de solicitação entre sites (CSRF). No entanto, aplicar `SameSite=Strict` pode causar problemas em determinados cenários, como:

- Widgets de URL no dashboard exibindo ”usuário não autenticado” ao incorporar iframes do mesmo domínio.
- Usuários acessando o dashboard via HTTP em vez de HTTPS podem enfrentar problemas de login.
- Impossibilidade de compartilhar URLs para seções específicas do menu do Zabbix ou hosts.

Para mitigar esses problemas, os usuários devem ter uma forma de ajustar a política SameSite.

1. Cookies seguros

Definir o parâmetro `secure` garante que os cookies sejam transmitidos apenas por HTTPS, evitando a exposição em conexões não criptografadas.

Para habilitar cookies seguros no Zabbix, adicione ou modifique a seguinte configuração no servidor web:

Para Apache:

```
Header always edit Set-Cookie ^(.*)$ $1;Secure
```

Para Nginx:

```
proxy_cookie_path / "/; Secure";
```

Certifique-se de que o frontend do Zabbix seja acessado via HTTPS; caso contrário, cookies com o parâmetro Secure não serão enviados.

2. Configurando o atributo SameSite

As configurações do servidor web também podem aplicar o atributo SameSite:

Para Apache:

```
<IfModule mod_headers.c>
    Header onsuccess edit Set-Cookie (.*) "$1; SameSite=Strict"
</IfModule>
```

Para Nginx (versão 1.19.3+):

```
proxy_cookie_flags ~ samesite=Strict; # Substitua ~ por 'zbx_session' para especificidade
```

Habilitando a Content Security Policy (CSP) no servidor web

Para proteger o frontend do Zabbix contra ataques de Cross Site Scripting (XSS), injeção de dados e ataques semelhantes, recomendamos habilitar a Content Security Policy no servidor web. Para isso, configure o servidor web para retornar o cabeçalho HTTP.

Attention:
A configuração do cabeçalho CSP a seguir é apenas para a instalação padrão do frontend do Zabbix e para casos em que todo o conteúdo se origina do domínio do site (excluindo subdomínios). Uma configuração diferente do cabeçalho CSP pode ser necessária se você estiver, por exemplo, configurando o widget *URL* para exibir conteúdo dos subdomínios do site ou de domínios externos, trocando o *OpenStreetMap* por outro mecanismo de mapa ou adicionando CSS ou widgets externos. Se você estiver usando o método de autenticação multifator Duo Universal Prompt multi-factor authentication, certifique-se de adicionar "duo.com" à diretiva CSP no arquivo de configuração do seu virtual host.

Para habilitar o CSP para o seu frontend do Zabbix na configuração do Apache, siga estas etapas:

1. Localize o arquivo de configuração do seu virtual host:

- `/etc/httpd/conf/httpd.conf` em sistemas baseados em RHEL
- `/etc/apache2/sites-available/000-default.conf` no Debian/Ubuntu

2. Adicione a seguinte diretiva ao arquivo de configuração do seu virtual host:

```
<VirtualHost *:*>
    Header set Content-Security-Policy: "default-src 'self' *.openstreetmap.org; script-src 'self' 'unsafe
</VirtualHost>
```

3. Reinicie o serviço Apache para aplicar as alterações:

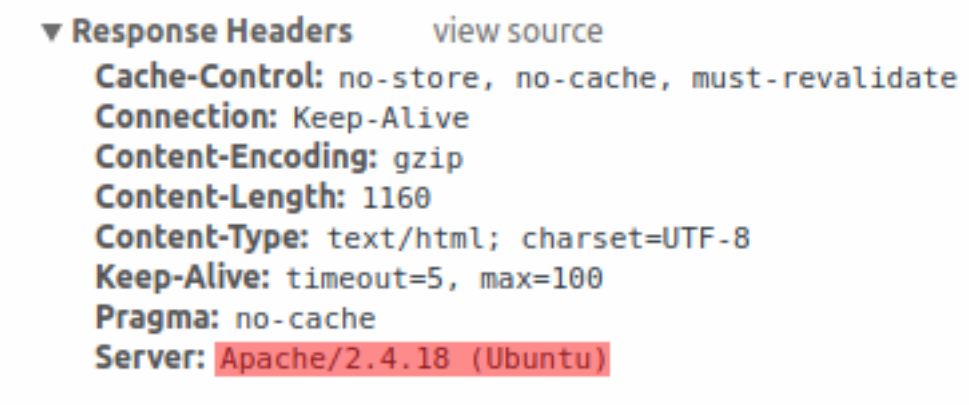
```
#### Em sistemas baseados em RHEL:
systemctl restart httpd.service

#### No Debian/Ubuntu
systemctl restart apache2.service
```

Desabilitando a exposição de informações do servidor web

Para melhorar a segurança, é recomendável desabilitar todas as assinaturas do servidor web.

Por padrão, o servidor web está expondo a assinatura do software:



A assinatura pode ser desabilitada adicionando os seguintes parâmetros ao arquivo de configuração do Apache:

```
ServerSignature Off
ServerTokens Prod
```

A assinatura do PHP (cabeçalho HTTP X-Powered-By) pode ser desabilitada alterando o arquivo de configuração `php.ini` (por padrão, a assinatura está desabilitada):

```
expose_php = Off
```

É necessário reiniciar o servidor web para que as alterações no arquivo de configuração sejam aplicadas.

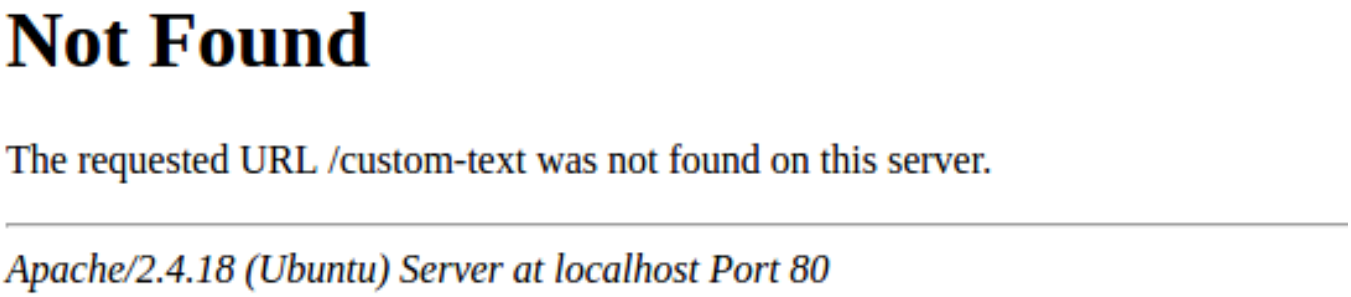
Para segurança adicional, você pode usar a ferramenta *mod_security* com o Apache (pacote *libapache2-mod-security2*). Esta ferramenta permite remover a assinatura do servidor em vez de remover apenas a versão da assinatura do servidor. A assinatura do servidor pode ser alterada para qualquer valor definindo "SecServerSignature" para qualquer valor desejado após instalar o *mod_security*.

Consulte a documentação do seu servidor web para obter ajuda sobre como remover/alterar assinaturas de software.

Desabilitando as páginas de erro padrão do servidor web

Para evitar a exposição de informações, recomenda-se desabilitar as páginas de erro padrão.

Por padrão, um servidor web usa páginas de erro internas:



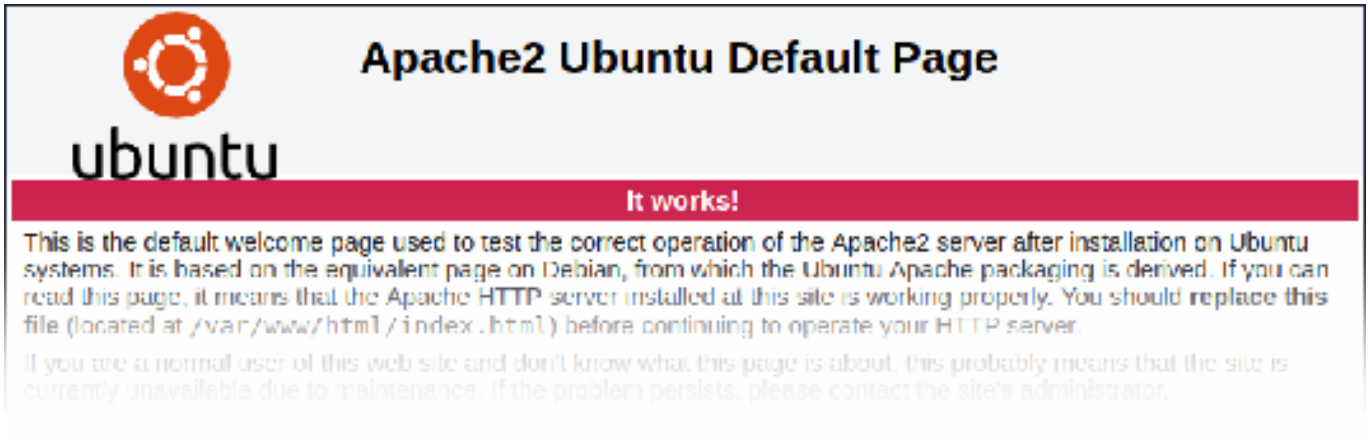
Essas páginas de erro padrão devem ser substituídas/removidas. Por exemplo, a diretiva "ErrorDocument" pode ser usada para definir uma página/texto de erro personalizado para o servidor web Apache.

Consulte a documentação do seu servidor web para obter ajuda sobre como substituir/remover as páginas de erro padrão.

Removendo a página de teste do servidor web

Para evitar a exposição de informações, é recomendada a remoção da página de teste do servidor web.

Por padrão, o diretório webroot do servidor web Apache contém a página de teste `index.html`:



Consulte a documentação do seu servidor web para obter ajuda sobre como remover as páginas de teste padrão.

Definir o cabeçalho de resposta HTTP X-Frame-Options

Por padrão, o Zabbix é configurado com o parâmetro *Usar o cabeçalho HTTP X-Frame-Options* definido como `SAMEORIGIN`. Isso significa que o conteúdo só pode ser carregado em um frame que tenha a mesma origem da própria página.

Os elementos do frontend do Zabbix que puxam conteúdo de URLs externas (nomeadamente, o widget de dashboard de URL) exibem o conteúdo recuperado em uma sandbox com todas as restrições de sandboxing ativadas.

Essas configurações aumentam a segurança do frontend do Zabbix e fornecem proteção contra ataques de XSS e clickjacking. Usuários *Super admin* podem modificar os parâmetros *Usar sandboxing de iframe* e *Usar o cabeçalho HTTP X-Frame-Options* conforme necessário. Por favor, avalie cuidadosamente os riscos e benefícios antes de alterar as configurações padrão. Não é recomendado desativar completamente o sandboxing de iframe ou o cabeçalho HTTP X-Frame-Options.

Ocultando o arquivo com a lista de senhas comuns

Para aumentar a complexidade dos ataques de força bruta de senha, recomenda-se limitar o acesso ao arquivo `ui/data/top_passwords.tx`
Este arquivo contém uma lista das senhas mais comuns e específicas de contexto e impede que os usuários definam tais senhas (se o parâmetro *Evitar senhas fáceis de adivinhar* estiver ativado na política de senha).

Para limitar o acesso ao arquivo `top_passwords.txt`, modifique a configuração do seu servidor web.

No Apache, o acesso ao arquivo pode ser limitado usando o arquivo `.htaccess`:

```
<Files "top_passwords.txt">
    Order Allow,Deny
    Deny from all
</Files>
```

No NGINX, o acesso ao arquivo pode ser limitado usando a diretiva `location`:

```
location = /data/top_passwords.txt {
    deny all;
    return 404;
}
```

2 Melhores práticas de configuração

Visão geral Esta seção descreve as melhores práticas para configurar o Zabbix a fim de obter o melhor desempenho e facilidade de uso. As recomendações são baseadas nos conselhos dos desenvolvedores do Zabbix e na experiência prática de instrutores e engenheiros de suporte do Zabbix.

Cada instalação do Zabbix é única e algumas dessas diretrizes podem não ser adequadas para sua configuração específica. No entanto, recomenda-se tentar seguir essas diretrizes tanto quanto possível para evitar problemas potenciais comuns.

> **Note:**
> Se você acredita que esta página pode ser melhorada, adoraríamos ouvir sua opinião! Por favor, destaque o texto em questão e pressione **ctrl+Enter** para relatar um erro ou compartilhar seu feedback.

Hosts e items Definindo um host

Um host no Zabbix não é uma máquina ou dispositivo físico, mas uma entidade lógica. Para fins de monitoramento, você pode criar hosts separados para um banco de dados ou, por exemplo, uma máquina virtual. Alternativamente, você pode criar um host genérico *Notebook do João* e monitorar todas as métricas sob esse host.

A melhor prática é criar um host separado para cada instância independente, como uma máquina virtual, um banco de dados, um container ou um switch de rede. Ao utilizar essa abordagem, você irá:

1. Evitar confusão nos dados de monitoramento, tendo itens, triggers e notificações de alerta separados para cada host.
2. Ajustar os níveis de acesso dos usuários. Você pode configurar funções de usuário para conceder acesso à visualização e/ou configuração de apenas hosts específicos. Veja também o princípio do menor privilégio.

Hosts com itens duplicados

Se você tiver vários hosts semelhantes, como *Network switch 1* e *Network switch 2*, o Zabbix oferece várias maneiras de recriar rapidamente o host. Você pode simplesmente clonar um host com todas as suas métricas pressionando o botão Clonar, mas, nesse caso, para atualizar um item posteriormente, será necessário fazê-lo manualmente em cada host.

A melhor prática é criar um template com todas as métricas necessárias, por exemplo, *Network switch template*. Em seguida, agrupe os hosts semelhantes em um grupo de hosts, para o exemplo acima, poderia ser *Network switches*. Agora, na seção *Coleta de dados -> Hosts*, você pode filtrar todos os hosts por grupo de hosts e usar o botão *Atualização em massa* para vincular o template a todos os seus switches de rede.

Items dependentes

Para minimizar o número de solicitações à entidade de destino, o Zabbix permite a criação de items mestre e dependentes. Neste caso, o item mestre coleta um grande conjunto de informações em uma única solicitação. Em seguida, os items dependentes podem ser configurados para extrair partes específicas desses dados por meio de pré-processamento e armazená-las como métricas individuais.

Por exemplo, o item mestre pode coletar uma resposta JSON ou XML contendo várias métricas ou executar uma consulta de banco de dados que retorna várias colunas de dados (por exemplo, número de conexões abertas, conexões abortadas, número máximo de conexões simultâneas permitidas e total acumulado de conexões desde a inicialização), e os items dependentes irão analisar e armazenar cada valor necessário separadamente.

A melhor prática para essa configuração é descartar o histórico do item mestre logo após a coleta e manter apenas os dados dos items dependentes.

Servidores e proxies Se todos os hosts estiverem na mesma rede local que o Zabbix server e não houver preocupações com escalabilidade ou desempenho, talvez você não precise de um proxy. Em ambientes maiores ou mais complexos, monitorar hosts diretamente com o Zabbix server pode não ser suficiente. Adicionar um proxy e atribuir parte dos hosts a esse proxy permite uma distribuição de carga mais equilibrada.

A melhor prática é adicionar um Zabbix proxy quando:

1. Você está monitorando vários hosts usando vários métodos de coleta de métricas atrás de um firewall. O proxy coletará dados dos hosts e os encaminhará para o Zabbix server, reduzindo a necessidade de abrir várias portas no firewall.
2. Você está monitorando locais remotos, filiais e/ou redes. Em caso de uma interrupção de rede entre o Zabbix server e seus locais remotos, os Zabbix proxies implantados nos locais remotos continuarão a coleta de dados e enviarão os dados coletados de volta ao Zabbix server sempre que a conexão de rede for restabelecida.
3. Você possui uma implantação em larga escala e deseja reduzir a carga no Zabbix server e melhorar o desempenho. A definição de uma implantação em larga escala é muito ampla e depende não apenas do número de hosts, mas também do número de valores coletados por segundo.

20 API

Visão geral A API do Zabbix permite recuperar e modificar programaticamente a configuração do Zabbix e fornece acesso aos dados históricos. Ela é amplamente utilizada para:

- criar novos aplicativos para trabalhar com o Zabbix;
- integrar o Zabbix a um software de terceiros;
- automatizar tarefas rotineiras.

A API do Zabbix é baseada em HTTP e é fornecida como parte do frontend web. Ela utiliza o protocolo JSON-RPC 2.0, o que significa duas coisas:

- a API consiste em um conjunto de métodos separados;
- as solicitações e respostas entre os clientes e a API são codificadas usando o formato JSON.

Para mais informações sobre o protocolo e JSON, consulte a especificação JSON-RPC 2.0 e a página inicial do formato JSON.

Para mais informações sobre como integrar a funcionalidade do Zabbix em seus aplicativos Python, consulte Biblioteca Python para Zabbix.

> **Note:**
> O acesso do usuário no Zabbix, incluindo configuração e dados históricos, depende do tipo de usuário, da função de usuário atribuída e dos grupos de usuários.

Estrutura A API consiste em vários métodos que são agrupados nominalmente em APIs separadas. Cada um dos métodos executa uma tarefa específica. Por exemplo, o método `host.create` pertence à API *host* e é usado para criar novos hosts. Historicamente, as APIs às vezes são chamadas de "classes".

> **Note:**
> A maioria das APIs contém pelo menos quatro métodos: `get`, `create`, `update` e `delete` para recuperar, criar, atualizar e excluir dados, respectivamente, mas algumas APIs podem fornecer um conjunto totalmente diferente de métodos.

Realizando solicitações Depois de configurar o frontend, você pode usar solicitações HTTP remotas para chamar a API. Para isso, é necessário enviar solicitações HTTP POST para o arquivo `api_jsonrpc.php` localizado no diretório do frontend. Por exemplo, se o seu frontend do Zabbix estiver instalado em `https://example.com/zabbix`, uma solicitação HTTP para chamar o método `apiinfo.version` pode ser assim:

```
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}'
```

O cabeçalho da solicitação deve ser definido como `Content-Type` com um dos seguintes valores: `application/json-rpc`, `application/json` ou `application/jsonrequest`.

O objeto de solicitação deve conter as seguintes propriedades:

- `jsonrpc` - a versão do protocolo JSON-RPC usada pela API (a API do Zabbix implementa a versão 2.0 do JSON-RPC);
- `method` - o método da API que está sendo chamado;
- `params` - os parâmetros que serão passados para o método da API;
- `id` - um identificador arbitrário da solicitação (se omitido, a API trata a solicitação como uma notificação).

Se a solicitação estiver correta, a resposta retornada pela API deve ser semelhante a esta:

```
{
    "jsonrpc": "2.0",
    "result": "7.0.0",
    "id": 1
}
```

O objeto de resposta, por sua vez, contém as seguintes propriedades:

- `jsonrpc` - a versão do protocolo JSON-RPC;
- `result` - os dados retornados pelo método;
- `id` - um identificador da solicitação correspondente.

Exemplo de fluxo de trabalho A seção a seguir irá guiá-lo por alguns exemplos de uso em maior detalhe.

Autenticação Para acessar qualquer dado no Zabbix, você precisa:

- usar um token de API existente (criado no frontend do Zabbix ou usando a API de Token);
- usar um token de autenticação obtido com o método user.login.

Por exemplo, se você quiser obter um novo token de autenticação fazendo login como o usuário padrão *Admin*, a solicitação JSON seria assim:

```
curl --request POST \
  --url 'https://example.com/zabbix/api_jsonrpc.php' \
  --header 'Content-Type: application/json-rpc' \
  --data '{"jsonrpc":"2.0","method":"user.login","params":{"username":"Admin","password":"zabbix"},"id":1}
```

Se você forneceu as credenciais corretamente, a resposta retornada pela API deve conter o token de autenticação do usuário:

```
{
    "jsonrpc": "2.0",
    "result": "0424bd59b807674191e7d77572075f33",
    "id": 1
}
```

Métodos de autorização Por cabeçalho de autorização

Todas as solicitações da API exigem autenticação ou um token de API. Você pode fornecer as credenciais usando o cabeçalho Authorization na solicitação:

```
curl --request POST \
  --url 'https://example.com/zabbix/api_jsonrpc.php' \
  --header 'Authorization: Bearer 0424bd59b807674191e7d77572075f33'
```

Attention:
Se você estiver enfrentando problemas de autenticação, consulte Encaminhamento do cabeçalho de autorização.

Desde o Zabbix 7.0.7, o cabeçalho Authorization é suportado em solicitações de origem cruzada (CORS). Desde o Zabbix 7.0.11, a API do Zabbix aceita cabeçalhos de forma case-insensitive (por exemplo, `authorization`, `Authorization` e `AUTHORIZATION` são tratados da mesma forma).

Pela propriedade `auth`

Uma solicitação de API pode ser autorizada pela propriedade `auth`.

Attention:
Observe que a propriedade `auth` está obsoleta. Ela será removida em versões futuras.

```
curl --request POST \
  --url 'https://example.com/zabbix/api_jsonrpc.php' \
  --header 'Content-Type: application/json-rpc' \
  --data '{"jsonrpc":"2.0","method":"host.get","params":{"output":["hostid"]},"auth":"0424bd59b807674191e7
```

Por cookie do Zabbix

Um cookie *"zbx_session"* é usado para autorizar uma solicitação de API da interface do Zabbix realizada usando JavaScript (de um módulo ou de um widget personalizado).

Recuperando hosts Agora você tem um token de autenticação de usuário válido (representado como uma variável nos exemplos a seguir) que pode ser usado para acessar os dados no Zabbix. Por exemplo, você pode usar o método host.get para recuperar os IDs, nomes de host e interfaces de todos os hosts configurados:

Requisição:

```
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` é um arquivo que contém uma consulta JSON. Em vez de um arquivo, você pode passar a consulta no argumento `--data`.

data.json

```
{
    "jsonrpc": "2.0",
```

```
    "method": "host.get",
    "params": {
        "output": [
            "hostid",
            "host"
        ],
        "selectInterfaces": [
            "interfaceid",
            "ip"
        ]
    },
    "id": 2
}
```

O objeto de resposta conterá os dados solicitados sobre os hosts:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "hostid": "10084",
            "host": "Zabbix server",
            "interfaces": [
                {
                    "interfaceid": "1",
                    "ip": "127.0.0.1"
                }
            ]
        }
    ],
    "id": 2
}
```

Note:
Por questões de desempenho, é sempre recomendável listar as propriedades do objeto que você deseja recuperar. Assim, você evitará recuperar tudo.

Criando um novo item Agora, crie um novo item no host "Zabbix server" usando os dados obtidos na solicitação `host.get` anterior. Isso pode ser feito usando o método item.create:

```
curl --request POST \
  --url 'https://example.com/zabbix/api_jsonrpc.php' \
  --header 'Authorization: Bearer ${AUTHORIZATION_TOKEN}' \
  --header 'Content-Type: application/json-rpc' \
  --data '{"jsonrpc":"2.0","method":"item.create","params":{"name":"Free disk space on /home/joe/","key_":
```

Uma resposta bem-sucedida conterá o ID do item recém-criado, que pode ser usado para referenciar o item nas solicitações seguintes:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "24759"
        ]
    },
    "id": 3
}
```

Note:
O método `item.create`, assim como outros *métodos de criação*, também pode aceitar arrays de objetos e criar múltiplos itens com uma única chamada à API.

Criando múltiplos triggers Assim, se os *métodos de criação* aceitarem arrays, você pode adicionar múltiplos triggers, por exemplo, este:

```
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
```

A resposta bem-sucedida conterá os IDs dos triggers recém-criados:

```
{
    "jsonrpc": "2.0",
    "result": {
        "triggerids": [
            "17369",
            "17370"
        ]
    },
    "id": 4
}
```

Atualizando um item Habilite um item definindo seu status como "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}'
```

A resposta bem-sucedida conterá o ID do item atualizado:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "10092"
        ]
    },
    "id": 5
}
```

Note:
O método `item.update`, assim como outros *métodos de atualização*, também pode aceitar arrays de objetos e atualizar vários itens com uma única chamada de API.

Atualizando múltiplos triggers Habilite múltiplos triggers definindo seu status como "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
```

A resposta bem-sucedida conterá os IDs dos triggers atualizados:

```
{
    "jsonrpc": "2.0",
    "result": {
        "triggerids": [
            "13938",
            "13939"
        ]
    },
    "id": 6
}
```

Note:
Este é o método preferido de atualização. Alguns métodos da API, como o `host.massupdate`, permitem escrever um código mais simples. No entanto, não é recomendado usar esses métodos, pois eles serão removidos em versões futuras.

Tratamento de erros Até o momento, tudo o que você tentou funcionou bem. Mas o que aconteceria se você tentasse fazer uma chamada incorreta para a API? Tente criar outro host chamando host.create, mas omitindo o parâmetro obrigatório `groups`:

```
curl --request POST \
  --url 'https://example.com/zabbix/api_jsonrpc.php' \
  --header 'Authorization: Bearer ${AUTHORIZATION_TOKEN}' \
  --header 'Content-Type: application/json-rpc' \
  --data '{"jsonrpc":"2.0","method":"host.create","params":{"host":"Linux server","interfaces":[{"type":1,
```

A resposta então conterá uma mensagem de erro:

```
{
    "jsonrpc": "2.0",
    "error": {
        "code": -32602,
        "message": "Invalid params.",
        "data": "No groups for host \"Linux server\"."
    },
    "id": 7
}
```

Se um erro ocorreu, em vez da propriedade `result`, o objeto de resposta conterá a propriedade `error` com os seguintes dados:

- `code` - um código de erro;
- `message` - um resumo curto do erro;
- `data` - uma mensagem de erro mais detalhada.

Erros podem ocorrer em vários casos, como uso de valores de entrada incorretos, expiração de sessão ou tentativa de acessar objetos inexistentes. Sua aplicação deve ser capaz de lidar adequadamente com esses tipos de erros.

Versões da API Para simplificar o versionamento da API, desde o Zabbix 2.0.4, a versão da API corresponde à versão do próprio Zabbix. Você pode usar o método apiinfo.version para descobrir a versão da API com a qual está trabalhando. Isso pode ser útil para ajustar seu aplicativo para usar recursos específicos de cada versão.

O Zabbix garante a compatibilidade de recursos dentro de uma versão principal. Ao fazer alterações incompatíveis entre versões principais, o Zabbix geralmente deixa os recursos antigos como obsoletos na próxima versão e só os remove na versão seguinte. Ocasionalmente, o Zabbix pode remover recursos entre versões principais sem fornecer nenhuma compatibilidade retroativa. É importante que você nunca dependa de recursos obsoletos e migre para alternativas mais recentes o mais rápido possível.

Note:
Você pode acompanhar todas as alterações feitas na API no API changelog.

Leitura adicional Agora, você tem conhecimento suficiente para começar a trabalhar com a API do Zabbix, no entanto, não pare por aqui. Para uma leitura adicional, recomenda-se dar uma olhada na lista de APIs disponíveis.

referência do método

Esta seção fornece uma visão geral das funções fornecidas pelo API Zabbix e irá ajudá-lo a encontrar o caminho para as classes disponíveis e métodos.

Monitoramento A API do Zabbix permite acessar o histórico e outros dados coletados durante o monitoramento.

Dashboards

Gerencie dashboards e crie relatórios agendados com base neles.

API de dashboard | API de dashboard de template | API de relatório

Cluster de alta disponibilidade

Recupere uma lista de nós do servidor e seus status.

API de cluster de alta disponibilidade

História

Recupere valores históricos coletados pelos processos de monitoramento do Zabbix para apresentação ou processamento posterior.

API de histórico

Trends

Recupere valores de trends calculados pelo servidor Zabbix para apresentação ou processamento adicional.

API de tendências

Eventos

Recupere eventos gerados por triggers, descoberta de rede e outros Sistemas Zabbix para gerenciamento de situação mais flexível ou de terceiros integração de ferramentas.

API de eventos

Problemas

Recupere problemas de acordo com os parâmetros fornecidos.

API de problema

Mapas

Configure mapas para criar representações dinâmicas detalhadas de sua infraestrutura de TI.

Mapa API

Tarefas

Interaja com o gerenciador de tarefas do servidor Zabbix, criando tarefas e recuperando resposta.

API de tarefas

Serviços A API do Zabbix permite acessar os dados coletados durante o monitoramento de serviços.

Acordo de Nível de Serviço

Definir objetivos de nível de serviço (SLO), recuperar indicadores de nível de serviço (SLI) detalhados informações sobre o desempenho do serviço.

API SLA

Serviços

Gerencie serviços para monitoramento de nível de serviço e recupere informações de SLA detalhadas sobre qualquer serviço.

Serviço API

Coleta de dados A API do Zabbix permite que você gerencie a configuração do seu sistema de monitoramento.

Hosts e grupos de hosts

Gerencie grupos de hosts, hosts e tudo relacionado a eles, incluindo interfaces de host, macros de host e períodos de manutenção.

API de Host | API de Grupo de Host | API de Interface de Host | API de Macro de Usuário | API de Mapa de Valores | API de Manutenção

Itens

Defina os itens a serem monitorados.

API do item

Gatilhos

Configure gatilhos para notificá-lo sobre problemas em seu sistema. Gerencie as dependências de gatilho.

API Trigger

Gráficos

Edite gráficos ou itens de gráfico separados para melhor apresentação dos dados coletados.

API de gráfico | Item do gráfico API

Templates e grupos de templates

Gerencie templates e vincule-os a hosts ou outros templates.

API de Template | API de Grupo de Template | API de Mapa de Valores

Descoberta de baixo nível (Low-level discovery)

Configure regras de descoberta de baixo nível, bem como item, gatilho e protótipos de gráfico para monitorar entidades dinâmicas.

regra LLD API | Protótipo de item API | Protótipo de gatilho API | Protótipo de gráfico API | Protótipo do host API

Correlação de eventos

Crie regras de correlação de eventos personalizados.

Correlação API

Descoberta de rede

Gerencie regras de descoberta em nível de rede para localizar e monitorar automaticamente novos hosts. Obtenha acesso total a informações sobre serviços descobertos e hosts.

Regra de descoberta API | Verificação de descoberta API | Host descoberto API | Serviço descoberto API

Exportar e importar

Exporte e importe dados de configuração do Zabbix para backups de configuração, migração ou atualizações de configuração em grande escala.

configuração API

Monitoramento da Web

Configure cenários da web para monitorar seus aplicativos e serviços da web.

Cenário da Web API

Alertas A API do Zabbix permite que você gerencie as ações e alertas do seu sistema de monitoramento.

Ações e alertas

Definir ações e operações para notificar os usuários sobre determinados eventos ou executar comandos remotos automaticamente. Tenha acesso a informações sobre alertas gerados e seus receptores.

Ação API | Alerta API

Tipos de mídia

Configure tipos de mídia e várias maneiras pelas quais os usuários receberão alertas.

API de tipo de mídia

Scripts

Configure e execute scripts para ajudá-lo em suas tarefas diárias.

API de Script

Usuários A API do Zabbix permite gerenciar os usuários do seu sistema de monitoramento.

Usuários e grupos de usuários

Adicione usuários que terão acesso ao Zabbix, atribua-os a grupos de usuários e conceda permissões. Crie funções para gerenciamento granular dos direitos dos usuários.

API de usuário | API de grupo de usuários | API de diretório de usuários | API de função de usuário

Tokens de API

Gerencie tokens de autorização.

Token API

Autenticação

Altere as opções de configuração de autenticação.

API de autenticação

Administração Com a API do Zabbix, você pode alterar as configurações de administração do seu sistema de monitoramento.

Geral

Altere certas opções globais de configuração.

API de Autoregistro | API de Mapa de Ícones | API de Imagem | API de Configurações | API de Expressão Regular | API de Módulo | API de Conector

Log de auditoria

Acompanhe as alterações de configuração feitas por cada usuário.

API de log de auditoria

Housekeeping

Configure o housekeeping.

Housekeeping API

Proxies e grupos de proxies

Gerencie os proxies usados em sua configuração de monitoramento distribuído.

API de proxy | API de grupo de proxy

Macros

Gerenciar macros.

API de macro de usuário

Informações da API Recupere a versão da API do Zabbix para que seu aplicativo possa usar recursos específicos da versão.

API info API

Action

Esta classe foi projetada para trabalhar com ações.

Referências de objetos:

- Action
- Action operation
  - Action operation message
  - Action operation condition
- Action recovery operation
- Action update operation
- Action filter
  - Action filter condition

Métodos disponíveis:

- action.create - criar novas ações
- action.delete - excluir ações
- action.get - recuperar ações
- action.update - atualizar ações

> Objeto ação

Os objetos a seguir estão diretamente relacionados à API ação.

Ação

O objeto action possui as seguintes propriedades.

Propriedade	Tipo	Descrição
actionid	ID	ID da ação. Comportamento da propriedade: - *somente leitura* - *obrigatório* para operações de atualização
esc_period	string	Duração padrão do passo da operação. Deve ser de pelo menos 60 segundos. Aceita segundos, unidade de tempo com sufixo ou uma macro de usuário. Comportamento da propriedade: - *suportado* se `eventsource` estiver definido como ”evento criado por um trigger”, ”evento interno” ou ”evento criado na atualização do status do serviço”
eventsource	integer	Tipo de eventos que a ação irá manipular. Consulte a propriedade `source` do evento para uma lista de tipos de eventos suportados. Comportamento da propriedade: - *constante* - *obrigatório* para operações de criação
name	string	Nome da ação. Comportamento da propriedade: - *obrigatório* para operações de criação
status	integer	Se a ação está habilitada ou desabilitada. Valores possíveis: 0 - *(padrão)* habilitada; 1 - desabilitada.
pause_symptoms	integer	Se deve pausar a escalonamento se o evento for um evento de sintoma. Valores possíveis: 0 - Não pausar escalonamento para problemas de sintoma; 1 - *(padrão)* Pausar escalonamento para problemas de sintoma. Comportamento da propriedade: - *suportado* se `eventsource` estiver definido como ”evento criado por um trigger”
pause_suppressed	integer	Se deve pausar o escalonamento durante períodos de manutenção ou não. Valores possíveis: 0 - Não pausar escalonamento; 1 - *(padrão)* Pausar escalonamento. Comportamento da propriedade: - *suportado* se `eventsource` estiver definido como ”evento criado por um trigger”
notify_if_canceled	integer	Se deve notificar quando o escalonamento for cancelado. Valores possíveis: 0 - Não notificar quando o escalonamento for cancelado; 1 - *(padrão)* Notificar quando o escalonamento for cancelado. Comportamento da propriedade: - *suportado* se `eventsource` estiver definido como ”evento criado por um trigger”

Operação de ação

O objeto de operação de ação define uma operação que será executada quando uma ação for executada. Possui as seguintes propriedades.

Propriedade	Tipo	Descrição
operationtype	integer	Tipo de operação. Valores possíveis: 0 - enviar mensagem; 1 - script global; 2 - adicionar host; 3 - remover host; 4 - adicionar ao grupo de hosts; 5 - remover do grupo de hosts; 6 - vincular template; 7 - desvincular template; 8 - habilitar host; 9 - desabilitar host; 10 - definir modo de inventário do host; 13 - adicionar tags de host; 14 - remover tags de host. Valores possíveis se `eventsource` do objeto Action estiver definido como ”evento criado por um trigger” ou ”evento criado na atualização do status do serviço”: 0 - ”enviar mensagem”; 1 - ”script global”. Valores possíveis se `eventsource` do objeto Action estiver definido como ”evento interno”: 0 - ”enviar mensagem”. Comportamento da propriedade: - *obrigatório*
esc_period	string	Duração de um passo de escalonamento em segundos. Deve ser maior que 60 segundos. Aceita segundos, unidade de tempo com sufixo ou uma macro de usuário. Se definido como 0 ou 0s, o período de escalonamento padrão da ação será usado. Padrão: 0s. Comportamento da propriedade: - *suportado* se `eventsource` do objeto Action estiver definido como ”evento criado por um trigger”, ”evento interno” ou ”evento criado na atualização do status do serviço”
esc_step_from	integer	Passo para iniciar o escalonamento. Padrão: 1. Comportamento da propriedade: - *suportado* se `eventsource` do objeto Action estiver definido como ”evento criado por um trigger”, ”evento interno” ou ”evento criado na atualização do status do serviço”
esc_step_to	integer	Passo para finalizar o escalonamento. Padrão: 1. Comportamento da propriedade: - *suportado* se `eventsource` do objeto Action estiver definido como ”evento criado por um trigger”, ”evento interno” ou ”evento criado na atualização do status do serviço”

Propriedade	Tipo	Descrição
evaltype	integer	Método de avaliação de condição da operação. Valores possíveis: 0 - *(padrão)* E/Ou; 1 - E; 2 - Ou.
opcommand	object	Script global a ser executado. O script global deve ter a propriedade `scriptid` definida. Comportamento da propriedade: - *obrigatório* se `operationtype` estiver definido como ”script global”
opcommand_grp	array	Grupos de hosts para executar scripts globais. Os grupos de hosts devem ter a propriedade `groupid` definida. Comportamento da propriedade: - *obrigatório* se `operationtype` estiver definido como ”script global” e `opcommand_hst` não estiver definido
opcommand_hst	array	Host para executar scripts globais. Os hosts devem ter a propriedade `hostid` definida. Comportamento da propriedade: - *obrigatório* se `operationtype` estiver definido como ”script global” e `opcommand_grp` não estiver definido
opconditions	array	Condições de operação usadas para ações de trigger. O objeto de condição de operação é descrito em detalhes abaixo.
opgroup	array	Grupos de hosts para adicionar hosts. Os grupos de hosts devem ter a propriedade `groupid` definida. Comportamento da propriedade: - *obrigatório* se `operationtype` estiver definido como ”adicionar ao grupo de hosts” ou ”remover do grupo de hosts”
opmessage	object	Objeto contendo os dados sobre a mensagem enviada pela operação. O objeto de mensagem de operação é descrito em detalhes abaixo. Comportamento da propriedade: - *obrigatório* se `operationtype` estiver definido como ”enviar mensagem”
opmessage_grp	array	Grupos de usuários para enviar mensagens. Os grupos de usuários devem ter a propriedade `usrgrpid` definida. Comportamento da propriedade: - *obrigatório* se `operationtype` estiver definido como ”enviar mensagem” e `opmessage_usr` não estiver definido
opmessage_usr	array	Usuários para enviar mensagens. Os usuários devem ter a propriedade `userid` definida. Comportamento da propriedade: - *obrigatório* se `operationtype` estiver definido como ”enviar mensagem” e `opmessage_grp` não estiver definido

Propriedade	Tipo	Descrição
optemplate	array	Templates para vincular aos hosts. Os templates devem ter a propriedade `templateid` definida. Comportamento da propriedade: - *obrigatório* se `operationtype` estiver definido como ”vincular template” ou ”desvincular template”
opinventory	object	Modo de inventário para definir no host. O inventário deve ter a propriedade `inventory_mode` definida. Comportamento da propriedade: - *obrigatório* se `operationtype` estiver definido como ”definir modo de inventário do host”
optag	array	Tags de host para adicionar ou remover. As tags devem ter a propriedade `tag` definida. A propriedade `value` é opcional. Comportamento da propriedade: - *suportado* se `operationtype` estiver definido como ”adicionar tags de host” ou ”remover tags de host”.

Mensagem de operação de ação

O objeto de mensagem de operação contém dados sobre a mensagem que será enviada pela operação. Ele possui as seguintes propriedades.

Propriedade	Tipo	Descrição
default_msg	integer	Se deve usar o texto e o assunto padrão da mensagem de ação. Valores possíveis: 0 - usar os dados da operação; 1 - *(padrão)* usar os dados do tipo de mídia.
mediatypeid	ID	ID do tipo de mídia que será usado para enviar a mensagem. Comportamento da propriedade: - *suportado* se `operationtype` do Objeto de operação de ação, Objeto de operação de recuperação de ação ou Objeto de operação de atualização de ação estiver definido como ”enviar mensagem”, ou se `operationtype` do Objeto de operação de atualização de ação estiver definido como ”notificar todos os envolvidos”
message	string	Texto da mensagem da operação. Comportamento da propriedade: - *suportado* se `default_msg` estiver definido como ”usar os dados da operação”
subject	string	Assunto da mensagem da operação. Comportamento da propriedade: - *suportado* se `default_msg` estiver definido como ”usar os dados da operação”

Condição de operação de ação

O objeto de condição de operação de ação define uma condição que deve ser atendida para executar a operação atual. Possui as seguintes propriedades.

Propriedade	Tipo	Descrição
conditiontype	integer	Tipo de condição. Valores possíveis: 14 - evento reconhecido. Comportamento da propriedade: - *obrigatório*
value	string	Valor para comparar. Comportamento da propriedade: - *obrigatório*
operator	integer	Operador da condição. Valores possíveis: 0 - *(padrão)* =

Os seguintes operadores e valores são suportados para cada tipo de condição de operação.

Condição	Nome da condição	Operadores suportados	Valor esperado
14	Evento reconhecido	=	Se o evento foi reconhecido. Valores possíveis: 0 - não reconhecido; 1 - reconhecido.

Operação de recuperação de ação

O objeto de operação de recuperação de ação define uma operação que será executada quando um problema for resolvido. As operações de recuperação são possíveis **apenas** para ações de trigger, internas e de serviço. Possui as seguintes propriedades.

Propriedade	Tipo	Descrição
operationtype	integer	Tipo de operação. Valores possíveis se o `eventsource` do objeto Action estiver definido como ”evento criado por um trigger” ou ”evento criado na atualização do status do serviço”: 0 - enviar mensagem; 1 - script global; 11 - notificar todos os envolvidos. Valores possíveis se o `eventsource` do objeto Action estiver definido como ”evento interno”: 0 - enviar mensagem; 11 - notificar todos os envolvidos. Comportamento da propriedade: - *obrigatório*
opcommand	object	Script global a ser executado. O script global deve ter a propriedade `scriptid` definida. Comportamento da propriedade: - *obrigatório* se `operationtype` estiver definido como ”script global”

Propriedade	Tipo	Descrição
opcommand_grp	array	Grupos de hosts para executar scripts globais. Os grupos de hosts devem ter a propriedade `groupid` definida. Comportamento da propriedade: - *obrigatório* se o `eventsource` do objeto Action estiver definido como "evento criado por um trigger", e `operationtype` estiver definido como "script global", e `opcommand_hst` não estiver definido
opcommand_hst	array	Host para executar scripts globais. Os hosts devem ter a propriedade `hostid` definida. Comportamento da propriedade: - *obrigatório* se o `eventsource` do objeto Action estiver definido como "evento criado por um trigger", e `operationtype` estiver definido como "script global", e `opcommand_grp` não estiver definido
opmessage	object	Objeto contendo os dados sobre a mensagem enviada pela operação de recuperação. O objeto de mensagem de operação é descrito em detalhes acima. Comportamento da propriedade: - *obrigatório* se `operationtype` estiver definido como "enviar mensagem"
opmessage_grp	array	Grupos de usuários para enviar mensagens. Os grupos de usuários devem ter a propriedade `usrgrpid` definida. Comportamento da propriedade: - *obrigatório* se `operationtype` estiver definido como "enviar mensagem" e `opmessage_usr` não estiver definido
opmessage_usr	array	Usuários para enviar mensagens. Os usuários devem ter a propriedade `userid` definida. Comportamento da propriedade: - *obrigatório* se `operationtype` estiver definido como "enviar mensagem" e `opmessage_grp` não estiver definido

Operação de atualização de ação

O objeto de operação de atualização de ação define uma operação que será executada quando um problema for atualizado (comentado, reconhecido, severidade alterada ou fechado manualmente). As operações de atualização são possíveis **apenas** para ações de trigger e serviço. Possui as seguintes propriedades.

Propriedade	Tipo	Descrição
operationtype	integer	Tipo de operação. Valores possíveis: 0 - enviar mensagem; 1 - script global; 12 - notificar todos os envolvidos. Comportamento da propriedade: - *obrigatório*

Propriedade	Tipo	Descrição
opcommand	object	Script global a ser executado. O script global deve ter a propriedade `scriptid` definida. Comportamento da propriedade: - *obrigatório* se `operationtype` estiver definido como ”script global”
opcommand_grp	array	Grupos de hosts para executar scripts globais. Os grupos de hosts devem ter a propriedade `groupid` definida. Comportamento da propriedade: - *obrigatório* se `eventsource` do objeto Action estiver definido como ”evento criado por um trigger”, e `operationtype` estiver definido como ”script global”, e `opcommand_hst` não estiver definido
opcommand_hst	array	Host para executar scripts globais. Os hosts devem ter a propriedade `hostid` definida. Comportamento da propriedade: - *obrigatório* se `eventsource` do objeto Action estiver definido como ”evento criado por um trigger”, e `operationtype` estiver definido como ”script global”, e `opcommand_grp` não estiver definido
opmessage	object	Objeto contendo os dados sobre a mensagem enviada pela operação de atualização. O objeto de mensagem de operação é descrito em detalhes acima.
opmessage_grp	array	Grupos de usuários para enviar mensagens. Os grupos de usuários devem ter a propriedade `usrgrpid` definida. Comportamento da propriedade: - *obrigatório* se `operationtype` estiver definido como ”enviar mensagem” e `opmessage_usr` não estiver definido
opmessage_usr	array	Usuários para enviar mensagens. Os usuários devem ter a propriedade `userid` definida. Comportamento da propriedade: - *obrigatório* se `operationtype` estiver definido como ”enviar mensagem” e `opmessage_grp` não estiver definido

Filtro de ação

O objeto de filtro de ação define um conjunto de condições que devem ser atendidas para executar as operações de ação configuradas. Ele possui as seguintes propriedades.

Propriedade	Tipo	Descrição
conditions	array	Conjunto de condições de filtro a serem usadas para filtrar os resultados. As condições serão classificadas na ordem de sua colocação na fórmula. Comportamento da propriedade: - *obrigatório*

Propriedade	Tipo	Descrição
evaltype	integer	Método de avaliação da condição do filtro. Valores possíveis: 0 - E/OU; 1 - E; 2 - OU; 3 - Expressão personalizada. Comportamento da propriedade: - *obrigatório*
eval_formula	string	Expressão gerada que será usada para avaliar as condições do filtro. A expressão contém IDs que fazem referência a condições de filtro específicas por seu `formulaid`. O valor de `eval_formula` é igual ao valor de `formula` para filtros com uma expressão personalizada. Comportamento da propriedade: - *somente leitura*
formula	string	Expressão definida pelo usuário a ser usada para avaliar as condições de filtros com uma expressão personalizada. A expressão deve conter IDs que fazem referência a condições de filtro específicas por seu `formulaid`. Os IDs usados na expressão devem corresponder exatamente aos definidos nas condições do filtro: nenhuma condição pode permanecer não utilizada ou omitida. Comportamento da propriedade: - *obrigatório* se `evaltype` estiver definido como ”expressão personalizada”

Condição do filtro de ação

O objeto de condição do filtro de ação define uma condição específica que deve ser verificada antes de executar as operações da ação.

Propriedade	Tipo	Descrição
conditiontype	integer	Tipo de condição. Valores possíveis se `eventsource` do objeto Action estiver definido como ”evento criado por um trigger”: 0 - grupo de hosts; 1 - host; 2 - trigger; 3 - nome do evento; 4 - severidade do trigger; 6 - período de tempo; 13 - template de host; 16 - problema está suprimido; 25 - tag do evento; 26 - valor da tag do evento. Valores possíveis se `eventsource` do objeto Action estiver definido como ”evento criado por uma regra de descoberta”: 7 - IP do host; 8 - tipo de serviço descoberto; 9 - porta do serviço descoberto; 10 - status da descoberta; 11 - duração do tempo de atividade ou inatividade; 12 - valor recebido; 18 - regra de descoberta; 19 - verificação de descoberta; 20 - proxy; 21 - objeto de descoberta. Valores possíveis se `eventsource` do objeto Action estiver definido como ”evento criado por autoregistro de agent ativo”: 20 - proxy; 22 - nome do host; 24 - metadados do host. Valores possíveis se `eventsource` do objeto Action estiver definido como ”evento interno”: 0 - grupo de hosts; 1 - host; 13 - template de host; 23 - tipo de evento; 25 - tag do evento; 26 - valor da tag do evento. Valores possíveis se `eventsource` do objeto Action estiver definido como ”evento criado na atualização do status do serviço”: 25 - tag do evento; 26 - valor da tag do evento; 27 - serviço; 28 - nome do serviço. Comportamento da propriedade: - *obrigatório*
value	string	Valor para comparar. Comportamento da propriedade: - *obrigatório*

Propriedade	Tipo	Descrição
value2	string	Valor secundário para comparar. Comportamento da propriedade: - *obrigatório* se `eventsource` do objeto Action estiver definido como "evento criado por um trigger", `conditiontype` estiver definido como qualquer valor possível para ações de trigger, e o tipo de condição (veja abaixo) for "26" - *obrigatório* se `eventsource` do objeto Action estiver definido como "evento interno", `conditiontype` estiver definido como qualquer valor possível para ações internas, e o tipo de condição (veja abaixo) for "26" - *obrigatório* se `eventsource` do objeto Action estiver definido como "evento criado na atualização do status do serviço", `conditiontype` estiver definido como qualquer valor possível para ações de serviço, e o tipo de condição (veja abaixo) for "26"
formulaid	string	ID único arbitrário que é usado para referenciar a condição a partir de uma expressão personalizada. Só pode conter letras maiúsculas. O ID deve ser definido pelo usuário ao modificar as condições do filtro, mas será gerado novamente ao solicitá-las posteriormente.
operator	integer	Operador da condição. Valores possíveis: 0 - *(padrão)* igual; 1 - diferente; 2 - contém; 3 - não contém; 4 - em; 5 - maior ou igual; 6 - menor ou igual; 7 - não está em; 8 - corresponde; 9 - não corresponde; 10 - Sim; 11 - Não.

Note:
Para entender melhor como usar filtros com vários tipos de expressões, veja exemplos nas páginas dos métodos action.get e action.create.

Os seguintes operadores e valores são suportados para cada tipo de condição.

Condição	Nome da condição	Operadores suportados	Valor esperado
0	Grupo de hosts	igual, diferente	ID do grupo de hosts.
1	Host	igual, diferente	ID do host.
2	Trigger	igual, diferente	ID do trigger.
3	Nome do evento	contém, não contém	Nome do evento.
4	Severidade do trigger	igual, diferente, maior ou igual, menor ou igual	Severidade do trigger. Consulte a propriedade `severity` do trigger para uma lista de severidades de trigger suportadas.
5	Valor do trigger	igual	Valor do trigger. Consulte a propriedade `value` do trigger para uma lista de valores de trigger suportados.
6	Período de tempo	em, não está em	Momento em que o evento foi disparado como um período de tempo.

Condição	Nome da condição	Operadores suportados	Valor esperado
7	IP do host	igual, diferente	Um ou vários intervalos de IP para verificar, separados por vírgulas. Consulte a seção configuração de descoberta de rede para mais informações sobre os formatos suportados de intervalos de IP.
8	Tipo de serviço descoberto	igual, diferente	Tipo de serviço descoberto. O tipo de serviço corresponde ao tipo da verificação de descoberta usada para detectar o serviço. Consulte a propriedade type da verificação de descoberta para uma lista de tipos suportados.
9	Porta do serviço descoberto	igual, diferente	Uma ou várias faixas de portas, separadas por vírgulas.
10	Status da descoberta	igual	Status de um objeto descoberto. Valores possíveis: 0 - host ou serviço ativo; 1 - host ou serviço inativo; 2 - host ou serviço descoberto; 3 - host ou serviço perdido.
11	Duração do tempo de atividade ou inatividade	maior ou igual, menor ou igual	Tempo indicando há quanto tempo o objeto descoberto está no status atual em segundos.
12	Valores recebidos	igual, diferente, maior ou igual, menor ou igual, contém, não contém	Valor retornado ao executar uma verificação de descoberta de agent Zabbix, SNMPv1, SNMPv2 ou SNMPv3.
13	Template de host	igual, diferente	ID do template vinculado.
16	Problema está suprimido	Sim, Não	Nenhum valor necessário: usar o operador ”Sim” significa que o problema deve ser suprimido, ”Não” - não suprimido.
18	Regra de descoberta	igual, diferente	ID da regra de descoberta.
19	Verificação de descoberta	igual, diferente	ID da verificação de descoberta.
20	Proxy	igual, diferente	ID do proxy.
21	Objeto de descoberta	igual	Tipo de objeto que disparou o evento de descoberta. Valores possíveis: 1 - host descoberto; 2 - serviço descoberto.
22	Nome do host	contém, não contém, corresponde, não corresponde	Nome do host. O uso de uma expressão regular é suportado para os operadores *corresponde* e *não corresponde* em condições de autoregistro.
23	Tipo de evento	igual	Evento interno específico. Valores possíveis: 0 - item em estado ”não suportado”; 1 - item em estado ”normal”; 2 - regra LLD em estado ”não suportado”; 3 - regra LLD em estado ”normal”; 4 - trigger em estado ”desconhecido”; 5 - trigger em estado ”normal”.
24	Metadados do host	contém, não contém, corresponde, não corresponde	Metadados do host autoregistrado. O uso de uma expressão regular é suportado para os operadores *corresponde* e *não corresponde*.

Condição	Nome da condição	Operadores suportados	Valor esperado
25	Tag	igual, diferente, contém, não contém	Tag do evento.
26	Valor da tag	igual, diferente, contém, não contém	Valor da tag do evento.
27	Serviço	igual, diferente	ID do serviço.
28	Nome do serviço	igual, diferente	Nome do serviço.

action.update

Descrição

`object action.update(object/array actions)`

Este método permite atualizar ações existentes.

> **Note:**
> Este método está disponível apenas para os tipos de usuário *Admin* e *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object/array)` Propriedades da ação a serem atualizadas.

A propriedade `actionid` deve ser definida para cada ação, todas as outras propriedades são opcionais. Somente as propriedades passadas serão atualizadas, todas as outras permanecerão inalteradas.

Além das propriedades padrão da ação, o método aceita os seguintes parâmetros.

Parâmetro	Tipo	Descrição
filter	object	Objeto filtro de ação para substituir o filtro atual.
operations	array	Operações de ação para substituir as operações existentes.
recovery_operations	array	Operações de recuperação de ação para substituir as operações de recuperação existentes. Comportamento do parâmetro: - *suportado* se `eventsource` do objeto Action estiver definido como "evento criado por um trigger", "evento interno" ou "evento criado na atualização do status do serviço"
update_operations	array	Operações de atualização de ação para substituir as operações de atualização existentes. Comportamento do parâmetro: - *suportado* se `eventsource` do objeto Action estiver definido como "evento criado por um trigger" ou "evento criado na atualização do status do serviço"

Valores de retorno

`(object)` Retorna um objeto contendo os IDs das ações atualizadas sob a propriedade `actionids`.

Exemplos

Desabilitar ação

Desabilite uma ação, ou seja, defina seu status como "1".

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "action.update",
    "params": {
        "actionid": "2",
        "status": "1"
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "actionids": [
            "2"
        ]
    },
    "id": 1
}
```

Veja também

- Filtro de ação
- Operação de ação

Fonte

CAction::update() em *ui/include/classes/api/services/CAction.php*.

#####ação.criar {#manual-api-reference-action-create}

Descrição

`object action.create(object/array actions)`

Este método permite criar novas ações.

Note:
Este método está disponível apenas para os tipos de usuário *Admin* e *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função de usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object/array)` Ações a serem criadas.

Além das propriedades padrão da ação, o método aceita os seguintes parâmetros.

Parâmetro	Tipo	Descrição
filter	object	Objeto Action filter para a ação.
operations	array	Operações de ação a serem criadas para a ação.
recovery_operations	array	Operações de recuperação de ação a serem criadas para a ação.
update_operations	array	Operações de atualização de ação a serem criadas para a ação.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs das ações criadas sob a propriedade `actionids`. A ordem dos IDs retornados corresponde a ordem das ações passadas.

Exemplos

Criar uma ação de trigger

Crie uma ação de trigger que será iniciada assim que uma trigger (com a palavra ”memory” em seu nome) do host ”10084” entrar em estado de PROBLEMA. A ação terá 4 operações configuradas. A primeira operação, imediata, enviará uma mensagem para todos os usuários do grupo de usuários ”7” via tipo de mídia ”1”. Se o evento não for resolvido em 30 minutos, a segunda operação executará o script ”5” (script com escopo ”Operação de ação”) em todos os hosts do grupo ”2”. Se o evento for resolvido, uma

operação de recuperação notificará todos os usuários que receberam qualquer mensagem sobre o problema. Se o evento for atualizado, uma operação de reconhecimento/atualização notificará (com assunto e mensagem personalizados) todos os usuários que receberam qualquer mensagem sobre o problema.

Requisição:

```
{
    "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
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "actionids": [
            "17"
        ]
    },
    "id": 1
}
```

Criar uma ação de descoberta

Crie uma ação de descoberta que vinculará o template ”10001” aos hosts descobertos.

Request:

```
{
    "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
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "actionids": [
            "18"
        ]
    },
    "id": 1
}
```

Usando um filtro de expressão personalizada

Crie uma ação de trigger que use uma expressão personalizada - ”A and (B or C)” - para avaliar as condições da ação. Assim que um trigger com severidade maior ou igual a ”Warning” do host ”10084” ou do host ”10106” entrar em estado de PROBLEMA, a ação enviará uma mensagem para todos os usuários no grupo de usuários ”7” via tipo de mídia ”1”. Os IDs de fórmula ”A”, ”B” e ”C” foram escolhidos arbitrariamente.

Requisição:

```
{
    "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
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "actionids": [
            "18"
        ]
    },
    "id": 1
}
```

Criar regra de auto-registro de agent

Crie uma ação de auto-registro que adicione um host ao grupo de hosts ”2” quando o nome do host contiver ”SRV” ou os metadados contiverem ”AlmaLinux”.

Requisição:

```
{
    "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
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "actionids": [
            19
        ]
    },
    "id": 1
}
```

Criar regra de auto-registro de agent com tags de host

Crie uma ação de auto-registro que adicione um host ao grupo de hosts "2" e adicione duas tags de host.

Requisição:

```
{
    "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
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "actionids": [
            20
        ]
    },
```

```
    "id": 1
}
```

Veja também

- Filtro de ação
- Operação de ação
- Script

Fonte

CAction::create() in *ui/include/classes/api/services/CAction.php*.

ação.excluir

Descrição

`object action.delete(array actionIds)`

Este método permite excluir ações.

> **Note:**
> Este método está disponível apenas para os tipos de usuário *Admin* e *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(array)` IDs das ações a serem excluídas.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs das ações excluídas sob a propriedade `actionids`.

Exemplos

Excluir várias ações

Excluir duas ações.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "action.delete",
    "params": [
        "17",
        "18"
    ],
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "actionids": [
            "17",
            "18"
        ]
    },
    "id": 1
}
```

Fonte

CAction::delete() in *ui/include/classes/api/services/CAction.php*.

#####ação.obter {#manual-api-reference-action-get}

Descrição

`integer/array action.get(object parameters)`

O método permite recuperar ações de acordo com os parâmetros fornecidos.

> **Note:**
> Este método está disponível para usuários de qualquer tipo. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object)` Parâmetros que definem a saída desejada.

O método suporta os seguintes parâmetros.

Parâmetro	Tipo	Descrição
actionids	ID/array	Retornar apenas ações com os IDs fornecidos.
groupids	ID/array	Retornar apenas ações que usam os grupos de hosts fornecidos nas condições da ação.
hostids	ID/array	Retornar apenas ações que usam os hosts fornecidos nas condições da ação.
triggerids	ID/array	Retornar apenas ações que usam os triggers fornecidos nas condições da ação.
mediatypeids	ID/array	Retornar apenas ações que usam os tipos de mídia fornecidos para enviar mensagens.
usrgrpids	ID/array	Retornar apenas ações que estão configuradas para enviar mensagens para os grupos de usuários fornecidos.
userids	ID/array	Retornar apenas ações que estão configuradas para enviar mensagens para os usuários fornecidos.
scriptids	ID/array	Retornar apenas ações que estão configuradas para executar os scripts fornecidos.
selectFilter	query	Retornar uma propriedade `filter` com o filtro de condição da ação.
selectOperations	query	Retornar uma propriedade `operations` com as operações da ação.
selectRecoveryOperations	query	Retornar uma propriedade `recovery_operations` com as operações de recuperação da ação.
selectUpdateOperations	query	Retornar uma propriedade `update_operations` com as operações de atualização da ação.
sortfield	string/array	Classificar o resultado pelas propriedades fornecidas. Valores possíveis: `actionid`, `name`, `status`.
countOutput	boolean	Esses parâmetros são descritos no comentário de referência.
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valores de retorno

`(integer/array)` Retorna ou:

- uma array de objetos;
- a contagem de objetos recuperados, se o parâmetro `countOutput` tiver sido usado.

Exemplos

Recuperar ações de trigger

Recupere todas as ações de trigger configuradas juntamente com as condições de ação e operações.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "action.get",
    "params": {
        "output": "extend",
        "selectOperations": "extend",
        "selectRecoveryOperations": "extend",
        "selectUpdateOperations": "extend",
        "selectFilter": "extend",
        "filter": {
            "eventsource": 0
        }
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "actionid": "3",
            "name": "Report problems to Zabbix administrators",
            "eventsource": "0",
            "status": "1",
            "esc_period": "1h",
            "pause_suppressed": "1",
            "filter": {
                "evaltype": "0",
                "formula": "",
                "conditions": [],
                "eval_formula": ""
            },
            "operations": [
                {
                    "operationid": "3",
                    "actionid": "3",
                    "operationtype": "0",
                    "esc_period": "0",
                    "esc_step_from": "1",
                    "esc_step_to": "1",
                    "evaltype": "0",
                    "opconditions": [],
                    "opmessage": [
                        {
                            "default_msg": "1",
                            "subject": "",
                            "message": "",
                            "mediatypeid" => "0"
                        }
                    ],
                    "opmessage_grp": [
                        {
                            "usrgrpid": "7"
                        }
                    ]
                }
            ],
            "recovery_operations": [
                {
                    "operationid": "7",
                    "actionid": "3",
```

```
                    "operationtype": "11",
                    "evaltype": "0",
                    "opconditions": [],
                    "opmessage": {
                        "default_msg": "0",
                        "subject": "{TRIGGER.STATUS}: {TRIGGER.NAME}",
                        "message": "Trigger: {TRIGGER.NAME}\r\nTrigger status: {TRIGGER.STATUS}\r\nTrigger
                        "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": "Updated: {TRIGGER.NAME}",
                        "message": "{USER.FULLNAME} updated problem at {EVENT.UPDATE.DATE} {EVENT.UPDATE.T
                        "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
}
```


Recuperar ações de descoberta

Recupere todas as ações de descoberta configuradas juntamente com as condições e operações da ação. O filtro usa o tipo de

avaliação ”and”, portanto, a propriedade `formula` está vazia e `eval_formula` é gerada automaticamente.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "action.get",
    "params": {
        "output": "extend",
        "selectOperations": "extend",
        "selectFilter": "extend",
        "filter": {
            "eventsource": 1
        }
    },
    "id": 1
}
```

Resposta:

```
{
    "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
}
```


Veja também

- Filtro de ação
- Operação de ação

Fonte

CAction::get() in *ui/include/classes/api/services/CAction.php*.

Alert

Esta classe foi projetada para trabalhar com alertas.

Referências do objeto:

- Alert

Métodos disponíveis:

- alert.get - recupera alertas

Objeto de alerta

Os seguintes objetos estão diretamente relacionados à API 'alert'.

Alerta

> **Note:**
> Alertas são criados pelo Zabbix server e não podem ser modificados via API.

O objeto alerta contém informações sobre se determinadas operações de ação foram executadas com sucesso. Ele possui as seguintes propriedades.

Propriedade	Tipo	Descrição
alertid	ID	ID do alerta.

Propriedade	Tipo	Descrição
actionid	ID	ID da ação que gerou o alerta.
alerttype	integer	Tipo de alerta. Valores possíveis: 0 - mensagem; 1 - comando remoto.
clock	timestamp	Hora em que o alerta foi gerado.
error	string	Texto do erro se houver problemas ao enviar uma mensagem ou executar um comando.
esc_step	integer	Etapa de escalonamento da ação durante a qual o alerta foi gerado.
eventid	ID	ID do evento que acionou a ação.
mediatypeid	ID	ID do tipo de mídia que foi usado para enviar a mensagem.
message	text	Texto da mensagem. Comportamento da propriedade: - *suportado* se `alerttype` estiver definido como ”mensagem”
retries	integer	Número de vezes que o Zabbix tentou enviar a mensagem.
sendto	string	Endereço, nome de usuário ou outro identificador do destinatário. Comportamento da propriedade: - *suportado* se `alerttype` estiver definido como ”mensagem”
status	integer	Status indicando se a operação de ação foi executada com sucesso. Valores possíveis se `alerttype` estiver definido como ”mensagem”: 0 - mensagem não enviada; 1 - mensagem enviada; 2 - falhou após várias tentativas; 3 - novo alerta ainda não processado pelo gerenciador de alertas. Valores possíveis se `alerttype` estiver definido como ”comando remoto”: 0 - comando não executado; 1 - comando executado; 2 - tentou executar o comando no Zabbix agent, mas ele estava indisponível.
subject	string	Assunto da mensagem. Comportamento da propriedade: - *suportado* se `alerttype` estiver definido como ”mensagem”
userid	ID	ID do usuário para o qual a mensagem foi enviada.
p_eventid	ID	ID do evento de problema que gerou o alerta.
acknowledgeid	ID	ID do reconhecimento que gerou o alerta.

#####alerta.obter {#manual-api-reference-alert-get}

Descrição

`integer/array alert.get(object parameters)`

O método permite recuperar alertas de acordo com os parâmetros fornecidos.

Note:
Este método está disponível para usuários de qualquer tipo. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object)` Parâmetros que definem a saída desejada.

O método suporta os seguintes parâmetros.

Parâmetro	Tipo	Descrição
alertids	ID/array	Retorna apenas alertas com os IDs fornecidos.
actionids	ID/array	Retorna apenas alertas gerados pelas ações fornecidas.
eventids	ID/array	Retorna apenas alertas gerados pelos eventos fornecidos.
groupids	ID/array	Retorna apenas alertas gerados por objetos dos grupos de host fornecidos.
hostids	ID/array	Retorna apenas alertas gerados por objetos dos hosts fornecidos.
mediatypeids	ID/array	Retorna apenas alertas de mensagem que usaram os tipos de mídia fornecidos.
objectids	ID/array	Retorna apenas alertas gerados pelos objetos fornecidos
userids	ID/array	Retorna apenas alertas de mensagem que foram enviados para os usuários fornecidos.
eventobject	integer	Retorna apenas alertas gerados por eventos relacionados a objetos do tipo fornecido. Veja o evento `object` para uma lista de tipos de objeto suportados. Padrão: 0 - trigger.
eventsource	integer	Retorna apenas alertas gerados por eventos do tipo fornecido. Veja o evento `source` para uma lista de tipos de evento suportados. Padrão: 0 - eventos de trigger.
time_from	timestamp	Retorna apenas alertas que foram gerados após o tempo fornecido.
time_till	timestamp	Retorna apenas alertas que foram gerados antes do tempo fornecido.
selectHosts	query	Retorna uma propriedade `hosts` com dados dos hosts que acionaram a operação da ação.
selectMediatypes	query	Retorna uma propriedade `mediatypes` com um array dos tipos de mídia que foram usados para o alerta de mensagem. Veja `mediatype.get` para restrições baseadas no tipo de usuário.
selectUsers	query	Retorna uma propriedade `users` com um array dos usuários para os quais a mensagem foi endereçada. Veja `user.get` para restrições baseadas no tipo de usuário.
sortfield	string/array	Ordena o resultado pelas propriedades fornecidas. Valores possíveis: `alertid`, `clock`, `eventid`, `mediatypeid`, `sendto`, `status`.
countOutput	boolean	Estes parâmetros são descritos no comentário de referência.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Note:
No Zabbix 7.0.9, usuários do tipo *Admin* e *User* podem recuperar dados de alerta do tipo ”mensagem” (0) apenas sobre seu próprio usuário.

Valores de retorno

`(integer/array)` Retorna:

- uma matriz de objetos;
- a contagem de objetos recuperados, se o parâmetro `countOutput` tiver sido usado.

Exemplos

Recuperar alertas por ID de ação

Recupere todos os alertas gerados pela ação "3".

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "alert.get",
    "params": {
        "output": "extend",
        "actionids": "3"
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "alertid": "1",
            "actionid": "3",
            "eventid": "21243",
            "userid": "1",
            "clock": "1362128008",
            "mediatypeid": "1",
            "sendto": "support@company.com",
            "subject": "PROBLEM: Zabbix agent on Linux server is unreachable for 5 minutes: ",
            "message": "Trigger: Zabbix agent on Linux server is unreachable for 5 minutes: \nTrigger stat
            "status": "0",
            "retries": "3",
            "error": "",
            "esc_step": "1",
            "alerttype": "0",
            "p_eventid": "0",
            "acknowledgeid": "0"
        }
    ],
    "id": 1
}
```

Veja também

- Host
- Tipo de mídia
- Usuário

Fonte

CAlert::get() em *ui/include/classes/api/services/CAlert.php*.

API info

Esta classe foi projetada para recuperar metainformações sobre a API.

Métodos disponíveis:

- apiinfo.version - recupera a versão da API do Zabbix

apiinfo.versão

Descrição

`string apiinfo.version(array)`

Este método permite recuperar a versão da API do Zabbix.

::: não importante Este método está disponível apenas para não autenticados usuários e deve ser chamado sem o parâmetro `auth` no JSON-RPC solicitação. :::

Parâmetros

`(array)` O método aceita um array vazio.

Valores de retorno

`(string)` Retorna a versão da API do Zabbix.

A partir do Zabbix 2.0.4 a versão da API corresponde à versão do Zabbix.

Exemplos

Recuperando a versão da API

Recupera a versão da API do Zabbix.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "apiinfo.version",
    "params": [],
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": "7.0.0",
    "id": 1
}
```

Fonte

CAPIInfo::version() em *ui/include/classes/api/services/CAPIInfo.php*.

Autenticação

Esta classe foi projetada para trabalhar com configurações de autenticação.

Referências de objeto:

- Autenticação

Métodos disponíveis:

- authentication.get - recuperar configurações de autenticação
- authentication.update - atualizar configurações de autenticação

Objeto de autenticação

Os objetos a seguir estão diretamente relacionados à API de autenticação.

Autenticação

O objeto de autenticação possui as seguintes propriedades.

Propriedade	Tipo	Descrição
authentication_type	integer	Autenticação padrão. Valores possíveis: 0 - *(padrão)* Interna; 1 - LDAP.

Propriedade	Tipo	Descrição
http_auth_enabled	integer	Autenticação HTTP. Valores possíveis: 0 - *(padrão)* Desabilitada; 1 - Habilitada. Comportamento da propriedade: - *suportado* se `$ALLOW_HTTP_AUTH` estiver habilitado no arquivo de configuração do frontend (*zabbix.conf.php*).
http_login_form	integer	Formulário de login padrão. Valores possíveis: 0 - *(padrão)* Formulário de login do Zabbix; 1 - Formulário de login HTTP. Comportamento da propriedade: - *suportado* se `$ALLOW_HTTP_AUTH` estiver habilitado no arquivo de configuração do frontend (*zabbix.conf.php*).
http_strip_domains	string	Nome do domínio a ser removido. Comportamento da propriedade: - *suportado* se `$ALLOW_HTTP_AUTH` estiver habilitado no arquivo de configuração do frontend (*zabbix.conf.php*).
http_case_sensitive	integer	Login HTTP com diferenciação de maiúsculas e minúsculas. Valores possíveis: 0 - Desligado; 1 - *(padrão)* Ligado. Comportamento da propriedade: - *suportado* se `$ALLOW_HTTP_AUTH` estiver habilitado no arquivo de configuração do frontend (*zabbix.conf.php*).
ldap_auth_enabled	integer	Autenticação LDAP. Valores possíveis: 0 - *(padrão)* Desabilitada; 1 - Habilitada.
ldap_case_sensitive	integer	Login LDAP com diferenciação de maiúsculas e minúsculas. Valores possíveis: 0 - Desligado; 1 - *(padrão)* Ligado.
ldap_userdirectoryid	ID	ID do diretório de usuário padrão para autenticação LDAP. Usado para grupos de usuários com `gui_access` definido como LDAP ou Padrão do sistema. Comportamento da propriedade: - *obrigatório* se `ldap_auth_enabled` estiver definido como "Habilitada"
saml_auth_enabled	integer	Autenticação SAML. Valores possíveis: 0 - *(padrão)* Desabilitada; 1 - Habilitada.
saml_case_sensitive	integer	Login SAML com diferenciação de maiúsculas e minúsculas. Valores possíveis: 0 - Desligado; 1 - *(padrão)* Ligado.

Propriedade	Tipo	Descrição
passwd_min_length	integer	Requisito de comprimento mínimo da senha. Os valores possíveis variam de 1 a 70. Padrão: 8.
passwd_check_rules	integer	Regras de verificação de senha. Valores possíveis de bitmap: 0 - Verificar comprimento da senha; 1 - Verificar se a senha usa letras latinas maiúsculas e minúsculas; 2 - Verificar se a senha usa dígitos; 4 - Verificar se a senha usa caracteres especiais; 8 - *(padrão)* Verificar se a senha não está na lista de senhas comumente usadas e não contém derivações da palavra "Zabbix" ou nome, sobrenome ou nome de usuário do usuário. Este é um campo de máscara de bits; qualquer soma dos valores possíveis de bitmap é aceitável (por exemplo, 15 para verificar todas as regras).
ldap_jit_status	integer	Status do provisionamento LDAP. Valores possíveis: 0 - Desabilitado para IdPs LDAP configurados; 1 - Habilitado para IdPs LDAP configurados.
saml_jit_status	integer	Status do provisionamento SAML. Valores possíveis: 0 - Desabilitado para IdPs SAML configurados; 1 - Habilitado para IdPs SAML configurados.
jit_provision_interval	string	Intervalo de tempo entre as solicitações de provisionamento JIT para o usuário logado. Aceita segundos e unidade de tempo com sufixo com suporte a mês e ano (3600s,60m,1h,1d,1M,1y). Valor mínimo: 1h. Padrão: 1h. Disponível apenas para provisionamento LDAP.
disabled_usrgrpid	ID	ID do grupo de usuários ao qual atribuir o usuário desprovisionado. O grupo de usuários deve estar desabilitado e não pode ser habilitado ou excluído quando configurado. Comportamento da propriedade: - *obrigatório* se `ldap_jit_status` estiver definido como "Habilitado para IdPs LDAP configurados" ou `saml_jit_status` estiver definido como "Habilitado para IdPs SAML configurados"
mfa_status	integer	Autenticação multifator. Valores possíveis: 0 - Desabilitada (para todos os métodos MFA configurados); 1 - Habilitada (para todos os métodos MFA configurados).
mfaid	ID	Método MFA padrão para grupos de usuários com MFA habilitado. Comportamento da propriedade: - *obrigatório* se `mfa_status` estiver definido como "Habilitada"

#####autenticação.get {#manual-api-reference-authentication-get}

Descrição

```
object authentication.get(object parameters)
```

O método permite recuperar o objeto de autenticação de acordo com os parâmetros fornecidos.

> **Note:**
> Este método está disponível apenas para o tipo de usuário *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Consulte Funções de usuário para mais informações.

Parâmetros

`(object)` Parâmetros que definem a saída desejada.

O método suporta apenas um parâmetro.

Parâmetro	Tipo	Descrição
output	query	Este parâmetro é descrito no comentário de referência.

Valores de retorno

`(object)` Retorna o objeto de autenticação.

Exemplos

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "authentication.get",
    "params": {
        "output": "extend"
    },
    "id": 1
}
```

Resposta:

```
{
    "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
}
```

Fonte

CAuthentication::get() em *ui/include/classes/api/services/CAuthentication.php*.

authentication.update

Descrição

`object authentication.update(object authentication)` Este método permite atualizar as configurações de autenticação existentes.

> **Note:**
> Este método está disponível apenas para o tipo de usuário *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Ver User roles para maiores informações.

Parâmetros

`(object)` Authentication properties a serem atualizados.

Valores de retorno

`(array)` Retorna o array com os nomes dos parâmetros atualizados.

Exemplos

Request:

```
{
    "jsonrpc": "2.0",
    "method": "authentication.update",
    "params": {
        "http_auth_enabled": 1,
        "http_case_sensitive": 0,
        "http_login_form": 1
    },
    "auth": "038e1d7b1735c6a5436ee9eae095879e",
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": [
        "http_auth_enabled",
        "http_case_sensitive",
        "http_login_form"
    ],
    "id": 1
}
```

Fonte

CAuthentication::update() em *ui/include/classes/api/services/CAuthentication.php*.

Autoregistro

Essa classe foi projetada para trabalhar com parâmetros de autoregistro.

Objetos de referência:

- ·Autoregistro

Métodos disponíveis:

-autoregistration.get - retrieve autoregistration -autoregistration.update - update autoregistration

Objeto de registro automático

Os seguintes objetos estão diretamente relacionados à API de `autoregistration`.

Autoregistration

O objeto de autoregistro possui as seguintes propriedades.

Propriedade	Tipo	Descrição
tls_accept	integer	Tipo de conexões de entrada permitidas para autoregistro. Valores possíveis: 1 - permitir conexões não criptografadas; 2 - permitir TLS com PSK; 3 - permitir conexões não criptografadas e TLS com PSK.
tls_psk_identity	string	Identidade PSK; deve ser pareada com apenas um PSK (entre autoregistration, hosts e proxies). Não inclua informações sensíveis na identidade PSK, pois ela é enviada sem criptografia pela rede para informar ao receptor qual PSK usar. Comportamento da propriedade: - *apenas gravação*
tls_psk	string	Chave pré-compartilhada (PSK); deve ter pelo menos 32 dígitos hexadecimais. Comportamento da propriedade: - *apenas gravação*

autoregistration.get

Descrição

`object autoregistration.get(object parameters)`

O método permite recuperar o objeto de auto-registro de acordo com os parâmetros fornecidos.

Note:
Este método está disponível apenas para o tipo de usuário *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object)` Parâmetros que definem a saída desejada.

O método suporta apenas um parâmetro.

Parâmetro	Tipo	Descrição
output	query	Este parâmetro é descrito no comentário de referência.

Valores de retorno

`(object)` Retorna o objeto de registro automático.

Exemplos

Requisição:

```
{
   "jsonrpc": "2.0",
    "method": "autoregistration.get",
    "params": {
        "output": "extend"
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
```

```
        "tls_accept": "3"
    },
    "id": 1
}
```

Fonte

CAutoregistration::get() in *ui/include/classes/api/services/CAutoregistration.php*.

autoregistration.update

Descrição

`object autoregistration.update(object autoregistration)`

Este método permite atualizar um registro automático existente.

> **Note:**
> Este método está disponível apenas para o tipo de usuário *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object)` Propriedades de autorregistro a serem atualizadas.

Valores de retorno

`(boolean )` Retorna boolean true como resultado de uma atualização bem-sucedida.

Exemplos

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "autoregistration.update",
    "params": {
        "tls_accept": "3",
        "tls_psk_identity": "PSK 001",
        "tls_psk": "11111595725ac58dd977beef14b97461a7c1045b9a1c923453302c5473193478"
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": true,
    "id": 1
}
```

Fonte

CAutoregistration::update() em *ui/include/classes/api/services/CAutoregistration.php*.

Cenário web

Esta classe foi projetada para trabalhar com cenários web.

Referências de objetos:

- Cenário web
- Etapa do cenário
    - Campo HTTP
- Tag do cenário web

Métodos disponíveis:

- httptest.create - criar novos cenários web

- httptest.delete - excluir cenários web
- httptest.get - recuperar cenários web
- httptest.update - atualizar cenários web

Objeto de cenário web

Os seguintes objetos estão diretamente relacionados à API `webcheck`.

Cenário web

O objeto cenário web possui as seguintes propriedades.

Propriedade	Tipo	Descrição
httptestid	ID	ID do cenário web. Comportamento da propriedade: - *somente leitura* - *obrigatório* para operações de atualização
hostid	ID	ID do host ao qual o cenário web pertence. Comportamento da propriedade: - *constante* - *obrigatório* para operações de criação
name	string	Nome do cenário web. Comportamento da propriedade: - *obrigatório* para operações de criação
agent	string	String do agente do usuário que será usada pelo cenário web. Padrão: Zabbix
authentication	integer	Método de autenticação que será usado pelo cenário web. Valores possíveis: 0 - *(padrão)* nenhum; 1 - autenticação HTTP básica; 2 - autenticação NTLM; 3 - autenticação Kerberos; 4 - autenticação Digest.
delay	string	Intervalo de execução do cenário web. Aceita segundos ou unidade de tempo com sufixo (por exemplo, 30s, 1m, 2h, 1d), ou uma macro de usuário. Padrão: 1m.
headers	array	Cabeçalhos HTTP que serão enviados ao realizar uma solicitação.
http_password	string	Senha usada para autenticação HTTP básica, NTLM, Kerberos ou Digest.
http_proxy	string	Proxy que será usado pelo cenário web fornecido como *http://[username[:password]@]proxy.example.com[:port]*.
http_user	string	Nome de usuário usado para autenticação HTTP básica, NTLM, Kerberos ou Digest.
retries	integer	Número de vezes que um cenário web tentará executar cada etapa antes de falhar. Padrão: 1.
ssl_cert_file	string	Nome do arquivo de certificado SSL usado para autenticação do cliente (deve estar no formato PEM).
ssl_key_file	string	Nome do arquivo de chave privada SSL usado para autenticação do cliente (deve estar no formato PEM).
ssl_key_password	string	Senha da chave privada SSL.

Propriedade	Tipo	Descrição
status	integer	Se o cenário web está habilitado. Valores possíveis: 0 - *(padrão)* habilitado; 1 - desabilitado.
templateid	ID	ID do cenário web do template pai. Comportamento da propriedade: - *somente leitura*
variables	array	Variáveis do cenário web.
verify_host	integer	Se deve validar que o nome do host para a conexão corresponde ao do certificado do host. Valores possíveis: 0 - *(padrão)* ignorar verificação do host; 1 - verificar host.
verify_peer	integer	Se deve validar que o certificado do host é autêntico. Valores possíveis: 0 - *(padrão)* ignorar verificação do peer; 1 - verificar peer.
uuid	string	Identificador global único, usado para vincular cenários web importados aos já existentes. Gerado automaticamente, se não fornecido. Comportamento da propriedade: - *suportado* se o cenário web pertencer a um template

Etapa do cenário

O objeto de etapa do cenário define uma verificação específica do cenário web. Ele possui as seguintes propriedades.

Propriedade	Tipo	Descrição
name	string	Nome da etapa do cenário. Comportamento da propriedade: - *obrigatório*
no	integer	Número sequencial da etapa em um cenário web. Comportamento da propriedade: - *obrigatório*
url	string	URL a ser verificada. Comportamento da propriedade: - *obrigatório*
follow_redirects	integer	Se deve seguir redirecionamentos HTTP. Valores possíveis: 0 - não seguir redirecionamentos; 1 - *(padrão)* seguir redirecionamentos.
headers	array	Cabeçalhos HTTP que serão enviados ao realizar uma requisição. Os cabeçalhos da etapa do cenário sobrescreverão os cabeçalhos especificados para o cenário web.
posts	string/array	Variáveis HTTP POST como uma string (dados brutos do post) ou como um array de campos HTTP (dados de campo de formulário).
required	string	Texto que deve estar presente na resposta.

Propriedade	Tipo	Descrição
retrieve_mode	integer	Parte da resposta HTTP que a etapa do cenário deve recuperar. Valores possíveis: 0 - *(padrão)* apenas corpo; 1 - apenas cabeçalhos; 2 - cabeçalhos e corpo.
status_codes	string	Intervalos de códigos de status HTTP necessários, separados por vírgulas.
timeout	string	Tempo limite da requisição em segundos. Aceita segundos, unidade de tempo com sufixo ou uma macro de usuário. Padrão: 15s. Máximo: 1h. Mínimo: 1s.
variables	array	Variáveis da etapa do cenário.
query_fields	array	Campos de consulta - array de campos HTTP que serão adicionados à URL ao realizar uma requisição.

Campo HTTP

O objeto de campo HTTP define o nome e o valor que são usados para especificar as variáveis do cenário web, cabeçalhos HTTP e campos POST ou campos de consulta. Ele possui as seguintes propriedades.

Propriedade	Tipo	Descrição
name	string	Nome do cabeçalho/variável/campo POST ou GET. Comportamento da propriedade: - *obrigatório*
value	string	Valor do cabeçalho/variável/campo POST ou GET. Comportamento da propriedade: - *obrigatório*

Tag de cenário web

O objeto de tag de cenário web possui as seguintes propriedades.

Propriedade	Tipo	Descrição
tag	string	Nome da tag do cenário web. Comportamento da propriedade: - *obrigatório*
value	string	Valor da tag do cenário web.

httptest.create

Descrição

`object httptest.create(object/array webScenarios)`

Este método permite criar novos cenários web.

Note:
A criação de um cenário web criará automaticamente um conjunto de itens de monitoramento web.

Note:
Este método está disponível apenas para os tipos de usuário *Admin* e *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Consulte Funções de usuário para mais informações.

Parâmetros

(object/array) Cenários web a serem criados.

Além das propriedades padrão do cenário web, o método aceita os seguintes parâmetros.

Parâmetro	Tipo	Descrição
steps	array	Etapas do cenário. Comportamento do parâmetro: - *obrigatório*
tags	array	Tags do cenário web.

Valores de retorno

(object) Retorna um objeto contendo os IDs dos cenários da web criados na propriedade `httptestids`. A ordem dos IDs retornados corresponde à ordem dos cenários da Web passados.

Exemplos

Criando um cenário web

Crie um cenário web para monitorar a página inicial da empresa. O cenário terá duas etapas, para verificar a página inicial e a página "Sobre" e certificar-se de que elas retornam o código de status HTTP 200.

Requisição:

```
{
    "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
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "httptestids": [
            "5"
        ]
    },
    "id": 1
}
```

Veja também

- Etapa do cenário

Fonte

CHttpTest::create() in *ui/include/classes/api/services/CHttpTest.php*.

httptest.delete

Descrição

`object httptest.delete(array webScenarioIds)`

Este método permite excluir cenários web.

> **Note:**
> Este método está disponível apenas para os tipos de usuário *Admin* e *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(array)` IDs dos cenários Web a serem excluídos.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos cenários da web excluídos na propriedade `httptestids`.

Exemplos

Excluindo vários cenários web

Exclua dois cenários web.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "httptest.delete",
    "params": [
        "2",
        "3"
    ],
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "httptestids": [
            "2",
            "3"
        ]
    },
    "id": 1
}
```

Fonte

CHttpTest::delete() in *ui/include/classes/api/services/CHttpTest.php*.

httptest.get

Descrição

`integer/array httptest.get(object parameters)`

O método permite recuperar cenários web de acordo com os parâmetros fornecidos.

> **Note:**
> Este método está disponível para usuários de qualquer tipo. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Consulte Funções de usuário para mais informações.

Parâmetros

(object) Parâmetros que definem a saída desejada.

O método suporta os seguintes parâmetros.

Parâmetro	Tipo	Descrição
groupids	ID/array	Retorna apenas cenários web que pertencem aos grupos de hosts informados.
hostids	ID/array	Retorna apenas cenários web que pertencem aos hosts informados.
httptestids	ID/array	Retorna apenas cenários web com os IDs informados.
inherited	boolean	Se definido como `true`, retorna apenas cenários web herdados de um template.
monitored	boolean	Se definido como `true`, retorna apenas cenários web habilitados que pertencem a hosts monitorados.
templated	boolean	Se definido como `true`, retorna apenas cenários web que pertencem a templates.
templateids	ID/array	Retorna apenas cenários web que pertencem aos templates informados.
expandName	flag	Expande macros no nome do cenário web.
expandStepName	flag	Expande macros nos nomes das etapas do cenário.
evaltype	integer	Tag método de avaliação. Valores possíveis: 0 - *(padrão)* E/Ou; 2 - Ou.
tags	array	Retorna apenas cenários web com as tags informadas. Formato: `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`. Um array vazio retorna todos os cenários web. Valores possíveis para operator: 0 - *(padrão)* Contém; 1 - Igual; 2 - Não contém; 3 - Diferente; 4 - Existe; 5 - Não existe.
selectHosts	query	Retorna os hosts aos quais o cenário web pertence como um array na propriedade `hosts`.
selectSteps	query	Retorna as etapas do cenário web na propriedade `steps`. Suporta `count`.
selectTags	query	Retorna as tags do cenário web na propriedade `tags`.
sortfield	string/array	Ordena o resultado pelas propriedades informadas. Valores possíveis: `httptestid`, `name`.
countOutput	boolean	Esses parâmetros são descritos no comentário de referência.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valores de retorno

(integer/array) Retorna:

- um array de objetos;

- a contagem de objetos recuperados, se o parâmetro `countOutput` tiver sido usado.

Exemplos

Recuperando um cenário web

Recupere todos os dados sobre o cenário web "9".

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "httptest.get",
    "params": {
        "output": "extend",
        "selectSteps": "extend",
        "httptestids": "9"
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "httptestid": "9",
            "name": "Homepage check",
            "delay": "1m",
            "status": "0",
            "variables": [],
            "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",
            "headers": [],
            "steps": [
                {
                    "httpstepid": "36",
                    "httptestid": "9",
                    "name": "Homepage",
                    "no": "1",
                    "url": "http://example.com",
                    "timeout": "15s",
                    "posts": "",
                    "required": "",
                    "status_codes": "200",
                    "variables": [
                        {
                            "name":"{var}",
                            "value":"12"
                        }
                    ],
                    "follow_redirects": "1",
                    "retrieve_mode": "0",
                    "headers": [],
```

```
                    "query_fields": []
                },
                {
                    "httpstepid": "37",
                    "httptestid": "9",
                    "name": "Homepage / About",
                    "no": "2",
                    "url": "http://example.com/about",
                    "timeout": "15s",
                    "posts": "",
                    "required": "",
                    "status_codes": "200",
                    "variables": [],
                    "follow_redirects": "1",
                    "retrieve_mode": "0",
                    "headers": [],
                    "query_fields": []
                }
            ]
        }
    ],
    "id": 1
}
```


Veja também

- Host
- Etapa do cenário

Fonte

CHttpTest::get() in *ui/include/classes/api/services/CHttpTest.php*.

httptest.update

Descrição

`object httptest.update(object/array webScenarios)`

Este método permite atualizar cenários web existentes.

> **Note:**
> Este método está disponível apenas para os tipos de usuário *Admin* e *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object/array)` Propriedades do cenário web a serem atualizadas.

A propriedade `httptestid` deve ser definida para cada cenário web, todas as outras propriedades são opcionais. Somente as propriedades passadas serão atualizadas, todas as outras permanecerão inalteradas.

Além das propriedades padrão do cenário web, o método aceita os seguintes parâmetros.

Parâmetro	Tipo	Descrição
steps	array	Etapas do cenário para substituir as etapas existentes.
tags	array	Tags do cenário web.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos cenários da web atualizados na propriedade `httptestid`.

Exemplos

Habilitando um cenário web

Habilite um cenário web, ou seja, defina seu status como ”0”.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "httptest.update",
    "params": {
        "httptestid": "5",
        "status": 0
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "httptestids": [
            "5"
        ]
    },
    "id": 1
}
```

Veja também

- Etapa do cenário

Fonte

CHttpTest::update() in *ui/include/classes/api/services/CHttpTest.php*.

Configuração

Esta classe é projetada para exportar e importar dados de configuração do Zabbix.

Métodos disponíveis:

- configuration.export - exportando a configuração
- configuration.import - importando a configuração

configuration.export

Descrição

`string configuration.export(object parameters)`

Este método permite exportar dados de configuração como uma string serializada.

> **Note:**
> Este método está disponível para usuários de qualquer tipo. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Consulte Funções de usuário para mais informações.

Parâmetros

`(object)` Parâmetros que definem os objetos a serem exportados e o formato a ser utilizado.

Parâmetro	Tipo	Descrição
format	string	Formato no qual os dados devem ser exportados. Valores possíveis: `yaml` - YAML; `xml` - XML; `json` - JSON; `raw` - array PHP não processado. Comportamento do parâmetro: - *obrigatório*
prettyprint	boolean	Torna a saída mais legível para humanos, adicionando indentação. Valores possíveis: `true` - adiciona indentação; `false` - *(padrão)* não adiciona indentação.
options	object	Objetos a serem exportados. O objeto `options` possui os seguintes parâmetros: `host_groups` - `(array)` IDs dos grupos de host a serem exportados; `hosts` - `(array)` IDs dos hosts a serem exportados; `images` - `(array)` IDs das imagens a serem exportadas; `maps` - `(array)` IDs dos mapas a serem exportados; `mediaTypes` - `(array)` IDs dos tipos de mídia a serem exportados; `template_groups` - `(array)` IDs dos grupos de template a serem exportados; `templates` - `(array)` IDs dos templates a serem exportados. Usuários dos tipos *Admin* e *User* podem exportar apenas aqueles objetos para os quais possuem permissão de *somente leitura* ou *leitura e gravação*, bem como imagens, mas não tipos de mídia. Comportamento do parâmetro: - *obrigatório*

Valores de retorno

`(string)` Retorna uma string serializada contendo os dados de configuração solicitados.

Exemplos

Exportando um template

Exporte a configuração do template ”10571” como uma string XML.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "configuration.export",
    "params": {
        "options": {
            "templates": [
                "10571"
            ]
        },
        "format": "xml"
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
```

```
    "result": "<?xml version=\"1.0\" encoding=\"UTF-8\"?>\n<zabbix_export><version>7.0</version><template_
    "id": 1
}
```


Fonte

CConfiguration::export() em *ui/include/classes/api/services/CConfiguration.php*.

configuration.import

Descrição

`boolean configuration.import(object parameters)`

Este método permite importar dados de configuração a partir de uma string serializada.

Note:
Este método está disponível para usuários de qualquer tipo. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Consulte Funções de usuário para mais informações.

Parâmetros

`(object)` Parâmetros contendo os dados a serem importados e regras de como os dados devem ser tratados.

Parâmetro	Tipo	Descrição
format	string	Formato da string serializada. Valores possíveis: `yaml` - YAML; `xml` - XML; `json` - JSON. Comportamento do parâmetro: - *obrigatório*
source	string	String serializada contendo os dados de configuração. Comportamento do parâmetro: - *obrigatório*
rules	object	Regras sobre como novos objetos e objetos existentes devem ser importados. Usuários do tipo *Admin* podem importar apenas aqueles objetos para os quais possuem permissão de *leitura-escrita* (permissão), bem como mapas. Por exemplo, um host e suas entidades (items, triggers, gráficos, etc.) só podem ser importados se o grupo de usuários do usuário tiver permissão para o grupo de hosts ao qual o host importado pertencerá. Imagens e tipos de mídia não podem ser importados por usuários do tipo *Admin*. O parâmetro `rules` é descrito em detalhes na tabela abaixo. Comportamento do parâmetro: - *obrigatório*

Attention:
Se nenhuma regra for fornecida, a configuração não será atualizada.

O objeto `rules` suporta os seguintes parâmetros.

Parâmetro	Tipo	Descrição
discoveryRules	object	Regras sobre como importar regras LLD. Parâmetros suportados: `createMissing` - `(boolean)` se definido como `true`, novas regras LLD serão criadas; padrão: `false`; `updateExisting` - `(boolean)` se definido como `true`, regras LLD existentes serão atualizadas; padrão: `false`; `deleteMissing` - `(boolean)` se definido como `true`, regras LLD não presentes nos dados importados serão excluídas do banco de dados; padrão: `false`.
graphs	object	Regras sobre como importar gráficos. Parâmetros suportados: `createMissing` - `(boolean)` se definido como `true`, novos gráficos serão criados; padrão: `false`; `updateExisting` - `(boolean)` se definido como `true`, gráficos existentes serão atualizados; padrão: `false`; `deleteMissing` - `(boolean)` se definido como `true`, gráficos não presentes nos dados importados serão excluídos do banco de dados; padrão: `false`.
host_groups	object	Regras sobre como importar grupos de hosts. Parâmetros suportados: `createMissing` - `(boolean)` se definido como `true`, novos grupos de hosts serão criados; padrão: `false`; `updateExisting` - `(boolean)` se definido como `true`, grupos de hosts existentes serão atualizados; padrão: `false`.
template_groups	object	Regras sobre como importar grupos de templates. Parâmetros suportados: `createMissing` - `(boolean)` se definido como `true`, novos grupos de templates serão criados; padrão: `false`; `updateExisting` - `(boolean)` se definido como `true`, grupos de templates existentes serão atualizados; padrão: `false`.
hosts	object	Regras sobre como importar hosts. Parâmetros suportados: `createMissing` - `(boolean)` se definido como `true`, novos hosts serão criados; padrão: `false`; `updateExisting` - `(boolean)` se definido como `true`, hosts existentes serão atualizados; padrão: `false`.
httptests	object	Regras sobre como importar cenários web. Parâmetros suportados: `createMissing` - `(boolean)` se definido como `true`, novos cenários web serão criados; padrão: `false`; `updateExisting` - `(boolean)` se definido como `true`, cenários web existentes serão atualizados; padrão: `false`; `deleteMissing` - `(boolean)` se definido como `true`, cenários web não presentes nos dados importados serão excluídos do banco de dados; padrão: `false`.
images	object	Regras sobre como importar imagens. Parâmetros suportados: `createMissing` - `(boolean)` se definido como `true`, novas imagens serão criadas; padrão: `false`; `updateExisting` - `(boolean)` se definido como `true`, imagens existentes serão atualizadas; padrão: `false`.

Parâmetro	Tipo	Descrição
items	object	Regras sobre como importar items. Parâmetros suportados: `createMissing` - `(boolean)` se definido como `true`, novos items serão criados; padrão: `false`; `updateExisting` - `(boolean)` se definido como `true`, items existentes serão atualizados; padrão: `false`; `deleteMissing` - `(boolean)` se definido como `true`, items não presentes nos dados importados serão excluídos do banco de dados; padrão: `false`.
maps	object	Regras sobre como importar mapas. Parâmetros suportados: `createMissing` - `(boolean)` se definido como `true`, novos mapas serão criados; padrão: `false`; `updateExisting` - `(boolean)` se definido como `true`, mapas existentes serão atualizados; padrão: `false`.
mediaTypes	object	Regras sobre como importar tipos de mídia. Parâmetros suportados: `createMissing` - `(boolean)` se definido como `true`, novos tipos de mídia serão criados; padrão: `false`; `updateExisting` - `(boolean)` se definido como `true`, tipos de mídia existentes serão atualizados; padrão: `false`.
templateLinkage	object	Regras sobre como importar links de template. Parâmetros suportados: `createMissing` - `(boolean)` se definido como `true`, templates que não estão vinculados ao host ou template sendo importado, mas estão presentes nos dados importados, serão vinculados; padrão: `false`; `deleteMissing` - `(boolean)` se definido como `true`, templates que estão vinculados ao host ou template sendo importado, mas não estão presentes nos dados importados, serão desvinculados sem remover entidades (items, triggers, etc.) herdadas dos templates desvinculados; padrão: `false`.
templates	object	Regras sobre como importar templates. Parâmetros suportados: `createMissing` - `(boolean)` se definido como `true`, novos templates serão criados; padrão: `false`; `updateExisting` - `(boolean)` se definido como `true`, templates existentes serão atualizados; padrão: `false`.
templateDashboards	object	Regras sobre como importar dashboards de template. Parâmetros suportados: `createMissing` - `(boolean)` se definido como `true`, novos dashboards de template serão criados; padrão: `false`; `updateExisting` - `(boolean)` se definido como `true`, dashboards de template existentes serão atualizados; padrão: `false`; `deleteMissing` - `(boolean)` se definido como `true`, dashboards de template não presentes nos dados importados serão excluídos do banco de dados; padrão: `false`.
triggers	object	Regras sobre como importar triggers. Parâmetros suportados: `createMissing` - `(boolean)` se definido como `true`, novos triggers serão criados; padrão: `false`; `updateExisting` - `(boolean)` se definido como `true`, triggers existentes serão atualizados; padrão: `false`; `deleteMissing` - `(boolean)` se definido como `true`, triggers não presentes nos dados importados serão excluídos do banco de dados; padrão: `false`.

Parâmetro	Tipo	Descrição
valueMaps	object	Regras sobre como importar mapas de valor de host ou template. Parâmetros suportados: `createMissing` - `(boolean)` se definido como `true`, novos mapas de valor serão criados; padrão: `false`; `updateExisting` - `(boolean)` se definido como `true`, mapas de valor existentes serão atualizados; padrão: `false`; `deleteMissing` - `(boolean)` se definido como `true`, mapas de valor não presentes nos dados importados serão excluídos do banco de dados; padrão: `false`.

Valores de retorno

`(boolean)` Retorna `true` se a importação foi bem sucedida.

Exemplos

Importando um template

Importa a configuração do template contida na string XML. Se algum item ou trigger na string XML estiver ausente, ele será excluído do banco de dados e todo o restante permanecerá inalterado.

Requisição:

```
{
    "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>7.0</version><templ
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": true,
    "id": 1
}
```

Fonte

CConfiguration::import() em *ui/include/classes/api/services/CConfiguration.php*.

configuração.importcompare

Descrição

`array configuration.importcompare(object parameters)`

Este método permite comparar o arquivo de importação com os elementos atuais do sistema e mostra o que será alterado se este arquivo de importação for importado.

Note:
Este método está disponível para usuários de qualquer tipo. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Consulte Funções de usuário para mais informações.

Parâmetros

`(object)` Parâmetros contendo os possíveis dados a serem importados e regras de como os dados devem ser tratados.

Parâmetro	Tipo	Descrição
format	string	Formato da string serializada. Valores possíveis: `yaml` - YAML; `xml` - XML; `json` - JSON. Comportamento do parâmetro: - *obrigatório*
source	string	String serializada contendo os dados de configuração. Comportamento do parâmetro: - *obrigatório*
rules	object	Regras sobre como novos objetos e objetos existentes devem ser comparados. Usuários do tipo *Admin* e *User* podem comparar novos objetos apenas com objetos existentes para os quais tenham permissão de *somente leitura* ou *leitura e gravação* permissão. O parâmetro `rules` é descrito em detalhes na tabela abaixo. Comportamento do parâmetro: - *obrigatório*

Attention:
Se nenhuma regra for fornecida, não haverá nada para atualizar e o resultado será vazio.

Note:
A comparação será feita apenas para grupos de hosts e templates. Triggers e gráficos serão comparados apenas para templates importados, qualquer outro será considerado como ”novo”.

O objeto `rules` suporta os seguintes parâmetros.

Parâmetro	Tipo	Descrição
discoveryRules	object	Regras sobre como importar regras LLD. Parâmetros suportados: `createMissing` - `(boolean)` se definido como `true`, novas regras LLD serão criadas; padrão: `false`; `updateExisting` - `(boolean)` se definido como `true`, regras LLD existentes serão atualizadas; padrão: `false`; `deleteMissing` - `(boolean)` se definido como `true`, regras LLD não presentes nos dados importados serão excluídas do banco de dados; padrão: `false`.
graphs	object	Regras sobre como importar gráficos. Parâmetros suportados: `createMissing` - `(boolean)` se definido como `true`, novos gráficos serão criados; padrão: `false`; `updateExisting` - `(boolean)` se definido como `true`, gráficos existentes serão atualizados; padrão: `false`; `deleteMissing` - `(boolean)` se definido como `true`, gráficos não presentes nos dados importados serão excluídos do banco de dados; padrão: `false`.
host_groups	object	Regras sobre como importar grupos de hosts. Parâmetros suportados: `createMissing` - `(boolean)` se definido como `true`, novos grupos de hosts serão criados; padrão: `false`; `updateExisting` - `(boolean)` se definido como `true`, grupos de hosts existentes serão atualizados; padrão: `false`.
template_groups	object	Regras sobre como importar grupos de templates. Parâmetros suportados: `createMissing` - `(boolean)` se definido como `true`, novos grupos de templates serão criados; padrão: `false`; `updateExisting` - `(boolean)` se definido como `true`, grupos de templates existentes serão atualizados; padrão: `false`.
hosts	object	Regras sobre como importar hosts. Parâmetros suportados: `createMissing` - `(boolean)` se definido como `true`, novos hosts serão criados; padrão: `false`; `updateExisting` - `(boolean)` se definido como `true`, hosts existentes serão atualizados; padrão: `false`. Este parâmetro não fará diferença na saída. É permitido apenas por consistência com `configuration.import`.
httptests	object	Regras sobre como importar cenários web. Parâmetros suportados: `createMissing` - `(boolean)` se definido como `true`, novos cenários web serão criados; padrão: `false`; `updateExisting` - `(boolean)` se definido como `true`, cenários web existentes serão atualizados; padrão: `false`; `deleteMissing` - `(boolean)` se definido como `true`, cenários web não presentes nos dados importados serão excluídos do banco de dados; padrão: `false`.

Parâmetro	Tipo	Descrição
images	object	Regras sobre como importar imagens. Parâmetros suportados: `createMissing` - `(boolean)` se definido como `true`, novas imagens serão criadas; padrão: `false`; `updateExisting` - `(boolean)` se definido como `true`, imagens existentes serão atualizadas; padrão: `false`. Este parâmetro não fará diferença na saída. É permitido apenas por consistência com `configuration.import`.
items	object	Regras sobre como importar items. Parâmetros suportados: `createMissing` - `(boolean)` se definido como `true`, novos items serão criados; padrão: `false`; `updateExisting` - `(boolean)` se definido como `true`, items existentes serão atualizados; padrão: `false`; `deleteMissing` - `(boolean)` se definido como `true`, items não presentes nos dados importados serão excluídos do banco de dados; padrão: `false`.
maps	object	Regras sobre como importar mapas. Parâmetros suportados: `createMissing` - `(boolean)` se definido como `true`, novos mapas serão criados; padrão: `false`; `updateExisting` - `(boolean)` se definido como `true`, mapas existentes serão atualizados; padrão: `false`. Este parâmetro não fará diferença na saída. É permitido apenas por consistência com `configuration.import`.
mediaTypes	object	Regras sobre como importar tipos de mídia. Parâmetros suportados: `createMissing` - `(boolean)` se definido como `true`, novos tipos de mídia serão criados; padrão: `false`; `updateExisting` - `(boolean)` se definido como `true`, tipos de mídia existentes serão atualizados; padrão: `false`. Este parâmetro não fará diferença na saída. É permitido apenas por consistência com `configuration.import`.
templateLinkage	object	Regras sobre como importar links de template. Parâmetros suportados: `createMissing` - `(boolean)` se definido como `true`, templates que não estão vinculados ao host ou template sendo importado, mas estão presentes nos dados importados, serão vinculados; padrão: `false`; `deleteMissing` - `(boolean)` se definido como `true`, templates que estão vinculados ao host ou template sendo importado, mas não estão presentes nos dados importados, serão desvinculados sem remover entidades (items, triggers, etc.) herdadas dos templates desvinculados; padrão: `false`.
templates	object	Regras sobre como importar templates. Parâmetros suportados: `createMissing` - `(boolean)` se definido como `true`, novos templates serão criados; padrão: `false`; `updateExisting` - `(boolean)` se definido como `true`, templates existentes serão atualizados; padrão: `false`.

Parâmetro	Tipo	Descrição
templateDashboards	object	Regras sobre como importar dashboards de template. Parâmetros suportados: `createMissing` - `(boolean)` se definido como `true`, novos dashboards de template serão criados; padrão: `false`; `updateExisting` - `(boolean)` se definido como `true`, dashboards de template existentes serão atualizados; padrão: `false`; `deleteMissing` - `(boolean)` se definido como `true`, dashboards de template não presentes nos dados importados serão excluídos do banco de dados; padrão: `false`.
triggers	object	Regras sobre como importar triggers. Parâmetros suportados: `createMissing` - `(boolean)` se definido como `true`, novos triggers serão criados; padrão: `false`; `updateExisting` - `(boolean)` se definido como `true`, triggers existentes serão atualizados; padrão: `false`; `deleteMissing` - `(boolean)` se definido como `true`, triggers não presentes nos dados importados serão excluídos do banco de dados; padrão: `false`.
valueMaps	object	Regras sobre como importar mapas de valor de host ou template. Parâmetros suportados: `createMissing` - `(boolean)` se definido como `true`, novos mapas de valor serão criados; padrão: `false`; `updateExisting` - `(boolean)` se definido como `true`, mapas de valor existentes serão atualizados; padrão: `false`; `deleteMissing` - `(boolean)` se definido como `true`, mapas de valor não presentes nos dados importados serão excluídos do banco de dados; padrão: `false`.

Valores de retorno

`(array)` Retorna um array com as alterações na configuração que serão feitas.

Exemplos

Comparando a importação de um template

Compara o template contido na string XML com os elementos atuais do sistema, e mostra o que será alterado se este template for importado.

Requisição:

```
{
    "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>7.0</version><templ
    },
    "id": 1
}
```

Resposta:

```
{
    "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
}
```

Fonte

CConfiguration::importcompare() em *ui/include/classes/api/services/CConfiguration.php*.

Configurações

Esta classe foi projetada para trabalhar com configurações comuns de administração.

Referências de objetos:

- Configurações

Métodos disponíveis:

- settings.get - recuperar configurações
- settings.update - atualizar configurações

Objeto settings

Os seguintes objetos estão diretamente relacionados à API `settings`.

Configurações

O objeto de configurações possui as seguintes propriedades.

Propriedade	Tipo	Descrição
default_lang	string	Idioma do sistema por padrão. Padrão: `en_US`.
default_timezone	string	Fuso horário do sistema por padrão. Padrão: `system` - padrão do sistema. Para a lista completa de fusos horários suportados, consulte a documentação do PHP.
default_theme	string	Tema padrão. Valores possíveis: `blue-theme` - *(padrão)* Azul; `dark-theme` - Escuro; `hc-light` - Alto contraste claro; `hc-dark` - Alto contraste escuro.
search_limit	integer	Limite para resultados de pesquisa e filtro. Padrão: 1000.
max_overview_table_size	integer	Número máximo de colunas e linhas nos widgets de painel Visão geral de dados e Visão geral de triggers. Padrão: 50.

Propriedade	Tipo	Descrição
max_in_table	integer	Quantidade máxima de elementos a serem exibidos dentro da célula da tabela. Padrão: 50.
server_check_interval	integer	Exibir aviso se o Zabbix server estiver inativo. Valores possíveis: 0 - Não exibir aviso; 10 - *(padrão)* Exibir aviso.
work_period	string	Horário de trabalho. Padrão: 1-5,09:00-18:00.
show_technical_errors	integer	Exibir erros técnicos (PHP/SQL) para usuários que não são Super administradores e para usuários que não fazem parte de grupos de usuários com modo de depuração ativado. Valores possíveis: 0 - *(padrão)* Não exibir erros técnicos; 1 - Exibir erros técnicos.
history_period	string	Período máximo para exibir dados históricos nos widgets de painel Últimos dados, Web e Visão geral de dados. Aceita segundos e unidade de tempo com sufixo. Padrão: 24h.
period_default	string	Período padrão do filtro de tempo. Aceita segundos e unidade de tempo com sufixo, com suporte a mês e ano (30s, 1m, 2h, 1d, 1M, 1y). Padrão: 1h.
max_period	string	Período máximo para o filtro de tempo. Aceita segundos e unidade de tempo com sufixo, com suporte a mês e ano (30s, 1m, 2h, 1d, 1M, 1y). Padrão: 2y.
severity_color_0	string	Cor para o nível de severidade ”Não classificado” como código hexadecimal. Padrão: 97AAB3.
severity_color_1	string	Cor para o nível de severidade ”Informação” como código hexadecimal. Padrão: 7499FF.
severity_color_2	string	Cor para o nível de severidade ”Aviso” como código hexadecimal. Padrão: FFC859.
severity_color_3	string	Cor para o nível de severidade ”Média” como código hexadecimal. Padrão: FFA059.
severity_color_4	string	Cor para o nível de severidade ”Alta” como código hexadecimal. Padrão: E97659.
severity_color_5	string	Cor para o nível de severidade ”Desastre” como código hexadecimal. Padrão: E45959.
severity_name_0	string	Nome para o nível de severidade ”Não classificado”. Padrão: Not classified.
severity_name_1	string	Nome para o nível de severidade ”Informação”. Padrão: Information.
severity_name_2	string	Nome para o nível de severidade ”Aviso”. Padrão: Warning.

Propriedade	Tipo	Descrição
severity_name_3	string	Nome para o nível de severidade ”Média”. Padrão: Average.
severity_name_4	string	Nome para o nível de severidade ”Alta”. Padrão: High.
severity_name_5	string	Nome para o nível de severidade ”Desastre”. Padrão: Disaster.
custom_color	integer	Usar cores personalizadas para status de eventos. Valores possíveis: 0 - *(padrão)* Não usar cores personalizadas para status de eventos; 1 - Usar cores personalizadas para status de eventos.
ok_period	string	Período de exibição de triggers OK. Aceita segundos e unidade de tempo com sufixo. Padrão: 5m.
blink_period	string	Período de piscar triggers na mudança de status. Aceita segundos e unidade de tempo com sufixo. Padrão: 2m.
problem_unack_color	string	Cor para eventos PROBLEM não reconhecidos como código hexadecimal. Padrão: CC0000.
problem_ack_color	string	Cor para eventos PROBLEM reconhecidos como código hexadecimal. Padrão: CC0000.
ok_unack_color	string	Cor para eventos RESOLVED não reconhecidos como código hexadecimal. Padrão: 009900.
ok_ack_color	string	Cor para eventos RESOLVED reconhecidos como código hexadecimal. Padrão: 009900.
problem_unack_style	integer	Piscar para eventos PROBLEM não reconhecidos. Valores possíveis: 0 - Não exibir piscando; 1 - *(padrão)* Exibir piscando.
problem_ack_style	integer	Piscar para eventos PROBLEM reconhecidos. Valores possíveis: 0 - Não exibir piscando; 1 - *(padrão)* Exibir piscando.
ok_unack_style	integer	Piscar para eventos RESOLVED não reconhecidos. Valores possíveis: 0 - Não exibir piscando; 1 - *(padrão)* Exibir piscando.
ok_ack_style	integer	Piscar para eventos RESOLVED reconhecidos. Valores possíveis: 0 - Não exibir piscando; 1 - *(padrão)* Exibir piscando.
url	string	URL do frontend.
discovery_groupid	ID	ID do grupo de hosts ao qual os hosts descobertos serão automaticamente atribuídos.

Propriedade	Tipo	Descrição
default_inventory_mode	integer	Modo de inventário de host padrão. Valores possíveis: -1 - *(padrão)* Desabilitado; 0 - Manual; 1 - Automático.
alert_usrgrpid	ID	ID do grupo de usuários para o qual será enviada a mensagem de alarme de banco de dados inativo. Se definido como ”0”, a mensagem de alarme não será enviada.
snmptrap_logging	integer	Registrar traps SNMP não correspondentes. Valores possíveis: 0 - Não registrar traps SNMP não correspondentes; 1 - *(padrão)* Registrar traps SNMP não correspondentes.
login_attempts	integer	Número de tentativas de login com falha após as quais o formulário de login será bloqueado. Padrão: 5.
login_block	string	Intervalo de tempo durante o qual o formulário de login será bloqueado se o número de tentativas de login com falha exceder o definido no campo login_attempts. Aceita segundos e unidade de tempo com sufixo. Padrão: 30s.
validate_uri_schemes	integer	Validar esquemas de URI. Valores possíveis: 0 - Não validar; 1 - *(padrão)* Validar.
uri_valid_schemes	string	Esquemas de URI válidos. Padrão: http,https,ftp,file,mailto,tel,ssh.
x_frame_options	string	Cabeçalho HTTP X-Frame-Options. Padrão: SAMEORIGIN.
iframe_sandboxing_enabled	integer	Usar sandboxing em iframe. Valores possíveis: 0 - Não usar; 1 - *(padrão)* Usar.
iframe_sandboxing_exceptions	string	Exceções de sandboxing para iframe.
connect_timeout	string	Tempo limite de conexão com o Zabbix server. Faixa de valores possíveis: 1-30s. Padrão: 3s. Comportamento da propriedade: - *obrigatório*
socket_timeout	string	Tempo limite de rede padrão. Faixa de valores possíveis: 1-300s. Padrão: 3s. Comportamento da propriedade: - *obrigatório*

Propriedade	Tipo	Descrição
media_type_test_timeout	string	Tempo limite de rede para teste de tipo de mídia. Faixa de valores possíveis: 1-300s. Padrão: 65s. Comportamento da propriedade: - *obrigatório*
item_test_timeout	string	Tempo limite de rede para testes de item. Faixa de valores possíveis: 1-600s. Padrão: 60s. Comportamento da propriedade: - *obrigatório*
script_timeout	string	Tempo limite de rede para execução de script. Faixa de valores possíveis: 1-300s. Padrão: 60s. Comportamento da propriedade: - *obrigatório*
report_test_timeout	string	Tempo limite de rede para teste de relatório agendado. Faixa de valores possíveis: 1-300s. Padrão: 60s. Comportamento da propriedade: - *obrigatório*
auditlog_enabled	integer	Se o registro de auditoria está habilitado. Valores possíveis: 0 - Desabilitar; 1 - *(padrão)* Habilitar.
auditlog_mode	integer	Se o registro de auditoria de descoberta de baixo nível, descoberta de rede e atividades de autorregistro realizadas pelo server (usuário do sistema) está habilitado. Valores possíveis: 0 - Desabilitar; 1 - *(padrão)* Habilitar.
ha_failover_delay	string	Atraso de failover em segundos. Padrão: 1m. Comportamento da propriedade: - *somente leitura*

Propriedade	Tipo	Descrição
geomaps_tile_provider	string	Provedor de tiles para geomapas. Valores possíveis: `OpenStreetMap.Mapnik` - *(padrão)* OpenStreetMap Mapnik; `OpenTopoMap` - OpenTopoMap; `Stamen.TonerLite` - Stamen Toner Lite; `Stamen.Terrain` - Stamen Terrain; `USGS.USTopo` - USGS US Topo; `USGS.USImagery` - USGS US Imagery. Suporta string vazia para especificar valores personalizados de `geomaps_tile_url`, `geomaps_max_zoom` e `geomaps_attribution`.
geomaps_tile_url	string	URL do tile do geomapa. Comportamento da propriedade: - *suportado* se `geomaps_tile_provider` estiver definido como string vazia
geomaps_max_zoom	integer	Nível máximo de zoom do geomapa. Valores possíveis: 0-30. Comportamento da propriedade: - *suportado* se `geomaps_tile_provider` estiver definido como string vazia
geomaps_attribution	string	Texto de atribuição do geomapa. Comportamento da propriedade: - *suportado* se `geomaps_tile_provider` estiver definido como string vazia
vault_provider	integer	Provedor de Vault. Valores possíveis: 0 - *(padrão)* HashiCorp Vault; 1 - CyberArk Vault.
timeout_zabbix_agent timeout_simple_check timeout_snmp_agent timeout_external_check timeout_db_monitor timeout_http_agent timeout_ssh_agent timeout_telnet_agent timeout_script timeout_browser	string	Não gastar mais do que `timeout_*` segundos no processamento. Aceita segundos ou unidade de tempo com sufixo (ex.: 30s, 1m). Também aceita macros de usuário. Faixa de valores possíveis: 1-600s. Padrão: 3s. Padrão para `timeout_browser`: 60s. Comportamento da propriedade: - *obrigatório*

settings.get

Descrição

`object settings.get(object parameters)`

O método permite recuperar o objeto de configurações de acordo com os parâmetros fornecidos.

> **Note:**
> Este método está disponível para usuários de qualquer tipo. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object)` Parâmetros que definem a saída desejada.

O método suporta apenas um parâmetro.

Parâmetro	Tipo	Descrição
output	query	Este parâmetro é descrito no comentário de referência.

Valores de retorno

`(object)` Retorna o objeto de configurações.

Exemplos

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "settings.get",
    "params": {
        "output": "extend"
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "default_theme": "blue-theme",
        "search_limit": "1000",
        "max_in_table": "50",
        "server_check_interval": "10",
        "work_period": "1-5,09:00-18:00",
        "show_technical_errors": "0",
        "history_period": "24h",
        "period_default": "1h",
        "max_period": "2y",
        "severity_color_0": "97AAB3",
        "severity_color_1": "7499FF",
        "severity_color_2": "FFC859",
        "severity_color_3": "FFA059",
        "severity_color_4": "E97659",
        "severity_color_5": "E45959",
        "severity_name_0": "Not classified",
        "severity_name_1": "Information",
        "severity_name_2": "Warning",
        "severity_name_3": "Average",
        "severity_name_4": "High",
        "severity_name_5": "Disaster",
        "custom_color": "0",
        "ok_period": "5m",
        "blink_period": "2m",
        "problem_unack_color": "CC0000",
        "problem_ack_color": "CC0000",
        "ok_unack_color": "009900",
        "ok_ack_color": "009900",
```

```
        "problem_unack_style": "1",
        "problem_ack_style": "1",
        "ok_unack_style": "1",
        "ok_ack_style": "1",
        "discovery_groupid": "5",
        "default_inventory_mode": "-1",
        "alert_usrgrpid": "7",
        "snmptrap_logging": "1",
        "default_lang": "en_US",
        "default_timezone": "system",
        "login_attempts": "5",
        "login_block": "30s",
        "validate_uri_schemes": "1",
        "uri_valid_schemes": "http,https,ftp,file,mailto,tel,ssh",
        "x_frame_options": "SAMEORIGIN",
        "iframe_sandboxing_enabled": "1",
        "iframe_sandboxing_exceptions": "",
        "max_overview_table_size": "50",
        "connect_timeout": "3s",
        "socket_timeout": "3s",
        "media_type_test_timeout": "65s",
        "script_timeout": "60s",
        "item_test_timeout": "60s",
        "url": "",
        "report_test_timeout": "60s",
        "auditlog_enabled": "1",
        "auditlog_mode": "1",
        "ha_failover_delay": "1m",
        "geomaps_tile_provider": "OpenStreetMap.Mapnik",
        "geomaps_tile_url": "",
        "geomaps_max_zoom": "0",
        "geomaps_attribution": "",
        "vault_provider": "0",
        "timeout_zabbix_agent": "3s",
        "timeout_simple_check": "3s",
        "timeout_snmp_agent": "3s",
        "timeout_external_check": "3s",
        "timeout_db_monitor": "3s",
        "timeout_http_agent": "3s",
        "timeout_ssh_agent": "3s",
        "timeout_telnet_agent": "3s",
        "timeout_script": "3s"
    },
    "id": 1
}
```


Fonte

CSettings::get() in *ui/include/classes/api/services/CSettings.php*.

settings.update

Descrição

`object settings.update(object settings)`

Este método permite atualizar as configurações comuns existentes.

> **Note:**
> Este método está disponível apenas para o tipo de usuário *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

(object) Propriedades de configurações a serem atualizadas.

Valores de retorno

(array) Retorna um array com os nomes dos parâmetros atualizados.

Exemplos

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "settings.update",
    "params": {
        "login_attempts": "1",
        "login_block": "1m"
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": [
        "login_attempts",
        "login_block"
    ],
    "id": 1
}
```

Fonte

CSettings::update() in *ui/include/classes/api/services/CSettings.php*.

Connector

Essa classe foi projetada para trabalhar com objetos de conexão.

Referências de objeto:

- Connector
- Connector tag filter

Métodos disponíveis:

- Connector
- Connector tag filter
- Connector
- Connector tag filter
- connector.create - creating new connectors
- connector.delete - deleting connectors
- connector.get - retrieving connectors
- connector.update - updating connectors

Objeto Connector

Os seguintes objetos estão diretamente relacionados à API connector.

Conector

O objeto conector possui as seguintes propriedades.

Propriedade	Tipo	Descrição
connectorid	ID	ID do conector. Comportamento da propriedade: - *somente leitura* - *obrigatório* para operações de atualização
name	string	Nome do conector. Comportamento da propriedade: - *obrigatório* para operações de criação
url	string	URL do endpoint, ou seja, URL do receptor. Macros de usuário são suportadas. Comportamento da propriedade: - *obrigatório* para operações de criação
protocol	integer	Protocolo de comunicação. Valores possíveis: 0 - *(padrão)* Zabbix Streaming Protocol v1.0.
data_type	integer	Tipo de dado. Valores possíveis: 0 - *(padrão)* Valores de item; 1 - Eventos.
item_value_type	integer	Uma soma dos tipos de valores de item a serem enviados. Valores possíveis: 1 - Numérico (float); 2 - Caractere; 4 - Log; 8 - Numérico (sem sinal); 16 - Texto; 32 - Binário (suportado desde o Zabbix 7.0.5). Padrão: 31 - Todos os tipos de item (exceto binário). Comportamento da propriedade: - *suportado* se `data_type` estiver definido como "Valores de item".
max_records	integer	Número máximo de eventos ou itens que podem ser enviados em uma mensagem. Valores possíveis: 0-2147483647 (valor máximo de um inteiro assinado de 32 bits). Padrão: 0 - Ilimitado.
max_senders	integer	Número de processos de envio a serem executados para este conector. Valores possíveis: 1-100. Padrão: 1.
max_attempts	integer	Número de tentativas. Valores possíveis: 1-5. Padrão: 1.

Propriedade	Tipo	Descrição
attempt_interval	string	O intervalo entre as tentativas de repetição. Aceita segundos. Valores possíveis: 0s-10s. Padrão: 5s. Comportamento da propriedade: - *suportado* se `max_attempts` for maior que 1.
timeout	string	Timeout. Sufixos de tempo são suportados (por exemplo, 30s, 1m). Macros de usuário são suportadas. Valores possíveis: 1s-60s. Padrão: 5s.
http_proxy	string	String de conexão do proxy HTTP(S) fornecida como *[protocol]://[username[:password]@]proxy.example.com[:port]*. Macros de usuário são suportadas.
authtype	integer	Método de autenticação HTTP. Valores possíveis: 0 - *(padrão)* Nenhum; 1 - Básico; 2 - NTLM; 3 - Kerberos; 4 - Digest; 5 - Bearer.
username	string	Nome de usuário. Macros de usuário são suportadas. Comportamento da propriedade: - *suportado* se `authtype` estiver definido como ”Básico”, ”NTLM”, ”Kerberos” ou ”Digest”
password	string	Senha. Macros de usuário são suportadas. Comportamento da propriedade: - *suportado* se `authtype` estiver definido como ”Básico”, ”NTLM”, ”Kerberos” ou ”Digest”
token	string	Token Bearer. Macros de usuário são suportadas. Comportamento da propriedade: - *obrigatório* se `authtype` estiver definido como ”Bearer”
verify_peer	integer	Se deve validar que o certificado do host é autêntico. Valores possíveis: 0 - Não validar; 1 - *(padrão)* Validar.
verify_host	integer	Se deve validar que o nome do host para a conexão corresponde ao do certificado do host. Valores possíveis: 0 - Não validar; 1 - *(padrão)* Validar.
ssl_cert_file	string	Caminho do arquivo da chave pública SSL. Macros de usuário são suportadas.
ssl_key_file	string	Caminho do arquivo da chave privada SSL. Macros de usuário são suportadas.

Propriedade	Tipo	Descrição
ssl_key_password	string	Senha para o arquivo da chave SSL. Macros de usuário são suportadas.
description	text	Descrição do conector.
status	integer	Se o conector está habilitado. Valores possíveis: 0 - Desabilitado; 1 - *(padrão)* Habilitado.
tags_evaltype	integer	Método de avaliação das tags. Valores possíveis: 0 - *(padrão)* And/Or; 2 - Or.

Filtro de tag

O filtro de tag permite exportar apenas valores de item ou eventos correspondentes. Se não for definido, tudo será exportado. O objeto de filtro de tag possui as seguintes propriedades.

Propriedade	Tipo	Descrição
tag	string	Nome da tag. Comportamento da propriedade: - *obrigatório*
operator	integer	Operador da condição. Valores possíveis: 0 - *(padrão)* Igual; 1 - Diferente; 2 - Contém; 3 - Não contém; 12 - Existe; 13 - Não existe.
value	string	Valor da tag.

connector.create

Descrição

`object connector.create(object/array connectors)`

Este método permite criar novos objetos conectores.

Note:
Este método está disponível apenas para o tipo de usuário *Super admin*. As permissões para executar o método podem ser revogadas nas configurações de função do usuário. Para mais informações, consulte funções do usuário User roles.

Parâmetros

`(object/array)` Objetos de conector a serem criados.

Além das propriedades padrão do conector, o método aceita os seguintes parâmetros.

Parâmetro	Tipo	Descrição
tags	array	Filtro de tag do conector.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos conectores criados sob a propriedade `connectorids`. A ordem dos IDs retornados corresponde à ordem dos conectores passados.

Exemplos

Criando um conector

Crie um conector para exportar eventos trigger com um filtro de tag. A autenticação HTTP será realizada usando um token Bearer.

Solicitação:

```
{
    "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
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "connectorid": [
            "3"
        ]
    },
    "id": 1
}
```

Fonte

CConnector::create() in *ui/include/classes/api/services/CConnector.php*.

connector.delete

Descrição

`object connector.delete(array connectorids)`

Este método permite excluir entradas de conectores.

> **Note:**
> Este método está disponível apenas para o tipo de usuário *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de Usuário para mais informações.

Parâmetros

`(array)` IDs dos conectores a serem excluídos.

Valores de Retorno

`(object)` Retorna um objeto contendo os IDs dos conectores excluídos na propriedade `connectorids`.

Exemplos

Excluindo múltiplos conectores

Exclua duas entradas de conector.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "connector.delete",
    "params": [
        3,
        5
    ],
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "connectorids": [
            "3",
            "5"
        ]
    },
    "id": 1
}
```

Fonte

CConnector::delete() em *ui/include/classes/api/services/CConnector.php*.

Método connector.get

######Descrição

`integer/array connector.get(object parameters)`

O método permite recuperar objetos de conectores de acordo com determinados parâmetros.

:::Nota Este método só está disponível para usuário do tipo *Super admin*. Permissões para chamar o método podem ser retiradas nas configurações de roles. Veja [User roles] (/manual/web_interface/frontend_sections/users/user_roles) para mais informações.

Parâmetros

`(object)` Parâmetros que definem a saída desejada.

O método suporta os seguintes parâmetros.

Parâmetro	Tipo	Descrição
connectorids	ID/array	Retorna apenas os connectors com os IDs fornecidos.
selectTags	query	Retorna uma propriedade `tags` com o filtro de tag do connector. Suporta `count`.
sortfield	string/array	Ordena o resultado pelas propriedades fornecidas. Valores possíveis: `connectorid`, `name`, `data_type`, `status`.
countOutput	boolean	Esses parâmetros são descritos no comentário de referência.
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valores de retorno

`(integer/array)` Retorna:

- um array (vetor) de objetos, ou;
- a contagem de objetos recuperados, se o parâmetro `countOutput` foi usado.

Exemplos

Recuperando todos os conectores

Recupere todos os dados sobre todos os conectores e suas propriedades.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "connector.get",
    "params": {
        "output": "extend",
        "selectTags": ["tag", "operator", "value"],
        "preservekeys": true
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "connectorid": "1",
            "name": "Export of item values",
            "protocol": "0",
            "data_type": "0",
            "url": "{$DATA_EXPORT_VALUES_URL}",
            "item_value_type": "31",
            "authtype": "4",
            "username": "{$DATA_EXPORT_VALUES_USERNAME}",
            "password": "{$DATA_EXPORT_VALUES_PASSWORD}",
            "token": "",
            "max_records": "0",
            "max_senders": "4",
            "max_attempts": "2",
            "attempt_interval": "10s",
            "timeout": "10s",
            "http_proxy": "{$DATA_EXPORT_VALUES_PROXY}",
            "verify_peer": "1",
            "verify_host": "1",
            "ssl_cert_file": "{$DATA_EXPORT_VALUES_SSL_CERT_FILE}",
            "ssl_key_file": "{$DATA_EXPORT_VALUES_SSL_KEY_FILE}",
            "ssl_key_password": "",
            "description": "",
            "status": "1",
            "tags_evaltype": "0",
            "tags": [
                {
                    "tag": "component",
                    "operator": "0",
                    "value": "memory"
                }
            ]
        },
        {
            "connectorid": "2",
            "name": "Export of events",
```

```
            "protocol": "0",
            "data_type": "1",
            "url": "{$DATA_EXPORT_EVENTS_URL}",
            "item_value_type": "31",
            "authtype": "5",
            "username": "",
            "password": "",
            "token": "{$DATA_EXPORT_EVENTS_BEARER_TOKEN}",
            "max_records": "0",
            "max_senders": "2",
            "max_attempts": "1",
            "attempt_interval": "5s",
            "timeout": "5s",
            "http_proxy": "",
            "verify_peer": "1",
            "verify_host": "1",
            "ssl_cert_file": "",
            "ssl_key_file": "",
            "ssl_key_password": "",
            "description": "",
            "status": "1",
            "tags_evaltype": "0",
            "tags": [
                {
                    "tag": "scope",
                    "operator": "0",
                    "value": "performance"
                }
            ]
        }
    ],
    "id": 1
}
```

Fonte

CConnector:get() em *ui/include/classes/api/services/CConnector.php*.

Método connector.update

Descrição

`object connector.update(object/array connectors)`

This method allows to update existing connectors.

Este método só é disponível para usuários do tipo *Super admin*. Permissões para chamar o método podem ser revogadas na configurações de role do usuário. Consulte [User roles] (/manual/web_interface/frontend_sections/users/user_roles) for more information.

Parâmetros

`(object/array)` Propriedades do conector a serem atualizadas.

A propriedade `connectorid` deve ser definida para cada conector, todas as outras propriedades são opcionais. Somente as propriedades passadas serão atualizadas, todas as outras permanecerão inalteradas.

Além das propriedades padrão do conector, o método aceita os seguintes parâmetros.

Parâmetro	Tipo	Descrição
tags	array	Filtro de tag do conector para substituir o filtro de tag atual.

Valores retornados

`(object)` Retorna um objeto contendo os IDs dos conectores atualizados sob a propriedade `connectorids`.

Exemplos

Alterando o tipo de autenticação HTTP

Altere o tipo de autenticação HTTP para Bearer para o conector com ID ”3”.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "connector.update",
    "params": {
        "connectorid": 3,
        "authtype": 5,
        "token": "{$DATA_EXPORT_BEARER_TOKEN}"
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "connectorids": [
            "3"
        ]
    },
    "id": 1
}
```

Atualização do filtro tag

Alterar o filtro de tag para o conector com ID ”5”.

Request:

```
{
    "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
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "connectorids": [
            "5"
```

```
        ]
    },
    "id": 1
}
```

Fonte

CConnector::update() in *ui/include/classes/api/services/CConnector.php*.

Correlation

Esta classe é projetada para trabalhar com correlações.

Referências de objetos:

- Correlation
- Correlation operation
- Correlation filter
  - Correlation filter condition

Métodos disponíveis:

- correlation.create - criar novas correlações
- correlation.delete - excluir correlações
- correlation.get - recuperar correlações
- correlation.update - atualizar correlações

Objeto de correlação

Os objetos a seguir estão diretamente relacionados à API `correlation`.

Correlação

O objeto de correlação possui as seguintes propriedades.

Propriedade	Tipo	Descrição
correlationid	ID	ID da correlação. Comportamento da propriedade: - *somente leitura* - *obrigatório* para operações de atualização
name	string	Nome da correlação. Comportamento da propriedade: - *obrigatório* para operações de criação
description	string	Descrição da correlação.
status	integer	Se a correlação está habilitada ou desabilitada. Valores possíveis: 0 - *(padrão)* habilitada; 1 - desabilitada.

Operação de correlação

O objeto de operação de correlação define uma operação que será executada quando uma correlação for executada. Ele possui as seguintes propriedades.

Propriedade	Tipo	Descrição
type	integer	Tipo de operação. Valores possíveis: 0 - fechar eventos antigos; 1 - fechar novo evento. Comportamento da propriedade: - *obrigatório*

Filtro de correlação

O objeto de filtro de correlação define um conjunto de condições que devem ser atendidas para executar as operações de correlação configuradas. Possui as seguintes propriedades.

Propriedade	Tipo	Descrição
conditions	array	Conjunto de condições de filtro a serem usadas para filtrar resultados. As condições serão classificadas na ordem de sua colocação na fórmula. Comportamento da propriedade: - *obrigatório*
evaltype	integer	Método de avaliação da condição do filtro. Valores possíveis: 0 - E/Ou; 1 - E; 2 - Ou; 3 - Expressão personalizada. Comportamento da propriedade: - *obrigatório*
eval_formula	string	Expressão gerada que será usada para avaliar as condições do filtro. A expressão contém IDs que referenciam condições de filtro específicas por seu `formulaid`. O valor de `eval_formula` é igual ao valor de `formula` para filtros com uma expressão personalizada. Comportamento da propriedade: - *somente leitura*
formula	string	Expressão definida pelo usuário a ser usada para avaliar as condições de filtros com uma expressão personalizada. A expressão deve conter IDs que referenciem condições de filtro específicas por seu `formulaid`. Os IDs usados na expressão devem corresponder exatamente aos definidos nas condições do filtro: nenhuma condição pode permanecer não utilizada ou omitida. Comportamento da propriedade: - *obrigatório* se `evaltype` estiver definido como ”expressão personalizada”

Condição do filtro de correlação

O objeto de condição do filtro de correlação define uma condição específica que deve ser verificada antes de executar as operações de correlação.

Propriedade	Tipo	Descrição
type	integer	Tipo de condição. Valores possíveis: 0 - tag de evento antigo; 1 - tag de evento novo; 2 - grupo de hosts do novo evento; 3 - par de tags de evento; 4 - valor da tag de evento antigo; 5 - valor da tag de evento novo. Comportamento da propriedade: - *obrigatório*
tag	string	Tag do evento (antigo ou novo). Comportamento da propriedade: - *obrigatório* se `type` estiver definido como ”tag de evento antigo”, ”tag de evento novo”, ”valor da tag de evento antigo” ou ”valor da tag de evento novo”
groupid	ID	ID do grupo de hosts. Comportamento da propriedade: - *obrigatório* se `type` estiver definido como ”grupo de hosts do novo evento”
oldtag	string	Tag do evento antigo. Comportamento da propriedade: - *obrigatório* se `type` estiver definido como ”par de tags de evento”
newtag	string	Tag do evento antigo. Comportamento da propriedade: - *obrigatório* se `type` estiver definido como ”par de tags de evento”
value	string	Valor da tag do evento (antigo ou novo). Comportamento da propriedade: - *obrigatório* se `type` estiver definido como ”valor da tag de evento antigo” ou ”valor da tag de evento novo”
formulaid	string	ID único arbitrário que é usado para referenciar a condição a partir de uma expressão personalizada. Só pode conter letras maiúsculas. O ID deve ser definido pelo usuário ao modificar as condições do filtro, mas será gerado novamente ao solicitá-las posteriormente.
operator	integer	Operador da condição. Comportamento da propriedade: - *obrigatório* se `type` estiver definido como ”grupo de hosts do novo evento”, ”valor da tag de evento antigo” ou ”valor da tag de evento novo”

Note:
Para entender melhor como usar filtros com vários tipos de expressões, consulte exemplos nas páginas dos métodos correlation.get e correlation.create.

Os seguintes operadores e valores são suportados para cada tipo de condição.

Condição	Nome da condição	Operadores suportados	Valor esperado
2	Grupo de hosts	=, <>	ID do grupo de hosts.
4	Valor da tag de evento antigo	=, <>, like, not like	string
5	Valor da tag de evento novo	=, <>, like, not like	string

Descrição

`integer/array correlation.get(object parameters)`

O método permite recuperar correlações de acordo com os parâmetros fornecidos.

> **Note:**
> Este método está disponível para usuários de qualquer tipo. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object)` Parâmetros que definem a saída desejada.

O método suporta os seguintes parâmetros.

Parâmetro	Tipo	Descrição
correlationids	ID/array	Retorna apenas as correlações com os IDs fornecidos.
selectFilter	query	Retorna uma propriedade `filter` com as condições de correlação.
selectOperations	query	Retorna uma propriedade `operations` com as operações de correlação.
sortfield	string/array	Ordena o resultado pelas propriedades fornecidas. Valores possíveis: `correlationid`, `name`, `status`.
countOutput	boolean	Esses parâmetros são descritos no comentário de referência.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valores de retorno

`(integer/array)` Retorna:

- uma matriz de objetos;
- a contagem de objetos recuperados, se o parâmetro `countOutput` tiver sido usado.

Exemplos

Recuperar correlações

Recupere todas as correlações configuradas juntamente com as condições e operações de correlação. O filtro usa o tipo de avaliação "and/or", portanto, a propriedade `formula` fica vazia e `eval_formula` é gerada automaticamente.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "correlation.get",
    "params": {
        "output": "extend",
        "selectOperations": "extend",
        "selectFilter": "extend"
    },
    "id": 1
}
```

Resposta:

```
{
    "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
}
```


Veja também

- Filtro de correlação
- Operação de correlação

Fonte

CCorrelation::get() in *ui/include/classes/api/services/CCorrelation.php*.

correlation.update

Descrição

`object correlation.update(object/array correlations)`

Este método permite atualizar correlações existentes.

Note:
Este método está disponível apenas para o tipo de usuário *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object/array)` Propriedades de correlação a serem atualizadas.

A propriedade `correlationid` deve ser definida para cada correlação, todas as outras propriedades são opcionais. Somente as propriedades passadas serão atualizadas, todas as outras permanecerão inalteradas.

Além das propriedades padrão de correlação, o método aceita os seguintes parâmetros.

Parâmetro	Tipo	Descrição
filter	object	Objeto Correlation filter para substituir o filtro atual.
operations	array	Correlation operations para substituir as operações existentes.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs do correlações sob a propriedade `correlationids`.

Exemplos

Desabilitar correlação

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "correlation.update",
    "params": {
        "correlationid": "1",
        "status": "1"
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "correlationids": [
            "1"
        ]
    },
    "id": 1
}
```

Substituir condições, mas manter o método de avaliação

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "correlation.update",
    "params": {
        "correlationid": "1",
        "filter": {
            "conditions": [
                {
                    "type": 3,
                    "oldtag": "error",
                    "newtag": "ok"
                }
            ]
        }
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "correlationids": [
            "1"
        ]
    },
    "id": 1
}
```

Veja também

- Filtro de correlação

- Operação de correlação

Fonte

CCorrelation::update() em *ui/include/classes/api/services/CCorrelation.php*.

Método correlation.delete

Descrição

`correlação de objeto.delete(ids de correlação de matriz)`

Este método permite excluir correlações.

> **Note:**
> Este método está disponível apenas para o tipo de usuário *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Ver Do utilizador funções Para maiores informações.

Parâmetros

`(array)` IDs das correlações a serem excluídas.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs do correlações sob a propriedade `correlationids`.

Exemplos

Excluir várias correlações

Delete duas correlações.

Solicitação:

```
{
    "jsonrpc": "2.0",
    "método": "correlação.excluir",
    "parâmetros": [
        "1",
        "2"
    ],
    "auth": "343baad4f88b4106b9b5961e77437688",
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "resultado": {
        "correlaionídeos": [
            "1",
            "2"
        ]
    },
    "id": 1
}
```

Fonte

CCorrelation::delete() em *ui/include/classes/api/services/CCorrelation.php*.

#####Método correlation.create {#manual-api-reference-correlation-create}

Descrição

`object correlation.create(object/array correlations)`

Este método permite criar novas correlações.

Note:
Este método está disponível apenas para o tipo de usuário *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object/array)` Correlações a serem criadas.

Além das propriedades padrão de correlação, o método aceita os seguintes parâmetros.

Parâmetro	Tipo	Descrição
operations	array	Operações de correlação a serem criadas para a correlação. Comportamento do parâmetro: - *obrigatório*
filter	object	Objeto filtro de correlação para a correlação. Comportamento do parâmetro: - *obrigatório*

Valores de retorno

`(object)` Retorna um objeto contendo os IDs do correlações sob a propriedade `correlationids`. A ordem do os IDs retornados correspondem à ordem das correlações passadas.

Exemplos

Criar uma nova correlação de tag de evento

Crie uma correlação usando o método de avaliação `AND/OR` com uma condição e uma operação. Por padrão, a correlação será ativada.

Requisição:

```
{
    "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
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "correlationids": [
            "1"
        ]
    },
```

```
    "id": 1
}
```

Usando um filtro de expressão personalizada

Crie uma correlação que usará uma condição de filtro personalizada. Os IDs de fórmula "A" ou "B" foram escolhidos arbitrariamente. O tipo de condição será "Grupo de hosts" com o operador "<>".

Requisição:

```
{
    "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
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "correlationids": [
            "2"
        ]
    },
    "id": 1
}
```

Veja também

- Filtro de correlação
- Operação de correlação

Fonte

CCorrelation::create() em *ui/include/classes/api/services/CCorrelation.php*.

Dashboard

Esta classe foi projetada para trabalhar com dashboards.

Referências de objetos:

- Dashboard
- Página do dashboard
  - Widget do dashboard
    - Campo do widget do dashboard
- Grupo de usuários do dashboard
- Usuário do dashboard

Métodos disponíveis:

- dashboard.create - criar novos dashboards
- dashboard.delete - excluir dashboards
- dashboard.get - recuperar dashboards
- dashboard.update - atualizar dashboards

Objeto do painel

Os objetos a seguir estão diretamente relacionados à API `dashboard`.

Dashboard

O objeto dashboard possui as seguintes propriedades.

Propriedade	Tipo	Descrição
dashboardid	ID	ID do dashboard. Comportamento da propriedade: - *somente leitura* - *obrigatório* para operações de atualização
name	string	Nome do dashboard. Comportamento da propriedade: - *obrigatório* para operações de criação
userid	ID	ID do usuário que é o proprietário do dashboard.
private	integer	Tipo de compartilhamento do dashboard. Valores possíveis: 0 - dashboard público; 1 - *(padrão)* dashboard privado.
display_period	integer	Período padrão de exibição da página (em segundos). Valores possíveis: 10, 30, 60, 120, 600, 1800, 3600. Padrão: 30.
auto_start	integer	Iniciar apresentação de slides automaticamente. Valores possíveis: 0 - não iniciar apresentação de slides automaticamente; 1 - *(padrão)* iniciar apresentação de slides automaticamente.

Página do dashboard

O objeto de página do dashboard possui as seguintes propriedades.

Propriedade	Tipo	Descrição
dashboard_pageid	ID	ID da página do dashboard. Comportamento da propriedade: - *somente leitura*
name	string	Nome da página do dashboard. Padrão: string vazia.

Propriedade	Tipo	Descrição
display_period	integer	Período de exibição da página do dashboard (em segundos). Valores possíveis: 0, 10, 30, 60, 120, 600, 1800, 3600. Padrão: 0 (usará o período de exibição padrão da página).
widgets	array	Array de objetos dashboard widget.

Widget do dashboard

O objeto widget do dashboard possui as seguintes propriedades.

Propriedade	Tipo	Descrição
widgetid	ID	ID do widget do dashboard. Comportamento da propriedade: - *somente leitura*
type	string	Tipo do widget do dashboard. Valores possíveis: actionlog - Log de ações; clock - Relógio; *(obsoleto)* dataover - Visão geral dos dados; discovery - Status da descoberta; favgraphs - Gráficos favoritos; favmaps - Mapas favoritos; gauge - Medidor; geomap - Geomapa; graph - Gráfico (clássico); graphprototype - Protótipo de gráfico; honeycomb - Colmeia; hostavail - Disponibilidade do host; hostnavigator - Navegador de host; itemhistory - Histórico do item; itemnavigator - Navegador de item; item - Valor do item; map - Mapa; navtree - Árvore de navegação do mapa; piechart - Gráfico de pizza; problemhosts - Hosts com problemas; problems - Problemas; problemsbysv - Problemas por severidade; slareport - Relatório de SLA; svggraph - Gráfico; systeminfo - Informações do sistema; tophosts - Principais hosts; toptriggers - Principais triggers; trigover - Visão geral das triggers; url - URL; web - Monitoramento web. Comportamento da propriedade: - *obrigatório*
name	string	Nome personalizado do widget.
x	integer	Posição horizontal a partir do lado esquerdo do dashboard. Os valores possíveis variam de 0 a 71.
y	integer	Posição vertical a partir do topo do dashboard. Os valores possíveis variam de 0 a 63.

Propriedade	Tipo	Descrição
width	integer	A largura do widget. Os valores possíveis variam de 1 a 72.
height	integer	A altura do widget. Os valores possíveis variam de 1 a 64.
view_mode	integer	O modo de visualização do widget. Valores possíveis: 0 - *(padrão)* visualização padrão do widget; 1 - com cabeçalho oculto;
fields	array	Array de objetos campo do widget do dashboard. Comportamento da propriedade: - veja widgets individuais em Campos do widget do dashboard

Campo do widget do dashboard

O objeto de campo do widget do dashboard possui as seguintes propriedades.

Propriedade	Tipo	Descrição
type	integer	Tipo do campo do widget. Valores possíveis: 0 - Inteiro; 1 - String; 2 - Grupo de hosts; 3 - Host; 4 - Item; 5 - Protótipo de item; 6 - Gráfico; 7 - Protótipo de gráfico; 8 - Mapa; 9 - Serviço; 10 - SLA; 11 - Usuário; 12 - Ação; 13 - Tipo de mídia. Comportamento da propriedade: - *obrigatório*
name	string	Nome do campo do widget. Valores possíveis: veja Campos do widget do dashboard. Comportamento da propriedade: - *obrigatório*
value	mixed	Valor do campo do widget dependendo do tipo. Valores possíveis: veja Campos do widget do dashboard. Comportamento da propriedade: - *obrigatório*

Grupo de usuários do dashboard

Lista de permissões do dashboard com base nos grupos de usuários. Possui as seguintes propriedades.

Propriedade	Tipo	Descrição
usrgrpid	ID	ID do grupo de usuários. Comportamento da propriedade: - *obrigatório*
permission	integer	Tipo de nível de permissão. Valores possíveis: 2 - somente leitura; 3 - leitura e gravação. Comportamento da propriedade: - *obrigatório*

Usuário do dashboard

Lista de permissões do dashboard com base nos usuários. Possui as seguintes propriedades.

Propriedade	Tipo	Descrição
userid	ID	ID do usuário. Comportamento da propriedade: - *obrigatório*
permission	integer	Tipo de nível de permissão. Valores possíveis: 2 - somente leitura; 3 - leitura e gravação. Comportamento da propriedade: - *obrigatório*

dashboard.create (criar dashboard)

Descrição

`object dashboard.create(object/array dashboards)`

Este método permite criar novos painéis.

Note:
Este método está disponível para usuários de qualquer tipo. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Consulte Funções de usuário para obter mais informações.

Parâmetros

`(object/array)` Dashboards a serem criados.

Além das propriedades padrão do dashboard, o método aceita os seguintes parâmetros.

Parâmetro	Tipo	Descrição
pages	array	Páginas do dashboard a serem criadas para o dashboard. As páginas do dashboard serão ordenadas na mesma ordem especificada. Comportamento do parâmetro: - *obrigatório*
users	array	Compartilhamentos de usuários do dashboard a serem criados no dashboard.
userGroups	array	Compartilhamentos de grupos de usuários do dashboard a serem criados no dashboard.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs do criado painéis sob a propriedade `dashboardids`. A ordem do retorno Os IDs correspondem à ordem dos painéis passados.

Exemplos

Criando um dashboard

Crie um dashboard chamado "My dashboard" com um widget de Problemas com tags e usando dois tipos de compartilhamento (grupo de usuários e usuário) em uma única página do dashboard.

Requisição:

```
{
    "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
```

```
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "2"
        ]
    },
    "id": 1
}
```

Veja também

- Página do dashboard
- Widget do dashboard
- Campo do widget do dashboard
- Usuário do dashboard
- Grupo de usuários do dashboard

Fonte

CDashboard::create() em *ui/include/classes/api/services/CDashboard.php*.

painel.delete

Descrição

`object dashboard.delete(array dashboardids)`

Este método permite excluir dashboards.

> **Note:**
> Este método está disponível para usuários de qualquer tipo. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(array)` IDs dos painéis a serem excluídos.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs do painéis sob a propriedade `dashboardids`.

Exemplos

Excluindo vários painéis

Exclua dois painéis.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.delete",
    "params": [
        "2",
        "3"
    ],
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
```

```
            "2",
            "3"
        ]
    },
    "id": 1
}
```

Fonte

CDashboard::delete() in *ui/include/classes/api/services/CDashboard.php*.

#####painel.get {#manual-api-reference-dashboard-get}

Descrição

`integer/array dashboard.get(object parameters)`

O método permite recuperar painéis de acordo com os parâmetros fornecidos.

> **Note:**
> Este método está disponível para usuários de qualquer tipo. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object)` Parâmetros que definem a saída desejada.

O método suporta os seguintes parâmetros.

Parâmetro	Tipo	Descrição
dashboardids	ID/array	Retornar apenas dashboards com os IDs fornecidos.
selectPages	query	Retornar uma propriedade `pages` com as páginas do dashboard, corretamente ordenadas.
selectUsers	query	Retornar uma propriedade `users` com os usuários com quem o dashboard é compartilhado.
selectUserGroups	query	Retornar uma propriedade `userGroups` com os grupos de usuários com quem o dashboard é compartilhado.
sortfield	string/array	Ordenar o resultado pelas propriedades fornecidas. Valores possíveis: `dashboardid`.
countOutput	boolean	Esses parâmetros são descritos no comentário de referência.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valores de retorno

`(integer/array)` Retorna:

- uma matriz de objetos;
- a contagem de objetos recuperados, se o parâmetro `countOutput` tiver sido usado.

Exemplos

Recuperando um dashboard pelo ID

Recupere todos os dados sobre os dashboards "1" e "2".

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.get",
    "params": {
        "output": "extend",
        "selectPages": "extend",
        "selectUsers": "extend",
        "selectUserGroups": "extend",
        "dashboardids": [
            "1",
            "2"
        ]
    },
    "id": 1
}
```

Resposta:

```
{
    "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
}
```

Veja também

- Página do dashboard
- Widget do dashboard
- Campo do widget do dashboard
- Usuário do dashboard
- Grupo de usuários do dashboard

Fonte

CDashboard::get() em *ui/include/classes/api/services/CDashboard.php*.

dashboard.update (atualização de dashboard)

Descrição

`painel de objetos.update(painéis de objetos/matrizes)`

Este método permite atualizar os painéis existentes.

> **Note:**
> Este método está disponível para usuários de qualquer tipo. Permissões para chamar o método pode ser revogado nas configurações de função do usuário. Veja Usuário funções Para maiores informações.

Parâmetros

`(object/array)` Propriedades do dashboard a serem atualizadas.

A propriedade `dashboardid` deve ser definida para cada dashboard, todas as outras propriedades são opcionais. Somente as propriedades passadas serão atualizadas, todas as outras permanecerão inalteradas.

Além das propriedades padrão do dashboard, o método aceita os seguintes parâmetros.

Parâmetro	Tipo	Descrição
pages	array	Páginas do dashboard para substituir as páginas existentes do dashboard. As páginas do dashboard são atualizadas pela propriedade `dashboard_pageid`. Novas páginas do dashboard serão criadas para objetos sem a propriedade `dashboard_pageid` e as páginas existentes do dashboard serão excluídas se não forem reutilizadas. As páginas do dashboard serão ordenadas na mesma ordem especificada. Somente as propriedades especificadas das páginas do dashboard serão atualizadas.
users	array	Compartilhamentos de usuários do dashboard para substituir os elementos existentes.
userGroups	array	Compartilhamentos de grupos de usuários do dashboard para substituir os elementos existentes.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs do painéis sob a propriedade `dashboardids`.

Exemplos

Renomeando um dashboard

Renomeie um dashboard para ”SQL server status”.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.update",
```

```
    "params": {
        "dashboardid": "2",
        "name": "SQL server status"
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "2"
        ]
    },
    "id": 1
}
```

Atualizando páginas do dashboard

Renomeie a primeira página do dashboard, substitua os widgets na segunda página do dashboard e adicione uma nova página como a terceira. Exclua todas as outras páginas do dashboard.

Requisição:

```
{
    "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
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "2"
        ]
    },
    "id": 1
```

```
}
```

Alterar o proprietário do dashboard

Disponível apenas para administradores e superadministradores.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.update",
    "params": {
        "dashboardid": "2",
        "userid": "1"
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "2"
        ]
    },
    "id": 1
}
```

Veja também

- Página do dashboard
- Widget do dashboard
- Campo do widget do dashboard
- Usuário do dashboard
- Grupo de usuários do dashboard

Fonte

CDashboard::update() em *ui/include/classes/api/services/CDashboard.php*.

Campos do widget do dashboard

Esta página contém links de navegação para parâmetros de widgets do dashboard e possíveis valores de propriedades para os respectivos objetos dashboard widget field.

Para ver os parâmetros e valores de propriedades de cada widget, acesse as páginas individuais dos widgets para:

- Log de ações
- Relógio
- Status de descoberta
- Gráficos favoritos
- Mapas favoritos
- Medidor
- Geomapa
- Gráfico
- Gráfico (clássico)
- Protótipo de gráfico
- Favo de mel
- Disponibilidade do host
- Navegador de host
- Histórico do item
- Navegador de item
- Valor do item
- Mapa
- Árvore de navegação do mapa

- Gráfico de pizza
- Hosts com problemas
- Problemas
- Relatório de SLA
- Informações do sistema
- Problemas por severidade
- Top hosts
- Top triggers
- Visão geral dos triggers
- URL
- Monitoramento web

Widgets obsoletos:

- Visão geral dos dados

Attention:
Os widgets obsoletos serão removidos na próxima versão principal.

1 Log de ação

Descrição

Esses parâmetros e os possíveis valores de propriedade para os respectivos objetos de campo do widget do dashboard permitem configurar o widget *Action log* nos métodos `dashboard.create` e `dashboard.update`.

Attention:
As propriedades `fields` do widget não são validadas durante a criação ou atualização de um dashboard. Isso permite que os usuários modifiquem widgets internos e criem widgets personalizados, mas também introduz o risco de criar ou atualizar widgets incorretamente. Para garantir a criação ou atualização bem-sucedida do widget *Action log*, consulte o comportamento do parâmetro descrito nas tabelas abaixo.

Parâmetros

Os seguintes parâmetros são suportados para o widget *Log de ações*.

Parâmetro	tipo	nome	valor
Intervalo de atualização	0	rf_rate	0 - Sem atualização; 10 - 10 segundos; 30 - 30 segundos; 60 - *(padrão)* 1 minuto; 120 - 2 minutos; 600 - 10 minutos; 900 - 15 minutos.
Destinatários	11	userids.0	ID do Usuário. Nota: Para configurar vários usuários, crie um objeto de campo de widget de dashboard para cada usuário com um número incrementado no nome da propriedade.
Ações	12	actionids.0	ID da Ação. Nota: Para configurar várias ações, crie um objeto de campo de widget de dashboard para cada ação com um número incrementado no nome da propriedade.
Tipos de mídia	13	mediatypeids.0	ID do Tipo de mídia. Nota: Para configurar vários tipos de mídia, crie um objeto de campo de widget de dashboard para cada tipo de mídia com um número incrementado no nome da propriedade.

Parâmetro	tipo	nome	valor
Status	0	statuses.0	0 - Em andamento; 1 - Enviado/Executado; 2 - Falhou. Nota: Para configurar vários valores, crie um objeto de campo de widget de dashboard para cada valor com um número incrementado no nome da propriedade.
String de pesquisa	1	message	Qualquer valor de string.
Período de tempo	1	time_period._referenceDASHBOARD._timeperiod - define o seletor de *Período de tempo* como fonte de dados; ABCDE._timeperiod - define um widget compatível (com seu parâmetro *Referência* definido como "ABCDE") como fonte de dados. Padrão: DASHBOARD._timeperiod Alternativamente, você pode definir o período de tempo apenas nos parâmetros *De* e *Até*.
De	1	time_period.from	String de tempo válida em sintaxe absoluta (YYYY-MM-DD hh:mm:ss) ou relativa (now, now/d, now/w-1w, etc.). Comportamento do parâmetro: - *suportado* se *Período de tempo* não estiver definido - *obrigatório* se time_period.to estiver definido
Até	1	time_period.to	String de tempo válida em sintaxe absoluta (YYYY-MM-DD hh:mm:ss) ou relativa (now, now/d, now/w-1w, etc.). Comportamento do parâmetro: - *suportado* se *Período de tempo* não estiver definido - *obrigatório* se time_period.from estiver definido
Ordenar entradas por	0	sort_triggers	3 - Tempo (ascendente); 4 - *(padrão)* Tempo (descendente); 5 - Tipo (ascendente); 6 - Tipo (descendente); 7 - Status (ascendente); 8 - Status (descendente); 11 - Destinatário (ascendente); 12 - Destinatário (descendente).
Mostrar linhas	0	show_lines	Os valores possíveis variam de 1-100. Padrão: 25.

Exemplos

Os exemplos a seguir têm como objetivo apenas descrever a configuração dos objetos de campo do widget do painel para o widget *Action log*. Para obter mais informações sobre como configurar um painel, consulte dashboard.create.

Configurando um widget *Action log*

Configure um widget *Action log* que exibe 10 entradas de detalhes de operações de ação, ordenadas por hora (em ordem crescente). Além disso, exiba detalhes apenas para aquelas operações de ação que tentaram enviar um e-mail para o usuário "1", mas não tiveram sucesso.

Requisição:

```
{
    "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
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
```

```
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Veja também

- Campo de widget do dashboard
- `dashboard.create`
- `dashboard.update`

2 Relógio (clock)

Descrição

Esses parâmetros e os possíveis valores de propriedade para os respectivos objetos de campo do widget de dashboard permitem configurar o widget *Relógio* nos métodos `dashboard.create` e `dashboard.update`.

Attention:
As propriedades `fields` do widget não são validadas durante a criação ou atualização de um dashboard. Isso permite que os usuários modifiquem widgets internos e criem widgets personalizados, mas também introduz o risco de criar ou atualizar widgets incorretamente. Para garantir a criação ou atualização bem-sucedida do widget *Relógio*, consulte o comportamento dos parâmetros descritos nas tabelas abaixo.

Parâmetros

Os seguintes parâmetros são suportados para o widget *Clock*.

Parâmetro	tipo	nome	valor
Intervalo de atualização	0	rf_rate	0 - Sem atualização; 10 - 10 segundos; 30 - 30 segundos; 60 - 1 minuto; 120 - 2 minutos; 600 - 10 minutos; 900 - *(padrão)* 15 minutos.
Tipo de hora	0	time_type	0 - *(padrão)* Hora local; 1 - Hora do servidor; 2 - Hora do host.
Tipo de relógio	0	clock_type	0 - *(padrão)* Analógico; 1 - Digital.

Os seguintes parâmetros são suportados se o *Tipo de hora* estiver definido como ”Hora do host”.

Parâmetro	tipo	nome	valor
Item	4	itemid.0	ID do item. Comportamento do parâmetro: - *obrigatório*

Os seguintes parâmetros são suportados se o *Tipo de relógio* estiver definido como ”Digital”.

Parâmetro	tipo	nome	valor
Mostrar	0	show.0	1 - Data; 2 - *(padrão)* Hora; 3 - Fuso horário. Observação: Para configurar vários valores, crie um objeto de campo de widget de dashboard para cada valor com um número incrementado no nome da propriedade.

Configuração avançada

Os seguintes parâmetros de configuração avançada são suportados se o *Tipo de relógio* estiver definido como ”Digital”.

Parâmetro	tipo	nome	valor
Cor de fundo	1	bg_color	Código de cor hexadecimal (por exemplo, FF0000). Padrão: "" (vazio).

Data

Os seguintes parâmetros de configuração avançada são suportados se o *Tipo de relógio* estiver definido como ”Digital” e *Mostrar* estiver definido como ”Data”.

Parâmetro	tipo	nome	valor
Negrito	0	date_bold	0 - *(padrão)* Desabilitado; 1 - Habilitado.
Cor	1	date_color	Código de cor hexadecimal (por exemplo, FF0000). Padrão: "" (vazio).

Hora

Os seguintes parâmetros de configuração avançada são suportados se o *Tipo de relógio* estiver definido como ”Digital” e *Mostrar* estiver definido como ”Hora”.

Parâmetro	tipo	nome	valor
Negrito	0	time_bold	0 - *(padrão)* Desabilitado; 1 - Habilitado.
Cor	1	time_color	Código de cor hexadecimal (por exemplo, FF0000). Padrão: "" (vazio).
Segundos	0	time_sec	0 - Desabilitado; 1 - *(padrão)* Habilitado.
Formato	0	time_format	0 - *(padrão)* 24 horas; 1 - 12 horas.

Fuso horário

Os seguintes parâmetros de configuração avançada são suportados se o *Tipo de relógio* estiver definido como ”Digital” e *Mostrar* estiver definido como ”Fuso horário”.

Parâmetro	tipo	nome	valor
Negrito	0	tzone_bold	0 - *(padrão)* Desabilitado; 1 - Habilitado.
Cor	1	tzone_color	Código de cor hexadecimal (por exemplo, FF0000). Padrão: "" (vazio).

Parâmetro	tipo	nome	valor
Fuso horário	1	tzone_timezone	String de fuso horário válida (por exemplo, `Europe/Riga`, `system`, `UTC`, etc.). Para a lista completa de fusos horários suportados, consulte a documentação do PHP. Padrão: `local`. Comportamento do parâmetro: - *suportado* se o *Tipo de hora* estiver definido como "Hora local" ou "Hora do servidor"
Formato	0	tzone_format	0 - *(padrão)* Curto; 1 - Completo. Comportamento do parâmetro: - *suportado* se o *Tipo de hora* estiver definido como "Hora local" ou "Hora do servidor"

Exemplos

Os seguintes exemplos visam apenas descrever a configuração do campo objetos de dashboard widget para o o widget *relógio* (clock). Para mais informações sobre configuração de deshboard, veja `dashboard.create`.

Configurando um widget *Clock*

Configure um widget *Clock* que exibe a data, hora e fuso horário local em um relógio digital personalizado.

Requisição:

```
{
    "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
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Veja também

-Campo widget do dashboard -`dashboard.create` -`dashboard.update`

Attention:
Este widget foi descontinuado e será removido na próxima grande atualização. Considere utilizar os widgets *Top hosts*.

Descrição

Esses parâmetros e os possíveis valores de propriedade para os respectivos objetos de campo do widget do dashboard permitem configurar o widget *Visão geral dos dados* nos métodos `dashboard.create` e `dashboard.update`.

Attention:
As propriedades `fields` do widget não são validadas durante a criação ou atualização de um dashboard. Isso permite que os usuários modifiquem widgets internos e criem widgets personalizados, mas também introduz o risco de criar ou atualizar widgets incorretamente. Para garantir a criação ou atualização bem-sucedida do widget *Visão geral dos dados*, consulte o comportamento dos parâmetros descrito nas tabelas abaixo.

Parâmetros

Os seguintes parâmetros são suportados para o widget *Visão geral dos dados*.

Parâmetro	type	name	value
Intervalo de atualização	0	rf_rate	0 - Sem atualização; 10 - 10 segundos; 30 - 30 segundos; 60 - *(padrão)* 1 minuto; 120 - 2 minutos; 600 - 10 minutos; 900 - 15 minutos.
Grupos de hosts	2	groupids.0	ID do Grupo de hosts. Nota: Para configurar múltiplos grupos de hosts, crie um objeto de campo de widget de dashboard para cada grupo de hosts com um número incrementado no nome da propriedade. Este parâmetro não é suportado se o widget estiver sendo configurado em um dashboard de template.
Grupos de hosts (Widget)	1	groupids._reference	Em vez do ID do Grupo de hosts: `ABCDE._hostgroupids` - defina um widget compatível (com seu parâmetro *Referência* definido como "ABCDE") como a fonte de dados para grupos de hosts. Este parâmetro não é suportado se o widget estiver sendo configurado em um dashboard de template.
Hosts	3	hostids.0	ID do Host. Nota: Para configurar múltiplos hosts, crie um objeto de campo de widget de dashboard para cada host com um número incrementado no nome da propriedade. Para múltiplos hosts, o parâmetro *Grupos de hosts* deve não estar configurado ou estar configurado com pelo menos um grupo de hosts ao qual os hosts configurados pertençam. Este parâmetro não é suportado se o widget estiver sendo configurado em um dashboard de template.

Parâmetro	type	name	value
Hosts (Widget/Dashboard)	1	hostids._reference	Em vez do ID do Host: `DASHBOARD.hostids` - defina o seletor *Host* como a fonte de dados para hosts; `ABCDE._hostids` - defina um widget compatível (com seu parâmetro *Referência* definido como "ABCDE") como a fonte de dados para hosts. Este parâmetro não é suportado se o widget estiver sendo configurado em um dashboard de template.
Tags de item			
Tipo de avaliação	0	evaltype	0 - *(padrão)* E/Ou; 2 - Ou.
Nome da tag	1	tags.0.tag	Qualquer valor de string. Nota: O número no nome da propriedade faz referência à ordem da tag na lista de avaliação de tags. Comportamento do parâmetro: - *obrigatório* se estiver configurando *Tags de item*
Operador	0	tags.0.operator	0 - Contém; 1 - Igual; 2 - Não contém; 3 - Não é igual; 4 - Existe; 5 - Não existe. Nota: O número no nome da propriedade faz referência à ordem da tag na lista de avaliação de tags. Comportamento do parâmetro: - *obrigatório* se estiver configurando *Tags de item*
Valor da tag	1	tags.0.value	Qualquer valor de string. Nota: O número no nome da propriedade faz referência à ordem da tag na lista de avaliação de tags. Comportamento do parâmetro: - *obrigatório* se estiver configurando *Tags de item*
Mostrar problemas suprimidos	0	show_suppressed	0 - *(padrão)* Desabilitado; 1 - Habilitado.
Localização dos hosts	0	style	0 - *(padrão)* Esquerda; 1 - Superior.

Exemplos

Os exemplos a seguir têm como objetivo apenas descrever a configuração dos objetos de campo do widget de painel para o widget *Visão geral dos dados*. Para obter mais informações sobre como configurar um painel, consulte `dashboard.create`.

Configurando um widget *Visão geral dos dados*

Configure um widget *Visão geral dos dados* que exibe dados para o host "10084" e apenas para os items para os quais a tag com o nome "component" contém o valor "cpu". Além disso, exiba os dados com os hosts localizados no topo.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "dataover",
                        "name": "Data overview",
                        "x": 0,
                        "y": 0,
                        "width": 36,
                        "height": 5,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 3,
                                "name": "hostids.0",
                                "value": 10084
                            },
                            {
                                "type": 1,
                                "name": "tags.0.tag",
                                "value": "component"
                            },
                            {
                                "type": 0,
                                "name": "tags.0.operator",
                                "value": 0
                            },
                            {
                                "type": 1,
                                "name": "tags.0.value",
                                "value": "cpu"
                            },
                            {
                                "type": 0,
                                "name": "style",
                                "value": 1
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
```

```
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Consulte também

- Dashboard widget field
- `dashboard.create`
- `dashboard.update`

4 Status de descoberta

Descrição

Esses parâmetros e os possíveis valores de propriedade para os respectivos objetos de campo de widget de dashboard permitem configurar o widget *Discovery status* nos métodos `dashboard.create` e `dashboard.update`.

Parâmetros

Os seguintes parâmetros são suportados para o widget *Discovery status*.

Parâmetro	type	nome	valor
Intervalo de atualização	0	rf_rate	0 - Sem atualização; 10 - 10 segundos; 30 - 30 segundos; 60 - *(padrão)* 1 minuto; 120 - 2 minutos; 600 - 10 minutos; 900 - 15 minutos.

Exemplos

Os exemplos a seguir têm como objetivo apenas descrever a configuração dos objetos de campo do widget do painel para o widget *Discovery status*. Para obter mais informações sobre como configurar um painel, consulte `dashboard.create`.

Configurando o widget *Discovery status*

Configure um widget *Discovery status* com o intervalo de atualização definido para 15 minutos.

Requisição:

```
{
    "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
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Veja também

- Campo do widget do dashboard
- `dashboard.create`
- `dashboard.update`

5 Gráficos favoritos

Descrição

Esses parâmetros e os possíveis valores de propriedade para os respectivos objetos de campo de widget de dashboard permitem configurar o widget *Gráficos favoritos* nos métodos `dashboard.create` e `dashboard.update`.

Parâmetros

Os seguintes parâmetros são suportados para o widget *Gráficos favoritos*.

Parâmetro	type	nome	valor
Intervalo de atualização	0	rf_rate	0 - Sem atualização; 10 - 10 segundos; 30 - 30 segundos; 60 - 1 minuto; 120 - 2 minutos; 600 - 10 minutos; 900 - *(padrão)* 15 minutos.

Exemplos

Os exemplos a seguir têm como objetivo apenas descrever a configuração dos objetos de campo do widget do dashboard para o widget *Gráficos favoritos*. Para obter mais informações sobre como configurar um dashboard, consulte `dashboard.create`.

Configurando um widget *Gráficos favoritos*

Configure um widget *Gráficos favoritos* com o intervalo de atualização definido para 10 minutos.

Requisição:

```
{
    "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
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Veja também

- Campo do widget do dashboard
- `dashboard.create`
- `dashboard.update`

6 Mapas favoritos

Descrição

Esses parâmetros e os possíveis valores de propriedade para os respectivos objetos de campo de widget de dashboard permitem configurar o widget *Mapas favoritos* nos métodos `dashboard.create` e `dashboard.update`.

Parâmetros

Os seguintes parâmetros são suportados para o widget *Mapas favoritos*.

Parâmetro	tipo	nome	valor
Intervalo de atualização	0	rf_rate	0 - Sem atualização; 10 - 10 segundos; 30 - 30 segundos; 60 - 1 minuto; 120 - 2 minutos; 600 - 10 minutos; 900 - *(padrão)* 15 minutos.

Exemplos

Os exemplos a seguir têm como objetivo apenas descrever a configuração dos objetos de campo do widget do dashboard para o widget *Mapas favoritos*. Para obter mais informações sobre como configurar um dashboard, consulte `dashboard.create`.

Configurando um widget *Mapas favoritos*

Configure um widget *Mapas favoritos* com o intervalo de atualização definido para 10 minutos.

Requisição:

```
{
    "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
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Veja também

- Campo do widget do dashboard
- `dashboard.create`
- `dashboard.update`

7 Medidor

Descrição

Esses parâmetros e os possíveis valores de propriedade para os respectivos objetos de campo do widget de dashboard permitem configurar o widget *Gauge* nos métodos `dashboard.create` e `dashboard.update`.

Attention:
As propriedades `fields` do widget não são validadas durante a criação ou atualização de um dashboard. Isso permite que os usuários modifiquem widgets internos e criem widgets personalizados, mas também introduz o risco de criar ou atualizar widgets incorretamente. Para garantir a criação ou atualização bem-sucedida do widget *Gauge*, consulte o comportamento dos parâmetros descritos nas tabelas abaixo.

Parâmetros

Os seguintes parâmetros são suportados para o widget *Gauge*.

Parâmetro	tipo	nome	valor
Intervalo de atualização	0	rf_rate	0 - Sem atualização; 10 - 10 segundos; 30 - 30 segundos; 60 - *(padrão)* 1 minuto; 120 - 2 minutos; 600 - 10 minutos; 900 - 15 minutos.
Item	4	itemid.0	ID do item.
Item (Widget)	1	itemid._reference	Comportamento do parâmetro: - *obrigatório* se *Item (Widget)* não estiver definido Em vez do ID do item: `ABCDE._itemid` - defina um widget compatível (com seu parâmetro *Referência* definido como "ABCDE") como fonte de dados para os items.
Mín	1	min	Comportamento do parâmetro: - *obrigatório* se *Item* não estiver definido Qualquer valor numérico. Sufixos (por exemplo, "1d", "2w", "4K", "8G") são suportados. Padrão: "0".
Máx	1	max	Qualquer valor numérico. Sufixos (por exemplo, "1d", "2w", "4K", "8G") são suportados. Padrão: "100".
Arco de valor	1	value_arc_color	Código de cor hexadecimal (por exemplo, `FF0000`). Padrão: `""` (vazio).
Fundo do arco	1	empty_color	Código de cor hexadecimal (por exemplo, `FF0000`). Padrão: `""` (vazio).
Fundo	1	bg_color	Código de cor hexadecimal (por exemplo, `FF0000`). Padrão: `""` (vazio).
Mostrar	0	show.0	1 - Descrição; 2 - Valor; 3 - Ponteiro; 4 - Escala; 5 - Arco de valor. Observação: Para configurar vários valores, crie um objeto de campo de widget de dashboard para cada valor com um número incrementado no nome da propriedade. Padrão: 1, 2, 4, 5. Os valores "Ponteiro" e "Escala" não são suportados se ambos: - nenhum objeto de campo de widget de dashboard para *Mostrar* com o valor "Arco de valor" estiver definido; - o parâmetro de configuração avançada *Mostrar arco* estiver definido como "Desabilitado". Os parâmetros de configuração avançada para opções de *Mostrar* não são suportados se nenhum objeto de campo de widget de dashboard com os respectivos valores estiver definido.

Parâmetro	tipo	nome	valor
Substituir host	1	override_hostid._referência	`ABCDE._hostid` - defina um widget compatível (com seu parâmetro *Referência* definido como "`ABCDE`") como fonte de dados para hosts; `DASHBOARD._hostid` - defina o seletor de *Host* do dashboard como fonte de dados para hosts. Este parâmetro não é suportado se estiver configurando o widget em um dashboard de template.

Configuração avançada

Os seguintes parâmetros de configuração avançada são suportados para o widget *Gauge*.

Note:
O número no nome da propriedade *Thresholds* (por exemplo, thresholds.0.color) faz referência à posição do threshold em uma lista, ordenada em ordem crescente. No entanto, se os thresholds forem configurados em uma ordem diferente, os valores serão ordenados em ordem crescente após a atualização da configuração do widget no frontend do Zabbix (por exemplo, `"thresholds.0.threshold":"5"` → `"thresholds.0.threshold":"1"`; `"thresholds.1.threshold":"1"` → `"thresholds.1.threshold": "5"`).

Parâmetro	tipo	nome	valor
Ângulo	0	angle	Valores possíveis: 180 *(padrão)* ou 270.
Descrição			
Descrição	1	description	Qualquer valor de string, incluindo macros. Macros suportadas: {HOST.*}, {ITEM.*}, {INVENTORY.*}, macros de usuário. Padrão: {ITEM.NAME}.
Tamanho	0	desc_size	Valores possíveis variam de 1-100. Padrão: 15.
Posição vertical	0	desc_v_pos	0 - Superior; 1 - *(padrão)* Inferior.
Negrito	0	desc_bold	0 - *(padrão)* Desabilitado; 1 - Habilitado.
Cor	1	desc_color	Código de cor hexadecimal (por exemplo, `FF0000`). Padrão: `""` (vazio).
Valor			
Casas decimais	0	decimal_places	Valores possíveis variam de 1-10. Padrão: 2.
Tamanho	0	value_size	Valores possíveis variam de 1-100. Padrão: 25.
Negrito	0	value_bold	0 - *(padrão)* Desabilitado; 1 - Habilitado.
Cor	1	value_color	Código de cor hexadecimal (por exemplo, `FF0000`). Padrão: `""` (vazio).
Unidades			
Unidades (checkbox)	0	units_show	0 - Desabilitado; 1 - *(padrão)* Habilitado.
Unidades (valor)	1	units	Qualquer valor de string. Comportamento do parâmetro: - *suportado* se *Unidades* (checkbox) estiver definido como "Habilitado"

Parâmetro	tipo	nome	valor
Tamanho	0	units_size	Valores possíveis variam de 1-100. Padrão: 25. Comportamento do parâmetro: - *suportado* se *Unidades* (checkbox) estiver definido como ”Habilitado”
Negrito	0	units_bold	0 - *(padrão)* Desabilitado; 1 - Habilitado. Comportamento do parâmetro: - *suportado* se *Unidades* (checkbox) estiver definido como ”Habilitado”
Posição	0	units_pos	0 - Antes do valor; 1 - Acima do valor; 2 - *(padrão)* Após o valor; 3 - Abaixo do valor. Comportamento do parâmetro: - *suportado* se *Unidades* (checkbox) estiver definido como ”Habilitado” Este parâmetro é ignorado se definido para uma das seguintes unidades relacionadas ao tempo: unixtime, uptime, s.
Cor	1	units_color	Código de cor hexadecimal (por exemplo, FF0000). Padrão: "" (vazio).
Arco do valor			
Tamanho do arco	0	value_arc_size	Valores possíveis variam de 1-100. Padrão: 20.
Agulha			
Cor	1	needle_color	Código de cor hexadecimal (por exemplo, FF0000). Padrão: "" (vazio). Comportamento do parâmetro: - *suportado* se um objeto de campo de widget de dashboard para *Mostrar* com o valor ”Arco do valor” estiver definido, ou *Mostrar arco* estiver definido como ”Habilitado”
Escala			
Mostrar unidades	0	scale_show_units	0 - Desabilitado; 1 - *(padrão)* Habilitado. Comportamento do parâmetro: - *suportado* se *Unidades* (checkbox) estiver definido como ”Habilitado” e um objeto de campo de widget de dashboard para *Mostrar* com o valor ”Arco do valor” estiver definido, ou *Mostrar arco* estiver definido como ”Habilitado”
Tamanho	0	scale_size	Valores possíveis variam de 1-100. Padrão: 15. Comportamento do parâmetro: - *suportado* se um objeto de campo de widget de dashboard para *Mostrar* com o valor ”Arco do valor” estiver definido, ou *Mostrar arco* estiver definido como ”Habilitado”
Casas decimais	0	scale_decimal_places	Valores possíveis variam de 1-10. Padrão: 0. Comportamento do parâmetro: - *suportado* se um objeto de campo de widget de dashboard para *Mostrar* com o valor ”Arco do valor” estiver definido, ou *Mostrar arco* estiver definido como ”Habilitado”

Parâmetro	tipo	nome	valor
Thresholds			
Cor	1	thresholds.0.color	Código de cor hexadecimal (por exemplo, FF0000).
Threshold	1	thresholds.0.threshold	Qualquer valor numérico. Sufixos (por exemplo, ”1d”, ”2w”, ”4K”, ”8G”) são suportados.
Mostrar rótulos	0	th_show_labels	0 - *(padrão)* Desabilitado; 1 - Habilitado. Comportamento do parâmetro: - *suportado* se *Thresholds* estiverem definidos e um objeto de campo de widget de dashboard para *Mostrar* com o valor ”Arco do valor” estiver definido ou *Mostrar arco* estiver definido como ”Habilitado”
Mostrar arco	0	th_show_arc	0 - *(padrão)* Desabilitado; 1 - Habilitado. Comportamento do parâmetro: - *suportado* se *Thresholds* estiverem definidos
Tamanho do arco	0	th_arc_size	Valores possíveis variam de 1-100. Padrão: 5. Comportamento do parâmetro: - *suportado* se *Mostrar arco* estiver definido como ”Habilitado”

Exemplos

Os exemplos a seguir têm como objetivo apenas descrever a configuração dos objetos de campo do widget do dashboard para o widget *Gauge*. Para obter mais informações sobre como configurar um dashboard, consulte `dashboard.create`.

Configurando um widget *Gauge*

Configure um widget *Gauge* que exibe o valor do item para o item ”44474” (Interface enp0s3: Bits enviados). Além disso, ajuste visualmente o widget com várias opções avançadas, incluindo limites.

Requisição:

```
{
    "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
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Veja também

- Campo do widget do dashboard
- `dashboard.create`
- `dashboard.update`

8 Geomapa

Descrição

Esses parâmetros e os possíveis valores de propriedade para os respectivos objetos de campo do widget do dashboard permitem configurar o widget *Geomap* nos métodos `dashboard.create` e `dashboard.update`.

Attention:
As propriedades `fields` do widget não são validadas durante a criação ou atualização de um dashboard. Isso permite que os usuários modifiquem widgets nativos e criem widgets personalizados, mas também introduz o risco de criar ou atualizar widgets incorretamente. Para garantir a criação ou atualização bem-sucedida do widget *Geomap*, consulte o comportamento dos parâmetros descritos nas tabelas abaixo.

Parâmetros

Os seguintes parâmetros são suportados para o widget *Geomap*.

Parâmetro	type	name	value
Intervalo de atualização	0	rf_rate	0 - Sem atualização; 10 - 10 segundos; 30 - 30 segundos; 60 - *(padrão)* 1 minuto; 120 - 2 minutos; 600 - 10 minutos; 900 - 15 minutos.

Parâmetro	type	name	value
Grupos de hosts	2	groupids.0	ID do Grupo de hosts. Nota: Para configurar múltiplos grupos de hosts, crie um objeto de campo de widget de dashboard para cada grupo de hosts com um número incrementado no nome da propriedade. Este parâmetro não é suportado se o widget estiver sendo configurado em um dashboard de template.
Grupos de hosts (Widget)	1	groupids._reference	Em vez do ID do Grupo de hosts: `ABCDE._hostgroupids` - defina um widget compatível (com seu parâmetro *Referência* definido como "`ABCDE`") como a fonte de dados para grupos de hosts. Este parâmetro não é suportado se o widget estiver sendo configurado em um dashboard de template.
Hosts	3	hostids.0	ID do Host. Nota: Para configurar múltiplos hosts, crie um objeto de campo de widget de dashboard para cada host com um número incrementado no nome da propriedade. Para múltiplos hosts, o parâmetro *Grupos de hosts* deve não estar configurado ou estar configurado com pelo menos um grupo de hosts ao qual os hosts configurados pertençam. Este parâmetro não é suportado se o widget estiver sendo configurado em um dashboard de template.
Hosts (Widget/Dashboard)	1	hostids._reference	Em vez do ID do Host: `DASHBOARD.hostids` - defina o seletor *Host* como a fonte de dados para hosts; `ABCDE._hostids` - defina um widget compatível (com seu parâmetro *Referência* definido como "`ABCDE`") como a fonte de dados para hosts. Este parâmetro não é suportado se o widget estiver sendo configurado em um dashboard de template.
Tags			
Tipo de avaliação	0	evaltype	0 - *(padrão)* E/Ou; 2 - Ou. Este parâmetro não é suportado se o widget estiver sendo configurado em um dashboard de template.
Nome da tag	1	tags.0.tag	Qualquer valor de string. Nota: O número no nome da propriedade faz referência à ordem da tag na lista de avaliação de tags. Comportamento do parâmetro: - *obrigatório* se estiver configurando *Tags* Este parâmetro não é suportado se o widget estiver sendo configurado em um dashboard de template.

Parâmetro	type	name	value
Operador	0	tags.0.operator	0 - Contém; 1 - Igual; 2 - Não contém; 3 - Diferente; 4 - Existe; 5 - Não existe. Nota: O número no nome da propriedade faz referência à ordem da tag na lista de avaliação de tags. Comportamento do parâmetro: - *obrigatório* se estiver configurando *Tags*
Valor da tag	1	tags.0.value	Este parâmetro não é suportado se o widget estiver sendo configurado em um dashboard de template. Qualquer valor de string. Nota: O número no nome da propriedade faz referência à ordem da tag na lista de avaliação de tags. Comportamento do parâmetro: - *obrigatório* se estiver configurando *Tags* Este parâmetro não é suportado se o widget estiver sendo configurado em um dashboard de template.
Visualização inicial	1	default_view	*latitude*, *longitude*, *nível de zoom* (*opcional*, valores possíveis variam de 0-30) separados por vírgula. Exemplo: `40.6892494,-74.0466891,10`.
Referência	1	reference	Qualquer valor de string consistindo em 5 caracteres (por exemplo, `ABCDE` ou `JBPNL`). Este valor deve ser único dentro do dashboard ao qual o widget pertence. Comportamento do parâmetro: - *obrigatório*

Exemplos

Os exemplos a seguir têm como objetivo apenas descrever a configuração dos objetos de campo do widget do painel para o widget *Geomap*. Para obter mais informações sobre como configurar um painel, consulte `dashboard.create`.

Configurando um widget *Geomap*

Configure um widget *Geomap* que exibe hosts dos grupos de hosts ”2” e ”22” com base na seguinte configuração de tag: a tag com o nome ”component” contém o valor ”node”, e a tag com o nome ”location” é igual ao valor ”New York”. Além disso, defina a visualização inicial do mapa para as coordenadas ”40.6892494” (latitude), ”-74.0466891” (longitude) com o nível de zoom ”10”.

Requisição:

```
{
    "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
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Veja também

- Campo de widget de dashboard
- `dashboard.create`
- `dashboard.update`

9 Gráfico

Descrição

Esses parâmetros e os possíveis valores de propriedade para os respectivos objetos de campo do widget de dashboard permitem configurar o widget *Gráfico* nos métodos `dashboard.create` e `dashboard.update`.

Attention:
As propriedades `fields` do widget não são validadas durante a criação ou atualização de um dashboard. Isso permite que os usuários modifiquem widgets internos e criem widgets personalizados, mas também introduz o risco de criar ou atualizar widgets incorretamente. Para garantir a criação ou atualização bem-sucedida do widget *Gráfico*, consulte o comportamento dos parâmetros descrito nas tabelas abaixo.

Parâmetros

Os seguintes parâmetros são suportados para o widget *Gráfico*.

Parâmetro	tipo	nome	valor
Intervalo de atualização	0	rf_rate	0 - Sem atualização; 10 - 10 segundos; 30 - 30 segundos; 60 - *(padrão)* 1 minuto; 120 - 2 minutos; 600 - 10 minutos; 900 - 15 minutos.
Referência	1	reference	Qualquer valor de string consistindo de 5 caracteres (por exemplo, `ABCDE` ou `JBPNL`). Este valor deve ser único dentro do dashboard ao qual o widget pertence. Comportamento do parâmetro: - *obrigatório*

Conjunto de dados

Os seguintes parâmetros são suportados para configurar um *Conjunto de dados*.

Note:
O primeiro número no nome da propriedade (por exemplo, ds.0.hosts.0, ds.0.items.0) representa o conjunto de dados específico, enquanto o segundo número, se presente, representa o host ou item configurado.

Parâmetro	tipo	nome	valor
Tipo de conjunto de dados	0	ds.0.dataset_type	0 - Lista de items; 1 - *(padrão)* Padrões de items.
Items	4	ds.0.itemids.0	ID do Item. Ao configurar o widget em um dashboard de template, apenas os items configurados no template devem ser definidos. Nota: Para configurar múltiplos items, crie um objeto de campo de widget de dashboard para cada item com um número incrementado no nome da propriedade. Comportamento do parâmetro: - *obrigatório* se *Tipo de conjunto de dados* estiver definido como "Lista de items" e *Items (Widget)* não estiver definido
Items (Widget)	1	ds.0.itemids.0._reference	Em vez do ID do Item: `ABCDE._itemid` - defina um widget compatível (com seu parâmetro *Referência* definido como "ABCDE") como a fonte de dados para os items. Nota: Para configurar múltiplos widgets, crie um objeto de campo de widget de dashboard para cada widget com um número incrementado no nome da propriedade. Este parâmetro é suportado desde o Zabbix 7.0.1. Comportamento do parâmetro: - *obrigatório* se *Tipo de conjunto de dados* estiver definido como "Lista de items" e *Items* não estiver definido
Cor	1	ds.0.color.0	Código hexadecimal da cor (por exemplo, `FF0000`). Comportamento do parâmetro: - *obrigatório* se *Tipo de conjunto de dados* estiver definido como "Lista de items"
Padrões de host	1	ds.0.hosts.0	Nome ou padrão do Host (por exemplo, "Zabbix*"). Comportamento do parâmetro: - *obrigatório* se *Tipo de conjunto de dados* estiver definido como "Padrões de items" Este parâmetro não é suportado ao configurar o widget em um dashboard de template.

Parâmetro	tipo	nome	valor
Padrões de items	1	ds.0.items.0	Nome ou padrão do Item (por exemplo, ”*: Número de valores processados por segundo”). Ao configurar o widget em um dashboard de template, apenas os padrões para items configurados no template devem ser definidos. Comportamento do parâmetro: - *obrigatório* se *Tipo de conjunto de dados* estiver definido como ”Padrões de items”
Cor	1	ds.0.color	Código hexadecimal da cor (por exemplo, FF0000). Padrão: FF465C. Comportamento do parâmetro: - *suportado* se *Tipo de conjunto de dados* estiver definido como ”Padrões de items”
Desenhar	0	ds.0.type	0 - *(padrão)* Linha; 1 - Pontos; 2 - Escada; 3 - Barra.
Empilhado	0	ds.0.stacked	0 - *(padrão)* Desabilitado; 1 - Habilitado. Comportamento do parâmetro: - *suportado* se *Desenhar* estiver definido como ”Linha”, ”Escada” ou ”Barra”
Largura	0	ds.0.width	Os valores possíveis variam de 1-10. Padrão: 1. Comportamento do parâmetro: - *suportado* se *Desenhar* estiver definido como ”Linha” ou ”Escada”
Tamanho do ponto	0	ds.0.pointsize	Os valores possíveis variam de 1-10. Padrão: 3. Comportamento do parâmetro: - *suportado* se *Desenhar* estiver definido como ”Pontos”
Transparência	0	ds.0.transparency	Os valores possíveis variam de 1-10. Padrão: 5.
Preenchimento	0	ds.0.fill	Os valores possíveis variam de 1-10. Padrão: 3. Comportamento do parâmetro: - *suportado* se *Desenhar* estiver definido como ”Linha” ou ”Escada”
Dados ausentes	0	ds.0.missingdatafunc	0 - *(padrão)* Nenhum; 1 - Conectado; 2 - Tratar como 0; 3 - Último conhecido. Comportamento do parâmetro: - *suportado* se *Desenhar* estiver definido como ”Linha” ou ”Escada”
Eixo Y	0	ds.0.axisy	0 - *(padrão)* Esquerda; 1 - Direita.
Deslocamento de tempo	1	ds.0.timeshift	String de tempo válida (por exemplo, 3600, 1h, etc.). Você pode usar sufixos de tempo. Valores negativos também são permitidos. Padrão: "" (vazio).

Parâmetro	tipo	nome	valor
Função de agregação	0	ds.0.aggregate_function	0 - *(padrão)* não usado; 1 - min; 2 - max; 3 - avg; 4 - count; 5 - sum; 6 - first; 7 - last.
Intervalo de agregação	1	ds.0.aggregate_interval	String de tempo válida (por exemplo, 3600, 1h, etc.). Você pode usar sufixos de tempo. Padrão: 1h.
Agregado	0	ds.0.aggregate_grouping	0 - *(padrão)* Cada item; 1 - Conjunto de dados. Comportamento do parâmetro: - *suportado* se *Função de agregação* estiver definido como ”min”, ”max”, ”avg”, ”count”, ”sum”, ”first” ou ”last”
Aproximação	0	ds.0.approximation	1 - min; 2 - *(padrão)* avg; 4 - max; 7 - all.
Rótulo do conjunto de dados	1	ds.0.data_set_label	Qualquer valor de string. Padrão: "" (vazio).

Opções de exibição

Os seguintes parâmetros são suportados para configurar as *Opções de exibição*.

Parâmetro	tipo	nome	valor
Seleção de dados históricos	0	source	0 - *(padrão)* Automático; 1 - Histórico; 2 - Tendências.
Triggers simples	0	simple_triggers	0 - *(padrão)* Desabilitado; 1 - Habilitado.
Horário de trabalho	0	working_time	0 - *(padrão)* Desabilitado; 1 - Habilitado.
Linha de percentil (esquerda)			
Status	0	percentile_left	0 - *(padrão)* Desabilitado; 1 - Habilitado. Comportamento do parâmetro: - *suportado* se o *Eixo Y* (na configuração do *Conjunto de dados*) estiver definido como ”Esquerda”

Parâmetro	tipo	nome	valor
Valor	0	percentile_left_value	Os valores possíveis variam de 1-100. Comportamento do parâmetro: - *suportado* se o *Eixo Y* (na configuração do *Conjunto de dados*) estiver definido como "Esquerda"
Linha de per-centil (di-reita)			
Status	0	percentile_right	0 - *(padrão)* Desabilitado; 1 - Habilitado. Comportamento do parâmetro: - *suportado* se o *Eixo Y* (na configuração do *Conjunto de dados*) estiver definido como "Direita"
Valor	0	percentile_right_value	Os valores possíveis variam de 1-100. Comportamento do parâmetro: - *suportado* se o *Eixo Y* (na configuração do *Conjunto de dados*) estiver definido como "Direita"

Período de tempo

Os seguintes parâmetros são suportados para configurar o *Período de tempo*.

Parâmetro	tipo	nome	valor
Período de tempo	1	time_period._referenceA	`DASHBOARD._timeperiod` - define o seletor *Período de tempo* como a fonte de dados; `ABCDE._timeperiod` - define um widget compatível (com seu parâmetro *Referência* definido como "`ABCDE`") como a fonte de dados. Padrão: `DASHBOARD._timeperiod` Alternativamente, você pode definir o período de tempo apenas nos parâmetros *De* e *Até*.
De	1	time_period.from	String de tempo válida na sintaxe absoluta (`YYYY-MM-DD hh:mm:ss`) ou relativa (`now`, `now/d`, `now/w-1w`, etc.). Comportamento do parâmetro: - *suportado* se *Período de tempo* não estiver definido - *obrigatório* se `time_period.to` estiver definido
Até	1	time_period.to	String de tempo válida na sintaxe absoluta (`YYYY-MM-DD hh:mm:ss`) ou relativa (`now`, `now/d`, `now/w-1w`, etc.). Comportamento do parâmetro: - *suportado* se *Período de tempo* não estiver definido - *obrigatório* se `time_period.from` estiver definido

Eixos

Os seguintes parâmetros são suportados para configurar *Eixos*.

Parâmetro	type	name	value
Y Esquerdo	0	lefty	0 - Desabilitado; 1 - *(padrão)* Habilitado. Comportamento do parâmetro: - *suportado* se o *Eixo Y* (na configuração do *Conjunto de dados*) estiver definido como ”Esquerdo”
Y Direito	0	righty	0 - *(padrão)* Desabilitado; 1 - Habilitado. Comportamento do parâmetro: - *suportado* se o *Eixo Y* (na configuração do *Conjunto de dados*) estiver definido como ”Direito”
Mín	1	lefty_min righty_min	Qualquer valor numérico. Padrão: "" (vazio).
Máx	1	lefty_max righty_max	Qualquer valor numérico. Padrão: "" (vazio).
Unidades (tipo)	0	lefty_units righty_units	0 - *(padrão)* Automático; 1 - Estático.
Unidades (valor)	1	lefty_static_units righty_static_units	Qualquer valor de string. Padrão: "" (vazio).
Eixo X	0	xaxis	0 - Desabilitado; 1 - *(padrão)* Habilitado.

Legenda

Os seguintes parâmetros são suportados para configurar a *Legenda*.

Parâmetro	tipo	nome	valor
Mostrar legenda	0	legend	0 - Desabilitado; 1 - *(padrão)* Habilitado.
Exibir min/méd/max	0	legend_statistic	Comportamento do parâmetro: - *suportado* se *Mostrar legenda* estiver definido como ”Habilitado” 0 - *(padrão)* Desabilitado; 1 - Habilitado.
Mostrar função de agregação	0	legend_aggregation	Comportamento do parâmetro: - *suportado* se *Mostrar legenda* estiver definido como ”Habilitado” 0 - *(padrão)* Desabilitado; 1 - Habilitado.
Linhas	0	legend_lines_mode	Comportamento do parâmetro: - *suportado* se *Mostrar legenda* estiver definido como ”Habilitado” 0 - *(padrão)* Fixo; 1 - Variável. Comportamento do parâmetro: - *suportado* se *Mostrar legenda* estiver definido como ”Habilitado”

Parâmetro	tipo	nome	valor
Número de linhas/ Número máximo de linhas	0	legend_lines	Os valores possíveis variam de 1 a 10. Padrão: 1. Comportamento do parâmetro: - *suportado* se *Mostrar legenda* estiver definido como ”Habilitado”
Número de colunas	0	legend_columns	Os valores possíveis variam de 1 a 4. Padrão: 4. Comportamento do parâmetro: - *suportado* se *Mostrar legenda* estiver definido como ”Habilitado” e *Exibir min/méd/max* estiver definido como ”Desabilitado”

Problemas

Os seguintes parâmetros são suportados para configurar *Problemas*.

Parâmetro	tipo	nome	valor
Mostrar problemas	0	show_problems	0 - *(padrão)* Desabilitado; 1 - Habilitado.
Apenas itens selecionados	0	graph_item_problems	0 - Desabilitado; 1 - *(padrão)* Habilitado.
Hosts de problema	1	problemhosts.0	Nome do host. Nota: O número no nome da propriedade faz referência ao host configurado. Para configurar vários hosts, crie um objeto de campo de widget de dashboard para cada host com um número incrementado no nome da propriedade. Este parâmetro não é suportado se estiver configurando o widget em um dashboard de template.
Gravidade	0	severities.0	0 - Não classificado; 1 - Informação; 2 - Aviso; 3 - Médio; 4 - Alto; 5 - Desastre. Padrão: vazio (todos habilitados). Nota: Para configurar vários valores, crie um objeto de campo de widget de dashboard para cada valor com um número incrementado no nome da propriedade.
Problema	1	problem_name	Nome do evento do problema (não diferencia maiúsculas de minúsculas, nome completo ou parte dele).
Tags de problema			
Tipo de avaliação	0	evaltype	0 - *(padrão)* E/Ou; 2 - Ou.

Parâmetro	tipo	nome	valor
Nome da tag	1	tags.0.tag	Qualquer valor de string. Nota: O número no nome da propriedade faz referência à ordem da tag na lista de avaliação de tags. Comportamento do parâmetro: - *obrigatório* se estiver configurando *Tags de problema*
Operador	0	tags.0.operator	0 - Contém; 1 - Igual; 2 - Não contém; 3 - Diferente; 4 - Existe; 5 - Não existe. Nota: O número no nome da propriedade faz referência à ordem da tag na lista de avaliação de tags. Comportamento do parâmetro: - *obrigatório* se estiver configurando *Tags de problema*
Valor da tag	1	tags.0.value	Qualquer valor de string. Nota: O número no nome da propriedade faz referência à ordem da tag na lista de avaliação de tags. Comportamento do parâmetro: - *obrigatório* se estiver configurando *Tags de problema*

Substituições

Os seguintes parâmetros são suportados para configurar *Substituições*.

Note:
O primeiro número no nome da propriedade (por exemplo, or.0.hosts.0, or.0.items.0) representa o conjunto de dados específico, enquanto o segundo número, se presente, representa o host ou item configurado.

Parâmetro	tipo	nome	valor
Padrões de host	1	or.0.hosts.0	Nome ou padrão do host (por exemplo, `Zabbix*`). Este parâmetro não é suportado se o widget estiver sendo configurado em um dashboard de template. Comportamento do parâmetro: - *obrigatório* se estiver configurando *Substituições*
Padrões de item	1	or.0.items.0	Nome ou padrão do item (por exemplo, `*: Number of processed *values per second`). Ao configurar o widget em um dashboard de template, apenas os padrões para os itens configurados no template devem ser definidos. Comportamento do parâmetro: - *obrigatório* se estiver configurando *Substituições*
Cor base	1	or.0.color	Código de cor hexadecimal (por exemplo, `FF0000`).
Largura	0	or.0.width	Os valores possíveis variam de 1-10.
Desenhar	0	or.0.type	0 - Linha; 1 - Pontos; 2 - Degrau; 3 - Barra.
Transparência	0	or.0.transparency	Os valores possíveis variam de 1-10.
Preenchimento	0	or.0.fill	Os valores possíveis variam de 1-10.
Tamanho do ponto	0	or.0.pointsize	Os valores possíveis variam de 1-10.

Parâmetro	tipo	nome	valor
Dados ausentes	0	or.0.missingdatafunc	0 - Nenhum; 1 - Conectado; 2 - Tratar como 0; 3 - Último conhecido.
Eixo Y	0	or.0.axisy	0 - Esquerda; 1 - Direita.
Deslocamento de tempo	1	or.0.timeshift	String de tempo válida (por exemplo, 3600, 1h, etc.). Você pode usar sufixos de tempo. Valores negativos são permitidos.

Exemplos

Os exemplos a seguir têm como objetivo apenas descrever a configuração dos objetos de campo do widget do dashboard para o widget *Gráfico*. Para obter mais informações sobre como configurar um dashboard, consulte `dashboard.create`.

Configurando um widget *Graph*

Configure um widget *Graph* da seguinte forma:

- 2 conjuntos de dados para um total de 9 items em 1 host.
- O primeiro conjunto de dados é do tipo ”Item list” e consiste em 3 items que são representados por linhas com cores diferentes, mas com a mesma largura, transparência e preenchimento.
- O segundo conjunto de dados é do tipo ”Item patterns”, consiste em 6 items, possui uma agregação configurada e é representado por uma linha com cor, largura, transparência e preenchimento personalizados.
- O segundo conjunto de dados também possui um rótulo de conjunto de dados personalizado.
- Os dados no gráfico são exibidos para um período de tempo das últimas 3 horas.
- Os problemas no gráfico são exibidos apenas para os items configurados.
- O gráfico possui dois eixos Y, dos quais o eixo Y direito exibe valores apenas para o segundo conjunto de dados.
- A legenda do gráfico exibe os items configurados em 4 linhas, bem como os valores mínimo, máximo e médio dos conjuntos de dados.

Request:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "svggraph",
                        "name": "Graph",
                        "x": 0,
                        "y": 0,
                        "width": 36,
                        "height": 5,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 0,
                                "name": "ds.0.dataset_type",
                                "value": 0
                            },
                            {
                                "type": 4,
                                "name": "ds.0.itemids.0",
                                "value": 23264
                            },
                            {
                                "type": 1,
                                "name": "ds.0.color.0",
```

```
                "value": "FF0000"
            },
            {
                "type": 4,
                "name": "ds.0.itemids.1",
                "value": 23269
            },
            {
                "type": 1,
                "name": "ds.0.color.1",
                "value": "BF00FF"
            },
            {
                "type": 4,
                "name": "ds.0.itemids.2",
                "value": 23257
            },
            {
                "type": 1,
                "name": "ds.0.color.2",
                "value": "0040FF"
            },
            {
                "type": 0,
                "name": "ds.0.width",
                "value": 3
            },
            {
                "type": 0,
                "name": "ds.0.transparency",
                "value": 3
            },
            {
                "type": 0,
                "name": "ds.0.fill",
                "value": 1
            },
            {
                "type": 1,
                "name": "ds.1.hosts.0",
                "value": "Zabbix server"
            },
            {
                "type": 1,
                "name": "ds.1.items.0",
                "value": "*: Number of processed *values per second"
            },
            {
                "type": 1,
                "name": "ds.1.color",
                "value": "000000"
            },
            {
                "type": 0,
                "name": "ds.1.transparency",
                "value": 0
            },
            {
                "type": 0,
                "name": "ds.1.fill",
                "value": 0
            },
```

```
                    {
                        "type": 0,
                        "name": "ds.1.axisy",
                        "value": 1
                    },
                    {
                        "type": 0,
                        "name": "ds.1.aggregate_function",
                        "value": 3
                    },
                    {
                        "type": 1,
                        "name": "ds.1.aggregate_interval",
                        "value": "1m"
                    },
                    {
                        "type": 0,
                        "name": "ds.1.aggregate_grouping",
                        "value": 1
                    },
                    {
                        "type": 1,
                        "name": "ds.1.data_set_label",
                        "value": "Number of processed values per second"
                    },
                    {
                        "type": 0,
                        "name": "graph_time",
                        "value": 1
                    },
                    {
                        "type": 1,
                        "name": "time_period.from",
                        "value": "now-3h"
                    },
                    {
                        "type": 1,
                        "name": "time_period.to",
                        "value": "now"
                    },
                    {
                        "type": 0,
                        "name": "legend_statistic",
                        "value": 1
                    },
                    {
                        "type": 0,
                        "name": "legend_lines",
                        "value": 4
                    },
                    {
                        "type": 0,
                        "name": "show_problems",
                        "value": 1
                    },
                    {
                        "type": 1,
                        "name": "reference",
                        "value": "YZABC"
                    }
                ]
            }
```

```
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Veja também

- Campo do widget do dashboard
- `dashboard.create`
- `dashboard.update`

10 Gráfico (clássico)

Descrição

Esses parâmetros e os possíveis valores de propriedade para os respectivos objetos de campo do widget do dashboard permitem configurar o widget *Gráfico (clássico)* nos métodos `dashboard.create` e `dashboard.update`.

Attention:
As propriedades `fields` do widget não são validadas durante a criação ou atualização de um dashboard. Isso permite que os usuários modifiquem widgets internos e criem widgets personalizados, mas também introduz o risco de criar ou atualizar widgets incorretamente. Para garantir a criação ou atualização bem-sucedida do widget *Gráfico (clássico)*, consulte o comportamento dos parâmetros descrito nas tabelas abaixo.

Parâmetros

Os seguintes parâmetros são suportados para o widget *Gráfico (clássico)*.

Parâmetro	type	name	value
Intervalo de atualização	0	rf_rate	0 - Sem atualização; 10 - 10 segundos; 30 - 30 segundos; 60 - *(padrão)* 1 minuto; 120 - 2 minutos; 600 - 10 minutos; 900 - 15 minutos.

Parâmetro	type	name	value
Fonte	0	source_type	0 - *(padrão)* Gráfico; 1 - Gráfico simples.
Gráfico	6	graphid.0	ID do Gráfico. Comportamento do parâmetro: - *obrigatório* se *Fonte* estiver definida como "Gráfico"
Gráfico (Widget)	1	graphid._reference	Em vez do ID do Gráfico: `ABCDE._graphid` - defina um widget compatível (com seu parâmetro *Referência* definido como "ABCDE") como a fonte de dados para gráficos. Comportamento do parâmetro: - *obrigatório* se *Fonte* estiver definida como "Gráfico simples" e *Gráfico* não estiver definido
Item	4	itemid.0	ID do Item. Comportamento do parâmetro: - *obrigatório* se *Fonte* estiver definida como "Gráfico simples" e *Item (Widget)* não estiver definido
Item (Widget)	1	itemid._reference	Em vez do ID do Item: `ABCDE._itemid` - defina um widget compatível (com seu parâmetro *Referência* definido como "ABCDE") como a fonte de dados para items. Comportamento do parâmetro: - *obrigatório* se *Fonte* estiver definida como "Gráfico simples" e *Item* não estiver definido
Período de tempo	1	time_period._reference	`DASHBOARD._timeperiod` - defina o seletor de *Período de tempo* como a fonte de dados; `ABCDE._timeperiod` - defina um widget compatível (com seu parâmetro *Referência* definido como "ABCDE") como a fonte de dados. Padrão: `DASHBOARD._timeperiod` Alternativamente, você pode definir o período de tempo apenas nos parâmetros *De* e *Até*.
De	1	time_period.from	String de tempo válida em sintaxe absoluta (`YYYY-MM-DD hh:mm:ss`) ou relativa (`now`, `now/d`, `now/w-1w`, etc.). Comportamento do parâmetro: - *suportado* se *Período de tempo* não estiver definido - *obrigatório* se `time_period.to` estiver definido
Até	1	time_period.to	String de tempo válida em sintaxe absoluta (`YYYY-MM-DD hh:mm:ss`) ou relativa (`now`, `now/d`, `now/w-1w`, etc.). Comportamento do parâmetro: - *suportado* se *Período de tempo* não estiver definido - *obrigatório* se `time_period.from` estiver definido
Exibir legenda	0	show_legend	0 - Desabilitado; 1 - *(padrão)* Habilitado.
Sobrescrever host	1	override_hostid._reference	`ABCDE._hostid` - defina um widget compatível (com seu parâmetro *Referência* definido como "ABCDE") como a fonte de dados para hosts; `DASHBOARD._hostid` - defina o seletor de *Host* do dashboard como a fonte de dados para hosts. Este parâmetro não é suportado ao configurar o widget em um dashboard de template.

Parâmetro	type	name	value
Referência	1	reference	Qualquer valor de string composto por 5 caracteres (por exemplo, `ABCDE` ou `JBPNL`). Este valor deve ser exclusivo dentro do dashboard ao qual o widget pertence. Comportamento do parâmetro: - *obrigatório*

Exemplos

Os exemplos a seguir têm como objetivo apenas descrever a configuração dos objetos de campo do widget de dashboard para o widget *Gráfico (clássico)*. Para obter mais informações sobre como configurar um dashboard, consulte `dashboard.create`.

Configurando um widget *Gráfico (clássico)*

Configure um widget *Gráfico (clássico)* que exibe um gráfico simples para o item "42269".

Requisição:

```
{
    "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
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Veja também

- Campo de widget de dashboard
- `dashboard.create`
- `dashboard.update`

11 Protótipo de gráfico

Descrição

Esses parâmetros e os possíveis valores de propriedade para os respectivos objetos de campo do widget do dashboard permitem configurar o widget *Protótipo de gráfico* nos métodos `dashboard.create` e `dashboard.update`.

Attention:
As propriedades `fields` do widget não são validadas durante a criação ou atualização de um dashboard. Isso permite que os usuários modifiquem widgets internos e criem widgets personalizados, mas também introduz o risco de criar ou atualizar widgets incorretamente. Para garantir a criação ou atualização bem-sucedida do widget *Protótipo de gráfico*, consulte o comportamento do parâmetro descrito nas tabelas abaixo.

Parâmetros

Os seguintes parâmetros são suportados para o widget *Protótipo de gráfico*.

Parâmetro	tipo	nome	valor
Intervalo de atualização	0	rf_rate	0 - Sem atualização; 10 - 10 segundos; 30 - 30 segundos; 60 - *(padrão)* 1 minuto; 120 - 2 minutos; 600 - 10 minutos; 900 - 15 minutos.
Fonte	0	source_type	2 - *(padrão)* Protótipo de gráfico; 3 - Protótipo de gráfico simples.
Protótipo de gráfico	7	graphid.0	ID do Protótipo de gráfico. Comportamento do parâmetro: - *obrigatório* se *Fonte* estiver definida como ”Protótipo de gráfico”

Parâmetro	tipo	nome	valor
Protótipo de item	5	itemid.0	ID do Protótipo de item. Comportamento do parâmetro: - *obrigatório* se *Fonte* estiver definida como "Protótipo de gráfico simples"
Período de tempo	1	time_period._referenceDASHBOARD	`DASHBOARD._timeperiod` - define o seletor de *Período de tempo* como fonte de dados; `ABCDE._timeperiod` - define um widget compatível (com seu parâmetro *Referência* definido como "ABCDE") como fonte de dados. Padrão: `DASHBOARD._timeperiod` Alternativamente, você pode definir o período de tempo apenas nos parâmetros *De* e *Até*.
De	1	time_period.from	String de tempo válida em sintaxe absoluta (`YYYY-MM-DD hh:mm:ss`) ou relativa (`now`, `now/d`, `now/w-1w`, etc.). Comportamento do parâmetro: - *suportado* se *Período de tempo* não estiver definido - *obrigatório* se `time_period.to` estiver definido
Até	1	time_period.to	String de tempo válida em sintaxe absoluta (`YYYY-MM-DD hh:mm:ss`) ou relativa (`now`, `now/d`, `now/w-1w`, etc.). Comportamento do parâmetro: - *suportado* se *Período de tempo* não estiver definido - *obrigatório* se `time_period.from` estiver definido
Exibir legenda	0	show_legend	0 - Desabilitado; 1 - *(padrão)* Habilitado.
Sobrescrever host	1	override_hostid._referenceABCDE	`ABCDE._hostid` - define um widget compatível (com seu parâmetro *Referência* definido como "ABCDE") como fonte de dados para hosts; `DASHBOARD._hostid` - define o seletor de *Host* do dashboard como fonte de dados para hosts. Este parâmetro não é suportado ao configurar o widget em um dashboard de template.
Colunas	0	columns	Os valores possíveis variam de 1-24. Padrão: 2.
Linhas	0	rows	Os valores possíveis variam de 1-16. Padrão: 1.
Referência	1	reference	Qualquer valor de string composto por 5 caracteres (por exemplo, `ABCDE` ou `JBPNL`). Este valor deve ser exclusivo dentro do dashboard ao qual o widget pertence. Comportamento do parâmetro: - *obrigatório*

Exemplos

Os exemplos a seguir têm como objetivo apenas descrever a configuração dos objetos de campo do widget do dashboard para o widget *Protótipo de gráfico*. Para obter mais informações sobre como configurar um dashboard, consulte `dashboard.create`.

Configurando um widget *Protótipo de gráfico*

Configure um widget *Protótipo de gráfico* que exibe uma grade de 3 gráficos (3 colunas, 1 linha) criados a partir de um protótipo de item (ID: "42316") por descoberta de baixo nível.

Requisição:

```
{
    "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
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Veja também

- Campo do widget do dashboard
- `dashboard.create`
- `dashboard.update`

12 Colmeia

Descrição

Esses parâmetros e os possíveis valores de propriedade para os respectivos objetos de campo do widget do dashboard permitem configurar o widget *Honeycomb* nos métodos `dashboard.create` e `dashboard.update`.

Attention:
As propriedades `fields` do widget não são validadas durante a criação ou atualização de um dashboard. Isso permite que os usuários modifiquem widgets internos e criem widgets personalizados, mas também introduz o risco de criar ou atualizar widgets incorretamente. Para garantir a criação ou atualização bem-sucedida do widget *Honeycomb*, consulte o comportamento dos parâmetros descritos nas tabelas abaixo.

Parâmetros

Os seguintes parâmetros são suportados para o widget *Honeycomb*.

Parâmetro	tipo	nome	valor
Intervalo de atualização	0	rf_rate	0 - Sem atualização; 10 - 10 segundos; 30 - 30 segundos; 60 - *(padrão)* 1 minuto; 120 - 2 minutos; 600 - 10 minutos; 900 - 15 minutos.
Grupos de hosts	2	groupids.0	ID do Grupo de hosts. Nota: Para configurar múltiplos grupos de hosts, crie um objeto de campo de widget de dashboard para cada grupo de hosts com um número incrementado no nome da propriedade. Este parâmetro não é suportado se o widget estiver sendo configurado em um dashboard de template.
Grupos de hosts (Widget)	1	groupids._reference	Em vez do ID do Grupo de hosts: `ABCDE._hostgroupids` - defina um widget compatível (com seu parâmetro *Referência* definido como "ABCDE") como a fonte de dados para grupos de hosts. Este parâmetro não é suportado se o widget estiver sendo configurado em um dashboard de template.

Parâmetro	tipo	nome	valor
Hosts	3	hostids.0	ID do Host. Nota: Para configurar múltiplos hosts, crie um objeto de campo de widget de dashboard para cada host com um número incrementado no nome da propriedade. Para múltiplos hosts, o parâmetro *Grupos de hosts* deve não estar configurado ou estar configurado com pelo menos um grupo de hosts ao qual os hosts configurados pertencem. Este parâmetro não é suportado se o widget estiver sendo configurado em um dashboard de template.
Hosts (Widget/Dashboard)	1	hostids._reference	Em vez do ID do Host: `DASHBOARD.hostids` - defina o seletor *Host* como a fonte de dados para hosts; `ABCDE._hostids` - defina um widget compatível (com seu parâmetro *Referência* definido como "`ABCDE`") como a fonte de dados para hosts. Este parâmetro não é suportado se o widget estiver sendo configurado em um dashboard de template.
Tags de host			
Tipo de avaliação	0	evaltype_host	0 - *(padrão)* E/Ou; 2 - Ou. Este parâmetro não é suportado se o widget estiver sendo configurado em um dashboard de template.
Nome da tag	1	host_tags.0.tag	Qualquer valor de string. Nota: O número no nome da propriedade faz referência à ordem da tag na lista de avaliação de tags. Comportamento do parâmetro: - *obrigatório* se estiver configurando *Tags de host* Este parâmetro não é suportado se o widget estiver sendo configurado em um dashboard de template.
Operador	0	host_tags.0.operator	0 - Contém; 1 - Igual; 2 - Não contém; 3 - Diferente; 4 - Existe; 5 - Não existe. Nota: O número no nome da propriedade faz referência à ordem da tag na lista de avaliação de tags. Comportamento do parâmetro: - *obrigatório* se estiver configurando *Tags de host* Este parâmetro não é suportado se o widget estiver sendo configurado em um dashboard de template.
Valor da tag	1	host_tags.0.value	Qualquer valor de string. Nota: O número no nome da propriedade faz referência à ordem da tag na lista de avaliação de tags. Comportamento do parâmetro: - *obrigatório* se estiver configurando *Tags de host* Este parâmetro não é suportado se o widget estiver sendo configurado em um dashboard de template.

Parâmetro	tipo	nome	valor
Padrões de item	1	items.0	Nome ou padrão do Item. Nota: Para configurar múltiplos padrões de item, crie um objeto de campo de widget de dashboard para cada padrão de item com um número incrementado no nome da propriedade. Comportamento do parâmetro: - *obrigatório*
Tags de item			
Tipo de avaliação	0	evaltype_item	0 - *(padrão)* E/Ou; 2 - Ou.
Nome da tag	1	item_tags.0.tag	Qualquer valor de string. Nota: O número no nome da propriedade faz referência à ordem da tag na lista de avaliação de tags. Comportamento do parâmetro: - *obrigatório* se estiver configurando *Tags de item*
Operador	0	item_tags.0.operator	0 - Contém; 1 - Igual; 2 - Não contém; 3 - Diferente; 4 - Existe; 5 - Não existe. Nota: O número no nome da propriedade faz referência à ordem da tag na lista de avaliação de tags. Comportamento do parâmetro: - *obrigatório* se estiver configurando *Tags de item*
Valor da tag	1	item_tags.0.value	Qualquer valor de string. Nota: O número no nome da propriedade faz referência à ordem da tag na lista de avaliação de tags. Comportamento do parâmetro: - *obrigatório* se estiver configurando *Tags de item*
Mostrar hosts em manutenção	0	maintenance	0 - *(padrão)* Desabilitado; 1 - Habilitado.
Mostrar	0	show.0	1 - Rótulo primário; 2 - Rótulo secundário. Nota: Para configurar múltiplos valores, crie um objeto de campo de widget de dashboard para cada valor com um número incrementado no nome da propriedade. Padrão: 1, 2.
Referência	1	reference	Qualquer valor de string composto por 5 caracteres (por exemplo, `ABCDE` ou `JBPNL`). Este valor deve ser único dentro do dashboard ao qual o widget pertence. Comportamento do parâmetro: - *obrigatório*

Configuração avançada

Os seguintes parâmetros de configuração avançada são suportados para o widget *Honeycomb*.

Note:
O número no nome da propriedade *Thresholds* (por exemplo, thresholds.0.color) faz referência à posição do limiar em uma lista, ordenada em ordem crescente. No entanto, se os limiares forem configurados em uma ordem diferente, os valores serão classificados em ordem crescente após a atualização da configuração do widget no frontend do Zabbix (por exemplo, `"thresholds.0.threshold":"5"` → `"thresholds.0.threshold":"1"`; `"thresholds.1.threshold":"1"` → `"thresholds.1.threshold": "5"`).

Parâmetro	tipo	nome	valor
Rótulo principal			
Tipo	0	primary_label_type	0 - *(padrão)* Texto; 1 - Valor.
Texto	1	primary_label	Qualquer valor de string, incluindo macros. Macros suportadas: {HOST.*}, {ITEM.*}, {INVENTORY.*}, macros de usuário. Padrão: {HOST.NAME} Comportamento do parâmetro: - *suportado* se *Tipo* estiver definido como "Texto"
Casas decimais	0	primary_label_decimal_places	Os valores possíveis variam de 0-6. Padrão: 2. Comportamento do parâmetro: - *suportado* se *Tipo* estiver definido como "Valor"
Tamanho (tipo)	0	primary_label_size_type	0 - *(padrão)* Automático; 1 - Personalizado.
Tamanho	0	primary_label_size	Os valores possíveis variam de 1-100. Padrão: 20. Comportamento do parâmetro: - *suportado* se *Tamanho* (tipo) estiver definido como "Personalizado"
Negrito	0	primary_label_bold	0 - *(padrão)* Desabilitado; 1 - Habilitado.
Cor	1	primary_label_color	Código de cor hexadecimal (por exemplo, FF0000). Padrão: baseado no `theme` do objeto Settings e objeto User: 1F2C33 para "blue-theme" ou "hc-light"; EEEEEE para "dark-theme" ou "hc-dark".
Unidades (checkbox)	0	primary_label_units_show	0 - Desabilitado; 1 - *(padrão)* Habilitado. Comportamento do parâmetro: - *suportado* se *Tipo* estiver definido como "Valor"
Unidades (valor)	1	primary_label_units	Qualquer valor de string. "" (vazio) Comportamento do parâmetro: - *suportado* se *Tipo* estiver definido como "Valor" e *Unidades* (checkbox) estiver definido como "Habilitado"
Posição	0	primary_label_units_pos	0 - Antes do valor; 1 - *(padrão)* Após o valor. Comportamento do parâmetro: - *suportado* se *Tipo* estiver definido como "Valor" e *Unidades* (checkbox) estiver definido como "Habilitado" Este parâmetro é ignorado se definido como uma das seguintes unidades relacionadas ao tempo: unixtime, uptime, s.

Parâmetro	tipo	nome	valor
Rótulo secundário			
Tipo	0	secondary_label_type	0 - Texto; 1 - *(padrão)* Valor.
Texto	1	secondary_label	Qualquer valor de string, incluindo macros. Macros suportadas: {HOST.*}, {ITEM.*}, {INVENTORY.*}, macros de usuário. Padrão: {{ITEM.LASTVALUE}.fmtnum(2)} Comportamento do parâmetro: - *suportado* se *Tipo* estiver definido como ”Texto”
Casas decimais	0	secondary_label_decimal_places	Os valores possíveis variam de 0-6. Padrão: 2. Comportamento do parâmetro: - *suportado* se *Tipo* estiver definido como ”Valor”
Tamanho (tipo)	0	secondary_label_size_type	0 - *(padrão)* Automático; 1 - Personalizado.
Tamanho	0	secondary_label_size	Os valores possíveis variam de 1-100. Padrão: 30. Comportamento do parâmetro: - *suportado* se *Tamanho* (tipo) estiver definido como ”Personalizado”
Negrito	0	secondary_label_bold	0 - Desabilitado; 1 - *(padrão)* Habilitado.
Cor	1	secondary_label_color	Código de cor hexadecimal (por exemplo, `FF0000`). Padrão: baseado no `theme` do objeto Settings e objeto User: `1F2C33` para ”blue-theme” ou ”hc-light”; `EEEEEE` para ”dark-theme” ou ”hc-dark”.
Unidades (checkbox)	0	secondary_label_units_show	0 - Desabilitado; 1 - *(padrão)* Habilitado. Comportamento do parâmetro: - *suportado* se *Tipo* estiver definido como ”Valor”
Unidades (valor)	1	secondary_label_units	Qualquer valor de string. "" (vazio) Comportamento do parâmetro: - *suportado* se *Tipo* estiver definido como ”Valor” e *Unidades* (checkbox) estiver definido como ”Habilitado”
Posição	0	secondary_label_position	0 - Antes do valor; 1 - *(padrão)* Após o valor. Comportamento do parâmetro: - *suportado* se *Tipo* estiver definido como ”Valor” e *Unidades* (checkbox) estiver definido como ”Habilitado” Este parâmetro é ignorado se definido como uma das seguintes unidades relacionadas ao tempo: unixtime, uptime, s.
Cor de fundo			
Cor de fundo	1	bg_color	Código de cor hexadecimal (por exemplo, `FF0000`). Padrão: baseado no `theme` do objeto Settings e objeto User: `D9E7ED` para ”blue-theme”; `3D5059` para ”dark-theme”; `AAD7E9` para ”hc-light”; `335463` para ”hc-dark”.

Parâmetro	tipo	nome	valor
Limiar			
Interpolação de cor	0	interpolation	0 - Desabilitado; 1 - *(padrão)* Habilitado.
Cor	1	thresholds.0.color	Código de cor hexadecimal (por exemplo, FF0000).
Limiar	1	thresholds.0.threshold	Qualquer valor numérico. Sufixos (por exemplo, ”1d”, ”2w”, ”4K”, ”8G”) são suportados.

Exemplos

Os exemplos a seguir têm como objetivo apenas descrever a configuração dos objetos de campo do widget do painel para o widget *Honeycomb*. Para obter mais informações sobre como configurar um painel, consulte `dashboard.create`.

Configurando um widget *Honeycomb*

Configure um widget *Honeycomb* que exibe a utilização dos processos do Zabbix server. Além disso, altere o rótulo primário das células do honeycomb e ajuste visualmente o widget com limites.

Requisição:

```
{
    "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
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Veja também

- Campo do widget do dashboard
- `dashboard.create`
- `dashboard.update`

13 Disponibilidade do host

Descrição

Esses parâmetros e os possíveis valores de propriedade para os respectivos objetos de campo de widget de dashboard permitem configurar o widget *Disponibilidade do host* nos métodos `dashboard.create` e `dashboard.update`.

Parâmetros

Os seguintes parâmetros são suportados para o widget *Disponibilidade do host*.

Parâmetro	tipo	nome	valor
Intervalo de atualização	0	rf_rate	0 - Sem atualização; 10 - 10 segundos; 30 - 30 segundos; 60 - 1 minuto; 120 - 2 minutos; 600 - 10 minutos; 900 - *(padrão)* 15 minutos.
Grupos de hosts	2	groupids.0	ID do Grupo de hosts. Nota: Para configurar múltiplos grupos de hosts, crie um objeto de campo de widget de dashboard para cada grupo de hosts com um número incrementado no nome da propriedade. Este parâmetro não é suportado se o widget estiver sendo configurado em um dashboard de template.
Grupos de hosts (Widget)	1	groupids._reference	Em vez do ID do Grupo de hosts: `ABCDE._hostgroupids` - defina um widget compatível (com seu parâmetro *Referência* definido como "`ABCDE`") como a fonte de dados para grupos de hosts. Este parâmetro não é suportado se o widget estiver sendo configurado em um dashboard de template.
Tipo de interface	0	interface_type.0	0 - Nenhum; 1 - Zabbix agent (checagens passivas); 2 - SNMP; 3 - IPMI; 4 - JMX; 5 - Zabbix agent (checagens ativas). Padrão: 1, 2, 3, 4, 5 (todos habilitados). Nota: Para configurar múltiplos valores, crie um objeto de campo de widget de dashboard para cada valor com um número incrementado no nome da propriedade.
Layout	0	layout	0 - *(padrão)* Horizontal; 1 - Vertical.
Mostrar hosts em manutenção	0	maintenance	0 - *(padrão)* Desabilitado; 1 - Habilitado.
Mostrar apenas totais	0	only_totals	0 - *(padrão)* Desabilitado; 1 - Habilitado.

Exemplos

Os exemplos a seguir têm como objetivo apenas descrever a configuração dos objetos de campo do widget do painel para o widget *Disponibilidade do host*. Para obter mais informações sobre como configurar um painel, consulte `dashboard.create`.

Configurando um widget *Host availability*

Configure um widget *Host availability* que exibe informações de disponibilidade (em layout vertical) para hosts no grupo de hosts "4" com interfaces "Zabbix agent" e "SNMP" configuradas.

Requisição:

```
{
    "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
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Veja também

- Campo de widget de dashboard
- `dashboard.create`
- `dashboard.update`

14 Navegador de host

Descrição

Esses parâmetros e os possíveis valores de propriedade para os respectivos objetos de campo do widget do dashboard permitem configurar o widget *Navegador de hosts* nos métodos `dashboard.create` e `dashboard.update`.

Attention:
As propriedades `fields` do widget não são validadas durante a criação ou atualização de um dashboard. Isso permite que os usuários modifiquem widgets internos e criem widgets personalizados, mas também introduz o risco de criar ou atualizar widgets incorretamente. Para garantir a criação ou atualização bem-sucedida do widget *Navegador de hosts*, consulte o comportamento dos parâmetros descrito nas tabelas abaixo.

Parâmetros

Os seguintes parâmetros são suportados para o widget *Navegador de hosts*.

Parâmetro	type	name	value
Intervalo de atualização	0	rf_rate	0 - Sem atualização; 10 - 10 segundos; 30 - 30 segundos; 60 - *(padrão)* 1 minuto; 120 - 2 minutos; 600 - 10 minutos; 900 - 15 minutos.
Grupos de hosts	2	groupids.0	ID do Grupo de hosts. Nota: Para configurar múltiplos grupos de hosts, crie um objeto de campo de widget de dashboard para cada grupo de hosts com um número incrementado no nome da propriedade. Este parâmetro não é suportado se o widget estiver sendo configurado em um dashboard de template.

Parâmetro	type	name	value
Grupos de hosts (Widget)	1	groupids._reference	Em vez do ID do Grupo de hosts: `ABCDE._hostgroupids` - defina um widget compatível (com seu parâmetro *Referência* definido como "ABCDE") como a fonte de dados para grupos de hosts. Este parâmetro não é suportado se o widget estiver sendo configurado em um dashboard de template.
Padrões de hosts	1	hosts.0	Nome ou padrão do Host. Nota: Para configurar múltiplos padrões de hosts, crie um objeto de campo de widget de dashboard para cada padrão de host com um número incrementado no nome da propriedade. Para múltiplos padrões de hosts, o parâmetro *Grupos de hosts* deve não estar configurado ou estar configurado com pelo menos um grupo de hosts ao qual os hosts correspondentes aos padrões configurados pertençam. Este parâmetro não é suportado se o widget estiver sendo configurado em um dashboard de template.
Status do host	0	status	-1 - *(padrão)* Qualquer; 0 - Habilitado; 1 - Desabilitado. Este parâmetro não é suportado se o widget estiver sendo configurado em um dashboard de template.
Tags do host			
Tipo de avaliação	0	host_tags_evaltype	0 - *(padrão)* E/Ou; 2 - Ou. Este parâmetro não é suportado se o widget estiver sendo configurado em um dashboard de template.
Nome da tag	1	host_tags.0.tag	Qualquer valor de string. Nota: O número no nome da propriedade faz referência à ordem da tag na lista de avaliação de tags. Comportamento do parâmetro: - *obrigatório* se estiver configurando *Tags do host* Este parâmetro não é suportado se o widget estiver sendo configurado em um dashboard de template.
Operador	0	host_tags.0.operator	0 - Contém; 1 - Igual; 2 - Não contém; 3 - Diferente; 4 - Existe; 5 - Não existe. Nota: O número no nome da propriedade faz referência à ordem da tag na lista de avaliação de tags. Comportamento do parâmetro: - *obrigatório* se estiver configurando *Tags do host* Este parâmetro não é suportado se o widget estiver sendo configurado em um dashboard de template.

Parâmetro	type	name	value
Valor da tag	1	host_tags.0.value	Qualquer valor de string. Nota: O número no nome da propriedade faz referência à ordem da tag na lista de avaliação de tags. Comportamento do parâmetro: - *obrigatório* se estiver configurando *Tags do host* Este parâmetro não é suportado se o widget estiver sendo configurado em um dashboard de template.
Severidade	0	severities.0	0 - Não classificado; 1 - Informação; 2 - Aviso; 3 - Médio; 4 - Alto; 5 - Desastre. Padrão: vazio (todas habilitadas). Nota: Para configurar múltiplos valores, crie um objeto de campo de widget de dashboard para cada valor com um número incrementado no nome da propriedade.
Mostrar hosts em manutenção	0	maintenance	0 - *(padrão)* Desabilitado; 1 - Habilitado.
Mostrar problemas	0	show_problems	0 - Todos; 1 - *(padrão)* Não suprimidos; 2 - Nenhum.
Agrupar por			
Atributo	0	group_by.0.attribute	0 - Grupo de hosts; 1 - Valor da tag; 2 - Severidade. Nota: O número no nome da propriedade faz referência à ordem do atributo na lista de atributos de agrupamento. Comportamento do parâmetro: - *obrigatório* se estiver configurando *Agrupar por*
Valor	1	group_by.0.tag_name	Qualquer valor de string. Nota: O número no nome da propriedade faz referência ao atributo de agrupamento definido no parâmetro *Atributo*. Comportamento do parâmetro: - *obrigatório* se estiver configurando *Agrupar por* e *Atributo* estiver definido como ”Valor da tag”
Limite de hosts	0	show_lines	Os valores possíveis variam de 1-9999. Padrão: 100. Este parâmetro não é suportado se o widget estiver sendo configurado em um dashboard de template.
Referência	1	reference	Qualquer valor de string composto por 5 caracteres (por exemplo, `ABCDE` ou `JBPNL`). Este valor deve ser único dentro do dashboard ao qual o widget pertence. Comportamento do parâmetro: - *obrigatório*

Exemplos

Os exemplos a seguir têm como objetivo apenas descrever a configuração dos objetos de campo do widget do painel para o widget *Navegador de hosts*. Para obter mais informações sobre como configurar um painel, consulte `dashboard.create`.

Configurando um widget *Host navigator*

Configure um widget *Host navigator* que exibe hosts agrupados por seu grupo de hosts e, em seguida, pelo valor da tag "city".

Request:

```
{
    "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
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Veja também

- Campo do widget do dashboard
- `dashboard.create`
- `dashboard.update`

15 Histórico de item

Descrição

Esses parâmetros e os possíveis valores de propriedade para os respectivos objetos de campo do widget de dashboard permitem configurar o widget *Histórico do item* nos métodos `dashboard.create` e `dashboard.update`.

Attention:
As propriedades `fields` do widget não são validadas durante a criação ou atualização de um dashboard. Isso permite que os usuários modifiquem widgets internos e criem widgets personalizados, mas também introduz o risco de criar ou atualizar widgets incorretamente. Para garantir a criação ou atualização bem-sucedida do widget *Histórico do item*, consulte o comportamento dos parâmetros descrito nas tabelas abaixo.

Parâmetros

Os seguintes parâmetros são suportados para o widget *Histórico do item*.

Parâmetro	tipo	nome	valor
Intervalo de atualização	0	rf_rate	0 - Sem atualização; 10 - 10 segundos; 30 - 30 segundos; 60 - *(padrão)* 1 minuto; 120 - 2 minutos; 600 - 10 minutos; 900 - 15 minutos.
Layout	0	layout	0 - *(padrão)* Horizontal; 1 - Vertical.
Colunas (veja abaixo)			

Parâmetro	tipo	nome	valor
Exibir linhas	0	show_lines	Os valores possíveis variam de 1 a 100. Padrão: 25.
Substituir host	1	override_hostid._reference	`ABCDE._hostid` - define um widget compatível (com seu parâmetro *Referência* definido como "`ABCDE`") como fonte de dados para hosts; `DASHBOARD._hostid` - define o seletor *Host* do dashboard como fonte de dados para hosts. Este parâmetro não é suportado se o widget estiver sendo configurado em um dashboard de template.
Configuração avançada (veja abaixo)			
Referência	1	reference	Qualquer valor de string composto por 5 caracteres (por exemplo, `ABCDE` ou `JBPNL`). Este valor deve ser exclusivo dentro do dashboard ao qual o widget pertence. Comportamento do parâmetro: - *obrigatório*

Colunas

As colunas têm parâmetros comuns e parâmetros adicionais dependendo da configuração do parâmetro *Item*.

Note:
Para todos os parâmetros relacionados às colunas, o número no nome da propriedade (por exemplo, columns.0.name) refere-se a uma coluna para a qual o parâmetro está configurado.

Os seguintes parâmetros são suportados para todas as colunas.

Parâmetro	tipo	nome	valor
Nome	1	columns.0.name	Qualquer valor de string. Comportamento do parâmetro: - *obrigatório*
Item	4	columns.0.itemid	ID do item. Ao configurar o widget em um dashboard de template, apenas os itens configurados no template devem ser definidos. Comportamento do parâmetro: - *obrigatório*
Cor base	1	columns.0.base_color	Código de cor hexadecimal (por exemplo, `FF0000`). Padrão: `""` (vazio).

Os seguintes parâmetros de coluna são suportados se o *Item* configurado for um item do tipo numérico.

Parâmetro	type	name	value
Exibição	0	columns.0.display	1 - *(padrão)* Como está; 2 - Barra; 3 - Indicadores.
Mínimo	1	columns.0.min	Qualquer valor numérico. Comportamento do parâmetro: - *suportado* se *Exibição* estiver definido como "Barra" ou "Indicadores"

Parâmetro	type	name	value
Máximo	1	columns.0.max	Qualquer valor numérico. Comportamento do parâmetro: - *suportado* se *Exibição* estiver definido como ”Barra” ou ”Indicadores”
Limiares			
Cor	1	columns.0.thresholds.color	Código de cor hexadecimal (por exemplo, FF0000).
Limiar	1	columns.0.thresholds.threshold	Qualquer valor numérico. Sufixos (por exemplo, ”1d”, ”2w”, ”4K”, ”8G”) são suportados.
Dados históricos	0	columns.0.history	0 - *(padrão)* Automático; 1 - Histórico; 2 - Tendências.

Os seguintes parâmetros de coluna são suportados se o *Item* configurado for um item do tipo caractere, texto ou log.

Parâmetro	tipo	nome	valor
Destaques			
Destaque	1	columns.0.highlights.color	Código de cor hexadecimal (por exemplo, FF0000).
Limite	1	columns.0.highlights.pattern	Qualquer expressão regular.
Exibição	0	columns.0.display	1 - *(padrão)* Como está; 4 - HTML; 5 - Linha única.
Linha única	0	columns.0.max_length	Os valores possíveis variam de 1-500. Padrão: 100. Comportamento do parâmetro: - *suportado* se *Exibição* estiver definida como ”Linha única”
Usar fonte monoespaçada	0	columns.0.monospace_font	0 - *(padrão)* Usar fonte padrão; 1 - Usar fonte monoespaçada.
Exibir hora local	0	columns.0.local_time	0 - *(padrão)* Exibir timestamp; 1 - Exibir hora local. Comportamento do parâmetro: - *suportado* se *Item* estiver definido como item do tipo log e *Exibir timestamp* estiver definido como ”Habilitado”

Os seguintes parâmetros de coluna são suportados se o *Item* configurado for um item do tipo binário.

Parâmetro	type	name	value
Mostrar miniatura	1	columns.0.show_thumbnail	0 - *(padrão)* Desabilitado; 1 - Habilitado.

Configuração avançada

Os seguintes parâmetros de configuração avançada são suportados para o widget *Histórico do item*.

Parâmetro	tipo	nome	valor
Novos valores	0	sortorder	0 - *(padrão)* Topo; 1 - Fundo.
Mostrar timestamp	0	show_timestamp	0 - *(padrão)* Desabilitado; 1 - Habilitado.

Parâmetro	tipo	nome	valor
Mostrar cabeçalho da coluna	0	show_column_header	0 - Desligado; 1 - Horizontal; 2 - *(padrão)* Vertical.
Período de tempo	1	time_period._reference	`DASHBOARD._timeperiod` - define o seletor de *Período de tempo* como fonte de dados; `ABCDE._timeperiod` - define um widget compatível (com seu parâmetro *Referência* definido como "ABCDE") como fonte de dados. Padrão: `DASHBOARD._timeperiod` Alternativamente, você pode definir o período de tempo apenas nos parâmetros *De* e *Até*.
De	1	time_period.from	String de tempo válida em sintaxe absoluta (`YYYY-MM-DD hh:mm:ss`) ou relativa (`now`, `now/d`, `now/w-1w`, etc.). Comportamento do parâmetro: - *suportado* se *Período de tempo* não estiver definido - *obrigatório* se `time_period.to` estiver definido
Até	1	time_period.to	String de tempo válida em sintaxe absoluta (`YYYY-MM-DD hh:mm:ss`) ou relativa (`now`, `now/d`, `now/w-1w`, etc.). Comportamento do parâmetro: - *suportado* se *Período de tempo* não estiver definido - *obrigatório* se `time_period.from` estiver definido

Exemplos

Os seguintes exemplos têm como objetivo descrever apenas a configuração dos objetos de campo do widget do painel para o widget *Histórico de item*. Para mais informações sobre como configurar um painel, consulte `dashboard.create`.

Configurando um widget *Histórico de item*

Configure um widget *Histórico de item* que exibe os dados mais recentes para dois itens numéricos "42269" e "42270". Além disso, configure as colunas dos itens para serem exibidas verticalmente, com os nomes das colunas exibidos horizontalmente; limite a exibição a 15 linhas de dados e inclua uma coluna de data/hora separada.

Requisição:

```
{
    "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
}
```


Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Veja também

- Campo do widget do painel
- `dashboard.create`
- `dashboard.update`

16 Navegador de itens

Descrição

Esses parâmetros e os possíveis valores de propriedade para os respectivos objetos de campo do widget de dashboard permitem configurar o widget *Navegador de item* nos métodos `dashboard.create` e `dashboard.update`.

Attention:
As propriedades `fields` do widget não são validadas durante a criação ou atualização de um dashboard. Isso permite que os usuários modifiquem widgets internos e criem widgets personalizados, mas também introduz o risco de criar ou atualizar widgets incorretamente. Para garantir a criação ou atualização bem-sucedida do widget *Navegador de item*, consulte o comportamento dos parâmetros descrito nas tabelas abaixo.

Parâmetros

Os seguintes parâmetros são suportados para o widget *Navegador de item*.

Parâmetro	type	name	value
Intervalo de atualização	0	rf_rate	0 - Sem atualização; 10 - 10 segundos; 30 - 30 segundos; 60 - *(padrão)* 1 minuto; 120 - 2 minutos; 600 - 10 minutos; 900 - 15 minutos.
Grupos de hosts	2	groupids.0	ID do Grupo de hosts. Nota: Para configurar múltiplos grupos de hosts, crie um objeto de campo de widget de dashboard para cada grupo de hosts com um número incrementado no nome da propriedade. Este parâmetro não é suportado se o widget estiver sendo configurado em um dashboard de template.
Grupos de hosts (Widget)	1	groupids._reference	Em vez do ID do Grupo de hosts: `ABCDE._hostgroupids` - defina um widget compatível (com seu parâmetro *Referência* definido como "ABCDE") como a fonte de dados para grupos de hosts. Este parâmetro não é suportado se o widget estiver sendo configurado em um dashboard de template.

Parâmetro	type	name	value
Hosts	3	hostids.0	ID do Host. Nota: Para configurar múltiplos hosts, crie um objeto de campo de widget de dashboard para cada host com um número incrementado no nome da propriedade. Para múltiplos hosts, o parâmetro *Grupos de hosts* deve não estar configurado ou estar configurado com pelo menos um grupo de hosts ao qual os hosts configurados pertencem. Este parâmetro não é suportado se o widget estiver sendo configurado em um dashboard de template.
Hosts (Widget/Dashboard)	1	hostids._reference	Em vez do ID do Host: `DASHBOARD.hostid` - defina o seletor de *Host* como a fonte de dados para hosts; `ABCDE._hostid` - defina um widget compatível (com seu parâmetro *Referência* definido como "`ABCDE`") como a fonte de dados para hosts. Este parâmetro não é suportado se o widget estiver sendo configurado em um dashboard de template.
Tags de host			
Tipo de avaliação	0	host_tags_evaltype	0 - *(padrão)* E/Ou; 2 - Ou. Este parâmetro não é suportado se o widget estiver sendo configurado em um dashboard de template.
Nome da tag	1	host_tags.0.tag	Qualquer valor de string. Nota: O número no nome da propriedade faz referência à ordem da tag na lista de avaliação de tags. Comportamento do parâmetro: - *obrigatório* se estiver configurando *Tags de host* Este parâmetro não é suportado se o widget estiver sendo configurado em um dashboard de template.
Operador	0	host_tags.0.operator	0 - Contém; 1 - Igual; 2 - Não contém; 3 - Diferente; 4 - Existe; 5 - Não existe. Nota: O número no nome da propriedade faz referência à ordem da tag na lista de avaliação de tags. Comportamento do parâmetro: - *obrigatório* se estiver configurando *Tags de host* Este parâmetro não é suportado se o widget estiver sendo configurado em um dashboard de template.
Valor da tag	1	host_tags.0.value	Qualquer valor de string. Nota: O número no nome da propriedade faz referência à ordem da tag na lista de avaliação de tags. Comportamento do parâmetro: - *obrigatório* se estiver configurando *Tags de host* Este parâmetro não é suportado se o widget estiver sendo configurado em um dashboard de template.

Parâmetro	type	name	value
Padrões de item	1	items.0	Nome ou padrão do Item. Nota: Para configurar múltiplos padrões de item, crie um objeto de campo de widget de dashboard para cada padrão de item com um número incrementado no nome da propriedade.
Tags de item			
Tipo de avaliação	0	item_tags_evaltype	0 - *(padrão)* E/Ou; 2 - Ou.
Nome da tag	1	item_tags.0.tag	Qualquer valor de string. Nota: O número no nome da propriedade faz referência à ordem da tag na lista de avaliação de tags. Comportamento do parâmetro: - *obrigatório* se estiver configurando *Tags de item*
Operador	0	item_tags.0.operator	0 - Contém; 1 - Igual; 2 - Não contém; 3 - Diferente; 4 - Existe; 5 - Não existe. Nota: O número no nome da propriedade faz referência à ordem da tag na lista de avaliação de tags. Comportamento do parâmetro: - *obrigatório* se estiver configurando *Tags de item*
Valor da tag	1	item_tags.0.value	Qualquer valor de string. Nota: O número no nome da propriedade faz referência à ordem da tag na lista de avaliação de tags. Comportamento do parâmetro: - *obrigatório* se estiver configurando *Tags de item*
Estado	0	state	-1 - *(padrão)* Todos; 0 - Normal; 1 - Não suportado.
Exibir problemas	0	show_problems	0 - Todos; 1 - *(padrão)* Não suprimidos; 2 - Nenhum.
Agrupar por			
Atributo	0	group_by.0.attribute	0 - Grupo de hosts; 1 - Nome do host; 2 - Valor da tag do host; 3 - Valor da tag do item. Nota: O número no nome da propriedade faz referência à ordem do atributo na lista de atributos de agrupamento. Comportamento do parâmetro: - *obrigatório* se estiver configurando *Agrupar por*

Parâmetro	type	name	value
Valor	1	group_by.0.tag_name	Qualquer valor de string. Nota: O número no nome da propriedade faz referência ao atributo de agrupamento definido no parâmetro *Atributo*. Comportamento do parâmetro: - *obrigatório* se estiver configurando *Agrupar por* e *Atributo* estiver definido como ”Valor da tag do host” ou ”Valor da tag do item”
Limite de itens	0	show_lines	Os valores possíveis variam de 1-9999. Padrão: 100.
Referência	1	reference	Qualquer valor de string composto por 5 caracteres (por exemplo, `ABCDE` ou `JBPNL`). Este valor deve ser exclusivo dentro do dashboard ao qual o widget pertence. Comportamento do parâmetro: - *obrigatório*

Exemplos

Os exemplos a seguir têm como objetivo apenas descrever a configuração dos objetos de campo do widget do painel para o widget *Navegador de item*. Para obter mais informações sobre como configurar um painel, consulte `dashboard.create`.

Configurando um widget *Navegador de items*

Configure um widget *Navegador de items* que exibe até 1000 items agrupados por seu host e, em seguida, pelo valor da tag de item ”component”.

Requisição:

```
{
    "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
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Veja também

- Campo do widget do dashboard
- `dashboard.create`
- `dashboard.update`

17 Valor do item

Descrição

Esses parâmetros e os possíveis valores de propriedade para os respectivos objetos de campo do widget de dashboard permitem configurar o widget *Valor do item* nos métodos `dashboard.create` e `dashboard.update`.

Attention:
As propriedades `fields` do widget não são validadas durante a criação ou atualização de um dashboard. Isso permite que os usuários modifiquem widgets internos e criem widgets personalizados, mas também introduz o risco de criar ou atualizar widgets incorretamente. Para garantir a criação ou atualização bem-sucedida do widget *Valor do item*, consulte o comportamento dos parâmetros descrito nas tabelas abaixo.

Parâmetros

Os seguintes parâmetros são suportados para o widget *Valor do item*.

Parâmetro	tipo	nome	valor
Intervalo de atualização	0	rf_rate	0 - Sem atualização; 10 - 10 segundos; 30 - 30 segundos; 60 - *(padrão)* 1 minuto; 120 - 2 minutos; 600 - 10 minutos; 900 - 15 minutos.
Item	4	itemid.0	ID do Item. Comportamento do parâmetro: - *obrigatório* se *Item (Widget)* não estiver definido
Item (Widget)	1	itemid._reference	Em vez do ID do Item: `ABCDE._itemid` - defina um widget compatível (com seu parâmetro *Referência* definido como "ABCDE") como a fonte de dados para os items. Comportamento do parâmetro: - *obrigatório* se *Item* não estiver definido
Mostrar	0	show.0	1 - Descrição; 2 - Valor; 3 - Hora; 4 - Indicador de alteração. Padrão: 1, 2, 3, 4 (todos habilitados). Nota: Para configurar vários valores, crie um objeto de campo de widget de dashboard para cada valor com um número incrementado no nome da propriedade.
Sobrescrever host	1	override_hostid._reference	`ABCDE._hostid` - defina um widget compatível (com seu parâmetro *Referência* definido como "ABCDE") como a fonte de dados para hosts; `DASHBOARD._hostid` - defina o seletor de *Host* do dashboard como a fonte de dados para hosts. Este parâmetro não é suportado se o widget estiver sendo configurado em um dashboard de template.

Configuração avançada

Os seguintes parâmetros de configuração avançada são suportados para o widget *Valor do item*.

Note:
O número no nome da propriedade *Thresholds* (por exemplo, thresholds.0.color) faz referência à posição do threshold em uma lista, ordenada em ordem crescente. No entanto, se os thresholds forem configurados em uma ordem diferente, os valores serão ordenados em ordem crescente após a atualização da configuração do widget no frontend do Zabbix (por exemplo, `"thresholds.0.threshold":"5"` → `"thresholds.0.threshold":"1"`; `"thresholds.1.threshold":"1"` → `"thresholds.1.threshold": "5"`).

Parâmetro	tipo	nome	valor
Cor de fundo	1	bg_color	Código de cor hexadecimal (por exemplo, FF0000). Padrão: "" (vazio).
Thresholds			
Cor	1	thresholds.0.color	Código de cor hexadecimal (por exemplo, FF0000).
Threshold	1	thresholds.0.threshold	Qualquer valor de string.

Parâmetro	tipo	nome	valor
Função de agre-gação	0	aggregate_function	0 - *(padrão)* não usado; 1 - min; 2 - max; 3 - avg; 4 - count; 5 - sum; 6 - first; 7 - last.
Período de tempo	1	time_period._referenda	`DASHBOARD._timeperiod` - define o seletor de *Período de tempo* como a fonte de dados; `ABCDE._timeperiod` - define um widget compatível (com seu parâmetro `reference` igual a `ABCDE`) como a fonte de dados. Padrão: `DASHBOARD._timeperiod` Alternativamente, você pode definir o período de tempo apenas nos parâmetros *From* e *To*.
From	1	time_period.from	Comportamento do parâmetro: - *suportado* se *Função de agregação* estiver definida como ”min”, ”max”, ”avg”, ”count”, ”sum”, ”first”, ”last” String de tempo válida em sintaxe absoluta (`YYYY-MM-DD hh:mm:ss`) ou relativa (`now`, `now/d`, `now/w-1w`, etc.).
To	1	time_period.to	Comportamento do parâmetro: - *suportado* se *Período de tempo* não estiver definido e *Função de agregação* estiver definida como ”min”, ”max”, ”avg”, ”count”, ”sum”, ”first”, ”last” - *obrigatório* se `time_period.to` estiver definido String de tempo válida em sintaxe absoluta (`YYYY-MM-DD hh:mm:ss`) ou relativa (`now`, `now/d`, `now/w-1w`, etc.).
Dados de histórico	0	history	Comportamento do parâmetro: - *suportado* se *Período de tempo* não estiver definido e *Função de agregação* estiver definida como ”min”, ”max”, ”avg”, ”count”, ”sum”, ”first”, ”last” - *obrigatório* se `time_period.from` estiver definido 0 - *(padrão)* Auto; 1 - Histórico; 2 - Tendências.

Descrição

Os seguintes parâmetros de configuração avançada são suportados se *Mostrar* estiver definido como ”Descrição”.

Parâmetro	tipo	nome	valor
Descrição	1	description	Qualquer valor de string, incluindo macros. Macros suportadas: {HOST.*}, {ITEM.*}, {INVENTORY.*}, Macros de usuário. Padrão: {ITEM.NAME}.
Posição horizontal	0	desc_h_pos	0 - Esquerda; 1 - *(padrão)* Centro; 2 - Direita. Dois ou mais elementos (Descrição, Valor, Hora) não podem compartilhar a mesma *Posição horizontal* e *Posição vertical*.

Parâmetro	tipo	nome	valor
Posição vertical	0	desc_v_pos	0 - Topo; 1 - Meio; 2 - *(padrão)* Inferior. Dois ou mais elementos (Descrição, Valor, Hora) não podem compartilhar a mesma *Posição horizontal* e *Posição vertical*.
Tamanho	0	desc_size	Os valores possíveis variam de 1 a 100. Padrão: 15.
Negrito	0	desc_bold	0 - *(padrão)* Desabilitado; 1 - Habilitado.
Cor	1	desc_color	Código de cor hexadecimal (por exemplo, FF0000). Padrão: "" (vazio).

Valor

Os seguintes parâmetros de configuração avançada são suportados se *Mostrar* estiver definido como ”Valor”.

Parâmetro	tipo	nome	valor
Casas decimais			
Casas decimais	0	decimal_places	Os valores possíveis variam de 1-10. Padrão: 2.
Tamanho	0	decimal_size	Os valores possíveis variam de 1-100. Padrão: 35.
Posição			
Posição horizontal	0	value_h_pos	0 - Esquerda; 1 - *(padrão)* Centro; 2 - Direita.
Posição vertical	0	value_v_pos	Dois ou mais elementos (Descrição, Valor, Hora) não podem compartilhar a mesma *Posição horizontal* e *Posição vertical*. 0 - Superior; 1 - *(padrão)* Meio; 2 - Inferior.
Tamanho	0	value_size	Dois ou mais elementos (Descrição, Valor, Hora) não podem compartilhar a mesma *Posição horizontal* e *Posição vertical*. Os valores possíveis variam de 1-100. Padrão: 45.
Negrito	0	value_bold	0 - Desabilitado; 1 - *(padrão)* Habilitado.
Cor	1	value_color	Código de cor hexadecimal (por exemplo, FF0000). Padrão: "" (vazio).
Unidades			
Unidades (checkbox)	0	units_show	0 - Desabilitado; 1 - *(padrão)* Habilitado.
Unidades (valor)	1	units	Qualquer valor de string.
Posição	0	units_pos	0 - Antes do valor; 1 - Acima do valor; 2 - *(padrão)* Após o valor; 3 - Abaixo do valor.

Parâmetro	tipo	nome	valor
Tamanho	0	units_size	Os valores possíveis variam de 1-100. Padrão: 35.
Negrito	0	units_bold	0 - Desabilitado; 1 - *(padrão)* Habilitado.
Cor	1	units_color	Código de cor hexadecimal (por exemplo, FF0000). Padrão: "" (vazio).

Hora

Os seguintes parâmetros de configuração avançada são suportados se *Mostrar* estiver definido como ”Hora”.

Parâmetro	tipo	nome	valor
Posição horizontal	0	time_h_pos	0 - Esquerda; 1 - *(padrão)* Centro; 2 - Direita. Dois ou mais elementos (Descrição, Valor, Hora) não podem compartilhar a mesma *Posição horizontal* e *Posição vertical*.
Posição vertical	0	time_v_pos	0 - *(padrão)* Superior; 1 - Meio; 2 - Inferior. Dois ou mais elementos (Descrição, Valor, Hora) não podem compartilhar a mesma *Posição horizontal* e *Posição vertical*.
Tamanho	0	time_size	Os valores possíveis variam de 1 a 100. Padrão: 15.
Negrito	0	time_bold	0 - *(padrão)* Desabilitado; 1 - Habilitado.
Cor	1	time_color	Código de cor hexadecimal (por exemplo, FF0000). Padrão: "" (vazio).

Indicador de alteração

Os seguintes parâmetros de configuração avançada são suportados se *Mostrar* estiver definido como ”Indicador de alteração”.

Parâmetro	tipo	nome	valor
Cor do indicador de alteração ↑	1	up_color	Código de cor hexadecimal (por exemplo, FF0000). Padrão: "" (vazio).
Cor do indicador de alteração ↓	1	down_color	Código de cor hexadecimal (por exemplo, FF0000). Padrão: "" (vazio).
Cor do indicador de alteração ↕	1	updown_color	Código de cor hexadecimal (por exemplo, FF0000). Padrão: "" (vazio).

Exemplos

Os exemplos a seguir têm como objetivo apenas descrever a configuração dos objetos de campo do widget do painel para o widget *Valor do item*. Para obter mais informações sobre como configurar um painel, consulte `dashboard.create`.

Configurando um widget *Valor do item*

Configure um widget *Valor do item* que exibe o valor do item para o item ”42266” (disponibilidade do agent Zabbix). Além disso, ajuste visualmente o widget com várias opções avançadas, incluindo uma cor de fundo dinâmica que muda com base no status de disponibilidade do agent Zabbix.

Requisição:

```
{
    "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
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Veja também

- Campo de widget de dashboard
- `dashboard.create`
- `dashboard.update`

18 Mapa

Descrição

Esses parâmetros e os possíveis valores de propriedade para os respectivos objetos de campo do widget de dashboard permitem configurar o widget *Mapa* nos métodos `dashboard.create` e `dashboard.update`.

Attention:
As propriedades `fields` do widget não são validadas durante a criação ou atualização de um dashboard. Isso permite que os usuários modifiquem widgets internos e criem widgets personalizados, mas também introduz o risco de criar ou atualizar widgets incorretamente. Para garantir a criação ou atualização bem-sucedida do widget *Mapa*, consulte o comportamento dos parâmetros descritos nas tabelas abaixo.

Parâmetros

Os seguintes parâmetros são suportados para o widget *Mapa*.

Parâmetro	tipo	nome	valor
Intervalo de atualização	0	rf_rate	0 - Sem atualização; 10 - 10 segundos; 30 - 30 segundos; 60 - 1 minuto; 120 - 2 minutos; 600 - 10 minutos; 900 - *(padrão)* 15 minutos.
Mapa	8	sysmapid.0	ID do Mapa. Comportamento do parâmetro: - *obrigatório* se *Mapa (Widget)* não estiver definido
Mapa (Widget)	1	sysmapid._reference	`ABCDE._mapid` - define um widget *Árvore de navegação de mapas* (com seu parâmetro *Referência* definido como "ABCDE") como a fonte de dados para mapas. Comportamento do parâmetro: - *obrigatório* se *Mapa* não estiver definido
Referência	1	reference	Qualquer valor de string consistindo de 5 caracteres (por exemplo, `ABCDE` ou `JBPNL`). Este valor deve ser exclusivo dentro do dashboard ao qual o widget pertence. Comportamento do parâmetro: - *obrigatório*

Exemplos

Os exemplos a seguir têm como objetivo apenas descrever a configuração dos objetos de campo do widget de painel para o widget *Mapa*. Para obter mais informações sobre como configurar um painel, consulte `dashboard.create`.

Configurando um widget *Mapa*

Configure um widget *Mapa* que exibe o mapa "1".

Requisição:

```
{
    "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
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Configurando um widget *Mapa* vinculado

Configure um widget *Mapa* que está vinculado a um widget *Árvore de navegação de mapas*.

Requisição:

```
{
    "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
```

```
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Veja também

- Campo do widget do dashboard
- `dashboard.create`
- `dashboard.update`
- Árvore de navegação do mapa

19 Árvore de navegação do mapa

Descrição

Esses parâmetros e os possíveis valores de propriedade para os respectivos objetos de campo do widget de dashboard permitem configurar o widget *Árvore de navegação de mapas* nos métodos `dashboard.create` e `dashboard.update`.

Attention:
As propriedades `fields` do widget não são validadas durante a criação ou atualização de um dashboard. Isso permite que os usuários modifiquem widgets internos e criem widgets personalizados, mas também introduz o risco de criar ou atualizar widgets incorretamente. Para garantir a criação ou atualização bem-sucedida do widget *Árvore de navegação de mapas*, consulte o comportamento dos parâmetros descrito nas tabelas abaixo.

Parâmetros

Os seguintes parâmetros são suportados para o widget *Árvore de navegação do mapa*.

Parâmetro	tipo	nome	valor
Intervalo de atualização	0	rf_rate	0 - Sem atualização; 10 - 10 segundos; 30 - 30 segundos; 60 - 1 minuto; 120 - 2 minutos; 600 - 10 minutos; 900 - *(padrão)* 15 minutos.
Mostrar mapas indisponíveis	1	show_unavailable	0 - *(padrão)* Desabilitado; 1 - Habilitado.
Referência	1	reference	Qualquer valor de string consistindo de 5 caracteres (por exemplo, `ABCDE` ou `JBPNL`). Este valor deve ser único dentro do dashboard ao qual o widget pertence. Comportamento do parâmetro: - *obrigatório*

Os seguintes parâmetros são suportados para configurar os elementos da árvore de navegação do mapa.

Parâmetro	tipo	nome	valor
Nome	1	navtree.1.name	Qualquer valor de string. Nota: O número no nome da propriedade define o número do elemento.

Parâmetro	tipo	nome	valor
Mapa vinculado	8	navtree.1.sysmapid	ID do Mapa. Nota: O número no nome da propriedade referencia o elemento ao qual o mapa está vinculado.
Parâmetros para criar hierarquia de elementos	0	navtree.1.parent	Número do elemento pai. Nota: O número no nome da propriedade referencia o elemento filho. O valor da propriedade referencia o elemento pai.
	0	navtree.1.order	Posição do elemento na árvore de navegação do mapa. Nota: O número no nome da propriedade referencia o número do elemento. O valor da propriedade referencia a posição do elemento na árvore de navegação do mapa. A posição do elemento pai é determinada dentro de toda a árvore de navegação do mapa. A posição do elemento filho é determinada dentro do elemento pai.

Exemplos

Os exemplos a seguir têm como objetivo apenas descrever a configuração dos objetos de campo do widget do painel para o widget *Árvore de navegação do mapa*. Para obter mais informações sobre como configurar um painel, consulte `dashboard.create`.

Configurando um widget *Árvore de navegação de mapas*

Configure um widget *Árvore de navegação de mapas* que exibe a seguinte árvore de navegação de mapas:

- Elemento A
  - Elemento A1
  - Elemento A2
- Elemento B
  - Elemento B1 (contém o mapa vinculado "1" que pode ser exibido em um *widget de mapa* vinculado)
  - Elemento B2
- Elemento C

Requisição:

```
{
    "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
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Veja também

- Campo do widget do dashboard

- `dashboard.create`
- `dashboard.update`
- Mapa

20 Gráfico de pizza

Descrição

Esses parâmetros e os possíveis valores de propriedade para os respectivos objetos de campo do widget do dashboard permitem configurar o widget *Gráfico de pizza* nos métodos `dashboard.create` e `dashboard.update`.

Attention:
As propriedades `fields` do widget não são validadas durante a criação ou atualização de um dashboard. Isso permite que os usuários modifiquem widgets internos e criem widgets personalizados, mas também introduz o risco de criar ou atualizar widgets incorretamente. Para garantir a criação ou atualização bem-sucedida do widget *Gráfico de pizza*, consulte o comportamento dos parâmetros descritos nas tabelas abaixo.

Parâmetros

Os seguintes parâmetros são suportados para o widget *Gráfico de pizza*.

Parâmetro	type	name	value
Intervalo de atualização	0	rf_rate	0 - Sem atualização; 10 - 10 segundos; 30 - 30 segundos; 60 - *(padrão)* 1 minuto; 120 - 2 minutos; 600 - 10 minutos; 900 - 15 minutos.

Conjunto de dados

Os seguintes parâmetros são suportados para configurar um *Conjunto de dados*.

Note:
O primeiro número no nome da propriedade (por exemplo, ds.0.hosts.0, ds.0.items.0) representa o conjunto de dados específico, enquanto o segundo número, se presente, representa o host ou item configurado.

Parâmetro	tipo	nome	valor
Tipo de conjunto de dados	0	ds.0.dataset_type	0 - Lista de items; 1 - *(padrão)* Padrões de items.
Items	4	ds.0.itemids.0	ID do item. Ao configurar o widget em um dashboard de template, apenas os items configurados no template devem ser definidos. Nota: Para configurar múltiplos items, crie um objeto de campo de widget de dashboard para cada item com um número incrementado no nome da propriedade. Comportamento do parâmetro: - *obrigatório* se o *Tipo de conjunto de dados* estiver definido como "Lista de items" e *Items (Widget)* não estiver definido

Parâmetro	tipo	nome	valor
Items (Widget)	1	ds.0.itemids.0._reference	Em vez do ID do item: `ABCDE._itemid` - defina um widget compatível (com seu parâmetro *Referência* definido como "ABCDE") como a fonte de dados para os items. Nota: Para configurar múltiplos widgets, crie um objeto de campo de widget de dashboard para cada widget com um número incrementado no nome da propriedade. Este parâmetro é suportado desde o Zabbix 7.0.1. Comportamento do parâmetro: - *obrigatório* se o *Tipo de conjunto de dados* estiver definido como "Lista de items" e *Items* não estiver definido
Cor	1	ds.0.color.0	Código de cor hexadecimal (por exemplo, FF0000). Comportamento do parâmetro: - *obrigatório* se o *Tipo de conjunto de dados* estiver definido como "Lista de items"
Tipo de item	0	ds.0.type.0	0 - *(padrão)* Normal; 1 - Total. O valor "Total" pode ser definido apenas para um item em todo o gráfico. Comportamento do parâmetro: - *suportado* se o *Tipo de conjunto de dados* estiver definido como "Lista de items"
Padrões de host	1	ds.0.hosts.0	Nome ou padrão do host (por exemplo, "Zabbix*"). Comportamento do parâmetro: - *obrigatório* se o *Tipo de conjunto de dados* estiver definido como "Padrões de items" Este parâmetro não é suportado se estiver configurando o widget em um dashboard de template.
Padrões de items	1	ds.0.items.0	Nome ou padrão do item (por exemplo, "*: Número de valores processados por segundo"). Ao configurar o widget em um dashboard de template, apenas os padrões para items configurados no template devem ser definidos. Comportamento do parâmetro: - *obrigatório* se o *Tipo de conjunto de dados* estiver definido como "Padrões de items"
Cor	1	ds.0.color	Código de cor hexadecimal (por exemplo, FF0000). Comportamento do parâmetro: - *suportado* se o *Tipo de conjunto de dados* estiver definido como "Padrões de items"
Função de agregação	0	ds.0.aggregate_function	1 - min; 2 - max; 3 - avg; 4 - count; 5 - sum; 6 - first; 7 - *(padrão)* last.

Parâmetro	tipo	nome	valor
Agregação do conjunto de dados	0	ds.0.dataset_aggregation	0 - *(padrão)* nenhum; 1 - min; 2 - max; 3 - avg; 4 - count; 5 - sum. Comportamento do parâmetro: - *suportado* se o *Tipo de item* estiver definido como "Total"
Rótulo do conjunto de dados	1	ds.0.data_set_label	Qualquer valor de string. Padrão: "" (vazio).

Opções de exibição

Os seguintes parâmetros são suportados para configurar as *Opções de exibição*.

Parâmetro	tipo	nome	valor
Seleção de dados históricos	0	source	0 - *(padrão)* Automático; 1 - Histórico; 2 - Tendências.
Desenhar	0	draw_type	0 - *(padrão)* Pizza; 1 - Rosquinha.
Largura	0	width	20 - 20% do raio; 30 - 30% do raio; 40 - 40% do raio; 50 - *(padrão)* 50% do raio. Comportamento do parâmetro: - *suportado* se *Desenhar* estiver definido como "Rosquinha"
Largura do traço	0	stroke	Os valores possíveis variam de 0-10. Padrão: 0. Este parâmetro é suportado desde o Zabbix 7.0.1. Comportamento do parâmetro: - *suportado* se *Desenhar* estiver definido como "Rosquinha"
Mostrar valor total	0	total_show	0 - *(padrão)* Desabilitado; 1 - Habilitado. Comportamento do parâmetro: - *suportado* se *Desenhar* estiver definido como "Rosquinha"
Tamanho	0	value_size_type	0 - *(padrão)* Automático; 1 - Personalizado. Comportamento do parâmetro: - *suportado* se *Mostrar valor total* estiver definido como "Habilitado"

Parâmetro	tipo	nome	valor
Tamanho (valor para tamanho personalizado)	0	value_size	Os valores possíveis variam de 1-100. Padrão: 20. Comportamento do parâmetro: - *suportado* se *Mostrar valor total* estiver definido como ”Habilitado”
Casas decimais	0	decimal_places	Os valores possíveis variam de 0-6. Padrão: 2. Comportamento do parâmetro: - *suportado* se *Mostrar valor total* estiver definido como ”Habilitado”
Unidades (caixa de seleção)	0	units_show	0 - *(padrão)* Desabilitado; 1 - Habilitado. Comportamento do parâmetro: - *suportado* se *Mostrar valor total* estiver definido como ”Habilitado”
Unidades (valor)	1	units	Qualquer valor de string. Comportamento do parâmetro: - *suportado* se *Unidades* (caixa de seleção) estiver definido como ”Habilitado”
Negrito	0	value_bold	0 - *(padrão)* Desabilitado; 1 - Habilitado. Comportamento do parâmetro: - *suportado* se *Mostrar valor total* estiver definido como ”Habilitado”
Cor	1	value_color	Código de cor hexadecimal (por exemplo, FF0000). Comportamento do parâmetro: - *suportado* se *Mostrar valor total* estiver definido como ”Habilitado”
Espaço entre setores	0	space	Os valores possíveis variam de 0-10. Padrão: 1.
Mesclar setores menores que N% (caixa de seleção)	0	merge	0 - *(padrão)* Desabilitado; 1 - Habilitado.
Mesclar setores menores que N% (valor)	0	merge_percent	Os valores possíveis variam de 1-10. Padrão: 1. Comportamento do parâmetro: - *suportado* se *Mesclar setores menores que N%* (caixa de seleção) estiver definido como ”Habilitado”
Mesclar setores menores que N% (cor)	1	merge_color	Código de cor hexadecimal (por exemplo, FF0000). Comportamento do parâmetro: - *suportado* se *Mesclar setores menores que N%* (caixa de seleção) estiver definido como ”Habilitado”

Período de tempo

Os seguintes parâmetros são suportados para configurar o *Período de tempo*.

Parâmetro	tipo	nome	valor
Período de tempo	1	time_period._referenceDASHBOARD	`DASHBOARD._timeperiod` - define o seletor de *Período de tempo* como a fonte de dados; `ABCDE._timeperiod` - define um widget compatível (com seu parâmetro *Referência* definido como "ABCDE") como a fonte de dados. Padrão: `DASHBOARD._timeperiod` Alternativamente, você pode definir o período de tempo apenas nos parâmetros *De* e *Até*.
De	1	time_period.from	String de tempo válida na sintaxe absoluta (`YYYY-MM-DD hh:mm:ss`) ou relativa (`now`, `now/d`, `now/w-1w`, etc.). Comportamento do parâmetro: - *suportado* se *Período de tempo* não estiver definido - *obrigatório* se `time_period.to` estiver definido
Até	1	time_period.to	String de tempo válida na sintaxe absoluta (`YYYY-MM-DD hh:mm:ss`) ou relativa (`now`, `now/d`, `now/w-1w`, etc.). Comportamento do parâmetro: - *suportado* se *Período de tempo* não estiver definido - *obrigatório* se `time_period.from` estiver definido

Legenda

Os seguintes parâmetros são suportados para configurar a *Legenda*.

Parâmetro	tipo	nome	valor
Exibir legenda	0	legend	0 - Desabilitado; 1 - *(padrão)* Habilitado.
Exibir valor	0	legend_value	0 - *(padrão)* Desabilitado; 1 - Habilitado. Comportamento do parâmetro: - *suportado* se *Exibir legenda* estiver definido como "Habilitado"
Exibir função de agregação	0	legend_aggregation	0 - *(padrão)* Desabilitado; 1 - Habilitado. Comportamento do parâmetro: - *suportado* se *Exibir legenda* estiver definido como "Habilitado"
Linhas	0	legend_lines_mode	0 - *(padrão)* Fixo; 1 - Variável. Comportamento do parâmetro: - *suportado* se *Exibir legenda* estiver definido como "Habilitado"
Número de linhas/ Número máximo de linhas	0	legend_lines	Os valores possíveis variam de 1-10. Padrão: 1. Comportamento do parâmetro: - *suportado* se *Exibir legenda* estiver definido como "Habilitado"

Parâmetro	tipo	nome	valor
Número de colunas	0	legend_columns	Os valores possíveis variam de 1-4. Padrão: 4. Comportamento do parâmetro: - *suportado* se *Exibir legenda* estiver definido como ”Habilitado” e *Exibir valor* estiver definido como ”Desabilitado”

Exemplos

Os exemplos a seguir têm como objetivo apenas descrever a configuração dos objetos de campo do widget do dashboard para o widget *Gráfico de pizza*. Para obter mais informações sobre como configurar um dashboard, consulte `dashboard.create`.

Configurando um widget *Pie chart*

Configure um widget *Pie chart* da seguinte forma:

- 2 conjuntos de dados para um total de 9 items em 1 host.
- O primeiro conjunto de dados é do tipo ”Lista de items” e consiste em 3 items, todos do tipo ”Normal” e representados por uma cor diferente.
- O segundo conjunto de dados é do tipo ”Padrões de items”, consiste em 6 items, possui uma agregação configurada para cada item e é representado por uma cor personalizada.
- O segundo conjunto de dados também possui um rótulo de conjunto de dados personalizado.
- Os dados no gráfico de pizza são exibidos como um gráfico de rosca com uma largura personalizada e valor total com unidades no centro.
- Os dados no gráfico de pizza são exibidos e agregados para um período de tempo personalizado das últimas 3 horas.
- A legenda do gráfico de pizza exibe os items configurados em 4 linhas.

Requisição:

```
{
    "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
}
```


Resposta:


```
{
    "jsonrpc": "2.0",
```

```
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Veja também

- Campo de widget do dashboard
- `dashboard.create`
- `dashboard.update`

21 Hosts com problemas

Descrição

Esses parâmetros e os possíveis valores de propriedade para os respectivos objetos de campo do widget de dashboard permitem configurar o widget *Hosts com problemas* nos métodos `dashboard.create` e `dashboard.update`.

Attention:
As propriedades `fields` do widget não são validadas durante a criação ou atualização de um dashboard. Isso permite que os usuários modifiquem widgets internos e criem widgets personalizados, mas também introduz o risco de criar ou atualizar widgets incorretamente. Para garantir a criação ou atualização bem-sucedida do widget *Hosts com problemas*, consulte o comportamento dos parâmetros descrito nas tabelas abaixo.

Parâmetros

Os seguintes parâmetros são suportados para o widget *Hosts com problemas*.

Parâmetro	type	name	value
Intervalo de atualização	0	rf_rate	0 - Sem atualização; 10 - 10 segundos; 30 - 30 segundos; 60 - *(padrão)* 1 minuto; 120 - 2 minutos; 600 - 10 minutos; 900 - 15 minutos.
Grupos de hosts	2	groupids.0	ID do Grupo de hosts. Nota: Para configurar múltiplos grupos de hosts, crie um objeto de campo de widget de dashboard para cada grupo de hosts com um número incrementado no nome da propriedade. Este parâmetro não é suportado se o widget estiver sendo configurado em um dashboard de template.
Grupos de hosts (Widget)	1	groupids._reference	Em vez do ID do Grupo de hosts: `ABCDE._hostgroupids` - defina um widget compatível (com seu parâmetro *Referência* definido como "ABCDE") como fonte de dados para grupos de hosts. Este parâmetro não é suportado se o widget estiver sendo configurado em um dashboard de template.
Excluir grupos de hosts	2	exclude_groupids.0	ID do Grupo de hosts. Nota: Para excluir múltiplos grupos de hosts, crie um objeto de campo de widget de dashboard para cada grupo de hosts com um número incrementado no nome da propriedade. Este parâmetro não é suportado se o widget estiver sendo configurado em um dashboard de template.

Parâmetro	type	name	value
Hosts	3	hostids.0	ID do Host. Nota: Para configurar múltiplos hosts, crie um objeto de campo de widget de dashboard para cada host com um número incrementado no nome da propriedade. Para múltiplos hosts, o parâmetro *Grupos de hosts* deve não estar configurado ou estar configurado com pelo menos um grupo de hosts ao qual os hosts configurados pertencem. Este parâmetro não é suportado se o widget estiver sendo configurado em um dashboard de template.
Hosts (Widget/Dashboard)	1	hostids._reference	Em vez do ID do Host: `DASHBOARD.hostids` - defina o seletor *Host* como fonte de dados para hosts; `ABCDE._hostids` - defina um widget compatível (com seu parâmetro *Referência* definido como "`ABCDE`") como fonte de dados para hosts. Este parâmetro não é suportado se o widget estiver sendo configurado em um dashboard de template.
Problema	1	problem	Nome do evento do problema (não diferencia maiúsculas de minúsculas, nome completo ou parte dele).
Gravidade	0	severities.0	0 - Não classificado; 1 - Informação; 2 - Aviso; 3 - Médio; 4 - Alto; 5 - Desastre. Padrão: vazio (todos habilitados). Nota: Para configurar múltiplos valores, crie um objeto de campo de widget de dashboard para cada valor com um número incrementado no nome da propriedade.
Tags do problema			
Tipo de avaliação	0	evaltype	0 - *(padrão)* E/Ou; 2 - Ou.
Nome da tag	1	tags.0.tag	Qualquer valor de string. Nota: O número no nome da propriedade faz referência à ordem da tag na lista de avaliação de tags. Comportamento do parâmetro: - *obrigatório* se estiver configurando *Tags do problema*
Operador	0	tags.0.operator	0 - Contém; 1 - Igual; 2 - Não contém; 3 - Diferente; 4 - Existe; 5 - Não existe. Nota: O número no nome da propriedade faz referência à ordem da tag na lista de avaliação de tags. Comportamento do parâmetro: - *obrigatório* se estiver configurando *Tags do problema*

Parâmetro	type	name	value
Valor da tag	1	tags.0.value	Qualquer valor de string. Nota: O número no nome da propriedade faz referência à ordem da tag na lista de avaliação de tags. Comportamento do parâmetro: - *obrigatório* se estiver configurando *Tags do problema*
Exibir problemas suprimidos	0	show_suppressed	0 - *(padrão)* Desabilitado; 1 - Habilitado.
Ocultar grupos sem problemas	0	hide_empty_groups	0 - *(padrão)* Desabilitado; 1 - Habilitado. Este parâmetro não é suportado se o widget estiver sendo configurado em um dashboard de template.
Exibição do problema	0	ext_ack	0 - *(padrão)* Todos; 1 - Apenas não reconhecidos; 2 - Separados.
Referência	1	reference	Qualquer valor de string composto por 5 caracteres (por exemplo, `ABCDE` ou `JBPNL`). Este valor deve ser exclusivo dentro do dashboard ao qual o widget pertence. Comportamento do parâmetro: - *obrigatório*

Exemplos

Os exemplos a seguir têm como objetivo apenas descrever a configuração dos objetos de campo do widget de dashboard para o widget *Hosts com problemas*. Para obter mais informações sobre como configurar um dashboard, consulte `dashboard.create`.

Configurando um widget *Problem hosts*

Configure um widget *Problem hosts* que exibe hosts dos grupos de hosts ”2” e ”4” que possuem problemas com um nome que inclua a string ”CPU” e que tenham as seguintes severidades: ”Warning”, ”Average”, ”High”, ”Disaster”.

Requisição:

```
{
    "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
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
```

```
        ]
    },
    "id": 1
}
```

Consulte também

- Dashboard widget field
- `dashboard.create`
- `dashboard.update`

22 Problemas

Descrição

Esses parâmetros e os possíveis valores de propriedade para os respectivos objetos de campo do widget de dashboard permitem configurar o widget *Problemas* nos métodos `dashboard.create` e `dashboard.update`.

Attention:
As propriedades `fields` do widget não são validadas durante a criação ou atualização de um dashboard. Isso permite que os usuários modifiquem widgets internos e criem widgets personalizados, mas também introduz o risco de criar ou atualizar widgets incorretamente. Para garantir a criação ou atualização bem-sucedida do widget *Problemas*, consulte o comportamento do parâmetro descrito nas tabelas abaixo.

Parâmetros

Os seguintes parâmetros são suportados para o widget *Problemas*.

Parâmetro	tipo	nome	valor
Intervalo de atualização	0	rf_rate	0 - Sem atualização; 10 - 10 segundos; 30 - 30 segundos; 60 - *(padrão)* 1 minuto; 120 - 2 minutos; 600 - 10 minutos; 900 - 15 minutos.
Mostrar	0	show	1 - *(padrão)* Problemas recentes; 2 - Histórico; 3 - Problemas.
Grupos de hosts	2	groupids.0	ID do Grupo de hosts. Nota: Para configurar múltiplos grupos de hosts, crie um objeto de campo de widget de dashboard para cada grupo de hosts com um número incrementado no nome da propriedade. Este parâmetro não é suportado se estiver configurando o widget em um dashboard de template.
Grupos de hosts (Widget)	1	groupids._reference	Em vez do ID do Grupo de hosts: `ABCDE._hostgroupids` - defina um widget compatível (com seu parâmetro *Referência* definido como "ABCDE") como a fonte de dados para grupos de hosts. Este parâmetro não é suportado se estiver configurando o widget em um dashboard de template.
Excluir grupos de hosts	2	exclude_groupids.0	ID do Grupo de hosts. Nota: Para excluir múltiplos grupos de hosts, crie um objeto de campo de widget de dashboard para cada grupo de hosts com um número incrementado no nome da propriedade. Este parâmetro não é suportado se estiver configurando o widget em um dashboard de template.

Parâmetro	tipo	nome	valor
Hosts	3	hostids.0	ID do Host. Nota: Para configurar múltiplos hosts, crie um objeto de campo de widget de dashboard para cada host com um número incrementado no nome da propriedade. Para múltiplos hosts, o parâmetro *Grupos de hosts* deve não estar configurado ou estar configurado com pelo menos um grupo de hosts ao qual os hosts configurados pertencem. Este parâmetro não é suportado se estiver configurando o widget em um dashboard de template.
Hosts (Widget/Dashboard)	1	hostids._reference	Em vez do ID do Host: `DASHBOARD.hostids` - defina o seletor *Host* como a fonte de dados para hosts; `ABCDE._hostids` - defina um widget compatível (com seu parâmetro *Referência* definido como "`ABCDE`") como a fonte de dados para hosts. Este parâmetro não é suportado se estiver configurando o widget em um dashboard de template.
Problema	1	problem	Nome do evento do problema (não diferencia maiúsculas de minúsculas, nome completo ou parte dele).
Gravidade	0	severities.0	0 - Não classificado; 1 - Informação; 2 - Aviso; 3 - Médio; 4 - Alto; 5 - Desastre. Padrão: vazio (todos habilitados). Nota: Para configurar múltiplos valores, crie um objeto de campo de widget de dashboard para cada valor com um número incrementado no nome da propriedade.
Tags do problema			
Tipo de avaliação	0	evaltype	0 - *(padrão)* E/Ou; 2 - Ou.
Nome da tag	1	tags.0.tag	Qualquer valor de string. Nota: O número no nome da propriedade faz referência à ordem da tag na lista de avaliação de tags. Comportamento do parâmetro: - *obrigatório* se estiver configurando *Tags do problema*
Operador	0	tags.0.operator	0 - Contém; 1 - Igual; 2 - Não contém; 3 - Não é igual; 4 - Existe; 5 - Não existe. Nota: O número no nome da propriedade faz referência à ordem da tag na lista de avaliação de tags. Comportamento do parâmetro: - *obrigatório* se estiver configurando *Tags do problema*

Parâmetro	tipo	nome	valor
Valor da tag	1	tags.0.value	Qualquer valor de string. Nota: O número no nome da propriedade faz referência à ordem da tag na lista de avaliação de tags. Comportamento do parâmetro: - *obrigatório* se estiver configurando *Tags do problema*
Mostrar tags	0	show_tags	0 - *(padrão)* Nenhuma; 1 - 1; 2 - 2; 3 - 3.
Nome da tag (formato)	0	tag_name_format	0 - *(padrão)* Completo; 1 - Abreviado; 2 - Nenhum. Comportamento do parâmetro: - *suportado* se *Mostrar tags* estiver definido como ”1”, ”2” ou ”3”
Prioridade de exibição da tag	1	tag_priority	Lista de tags separadas por vírgula. Comportamento do parâmetro: - *suportado* se *Mostrar tags* estiver definido como ”1”, ”2” ou ”3”
Mostrar dados operacionais	0	show_opdata	0 - *(padrão)* Nenhum; 1 - Separadamente; 2 - Com o nome do problema.
Mostrar problemas suprimidos	0	show_suppressed	0 - *(padrão)* Desabilitado; 1 - Habilitado.
Status de reconhecimento	0	acknowledgement_status	0 - *(padrão)* todos; 1 - Não reconhecido; 2 - Reconhecido.
Por mim	0	acknowledged_by_me	0 - *(padrão)* Desabilitado; 1 - Habilitado.
Ordenar entradas por	0	sort_triggers	1 - Gravidade (decrescente); 2 - Host (crescente); 3 - Hora (crescente); 4 - *(padrão)* Hora (decrescente); 13 - Gravidade (crescente); 14 - Host (decrescente); 15 - Problema (crescente); 16 - Problema (decrescente). Para todos os valores, exceto ”Hora (decrescente)” e ”Hora (crescente)”, o parâmetro *Mostrar linha do tempo* deve ser definido como ”Desabilitado”. Os valores ”Host (crescente)” e ”Host (decrescente)” não são suportados se estiver configurando o widget em um dashboard de template.

Parâmetro	tipo	nome	valor
Mostrar linha do tempo	0	show_timeline	0 - Desabilitado; 1 - *(padrão)* Habilitado. Comportamento do parâmetro: - *suportado* se *Ordenar entradas por* estiver definido como ”Hora (decrescente)” ou ”Hora (crescente)”
Destacar linha in-teira	0	highlight_row	0 - *(padrão)* Desabilitado; 1 - Habilitado. Este parâmetro não é suportado nos temas de alto contraste. Esta opção é suportada desde o Zabbix 7.0.19.
Mostrar lin-has	0	show_lines	Os valores possíveis variam de 1-100. Padrão: 25.
Referência	1	reference	Qualquer valor de string consistindo de 5 caracteres (por exemplo, `ABCDE` ou `JBPNL`). Este valor deve ser exclusivo dentro do dashboard ao qual o widget pertence. Comportamento do parâmetro: - *obrigatório*

Exemplos

Os exemplos a seguir têm como objetivo apenas descrever a configuração dos objetos de campo do widget de dashboard para o widget *Problemas*. Para obter mais informações sobre como configurar um dashboard, consulte `dashboard.create`.

Configurando um widget *Problems*

Configure um widget *Problems* que exiba problemas para o grupo de hosts ”4” que satisfaçam as seguintes condições:

- Problemas que tenham uma tag com o nome ”scope” que contenha os valores ”performance” ou ”availability”, ou ”capacity”.
- Problemas que tenham as seguintes severidades: ”Warning”, ”Average”, ”High”, ”Disaster”.

Além disso, configure o widget para mostrar tags e dados operacionais.

Requisição:

```
{
    "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
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Veja também

- Campo de widget do dashboard
- `dashboard.create`
- `dashboard.update`

23 Problemas por gravidade

Descrição

Esses parâmetros e os possíveis valores de propriedade para os respectivos objetos de campo do widget do dashboard permitem configurar o widget *Problemas por severidade* nos métodos `dashboard.create` e `dashboard.update`.

Attention:
As propriedades `fields` do widget não são validadas durante a criação ou atualização de um dashboard. Isso permite que os usuários modifiquem widgets internos e criem widgets personalizados, mas também introduz o risco de criar ou atualizar widgets incorretamente. Para garantir a criação ou atualização bem-sucedida do widget *Problemas por severidade*, consulte o comportamento dos parâmetros descrito nas tabelas abaixo.

Parâmetros

Os seguintes parâmetros são suportados para o widget *Problemas por severidade*.

Parâmetro	tipo	nome	valor
Intervalo de atualização	0	rf_rate	0 - Sem atualização; 10 - 10 segundos; 30 - 30 segundos; 60 - *(padrão)* 1 minuto; 120 - 2 minutos; 600 - 10 minutos; 900 - 15 minutos.
Grupos de hosts	2	groupids.0	ID do Grupo de hosts. Nota: Para configurar vários grupos de hosts, crie um objeto de campo de widget de dashboard para cada grupo de hosts com um número incrementado no nome da propriedade. Este parâmetro não é suportado se o widget estiver sendo configurado em um dashboard de template.
Grupos de hosts (Widget)	1	groupids._reference	Em vez do ID do Grupo de hosts: `ABCDE._hostgroupids` - defina um widget compatível (com seu parâmetro *Referência* definido como "`ABCDE`") como a fonte de dados para grupos de hosts. Este parâmetro não é suportado se o widget estiver sendo configurado em um dashboard de template.
Excluir grupos de hosts	2	exclude_groupids.0	ID do Grupo de hosts. Nota: Para excluir vários grupos de hosts, crie um objeto de campo de widget de dashboard para cada grupo de hosts com um número incrementado no nome da propriedade. Este parâmetro não é suportado se o widget estiver sendo configurado em um dashboard de template.
Hosts	3	hostids.0	ID do Host. Nota: Para configurar vários hosts, crie um objeto de campo de widget de dashboard para cada host com um número incrementado no nome da propriedade. Para múltiplos hosts, o parâmetro *Grupos de hosts* deve não estar configurado ou estar configurado com pelo menos um grupo de hosts ao qual os hosts configurados pertencem. Este parâmetro não é suportado se o widget estiver sendo configurado em um dashboard de template.
Hosts (Widget/Dashboard)	1	hostids._reference	Em vez do ID do Host: `DASHBOARD.hostids` - defina o seletor *Host* como a fonte de dados para hosts; `ABCDE._hostids` - defina um widget compatível (com seu parâmetro *Referência* definido como "`ABCDE`") como a fonte de dados para hosts. Este parâmetro não é suportado se o widget estiver sendo configurado em um dashboard de template.
Problema	1	problem	Nome do evento do problema (não diferencia maiúsculas de minúsculas, nome completo ou parte dele).

Parâmetro	tipo	nome	valor
Severidade	0	severities.0	0 - Não classificado; 1 - Informação; 2 - Aviso; 3 - Médio; 4 - Alto; 5 - Desastre. Padrão: vazio (todas habilitadas). Nota: Para configurar vários valores, crie um objeto de campo de widget de dashboard para cada valor com um número incrementado no nome da propriedade.
Tags do problema			
Tipo de avaliação	0	evaltype	0 - *(padrão)* E/Ou; 2 - Ou.
Nome da tag	1	tags.0.tag	Qualquer valor de string. Nota: O número no nome da propriedade faz referência à ordem da tag na lista de avaliação de tags. Comportamento do parâmetro: - *obrigatório* se estiver configurando *Tags do problema*
Operador	0	tags.0.operator	0 - Contém; 1 - Igual; 2 - Não contém; 3 - Diferente; 4 - Existe; 5 - Não existe. Nota: O número no nome da propriedade faz referência à ordem da tag na lista de avaliação de tags. Comportamento do parâmetro: - *obrigatório* se estiver configurando *Tags do problema*
Valor da tag	1	tags.0.value	Qualquer valor de string. Nota: O número no nome da propriedade faz referência à ordem da tag na lista de avaliação de tags. Comportamento do parâmetro: - *obrigatório* se estiver configurando *Tags do problema*
Mostrar	0	show_type	0 - *(padrão)* Grupos de hosts; 1 - Totais. Este parâmetro não é suportado se o widget estiver sendo configurado em um dashboard de template, e por padrão está definido como "Totais".
Layout	0	layout	0 - *(padrão)* Horizontal; 1 - Vertical. Comportamento do parâmetro: - *suportado* se *Mostrar* estiver definido como "Totais"
Mostrar dados operacionais	0	show_opdata	0 - *(padrão)* Nenhum; 1 - Separadamente; 2 - Com o nome do problema.

Parâmetro	tipo	nome	valor
Mostrar problemas suprimidos	0	show_suppressed	0 - *(padrão)* Desabilitado; 1 - Habilitado.
Ocultar grupos sem problemas	0	hide_empty_groups	0 - *(padrão)* Desabilitado; 1 - Habilitado. Comportamento do parâmetro: - *suportado* se *Mostrar* estiver definido como "Grupos de hosts" Este parâmetro não é suportado se o widget estiver sendo configurado em um dashboard de template.
Exibição do problema	0	ext_ack	0 - *(padrão)* Todos; 1 - Apenas não reconhecidos; 2 - Separados.
Mostrar linha do tempo	0	show_timeline	0 - Desabilitado; 1 - *(padrão)* Habilitado.
Referência	1	reference	Qualquer valor de string composto por 5 caracteres (por exemplo, `ABCDE` ou `JBPNL`). Este valor deve ser exclusivo dentro do dashboard ao qual o widget pertence. Comportamento do parâmetro: - *obrigatório*

Exemplos

Os exemplos a seguir têm como objetivo apenas descrever a configuração dos objetos de campo do widget do painel para o widget *Problemas por severidade*. Para obter mais informações sobre como configurar um painel, consulte `dashboard.create`.

Configurando um widget *Problemas por severidade*

Configure um widget *Problemas por severidade* que exibe totais de problemas para todos os grupos de hosts.

Requisição:

```
{
    "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
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Veja também

- Campo de widget de dashboard
- `dashboard.create`
- `dashboard.update`

24 Relatório de SLA

Descrição

Esses parâmetros e os possíveis valores de propriedade para os respectivos objetos de campo de widget de dashboard permitem configurar o widget *Relatório de SLA* nos métodos `dashboard.create` e `dashboard.update`.

Attention:
As propriedades `fields` do widget não são validadas durante a criação ou atualização de um dashboard. Isso permite que os usuários modifiquem widgets internos e criem widgets personalizados, mas também introduz o risco de criar ou atualizar widgets incorretamente. Para garantir a criação ou atualização bem-sucedida do widget *Relatório de SLA*, consulte o comportamento dos parâmetros descritos nas tabelas abaixo.

Parâmetros

Os seguintes parâmetros são suportados para o widget *Relatório de SLA*.

Parâmetro	type	name	value
Intervalo de atualização	0	rf_rate	0 - *(padrão)* Sem atualização; 10 - 10 segundos; 30 - 30 segundos; 60 - 1 minuto; 120 - 2 minutos; 600 - 10 minutos; 900 - 15 minutos.
SLA	10	slaid.0	ID do SLA. Comportamento do parâmetro: - *obrigatório*
Serviço	9	serviceid.0	ID do Serviço.
Mostrar períodos	0	show_periods	Os valores possíveis variam de 1 a 100. Padrão: 20.
De	1	date_from	String de data válida no formato `YYYY-MM-DD`. Datas relativas com modificadores `d`, `w`, `M`, `y` (por exemplo, `now`, `now/d`, `now/w-1w`, etc.) são suportadas.
Até	1	date_to	String de data válida no formato `YYYY-MM-DD`. Datas relativas com modificadores `d`, `w`, `M`, `y` (por exemplo, `now`, `now/d`, `now/w-1w`, etc.) são suportadas.

Exemplos

Os exemplos a seguir têm como objetivo apenas descrever a configuração dos objetos de campo do widget do painel para o widget *Relatório de SLA*. Para obter mais informações sobre como configurar um painel, consulte `dashboard.create`.

Configurando um widget *Relatório de SLA*

Configure um widget *Relatório de SLA* que exibe o relatório de SLA para o serviço ”2” do SLA ”4” para o período dos últimos 30 dias.

Requisição:

```
{
    "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
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Consulte também

- Dashboard widget field
- `dashboard.create`
- `dashboard.update`

25 Informações do sistema

Descrição

Esses parâmetros e os possíveis valores de propriedade para os respectivos objetos de campo do widget do dashboard permitem configurar o widget *Informações do sistema* nos métodos `dashboard.create` e `dashboard.update`.

Attention:
As propriedades `fields` do widget não são validadas durante a criação ou atualização de um dashboard. Isso permite que os usuários modifiquem widgets internos e criem widgets personalizados, mas também introduz o risco de criar ou atualizar widgets incorretamente. Para garantir a criação ou atualização bem-sucedida do widget *Informações do sistema*, consulte o comportamento dos parâmetros descrito nas tabelas abaixo.

Parâmetros

Os seguintes parâmetros são suportados para o widget *Informações do sistema*.

Parâmetro	tipo	nome	valor
Intervalo de atualização	0	rf_rate	0 - Sem atualização; 10 - 10 segundos; 30 - 30 segundos; 60 - 1 minuto; 120 - 2 minutos; 600 - 10 minutos; 900 - *(padrão)* 15 minutos.
Mostrar	0	info_type	0 - *(padrão)* Estatísticas do sistema; 1 - Nós de alta disponibilidade.
Mostrar detalhes da verificação de atualização de software	0	show_software_update_check_details	0 - *(padrão)* Desabilitado; 1 - Habilitado. Comportamento do parâmetro: - *suportado* se `AllowSoftwareUpdateCheck` na configuração do Zabbix server estiver habilitado e *Mostrar* estiver definido como "Estatísticas do sistema"

Exemplos

Os exemplos a seguir visam apenas descrever a configuração dos objetos de campo do widget do painel para o widget *Informações do sistema*. Para obter mais informações sobre como configurar um painel, consulte `dashboard.create`.

Configurando um widget *Informações do sistema*

Configure um widget *Informações do sistema* que exibe estatísticas do sistema com um intervalo de atualização de 10 minutos e verificação de atualização de software ativada.

Requisição:

```
{
    "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
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Veja também

- Campo do widget do dashboard
- `dashboard.create`
- `dashboard.update`

26 Principais hosts

Descrição

Esses parâmetros e os possíveis valores de propriedade para os respectivos objetos de campo de widget de dashboard permitem configurar o widget *Top hosts* nos métodos `dashboard.create` e `dashboard.update`.

Attention:
As propriedades `fields` do widget não são validadas durante a criação ou atualização de um dashboard. Isso permite que os usuários modifiquem widgets internos e criem widgets personalizados, mas também introduz o risco de criar ou atualizar widgets incorretamente. Para garantir a criação ou atualização bem-sucedida do widget *Top hosts*, consulte o comportamento dos parâmetros descrito nas tabelas abaixo.

Parâmetros

Os seguintes parâmetros são suportados para o widget *Top hosts*.

Parâmetro	tipo	nome	valor
Intervalo de atualização	0	rf_rate	0 - Sem atualização; 10 - 10 segundos; 30 - 30 segundos; 60 - *(padrão)* 1 minuto; 120 - 2 minutos; 600 - 10 minutos; 900 - 15 minutos.

Parâmetro	tipo	nome	valor
Grupos de hosts	2	groupids.0	ID do Grupo de hosts. Nota: Para configurar vários grupos de hosts, crie um objeto de campo de widget de dashboard para cada grupo de hosts com um número incrementado no nome da propriedade. Este parâmetro não é suportado se o widget estiver sendo configurado em um dashboard de template.
Grupos de hosts (Widget)	1	groupids._reference	Em vez do ID do Grupo de hosts: `ABCDE._hostgroupids` - defina um widget compatível (com seu parâmetro *Referência* definido como "`ABCDE`") como a fonte de dados para grupos de hosts. Este parâmetro não é suportado se o widget estiver sendo configurado em um dashboard de template.
Hosts	3	hostids.0	ID do Host. Nota: Para configurar vários hosts, crie um objeto de campo de widget de dashboard para cada host com um número incrementado no nome da propriedade. Para múltiplos hosts, o parâmetro *Grupos de hosts* deve não estar configurado ou estar configurado com pelo menos um grupo de hosts ao qual os hosts configurados pertençam. Este parâmetro não é suportado se o widget estiver sendo configurado em um dashboard de template.
Hosts (Widget/Dashboard)	1	hostids._reference	Em vez do ID do Host: `DASHBOARD.hostids` - defina o seletor *Host* como a fonte de dados para hosts; `ABCDE._hostids` - defina um widget compatível (com seu parâmetro *Referência* definido como "`ABCDE`") como a fonte de dados para hosts. Este parâmetro não é suportado se o widget estiver sendo configurado em um dashboard de template.
Tags de host			
Tipo de avaliação	0	evaltype	0 - *(padrão)* E/Ou; 2 - Ou. Este parâmetro não é suportado se o widget estiver sendo configurado em um dashboard de template.
Nome da tag	1	tags.0.tag	Qualquer valor de string. Nota: O número no nome da propriedade faz referência à ordem da tag na lista de avaliação de tags. Comportamento do parâmetro: - *obrigatório* se estiver configurando *Tags de host* Este parâmetro não é suportado se o widget estiver sendo configurado em um dashboard de template.

Parâmetro	tipo	nome	valor
Operador	0	tags.0.operator	0 - Contém; 1 - Igual; 2 - Não contém; 3 - Não é igual; 4 - Existe; 5 - Não existe. Nota: O número no nome da propriedade faz referência à ordem da tag na lista de avaliação de tags. Comportamento do parâmetro: - *obrigatório* se estiver configurando *Tags de host* Este parâmetro não é suportado se o widget estiver sendo configurado em um dashboard de template.
Valor da tag	1	tags.0.value	Qualquer valor de string. Nota: O número no nome da propriedade faz referência à ordem da tag na lista de avaliação de tags. Comportamento do parâmetro: - *obrigatório* se estiver configurando *Tags de host* Este parâmetro não é suportado se o widget estiver sendo configurado em um dashboard de template.
Mostrar hosts em manutenção	0	maintenance	0 - *(padrão)* Desabilitado; 1 - Habilitado.
Colunas (veja abaixo)			
Ordenar por	0	column	Valor numérico da coluna das colunas configuradas.
Ordem	0	order	2 - *(padrão)* Top N; 3 - Bottom N.
Limite de hosts	0	show_lines	Os valores possíveis variam de 1-100. Padrão: 10. Este parâmetro não é suportado se o widget estiver sendo configurado em um dashboard de template.

Colunas

As colunas possuem parâmetros comuns e parâmetros adicionais dependendo da configuração do parâmetro *Data*.

Note:
Para todos os parâmetros relacionados às colunas, o número no nome da propriedade (por exemplo, columns.0.name) faz referência a uma coluna para a qual o parâmetro está configurado.

Os seguintes parâmetros são suportados para todas as colunas.

Parâmetro	tipo	nome	valor
Nome	1	columns.0.name	Qualquer valor de string. Comportamento do parâmetro: - *obrigatório*

Parâmetro	tipo	nome	valor
Dados	0	columns.0.data	1 - Valor do item; 2 - Nome do host; 3 - Texto. Comportamento do parâmetro: - *obrigatório*
Cor base	1	columns.0.base_color	Código de cor hexadecimal (por exemplo, FF0000). Comportamento do parâmetro: - *obrigatório*

Valor do item

Os seguintes parâmetros são suportados se *Dados* estiver definido como ”Valor do item”.

Note:
O primeiro número no nome da propriedade *Thresholds* (por exemplo, columnsthresholds.0.color.0) faz referência à coluna para a qual os thresholds estão configurados, enquanto o segundo número faz referência à posição do threshold em uma lista, ordenada em ordem crescente. No entanto, se os thresholds forem configurados em uma ordem diferente, os valores serão ordenados em ordem crescente após a atualização da configuração do widget no frontend do Zabbix (por exemplo, `"threshold.0.threshold":"5"` → `"threshold.0.threshold":"1"`; `"threshold.1.threshold":"1"` → `"threshold.1.threshold": "5"`).

Parâmetro	tipo	nome	valor
Item	1	columns.0.item	Nome de item válido. Ao configurar o widget em um dashboard de template, apenas os items configurados no template devem ser definidos.
Exibir	0	columns.0.display	1 - *(padrão)* Como está; 2 - Barra; 3 - Indicadores.
Mínimo	1	columns.0.min	Qualquer valor numérico. Comportamento do parâmetro: - *suportado* se *Exibir* estiver definido como ”Barra” ou ”Indicadores”
Máximo	1	columns.0.max	Qualquer valor numérico. Comportamento do parâmetro: - *suportado* se *Exibir* estiver definido como ”Barra” ou ”Indicadores”
Casas decimais	0	columns.0.decimal_places	Os valores possíveis variam de 0 a 10. Padrão: 2.
Thresholds			
Cor	1	columnsthresholds.0.color.0	Código de cor hexadecimal (por exemplo, FF0000). Padrão: "" (vazio).
Threshold	1	columnsthresholds.0.threshold.0	Qualquer valor de string.
Função de agregação	0	columns.0.aggregate_function	0 - *(padrão)* não usado; 1 - min; 2 - max; 3 - avg; 4 - count; 5 - sum; 6 - first; 7 - last.

Parâmetro	tipo	nome	valor
Período de tempo	1	columns.0.time_period.reference	`DASHBOARD._timeperiod` - define o seletor de *Período de tempo* como fonte de dados; `ABCDE._timeperiod` - define um widget compatível (com seu parâmetro `reference` igual a `ABCDE`) como fonte de dados. Padrão: `DASHBOARD._timeperiod` Alternativamente, você pode definir o período de tempo apenas nos parâmetros *De* e *Até*. Comportamento do parâmetro: - *suportado* se *Função de agregação* estiver definida como ”min”, ”max”, ”avg”, ”count”, ”sum”, ”first”, ”last”
De	1	columns.0.time_period.from	String de tempo válida no formato absoluto (`YYYY-MM-DD hh:mm:ss`) ou sintaxe de tempo relativa (`now`, `now/d`, `now/w-1w`, etc.). Comportamento do parâmetro: - *suportado* se *Período de tempo* não estiver definido e *Função de agregação* estiver definida como ”min”, ”max”, ”avg”, ”count”, ”sum”, ”first”, ”last” - *obrigatório* se `columns.0.time_period.to` estiver definido
Até	1	columns.0.time_period.to	String de tempo válida no formato absoluto (`YYYY-MM-DD hh:mm:ss`) ou sintaxe de tempo relativa (`now`, `now/d`, `now/w-1w`, etc.). Comportamento do parâmetro: - *suportado* se *Período de tempo* não estiver definido e *Função de agregação* estiver definida como ”min”, ”max”, ”avg”, ”count”, ”sum”, ”first”, ”last” - *obrigatório* se `columns.0.time_period.from` estiver definido
Dados históricos	0	columns.0.history	0 - *(padrão)* Auto; 1 - Histórico; 2 - Tendências.
Referência	1	reference	Qualquer valor de string composto por 5 caracteres (por exemplo, `ABCDE` ou `JBPNL`). Este valor deve ser exclusivo dentro do dashboard ao qual o widget pertence. Comportamento do parâmetro: - *obrigatório*

Texto

Os seguintes parâmetros são suportados se *Dados* estiver definido como ”Texto”.

Parâmetro	tipo	nome	valor
Texto	1	columns.0.text	Qualquer valor de string, incluindo macros. Macros suportadas: {HOST.*}, {INVENTORY.*}. Comportamento do parâmetro: - *obrigatório* se *Dados* estiver definido como ”Texto”

Exemplos

Os exemplos a seguir têm como objetivo apenas descrever a configuração dos objetos de campo do widget de dashboard para o widget *Top hosts*. Para obter mais informações sobre como configurar um dashboard, consulte `dashboard.create`.

Configurando um widget *Top hosts*

Configure um widget *Top hosts* que exibe os principais hosts por utilização de CPU no grupo de hosts ”4”. Além disso, configure as seguintes colunas personalizadas: ”Nome do host”, ”Utilização da CPU em %”, ”Média de 1m”, ”Média de 5m”, ”Média de 15m”, ”Processos”.

Requisição:

```
{
    "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
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Veja também

- Campo do widget do dashboard
- `dashboard.create`
- `dashboard.update`

27 Principais triggers

Descrição

Esses parâmetros e os possíveis valores de propriedade para os respectivos objetos de campo do widget do dashboard permitem configurar o widget *Top triggers* nos métodos `dashboard.create` e `dashboard.update`.

Attention:
As propriedades `fields` do widget não são validadas durante a criação ou atualização de um dashboard. Isso permite que os usuários modifiquem widgets internos e criem widgets personalizados, mas também introduz o risco de criar ou atualizar widgets incorretamente. Para garantir a criação ou atualização bem-sucedida do widget *Top triggers*, consulte o comportamento dos parâmetros descritos nas tabelas abaixo.

Parâmetros

Os seguintes parâmetros são suportados para o widget *Top triggers*.

Parâmetro	type	name	value
Intervalo de atualização	0	rf_rate	0 - *(padrão)* Sem atualização; 10 - 10 segundos; 30 - 30 segundos; 60 - 1 minuto; 120 - 2 minutos; 600 - 10 minutos; 900 - 15 minutos.
Grupos de hosts	2	groupids.0	ID do grupo de hosts. Nota: Para configurar vários grupos de hosts, crie um objeto de campo de widget de dashboard para cada grupo de hosts com um número incrementado no nome da propriedade. Este parâmetro não é suportado se o widget estiver sendo configurado em um dashboard de template.
Hosts	3	hostids.0	ID do host. Nota: Para configurar vários hosts, crie um objeto de campo de widget de dashboard para cada host com um número incrementado no nome da propriedade. Para múltiplos hosts, o parâmetro *Grupos de hosts* deve não estar configurado ou estar configurado com pelo menos um grupo de hosts ao qual os hosts configurados pertençam. Este parâmetro não é suportado se o widget estiver sendo configurado em um dashboard de template.
Problema	1	problem	Nome do evento do problema (não diferencia maiúsculas de minúsculas, nome completo ou parte dele).
Severidade	0	severities.0	Severidades do trigger. 0 - Não classificado; 1 - Informação; 2 - Aviso; 3 - Médio; 4 - Alto; 5 - Desastre. Padrão: vazio (todas habilitadas). Nota: Para configurar múltiplos valores, crie um objeto de campo de widget de dashboard para cada valor com um número incrementado no nome da propriedade.
Tags do problema			
Tipo de avaliação	0	evaltype	0 - *(padrão)* E/Ou; 2 - Ou.
Nome da tag	1	tags.0.tag	Qualquer valor de string. Nota: O número no nome da propriedade faz referência à ordem da tag na lista de avaliação de tags. Comportamento do parâmetro: - *obrigatório* se estiver configurando *Tags do problema*

Parâmetro	type	name	value
Operador	0	tags.0.operator	0 - Contém; 1 - Igual; 2 - Não contém; 3 - Não é igual; 4 - Existe; 5 - Não existe. Nota: O número no nome da propriedade faz referência à ordem da tag na lista de avaliação de tags. Comportamento do parâmetro: - *obrigatório* se estiver configurando *Tags do problema*
Valor da tag	1	tags.0.value	Qualquer valor de string. Nota: O número no nome da propriedade faz referência à ordem da tag na lista de avaliação de tags. Comportamento do parâmetro: - *obrigatório* se estiver configurando *Tags do problema*
Período de tempo	1	time_period._referenceDASHBOARD._timeperiod	`DASHBOARD._timeperiod` - define o seletor de *Período de tempo* como a fonte de dados; `ABCDE._timeperiod` - define um widget compatível (com seu parâmetro *Referência* definido como "ABCDE") como a fonte de dados. Padrão: `DASHBOARD._timeperiod` Alternativamente, você pode definir o período de tempo apenas nos parâmetros *De* e *Até*.
De	1	time_period.from	String de tempo válida no formato absoluto (`YYYY-MM-DD hh:mm:ss`) ou na sintaxe de tempo relativo (`now`, `now/d`, `now/w-1w`, etc.). Comportamento do parâmetro: - *suportado* se *Período de tempo* não estiver definido - *obrigatório* se `time_period.to` estiver definido
Até	1	time_period.to	String de tempo válida no formato absoluto (`YYYY-MM-DD hh:mm:ss`) ou na sintaxe de tempo relativo (`now`, `now/d`, `now/w-1w`, etc.). Comportamento do parâmetro: - *suportado* se *Período de tempo* não estiver definido - *obrigatório* se `time_period.from` estiver definido
Limite de triggers	0	show_lines	Os valores possíveis variam de 1 a 100. Padrão: 10.

Exemplos

Os exemplos a seguir têm como objetivo apenas descrever a configuração dos objetos de campo do widget de dashboard para o widget *Top triggers*. Para obter mais informações sobre como configurar um dashboard, consulte `dashboard.create`.

Configurando um widget *Top triggers*

Configure um widget *Top triggers* que exibe os 5 principais triggers para o grupo de hosts "4" com a contagem de todos os problemas para cada trigger. O widget exibe apenas triggers que possuem severidades "Warning", "Average", "High" ou "Disaster", e problemas que possuem uma tag com o nome "scope" que contenha os valores "performance" ou "availability", ou "capacity".

Requisição:

```
{
    "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
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Veja também

- Campo do widget do dashboard
- `dashboard.create`
- `dashboard.update`

28 Visão geral dos triggers

Descrição

Esses parâmetros e os possíveis valores de propriedade para os respectivos objetos de campo do widget de dashboard permitem configurar o widget *Visão geral de triggers* nos métodos `dashboard.create` e `dashboard.update`.

Attention:
As propriedades `fields` do widget não são validadas durante a criação ou atualização de um dashboard. Isso permite que os usuários modifiquem widgets internos e criem widgets personalizados, mas também introduz o risco de criar ou atualizar widgets incorretamente. Para garantir a criação ou atualização bem-sucedida do widget *Visão geral de triggers*, consulte o comportamento dos parâmetros descrito nas tabelas abaixo.

Parâmetros

Os seguintes parâmetros são suportados para o widget *Visão geral dos triggers*.

Parâmetro	tipo	nome	valor
Intervalo de atualização	0	rf_rate	0 - Sem atualização; 10 - 10 segundos; 30 - 30 segundos; 60 - *(padrão)* 1 minuto; 120 - 2 minutos; 600 - 10 minutos; 900 - 15 minutos.
Mostrar	0	show	1 - *(padrão)* Problemas recentes; 2 - Qualquer; 3 - Problemas.
Grupos de hosts	2	groupids.0	ID do Grupo de hosts. Nota: Para configurar múltiplos grupos de hosts, crie um objeto de campo de widget de dashboard para cada grupo de hosts com um número incrementado no nome da propriedade. Este parâmetro não é suportado se o widget estiver sendo configurado em um dashboard de template.
Grupos de hosts (Widget)	1	groupids._reference	Em vez do ID do Grupo de hosts: `ABCDE._hostgroupids` - defina um widget compatível (com seu parâmetro *Referência* definido como "ABCDE") como a fonte de dados para grupos de hosts. Este parâmetro não é suportado se o widget estiver sendo configurado em um dashboard de template.
Hosts	3	hostids.0	ID do Host. Nota: Para configurar múltiplos hosts, crie um objeto de campo de widget de dashboard para cada host com um número incrementado no nome da propriedade. Para múltiplos hosts, o parâmetro *Grupos de hosts* deve não estar configurado ou estar configurado com pelo menos um grupo de hosts ao qual os hosts configurados pertençam. Este parâmetro não é suportado se o widget estiver sendo configurado em um dashboard de template.

Parâmetro	tipo	nome	valor
Hosts (Widget/Dashboard)	1	hostids._reference	Em vez do ID do Host: `DASHBOARD.hostids` - defina o seletor *Host* como a fonte de dados para hosts; `ABCDE._hostids` - defina um widget compatível (com seu parâmetro *Referência* definido como "ABCDE") como a fonte de dados para hosts. Este parâmetro não é suportado se o widget estiver sendo configurado em um dashboard de template.
Tags de problema			
Tipo de avaliação	0	evaltype	0 - *(padrão)* E/Ou; 2 - Ou.
Nome da tag	1	tags.0.tag	Qualquer valor de string. Nota: O número no nome da propriedade faz referência à ordem da tag na lista de avaliação de tags. Comportamento do parâmetro: - *obrigatório* se estiver configurando *Tags de problema*
Operador	0	tags.0.operator	0 - Contém; 1 - Igual; 2 - Não contém; 3 - Não é igual; 4 - Existe; 5 - Não existe. Nota: O número no nome da propriedade faz referência à ordem da tag na lista de avaliação de tags. Comportamento do parâmetro: - *obrigatório* se estiver configurando *Tags de problema*
Valor da tag	1	tags.0.value	Qualquer valor de string. Nota: O número no nome da propriedade faz referência à ordem da tag na lista de avaliação de tags. Comportamento do parâmetro: - *obrigatório* se estiver configurando *Tags de problema*
Mostrar problemas suprimidos	0	show_suppressed	0 - *(padrão)* Desabilitado; 1 - Habilitado.
Localização dos hosts	0	style	0 - *(padrão)* Esquerda; 1 - Superior.

Exemplos

Os exemplos a seguir têm como objetivo apenas descrever a configuração dos objetos de campo do widget do painel para o widget *Visão geral dos triggers*. Para obter mais informações sobre como configurar um painel, consulte `dashboard.create`.

Configurando um widget *Visão geral dos triggers*

Configure um widget *Visão geral dos triggers* que exibe os estados dos triggers para todos os grupos de hosts que possuem triggers com uma tag cujo nome é "scope" e contém o valor "availability".

Requisição:

```
{
    "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
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
```

```
    },
    "id": 1
}
```

Veja também

- Campo do widget do dashboard
- `dashboard.create`
- `dashboard.update`

29 URL

Descrição

Esses parâmetros e os possíveis valores de propriedade para os respectivos objetos de campo do widget de dashboard permitem configurar o widget *URL* nos métodos `dashboard.create` e `dashboard.update`.

Attention:
As propriedades `fields` do widget não são validadas durante a criação ou atualização de um dashboard. Isso permite que os usuários modifiquem widgets internos e criem widgets personalizados, mas também introduz o risco de criar ou atualizar widgets incorretamente. Para garantir a criação ou atualização bem-sucedida do widget *URL*, consulte o comportamento do parâmetro descrito nas tabelas abaixo.

Parâmetros

Os seguintes parâmetros são suportados para o widget *URL*.

Parâmetro	tipo	nome	valor
Intervalo de atualização	0	rf_rate	0 - *(padrão)* Sem atualização; 10 - 10 segundos; 30 - 30 segundos; 60 - 1 minuto; 120 - 2 minutos; 600 - 10 minutos; 900 - 15 minutos.
URL	1	url	String de URL válida. Comportamento do parâmetro: - *obrigatório*
Substituir host	1	override_hostid._referenceABCDE	`ABCDE._hostid` - define um widget compatível (com seu parâmetro *Referência* definido como "ABCDE") como fonte de dados para hosts; `DASHBOARD._hostid` - define o seletor *Host* do dashboard como fonte de dados para hosts. Este parâmetro não é suportado se o widget estiver sendo configurado em um dashboard de template.

Exemplos

Os exemplos a seguir têm como objetivo apenas descrever a configuração dos objetos de campo do widget de dashboard para o widget *URL*. Para obter mais informações sobre como configurar um dashboard, consulte `dashboard.create`.

Configurando um widget *URL*

Configure um widget *URL* que exibe a página inicial do manual do Zabbix.

Requisição:

```
{
    "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/7.0/en"
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Veja também

- Campo do widget do dashboard
- `dashboard.create`
- `dashboard.update`

30 Monitoramento web

Descrição

Esses parâmetros e os possíveis valores de propriedade para os respectivos objetos de campo do widget de dashboard permitem

configurar o widget *Monitoramento web* nos métodos `dashboard.create` e `dashboard.update`.

Attention:
As propriedades `fields` do widget não são validadas durante a criação ou atualização de um dashboard. Isso permite que os usuários modifiquem widgets internos e criem widgets personalizados, mas também introduz o risco de criar ou atualizar widgets incorretamente. Para garantir a criação ou atualização bem-sucedida do widget *Monitoramento web*, consulte o comportamento dos parâmetros descrito nas tabelas abaixo.

Parâmetros

Os seguintes parâmetros são suportados para o widget *Web monitoring*.

Parâmetro	tipo	nome	valor
Intervalo de atualização	0	rf_rate	0 - Sem atualização; 10 - 10 segundos; 30 - 30 segundos; 60 - *(padrão)* 1 minuto; 120 - 2 minutos; 600 - 10 minutos; 900 - 15 minutos.
Grupos de hosts	2	groupids.0	ID do Grupo de hosts. Nota: Para configurar vários grupos de hosts, crie um objeto de campo de widget de dashboard para cada grupo de hosts com um número incrementado no nome da propriedade. Este parâmetro não é suportado se o widget estiver sendo configurado em um dashboard de template.
Grupos de hosts (Widget)	1	groupids._reference	Em vez do ID do Grupo de hosts: `ABCDE._hostgroupids` - defina um widget compatível (com seu parâmetro *Referência* definido como "`ABCDE`") como a fonte de dados para grupos de hosts. Este parâmetro não é suportado se o widget estiver sendo configurado em um dashboard de template.
Excluir grupos de hosts	2	exclude_groupids.0	ID do Grupo de hosts. Nota: Para excluir vários grupos de hosts, crie um objeto de campo de widget de dashboard para cada grupo de hosts com um número incrementado no nome da propriedade. Este parâmetro não é suportado se o widget estiver sendo configurado em um dashboard de template.
Hosts	3	hostids.0	ID do Host. Nota: Para configurar vários hosts, crie um objeto de campo de widget de dashboard para cada host com um número incrementado no nome da propriedade. Para múltiplos hosts, o parâmetro *Grupos de hosts* deve não estar configurado ou estar configurado com pelo menos um grupo de hosts ao qual os hosts configurados pertencem. Este parâmetro não é suportado se o widget estiver sendo configurado em um dashboard de template.
Hosts (Widget/Dashboard)	1	hostids._reference	Em vez do ID do Host: `DASHBOARD.hostids` - defina o seletor *Host* como a fonte de dados para hosts; `ABCDE._hostids` - defina um widget compatível (com seu parâmetro *Referência* definido como "`ABCDE`") como a fonte de dados para hosts. Este parâmetro não é suportado se o widget estiver sendo configurado em um dashboard de template.

Parâmetro	tipo	nome	valor
Tags de cenário			
Tipo de avaliação	0	evaltype	0 - *(padrão)* E/Ou; 2 - Ou.
Nome da tag	1	tags.0.tag	Qualquer valor de string. Nota: O número no nome da propriedade faz referência à ordem da tag na lista de avaliação de tags. Comportamento do parâmetro: - *obrigatório* se estiver configurando *Tags de cenário*
Operador	0	tags.0.operator	0 - Contém; 1 - Igual; 2 - Não contém; 3 - Não é igual; 4 - Existe; 5 - Não existe. Nota: O número no nome da propriedade faz referência à ordem da tag na lista de avaliação de tags. Comportamento do parâmetro: - *obrigatório* se estiver configurando *Tags de cenário*
Valor da tag	1	tags.0.value	Qualquer valor de string. Nota: O número no nome da propriedade faz referência à ordem da tag na lista de avaliação de tags. Comportamento do parâmetro: - *obrigatório* se estiver configurando *Tags de cenário*
Mostrar hosts em manutenção	0	maintenance	0 - Desabilitado; 1 - *(padrão)* Habilitado.
Referência	1	reference	Qualquer valor de string consistindo de 5 caracteres (por exemplo, `ABCDE` ou `JBPNL`). Este valor deve ser exclusivo dentro do dashboard ao qual o widget pertence. Comportamento do parâmetro: - *obrigatório*

Exemplos

Os exemplos a seguir têm como objetivo apenas descrever a configuração dos objetos de campo do widget do painel para o widget *Web monitoring*. Para obter mais informações sobre como configurar um painel, consulte `dashboard.create`.

Configurando um widget de *Monitoramento web*

Configure um widget de *Monitoramento web* que exibe um resumo do status dos cenários de monitoramento web ativos para o grupo de hosts "4".

Requisição:

```
{
    "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
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Veja também

- Campo de widget do dashboard
- `dashboard.create`
- `dashboard.update`

Diretório de usuários

Esta classe foi projetada para trabalhar com diretórios de usuários.

Referências de objetos:

- Diretório de usuários
  - Mapeamentos de tipos de mídia
  - Mapeamentos de grupos de provisionamento

Métodos disponíveis:

- userdirectory.create - criar novo diretório de usuários
- userdirectory.delete - excluir diretório de usuários
- userdirectory.get - recuperar diretório de usuários
- userdirectory.update - atualizar diretório de usuários
- userdirectory.test - testar conexão do diretório de usuários

Objeto de diretório de usuário

Os seguintes objetos estão diretamente relacionados à API `userdirectory`.

Diretório de usuários

O objeto de diretório de usuários possui as seguintes propriedades.

Propriedade	Tipo	Descrição
userdirectoryid	ID	ID do diretório de usuários. Se um diretório de usuários for excluído, o valor da propriedade `userdirectoryid` do objeto Usuário será definido como ”0” para todos os usuários vinculados ao diretório de usuários excluído. Comportamento da propriedade: - *somente leitura* - *obrigatório* para operações de atualização
idp_type	integer	Tipo de protocolo de autenticação usado pelo provedor de identidade para o diretório de usuários. Observe que só pode existir um diretório de usuários do tipo SAML. Valores possíveis: 1 - Diretório de usuários do tipo LDAP; 2 - Diretório de usuários do tipo SAML. Comportamento da propriedade: - *obrigatório* para operações de criação
group_name	string	Atributo do diretório de usuários LDAP/SAML que contém o nome do grupo usado para mapear grupos entre o diretório de usuários LDAP/SAML e o Zabbix. Exemplo: *cn* Comportamento da propriedade: - *obrigatório* se `provision_status` estiver definido como ”Habilitado” e `saml_jit_status` do objeto Autenticação estiver definido como ”Habilitado para IdPs SAML configurados”
user_username	string	Atributo do diretório de usuários LDAP/SAML (também atributo SCIM se `scim_status` estiver definido como ”Provisionamento SCIM habilitado”) que contém o nome do usuário que é usado como valor para a propriedade `name` do objeto Usuário quando o usuário é provisionado. Exemplos: *cn*, *commonName*, *displayName*, *name*
user_lastname	string	Atributo do diretório de usuários LDAP/SAML (também atributo SCIM se `scim_status` estiver definido como ”Provisionamento SCIM habilitado”) que contém o sobrenome do usuário que é usado como valor para a propriedade `surname` do objeto Usuário quando o usuário é provisionado. Exemplos: *sn*, *surname*, *lastName*

Propriedade	Tipo	Descrição
provision_status	integer	Status de provisionamento do diretório de usuários. Valores possíveis: 0 - *(padrão)* Desabilitado (o provisionamento de usuários criados por este diretório de usuários está desabilitado); 1 - Habilitado (o provisionamento de usuários criados por este diretório de usuários está habilitado; adicionalmente, o status do provisionamento LDAP ou SAML (`ldap_jit_status` ou `saml_jit_status` do objeto Autenticação) deve estar habilitado).
provision_groups	array	Array de objetos de mapeamentos de grupos de provisionamento para mapear o padrão de grupo de usuários LDAP/SAML para o grupo de usuários e função de usuário do Zabbix. Comportamento da propriedade: - *obrigatório* se `provision_status` estiver definido como "Habilitado"
provision_media	array	Array de objetos de mapeamentos de tipos de mídia para mapear os atributos de mídia LDAP/SAML do usuário (por exemplo, e-mail) para a mídia de usuário do Zabbix para envio de notificações.
Propriedades específicas do LDAP:		
name	string	Nome exclusivo do diretório de usuários. Comportamento da propriedade: - *obrigatório* se `idp_type` estiver definido como "Diretório de usuários do tipo LDAP"
host	string	Nome do host, IP ou URI do servidor LDAP. A URI deve conter o esquema (`ldap://` ou `ldaps://`), host e porta (opcional). Exemplos: *host.example.com* *127.0.0.1* *ldap://ldap.example.com:389* Comportamento da propriedade: - *obrigatório* se `idp_type` estiver definido como "Diretório de usuários do tipo LDAP"
port	integer	Porta do servidor LDAP. Comportamento da propriedade: - *obrigatório* se `idp_type` estiver definido como "Diretório de usuários do tipo LDAP"
base_dn	string	Caminho base do diretório de usuários LDAP para contas de usuário. Exemplos: *ou=Users,dc=example,dc=org* *ou=Users,ou=system* (para OpenLDAP) *DC=company,DC=com* (para Microsoft Active Directory) *uid=%{user},dc=example,dc=com* (para vinculação direta de usuário; o placeholder "*%{user}*" é obrigatório) Comportamento da propriedade: - *obrigatório* se `idp_type` estiver definido como "Diretório de usuários do tipo LDAP"

Propriedade	Tipo	Descrição
search_attribute	string	Atributo do diretório de usuários LDAP pelo qual identificar a conta de usuário a partir das informações fornecidas na solicitação de login. Exemplos: *uid* (para OpenLDAP) *sAMAccountName* (para Microsoft Active Directory) Comportamento da propriedade: - *obrigatório* se `idp_type` estiver definido como ”Diretório de usuários do tipo LDAP”
bind_dn	string	Conta do servidor LDAP para vinculação e pesquisa no servidor LDAP. Para vinculação direta de usuário e vinculação anônima, `bind_dn` deve estar vazio. Exemplos: *uid=ldap_search,ou=system* (para OpenLDAP) *CN=ldap_search,OU=user_group,DC=company,DC=com* (para Microsoft Active Directory) *CN=Admin,OU=Users,OU=Zabbix,DC=zbx,DC=local* Comportamento da propriedade: - *suportado* se `idp_type` estiver definido como ”Diretório de usuários do tipo LDAP”
bind_password	string	Senha LDAP da conta para vinculação e pesquisa no servidor LDAP. Para vinculação direta de usuário e vinculação anônima, `bind_password` deve estar vazio. Comportamento da propriedade: - *suportado* se `idp_type` estiver definido como ”Diretório de usuários do tipo LDAP”
description	string	Descrição do diretório de usuários. Comportamento da propriedade: - *suportado* se `idp_type` estiver definido como ”Diretório de usuários do tipo LDAP”
group_basedn	string	Caminho base do diretório de usuários LDAP para grupos; usado para configurar uma verificação de associação de usuário no diretório de usuários LDAP. Ignorado ao provisionar um usuário se `group_membership` estiver definido. Exemplo: *ou=Groups,dc=example,dc=com* Comportamento da propriedade: - *suportado* se `idp_type` estiver definido como ”Diretório de usuários do tipo LDAP”

Propriedade	Tipo	Descrição
group_filter	string	String de filtro para recuperar grupos do diretório de usuários LDAP dos quais o usuário é membro; usado para configurar uma verificação de associação de usuário no diretório de usuários LDAP. Ignorado ao provisionar um usuário se `group_membership` estiver definido. Placeholders suportados em `group_filter`: *%{attr}* - atributo de pesquisa (substituído pelo valor da propriedade `search_attribute`); *%{groupattr}* - atributo de grupo (substituído pelo valor da propriedade `group_member`); *%{host}* - nome do host, IP ou URI do servidor LDAP (substituído pelo valor da propriedade `host`); *%{user}* - nome de usuário do Zabbix. Padrão: *(%{groupattr}=%{user})* Exemplos: - *(member=uid=%{ref},ou=Users,dc=example,dc=com)* corresponderá a "User1" se um objeto de grupo LDAP contiver o atributo "*member*" com o valor "*uid=User1,ou=Users,dc=example,dc=com*", e retornará o grupo do qual "User1" é membro; - *(%{groupattr}=cn=%{ref},ou=Users,ou=Zabbix,DC=example,DC=com)* corresponderá a "User1" se um objeto de grupo LDAP contiver o atributo especificado na propriedade `group_member` com o valor "*cn=User1,ou=Users,ou=Zabbix,DC=example,DC=com*", e retornará o grupo do qual "User1" é membro. Comportamento da propriedade: - *suportado* se `idp_type` estiver definido como "Diretório de usuários do tipo LDAP"
group_member	string	Atributo do diretório de usuários LDAP que contém informações sobre os membros do grupo; usado para configurar uma verificação de associação de usuário no diretório de usuários LDAP. Ignorado ao provisionar um usuário se `group_membership` estiver definido. Comportamento da propriedade: - *suportado* se `idp_type` estiver definido como "Diretório de usuários do tipo LDAP"
group_membership	string	Atributo do diretório de usuários LDAP que contém informações sobre os grupos aos quais um usuário pertence. Exemplo: *memberOf* Comportamento da propriedade: - *suportado* se `idp_type` estiver definido como "Diretório de usuários do tipo LDAP"

Propriedade	Tipo	Descrição
search_filter	string	String de filtro personalizada usada para localizar e autenticar um usuário em um diretório de usuários LDAP com base nas informações fornecidas na solicitação de login. Placeholders suportados em `search_filter`: *%{attr}* - nome do atributo de pesquisa (por exemplo, *uid*, *sAMAccountName*); *%{user}* - nome de usuário do Zabbix. Padrão: *(%{attr}=%{user})* Comportamento da propriedade: - *suportado* se `idp_type` estiver definido como ”Diretório de usuários do tipo LDAP”
start_tls	integer	Opção de configuração do servidor LDAP que permite que a comunicação com o servidor LDAP seja protegida usando Transport Layer Security (TLS). Observe que `start_tls` deve ser definido como ”Desabilitado” para hosts usando o protocolo `ldaps://`. Valores possíveis: 0 - *(padrão)* Desabilitado; 1 - Habilitado. Comportamento da propriedade: - *suportado* se `idp_type` estiver definido como ”Diretório de usuários do tipo LDAP”
user_ref_attr	string	Atributo do diretório de usuários LDAP usado para referenciar um objeto de usuário. O valor de `user_ref_attr` é usado para obter valores do atributo especificado no diretório de usuários e colocá-los no lugar do placeholder *%{ref}* na string `group_filter`. Exemplos: *cn*, *uid*, *member*, *uniqueMember* Comportamento da propriedade: - *suportado* se `idp_type` estiver definido como ”Diretório de usuários do tipo LDAP”
Propriedades específicas do SAML:		
idp_entityid	string	URI que identifica o provedor de identidade e é usada para se comunicar com o provedor de identidade em mensagens SAML. Exemplo: *https://idp.example.com/idp* Comportamento da propriedade: - *obrigatório* se `idp_type` estiver definido como ”Diretório de usuários do tipo SAML”
sp_entityid	string	URL ou qualquer string que identifica o provedor de serviços do provedor de identidade. Exemplos: *https://idp.example.com/sp* *zabbix* Comportamento da propriedade: - *obrigatório* se `idp_type` estiver definido como ”Diretório de usuários do tipo SAML”

Propriedade	Tipo	Descrição
username_attribute	string	Atributo do diretório de usuários SAML (também atributo SCIM se `scim_status` estiver definido como "Provisionamento SCIM habilitado") que contém o nome de usuário do usuário que é comparado com o valor da propriedade `username` do objeto Usuário ao autenticar. Exemplos: *uid*, *userprincipalname*, *samaccountname*, *username*, *userusername*, *urn:oid:0.9.2342.19200300.100.1.1*, *urn:oid:1.3.6.1.4.1.5923.1.1.1.13*, *urn:oid:0.9.2342.19200300.100.1.44* Comportamento da propriedade: - *obrigatório* se `idp_type` estiver definido como "Diretório de usuários do tipo SAML"
sso_url	string	URL do serviço de logon único SAML do provedor de identidade, para o qual o Zabbix enviará as solicitações de autenticação SAML. Exemplo: *http://idp.example.com/idp/sso/saml* Comportamento da propriedade: - *obrigatório* se `idp_type` estiver definido como "Diretório de usuários do tipo SAML"
slo_url	string	URL do serviço de logout único SAML do provedor de identidade, para o qual o Zabbix enviará as solicitações de logout SAML. Exemplo: *https://idp.example.com/idp/slo/saml* Comportamento da propriedade: - *suportado* se `idp_type` estiver definido como "Diretório de usuários do tipo SAML"
encrypt_nameid	integer	Se o ID de nome SAML deve ser criptografado. Valores possíveis: 0 - *(padrão)* Não criptografar o ID de nome; 1 - Criptografar o ID de nome. Comportamento da propriedade: - *suportado* se `idp_type` estiver definido como "Diretório de usuários do tipo SAML"
encrypt_assertions	integer	Se as asserções SAML devem ser criptografadas. Valores possíveis: 0 - *(padrão)* Não criptografar asserções; 1 - Criptografar asserções. Comportamento da propriedade: - *suportado* se `idp_type` estiver definido como "Diretório de usuários do tipo SAML"
nameid_format	string	Formato do ID de nome do provedor de serviços do provedor de identidade SAML. Exemplos: *urn:oasis:names:tc:SAML:2.0:nameid-format:persistent* *urn:oasis:names:tc:SAML:2.0:nameid-format:transient* *urn:oasis:names:tc:SAML:2.0:nameid-format:kerberos* *urn:oasis:names:tc:SAML:2.0:nameid-format:entity* Comportamento da propriedade: - *suportado* se `idp_type` estiver definido como "Diretório de usuários do tipo SAML"

Propriedade	Tipo	Descrição
scim_status	integer	Se o provisionamento SCIM para SAML está habilitado ou desabilitado. Valores possíveis: 0 - *(padrão)* Provisionamento SCIM desabilitado; 1 - Provisionamento SCIM habilitado. Comportamento da propriedade: - *suportado* se `idp_type` estiver definido como ”Diretório de usuários do tipo SAML”
sign_assertions	integer	Se as asserções SAML devem ser assinadas com uma assinatura SAML. Valores possíveis: 0 - *(padrão)* Não assinar asserções; 1 - Assinar asserções. Comportamento da propriedade: - *suportado* se `idp_type` estiver definido como ”Diretório de usuários do tipo SAML”
sign_authn_requests	integer	Se as solicitações SAML AuthN devem ser assinadas com uma assinatura SAML. Valores possíveis: 0 - *(padrão)* Não assinar solicitações AuthN; 1 - Assinar solicitações AuthN. Comportamento da propriedade: - *suportado* se `idp_type` estiver definido como ”Diretório de usuários do tipo SAML”
sign_messages	integer	Se as mensagens SAML devem ser assinadas com uma assinatura SAML. Valores possíveis: 0 - *(padrão)* Não assinar mensagens; 1 - Assinar mensagens. Comportamento da propriedade: - *suportado* se `idp_type` estiver definido como ”Diretório de usuários do tipo SAML”
sign_logout_requests	integer	Se as solicitações de logout SAML devem ser assinadas com uma assinatura SAML. Valores possíveis: 0 - *(padrão)* Não assinar solicitações de logout; 1 - Assinar solicitações de logout. Comportamento da propriedade: - *suportado* se `idp_type` estiver definido como ”Diretório de usuários do tipo SAML”
sign_logout_responses	integer	Se as respostas de logout SAML devem ser assinadas com uma assinatura SAML. Valores possíveis: 0 - *(padrão)* Não assinar respostas de logout; 1 - Assinar respostas de logout. Comportamento da propriedade: - *suportado* se `idp_type` estiver definido como ”Diretório de usuários do tipo SAML”

Mapeamentos de tipo de mídia

O objeto de mapeamentos de tipo de mídia possui as seguintes propriedades.

Propriedade	Tipo	Descrição
userdirectory_mediaid	ID	ID do mapeamento de tipo de mídia. Comportamento da propriedade: - *somente leitura*
name	string	Nome visível na lista de mapeamentos de tipo de mídia. Comportamento da propriedade: - *obrigatório*
mediatypeid	ID	ID do tipo de mídia a ser criado; usado como valor para a propriedade `mediatypeid` do objeto Media. Comportamento da propriedade: - *obrigatório*
attribute	string	Atributo do diretório de usuários LDAP/SAML (também atributo SCIM se `scim_status` estiver definido como "Provisionamento SCIM está ativado") que contém a mídia do usuário (por exemplo, *user@example.com*) que é usada como valor para a propriedade `sendto` do objeto Media. Se presente nos dados recebidos do provedor de identidade LDAP/SAML, e o valor não estiver vazio, isso acionará a criação da mídia para o usuário provisionado. Comportamento da propriedade: - *obrigatório*
active	integer	Valor da propriedade `active` da mídia do usuário quando a mídia é criada para o usuário provisionado. Valores possíveis: 0 - *(padrão)* ativado; 1 - desativado.
severity	integer	Valor da propriedade `severity` da mídia do usuário quando a mídia é criada para o usuário provisionado. Padrão: 63.
period	string	Valor da propriedade `period` da mídia do usuário quando a mídia é criada para o usuário provisionado. Padrão: 1-7,00:00-24:00.

Mapeamentos de grupos de provisionamento

Os mapeamentos de grupos de provisionamento possuem as seguintes propriedades.

Propriedade	Tipo	Descrição
name	string	Nome completo de um grupo (por exemplo, *Zabbix administrators*) no diretório de usuários LDAP/SAML (também SCIM se `scim_status` estiver definido como "Provisionamento SCIM está ativado"). Suporta o caractere curinga "*". Único em todos os mapeamentos de grupos de provisionamento. Comportamento da propriedade: - *obrigatório*

Propriedade	Tipo	Descrição
roleid	ID	ID da função de usuário a ser atribuída ao usuário. Se vários mapeamentos de grupos de provisionamento forem correspondidos, a função do tipo de usuário mais alto (*User*, *Admin* ou *Super admin*) será atribuída ao usuário. Se houver várias funções com o mesmo tipo de usuário, a primeira função (ordenada em ordem alfabética) será atribuída ao usuário. Comportamento da propriedade: - *obrigatório*
user_groups	array	Array de objetos de ID de grupo de usuários do Zabbix. Cada objeto possui as seguintes propriedades: `usrgrpid` - `(ID)` ID do grupo de usuários do Zabbix a ser atribuído ao usuário. Se vários mapeamentos de grupos de provisionamento forem correspondidos, os grupos de usuários do Zabbix de todos os mapeamentos correspondentes serão atribuídos ao usuário. Comportamento da propriedade: - *obrigatório*

userdirectory.create

Descrição

`object userdirectory.create(object/array userDirectory)`

Este método permite criar novos diretórios de usuários.

> **Note:**
> Este método está disponível apenas para o tipo de usuário *Super admin*.

Parâmetros

`(object/array)` Diretórios de usuário a serem criados.

O método aceita diretórios de usuário com as propriedades padrão do diretório de usuário.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos diretórios de usuários criados na propriedade `userdirectoryids`. A ordem dos IDs retornados corresponde à ordem dos diretórios de usuários passados.

Exemplos

Criando um diretório de usuários

Crie um diretório de usuários para autenticar usuários com StartTLS via LDAP. Observe que, para autenticar usuários via LDAP, a autenticação LDAP deve estar habilitada.

Requisição:

```
{
    "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
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "userdirectoryids": [
            "3"
        ]
    },
    "id": 1
}
```

Criando um diretório de usuários (provisionamento JIT ativado)

Crie um diretório de usuários para autenticar usuários via LDAP (com provisionamento JIT ativado). Observe que, para autenticar usuários via LDAP, a autenticação LDAP deve estar ativada.

Requisição:

```
{
    "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
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "userdirectoryids": [
            "2"
        ]
    },
    "id": 1
}
```

Fonte

CUserDirectory::create() in *ui/include/classes/api/services/CUserDirectory.php*.

userdirectory.delete

Descrição

`object userdirectory.delete(array userDirectoryIds)`

Este método permite excluir diretórios de usuários. O diretório de usuários não pode ser excluído quando é usado diretamente por pelo menos um grupo de usuários.
 O diretório de usuários padrão do LDAP não pode ser excluído quando `authentication.ldap_configured` está definido como 1 ou quando restam mais diretórios de usuários.

> **Note:**
> Este método está disponível apenas para o tipo de usuário *Super admin*.

Parâmetros

`(array)` IDs dos diretórios de usuários a serem excluídos.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos diretórios de usuários excluídos na propriedade `userdirectoryids`.

Exemplos

Excluindo vários diretórios de usuários

Exclua dois diretórios de usuários.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "userdirectory.delete",
    "params": [
        "2",
        "12"
    ],
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "userdirectoryids": [
            "2",
            "12"
        ]
    },
    "id": 1
}
```

Fonte

CUserDirectory::delete() in *ui/include/classes/api/services/CUserDirectory.php*.

userdirectory.get

Descrição

`integer/array userdirectory.get(object parameters)`

O método permite recuperar diretórios de usuários de acordo com os parâmetros fornecidos.

> **Note:**
> Este método está disponível apenas para tipos de usuários *Super admin*.

Parâmetros

`(object)` Parâmetros que definem a saída desejada.

O método suporta os seguintes parâmetros.

Parâmetro	Tipo	Descrição
userdirectoryids	ID/array	Retorna apenas diretórios de usuários com os IDs fornecidos.
selectUsrgrps	query	Retorna uma propriedade `usrgrps` com grupos de usuários associados a um diretório de usuário. Suporta `count`.
selectProvisionMedia	query	Retorna uma propriedade `provision_media` com mapeamentos de tipos de mídia associados a um diretório de usuário.
selectProvisionGroups	query	Retorna uma propriedade `provision_groups` com mapeamentos de grupos de provisionamento associados a um diretório de usuário.
sortfield	string/array	Ordena o resultado pelas propriedades fornecidas. Valores possíveis: `name`.
filter	object	Retorna apenas os resultados que correspondem exatamente ao filtro fornecido. Aceita um objeto, onde as chaves são nomes de propriedades e os valores são um único valor ou um array de valores. Suporta propriedades: `userdirectoryid`, `idp_type`, `provision_status`.

Parâmetro	Tipo	Descrição
search	object	Retorna resultados que correspondem ao padrão fornecido (case-insensitive). Aceita um objeto, onde as chaves são nomes de propriedades e os valores são strings a serem pesquisadas. Se nenhuma opção adicional for fornecida, será realizada uma pesquisa `LIKE "%...%"`. Propriedades suportadas: `name`, `description`. Diretórios de usuário do tipo SAML terão valor vazio para as propriedades `name` e `description`. Ambas as propriedades podem ser alteradas com a operação `userdirectory.update`.
countOutput	boolean	Esses parâmetros são descritos no comentário de referência.
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valores de retorno

`(integer/array)` Retorna:

- um array de objetos;
- a contagem de objetos recuperados, se o parâmetro `countOutput` foi utilizado.

Exemplos

Recuperando diretórios de usuários

Recupere todos os diretórios de usuários com propriedades adicionais que exibem mapeamentos de tipos de mídia e mapeamentos de grupos de provisionamento associados a cada diretório de usuário.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "userdirectory.get",
    "params": {
        "output": "extend",
        "selectProvisionMedia": "extend",
        "selectProvisionGroups": "extend"
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": [
       {
            "userdirectoryid": "1",
            "idp_type": "2",
            "name": "",
            "provision_status": "1",
            "description": "",
            "group_name": "groups",
            "user_username": "",
            "user_lastname": "",
            "idp_entityid": "http://example.com/simplesaml/saml2/idp/metadata.php",
            "sso_url": "http://example.com/simplesaml/saml2/idp/SSOService.php",
```

```
            "slo_url": "",
            "username_attribute": "uid",
            "sp_entityid": "zabbix",
            "nameid_format": "",
            "sign_messages": "0",
            "sign_assertions": "0",
            "sign_authn_requests": "0",
            "sign_logout_requests": "0",
            "sign_logout_responses": "0",
            "encrypt_nameid": "0",
            "encrypt_assertions": "0",
            "scim_status": "1",
            "provision_media": [
                {
                    "userdirectory_mediaid": "1",
                    "name": "example.com",
                    "mediatypeid": "1",
                    "attribute": "user@example.com",
                    "active": "0",
                    "severity": "63",
                    "period": "1-7,00:00-24:00"
                }
            ],
            "provision_groups": [
                {
                    "name": "*",
                    "roleid": "1",
                    "user_groups": [
                        {
                            "usrgrpid": "13"
                        }
                    ]
                }
            ]
        },
        {
            "userdirectoryid": "2",
            "idp_type": "1",
            "name": "AD server",
            "provision_status": "1",
            "description": "",
            "host": "host.example.com",
            "port": "389",
            "base_dn": "DC=zbx,DC=local",
            "search_attribute": "sAMAccountName",
            "bind_dn": "CN=Admin,OU=Users,OU=Zabbix,DC=zbx,DC=local",
            "start_tls": "0",
            "search_filter": "",
            "group_basedn": "OU=Zabbix,DC=zbx,DC=local",
            "group_name": "CN",
            "group_member": "member",
            "group_filter": "(%{groupattr}=CN=%{ref},OU=Users,OU=Zabbix,DC=zbx,DC=local)",
            "group_membership": "",
            "user_username": "givenName",
            "user_lastname": "sn",
            "user_ref_attr": "CN",
            "provision_media": [
                {
                    "userdirectory_mediaid": "2",
                    "name": "example.com",
                    "mediatypeid": "1",
                    "attribute": "user@example.com",
```

```
                    "active": "0",
                    "severity": "63",
                    "period": "1-7,00:00-24:00"
                }
            ],
            "provision_groups": [
                {
                    "name": "*",
                    "roleid": "4",
                    "user_groups": [
                        {
                            "usrgrpid": "8"
                        }
                    ]
                },
                {
                    "name": "Zabbix administrators",
                    "roleid": "2",
                    "user_groups": [
                        {
                            "usrgrpid": "7"
                        },
                        {
                            "usrgrpid": "8"
                        }
                    ]
                }
            ]
        },
        {
            "userdirectoryid": "3",
            "idp_type": "1",
            "name": "LDAP API server #1",
            "provision_status": "0",
            "description": "",
            "host": "ldap://local.ldap",
            "port": "389",
            "base_dn": "ou=Users,dc=example,dc=org",
            "search_attribute": "uid",
            "bind_dn": "cn=ldap_search,dc=example,dc=org",
            "start_tls": "1",
            "search_filter": "",
            "group_basedn": "",
            "group_name": "",
            "group_member": "",
            "group_filter": "",
            "group_membership": "",
            "user_username": "",
            "user_lastname": "",
            "user_ref_attr": "",
            "provision_media": [],
            "provision_groups": []
        }
    ],
    "id": 1
}
```


Veja também

- Grupo de usuários

Fonte

CUserDirectory::get() em *ui/include/classes/api/services/CUserDirectory.php*.

userdirectory.test

Descrição

`object userdirectory.test(array userDirectory)`

Este método permite testar as configurações de conexão do diretório de usuários.

> **Note:**
> Este método também permite testar quais dados configurados correspondem às configurações do diretório de usuários para provisionamento de usuários (por exemplo, qual função de usuário, grupos de usuários, mídias de usuários serão atribuídos ao usuário). Para este tipo de teste, a requisição da API deve ser feita para um diretório de usuários que tenha `provision_status` definido como habilitado.

> **Note:**
> Este método está disponível apenas para o tipo de usuário *Super admin*.

Parâmetros

`(object)` Propriedades do diretório de usuários.

Como a API `userdirectory.get` não retorna o campo `bind_password`, `userdirectoryid` e/ou `bind_password` devem ser fornecidos.
 Além das propriedades padrão do diretório de usuários, o método aceita os seguintes parâmetros.

Parâmetro	Tipo	Descrição
test_username	string	Nome de usuário para testar no diretório de usuários.
test_password	string	Senha associada ao nome de usuário para testar no diretório de usuários.

Valores de retorno

`(bool)` Retorna verdadeiro em caso de sucesso.

Exemplos

Testar diretório de usuários para usuário existente

Testar o diretório de usuários ”3” para ”user1”.

Requisição:

```
{
    "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
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": true,
    "id": 1
}
```

Testar diretório de usuário para usuário inexistente

Testar o diretório de usuário ”3” para o ”user2” inexistente.

Requisição:

```
{
    "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
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "error": {
        "code": -32500,
        "message": "Application error.",
        "data": "Incorrect user name or password or account is temporarily blocked."
    },
    "id": 1
}
```

Testar diretório de usuário para provisionamento de usuário

Testar o diretório de usuário ”3” para quais dados configurados correspondem às configurações do diretório de usuário para o provisionamento de ”user3” (por exemplo, quais funções de usuário, grupos de usuários, mídias de usuário serão atribuídos ao usuário).

Requisição:

```
{
    "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
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "username": "user3",
        "name": "John",
        "surname": "Doe",
        "medias": [],
        "usrgrps": [
```

```
            {
                "usrgrpid": "8"
            },
            {
                "usrgrpid": "7"
            }
        ],
        "roleid": "2",
        "userdirectoryid": "2"
    },
    "id": 1
}
```

Fonte

CUserDirectory::test() em *ui/include/classes/api/services/CUserDirectory.php*.

userdirectory.update

Descrição

`object userdirectory.update(object/array userDirectory)`

Este método permite atualizar diretórios de usuários existentes.

> **Note:**
> Este método está disponível apenas para o tipo de usuário *Super admin*.

Parâmetros

`(object/array)` Propriedades do diretório de usuário a serem atualizadas.

A propriedade `userdirectoryid` deve ser definida para cada diretório de usuário, todas as outras propriedades são opcionais. Apenas as propriedades passadas serão atualizadas, todas as outras permanecerão inalteradas.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos diretórios de usuários atualizados na propriedade `userdirectoryids`.

Exemplos

Atualizar a senha de bind para o diretório de usuários

Defina uma nova senha de bind para um diretório de usuários.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "userdirectory.update",
    "params": {
        "userdirectoryid": "3",
        "bind_password": "newldappassword"
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "userdirectoryids": [
            "3"
        ]
    },
    "id": 1
}
```

Atualizar mapeamentos para diretório de usuários

Atualize os mapeamentos de grupos de provisionamento e mapeamentos de tipos de mídia para o diretório de usuários "2".

Requisição:

```
{
    "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
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "userdirectoryids": [
            "2"
        ]
    },
    "id": 1
}
```

Fonte

CUserDirectory::update() em *ui/include/classes/api/services/CUserDirectory.php*.

Event

Esta classe foi projetada para trabalhar com eventos.

Referências de objetos:

- Event
- Event tag
- Media type URL

Métodos disponíveis:

- event.get - recuperar eventos
- event.acknowledge - reconhecer eventos

Objeto event

Os seguintes objetos estão diretamente relacionados à API `event`.

Evento

> **Note:**
> Os eventos são criados pelo Zabbix server e não podem ser modificados via API.

O objeto evento possui as seguintes propriedades.

Propriedade	Tipo	Descrição
eventid	ID	ID do evento.
source	integer	Tipo do evento. Valores possíveis: 0 - evento criado por um trigger; 1 - evento criado por uma regra de descoberta; 2 - evento criado por autoregistro de agent ativo; 3 - evento interno; 4 - evento criado na atualização do status do serviço.
object	integer	Tipo de objeto relacionado ao evento. Valores possíveis se `source` estiver definido como "evento criado por um trigger": 0 - trigger. Valores possíveis se `source` estiver definido como "evento criado por uma regra de descoberta": 1 - host descoberto; 2 - serviço descoberto. Valores possíveis se `source` estiver definido como "evento criado por autoregistro de agent ativo": 3 - host autoregistrado. Valores possíveis se `source` estiver definido como "evento interno": 0 - trigger; 4 - item; 5 - regra LLD. Valores possíveis se `source` estiver definido como "evento criado na atualização do status do serviço": 6 - serviço.
objectid	ID	ID do objeto relacionado.
acknowledged	integer	Se o evento foi reconhecido.
clock	timestamp	Hora em que o evento foi criado.
ns	integer	Nanosegundos em que o evento foi criado.
name	string	Nome do evento resolvido.

Propriedade	Tipo	Descrição
value	integer	Estado do objeto relacionado. Valores possíveis se `source` estiver definido como ”evento criado por um trigger” ou ”evento criado na atualização do status do serviço”: 0 - OK; 1 - problema. Valores possíveis se `source` estiver definido como ”evento criado por uma regra de descoberta”: 0 - host ou serviço ativo; 1 - host ou serviço inativo; 2 - host ou serviço descoberto; 3 - host ou serviço perdido. Valores possíveis se `source` estiver definido como ”evento interno”: 0 - estado ”normal”; 1 - estado ”desconhecido” ou ”não suportado”. Comportamento da propriedade: - *suportado* se `source` estiver definido como ”evento criado por um trigger”, ”evento criado por uma regra de descoberta”, ”evento interno” ou ”evento criado na atualização do status do serviço”
severity	integer	Gravidade atual do evento. Valores possíveis: 0 - não classificado; 1 - informação; 2 - aviso; 3 - média; 4 - alta; 5 - desastre.
r_eventid	ID	ID do evento de recuperação.
c_eventid	ID	ID do evento que foi usado para sobrescrever (fechar) o evento atual sob a regra de correlação global. Veja `correlationid` para identificar a regra de correlação exata. Este parâmetro só é definido quando o evento é fechado por regra de correlação global.
cause_eventid	ID	ID do evento de causa.
correlationid	ID	ID da regra de correlação que gerou o fechamento do problema. Este parâmetro só é definido quando o evento é fechado por regra de correlação global.
userid	ID	ID do usuário que fechou o evento (se o evento foi fechado manualmente).
suppressed	integer	Se o evento está suprimido. Valores possíveis: 0 - evento está em estado normal; 1 - evento está suprimido.
opdata	string	Dados operacionais com macros expandidas.
urls	array	URLs de tipo de mídia ativas.

Tag de evento

O objeto de tag de evento possui as seguintes propriedades.

Propriedade	Tipo	Descrição
tag	string	Nome da tag de evento.
value	string	Valor da tag de evento.

URL do tipo de mídia

O objeto de URL do tipo de mídia possui as seguintes propriedades.

Propriedade	Tipo	Descrição
name	string	Nome da URL definida pelo tipo de mídia.
url	string	Valor da URL definida pelo tipo de mídia.

Os resultados conterão entradas apenas para tipos de mídia ativos com entrada de menu de evento habilitada. Macros usadas nas propriedades serão expandidas, mas se uma das propriedades contiver uma macro não expandida, ambas as propriedades serão excluídas dos resultados. Para macros suportadas, consulte *Macros suportadas*.

event.acknowledge

Descrição

`object event.acknowledge(object/array parameters)`

Este método permite atualizar eventos. As seguintes ações de atualização podem ser realizadas:

- Fechar evento. Se o evento já estiver resolvido, esta ação será ignorada.
- Reconhecer evento. Se o evento já estiver reconhecido, esta ação será ignorada.
- Desfazer reconhecimento do evento. Se o evento não estiver reconhecido, esta ação será ignorada.
- Adicionar mensagem.
- Alterar a severidade do evento. Se o evento já tiver a mesma severidade, esta ação será ignorada.
- Suprimir evento. Se o evento já estiver suprimido, esta ação será ignorada.
- Remover supressão do evento. Se o evento não estiver suprimido, esta ação será ignorada.
- Alterar a classificação do evento para causa. Se o evento já estiver classificado como causa, esta ação será ignorada.
- Alterar a classificação do evento para sintoma. Se o evento já estiver classificado como sintoma, esta ação será ignorada.

Attention:
Apenas eventos de trigger podem ser atualizados.
 Apenas eventos de problema podem ser atualizados.
 Permissões de leitura/gravação para trigger são necessárias para fechar o evento ou alterar a severidade do evento.
 Para fechar um evento, o fechamento manual deve ser permitido na trigger.

Note:
Este método está disponível para usuários de qualquer tipo. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object/array)` Parâmetros contendo os IDs dos eventos e operações de atualização que devem ser executadas.

Parâmetro	Tipo	Descrição
eventids	ID/array	IDs dos eventos a serem reconhecidos. Comportamento do parâmetro: - *obrigatório*

Parâmetro	Tipo	Descrição
action	integer	Ação(ões) de atualização do evento. Valores possíveis do bitmap: 1 - fechar problema; 2 - reconhecer evento; 4 - adicionar mensagem; 8 - alterar severidade; 16 - desmarcar reconhecimento do evento; 32 - suprimir evento; 64 - dessuprimir evento; 128 - alterar classificação do evento para causa; 256 - alterar classificação do evento para sintoma. Este é um campo de máscara de bits; qualquer soma dos valores possíveis do bitmap é aceitável (por exemplo, 34 para reconhecer e suprimir evento). Comportamento do parâmetro: - *obrigatório*
cause_eventid	ID	ID do evento causa. Comportamento do parâmetro: - *obrigatório* se `action` contiver o bit ”alterar classificação do evento para sintoma”
message	string	Texto da mensagem. Comportamento do parâmetro: - *obrigatório* se `action` contiver o bit ”adicionar mensagem”
severity	integer	Nova severidade para os eventos. Valores possíveis: 0 - não classificado; 1 - informação; 2 - aviso; 3 - médio; 4 - alto; 5 - desastre. Comportamento do parâmetro: - *obrigatório* se `action` contiver o bit ”alterar severidade”
suppress_until	integer	Timestamp Unix até o qual o evento deve ser suprimido. Se definido como ”0”, a supressão será indefinida. Comportamento do parâmetro: - *obrigatório* se `action` contiver o bit ”suprimir evento”

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos eventos atualizados sob a propriedade `eventids`.

Exemplos

Reconhecendo um evento

Reconheça um único evento e deixe uma mensagem.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "event.acknowledge",
    "params": {
        "eventids": "20427",
```

```
        "action": 6,
        "message": "Problem resolved."
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "eventids": [
            "20427"
        ]
    },
    "id": 1
}
```

Alterando a gravidade do evento

Altere a gravidade de vários eventos e deixe uma mensagem.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "event.acknowledge",
    "params": {
        "eventids": ["20427", "20428"],
        "action": 12,
        "message": "Maintenance required to fix it.",
        "severity": 4
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "eventids": [
            "20427",
            "20428"
        ]
    },
    "id": 1
}
```

Fonte

CEvent::acknowledge() em *ui/include/classes/api/services/CEvent.php*.

event.get

Descrição

`integer/array event.get(object parameters)`

O método permite recuperar eventos de acordo com os parâmetros fornecidos.

> **Attention:**
> Este método pode retornar eventos de uma entidade excluída se esses eventos ainda não tiverem sido removidos pelo housekeeper.

Note:
Este método está disponível para usuários de qualquer tipo. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object)` Parâmetros que definem a saída desejada.

O método suporta os seguintes parâmetros.

Parâmetro	Tipo	Descrição
eventids	ID/array	Retorna apenas eventos com os IDs fornecidos.
groupids	ID/array	Retorna apenas eventos criados por objetos que pertencem aos grupos de hosts fornecidos.
hostids	ID/array	Retorna apenas eventos criados por objetos que pertencem aos hosts fornecidos.
objectids	ID/array	Retorna apenas eventos criados pelos objetos fornecidos.
source	integer	Retorna apenas eventos com o tipo fornecido. Consulte a página do objeto event para uma lista dos tipos de eventos suportados. Padrão: 0 - eventos de trigger.
object	integer	Retorna apenas eventos criados por objetos do tipo fornecido. Consulte a página do objeto event para uma lista dos tipos de objetos suportados. Padrão: 0 - trigger.
acknowledged	boolean	Se definido como `true`, retorna apenas eventos reconhecidos.
action	integer	Retorna apenas eventos para os quais as ações de atualização de evento fornecidas foram realizadas. Para várias ações, use uma soma de quaisquer valores de bitmap aceitáveis como máscara de bits (por exemplo, 34 para reconhecer e suprimir evento).
action_userids	ID/array	Retorna apenas eventos com os IDs dos usuários que realizaram as ações de atualização de evento.
suppressed	boolean	Se definido como `true`, retorna apenas eventos suprimidos.
symptom	boolean	Se definido como `true`, retorna apenas eventos de sintoma.
severities	integer/array	Retorna apenas eventos com as gravidades fornecidas. Aplica-se apenas se o objeto for trigger.
trigger_severities	integer/array	Retorna apenas eventos com as gravidades de trigger fornecidas. Aplica-se apenas se o objeto for trigger.
evaltype	integer	Método de avaliação de tag. Valores possíveis: 0 - *(padrão)* E/Ou; 2 - Ou.
tags	array	Retorna apenas eventos com as tags fornecidas. Formato: `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`. Um array vazio retorna todos os eventos. Valores possíveis para operator: 0 - *(padrão)* Contém; 1 - Igual; 2 - Não contém; 3 - Não é igual; 4 - Existe; 5 - Não existe.
eventid_from	string	Retorna apenas eventos com IDs maiores ou iguais ao ID fornecido.
eventid_till	string	Retorna apenas eventos com IDs menores ou iguais ao ID fornecido.
time_from	timestamp	Retorna apenas eventos criados no ou após o horário fornecido.
time_till	timestamp	Retorna apenas eventos criados no ou antes do horário fornecido.

Parâmetro	Tipo	Descrição
problem_time_from	timestamp	Retorna apenas eventos que estavam no estado de problema a partir de `problem_time_from`, independentemente de seu status atual. Aplica-se apenas quando a origem é um evento de trigger e o objeto é um trigger. Este parâmetro é obrigatório se `problem_time_till` for especificado.
problem_time_till	timestamp	Retorna apenas eventos que estavam no estado de problema até `problem_time_till`, independentemente de seu status atual. Aplica-se apenas quando a origem é um evento de trigger e o objeto é um trigger. Este parâmetro é obrigatório se `problem_time_from` for especificado.
value	integer/array	Retorna apenas eventos com os valores fornecidos.
selectAcknowledges	query	Retorna uma propriedade `acknowledges` com as atualizações do evento. As atualizações de evento são classificadas em ordem cronológica inversa. O objeto de atualização de evento possui as seguintes propriedades: `acknowledgeid` - `(ID)` ID do reconhecimento; `userid` - `(ID)` ID do usuário que atualizou o evento; `clock` - `(timestamp)` horário em que o evento foi atualizado; `message` - `(string)` texto da mensagem; `action` - `(integer)` ação de atualização que foi realizada, consulte `event.acknowledge`; `old_severity` - `(integer)` gravidade do evento antes desta ação de atualização; `new_severity` - `(integer)` gravidade do evento após esta ação de atualização; `suppress_until` - `(timestamp)` horário até o qual o evento será suprimido; `taskid` - `(ID)` ID da tarefa se o evento atual estiver passando por uma alteração de classificação; `username` - `(string)` nome de usuário do usuário que atualizou o evento; `name` - `(string)` nome do usuário que atualizou o evento; `surname` - `(string)` sobrenome do usuário que atualizou o evento. Suporta `count`.
selectAlerts	query	Retorna uma propriedade `alerts` com os alertas gerados pelo evento. Os alertas são classificados em ordem cronológica inversa.
selectHosts	query	Retorna uma propriedade `hosts` com os hosts que contêm o objeto que criou o evento. Suportado apenas para eventos gerados por triggers, items ou regras LLD.
selectRelatedObject	query	Retorna uma propriedade `relatedObject` com o objeto que criou o evento. O tipo de objeto retornado depende do tipo de evento.
selectSuppressionData	query	Retorna uma propriedade `suppression_data` com a lista de manutenções ativas e supressões manuais: `maintenanceid` - `(ID)` ID da manutenção; `userid` - `(ID)` ID do usuário que suprimiu o evento; `suppress_until` - `(integer)` horário até o qual o evento será suprimido.
selectTags	query	Retorna uma propriedade `tags` com as tags do evento.
filter	object	Retorna apenas os resultados que correspondem exatamente ao filtro fornecido. Aceita um objeto, onde as chaves são nomes de propriedades e os valores são um único valor ou um array de valores para comparar. Não suporta propriedades do tipo de dado `text` data type.

Parâmetro	Tipo	Descrição
sortfield	string/array	Classifica o resultado pelas propriedades fornecidas. Valores possíveis: `eventid`, `objectid`, `clock`. Valores possíveis quando usado junto com `groupBy`: `objectid`. Valores possíveis quando usado junto com `countOutput` e `groupBy`: `objectid`, `rowscount`.
groupBy	string/array	Agrupa os resultados pelas propriedades fornecidas. As propriedades especificadas serão retornadas nos resultados. Valores possíveis: `objectid`.
countOutput	boolean	Esses parâmetros são descritos no comentário de referência.
editable	boolean	
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	
select_acknowledges (obsoleto)	query	Este parâmetro está obsoleto, use `selectAcknowledges` em seu lugar. Retorna uma propriedade `acknowledges` com as atualizações do evento. As atualizações de evento são classificadas em ordem cronológica inversa. O objeto de atualização de evento possui as seguintes propriedades: `acknowledgeid` - `(ID)` ID do reconhecimento; `userid` - `(ID)` ID do usuário que atualizou o evento; `clock` - `(timestamp)` horário em que o evento foi atualizado; `message` - `(string)` texto da mensagem; `action` - `(integer)` ação de atualização que foi realizada, consulte `event.acknowledge`; `old_severity` - `(integer)` gravidade do evento antes desta ação de atualização; `new_severity` - `(integer)` gravidade do evento após esta ação de atualização; `suppress_until` - `(timestamp)` horário até o qual o evento será suprimido; `taskid` - `(ID)` ID da tarefa se o evento atual estiver passando por uma alteração de classificação; `username` - `(string)` nome de usuário do usuário que atualizou o evento; `name` - `(string)` nome do usuário que atualizou o evento; `surname` - `(string)` sobrenome do usuário que atualizou o evento. Suporta `count`.
select_alerts (obsoleto)	query	Este parâmetro está obsoleto, use `selectAlerts` em seu lugar. Retorna uma propriedade `alerts` com os alertas gerados pelo evento. Os alertas são classificados em ordem cronológica inversa.

Valores de retorno

`(integer/array)` Retorna:

- um array de objetos;
- a contagem de objetos recuperados, se o parâmetro `countOutput` foi usado, mas o parâmetro `groupBy` não foi usado;
- um array de objetos com resultados de agregação, se o parâmetro `groupBy` foi usado.

Exemplos

Recuperando eventos de trigger

Recupere os eventos mais recentes do trigger ”22395”.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "event.get",
    "params": {
        "output": "extend",
        "selectAcknowledges": "extend",
        "selectSuppressionData": "extend",
        "selectTags": "extend",
        "objectids": "22395",
        "sortfield": ["clock", "eventid"],
        "sortorder": "DESC"
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "eventid": "20",
            "source": "0",
            "object": "0",
            "objectid": "22395",
            "clock": "1728658089",
            "value": "0",
            "acknowledged": "0",
            "ns": "461809482",
            "name": "Load average is too high (per CPU load over 1.5 for 5m)",
            "severity": "0",
            "r_eventid": "0",
            "c_eventid": "0",
            "correlationid": "0",
            "userid": "0",
            "cause_eventid": "0",
            "acknowledges": [],
            "opdata": "Load averages(1m 5m 15m): (1.6328 3.0522 2.2515), # of CPUs: 2",
            "suppression_data": [],
            "suppressed": "0",
            "tags": [
                {
                    "tag": "scope",
                    "value": "capacity"
                },
                {
                    "tag": "scope",
                    "value": "performance"
                },
                {
                    "tag": "component",
                    "value": "cpu"
                },
                {
                    "tag": "class",
                    "value": "os"
                },
                {
```

```
                "tag": "target",
                "value": "linux"
            }
        ],
        "urls": []
    },
    {
        "eventid": "4",
        "source": "0",
        "object": "0",
        "objectid": "22395",
        "clock": "1728657737",
        "value": "1",
        "acknowledged": "1",
        "ns": "460759366",
        "name": "Load average is too high (per CPU load over 1.5 for 5m)",
        "severity": "3",
        "r_eventid": "20",
        "c_eventid": "0",
        "correlationid": "0",
        "userid": "0",
        "cause_eventid": "0",
        "acknowledges": [
            {
                "acknowledgeid": "1",
                "userid": "1",
                "clock": "1728657938",
                "message": "Testing environment. Please disregard this alert.",
                "action": "38",
                "old_severity": "0",
                "new_severity": "0",
                "suppress_until": "1728744338",
                "taskid": "0",
                "username": "Admin",
                "name": "Zabbix",
                "surname": "Administrator"
            }
        ],
        "opdata": "Load averages(1m 5m 15m): (1.6328 3.0522 2.2515), # of CPUs: 2",
        "suppression_data": [
            {
                "maintenanceid": "0",
                "suppress_until": "1728744338",
                "userid": "1"
            }
        ],
        "suppressed": "1",
        "tags": [
            {
                "tag": "scope",
                "value": "capacity"
            },
            {
                "tag": "scope",
                "value": "performance"
            },
            {
                "tag": "component",
                "value": "cpu"
            },
            {
                "tag": "class",
```

```
                    "value": "os"
                },
                {
                    "tag": "target",
                    "value": "linux"
                }
            ],
            "urls": []
        }
    ],
    "id": 1
}
```

Recuperando eventos por período de tempo

Recupere todos os eventos que foram criados entre 17 e 18 de outubro de 2012, em ordem cronológica inversa.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "event.get",
    "params": {
        "output": "extend",
        "time_from": "1350432000",
        "time_till": "1350518400",
        "sortfield": ["clock", "eventid"],
        "sortorder": "DESC"
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "eventid": "20617",
            "source": "0",
            "object": "0",
            "objectid": "14282",
            "clock": "1350477816",
            "value": "1",
            "acknowledged": "0",
            "ns": "0",
            "name": "Less than 25% free in the history cache",
            "severity": "3",
            "r_eventid": "0",
            "c_eventid": "0",
            "correlationid": "0",
            "userid": "0",
            "cause_eventid": "0",
            "opdata": "",
            "suppressed": "0",
            "urls": []
        },
        {
            "eventid": "20616",
            "source": "0",
            "object": "0",
            "objectid": "14281",
            "clock": "1350477814",
            "value": "0",
            "acknowledged": "0",
```

```
            "ns": "0",
            "name": "Zabbix trapper processes more than 75% busy",
            "severity": "0",
            "r_eventid": "0",
            "c_eventid": "0",
            "correlationid": "0",
            "userid": "0",
            "cause_eventid": "0",
            "opdata": "",
            "suppressed": "0",
            "urls": []
        },
        {
            "eventid": "20615",
            "source": "0",
            "object": "0",
            "objectid": "14281",
            "clock": "1350477541",
            "value": "1",
            "acknowledged": "0",
            "ns": "0",
            "name": "Zabbix trapper processes more than 75% busy",
            "severity": "3",
            "r_eventid": "20616",
            "c_eventid": "0",
            "correlationid": "0",
            "userid": "0",
            "cause_eventid": "0",
            "opdata": "",
            "suppressed": "0",
            "urls": []
        }
    ],
    "id": 1
}
```


Recuperando eventos reconhecidos por um usuário específico

Recuperando eventos reconhecidos pelo usuário com ID=10

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "event.get",
    "params": {
        "output": "extend",
        "action": 2,
        "action_userids": [10],
        "selectAcknowledges": ["userid", "action"],
        "sortfield": ["eventid"],
        "sortorder": "DESC"
    },
    "id": 1
}
```

Resposta:


```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "eventid": "503",
            "source": "0",
            "object": "0",
```

```
            "objectid": "23162",
            "clock": "1747212236",
            "value": "1",
            "acknowledged": "1",
            "ns": "413470863",
            "name": "Number of installed packages has been changed",
            "severity": "2",
            "r_eventid": "0",
            "c_eventid": "0",
            "correlationid": "0",
            "userid": "0",
            "cause_eventid": "0",
            "acknowledges": [
                {
                    "userid": "10",
                    "action": "2"
                }
            ],
            "opdata": "",
            "suppressed": "0",
            "urls": []
        }
    ],
    "id": 1
}
```

Recuperando triggers principais com contagem de eventos de problema

Recupere as 5 triggers principais que possuem severidades ”Aviso”, ”Média”, ”Alta” ou ”Desastre”, juntamente com o número de eventos de problema dentro de um período de tempo especificado.

Requisição:

```
{
    "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
}
```

Resposta:

```
{
    "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
}
```

Veja também

- Alerta
- Item
- Host
- Regra LLD
- Serviço
- Trigger

Fonte

CEvent::get() em *ui/include/classes/api/services/CEvent.php*.

Expressão regular

Esta classe foi projetada para trabalhar com expressões regulares globais.

Referências de objetos:

- Expressão regular
- Expressões

Métodos disponíveis:

- regexp.create - criar novas expressões regulares
- regexp.delete - excluir expressões regulares
- regexp.get - recuperar expressões regulares
- regexp.update - atualizar expressões regulares

Objeto de expressão regular

Os seguintes objetos estão diretamente relacionados à API `regexp`.

Expressão regular

O objeto de expressão regular global possui as seguintes propriedades.

Propriedade	Tipo	Descrição
regexpid	ID	ID da expressão regular. Comportamento da propriedade: - *somente leitura* - *obrigatório* para operações de atualização
name	string	Nome da expressão regular. Comportamento da propriedade: - *obrigatório* para operações de criação
test_string	string	String de teste.

Expressões

O objeto de expressões possui as seguintes propriedades.

Propriedade	Tipo	Descrição
expression	string	Expressão regular. Comportamento da propriedade: - *obrigatório*
expression_type	integer	Tipo de expressão regular. Valores possíveis: 0 - String de caracteres incluída; 1 - Qualquer string de caracteres incluída; 2 - String de caracteres não incluída; 3 - Resultado é VERDADEIRO; 4 - Resultado é FALSO. Comportamento da propriedade: - *obrigatório*
exp_delimiter	string	Delimitador de expressão. Valor padrão: `","`. Valores possíveis: `","` ou `"."`, ou `"/"`. Comportamento da propriedade: - *suportado* se `expression_type` estiver definido como "Qualquer string de caracteres incluída"
case_sensitive	integer	Sensibilidade a maiúsculas/minúsculas. Valor padrão: 0. Valores possíveis: 0 - Não diferencia maiúsculas de minúsculas; 1 - Diferencia maiúsculas de minúsculas.

regexp.create

Descrição

`object regexp.create(object/array regularExpressions)`

Este método permite criar novas expressões regulares globais.

> **Note:**
> Este método está disponível apenas para tipos de usuário *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função de usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object/array)` Expressões regulares a serem criadas.

Além das propriedades padrão, o método aceita os seguintes parâmetros.

Parâmetro	Tipo	Descrição
expressions	array	Opções de expressões. Comportamento do parâmetro: - *obrigatório*

Valores de retorno

(`object`) Retorna um objeto contendo os IDs das expressões regulares criadas sob a propriedade `regexpids`.

Exemplos

Criando uma nova expressão regular global.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "regexp.create",
    "params": {
      "name": "Storage devices for SNMP discovery",
      "test_string": "/boot",
      "expressions": [
        {
          "expression": "^(Physical memory|Virtual memory|Memory buffers|Cached memory|Swap space)$",
          "expression_type": "4",
          "case_sensitive": "1"
        }
      ]
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "regexpids": [
            "16"
        ]
    },
    "id": 1
}
```

Fonte

CRegexp::create() in *ui/include/classes/api/services/CRegexp.php*.

regexp.delete

Descrição

`object regexp.delete(array regexpids)`

Este método permite excluir expressões regulares globais.

> **Note:**
> Este método está disponível apenas para tipos de usuário *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

(`array`) IDs das expressões regulares a serem excluídas.

Valores de retorno

(`object`) Retorna um objeto contendo os IDs das expressões regulares excluídas sob a propriedade `regexpids`

Exemplos

Excluindo várias expressões regulares globais.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "regexp.delete",
```

```
    "params": [
        "16",
        "17"
    ],
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "regexpids": [
            "16",
            "17"
        ]
    },
    "id": 1
}
```

Fonte

CRegexp::delete() in *ui/include/classes/api/services/CRegexp.php*.

regexp.get

Descrição

`integer/array regexp.get(object parameters)`

O método permite recuperar expressões regulares globais de acordo com os parâmetros fornecidos.

Note:
Este método está disponível apenas para *Super Admin*. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object)` Parâmetros que definem a saída desejada.

O método suporta os seguintes parâmetros.

Parâmetro	Tipo	Descrição
regexpids	ID/array	Retorna apenas as expressões regulares com os IDs fornecidos.
selectExpressions	query	Retorna uma propriedade `expressions`.
sortfield	string/array	Ordena o resultado pelas propriedades fornecidas. Valores possíveis: `regexpid`, `name`.
countOutput	boolean	Estes parâmetros são descritos no comentário de referência.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valores de retorno

`(integer/array)` Retorna:

- uma matriz de objetos;
- a contagem de objetos recuperados, se o parâmetro `countOutput` foi usado.

Exemplos

Recuperando expressões regulares globais.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "regexp.get",
    "params": {
        "output": ["regexpid", "name"],
        "selectExpressions": ["expression", "expression_type"],
        "regexpids": [1, 2],
        "preservekeys": true
    },
    "id": 1
}
```

Resposta:

```
{
  "jsonrpc": "2.0",
  "result": {
    "1": {
      "regexpid": "1",
      "name": "File systems for discovery",
      "expressions": [
        {
          "expression": "^(btrfs|ext2|ext3|ext4|reiser|xfs|ffs|ufs|jfs|jfs2|vxfs|hfs|apfs|refs|ntfs|fat32|
          "expression_type": "3"
        }
      ]
    },
    "2": {
      "regexpid": "2",
      "name": "Network interfaces for discovery",
      "expressions": [
        {
          "expression": "^Software Loopback Interface",
          "expression_type": "4"
        },
        {
          "expression": "^(In)?[Ll]oop[Bb]ack[0-9._]*$",
          "expression_type": "4"
        },
        {
          "expression": "^NULL[0-9.]*$",
          "expression_type": "4"
        },
        {
          "expression": "^[Ll]o[0-9.]*$",
          "expression_type": "4"
        },
        {
          "expression": "^[Ss]ystem$",
          "expression_type": "4"
        },
        {
          "expression": "^Nu[0-9.]*$",
          "expression_type": "4"
        }
      ]
```

```
    }
  },
  "id": 1
}
```

Fonte

CRegexp::get() in *ui/include/classes/api/services/CRegexp.php*.

regexp.update

Descrição

`object regexp.update(object/array regularExpressions)`

Este método permite atualizar expressões regulares globais existentes.

> **Note:**
> Este método está disponível apenas para tipos de usuário *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object/array)` Propriedades da expressão regular a serem atualizadas.

A propriedade `regexpid` deve ser definida para cada objeto, todas as outras propriedades são opcionais. Somente as propriedades passadas serão atualizadas, todas as outras permanecerão inalteradas.

Além das propriedades padrão, o método aceita os seguintes parâmetros.

Parâmetro	Tipo	Descrição
expressions	array	Opções de expressões.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs das expressões regulares atualizadas na propriedade `regexpids`

Exemplos

Atualizando a expressão regular global para descoberta de sistemas de arquivos.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "regexp.update",
    "params": {
      "regexpid": "1",
      "name": "File systems for discovery",
      "test_string": "",
      "expressions": [
        {
          "expression": "^(btrfs|ext2|ext3|ext4|reiser|xfs|ffs|ufs|jfs|jfs2|vxfs|hfs|apfs|refs|zfs)$",
          "expression_type": "3",
          "exp_delimiter": ",",
          "case_sensitive": "0"
        },
        {
          "expression": "^(ntfs|fat32|fat16)$",
          "expression_type": "3",
          "exp_delimiter": ",",
          "case_sensitive": "0"
        }
      ]
    },
```

```
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "regexpids": [
            "1"
        ]
    },
    "id": 1
}
```

Fonte

CRegexp::update() in *ui/include/classes/api/services/CRegexp.php*.

Função

Esta classe foi projetada para trabalhar com funções de usuário.

Referências de objetos:

- Função
- Regras de função
  - Elemento da interface
  - Serviço
  - Tag de serviço
  - Módulo
  - Ação

Métodos disponíveis:

- role.create - criar novas funções de usuário
- role.delete - excluir funções de usuário
- role.get - recuperar funções de usuário
- role.update - atualizar funções de usuário

Objeto role

Os seguintes objetos estão diretamente relacionados à API `role`.

Role

O objeto role possui as seguintes propriedades:

Propriedade	Tipo	Descrição
roleid	ID	ID da role. Comportamento da propriedade: - *somente leitura* - *obrigatório* para operações de atualização
name	string	Nome da role. Comportamento da propriedade: - *obrigatório* para operações de criação

Propriedade	Tipo	Descrição
type	integer	Tipo de usuário. Valores possíveis: 1 - *(padrão)* Usuário; 2 - Admin; 3 - Super admin. Comportamento da propriedade: - *obrigatório* para operações de criação
readonly	integer	Se a role é somente leitura. Valores possíveis: 0 - *(padrão)* Não; 1 - Sim. Comportamento da propriedade: - *somente leitura*

Regras de função

O objeto de regras de função possui as seguintes propriedades:

Propriedade	Tipo	Descrição
ui	array	Array de objetos de elemento da interface.
ui.default_access	integer	Se o acesso a novos elementos da interface está habilitado. Valores possíveis: 0 - Desabilitado; 1 - *(padrão)* Habilitado.
services.read.mode	integer	Acesso somente leitura aos serviços. Valores possíveis: 0 - Acesso somente leitura aos serviços especificados pelas propriedades `services.read.list` ou correspondentes à `services.read.tag`; 1 - *(padrão)* Acesso somente leitura a todos os serviços.
services.read.list	array	Array de objetos de Serviço. Os serviços especificados, incluindo serviços filhos, terão acesso somente leitura concedido à função de usuário. O acesso somente leitura não substituirá o acesso de leitura e gravação aos serviços. Comportamento da propriedade: - *suportado* se `services.read.mode` estiver definido como "0"
services.read.tag	object	Array de objetos de Tag de serviço. Os serviços correspondentes à tag, incluindo serviços filhos, terão acesso somente leitura concedido à função de usuário. O acesso somente leitura não substituirá o acesso de leitura e gravação aos serviços. Comportamento da propriedade: - *suportado* se `services.read.mode` estiver definido como "0"
services.write.mode	integer	Acesso de leitura e gravação aos serviços. Valores possíveis: 0 - *(padrão)* Acesso de leitura e gravação aos serviços especificados pelas propriedades `services.write.list` ou correspondentes à `services.write.tag`; 1 - Acesso de leitura e gravação a todos os serviços.

Propriedade	Tipo	Descrição
services.write.list	array	Array de objetos de Serviço. Os serviços especificados, incluindo serviços filhos, terão acesso de leitura e gravação concedido à função de usuário. O acesso de leitura e gravação substituirá o acesso somente leitura aos serviços. Comportamento da propriedade: - *suportado* se `services.write.mode` estiver definido como "0"
services.write.tag	object	Array de objetos de Tag de serviço. Os serviços correspondentes à tag, incluindo serviços filhos, terão acesso de leitura e gravação concedido à função de usuário. O acesso de leitura e gravação substituirá o acesso somente leitura aos serviços. Comportamento da propriedade: - *suportado* se `services.write.mode` estiver definido como "0"
modules	array	Array de objetos de módulo.
modules.default_access	integer	Se o acesso a novos módulos está habilitado. Valores possíveis: 0 - Desabilitado; 1 - *(padrão)* Habilitado.
api.access	integer	Se o acesso à API está habilitado. Valores possíveis: 0 - Desabilitado; 1 - *(padrão)* Habilitado.
api.mode	integer	Modo de tratamento dos métodos da API listados na propriedade `api`. Valores possíveis: 0 - *(padrão)* Lista de negação; 1 - Lista de permissão.
api	array	Array de métodos da API.
actions	array	Array de objetos de ação.
actions.default_access	integer	Se o acesso a novas ações está habilitado. Valores possíveis: 0 - Desabilitado; 1 - *(padrão)* Habilitado.

Elemento da interface do usuário

O objeto de elemento da interface do usuário possui as seguintes propriedades:

Propriedade	Tipo	Descrição
name	string	Nome do elemento da interface do usuário. Valores possíveis se o `type` do objeto Role estiver definido como "User", "Admin" ou "Super admin": `monitoring.dashboard` - *Dashboards*; `monitoring.problems` - *Monitoramento → Problemas*; `monitoring.hosts` - *Monitoramento → Hosts*; `monitoring.latest_data` - *Monitoramento → Dados mais recentes*; `monitoring.maps` - *Monitoramento → Mapas*; `services.services` - *Serviços → Serviços*; `services.sla_report` - *Serviços → Relatório de SLA*; `inventory.overview` - *Inventário → Visão geral*; `inventory.hosts` - *Inventário → Hosts*; `reports.availability_report` - *Relatórios → Relatório de disponibilidade*; `reports.top_triggers` - *Relatórios → Top 100 triggers*. Valores possíveis se o `type` do objeto Role estiver definido como "Admin" ou "Super admin": `monitoring.discovery` - *Monitoramento → Descoberta*; `services.sla` - *Serviços → SLA*; `reports.scheduled_reports` - *Relatórios → Relatórios agendados*; `reports.notifications` - *Relatórios → Notificações*; `configuration.template_groups` - *Coleta de dados → Grupos de templates*; `configuration.host_groups` - *Coleta de dados → Grupos de hosts*; `configuration.templates` - *Coleta de dados → Templates*; `configuration.hosts` - *Coleta de dados → Hosts*; `configuration.maintenance` - *Coleta de dados → Manutenção*; `configuration.discovery` - *Coleta de dados → Descoberta*; `configuration.trigger_actions` - *Alertas → Ações → Ações de trigger*; `configuration.service_actions` - *Alertas → Ações → Ações de serviço*; `configuration.discovery_actions` - *Alertas → Ações → Ações de descoberta*; `configuration.autoregistration_actions` - *Alertas → Ações → Ações de autorregistro*; `configuration.internal_actions` - *Alertas → Ações → Ações internas*. Valores possíveis se o `type` do objeto Role estiver definido como "Super admin": `reports.system_info` - *Relatórios → Informações do sistema*; `reports.audit` - *Relatórios → Log de auditoria*; `reports.action_log` - *Relatórios → Log de ações*; `configuration.event_correlation` - *Coleta de dados → Correlação de eventos*; `administration.media_types` - *Alertas → Tipos de mídia*; `administration.scripts` - *Alertas → Scripts*; `administration.user_groups` - *Usuários → Grupos de usuários*; `administration.user_roles` - *Usuários → Funções de usuário*; `administration.users` - *Usuários → Usuários*; `administration.api_tokens` - *Usuários → Tokens de API*; `administration.authentication` - *Usuários → Autenticação*; `administration.general` - *Administração → Geral*; `administration.audit_log` - *Administração → Log de auditoria*; `administration.housekeeping` - *Administração → Limpeza*; `administration.proxy_groups` - *Administração → Grupos de proxy*; `administration.proxies` - *Administração → Proxies*; `administration.macros` - *Administração → Macros*; `administration.queue` - *Administração → Fila*. Comportamento da propriedade:

Propriedade	Tipo	Descrição
status	integer	Se o acesso ao elemento da interface do usuário está habilitado. Valores possíveis: 0 - Desabilitado; 1 - *(padrão)* Habilitado.

Serviço

Propriedade	Tipo	Descrição
serviceid	ID	ID do serviço. Comportamento da propriedade: - *obrigatório*

Tag de serviço

Propriedade	Tipo	Descrição
tag	string	Nome da tag. Se uma string vazia for especificada, a tag de serviço não será usada para correspondência de serviço. Comportamento da propriedade: - *obrigatório*
value	string	Valor da tag. Se nenhum valor ou uma string vazia for especificada, apenas o nome da tag será usado para correspondência de serviço.

Módulo

O objeto módulo possui as seguintes propriedades:

Propriedade	Tipo	Descrição
moduleid	ID	ID do módulo. Comportamento da propriedade: - *obrigatório*
status	integer	Se o acesso ao módulo está habilitado. Valores possíveis: 0 - Desabilitado; 1 - *(padrão)* Habilitado.

Ação

O objeto action possui as seguintes propriedades:

Propriedade	Tipo	Descrição
name	string	Nome da ação. Valores possíveis se o `type` do objeto Role estiver definido como "User", "Admin" ou "Super admin": `edit_dashboards` - Criar e editar dashboards; `edit_maps` - Criar e editar mapas; `add_problem_comments` - Adicionar comentários de problema; `change_severity` - Alterar a severidade do problema; `acknowledge_problems` - Reconhecer problemas; `suppress_problems` - Suprimir problemas; `close_problems` - Fechar problemas; `execute_scripts` - Executar scripts; `manage_api_tokens` - Gerenciar tokens de API; `change_problem_ranking` - Alterar a classificação do problema de causa para sintoma e vice-versa. Valores possíveis se o `type` do objeto Role estiver definido como "Admin" ou "Super admin": `edit_maintenance` - Criar e editar manutenções; `manage_scheduled_reports` - Gerenciar relatórios agendados, `manage_sla` - Gerenciar SLA. Valores possíveis se o `type` do objeto Role estiver definido como "User" ou "Admin": `invoke_execute_now` - permite executar verificações de item para usuários que possuem apenas permissões de leitura no host. Comportamento da propriedade: - *obrigatório*
status	integer	Se o acesso para executar a ação está habilitado. Valores possíveis: 0 - Desabilitado; 1 - *(padrão)* Habilitado.

role.create

Descrição

`object role.create(object/array roles)`

Este método permite criar novas funções.

Note:
Este método está disponível apenas para o tipo de usuário *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object/array)` Funções a serem criadas.

Além das propriedades padrão da função, o método aceita os seguintes parâmetros.

Parâmetro	Tipo	Descrição
rules	array	Regras de função a serem criadas para a função.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos papéis criados na propriedade `roleids`. A ordem dos IDs retornados corresponde à ordem das funções passadas.

Exemplos

Criando uma função

Crie uma função com o tipo "Usuário" e acesso negado a dois elementos da interface do usuário.

Requisição:

```
{
    "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
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "roleids": [
            "5"
        ]
    },
    "id": 1
}
```

Veja também

- Regras de função
- Elemento da interface
- Módulo
- Ação

Fonte

CRole::create() in *ui/include/classes/api/services/CRole.php*.

role.delete

Descrição

`object role.delete(array roleids)`

Este método permite excluir funções.

> **Note:**
> Este método está disponível apenas para o tipo de usuário *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(array)` IDs das funções a serem excluídas.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos papéis excluídos na propriedade `roleids`.

Exemplos

Excluindo múltiplas funções de usuário

Exclua duas funções de usuário.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "role.delete",
    "params": [
        "4",
        "5"
    ],
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "roleids": [
            "4",
            "5"
        ]
    },
    "id": 1
}
```

Fonte

CRole::delete() in *ui/include/classes/api/services/CRole.php*.

role.get

Descrição

`integer/array role.get(object parameters)`

O método permite recuperar funções de acordo com os parâmetros fornecidos.

> **Note:**
> Este método está disponível para usuários de qualquer tipo. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object)` Parâmetros que definem a saída desejada.

O método suporta os seguintes parâmetros.

Parâmetro	Tipo	Descrição
roleids	ID/array	Retornar apenas funções com os IDs fornecidos.
selectRules	query	Retornar uma propriedade `rules` com as regras da função.
selectUsers	query	Retornar uma propriedade `users` com os usuários aos quais a função está atribuída. Veja `user.get` para restrições baseadas no tipo de usuário.
sortfield	string/array	Classificar o resultado pelas propriedades fornecidas. Valores possíveis: `roleid`, `name`.
countOutput	boolean	Esses parâmetros são descritos no comentário de referência.
editable	boolean	
excludeSearch	boolean	

Parâmetro	Tipo	Descrição
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valores de retorno

`(integer/array)` Retorna:

- uma matriz de objetos;
- a contagem de objetos recuperados, se o parâmetro `countOutput` foi usado.

Exemplos

Recuperando dados de função

Recupere os dados da função "Super admin role" e suas regras de acesso.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "role.get",
    "params": {
        "output": "extend",
        "selectRules": "extend",
        "roleids": "3"
    },
    "id": 1
}
```

Resposta:

```
{
    "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"
            }
        ],
        "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"
                    }
                ],
                "actions.default_access": "1"
            }
        }
    ],
    "id": 1
}
```


Veja também

- Regras de função
- Usuário

Fonte

CRole::get() in *ui/include/classes/api/services/CRole.php*.

role.update

Descrição

`object role.update(object/array roles)`

Este método permite atualizar funções existentes.

> **Note:**
> Este método está disponível apenas para o tipo de usuário *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object/array)` Propriedades da role a serem atualizadas.

A propriedade `roleid` deve ser definida para cada role, todas as outras propriedades são opcionais. Somente as propriedades passadas serão atualizadas, todas as outras permanecerão inalteradas.

Além das propriedades padrão da role, o método aceita os seguintes parâmetros.

Parâmetro	Tipo	Descrição
rules	array	Regras de acesso a serem atualizadas para a role.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos papéis atualizados na propriedade `roleids`.

Exemplos

Desabilitando a capacidade de executar scripts

Atualize a função com o ID "5", desabilite a capacidade de executar scripts.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "role.update",
    "params": [
        {
            "roleid": "5",
            "rules": {
                "actions": [
                    {
                        "name": "execute_scripts",
                        "status": "0"
                    }
                ]
            }
        }
    ],
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "roleids": [
            "5"
        ]
    },
    "id": 1
}
```

Limitando o acesso à API

Atualize a função com ID ”5”, negando a chamada de qualquer método ”create”, ”update” ou ”delete”.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "role.update",
    "params": [
        {
            "roleid": "5",
            "rules": {
                "api.access": "1",
                "api.mode": "0",
                "api": ["*.create", "*.update", "*.delete"]
            }
        }
    ],
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "roleids": [
            "5"
        ]
    },
    "id": 1
}
```

Fonte

CRole::update() in *ui/include/classes/api/services/CRole.php*.

Graph

Esta classe foi projetada para trabalhar com gráficos.

Referências de objetos:

- Graph

Métodos disponíveis:

- graph.create - criar novos gráficos
- graph.delete - excluir gráficos
- graph.get - recuperar gráficos
- graph.update - atualizar gráficos

Objeto gráfico

Os seguintes objetos estão diretamente relacionados à API `graph`.

Gráfico

O objeto gráfico possui as seguintes propriedades.

Propriedade	Tipo	Descrição
graphid	ID	ID do gráfico. Comportamento da propriedade: - *somente leitura* - *obrigatório* para operações de atualização

Propriedade	Tipo	Descrição
height	integer	Altura do gráfico em pixels. Comportamento da propriedade: - *obrigatório* para operações de criação
name	string	Nome do gráfico. Comportamento da propriedade: - *obrigatório* para operações de criação
width	integer	Largura do gráfico em pixels. Comportamento da propriedade: - *obrigatório* para operações de criação
flags	integer	Origem do gráfico. Valores possíveis: 0 - *(padrão)* um gráfico simples; 4 - um gráfico descoberto. Comportamento da propriedade: - *somente leitura*
graphtype	integer	Tipo de layout do gráfico. Valores possíveis: 0 - *(padrão)* normal; 1 - empilhado; 2 - pizza; 3 - pizza explodida.
percent_left	float	Percentil esquerdo. Padrão: 0.
percent_right	float	Percentil direito. Padrão: 0.
show_3d	integer	Se deve mostrar gráficos de pizza e pizza explodida em 3D. Valores possíveis: 0 - *(padrão)* mostrar em 2D; 1 - mostrar em 3D.
show_legend	integer	Se deve mostrar a legenda no gráfico. Valores possíveis: 0 - ocultar; 1 - *(padrão)* mostrar.
show_work_period	integer	Se deve mostrar o período de trabalho no gráfico. Valores possíveis: 0 - ocultar; 1 - *(padrão)* mostrar.
show_triggers	integer	Se deve mostrar a linha do trigger no gráfico. Valores possíveis: 0 - ocultar; 1 - *(padrão)* mostrar.
templateid	ID	ID do gráfico do template pai. Comportamento da propriedade: - *somente leitura*
yaxismax	float	O valor máximo fixo para o eixo Y. Padrão: 100.

Propriedade	Tipo	Descrição
yaxismin	float	O valor mínimo fixo para o eixo Y. Padrão: 0.
ymax_itemid	ID	ID do item que é usado como valor máximo para o eixo Y. Se um usuário não tiver acesso ao item especificado, o gráfico será renderizado como se `ymax_type` estivesse definido como "calculado".
ymax_type	integer	Método de cálculo do valor máximo para o eixo Y. Valores possíveis: 0 - *(padrão)* calculado; 1 - fixo; 2 - item.
ymin_itemid	ID	ID do item que é usado como valor mínimo para o eixo Y. Se um usuário não tiver acesso ao item especificado, o gráfico será renderizado como se `ymin_type` estivesse definido como "calculado".
ymin_type	integer	Método de cálculo do valor mínimo para o eixo Y. Valores possíveis: 0 - *(padrão)* calculado; 1 - fixo; 2 - item.
uuid	string	Identificador único universal, usado para vincular gráficos importados a já existentes. Gerado automaticamente, se não for fornecido. Comportamento da propriedade: - *suportado* se o gráfico pertencer a um template

graph.create

Descrição

`object graph.create(object/array graphs)`

Este método permite criar novos gráficos.

Note:
Este método está disponível apenas para os tipos de usuário *Admin* e *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object/array)` Gráficos a serem criados.

Além das propriedades padrão do gráfico, o método aceita os seguintes parâmetros.

Parâmetro	Tipo	Descrição
gitems	array	Itens do gráfico a serem criados para o gráfico. Comportamento do parâmetro: - *obrigatório*

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos gráficos criados sob a propriedade `graphids`. A ordem dos IDs retornados corresponde à ordem dos gráficos passados.

Exemplos

Criando um gráfico

Crie um gráfico com dois items.

Requisição:

```
{
    "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
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "graphids": [
            "652"
        ]
    },
    "id": 1
}
```

Veja também

- Item de gráfico

Fonte

CGraph::create() em *ui/include/classes/api/services/CGraph.php*.

graph.delete

Descrição

`object graph.delete(array graphIds)`

Este método permite excluir gráficos.

> **Note:**
> Este método está disponível apenas para os tipos de usuário *Admin* e *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(array)` IDs dos gráficos a serem excluídos.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos gráficos excluídos sob a propriedade `graphids`.

Exemplos

Excluindo múltiplos gráficos

Exclua dois gráficos.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "graph.delete",
    "params": [
        "652",
        "653"
    ],
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "graphids": [
            "652",
            "653"
        ]
    },
    "id": 1
}
```

Fonte

CGraph::delete() em *ui/include/classes/api/services/CGraph.php*.

graph.get

Descrição

`integer/array graph.get(object parameters)`

O método permite recuperar gráficos de acordo com os parâmetros fornecidos.

> **Note:**
> Este método está disponível para usuários de qualquer tipo. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object)` Parâmetros que definem a saída desejada.

O método suporta os seguintes parâmetros.

Parâmetro	Tipo	Descrição
graphids	ID/array	Retorna apenas gráficos com os IDs fornecidos.
groupids	ID/array	Retorna apenas gráficos que pertencem a hosts ou templates nos grupos de hosts ou grupos de templates fornecidos.
templateids	ID/array	Retorna apenas gráficos que pertencem aos templates fornecidos.
hostids	ID/array	Retorna apenas gráficos que pertencem aos hosts fornecidos.
itemids	ID/array	Retorna apenas gráficos que contêm os items fornecidos.
templated	boolean	Se definido como `true`, retorna apenas gráficos que pertencem a templates.
inherited	boolean	Se definido como `true`, retorna apenas gráficos herdados de um template.
expandName	flag	Expande macros no nome do gráfico.
selectHostGroups	query	Retorna uma propriedade `hostgroups` com os grupos de hosts aos quais o gráfico pertence.
selectTemplateGroups	query	Retorna uma propriedade `templategroups` com os grupos de templates aos quais o gráfico pertence.

Parâmetro	Tipo	Descrição
selectTemplates	query	Retorna uma propriedade `templates` com os templates aos quais o gráfico pertence.
selectHosts	query	Retorna uma propriedade `hosts` com os hosts aos quais o gráfico pertence.
selectItems	query	Retorna uma propriedade `items` com os items usados no gráfico.
selectGraphDiscovery	query	Retorna uma propriedade `graphDiscovery` com o objeto de descoberta de gráfico. O objeto de descoberta de gráfico vincula o gráfico a um protótipo de gráfico a partir do qual ele foi criado. Possui as seguintes propriedades: `graphid` - `(ID)` ID do gráfico; `parent_graphid` - `(ID)` ID do protótipo de gráfico a partir do qual o gráfico foi criado; `lastcheck` - `(timestamp)` hora em que o gráfico foi descoberto pela última vez; `status` - `(int)` status da descoberta do gráfico: 0 - *(padrão)* gráfico descoberto, 1 - gráfico não é mais descoberto; `ts_delete` - `(timestamp)` hora em que um gráfico que não é mais descoberto será excluído.
selectGraphItems	query	Retorna uma propriedade `gitems` com os items usados no gráfico.
selectDiscoveryRule	query	Retorna uma propriedade `discoveryRule` com a regra de descoberta de baixo nível que criou o gráfico.
filter	object	Retorna apenas os resultados que correspondem exatamente ao filtro fornecido. Aceita um objeto, onde as chaves são nomes de propriedades e os valores são um único valor ou um array de valores para comparar. Não suporta propriedades do tipo de dado `text` data type. Suporta propriedades adicionais: `host` - nome técnico do host ao qual o gráfico pertence; `hostid` - ID do host ao qual o gráfico pertence.
sortfield	string/array	Ordena o resultado pelas propriedades fornecidas. Valores possíveis: `graphid`, `name`, `graphtype`.
countOutput	boolean	Esses parâmetros são descritos no comentário de referência.
editable	boolean	
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	
selectGroups (obsoleto)	query	Este parâmetro está obsoleto, use `selectHostGroups` ou `selectTemplateGroups` em seu lugar. Retorna uma propriedade `groups` com os grupos de hosts e grupos de templates aos quais o gráfico pertence.

Valores de retorno

`(integer/array)` Retorna:

- uma matriz de objetos;
- a contagem de objetos recuperados, se o parâmetro `countOutput` tiver sido usado.

Exemplos

Recuperando gráficos de hosts

Recupere todos os gráficos do host ”10107” e classifique-os por nome.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "graph.get",
    "params": {
        "output": "extend",
        "hostids": 10107,
        "sortfield": "name"
    },
    "id": 1
}
```

Resposta:

```
{
    "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
}
```


Veja também

- Regra de descoberta
- Item do gráfico
- Item
- Host
- Grupo de hosts
- Template
- Grupo de templates

Fonte

CGraph::get() em *ui/include/classes/api/services/CGraph.php*.

graph.update

Descrição

`object graph.update(object/array graphs)`

Este método permite atualizar gráficos existentes.

Note:
Este método está disponível apenas para os tipos de usuário *Admin* e *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object/array)` Propriedades do gráfico a serem atualizadas.

A propriedade `graphid` deve ser definida para cada gráfico, todas as outras propriedades são opcionais. Somente as propriedades passadas serão atualizadas, todas as outras permanecerão inalteradas.

Além das propriedades padrão do gráfico, o método aceita os seguintes parâmetros.

Parâmetro	Tipo	Descrição
gitems	array	Itens do gráfico para substituir os itens do gráfico existentes. Se um item do gráfico tiver a propriedade `gitemid` definida, ele será atualizado; caso contrário, um novo item do gráfico será criado.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos gráficos atualizados sob a propriedade `graphids`.

Exemplos

Definindo o máximo para a escala Y

Defina o máximo da escala Y para um valor fixo de 100.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "graph.update",
    "params": {
        "graphid": "439",
        "ymax_type": 1,
        "yaxismax": 100
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "graphids": [
            "439"
        ]
    },
    "id": 1
}
```

Fonte

CGraph::update() em *ui/include/classes/api/services/CGraph.php*.

Grupo de hosts

Esta classe foi projetada para trabalhar com grupos de hosts.

Referências de objetos:

- Grupo de hosts

Métodos disponíveis:

- hostgroup.create - criar novos grupos de hosts

- hostgroup.delete - excluir grupos de hosts
- hostgroup.get - recuperar grupos de hosts
- hostgroup.massadd - adicionar objetos relacionados aos grupos de hosts
- hostgroup.massremove - remover objetos relacionados dos grupos de hosts
- hostgroup.massupdate - substituir ou remover objetos relacionados dos grupos de hosts
- hostgroup.propagate - propagar permissões e filtros de tags para subgrupos dos grupos de hosts
- hostgroup.update - atualizar grupos de hosts

Objeto de grupo de hosts

Os seguintes objetos estão diretamente relacionados à API `hostgroup`.

Grupo de hosts

O objeto grupo de hosts possui as seguintes propriedades.

Propriedade	Tipo	Descrição
groupid	ID	ID do grupo de hosts. Comportamento da propriedade: - *somente leitura* - *obrigatório* para operações de atualização
name	string	Nome do grupo de hosts. Comportamento da propriedade: - *obrigatório* para operações de criação
flags	integer	Origem do grupo de hosts. Valores possíveis: 0 - um grupo de hosts comum; 4 - um grupo de hosts descoberto. Comportamento da propriedade: - *somente leitura*
uuid	string	Identificador universal único, usado para vincular grupos de hosts importados aos já existentes. Gerado automaticamente, se não fornecido.

hostgroup.create

Descrição

`object hostgroup.create(object/array hostGroups)`

Este método permite criar novos grupos de hosts.

> **Note:**
> Este método está disponível apenas para o tipo de usuário *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object/array)` Grupos de hosts a serem criados.

O método aceita grupos de hosts com as propriedades padrão do grupo de hosts.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs do host criado grupos sob a propriedade `groupids`. A ordem dos IDs retornados corresponde à ordem dos grupos de hosts passados.

Exemplos

Criando um grupo de hosts

Crie um grupo de hosts chamado ”Linux servers”.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "hostgroup.create",
    "params": {
        "name": "Linux servers"
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
            "107819"
        ]
    },
    "id": 1
}
```

Fonte

CHostGroup::create() em *ui/include/classes/api/services/CHostGroup.php*.

hostgroup.delete

Descrição

`object hostgroup.delete(array hostGroupIds)`

Este método permite excluir grupos de hosts.

Um grupo de hosts não pode ser excluído se:

- ele contiver hosts que pertencem apenas a este grupo;
- ele estiver marcado como interno;
- ele for usado por um protótipo de host;
- ele for usado em um script global;
- ele for usado em uma condição de correlação.

Note:
Este método está disponível apenas para os tipos de usuário *Admin* e *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função de usuário. Consulte Funções de usuário para obter mais informações.

Parâmetros

`(array)` IDs dos grupos de hosts a serem excluídos.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs do host excluído grupos sob a propriedade `groupids`.

Exemplos

Excluindo vários grupos de hosts

Exclua dois grupos de hosts.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "hostgroup.delete",
    "params": [
        "107824",
```

```
        "107825"
    ],
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
            "107824",
            "107825"
        ]
    },
    "id": 1
}
```

Fonte

CHostGroup::delete() in *ui/include/classes/api/services/CHostGroup.php*.

hostgroup.get

Descrição

`integer/array hostgroup.get(object parameters)`

O método permite recuperar grupos de hosts de acordo com os parâmetros fornecidos.

> **Note:**
> Este método está disponível para usuários de qualquer tipo. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object)` Parâmetros que definem a saída desejada.

O método suporta os seguintes parâmetros.

Parâmetro	Tipo	Descrição
graphids	ID/array	Retorna apenas grupos de hosts que contenham hosts com os gráficos informados.
groupids	ID/array	Retorna apenas grupos de hosts com os IDs de grupo de host informados.
hostids	ID/array	Retorna apenas grupos de hosts que contenham os hosts informados.
maintenanceids	ID/array	Retorna apenas grupos de hosts que são afetados pelas manutenções informadas.
triggerids	ID/array	Retorna apenas grupos de hosts que contenham hosts com os triggers informados.
with_graphs	boolean	Retorna apenas grupos de hosts que contenham hosts com gráficos.
with_graph_prototypes	boolean	Retorna apenas grupos de hosts que contenham hosts com protótipos de gráficos.
with_hosts	boolean	Retorna apenas grupos de hosts que contenham hosts.
with_httptests	boolean	Retorna apenas grupos de hosts que contenham hosts com verificações web. Sobrescreve o parâmetro `with_monitored_httptests`.
with_items	boolean	Retorna apenas grupos de hosts que contenham hosts com items. Sobrescreve os parâmetros `with_monitored_items` e `with_simple_graph_items`.

Parâmetro	Tipo	Descrição
with_item_prototypes	boolean	Retorna apenas grupos de hosts que contenham hosts com protótipos de item. Sobrescreve o parâmetro `with_simple_graph_item_prototypes`.
with_simple_graph_item_prototypes	boolean	Retorna apenas grupos de hosts que contenham hosts com protótipos de item, que estão habilitados para criação e possuem tipo de informação numérica.
with_monitored_httptests	boolean	Retorna apenas grupos de hosts que contenham hosts com verificações web habilitadas.
with_monitored_hosts	boolean	Retorna apenas grupos de hosts que contenham hosts monitorados.
with_monitored_items	boolean	Retorna apenas grupos de hosts que contenham hosts com items habilitados. Sobrescreve o parâmetro `with_simple_graph_items`.
with_monitored_triggers	boolean	Retorna apenas grupos de hosts que contenham hosts com triggers habilitados. Todos os items usados no trigger também devem estar habilitados.
with_simple_graph_items	boolean	Retorna apenas grupos de hosts que contenham hosts com items numéricos.
with_triggers	boolean	Retorna apenas grupos de hosts que contenham hosts com triggers. Sobrescreve o parâmetro `with_monitored_triggers`.
selectDiscoveryRules	query	Retorna uma propriedade `discoveryRules` com as regras LLD que descobriram o grupo de hosts.
selectGroupDiscoveries	query	Retorna uma propriedade `groupDiscoveries` com os objetos de descoberta de grupo de hosts. Cada objeto de descoberta de grupo de hosts é um protótipo de grupo de hosts vinculado ao grupo de hosts descoberto e possui as seguintes propriedades: `parent_group_prototypeid` - `(ID)` ID do protótipo de grupo de hosts a partir do qual o grupo de hosts foi descoberto; `name` - `(string)` nome do protótipo de grupo de hosts; `lastcheck` - `(timestamp)` hora em que o grupo de hosts foi descoberto pela última vez; `ts_delete` - `(timestamp)` hora em que o grupo de hosts que não é mais descoberto será excluído; `status` - `(int)` status da descoberta do grupo de hosts: 0 - *(padrão)* grupo de hosts foi descoberto, 1 - grupo de hosts não é mais descoberto.
selectHostPrototypes	query	Retorna uma propriedade `hostPrototypes` com os protótipos de host que descobriram este grupo de hosts.
selectHosts	query	Retorna uma propriedade `hosts` com os hosts que pertencem ao grupo de hosts. Suporta `count`.
limitSelects	integer	Limita o número de registros retornados por subconsultas. Aplica-se às seguintes subconsultas: `selectHosts` - os resultados serão classificados por `host`.
sortfield	string/array	Classifica o resultado pelas propriedades informadas. Valores possíveis: `groupid`, `name`.
countOutput	boolean	Esses parâmetros são descritos no comentário de referência.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	

Parâmetro	Tipo	Descrição
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	
monitored_hosts (obsoleto)	flag	Este parâmetro está obsoleto, por favor use `with_monitored_hosts` em seu lugar. Retorna apenas grupos de hosts que contenham hosts monitorados.
real_hosts (obsoleto)	flag	Este parâmetro está obsoleto, por favor use `with_hosts` em seu lugar. Retorna apenas grupos de hosts que contenham hosts.

Valores de retorno

`(integer/array)` Retorna:

- um array de objetos
- a contagem de objetos recuperados, se o parâmetro `countOutput` tiver sido usado

Exemplos

Recuperando dados pelo nome

Recupere todos os dados sobre dois grupos de hosts chamados ”Zabbix servers” e ”Linux servers”.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "hostgroup.get",
    "params": {
        "output": "extend",
        "filter": {
            "name": [
                "Zabbix servers",
                "Linux servers"
            ]
        }
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "groupid": "2",
            "name": "Linux servers",
            "flags": "0",
            "uuid": "dc579cd7a1a34222933f24f52a68bcd8"
        },
        {
            "groupid": "4",
            "name": "Zabbix servers",
            "flags": "0",
            "uuid": "6f6799aa69e844b4b3918f779f2abf08"
        }
    ],
    "id": 1
}
```

Veja também

- Host

Fonte

CHostGroup::get() em *ui/include/classes/api/services/CHostGroup.php*.

hostgroup.massadd

Descrição

`object hostgroup.massadd(object parameters)`

Este método permite adicionar simultaneamente vários objetos relacionados a todos os grupos de hosts fornecidos.

> **Note:**
> Este método está disponível apenas para os tipos de usuário *Admin* e *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object)` Parâmetros contendo os IDs dos grupos de hosts a serem atualizados e os objetos a serem adicionados a todos os grupos de hosts.

O método aceita os seguintes parâmetros.

Parâmetro	Tipo	Descrição
groups	object/array	Grupos de hosts a serem atualizados. Os grupos de hosts devem ter apenas a propriedade `groupid` definida. Comportamento do parâmetro: - *obrigatório*
hosts	object/array	Hosts a serem adicionados a todos os grupos de hosts. Os hosts devem ter apenas a propriedade `hostid` definida.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos grupos de hosts atualizados na propriedade `groupids`.

Exemplos

Adicionando hosts a grupos de hosts

Adicione dois hosts a grupos de hosts com IDs 5 e 6.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "hostgroup.massadd",
    "params": {
        "groups": [
            {
                "groupid": "5"
            },
            {
                "groupid": "6"
            }
        ],
        "hosts": [
            {
                "hostid": "30050"
            },
            {
                "hostid": "30001"
            }
        ]
```

```
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
            "5",
            "6"
        ]
    },
    "id": 1
}
```

Veja também

- Host

Fonte

CHostGroup::massAdd() in *ui/include/classes/api/services/CHostGroup.php*.

hostgroup.massremove

Descrição

`object hostgroup.massremove(object parameters)`

Este método permite remover objetos relacionados de vários grupos de hosts.

Note:
Este método está disponível apenas para os tipos de usuário *Admin* e *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função de usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object)` Parâmetros contendo os IDs dos grupos de hosts a serem atualizados e os objetos que devem ser removidos.

Parâmetro	Tipo	Descrição
groupids	ID/array	IDs dos grupos de hosts a serem atualizados. Comportamento do parâmetro: - *obrigatório*
hostids	ID/array	IDs dos hosts a serem removidos de todos os grupos de hosts.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs do host atualizado grupos sob a propriedade `groupids`.

Exemplos

Removendo hosts de grupos de hosts

Remova dois hosts dos grupos de hosts fornecidos.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "hostgroup.massremove",
    "params": {
        "groupids": [
            "5",
            "6"
```

```
        ],
        "hostids": [
            "30050",
            "30001"
        ]
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
            "5",
            "6"
        ]
    },
    "id": 1
}
```

Fonte

CHostGroup::massRemove() in *ui/include/classes/api/services/CHostGroup.php*.

hostgroup.massupdate

Descrição

`object hostgroup.massupdate(object parameters)`

Este método permite substituir hosts e templates pelos especificados em vários grupos de hosts.

Note:
Este método está disponível apenas para os tipos de usuário *Admin* e *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object)` Parâmetros contendo os IDs dos grupos de hosts a serem atualizados e os objetos que devem ser atualizados.

Parâmetro	Tipo	Descrição
groups	object/array	Grupos de hosts a serem atualizados. Os grupos de hosts devem ter apenas a propriedade `groupid` definida. Comportamento do parâmetro: - *obrigatório*
hosts	object/array	Hosts para substituir os hosts atuais nos grupos de hosts fornecidos. Todos os outros hosts, exceto os mencionados, serão excluídos dos grupos de hosts. Hosts descobertos não serão afetados. Os hosts devem ter apenas a propriedade `hostid` definida. Comportamento do parâmetro: - *obrigatório*

Valores de retorno

`(object)` Retorna um objeto contendo os IDs do host atualizado grupos sob a propriedade `groupids`.

Exemplos

Substituindo hosts em um grupo de hosts

Substitua todos os hosts em um grupo de hosts pelos hosts mencionados.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "hostgroup.massupdate",
    "params": {
        "groups": [
            {
                "groupid": "6"
            }
        ],
        "hosts": [
            {
                "hostid": "30050"
            }
        ]
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
            "6",
        ]
    },
    "id": 1
}
```

Veja também

- hostgroup.update
- hostgroup.massadd
- Host

Fonte

CHostGroup::massUpdate() em *ui/include/classes/api/services/CHostGroup.php*.

hostgroup.propagate

Descrição

`object hostgroup.propagate(object parameters)`

Este método permite aplicar permissões e filtros de tags a todos os subgrupos de um grupo de hosts.

> **Note:**
> Este método está disponível apenas para tipos de usuário *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função de usuário. Consulte Funções de usuário para mais informações.

Parâmetros

`(object)` Parâmetros que definem a saída desejada.

O método suporta os seguintes parâmetros.

Parâmetro	Tipo	Descrição
groups	object/array	Grupos de hosts a serem propagados. Os grupos de hosts devem ter a propriedade `groupid` definida. Comportamento do parâmetro: - *obrigatório*
permissions	boolean	Defina como ”true” para propagar permissões. Comportamento do parâmetro: - *obrigatório* se `tag_filters` não estiver definido
tag_filters	boolean	Defina como ”true” para propagar filtros de tags. Comportamento do parâmetro: - *obrigatório* se `permissions` não estiver definido

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos grupos de hosts propagados na propriedade `groupids`.

Exemplos

Propagando permissões de grupos de hosts e filtros de tags para seus subgrupos.

Propaga permissões de grupos de hosts e filtros de tags para seus subgrupos.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "hostgroup.propagate",
    "params": {
        "groups": [
            {
                "groupid": "6"
            }
        ],
        "permissions": true,
        "tag_filters": true
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
            "6",
        ]
    },
    "id": 1
}
```

Veja também

- hostgroup.update
- hostgroup.massadd
- Host

Fonte

CHostGroup::propagate() em *ui/include/classes/api/services/CHostGroup.php*.

hostgroup.update

Descrição

`object hostgroup.update(object/array hostGroups)`

Este método permite atualizar grupos de hosts existentes.

> **Note:**
> Este método está disponível apenas para os tipos de usuário *Admin* e *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Consulte Funções de usuário para mais informações.

Parâmetros

`(object/array)` Propriedades do grupo de hosts a serem atualizadas.

A propriedade `groupid` deve ser definida para cada grupo de hosts, todas as outras propriedades são opcionais. Somente as propriedades fornecidas serão atualizadas, todas as outras permanecerão inalteradas.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs do host atualizado grupos sob a propriedade `groupids`.

Exemplos

Renomeando um grupo de hosts

Renomeie um grupo de hosts para ”Linux hosts”.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "hostgroup.update",
    "params": {
        "groupid": "7",
        "name": "Linux hosts"
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
            "7"
        ]
    },
    "id": 1
}
```

Fonte

CHostGroup::update() em *ui/include/classes/api/services/CHostGroup.php*.

Grupo de proxy

Esta classe foi projetada para trabalhar com grupos de proxy.

Referências de objetos:

- Grupo de proxy

Métodos disponíveis:

- proxygroup.create - criar novos grupos de proxy
- proxygroup.delete - excluir grupos de proxy
- proxygroup.get - recuperar grupos de proxy
- proxygroup.update - atualizar grupos de proxy

Objeto de grupo de proxy

Os seguintes objetos estão diretamente relacionados à API `proxygroup`.

Grupo de proxy

O objeto grupo de proxy possui as seguintes propriedades.

Propriedade	Tipo	Descrição
proxy_groupid	ID	ID do grupo de proxy. Comportamento da propriedade: - *somente leitura* - *obrigatório* para operações de atualização
name	string	Nome do grupo de proxy. Comportamento da propriedade: - *obrigatório* para operações de criação
description	text	Descrição do grupo de proxy.
failover_delay	string	Período durante o qual um proxy no grupo de proxy deve se comunicar com o Zabbix server para ser considerado online. Sufixos de tempo são suportados, por exemplo, 30s, 1m. Macros de usuário são suportadas. Valores possíveis: 10s-15m. Padrão: 1m.
min_online	string	Número mínimo de proxies online necessários para manter o grupo de proxy online. Macros de usuário são suportadas. Faixa de valores possíveis: 1-1000. Padrão: 1.
state	integer	Estado do grupo de proxy. Valores possíveis: 0 - Desconhecido; 1 - Offline; 2 - Recuperando; 3 - Online; 4 - Degradando. Comportamento da propriedade: - *somente leitura*

proxygroup.create

Descrição

`object proxygroup.create(object/array proxyGroups)`

Este método permite criar novos grupos de proxy.

Note:
Este método está disponível apenas para o tipo de usuário *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função de usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object/array)` Grupos de proxy a serem criados.

O método aceita grupos de proxy com as propriedades padrão de grupo de proxy.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos grupos de proxy criados na propriedade `proxy_groupids`. A ordem dos IDs retornados corresponde à ordem dos grupos de proxy passados.

Exemplos

Criar um grupo de proxy

Crie um grupo de proxy com configurações personalizadas.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "proxygroup.create",
    "params": {
        "name": "Proxy group",
        "failover_delay": "5m",
        "min_online": "10"
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "proxy_groupids": [
            "5"
        ]
    },
    "id": 1
}
```

Fonte

CProxyGroup::create() em *ui/include/classes/api/services/CProxyGroup.php*.

proxygroup.delete

Descrição

`object proxygroup.delete(array proxyGroupIds)`

Este método permite excluir grupos de proxy.

> **Note:**
> Este método está disponível apenas para o tipo de usuário *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(array)` IDs dos grupos de proxy a serem excluídos.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos grupos de proxy excluídos na propriedade `proxy_groupids`.

Exemplos

Excluir vários grupos de proxy

Exclua dois grupos de proxy.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "proxygroup.delete",
    "params": [
        "5",
        "10"
    ],
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "proxy_groupids": [
            "5",
            "10"
        ]
    },
    "id": 1
}
```

Fonte

CProxyGroup::delete() em *ui/include/classes/api/services/CProxyGroup.php*.

proxygroup.get

Descrição

`integer/array proxygroup.get(object parameters)`

O método permite recuperar grupos de proxy de acordo com os parâmetros fornecidos.

Note:
Este método está disponível para usuários de qualquer tipo. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object)` Parâmetros que definem a saída desejada.

O método suporta os seguintes parâmetros.

Parâmetro	Tipo	Descrição
proxy_groupids	ID/array	Retorna apenas grupos de proxy com os IDs fornecidos.
proxyids	ID/array	Retorna apenas grupos de proxy que contenham os proxies fornecidos.
selectProxies	query	Retorna uma propriedade `proxies` com os proxies que pertencem ao grupo de proxy. Suporta `count`.
sortfield	string/array	Ordena o resultado pelas propriedades fornecidas. Valores possíveis: `proxy_groupid`, `name`.
countOutput	boolean	Esses parâmetros são descritos no comentário de referência.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	

Parâmetro	Tipo	Descrição
sortorder	string/array	
startSearch	boolean	

Valores de retorno

`(integer/array)` Retorna:

- um array de objetos;
- a contagem de objetos recuperados, se o parâmetro `countOutput` tiver sido usado.

Exemplos

Recuperar todos os grupos de proxy

Recupere todos os grupos de proxy configurados com proxies.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "proxygroup.get",
    "params": {
        "output": "extend",
        "selectProxies": ["proxyid", "name"]
    },
    "id": 1
}
```

Resposta:

```
{
    "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
}
```

Veja também

- Proxy

Fonte

CProxyGroup::get() em *ui/include/classes/api/services/CProxyGroup.php*.

proxygroup.update

Descrição

`object proxygroup.update(object/array proxyGroups)`

Este método permite atualizar grupos de proxy existentes.

> **Note:**
> Este método está disponível apenas para o tipo de usuário *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função de usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object/array)` Propriedades do grupo de proxy a serem atualizadas.

A propriedade `proxy_groupid` deve ser definida para cada grupo de proxy, todas as outras propriedades são opcionais. Apenas as propriedades passadas serão atualizadas, todas as outras permanecerão inalteradas.

O método aceita grupos de proxy com as propriedades padrão do grupo de proxy.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos grupos de proxy atualizados na propriedade `proxy_groupids`.

Exemplos

Alterar o número mínimo de proxies online

Altere o número mínimo de proxies online necessários para que o grupo esteja online.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "proxygroup.update",
    "params": {
        "proxy_groupid": "5",
        "min_online": "3"
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "proxy_groupids": [
            "5"
```

```
        ]
    },
    "id": 1
}
```

Fonte

CProxyGroup::update() em *ui/include/classes/api/services/CProxyGroup.php*.

Grupo de template

Esta classe foi projetada para trabalhar com grupos de template.

Referências de objetos:

- Grupo de template

Métodos disponíveis:

- templategroup.create - criar novos grupos de template
- templategroup.delete - excluir grupos de template
- templategroup.get - recuperar grupos de template
- templategroup.massadd - adicionar objetos relacionados aos grupos de template
- templategroup.massremove - remover objetos relacionados dos grupos de template
- templategroup.massupdate - substituir ou remover objetos relacionados dos grupos de template
- templategroup.propagate - propagar permissões para subgrupos dos grupos de template
- templategroup.update - atualizar grupos de template

Objeto de grupo de template

Os seguintes objetos estão diretamente relacionados à API `templategroup`.

Grupo de template

O objeto de grupo de template possui as seguintes propriedades.

Propriedade	Tipo	Descrição
groupid	ID	ID do grupo de template. Comportamento da propriedade: - *somente leitura* - *obrigatório* para operações de atualização
name	string	Nome do grupo de template. Comportamento da propriedade: - *obrigatório* para operações de criação
uuid	string	Identificador universal único, usado para vincular grupos de template importados a já existentes. Gerado automaticamente, se não fornecido.

templategroup.create

Descrição

`object templategroup.create(object/array templateGroups)`

Este método permite criar novos grupos de templates.

> **Note:**
> Este método está disponível apenas para o tipo de usuário *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object/array)` Grupos de template a serem criados. O método aceita grupos de template com as propriedades padrão de grupo de template.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos grupos de template criados na propriedade `groupids`. A ordem dos IDs retornados corresponde à ordem dos grupos de template passados.

Exemplos

Criando um grupo de templates

Crie um grupo de templates chamado "Templates/Databases".

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "templategroup.create",
    "params": {
        "name": "Templates/Databases"
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
            "107820"
        ]
    },
    "id": 1
}
```

Fonte

CTemplateGroup::create() em *ui/include/classes/api/services/CTemplateGroup.php*.

templategroup.delete

Descrição

`object templategroup.delete(array templateGroupIds)`

Este método permite excluir grupos de templates.

Um grupo de templates não pode ser excluído se contiver templates que pertencem apenas a este grupo.

> **Note:**
> Este método está disponível apenas para os tipos de usuário *Admin* e *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(array)` IDs dos grupos de template a serem excluídos.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos grupos de template excluídos na propriedade `groupids`.

Exemplos

Excluindo vários grupos de template

Exclua dois grupos de template.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "templategroup.delete",
    "params": [
        "107814",
        "107815"
    ],
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
            "107814",
            "107815"
        ]
    },
    "id": 1
}
```

Fonte

CTemplateGroup::delete() em *ui/include/classes/api/services/CTemplateGroup.php*.

templategroup.get

Descrição

`integer/array templategroup.get(object parameters)`

O método permite recuperar grupos de templates de acordo com os parâmetros fornecidos.

> **Note:**
> Este método está disponível para usuários de qualquer tipo. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object)` Parâmetros que definem a saída desejada.

O método suporta os seguintes parâmetros.

Parâmetro	Tipo	Descrição
graphids	ID/array	Retorna apenas grupos de templates que contenham templates com os gráficos informados.
groupids	ID/array	Retorna apenas grupos de templates com os IDs de grupo de template informados.
templateids	ID/array	Retorna apenas grupos de templates que contenham os templates informados.
triggerids	ID/array	Retorna apenas grupos de templates que contenham templates com os triggers informados.
with_graphs	boolean	Retorna apenas grupos de templates que contenham templates com gráficos.
with_graph_prototypes	boolean	Retorna apenas grupos de templates que contenham templates com protótipos de gráficos.
with_httptests	boolean	Retorna apenas grupos de templates que contenham templates com verificações web.
with_items	boolean	Retorna apenas grupos de templates que contenham templates com items. Sobrescreve os parâmetros `with_simple_graph_items`.

Parâmetro	Tipo	Descrição
with_item_prototypes	boolean	Retorna apenas grupos de templates que contenham templates com protótipos de item. Sobrescreve o parâmetro `with_simple_graph_item_prototypes`.
with_simple_graph_item_prototypes	boolean	Retorna apenas grupos de templates que contenham templates com protótipos de item, que estejam habilitados para criação e tenham tipo de informação numérica.
with_simple_graph_items	boolean	Retorna apenas grupos de templates que contenham templates com items numéricos.
with_templates	boolean	Retorna apenas grupos de templates que contenham templates.
with_triggers	boolean	Retorna apenas grupos de templates que contenham templates com triggers.
selectTemplates	query	Retorna uma propriedade `templates` com os templates que pertencem ao grupo de template. Suporta `count`.
limitSelects	integer	Limita o número de registros retornados por subconsultas. Aplica-se às seguintes subconsultas: `selectTemplates` - os resultados serão classificados por `template`.
sortfield	string/array	Classifica o resultado pelas propriedades informadas. Valores possíveis: `groupid`, `name`.
countOutput	boolean	Esses parâmetros são descritos no comentário de referência.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valores de retorno

`(integer/array)` Retorna:

- um array de objetos;
- a contagem de objetos recuperados, se o parâmetro `countOutput` foi utilizado.

Exemplos

Recuperando dados pelo nome

Recupere todos os dados sobre dois grupos de template chamados ”Templates/Databases” e ”Templates/Modules”.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "templategroup.get",
    "params": {
        "output": "extend",
        "filter": {
            "name": [
                "Templates/Databases",
                "Templates/Modules"
            ]
        }
    },
```

```
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "groupid": "13",
            "name": "Templates/Databases",
            "uuid": "748ad4d098d447d492bb935c907f652f"
        },
        {
            "groupid": "8",
            "name": "Templates/Modules",
            "uuid": "57b7ae836ca64446ba2c296389c009b7"
        }
    ],
    "id": 1
}
```

Veja também

- Template

Fonte

CTemplateGroup::get() em *ui/include/classes/api/services/CTemplateGroup.php*.

templategroup.massadd

Descrição

`object templategroup.massadd(object parameters)`

Este método permite adicionar simultaneamente vários objetos relacionados a todos os grupos de templates fornecidos.

Note:
Este método está disponível apenas para os tipos de usuário *Admin* e *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object)` Parâmetros contendo os IDs dos grupos de templates a serem atualizados e os objetos a serem adicionados a todos os grupos de templates.

O método aceita os seguintes parâmetros.

Parâmetro	Tipo	Descrição
groups	object/array	Grupos de templates a serem atualizados. Os grupos de templates devem ter apenas a propriedade `groupid` definida. Comportamento do parâmetro: - *obrigatório*
templates	object/array	Templates a serem adicionados a todos os grupos de templates. Os templates devem ter apenas a propriedade `templateid` definida. Comportamento do parâmetro: - *obrigatório*

Valores de retorno

(`object`) Retorna um objeto contendo os IDs dos grupos de template atualizados na propriedade `groupids`.

Exemplos

Adicionando templates a grupos de templates

Adicione dois templates a grupos de templates com IDs 12 e 13.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "templategroup.massadd",
    "params": {
        "groups": [
            {
                "groupid": "12"
            },
            {
                "groupid": "13"
            }
        ],
        "templates": [
            {
                "templateid": "10486"
            },
            {
                "templateid": "10487"
            }
        ]
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
            "12",
            "13"
        ]
    },
    "id": 1
}
```

Veja também

- Template

Fonte

CTemplateGroup::massAdd() em *ui/include/classes/api/services/CTemplateGroup.php*.

templategroup.massremove

Descrição

`object templategroup.massremove(object parameters)`

Este método permite remover objetos relacionados de vários grupos de templates.

> **Note:**
> Este método está disponível apenas para os tipos de usuário *Admin* e *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object)` Parâmetros contendo os IDs dos grupos de template a serem atualizados e os objetos que devem ser removidos.

Parâmetro	Tipo	Descrição
groupids	ID/array	IDs dos grupos de template a serem atualizados. Comportamento do parâmetro: - *obrigatório*
templateids	ID/array	IDs dos templates a serem removidos de todos os grupos de template. Comportamento do parâmetro: - *obrigatório*

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos grupos de template atualizados na propriedade `groupids`.

Exemplos

Removendo templates de grupos de templates

Remove dois templates dos grupos de templates fornecidos.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "templategroup.massremove",
    "params": {
        "groupids": [
            "5",
            "6"
        ],
        "templateids": [
            "30050",
            "30001"
        ]
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
            "5",
            "6"
        ]
    },
    "id": 1
}
```

Fonte

CTemplateGroup::massRemove() em *ui/include/classes/api/services/CTemplateGroup.php*.

templategroup.massupdate

Descrição

`object templategroup.massupdate(object parameters)`

Este método permite substituir templates pelos especificados em vários grupos de templates.

Note:
Este método está disponível apenas para os tipos de usuário *Admin* e *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object)` Parâmetros contendo os IDs dos grupos de templates a serem atualizados e os objetos que devem ser atualizados.

Parâmetro	Tipo	Descrição
groups	object/array	Grupos de templates a serem atualizados. Os grupos de templates devem ter apenas a propriedade `groupid` definida. Comportamento do parâmetro: - *obrigatório*
templates	object/array	Templates para substituir o template atual nos grupos de templates fornecidos. Todos os outros templates, exceto os mencionados, serão excluídos dos grupos de templates. Os templates devem ter apenas a propriedade `templateid` definida. Comportamento do parâmetro: - *obrigatório*

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos grupos de template atualizados na propriedade `groupids`.

Exemplos

Substituindo templates em um grupo de templates

Substitua todos os templates em um grupo de templates pelos templates mencionados.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "templategroup.massupdate",
    "params": {
        "groups": [
            {
                "groupid": "8"
            }
        ],
        "templates": [
            {
                "templateid": "40050"
            }
        ]
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
            "8",
        ]
    },
```

```
    "id": 1
}
```

Veja também

- templategroup.update
- templategroup.massadd
- Template

Fonte

CTemplateGroup::massUpdate() em *ui/include/classes/api/services/CTemplateGroup.php*.

templategroup.propagate

Descrição

`object templategroup.propagate(object parameters)`

Este método permite aplicar permissões a todos os subgrupos de grupos de template.

> **Note:**
> Este método está disponível apenas para tipos de usuário *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Consulte Funções de usuário para mais informações.

Parâmetros

`(object)` Parâmetros que definem a saída desejada.

O método suporta os seguintes parâmetros.

Parâmetro	Tipo	Descrição
groups	object/array	Grupos de template a serem propagados. Os grupos de template devem ter apenas a propriedade `groupid` definida. Comportamento do parâmetro: - *obrigatório*
permissions	boolean	Defina como `true` se precisar propagar permissões. Comportamento do parâmetro: - *obrigatório*

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos grupos de template propagados na propriedade `groupids`.

Exemplos

Propagando permissões de grupo de template para seus subgrupos.

Propaga permissões de grupo de template para seus subgrupos.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "templategroup.propagate",
    "params": {
        "groups": [
            {
                "groupid": "15"
            }
        ],
        "permissions": true
    },
```

```
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
            "15",
        ]
    },
    "id": 1
}
```

Veja também

- templategroup.update
- templategroup.massadd
- Template

Fonte

CTemplateGroup::propagate() em *ui/include/classes/api/services/CTemplateGroup.php*.

templategroup.update

Descrição

`object templategroup.update(object/array templateGroups)`

Este método permite atualizar grupos de templates existentes.

> **Note:**
> Este método está disponível apenas para os tipos de usuário *Admin* e *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Consulte Funções de usuário para mais informações.

Parâmetros

`(object/array)` Propriedades do grupo de template a serem atualizadas.

A propriedade `groupid` deve ser definida para cada grupo de template, todas as outras propriedades são opcionais. Somente as propriedades fornecidas serão atualizadas, todas as outras permanecerão inalteradas.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos grupos de template atualizados na propriedade `groupids`.

Exemplos

Renomeando um grupo de templates

Renomeie um grupo de templates para "Templates/Databases"

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "templategroup.update",
    "params": {
        "groupid": "7",
        "name": "Templates/Databases"
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
```

```
        "groupids": [
            "7"
        ]
    },
    "id": 1
}
```

Fonte

CTemplateGroup::update() em *ui/include/classes/api/services/CTemplateGroup.php*.

Grupo de usuários

Esta classe foi projetada para trabalhar com grupos de usuários.

Referências de objetos:

- Grupo de usuários
- Permissão
- Permissão baseada em tag

Métodos disponíveis:

- usergroup.create - criar novos grupos de usuários
- usergroup.delete - excluir grupos de usuários
- usergroup.get - recuperar grupos de usuários
- usergroup.update - atualizar grupos de usuários

Objeto de grupo de usuários

Os seguintes objetos estão diretamente relacionados à API `usergroup`.

Grupo de usuários

O objeto grupo de usuários possui as seguintes propriedades.

Propriedade	Tipo	Descrição
usrgrpid	ID	ID do grupo de usuários. Comportamento da propriedade: - *somente leitura* - *obrigatório* para operações de atualização
name	string	Nome do grupo de usuários. Comportamento da propriedade: - *obrigatório* para operações de criação
debug_mode	integer	Se o modo de depuração está habilitado ou desabilitado. Valores possíveis: 0 - *(padrão)* desabilitado; 1 - habilitado.
gui_access	integer	Método de autenticação no frontend dos usuários do grupo. Valores possíveis: 0 - *(padrão)* usar o método de autenticação padrão do sistema; 1 - usar autenticação interna; 2 - usar autenticação LDAP; 3 - desabilitar o acesso ao frontend.
mfa_status	integer	Se a MFA está habilitada ou desabilitada para os usuários do grupo. Valores possíveis: 0 - desabilitada (para todos os métodos de MFA configurados); 1 - habilitada (para todos os métodos de MFA configurados).

Propriedade	Tipo	Descrição
mfaid	ID	Método de MFA usado para os usuários do grupo. Comportamento da propriedade: - *suportado* para usuários do tipo *Super admin* (desde o Zabbix 7.0.9) - *suportado* se `mfa_status` do objeto Authentication estiver definido como ”Habilitado”
users_status	integer	Se o grupo de usuários está habilitado ou desabilitado. Para usuários desprovisionados, o grupo de usuários não pode ser habilitado. Valores possíveis: 0 - *(padrão)* habilitado; 1 - desabilitado.
userdirectoryid	ID	ID do diretório de usuários usado para autenticação. Comportamento da propriedade: - *suportado* para usuários do tipo *Super admin* (desde o Zabbix 7.0.9) - *suportado* se `gui_access` estiver definido como ”usar o método de autenticação padrão do sistema” ou ”usar autenticação LDAP”

Permissão

O objeto de permissão possui as seguintes propriedades.

Propriedade	Tipo	Descrição
id	ID	ID do grupo de hosts ou grupo de templates para adicionar permissão. Comportamento da propriedade: - *obrigatório*
permission	integer	Nível de acesso ao grupo de hosts ou grupo de templates. Valores possíveis: 0 - acesso negado; 2 - acesso somente leitura; 3 - acesso leitura e escrita. Comportamento da propriedade: - *obrigatório* para operações de criação

Permissão baseada em tag

O objeto de permissão baseada em tag possui as seguintes propriedades.

Propriedade	Tipo	Descrição
groupid	ID	ID do grupo de hosts ao qual adicionar permissão. Comportamento da propriedade: - *obrigatório*
tag	string	Nome da tag.
value	string	Valor da tag.

usergroup.create

Descrição

`object usergroup.create(object/array userGroups)`

Este método permite criar novos grupos de usuários.

Note:
Este método está disponível apenas para o tipo de usuário *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Consulte Funções de usuário para mais informações.

Parâmetros

`(object/array)` Grupos de usuários a serem criados.

Além das propriedades padrão do grupo de usuários, o método aceita os seguintes parâmetros.

Parâmetro	Tipo	Descrição
hostgroup_rights	object/array	Permissões de grupo de host a serem atribuídas ao grupo de usuários.
templategroup_rights	object/array	Permissões de grupo de template a serem atribuídas ao grupo de usuários.
tag_filters	array	Permissões baseadas em tags a serem atribuídas ao grupo de usuários.
users	object/array	Usuários a serem adicionados ao grupo de usuários. O usuário deve ter apenas a propriedade `userid` definida.
rights (obsoleto)	object/array	Este parâmetro está obsoleto, por favor, use `hostgroup_rights` ou `templategroup_rights`. Permissões a serem atribuídas ao grupo de usuários.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos grupos de usuários criados na propriedade `usrgrpids`. A ordem dos IDs retornados corresponde à ordem dos grupos de usuários passados.

Exemplos

Criando um grupo de usuários

Crie um grupo de usuários *Operation managers* com acesso negado ao grupo de hosts "2" e adicione um usuário a ele.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "usergroup.create",
    "params": {
        "name": "Operation managers",
        "hostgroup_rights": {
            "id": "2",
            "permission": 0
        },
        "users": [
            {
                "userid": "12"
            }
        ]
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "usrgrpids": [
            "20"
        ]
    },
    "id": 1
}
```

Veja também

- Permissão

Fonte

CUserGroup::create() em *ui/include/classes/api/services/CUserGroup.php*.

usergroup.delete

Descrição

`object usergroup.delete(array userGroupIds)`

Este método permite excluir grupos de usuários.

Attention:
O grupo de usuários Desprovisioned (o grupo de usuários especificado para `disabled_usrgrpid` em Authentication) não pode ser excluído.

Note:
Este método está disponível apenas para o tipo de usuário *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(array)` IDs dos grupos de usuários a serem excluídos.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos grupos de usuários excluídos sob a propriedade `usrgrpids`.

Exemplos

Excluindo vários grupos de usuários

Exclua dois grupos de usuários.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "usergroup.delete",
    "params": [
        "20",
        "21"
    ],
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "usrgrpids": [
            "20",
            "21"
        ]
    },
    "id": 1
}
```

Fonte

CUserGroup::delete() in *ui/include/classes/api/services/CUserGroup.php*.

usergroup.get

Descrição

`integer/array usergroup.get(object parameters)`

O método permite recuperar grupos de usuários de acordo com os parâmetros fornecidos.

> **Note:**
> Este método está disponível para usuários de qualquer tipo. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object)` Parâmetros que definem a saída desejada.

O método suporta os seguintes parâmetros.

Parâmetro	Tipo	Descrição
mfaids	ID/array	Retorna apenas grupos de usuários com os métodos MFA fornecidos. Comportamento do parâmetro: - *suportado* para usuários do tipo *Super admin* (desde o Zabbix 7.0.9)
mfa_status	integer	Retorna apenas grupos de usuários com o status MFA fornecido. Consulte a página do grupo de usuários para obter uma lista de status suportados.
status	integer	Retorna apenas grupos de usuários com o status fornecido. Consulte a página do grupo de usuários para obter uma lista de status suportados.
userids	ID/array	Retorna apenas grupos de usuários que contenham os usuários fornecidos.
usrgrpids	ID/array	Retorna apenas grupos de usuários com os IDs fornecidos.
selectTagFilters	query	Retorna permissões baseadas em tags do grupo de usuários na propriedade `tag_filters`.
selectUsers	query	Retorna os usuários do grupo de usuários na propriedade `users`. Veja `user.get` para restrições baseadas no tipo de usuário.
selectHostGroupRights	query	Retorna as permissões do grupo de usuários para grupos de hosts na propriedade `hostgroup_rights`. Consulte a página do grupo de usuários para obter uma lista de níveis de acesso aos grupos de hosts.
selectTemplateGroupRights	query	Retorna as permissões do grupo de usuários para grupos de templates na propriedade `templategroup_rights`. Consulte a página do grupo de usuários para obter uma lista de níveis de acesso aos grupos de templates.
limitSelects	integer	Limita o número de registros retornados por subseleções.
output	query	Propriedades do objeto grupo de usuários a serem retornadas. Desde o Zabbix 7.0.9, usuários do tipo *Admin* e *User* podem recuperar apenas as seguintes propriedades: `usrgrpid`, `name`, `gui_access`, `users_status`, `debug_mode`, `mfa_status`. Padrão: `extend`.
sortfield	string/array	Ordena o resultado pelas propriedades fornecidas. Valores possíveis: `usrgrpid`, `name`.
countOutput	boolean	Esses parâmetros são descritos no comentário de referência.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
preservekeys	boolean	
search	object	
searchByAny	boolean	

Parâmetro	Tipo	Descrição
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	
selectRights (obsoleto)	query	Este parâmetro está obsoleto, use `selectHostGroupRights` ou `selectTemplateGroupRights` em seu lugar. Retorna os direitos do grupo de usuários na propriedade `rights`. Consulte a página do grupo de usuários para obter uma lista de níveis de acesso aos grupos de hosts.

Valores de retorno

`(integer/array)` Retorna:

- uma array de objetos;
- a contagem de objetos recuperados, se o parâmetro `countOutput` foi usado.

Exemplos

Recuperando grupos de usuários habilitados

Recupere todos os grupos de usuários habilitados.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "usergroup.get",
    "params": {
        "output": "extend",
        "status": 0
    },
    "id": 1
}
```

Resposta:

```
{
    "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
}
```


Veja também

- Usuário

Fonte

CUserGroup::get() in *ui/include/classes/api/services/CUserGroup.php*.

usergroup.update

Descrição

`object usergroup.update(object/array userGroups)`

Este método permite atualizar grupos de usuários existentes.

> **Note:**
> Este método está disponível apenas para o tipo de usuário *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função de usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object/array)` Propriedades do grupo de usuários a serem atualizadas.

A propriedade `usrgrpid` deve ser definida para cada grupo de usuários, todas as outras propriedades são opcionais. Somente as propriedades passadas serão atualizadas, todas as outras permanecerão inalteradas.

Além das propriedades padrão do grupo de usuários, o método aceita os seguintes parâmetros.

Parâmetro	Tipo	Descrição
hostgroup_rights	object/array	Permissões do grupo de hosts para substituir as permissões atuais atribuídas ao grupo de usuários.
templategroup_rights	object/array	Permissões do grupo de templates para substituir as permissões atuais atribuídas ao grupo de usuários.
tag_filters	array	Permissões baseadas em tags para substituir as permissões atuais atribuídas ao grupo de usuários.
users	object/array	Usuários para substituir os usuários atuais atribuídos ao grupo de usuários. O usuário deve ter apenas a propriedade `userid` definida.
rights (obsoleto)	object/array	Este parâmetro está obsoleto, por favor, use `hostgroup_rights` ou `templategroup_rights`. Permissões a serem atribuídas ao grupo de usuários.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos grupos de usuários atualizados sob a propriedade `usrgrpids`.

Exemplos

Habilitando um grupo de usuários e atualizando permissões

Habilite um grupo de usuários e forneça acesso de leitura e gravação para ele aos grupos de hosts ”2” e ”4”.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "usergroup.update",
    "params": {
        "usrgrpid": "17",
        "users_status": "0",
        "hostgroup_rights": [
            {
                "id": "2",
                "permission": 3
            },
            {
                "id": "4",
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "usrgrpids": [
            "17"
        ]
    },
    "id": 1
}
```

Veja também

- Permissão

Fonte

CUserGroup::update() in *ui/include/classes/api/services/CUserGroup.php*.

History

Esta classe foi projetada para trabalhar com dados de histórico.

Referências de objetos:

- Histórico de ponto flutuante
- Histórico de inteiro
- Histórico de string
- Histórico de texto
- Histórico de log

Métodos disponíveis:

- history.clear - limpa os dados de histórico
- history.get - recupera os dados de histórico
- history.push - envia dados de histórico para o Zabbix server

Objeto de histórico

Os seguintes objetos estão diretamente relacionados à API `history`.

Note:
Os objetos de histórico diferem dependendo do tipo de informação do item. Eles são criados pelo Zabbix server e não podem ser modificados via API.

Histórico de ponto flutuante

O objeto de histórico de ponto flutuante possui as seguintes propriedades.

Propriedade	Tipo	Descrição
clock	timestamp	Hora em que o valor foi recebido.
itemid	ID	ID do item relacionado.
ns	integer	Nanosegundos em que o valor foi recebido.
value	float	Valor recebido.

Histórico de inteiros

O objeto de histórico de inteiros possui as seguintes propriedades.

Propriedade	Tipo	Descrição
clock	timestamp	Hora em que o valor foi recebido.
itemid	ID	ID do item relacionado.
ns	integer	Nanosegundos em que o valor foi recebido.
value	integer	Valor recebido.

Histórico de string

O objeto de histórico de string possui as seguintes propriedades.

Propriedade	Tipo	Descrição
clock	timestamp	Hora em que o valor foi recebido.
itemid	ID	ID do item relacionado.
ns	integer	Nanosegundos em que o valor foi recebido.
value	string	Valor recebido.

Histórico de texto

O objeto de histórico de texto possui as seguintes propriedades.

Propriedade	Tipo	Descrição
id	ID	ID da entrada de histórico.
clock	timestamp	Hora em que o valor foi recebido.
itemid	ID	ID do item relacionado.
ns	integer	Nanosegundos em que o valor foi recebido.
value	text	Valor recebido.

Histórico de log

O objeto de histórico de log possui as seguintes propriedades.

Propriedade	Tipo	Descrição
id	ID	ID da entrada do histórico.
clock	timestamp	Hora em que o valor foi recebido.
itemid	ID	ID do item relacionado.
logeventid	integer	ID da entrada do log de eventos do Windows.
ns	integer	Nanosegundos em que o valor foi recebido.
severity	integer	Nível da entrada do log de eventos do Windows.
source	string	Fonte da entrada do log de eventos do Windows.
timestamp	timestamp	Hora da entrada do log de eventos do Windows.
value	text	Valor recebido.

Histórico binário

O objeto de histórico binário possui as seguintes propriedades.

Propriedade	Tipo	Descrição
clock	timestamp	Hora em que o valor foi recebido.
itemid	ID	ID do item relacionado.
ns	integer	Nanosegundos em que o valor foi recebido.
value	text	Valor recebido.

history.get

Descrição

`integer/array history.get(object parameters)`

O método permite recuperar dados históricos de acordo com os parâmetros fornecidos.

Attention:
Este método pode retornar dados históricos de uma entidade excluída se esses dados ainda não tiverem sido removidos pelo housekeeper.

Note:
Este método está disponível para usuários de qualquer tipo. As permissões para chamar o método podem ser revogadas nas configurações de função de usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object)` Parâmetros que definem a saída desejada.

O método suporta os seguintes parâmetros.

Parâmetro	Tipo	Descrição
history	integer	Tipos de objetos de histórico a serem retornados. Valores possíveis: 0 - numérico float; 1 - caractere; 2 - log; 3 - *(padrão)* numérico unsigned; 4 - texto; 5 - binário.
hostids	ID/array	Retornar apenas histórico dos hosts fornecidos.
itemids	ID/array	Retornar apenas histórico dos items fornecidos.
time_from	timestamp	Retornar apenas valores que foram recebidos após ou no horário fornecido.
time_till	timestamp	Retornar apenas valores que foram recebidos antes do horário fornecido.
sortfield	string/array	Ordenar o resultado pelas propriedades fornecidas. Valores possíveis: `itemid`, `clock`, `ns`.
countOutput	boolean	Esses parâmetros são descritos no comentário de referência.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valores de retorno

`(integer/array)` Retorna:

- uma matriz de objetos;
- a contagem de objetos recuperados, se o parâmetro `countOutput` tiver sido usado.

Exemplos

Recuperando dados históricos de item

Retorna os 10 últimos valores recebidos de um item numérico (float).

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "history.get",
    "params": {
        "output": "extend",
        "history": 0,
        "itemids": "23296",
        "sortfield": "clock",
        "sortorder": "DESC",
        "limit": 10
    },
    "id": 1
}
```

Resposta:

```
{
    "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
}
```

Fonte

CHistory::get() em *ui/include/classes/api/services/CHistory.php*.

history.push

Descrição

`object history.push(object/array itemHistoryData)`

Este método permite enviar dados do histórico de item para o Zabbix server.

> **Note:**
> Este método está disponível para usuários de qualquer tipo. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Consulte Funções de usuário para mais informações.

Parâmetros

`(object/array)` Dados do histórico do item a serem enviados.

O método suporta os seguintes parâmetros.

Parâmetro	Tipo	Descrição
itemid	ID	ID do item relacionado. Comportamento do parâmetro: - *obrigatório* se `host` e `key` não estiverem definidos
host	string	Nome técnico do host. Comportamento do parâmetro: - *obrigatório* se `itemid` não estiver definido
key	string	Chave do item. Comportamento do parâmetro: - *obrigatório* se `itemid` não estiver definido
value	mixed	Valor do item. Comportamento do parâmetro: - *obrigatório*
clock	timestamp	Hora em que o valor foi recebido.
ns	integer	Nanosegundos em que o valor foi recebido.

Valores de retorno

`(object)` Retorna o resultado da operação de envio de dados.

Exemplos

Enviar dados de histórico de item

Enviar dados de histórico de item para os items ”10600”, ”10601” e ”999999”.

Requisição:

```
{
    "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
}
```


Resposta:


```
{
    "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
}
```


Veja também

- Itens trapper
- Itens HTTP agent
- Host
- Item

Origem

CHistory::push() em *ui/include/classes/api/services/CHistory.php*.

histórico.limpar

Descrição

`object history.clear(array itemids)`

Este método permite limpar o histórico do item.

> **Note:**
> Este método está disponível apenas para os tipos de usuário *Admin* e *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Consulte Funções de usuário para mais informações.

Parâmetros

(`array`) IDs de itens a serem limpos.

Valores de retorno

(`object`) Retorna um objeto contendo os IDs dos itens compensados sob a propriedade `itemids`.

Exemplos

Limpar histórico

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "history.clear",
    "params": [
        "10325",
        "13205"
    ],
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "10325",
            "13205"
        ]
    },
    "id": 1
}
```

Fonte

CHistory::clear() em *ui/include/classes/api/services/CHistory.php*.

Host

Esta classe foi projetada para trabalhar com hosts.

Referências de objetos:

- Host
- Inventário do host
- Tag do host

Métodos disponíveis:

- host.create - criar novos hosts
- host.delete - excluir hosts
- host.get - recuperar hosts
- host.massadd - adicionar objetos relacionados aos hosts
- host.massremove - remover objetos relacionados dos hosts
- host.massupdate - substituir ou remover objetos relacionados dos hosts
- host.update - atualizar hosts

Objeto host

Os seguintes objetos estão diretamente relacionados à API de host.

Host

O objeto host possui as seguintes propriedades.

Propriedade	Tipo	Descrição
hostid	ID	ID do host. Comportamento da propriedade: - *somente leitura* - *obrigatório* para operações de atualização
host	string	Nome técnico do host. Comportamento da propriedade: - *obrigatório* para operações de criação
description	text	Descrição do host.
flags	integer	Origem do host. Valores possíveis: 0 - um host comum; 4 - um host descoberto. Comportamento da propriedade: - *somente leitura*
inventory_mode	integer	Modo de preenchimento do inventário do host. Valores possíveis: -1 - *(padrão)* desabilitado; 0 - manual; 1 - automático.
ipmi_authtype	integer	Algoritmo de autenticação IPMI. Valores possíveis: -1 - *(padrão)* padrão; 0 - nenhum; 1 - MD2; 2 - MD5 4 - straight; 5 - OEM; 6 - RMCP+.
ipmi_password	string	Senha IPMI.
ipmi_privilege	integer	Nível de privilégio IPMI. Valores possíveis: 1 - callback; 2 - *(padrão)* usuário; 3 - operador; 4 - admin; 5 - OEM.
ipmi_username	string	Nome de usuário IPMI.
maintenance_from	timestamp	Hora de início da manutenção efetiva. Comportamento da propriedade: - *somente leitura*
maintenance_status	integer	Status de manutenção efetiva. Valores possíveis: 0 - *(padrão)* sem manutenção; 1 - manutenção em vigor. Comportamento da propriedade: - *somente leitura*

Propriedade	Tipo	Descrição
maintenance_type	integer	Tipo de manutenção efetiva. Valores possíveis: 0 - *(padrão)* manutenção com coleta de dados; 1 - manutenção sem coleta de dados. Comportamento da propriedade: - *somente leitura*
maintenanceid	ID	ID da manutenção que está atualmente em vigor no host. Comportamento da propriedade: - *somente leitura*
name	string	Nome visível do host. Padrão: valor da propriedade `host`.
monitored_by	integer	Origem utilizada para monitorar o host. Valores possíveis: 0 - *(padrão)* Zabbix server; 1 - Proxy; 2 - Grupo de proxy.
proxyid	ID	ID do proxy utilizado para monitorar o host. Comportamento da propriedade: - *obrigatório* se `monitored_by` estiver definido como "Proxy"
proxy_groupid	ID	ID do grupo de proxy utilizado para monitorar o host. Comportamento da propriedade: - *obrigatório* se `monitored_by` estiver definido como "Grupo de proxy"
status	integer	Status e função do host. Valores possíveis: 0 - *(padrão)* habilitado; 1 - desabilitado.
tls_connect	integer	Conexões para o host. Valores possíveis: 1 - *(padrão)* Sem criptografia; 2 - PSK; 4 - certificado.
tls_accept	integer	Conexões do host. Valores possíveis de bitmap: 1 - *(padrão)* Sem criptografia; 2 - PSK; 4 - certificado. Este é um campo de máscara de bits; qualquer soma dos valores possíveis de bitmap é aceitável (por exemplo, 6 para PSK e certificado).
tls_issuer	string	Emissor do certificado.
tls_subject	string	Assunto do certificado.
tls_psk_identity	string	Identidade PSK; deve ser pareada com apenas um PSK (entre autoregistro, hosts e proxies). Não inclua informações sensíveis na identidade PSK, pois ela é enviada sem criptografia pela rede para informar ao receptor qual PSK usar. Comportamento da propriedade: - *somente escrita* - *obrigatório* se `tls_connect` estiver definido como "PSK" ou `tls_accept` contiver o bit "PSK"

Propriedade	Tipo	Descrição
tls_psk	string	Chave pré-compartilhada (PSK); deve ter pelo menos 32 dígitos hexadecimais. Comportamento da propriedade: - *somente escrita* - *obrigatório* se `tls_connect` estiver definido como ”PSK” ou `tls_accept` contiver o bit ”PSK”
active_available	integer	Status de disponibilidade da interface ativa do host. Valores possíveis: 0 - status da interface é desconhecido; 1 - interface está disponível; 2 - interface não está disponível. Comportamento da propriedade: - *somente leitura*
assigned_proxyid	ID	ID do proxy atribuído pelo Zabbix server, se o host for monitorado por um grupo de proxy. Comportamento da propriedade: - *somente leitura*

Inventário de host

O objeto de inventário de host possui as seguintes propriedades.

Note:
Cada propriedade possui seu próprio número de ID exclusivo, que é usado para associar campos de inventário de host com items.

ID	Propriedade	Tipo	Descrição	Comprimento máximo
4	alias	string	Apelido.	128 caracteres
11	asset_tag	string	Tag de ativo.	64 caracteres
28	chassis	string	Chassi.	64 caracteres
23	contact	string	Pessoa de contato.	Depende do banco de dados utilizado: - 65535 caracteres para bancos de dados SQL - 2048 caracteres para bancos de dados Oracle
32	contract_number	string	Número do contrato.	64 caracteres
47	date_hw_decomm	string	Data de descomissionamento do hardware.	64 caracteres
46	date_hw_expiry	string	Data de expiração da manutenção do hardware.	64 caracteres
45	date_hw_install	string	Data de instalação do hardware.	64 caracteres
44	date_hw_purchase	string	Data de compra do hardware.	64 caracteres
34	deployment_status	string	Status de implantação.	64 caracteres
14	hardware	string	Hardware.	255 caracteres
15	hardware_full	string	Hardware detalhado.	Depende do banco de dados utilizado: - 65535 caracteres para bancos de dados SQL - 2048 caracteres para bancos de dados Oracle
39	host_netmask	string	Máscara de sub-rede do host.	39 caracteres
38	host_networks	string	Redes do host.	Depende do banco de dados utilizado: - 65535 caracteres para bancos de dados SQL - 2048 caracteres para bancos de dados Oracle
40	host_router	string	Roteador do host.	39 caracteres

ID	Propriedade	Tipo	Descrição	Comprimento máximo
30	hw_arch	string	Arquitetura do hardware.	32 caracteres
33	installer_name	string	Nome do instalador.	64 caracteres
24	location	string	Localização.	Depende do banco de dados utilizado: - 65535 caracteres para bancos de dados SQL - 2048 caracteres para bancos de dados Oracle
25	location_lat	string	Latitude da localização.	16 caracteres
26	location_lon	string	Longitude da localização.	16 caracteres
12	macaddress_a	string	Endereço MAC A.	64 caracteres
13	macaddress_b	string	Endereço MAC B.	64 caracteres
29	model	string	Modelo.	64 caracteres
3	name	string	Nome.	128 caracteres
27	notes	string	Notas.	Depende do banco de dados utilizado: - 65535 caracteres para bancos de dados SQL - 2048 caracteres para bancos de dados Oracle
41	oob_ip	string	Endereço IP OOB.	39 caracteres
42	oob_netmask	string	Máscara de sub-rede do host OOB.	39 caracteres
43	oob_router	string	Roteador OOB.	39 caracteres
5	os	string	Nome do sistema operacional.	128 caracteres
6	os_full	string	Nome detalhado do sistema operacional.	255 caracteres
7	os_short	string	Nome curto do sistema operacional.	128 caracteres
61	poc_1_cell	string	Número de celular do POC primário.	64 caracteres
58	poc_1_email	string	E-mail primário.	128 caracteres
57	poc_1_name	string	Nome do POC primário.	128 caracteres
63	poc_1_notes	string	Notas do POC primário.	Depende do banco de dados utilizado: - 65535 caracteres para bancos de dados SQL - 2048 caracteres para bancos de dados Oracle
59	poc_1_phone_a	string	Telefone A do POC primário.	64 caracteres
60	poc_1_phone_b	string	Telefone B do POC primário.	64 caracteres
62	poc_1_screen	string	Nome de tela do POC primário.	64 caracteres
68	poc_2_cell	string	Número de celular do POC secundário.	64 caracteres
65	poc_2_email	string	E-mail do POC secundário.	128 caracteres
64	poc_2_name	string	Nome do POC secundário.	128 caracteres
70	poc_2_notes	string	Notas do POC secundário.	Depende do banco de dados utilizado: - 65535 caracteres para bancos de dados SQL - 2048 caracteres para bancos de dados Oracle
66	poc_2_phone_a	string	Telefone A do POC secundário.	64 caracteres
67	poc_2_phone_b	string	Telefone B do POC secundário.	64 caracteres
69	poc_2_screen	string	Nome de tela do POC secundário.	64 caracteres
8	serialno_a	string	Número de série A.	64 caracteres
9	serialno_b	string	Número de série B.	64 caracteres
48	site_address_a	string	Endereço do site A.	128 caracteres
49	site_address_b	string	Endereço do site B.	128 caracteres
50	site_address_c	string	Endereço do site C.	128 caracteres
51	site_city	string	Cidade do site.	128 caracteres
53	site_country	string	País do site.	64 caracteres
56	site_notes	string	Notas do site.	Depende do banco de dados utilizado: - 65535 caracteres para bancos de dados SQL - 2048 caracteres para bancos de dados Oracle
55	site_rack	string	Localização do rack do site.	128 caracteres
52	site_state	string	Estado do site.	64 caracteres

ID	Propriedade	Tipo	Descrição	Comprimento máximo
54	site_zip	string	CEP/código postal do site.	64 caracteres
16	software	string	Software.	255 caracteres
18	software_app_a	string	Aplicativo de software A.	64 caracteres
19	software_app_b	string	Aplicativo de software B.	64 caracteres
20	software_app_c	string	Aplicativo de software C.	64 caracteres
21	software_app_d	string	Aplicativo de software D.	64 caracteres
22	software_app_e	string	Aplicativo de software E.	64 caracteres
17	software_full	string	Detalhes do software.	Depende do banco de dados utilizado: - 65535 caracteres para bancos de dados SQL - 2048 caracteres para bancos de dados Oracle
10	tag	string	Tag.	64 caracteres
1	type	string	Tipo.	64 caracteres
2	type_full	string	Detalhes do tipo.	64 caracteres
35	url_a	string	URL A.	2048 caracteres
36	url_b	string	URL B.	2048 caracteres
37	url_c	string	URL C.	2048 caracteres
31	vendor	string	Fornecedor.	64 caracteres

Tag de host

O objeto de tag de host possui as seguintes propriedades.

Propriedade	Tipo	Descrição
tag	string	Nome da tag do host. Comportamento da propriedade: - *obrigatório*
value	string	Valor da tag do host.
automatic	integer	Tipo de tag de host. Valores possíveis: 0 - *(padrão)* manual (tag criada pelo usuário); 1 - automática (tag criada por descoberta de baixo nível) Comportamento da propriedade: - *somente leitura*

host.create

Descrição

`object host.create(object/array hosts)`

Este método permite criar novos hosts.

Note:
Este método está disponível apenas para os tipos de usuário *Admin* e *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object/array)` Hosts a serem criados.

Além das propriedades padrão do host, o método aceita os seguintes parâmetros.

Parâmetro	Tipo	Descrição
groups	object/array	Grupos de hosts aos quais o host será adicionado. Os grupos de hosts devem ter apenas a propriedade `groupid` definida. Comportamento do parâmetro: - *obrigatório*
interfaces	object/array	Interfaces a serem criadas para o host.
tags	object/array	Tags do host.
templates	object/array	Templates a serem vinculados ao host. Os templates devem ter apenas a propriedade `templateid` definida.
macros	object/array	Macros de usuário a serem criadas para o host.
inventory	object	Propriedades do inventário do host.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos hosts criados sob a propriedade `hostids`. A ordem dos IDs retornados corresponde à ordem dos hosts passados.

Exemplos

Criando um host

Crie um host chamado "Linux server" com uma interface IP e tags, adicione-o a um grupo, vincule um template a ele e defina os endereços MAC no inventário do host.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "host.create",
    "params": {
        "host": "Linux server",
        "interfaces": [
            {
                "type": 1,
                "main": 1,
                "useip": 1,
                "ip": "192.168.3.1",
                "dns": "",
                "port": "10050"
            }
        ],
        "groups": [
            {
                "groupid": "50"
            }
        ],
        "tags": [
            {
                "tag": "host-name",
                "value": "linux-server"
            }
        ],
        "templates": [
            {
                "templateid": "20045"
            }
        ],
        "macros": [
            {
                "macro": "{$USER_ID}",
                "value": "123321"
            },
```

```
            {
                "macro": "{$USER_LOCATION}",
                "value": "0:0:0",
                "description": "latitude, longitude and altitude coordinates"
            }
        ],
        "inventory_mode": 0,
        "inventory": {
            "macaddress_a": "01234",
            "macaddress_b": "56768"
        }
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "107819"
        ]
    },
    "id": 1
}
```

Criando um host com interface SNMP

Crie um host chamado ”SNMP host” com uma interface SNMPv3 com detalhes.

Request:

```
{
    "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
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10658"
        ]
    },
    "id": 1
}
```

Criando um host com criptografia PSK

Crie um host chamado "PSK host" com criptografia PSK configurada. Observe que o host deve ser pré-configurado para usar PSK.

Requisição:

```
{
    "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
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10590"
        ]
    },
    "id": 1
}
```

Criando um host monitorado por um proxy

Crie um host que será monitorado pelo proxy com ID "1".

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "host.create",
    "params": {
        "host": "Host monitored by proxy",
```

```
        "groups": [
            {
                "groupid": "2"
            }
        ],
        "monitored_by": 1,
        "proxyid": 1
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10591"
        ]
    },
    "id": 1
}
```

Criando um host monitorado por um grupo de proxy

Crie um host que será monitorado por um grupo de proxy com ID ”1”.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "host.create",
    "params": {
        "host": "Host monitored by proxy group",
        "groups": [
            {
                "groupid": "2"
            }
        ],
        "monitored_by": 2,
        "proxy_groupid": 1
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10592"
        ]
    },
    "id": 1
}
```

Veja também

- Grupo de hosts
- Template
- Macro de usuário
- Interface de host
- Inventário de host
- Tag de host
- Proxy
- Grupo de proxies

Fonte

CHost::create() em *ui/include/classes/api/services/CHost.php*.

host.delete

Descrição

`object host.delete(array hosts)`

Este método permite excluir hosts.

> **Note:**
> Este método está disponível apenas para os tipos de usuário *Admin* e *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(array)` IDs de hosts a serem excluídos.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos hosts excluídos sob a propriedade `hostids`.

Exemplos

Excluindo múltiplos hosts

Exclua dois hosts.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "host.delete",
    "params": [
        "13",
        "32"
    ],
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "13",
            "32"
        ]
    },
    "id": 1
}
```

Fonte

CHost::delete() em *ui/include/classes/api/services/CHost.php*.

host.get

Descrição

`integer/array host.get(object parameters)`

O método permite recuperar hosts de acordo com os parâmetros fornecidos.

Note:
Este método está disponível para usuários de qualquer tipo. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object)` Parâmetros que definem a saída desejada.

O método suporta os seguintes parâmetros.

Parâmetro	Tipo	Descrição
groupids	ID/array	Retorna apenas hosts que pertencem aos grupos informados.
dserviceids	ID/array	Retorna apenas hosts que estão relacionados aos serviços descobertos informados.
graphids	ID/array	Retorna apenas hosts que possuem os gráficos informados.
hostids	ID/array	Retorna apenas hosts com os IDs de host informados.
httptestids	ID/array	Retorna apenas hosts que possuem as verificações web informadas.
interfaceids	ID/array	Retorna apenas hosts que usam as interfaces informadas.
itemids	ID/array	Retorna apenas hosts que possuem os items informados.
maintenanceids	ID/array	Retorna apenas hosts que são afetados pelas manutenções informadas.
monitored_hosts	flag	Retorna apenas hosts monitorados.
proxyids	ID/array	Retorna apenas hosts que são monitorados pelos proxies informados.
proxy_groupids	ID/array	Retorna apenas hosts que são monitorados pelos grupos de proxy informados.
templated_hosts	flag	Retorna hosts e templates.
templateids	ID/array	Retorna apenas hosts que estão vinculados aos templates informados.
triggerids	ID/array	Retorna apenas hosts que possuem os triggers informados.
with_items	flag	Retorna apenas hosts que possuem items. Sobrescreve os parâmetros `with_monitored_items` e `with_simple_graph_items`.
with_item_prototypes	flag	Retorna apenas hosts que possuem protótipos de item. Sobrescreve o parâmetro `with_simple_graph_item_prototypes`.
with_simple_graph_item_prototypes	flag	Retorna apenas hosts que possuem protótipos de item, que estão habilitados para criação e possuem tipo de informação numérica.
with_graphs	flag	Retorna apenas hosts que possuem gráficos.
with_graph_prototypes	flag	Retorna apenas hosts que possuem protótipos de gráfico.
with_httptests	flag	Retorna apenas hosts que possuem verificações web. Sobrescreve o parâmetro `with_monitored_httptests`.
with_monitored_httptests	flag	Retorna apenas hosts que possuem verificações web habilitadas.
with_monitored_items	flag	Retorna apenas hosts que possuem items habilitados. Sobrescreve o parâmetro `with_simple_graph_items`.
with_monitored_triggers	flag	Retorna apenas hosts que possuem triggers habilitados. Todos os items usados no trigger também devem estar habilitados.
with_simple_graph_items	flag	Retorna apenas hosts que possuem items com tipo de informação numérica.
with_triggers	flag	Retorna apenas hosts que possuem triggers. Sobrescreve o parâmetro `with_monitored_triggers`.
withProblemsSuppressed	boolean	Se definido como `true`, retorna apenas hosts com problemas suprimidos.
evaltype	integer	Método de avaliação da tag. Valores possíveis: 0 - *(padrão)* E/Ou; 2 - Ou.
severities	integer/array	Retorna hosts que possuem apenas problemas com as severidades informadas. Aplica-se apenas se o objeto do problema for trigger.

Parâmetro	Tipo	Descrição
tags	object/array	Retorna apenas hosts com as tags informadas. Formato: `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`. Um array vazio retorna todos os hosts. Valores possíveis para operator: 0 - *(padrão)* Contém; 1 - Igual; 2 - Não contém; 3 - Diferente; 4 - Existe; 5 - Não existe.
inheritedTags	boolean	Retorna hosts que possuem as `tags` informadas também em todos os seus templates vinculados. Valores possíveis: `true` - os templates vinculados também devem possuir as `tags` informadas; `false` - *(padrão)* as tags dos templates vinculados são ignoradas.
selectDiscoveries	query	Retorna uma propriedade `discoveries` com as regras de descoberta de baixo nível do host. Suporta `count`.
selectDiscoveryRule	query	Retorna uma propriedade `discoveryRule` com a regra de descoberta de baixo nível que criou o host (a partir do protótipo de host no monitoramento VMware).
selectGraphs	query	Retorna uma propriedade `graphs` com os gráficos do host. Suporta `count`.
selectHostDiscovery	query	Retorna uma propriedade `hostDiscovery` com os dados do objeto de descoberta de host. O objeto de descoberta de host vincula um host descoberto a um protótipo de host ou protótipos de host a uma regra LLD e possui as seguintes propriedades: `host` - *(string)* host do protótipo de host; `hostid` - *(string)* ID do host descoberto ou protótipo de host; `parent_hostid` - *(string)* ID do protótipo de host a partir do qual o host foi criado; `parent_itemid` - *(string)* ID da regra LLD que criou o host descoberto; `lastcheck` - *(timestamp)* hora em que o host foi descoberto pela última vez; `status` - *(int)* status da descoberta do host: 0 - *(padrão)* host foi descoberto, 1 - host não é mais descoberto; `ts_delete` - *(timestamp)* hora em que um host que não é mais descoberto será excluído; `ts_disable` - *(timestamp)* hora em que um host que não é mais descoberto será desabilitado; `disable_source` - *(int)* indicador se o host foi desabilitado por uma regra LLD ou manualmente: 0 - *(padrão)* desabilitado automaticamente, 1 - desabilitado por uma regra LLD.
selectHostGroups	query	Retorna uma propriedade `hostgroups` com os dados dos grupos de host aos quais o host pertence.
selectHttpTests	query	Retorna uma propriedade `httpTests` com os cenários web do host. Suporta `count`.
selectInterfaces	query	Retorna uma propriedade `interfaces` com as interfaces do host. Suporta `count`.

Parâmetro	Tipo	Descrição
selectInventory	query	Retorna uma propriedade `inventory` com os dados de inventário do host.
selectItems	query	Retorna uma propriedade `items` com os items do host. Suporta `count`.
selectMacros	query	Retorna uma propriedade `macros` com as macros do host.
selectParentTemplates	query	Retorna uma propriedade `parentTemplates` com os templates aos quais o host está vinculado. Além dos campos do objeto Template, contém `link_type` - `(integer)` a forma como o template está vinculado ao host. Valores possíveis: 0 - *(padrão)* vinculado manualmente; 1 - vinculado automaticamente por LLD. Suporta `count`.
selectDashboards	query	Retorna uma propriedade `dashboards`. Suporta `count`.
selectTags	query	Retorna uma propriedade `tags` com as tags do host.
selectInheritedTags	query	Retorna uma propriedade `inheritedTags` com as tags que estão em todos os templates vinculados ao host.
selectTriggers	query	Retorna uma propriedade `triggers` com os triggers do host. Suporta `count`.
selectValueMaps	query	Retorna uma propriedade `valuemaps` com os mapas de valores do host.
filter	object	Retorna apenas os resultados que correspondem exatamente ao filtro informado. Aceita um objeto, onde as chaves são nomes de propriedades e os valores são um único valor ou um array de valores para comparar. Não suporta propriedades do tipo de dado `text` data type. Suporta propriedades adicionais: propriedades de interface de host.
limitSelects	integer	Limita o número de registros retornados por subconsultas. Aplica-se às seguintes subconsultas: `selectParentTemplates` - os resultados serão ordenados por `host`; `selectInterfaces`; `selectItems` - ordenados por `name`; `selectDiscoveries` - ordenados por `name`; `selectTriggers` - ordenados por `description`; `selectGraphs` - ordenados por `name`; `selectDashboards` - ordenados por `name`.
search	object	Retorna resultados que correspondam ao padrão informado (case-insensitive). Aceita um objeto, onde as chaves são nomes de propriedades e os valores são strings para pesquisar. Se nenhuma opção adicional for informada, será realizada uma busca `LIKE "%...%"`. Suporta apenas propriedades dos tipos de dado `string` e `text` data type. Suporta propriedades adicionais: propriedades de interface de host.

Parâmetro	Tipo	Descrição
searchInventory	object	Retorna hosts que possuem dados de inventário que correspondam ao padrão informado (case-insensitive). Aceita um objeto, onde as chaves são nomes de propriedades e os valores são strings para pesquisar. Se nenhuma opção adicional for informada, será realizada uma busca `LIKE "%…%"`. Suporta apenas propriedades dos tipos de dado `string` e `text` data type.
sortfield	string/array	Ordena o resultado pelas propriedades informadas. Valores possíveis: `hostid`, `host`, `name`, `status`.
countOutput	boolean	Esses parâmetros estão descritos no comentário de referência.
editable	boolean	
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	
selectGroups (obsoleto)	query	Este parâmetro está obsoleto, por favor use `selectHostGroups`. Retorna uma propriedade `groups` com os dados dos grupos de host aos quais o host pertence.

Valores de retorno

`(integer/array)` Retorna:

- uma matriz de objetos;
- a contagem de objetos recuperados, se o parâmetro `countOutput` tiver sido usado.

Exemplos

Recuperando dados pelo nome

Recupere todos os dados sobre dois hosts chamados ”Zabbix server” e ”Linux server”.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "filter": {
            "host": [
                "Zabbix server",
                "Linux server"
            ]
        }
    },
    "id": 1
}
```

Resposta:

```
{
    "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": "The Zabbix monitoring server.",
            "tls_connect": "1",
            "tls_accept": "1",
            "tls_issuer": "",
            "tls_subject": "",
            "proxy_groupid": "0",
            "monitored_by": "0",
            "inventory_mode": "1",
            "active_available": "1",
            "assigned_proxyid": "0"
        },
        {
            "hostid": "10167",
            "proxyid": "0",
            "host": "Linux server",
            "status": "0",
            "ipmi_authtype": "-1",
            "ipmi_privilege": "2",
            "ipmi_username": "",
            "ipmi_password": "",
            "maintenanceid": "0",
            "maintenance_status": "0",
            "maintenance_type": "0",
            "maintenance_from": "0",
            "name": "Linux server",
            "flags": "0",
            "description": "",
            "tls_connect": "1",
            "tls_accept": "1",
            "tls_issuer": "",
            "tls_subject": "",
            "proxy_groupid": "0",
            "monitored_by": "0",
            "inventory_mode": "1",
            "active_available": "1",
            "assigned_proxyid": "0"
        }
    ],
    "id": 1
}
```


Recuperando grupos de hosts

Recupere os grupos de hosts dos quais o host ”Zabbix server” é membro.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": ["hostid"],
        "selectHostGroups": "extend",
        "filter": {
```

```
            "host": [
                "Zabbix server"
            ]
        }
    },
    "id": 1
}
```

Resposta:

```
{
    "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
}
```

Recuperando templates vinculados

Recupere os IDs e nomes dos templates vinculados ao host ”10084”.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": ["hostid"],
        "selectParentTemplates": [
            "templateid",
            "name"
        ],
        "hostids": "10084"
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "hostid": "10084",
            "parentTemplates": [
                {
                    "name": "Linux",
                    "templateid": "10001"
                },
```

```
                {
                    "name": "Zabbix Server",
                    "templateid": "10047"
                }
            ]
        }
    ],
    "id": 1
}
```

Recuperando hosts por template

Recupere hosts que tenham o template ”10001” (*Linux por Zabbix agent*) vinculado a eles.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": ["hostid", "name"],
        "templateids": "10001"
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "templateid": "10001",
            "hosts": [
                {
                    "hostid": "10084",
                    "name": "Zabbix server"
                },
                {
                    "hostid": "10603",
                    "name": "Host 1"
                },
                {
                    "hostid": "10604",
                    "name": "Host 2"
                }
            ]
        }
    ],
    "id": 1
}
```

Pesquisa por dados de inventário do host

Recupera hosts que contenham ”Linux” no campo ”OS” do inventário do host.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": [
            "host"
        ],
        "selectInventory": [
            "os"
```

```
        ],
        "searchInventory": {
            "os": "Linux"
        }
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "hostid": "10084",
            "host": "Zabbix server",
            "inventory": {
                "os": "Linux Ubuntu"
            }
        },
        {
            "hostid": "10107",
            "host": "Linux server",
            "inventory": {
                "os": "Linux Mint"
            }
        }
    ],
    "id": 1
}
```

Pesquisa por tags de host

Recuperar hosts que possuem a tag ”host-name” igual a ”linux-server”.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": ["hostid"],
        "selectTags": "extend",
        "evaltype": 0,
        "tags": [
            {
                "tag": "host-name",
                "value": "linux-server",
                "operator": 1
            }
        ]
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "hostid": "10085",
            "tags": [
                {
                    "tag": "host-name",
                    "value": "linux-server",
```

```
                    "automatic": "0"
                },
                {
                    "tag": "os",
                    "value": "rhel-7",
                    "automatic": "0"
                }
            ]
        }
    ],
    "id": 1
}
```

Recuperar hosts que possuem essas tags não apenas no nível do host, mas também em seus templates pai vinculados.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": ["name"],
        "tags": [
            {
                "tag": "os",
                "value": "rhel-7",
                "operator": 1
            }
        ],
        "inheritedTags": true
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "hostid": "10623",
            "name": "PC room 1"
        },
        {
            "hostid": "10601",
            "name": "Office"
        }
    ],
    "id": 1
}
```

Recuperar hosts excluindo aqueles com a tag ”component”.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": ["hostid"],
        "selectTags": "extend",
        "evaltype": 0,
        "tags": [
            {
                "tag": "component",
                "value": "",
```

```
                "operator": 5
            }
        ]
    },
    "id": 1
}
```

Resposta:

```
{
    "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
}
```

Pesquisando host com tags e tags de template

Recupera um host com tags e todas as tags que estão vinculadas aos templates pai.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": ["name"],
        "hostids": 10502,
        "selectTags": ["tag", "value"],
        "selectInheritedTags": ["tag", "value"]
    },
    "id": 1
}
```

Resposta:

```
{
    "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
}
```


Pesquisando hosts por gravidade do problema

Recuperar hosts que possuem problemas de ”Desastre”.

Requisição:


```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": ["name"],
        "severities": 5
    },
    "id": 1
}
```


Resposta:


```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "hostid": "10160",
            "name": "Zabbix server"
        }
    ],
    "id": 1
}
```


Recuperar hosts que possuem problemas de ”Média” e ”Alta” gravidade.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": ["name"],
        "severities": [3, 4]
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "hostid": "20170",
            "name": "Database"
        },
        {
            "hostid": "20183",
            "name": "workstation"
        }
    ],
    "id": 1
}
```

Recuperando itens do host

Recupere itens para o host "Zabbix server". A solicitação é limitada às propriedades do item `itemid`, `name` e `status`. Devido ao grande tamanho da resposta, apenas um subconjunto de itens é mostrado no exemplo.

Solicitação:

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": ["hostid"],
        "selectItems": ["itemid","name","status"],
        "filter": {
            "host": [
                "Zabbix server"
            ]
        }
    },
    "id": 1
}
```

Resposta:

```
{
  "jsonrpc": "2.0",
  "result": [
    {
      "hostid": "10084",
      "items": [
        {
          "itemid": "42227",
          "name": "Host name of Zabbix agent running",
          "status": "0"
        },
        {
          "itemid": "42237",
          "name": "Zabbix agent ping",
```

```
      "status": "0"
    },
    {
      "itemid": "42250",
      "name": "Version of Zabbix agent running",
      "status": "0"
    },
    {
      "itemid": "42251",
      "name": "Maximum number of open file descriptors",
      "status": "0"
    },
    {
      "itemid": "42252",
      "name": "Maximum number of processes",
      "status": "0"
    },
    {
      "itemid": "69869",
      "name": "Interface enp0s3: Inbound packets discarded",
      "status": "0"
    },
    {
      "itemid": "69870",
      "name": "Interface enp0s3: Inbound packets with errors",
      "status": "0"
    },
    {
      "itemid": "69871",
      "name": "Interface enp0s3: Bits received",
      "status": "0"
    },
    {
      "itemid": "69872",
      "name": "Interface enp0s3: Outbound packets discarded",
      "status": "0"
    },
    {
      "itemid": "69873",
      "name": "Interface enp0s3: Outbound packets with errors",
      "status": "0"
    },
    {
      "itemid": "69874",
      "name": "Interface enp0s3: Bits sent",
      "status": "0"
    },
    {
      "itemid": "42253",
      "name": "Number of processes",
      "status": "0"
    },
    {
      "itemid": "42254",
      "name": "Number of running processes",
      "status": "0"
    },
    {
      "itemid": "42255",
      "name": "System boot time",
      "status": "0"
    }
```

```
        ]
      }
    ],
    "id": 1
}
```

Veja também

- Grupo de hosts
- Template
- Macro de usuário
- Interface de host
- Proxy
- Grupo de proxies

Fonte

CHost::get() em *ui/include/classes/api/services/CHost.php*.

host.massadd

Descrição

`object host.massadd(object parameters)`

Este método permite adicionar simultaneamente vários objetos relacionados a todos os hosts fornecidos.

> **Note:**
> Este método está disponível apenas para os tipos de usuário *Admin* e *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object)` Parâmetros contendo os IDs dos hosts a serem atualizados e os objetos a serem adicionados a todos os hosts.

O método aceita os seguintes parâmetros.

Parâmetro	Tipo	Descrição
hosts	object/array	Hosts a serem atualizados. Os hosts devem ter apenas a propriedade `hostid` definida. Comportamento do parâmetro: - *obrigatório*
groups	object/array	Grupos de hosts a serem adicionados aos hosts fornecidos. Os grupos de hosts devem ter apenas a propriedade `groupid` definida.
interfaces	object/array	Interfaces de host a serem criadas para os hosts fornecidos.
macros	object/array	Macros de usuário a serem criadas para os hosts fornecidos.
templates	object/array	Templates a serem vinculados aos hosts fornecidos. Os templates devem ter apenas a propriedade `templateid` definida.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos hosts atualizados sob a propriedade `hostids`.

Exemplos

Adicionando macros

Adicione duas novas macros a dois hosts.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "host.massadd",
    "params": {
        "hosts": [
            {
                "hostid": "10160"
            },
            {
                "hostid": "10167"
            }
        ],
        "macros": [
            {
                "macro": "{$TEST1}",
                "value": "MACROTEST1"
            },
            {
                "macro": "{$TEST2}",
                "value": "MACROTEST2",
                "description": "Test description"
            }
        ]
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10160",
            "10167"
        ]
    },
    "id": 1
}
```

Veja também

- host.update
- Grupo de hosts
- Template
- Macro de usuário
- Interface de host

Fonte

CHost::massAdd() em *ui/include/classes/api/services/CHost.php*.

host.massremove

Descrição

`object host.massremove(object parameters)`

Este método permite remover objetos relacionados de vários hosts.

> **Note:**
> Este método está disponível apenas para os tipos de usuário *Admin* e *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object)` Parâmetros contendo os IDs dos hosts a serem atualizados e os objetos que devem ser removidos.

Parâmetro	Tipo	Descrição
hostids	ID/array	IDs dos hosts a serem atualizados. Comportamento do parâmetro: - *obrigatório*
groupids	ID/array	IDs dos grupos de hosts dos quais os hosts fornecidos serão removidos.
interfaces	object/array	Interfaces de host a serem removidas dos hosts fornecidos. O objeto de interface de host deve ter apenas as propriedades `ip`, `dns` e `port` definidas.
macros	string/array	Macros de usuário a serem excluídas dos hosts fornecidos.
templateids	ID/array	IDs dos templates a serem desvinculados dos hosts fornecidos.
templateids_clear	ID/array	IDs dos templates a serem desvinculados e limpos dos hosts fornecidos.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos hosts atualizados sob a propriedade `hostids`.

Exemplos

Desvinculando templates

Desvincule um template de dois hosts e exclua todas as entidades do template.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "host.massremove",
    "params": {
        "hostids": ["69665", "69666"],
        "templateids_clear": "325"
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "69665",
            "69666"
        ]
    },
    "id": 1
}
```

Veja também

- host.update
- Macro de usuário
- Interface de host

Fonte

CHost::massRemove() em *ui/include/classes/api/services/CHost.php*.

host.massupdate

Descrição

`object host.massupdate(object parameters)`

Este método permite substituir ou remover objetos relacionados simultaneamente e atualizar propriedades em vários hosts.

> **Note:**
> Este método está disponível apenas para os tipos de usuário *Admin* e *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função de usuário. Consulte Funções de usuário para mais informações.

Parâmetros

`(object)` Parâmetros contendo os IDs dos hosts a serem atualizados e as propriedades que devem ser atualizadas.

Além das propriedades padrão do host, o método aceita os seguintes parâmetros.

Parâmetro	Tipo	Descrição
hosts	object/array	Hosts a serem atualizados. Os hosts devem ter apenas a propriedade `hostid` definida. Comportamento do parâmetro: - *obrigatório*
groups	object/array	Grupos de hosts para substituir os grupos de hosts atuais aos quais os hosts pertencem. Os grupos de hosts devem ter apenas a propriedade `groupid` definida.
interfaces	object/array	Interfaces de host para substituir as interfaces de host atuais nos hosts fornecidos.
inventory	object	Propriedades do Inventário do host. O modo de inventário do host não pode ser atualizado usando o parâmetro `inventory`, use `inventory_mode` em vez disso.
macros	object/array	Macros de usuário para substituir as macros de usuário atuais nos hosts fornecidos.
templates	object/array	Templates para substituir os templates atualmente vinculados nos hosts fornecidos. Os templates devem ter apenas a propriedade `templateid` definida.
templates_clear	object/array	Templates para desvincular e limpar dos hosts fornecidos. Os templates devem ter apenas a propriedade `templateid` definida.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos hosts atualizados sob a propriedade `hostids`.

Exemplos

Habilitando múltiplos hosts

Habilite o monitoramento de dois hosts, ou seja, defina seu status como "0".

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "host.massupdate",
    "params": {
        "hosts": [
            {
                "hostid": "69665"
            },
            {
                "hostid": "69666"
            }
```

```
        ],
        "status": 0
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "69665",
            "69666"
        ]
    },
    "id": 1
}
```

Veja também

- host.update
- host.massadd
- host.massremove
- Grupo de host
- Template
- Macro de usuário
- Interface de host

Fonte

CHost::massUpdate() em *ui/include/classes/api/services/CHost.php*.

host.update

Descrição

`object host.update(object/array hosts)`

Este método permite atualizar hosts existentes.

Note:
Este método está disponível apenas para os tipos de usuário *Admin* e *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object/array)` Propriedades do host a serem atualizadas.

A propriedade `hostid` deve ser definida para cada host, todas as outras propriedades são opcionais. Somente as propriedades fornecidas serão atualizadas, todas as outras permanecerão inalteradas.

Observe, no entanto, que atualizar o nome técnico do host também atualizará o nome visível do host (se não for fornecido separadamente) pelo valor do nome técnico do host.

Além das propriedades padrão do host, o método aceita os seguintes parâmetros.

Parâmetro	Tipo	Descrição
groups	object/array	Grupos de hosts para substituir os grupos de hosts atuais aos quais o host pertence. Todos os grupos de hosts que não estiverem listados na solicitação serão desvinculados. Os grupos de hosts devem ter apenas a propriedade `groupid` definida.

Parâmetro	Tipo	Descrição
interfaces	object/array	Interfaces de host para substituir as interfaces de host atuais. Todas as interfaces que não estiverem listadas na solicitação serão removidas.
tags	object/array	Tags do host para substituir as tags de host atuais. Todas as tags que não estiverem listadas na solicitação serão removidas.
inventory	object	Propriedades do inventário do host.
macros	object/array	Macros de usuário para substituir as macros de usuário atuais. Todas as macros que não estiverem listadas na solicitação serão removidas.
templates	object/array	Templates para substituir os templates atualmente vinculados. Todos os templates que não estiverem listados na solicitação serão apenas desvinculados. Os templates devem ter apenas a propriedade `templateid` definida.
templates_clear	object/array	Templates para desvincular e limpar do host. Os templates devem ter apenas a propriedade `templateid` definida.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos hosts atualizados sob a propriedade `hostids`.

Exemplos

Habilitando um host

Habilite o monitoramento do host, ou seja, defina seu status como ”0”.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "host.update",
    "params": {
        "hostid": "10126",
        "status": 0
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10126"
        ]
    },
    "id": 1
}
```

Desvinculando templates

Desvincule e limpe dois templates do host.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "host.update",
    "params": {
        "hostid": "10126",
        "templates_clear": [
            {
```

```
                "templateid": "10124"
            },
            {
                "templateid": "10125"
            }
        ]
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10126"
        ]
    },
    "id": 1
}
```

Atualizando macros de host

Substitua todas as macros de host por duas novas.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "host.update",
    "params": {
        "hostid": "10126",
        "macros": [
            {
                "macro": "{$PASS}",
                "value": "password"
            },
            {
                "macro": "{$DISC}",
                "value": "sda",
                "description": "Updated description"
            }
        ]
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10126"
        ]
    },
    "id": 1
}
```

Atualizando o inventário do host

Alterar o modo de inventário e adicionar localização

Requisição:

```
{
    "jsonrpc": "2.0",
```

```
    "method": "host.update",
    "params": {
        "hostid": "10387",
        "inventory_mode": 0,
        "inventory": {
            "location": "Latvia, Riga"
        }
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10387"
        ]
    },
    "id": 1
}
```

Atualizando tags de host

Substitua todas as tags de host por uma nova.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "host.update",
    "params": {
        "hostid": "10387",
        "tags": {
            "tag": "os",
            "value": "rhel-7"
        }
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10387"
        ]
    },
    "id": 1
}
```

Atualizando macros de host descobertas

Converta a macro ”automática” criada pela regra de descoberta para ”manual” e altere seu valor para ”novo-valor”.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "host.update",
    "params": {
        "hostid": "10387",
        "macros": {
            "hostmacroid": "5541",
            "value": "new-value",
```

```
            "automatic": "0"
        }
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10387"
        ]
    },
    "id": 1
}
```

Atualizando a criptografia do host

Atualize o host ”10590” para usar apenas criptografia PSK para conexões do host para o Zabbix server e altere a identidade PSK e a chave PSK. Observe que o host deve ser pré-configurado para usar PSK.

Requisição:

```
{
    "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
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10590"
        ]
    },
    "id": 1
}
```

Veja também

- host.massadd
- host.massupdate
- host.massremove
- Grupo de hosts
- Template
- Macro de usuário
- Interface de host
- Inventário de host
- Tag de host
- Proxy
- Grupo de proxies

Fonte

CHost::update() em *ui/include/classes/api/services/CHost.php*.

Host descoberto

Esta classe foi projetada para trabalhar com hosts descobertos.

Referências de objeto:

- Host descoberto

Métodos disponíveis:

- dhost.get - recuperar hosts descobertos

Objeto host descoberto

Os seguintes objetos estão diretamente relacionados à API `dhost`.

Host descoberto

> **Note:**
> Hosts descobertos são criados pelo Zabbix server e não podem ser modificados via API.

O objeto host descoberto contém informações sobre um host descoberto por uma regra de descoberta de rede. Ele possui as seguintes propriedades.

Propriedade	Tipo	Descrição
dhostid	ID	ID do host descoberto.
druleid	ID	ID da regra de descoberta que detectou o host.
lastdown	timestamp	Hora em que o host descoberto ficou inativo pela última vez.
lastup	timestamp	Hora em que o host descoberto ficou ativo pela última vez.
status	integer	Se o host descoberto está ativo ou inativo. Um host está ativo se tiver pelo menos um serviço descoberto ativo. Valores possíveis: 0 - host ativo; 1 - host inativo.

#####dhost.get {#manual-api-reference-dhost-get}

Descrição

`integer/array dhost.get(object parameters)`

O método permite recuperar hosts descobertos de acordo com os parâmetros fornecidos.

> **Note:**
> Este método está disponível para usuários de qualquer tipo. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object)` Parâmetros que definem a saída desejada.

O método suporta os seguintes parâmetros.

Parâmetro	Tipo	Descrição
dhostids	ID/array	Retorna apenas hosts descobertos com os IDs fornecidos.
druleids	ID/array	Retorna apenas hosts descobertos que foram criados pelas regras de descoberta fornecidas.
dserviceids	ID/array	Retorna apenas hosts descobertos que estão executando os serviços fornecidos.
selectDRules	query	Retorna uma propriedade `drules` com um array das regras de descoberta que detectaram o host.

Parâmetro	Tipo	Descrição
selectDServices	query	Retorna uma propriedade `dservices` com os serviços descobertos em execução no host. Suporta `count`.
limitSelects	integer	Limita o número de registros retornados por subconsultas. Aplica-se às seguintes subconsultas: `selectDServices` - os resultados serão ordenados por `dserviceid`.
sortfield	string/array	Ordena o resultado pelas propriedades fornecidas. Valores possíveis: `dhostid`, `druleid`.
countOutput	boolean	Esses parâmetros são descritos no comentário de referência.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valores de retorno

`(integer/array)` Retorna:

- uma matriz de objetos;
- a contagem de objetos recuperados, se o parâmetro `countOutput` tiver sido usado.

Exemplos

Recuperar hosts descobertos por regra de descoberta

Recupere todos os hosts e os serviços descobertos que estão sendo executados e que foram detectados pela regra de descoberta "4".

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "dhost.get",
    "params": {
        "output": "extend",
        "selectDServices": "extend",
        "druleids": "4"
    },
    "id": 1
}
```

Resposta:

```
{
    "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
}
```


Veja também

- Serviço descoberto
- Regra de descoberta

Fonte

CDHost::get() em *ui/include/classes/api/services/CDHost.php*.

Housekeeping

Esta classe foi projetada para trabalhar com housekeeping.

Referências de objetos:

- Housekeeping

Métodos disponíveis:

- housekeeping.get - recuperar housekeeping
- housekeeping.update - atualizar housekeeping

Objeto de limpeza

Os seguintes objetos estão diretamente relacionados à API `housekeeping`.

Housekeeping

O objeto de configurações possui as seguintes propriedades.

Propriedade	Tipo	Descrição
hk_events_mode	integer	Habilitar housekeeping interno para eventos e alertas. Valores possíveis: 0 - Desabilitar; 1 - *(padrão)* Habilitar.
hk_events_trigger	string	Período de armazenamento dos dados de trigger. Aceita segundos e unidade de tempo com sufixo. Padrão: 365d.

Propriedade	Tipo	Descrição
hk_events_service	string	Período de armazenamento dos dados de serviço. Aceita segundos e unidade de tempo com sufixo. Padrão: 1d.
hk_events_internal	string	Período de armazenamento dos dados internos. Aceita segundos e unidade de tempo com sufixo. Padrão: 1d.
hk_events_discovery	string	Período de armazenamento dos dados de descoberta de rede. Aceita segundos e unidade de tempo com sufixo. Padrão: 1d.
hk_events_autoreg	string	Período de armazenamento dos dados de auto-registro. Aceita segundos e unidade de tempo com sufixo. Padrão: 1d.
hk_services_mode	integer	Habilitar housekeeping interno para serviços. Valores possíveis: 0 - Desabilitar; 1 - *(padrão)* Habilitar.
hk_services	string	Período de armazenamento dos dados de serviços. Aceita segundos e unidade de tempo com sufixo. Padrão: 365d.
hk_audit_mode	integer	Habilitar housekeeping interno para auditoria. Valores possíveis: 0 - Desabilitar; 1 - *(padrão)* Habilitar.
hk_audit	string	Período de armazenamento dos dados de auditoria. Aceita segundos e unidade de tempo com sufixo. Padrão: 31d.
hk_sessions_mode	integer	Habilitar housekeeping interno para sessões. Valores possíveis: 0 - Desabilitar; 1 - *(padrão)* Habilitar.
hk_sessions	string	Período de armazenamento dos dados de sessões. Aceita segundos e unidade de tempo com sufixo. Padrão: 365d.
hk_history_mode	integer	Habilitar housekeeping interno para histórico. Valores possíveis: 0 - Desabilitar; 1 - *(padrão)* Habilitar.
hk_history_global	integer	Sobrescrever o período de histórico do item. Valores possíveis: 0 - Não sobrescrever; 1 - *(padrão)* Sobrescrever.
hk_history	string	Período de armazenamento dos dados de histórico. Aceita segundos e unidade de tempo com sufixo. Padrão: 31d.
hk_trends_mode	integer	Habilitar housekeeping interno para tendências. Valores possíveis: 0 - Desabilitar; 1 - *(padrão)* Habilitar.

Propriedade	Tipo	Descrição
hk_trends_global	integer	Sobrescrever o período de tendências do item. Valores possíveis: 0 - Não sobrescrever; 1 - *(padrão)* Sobrescrever.
hk_trends	string	Período de armazenamento dos dados de tendências. Aceita segundos e unidade de tempo com sufixo. Padrão: 365d.
db_extension	string	Flag de configuração de extensão do banco de dados. Se esta flag estiver definida como ”timescaledb”, o server altera seu comportamento para housekeeping e exclusão de item. Comportamento da propriedade: - *somente leitura*
compression_availability	integer	Se a compactação de dados é suportada pelo banco de dados (ou sua extensão). Valores possíveis: 0 - Indisponível; 1 - Disponível. Comportamento da propriedade: - *somente leitura*
compression_status	integer	Habilitar a compactação do TimescaleDB para histórico e tendências. Valores possíveis: 0 - *(padrão)* Desligado; 1 - Ligado.
compress_older	string	Compactar registros de histórico e tendências mais antigos que o período especificado. Aceita segundos e unidade de tempo com sufixo. Padrão: 7d.

housekeeping.get

Descrição

`object housekeeping.get(object parameters)`

O método permite recuperar o objeto housekeeping de acordo com os parâmetros fornecidos.

> **Note:**
> Este método está disponível para usuários de qualquer tipo. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object)` Parâmetros que definem a saída desejada.

O método suporta apenas um parâmetro.

Parâmetro	Tipo	Descrição
output	query	Este parâmetro é descrito no comentário de referência.

Valores de retorno

`(object)` Retorna o objeto de limpeza.

Exemplos

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "housekeeping.get",
    "params": {
        "output": "extend"
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hk_events_mode": "1",
        "hk_events_trigger": "365d",
        "hk_events_service": "1d",
        "hk_events_internal": "1d",
        "hk_events_discovery": "1d",
        "hk_events_autoreg": "1d",
        "hk_services_mode": "1",
        "hk_services": "365d",
        "hk_audit_mode": "1",
        "hk_audit": "31d",
        "hk_sessions_mode": "1",
        "hk_sessions": "365d",
        "hk_history_mode": "1",
        "hk_history_global": "0",
        "hk_history": "31d",
        "hk_trends_mode": "1",
        "hk_trends_global": "0",
        "hk_trends": "365d",
        "db_extension": "",
        "compression_status": "0",
        "compress_older": "7d"
    },
    "id": 1
}
```

Fonte

CHhousekeeping ::get() in *ui/include/classes/api/services/CHousekeeping.php*.

housekeeping.update

Descrição

`object housekeeping.update(object housekeeping)`

Este método permite atualizar as configurações de housekeeping existentes.

> **Note:**
> Este método está disponível apenas para o tipo de usuário *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object)` Propriedades de housekeeping a serem atualizadas.

Valores de retorno

`(array)` Retorna um array com os nomes dos parâmetros atualizados.

Exemplos

Requisição:

```
{
    "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
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": [
        "hk_events_mode",
        "hk_events_trigger",
        "hk_events_internal",
        "hk_events_discovery"
    ],
    "id": 1
}
```

Fonte

CHousekeeping::update() em *ui/include/classes/api/services/CHousekeeping.php*.

Imagem

Esta classe foi projetada para trabalhar com imagens.

Referências de objetos:

- Image

Métodos disponíveis:

- image.create - criar novas imagens
- image.delete - excluir imagens
- image.get - recuperar imagens
- image.update - atualizar imagens

Objeto image

Os seguintes objetos estão diretamente relacionados à API `image`.

Imagem

O objeto imagem possui as seguintes propriedades.

Propriedade	Tipo	Descrição
imageid	ID	ID da imagem. Comportamento da propriedade: - *somente leitura* - *obrigatório* para operações de atualização
name	string	Nome da imagem. Comportamento da propriedade: - *obrigatório* para operações de criação

Propriedade	Tipo	Descrição
imagetype	integer	Tipo de imagem. Valores possíveis: 1 - *(padrão)* ícone; 2 - imagem de fundo. Comportamento da propriedade: - *constante* - *obrigatório* para operações de criação
image	string	Imagem codificada em Base64. O tamanho máximo da imagem codificada é de 1 MB. O tamanho máximo pode ser ajustado alterando o valor da constante ZBX_MAX_IMAGE_SIZE. Formatos de imagem suportados: PNG, JPEG, GIF e WebP. Comportamento da propriedade: - *obrigatório* para operações de criação

image.create

Descrição

`object image.create(object/array images)`

Este método permite criar novas imagens.

> **Note:**
> Este método está disponível apenas para o tipo de usuário *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Consulte Funções de usuário para mais informações.

Parâmetros

`(object/array)` Imagens a serem criadas.

O método aceita imagens com as propriedades padrão da imagem.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs das imagens criadas sob a propriedade `imageids`. A ordem dos IDs retornados corresponde à ordem das imagens passadas.

Exemplos

Criar uma imagem de ícone

Crie um ícone de nuvem.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "image.create",
    "params": {
        "imagetype": 1,
        "name": "Cloud_(24)",
        "image": "iVBORw0KGgoAAAANSUhEUgAAABgAAAANCAYAAACzbK7QAAAABHNCSVQICAgIfAhkiAAAAAlwSFlzAAACmAAAApgB
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "imageids": [
```

```
            "188"
        ]
    },
    "id": 1
}
```

Criar uma imagem de fundo

Crie uma pequena imagem de fundo para ser usada como padrão.

Request:

```
{
    "jsonrpc": "2.0",
    "method": "image.create",
    "params": {
        "imagetype": 2,
        "name": "background_image",
        "image": "iVBORw0KGgoAAAANSUhEUgAAABgAAAAYCAIAAABvFaqvAAABhWlDQ1BJQ0MgcHJvZmlsZQAAKJF9kb1Lw0AYxp+m
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "imageids": [
            "189"
        ]
    },
    "id": 1
}
```

Fonte

CImage::create() em *ui/include/classes/api/services/CImage.php*.

image.delete

Descrição

`object image.delete(array imageIds)`

Este método permite excluir imagens.

> **Note:**
> Este método está disponível apenas para o tipo de usuário *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(array)` IDs das imagens a serem excluídas.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs das imagens excluídas sob a propriedade `imageids`.

Exemplos

Excluir várias imagens

Exclua duas imagens.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "image.delete",
```

```
    "params": [
        "188",
        "192"
    ],
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "imageids": [
            "188",
            "192"
        ]
    },
    "id": 1
}
```

Fonte

CImage::delete() em *ui/include/classes/api/services/CImage.php*.

image.get

Descrição

`integer/array image.get(object parameters)`

O método permite recuperar imagens de acordo com os parâmetros fornecidos.

Note:
Este método está disponível para usuários de qualquer tipo. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object)` Parâmetros que definem a saída desejada.

O método suporta os seguintes parâmetros.

Parâmetro	Tipo	Descrição
imageids	ID/array	Retorna apenas imagens com os IDs fornecidos.
sysmapids	ID/array	Retorna imagens que são usadas nos mapas fornecidos.
select_image	flag	Retorna uma propriedade `image` com a imagem codificada em Base64.
sortfield	string/array	Ordena o resultado pelas propriedades fornecidas. Valores possíveis: `imageid`, `name`.
countOutput	boolean	Estes parâmetros são descritos no comentário de referência.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valores de retorno

`(integer/array)` Retorna:

- uma matriz de objetos;
- a contagem de objetos recuperados, se o parâmetro `countOutput` tiver sido usado.

Exemplos

Recuperar uma imagem

Recupere todos os dados da imagem com o ID "2".

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "image.get",
    "params": {
        "output": "extend",
        "select_image": true,
        "imageids": "2"
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "imageid": "2",
            "imagetype": "1",
            "name": "Cloud_(24)",
            "image": "iVBORw0KGgoAAAANSUhEUgAAABgAAAANCAYAAACzbK7QAAAABHNCSVQICAgIfAhkiAAAAAlwSFlzAAACmAAA
        }
    ],
    "id": 1
}
```

Fonte

CImage::get() em *ui/include/classes/api/services/CImage.php*.

imagem.atualização

Descrição

`object image.update(object/array images)`

Este método permite atualizar imagens existentes.

> **Note:**
> Este método está disponível apenas para o tipo de usuário *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object/array)` Propriedades da imagem a serem atualizadas.

A propriedade `imageid` deve ser definida para cada imagem, todas as outras propriedades são opcionais. Somente as propriedades passadas serão atualizadas, todas as outras permanecerão inalteradas.

O método aceita imagens com as propriedades padrão da imagem.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs das imagens atualizadas sob a propriedade `imageids`.

Exemplos

Renomear imagem

Renomear imagem para "Cloud icon".

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "image.update",
    "params": {
        "imageid": "2",
        "name": "Cloud icon"
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "imageids": [
            "2"
        ]
    },
    "id": 1
}
```

Fonte

CImage::update() em *ui/include/classes/api/services/CImage.php*.

Interface de host

Esta classe foi projetada para trabalhar com interfaces de host.

Referências de objetos:

- Interface de host
  - Detalhes

Métodos disponíveis:

- hostinterface.create - criar novas interfaces de host
- hostinterface.delete - excluir interfaces de host
- hostinterface.get - recuperar interfaces de host
- hostinterface.massadd - adicionar interfaces de host a hosts
- hostinterface.massremove - remover interfaces de host de hosts
- hostinterface.replacehostinterfaces - substituir interfaces de host em um host
- hostinterface.update - atualizar interfaces de host

Objeto de interface de host

Os seguintes objetos estão diretamente relacionados à API `hostinterface`.

Interface do host

O objeto de interface do host possui as seguintes propriedades.

Attention:
Observe que as propriedades `ip` e `dns` são *obrigatórias* para operações de criação. Se você não quiser usar DNS, defina-o como uma string vazia.

Propriedade	Tipo	Descrição
interfaceid	ID	ID da interface. Comportamento da propriedade: - *somente leitura* - *obrigatório* para operações de atualização

Propriedade	Tipo	Descrição
available	integer	Disponibilidade da interface do host. Valores possíveis: 0 - *(padrão)* desconhecido; 1 - disponível; 2 - indisponível. Comportamento da propriedade: - *somente leitura*
hostid	ID	ID do host ao qual a interface pertence. Comportamento da propriedade: - *constante* - *obrigatório* para operações de criação
type	integer	Tipo de interface. Valores possíveis: 1 - Agent; 2 - SNMP; 3 - IPMI; 4 - JMX. Comportamento da propriedade: - *obrigatório* para operações de criação
ip	string	Endereço IP usado pela interface. Pode ser vazio se a conexão for feita via DNS. Comportamento da propriedade: - *obrigatório* para operações de criação
dns	string	Nome DNS usado pela interface. Pode ser vazio se a conexão for feita via IP. Comportamento da propriedade: - *obrigatório* para operações de criação
port	string	Número da porta usada pela interface. Pode conter macros de usuário. Comportamento da propriedade: - *obrigatório* para operações de criação
useip	integer	Se a conexão deve ser feita via IP. Valores possíveis: 0 - conectar usando o nome DNS do host; 1 - conectar usando o endereço IP do host. Comportamento da propriedade: - *obrigatório* para operações de criação
main	integer	Se a interface é usada como padrão no host. Apenas uma interface de cada tipo pode ser definida como padrão em um host. Valores possíveis: 0 - não é padrão; 1 - padrão. Comportamento da propriedade: - *obrigatório* para operações de criação
details	object	Objeto details adicional para a interface. Comportamento da propriedade: - *obrigatório* se `type` for definido como ”SNMP”

Propriedade	Tipo	Descrição
disable_until	timestamp	O próximo horário de coleta de uma interface de host indisponível. Comportamento da propriedade: - *somente leitura*
error	string	Texto do erro se a interface do host estiver indisponível. Comportamento da propriedade: - *somente leitura*
errors_from	timestamp	Horário em que a interface do host ficou indisponível. Comportamento da propriedade: - *somente leitura*

Detalhes

O objeto details possui as seguintes propriedades.

Propriedade	Tipo	Descrição
version	integer	Versão da interface SNMP. Valores possíveis: 1 - SNMPv1; 2 - SNMPv2c; 3 - SNMPv3. Comportamento da propriedade: - *obrigatório*
bulk	integer	Se deve usar requisições SNMP bulk. Valores possíveis: 0 - não usar requisições bulk; 1 - (padrão) - usar requisições bulk.
community	string	Comunidade SNMP. Usado apenas por interfaces SNMPv1 e SNMPv2. Comportamento da propriedade: - *obrigatório* se `version` estiver definido como "SNMPv1" ou "SNMPv2c"
max_repetitions	integer	Valor máximo de repetição para requisições SNMP bulk nativas (GetBulkRequest-PDUs). Usado apenas para items `discovery[]` e `walk[]` em SNMPv2 e v3. Padrão: 10.
securityname	string	Nome de segurança SNMPv3. Usado apenas por interfaces SNMPv3.
securitylevel	integer	Nível de segurança SNMPv3. Usado apenas por interfaces SNMPv3. Valores possíveis: 0 - (padrão) - noAuthNoPriv; 1 - authNoPriv; 2 - authPriv.
authpassphrase	string	Senha de autenticação SNMPv3. Usado apenas por interfaces SNMPv3.
privpassphrase	string	Senha de privacidade SNMPv3. Usado apenas por interfaces SNMPv3.

Propriedade	Tipo	Descrição
authprotocol	integer	Protocolo de autenticação SNMPv3. Usado apenas por interfaces SNMPv3. Valores possíveis: 0 - (padrão) - MD5; 1 - SHA1; 2 - SHA224; 3 - SHA256; 4 - SHA384; 5 - SHA512.
privprotocol	integer	Protocolo de privacidade SNMPv3. Usado apenas por interfaces SNMPv3. Valores possíveis: 0 - (padrão) - DES; 1 - AES128; 2 - AES192; 3 - AES256; 4 - AES192C; 5 - AES256C.
contextname	string	Nome do contexto SNMPv3. Usado apenas por interfaces SNMPv3.

hostinterface.create

Descrição

`object hostinterface.create(object/array hostInterfaces)`

Este método permite criar novas interfaces de host.

Note:
Este método está disponível apenas para os tipos de usuário *Admin* e *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função de usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object/array)` Interfaces de host a serem criadas.

O método aceita interfaces de host com as propriedades padrão da interface de host.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs do host criado interfaces sob a propriedade `interfaceids`. A ordem do retorno Os IDs correspondem à ordem das interfaces de host passadas.

Exemplos

Criar uma nova interface

Crie uma interface de agent secundária de IP no host "30052".

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "hostinterface.create",
    "params": {
        "hostid": "30052",
        "main": "0",
        "type": "1",
        "useip": "1",
        "ip": "127.0.0.1",
        "dns": "",
        "port": "10050"
```

```
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "interfaceids": [
            "30062"
        ]
    },
    "id": 1
}
```

Criar uma interface com detalhes SNMP

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "hostinterface.create",
    "params": {
        "hostid": "10456",
        "main": "0",
        "type": "2",
        "useip": "1",
        "ip": "127.0.0.1",
        "dns": "",
        "port": "1601",
        "details": {
            "version": "2",
            "bulk": "1",
            "community": "{$SNMP_COMMUNITY}"
        }
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "interfaceids": [
            "30063"
        ]
    },
    "id": 1
}
```

Veja também

- hostinterface.massadd
- host.massadd

Fonte

CHostInterface::create() em *ui/include/classes/api/services/CHostInterface.php*.

hostinterface.delete

Descrição

`object hostinterface.delete(array hostInterfaceIds)`

Este método permite excluir interfaces de host.

Note:
Este método está disponível apenas para os tipos de usuário *Admin* e *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(array)` IDs das interfaces de host a serem excluídas.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs do host excluído interfaces sob a propriedade `interfaceids`.

Exemplos

Excluir uma interface de host

Exclua a interface de host com o ID 30062.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "hostinterface.delete",
    "params": [
        "30062"
    ],
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "interfaceids": [
            "30062"
        ]
    },
    "id": 1
}
```

Veja também

- hostinterface.massremove
- host.massremove

Fonte

CHostInterface::delete() em *ui/include/classes/api/services/CHostInterface.php*.

hostinterface.get

Descrição

`integer/array hostinterface.get(object parameters)`

O método permite recuperar interfaces de host de acordo com os parâmetros fornecidos.

Note:
Este método está disponível para usuários de qualquer tipo. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object)` Parâmetros que definem a saída desejada.

O método suporta os seguintes parâmetros.

Parâmetro	Tipo	Descrição
hostids	ID/array	Retorna apenas as interfaces de host usadas pelos hosts fornecidos.
interfaceids	ID/array	Retorna apenas as interfaces de host com os IDs fornecidos.
itemids	ID/array	Retorna apenas as interfaces de host usadas pelos itens fornecidos.
triggerids	ID/array	Retorna apenas as interfaces de host usadas por itens nos triggers fornecidos.
selectItems	query	Retorna uma propriedade `items` com os itens que usam a interface. Suporta `count`.
selectHosts	query	Retorna uma propriedade `hosts` com um array de hosts que usam a interface.
limitSelects	integer	Limita o número de registros retornados por subconsultas. Aplica-se às seguintes subconsultas: `selectItems`.
sortfield	string/array	Ordena o resultado pelas propriedades fornecidas. Valores possíveis: `interfaceid`, `dns`, `ip`.
countOutput	boolean	Esses parâmetros são descritos no comentário de referência.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valores de retorno

`(integer/array)` Retorna:

- uma matriz de objetos;
- a contagem de objetos recuperados, se o parâmetro `countOutput` tiver sido usado.

Exemplos

Recuperar interfaces de host

Recupere todos os dados sobre as interfaces usadas pelo host "30057".

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "hostinterface.get",
    "params": {
        "output": "extend",
        "hostids": "30057"
    },
    "id": 1
}
```

Resposta:

```
{
    "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
}
```


Veja também

- Host
- Item

Fonte

CHostInterface::get() em *ui/include/classes/api/services/CHostInterface.php*.

hostinterface.massadd

Descrição

`object hostinterface.massadd(object parameters)`

Este método permite adicionar simultaneamente interfaces de host a vários hosts.

> **Note:**
> Este método está disponível apenas para os tipos de usuário *Admin* e *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Consulte Funções de usuário para mais informações.

Parâmetros

`(object)` Parâmetros contendo as interfaces de host a serem criadas nos hosts fornecidos.

O método aceita os seguintes parâmetros.

Parâmetro	Tipo	Descrição
interfaces	object/array	Interfaces de host a serem criadas nos hosts fornecidos. Comportamento do parâmetro: - *obrigatório*
hosts	object/array	Hosts a serem atualizados. Os hosts devem ter apenas a propriedade `hostid` definida. Comportamento do parâmetro: - *obrigatório*

Valores de retorno

`(object)` Retorna um objeto contendo os IDs do host criado interfaces sob a propriedade `interfaceids`.

Exemplos

Criando interfaces

Crie uma interface em dois hosts.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "hostinterface.massadd",
    "params": {
        "hosts": [
            {
                "hostid": "30050"
            },
            {
                "hostid": "30052"
            }
        ],
        "interfaces": {
            "dns": "",
            "ip": "127.0.0.1",
            "main": 0,
            "port": "10050",
            "type": 1,
            "useip": 1
        }
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "interfaceids": [
            "30069",
            "30070"
        ]
    },
    "id": 1
}
```

Veja também

- hostinterface.create
- host.massadd
- Host

Fonte

CHostInterface::massAdd() em *ui/include/classes/api/services/CHostInterface.php*.

hostinterface.massremove

Descrição

`object hostinterface.massremove(object parameters)`

Este método permite remover interfaces de host dos hosts fornecidos.

> **Note:**
> Este método está disponível apenas para os tipos de usuário *Admin* e *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object)` Parâmetros contendo os IDs dos hosts a serem atualizados e as interfaces a serem removidas.

Parâmetro	Tipo	Descrição
interfaces	object/array	Interfaces de host a serem removidas dos hosts fornecidos. O objeto de interface de host deve ter apenas as propriedades `ip`, `dns` e `port` definidas. Comportamento do parâmetro: - *obrigatório*
hostids	ID/array	IDs dos hosts a serem atualizados. Comportamento do parâmetro: - *obrigatório*

Valores de retorno

`(object)` Retorna um objeto contendo os IDs do host excluído interfaces sob a propriedade `interfaceids`.

Exemplos

Removendo interfaces

Remove a interface SNMP "127.0.0.1" de dois hosts.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "hostinterface.massremove",
    "params": {
        "hostids": [
            "30050",
            "30052"
        ],
        "interfaces": {
            "dns": "",
            "ip": "127.0.0.1",
            "port": "161"
        }
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "interfaceids": [
            "30069",
            "30070"
        ]
    },
    "id": 1
}
```

Veja também

- hostinterface.delete
- host.massremove

Fonte

CHostInterface::massRemove() em *ui/include/classes/api/services/CHostInterface.php*.

hostinterface.replacehostinterfaces

Descrição

`object hostinterface.replacehostinterfaces(object parameters)`

Este método permite substituir todas as interfaces de host em um determinado host.

> **Note:**
> Este método está disponível apenas para os tipos de usuário *Admin* e *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object)` Parâmetros contendo o ID do host a ser atualizado e as novas interfaces do host.

Parâmetro	Tipo	Descrição
interfaces	object/array	Interfaces do host para substituir as interfaces atuais do host. Comportamento do parâmetro: - *obrigatório*
hostid	ID	ID do host a ser atualizado. Comportamento do parâmetro: - *obrigatório*

Valores de retorno

`(object)` Retorna um objeto contendo os IDs do host criado interfaces sob a propriedade `interfaceids`.

Exemplos

Substituindo interfaces de host

Substitua todas as interfaces de host por uma única interface de agent.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "hostinterface.replacehostinterfaces",
    "params": {
        "hostid": "30052",
        "interfaces": {
            "dns": "",
            "ip": "127.0.0.1",
```

```
            "main": 1,
            "port": "10050",
            "type": 1,
            "useip": 1
        }
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "interfaceids": [
            "30081"
        ]
    },
    "id": 1
}
```

Veja também

- host.update
- host.massupdate

Fonte

CHostInterface::replaceHostInterfaces() em *ui/include/classes/api/services/CHostInterface.php*.

#####hostinterface.update {#manual-api-reference-hostinterface-update}

Descrição

`object hostinterface.update(object/array hostInterfaces)`

Este método permite atualizar interfaces de host existentes.

> **Note:**
> Este método está disponível apenas para os tipos de usuário *Admin* e *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função de usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object/array)` Propriedades da interface do host a serem atualizadas.

A propriedade `interfaceid` deve ser definida para cada interface do host, todas as outras propriedades são opcionais. Somente as propriedades fornecidas serão atualizadas, todas as outras permanecerão inalteradas.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs do host atualizado interfaces sob a propriedade `interfaceids`.

Exemplos

Alterando a porta da interface do host

Altere a porta de uma interface do host.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "hostinterface.update",
    "params": {
        "interfaceid": "30048",
        "port": "30050"
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "interfaceids": [
            "30048"
        ]
    },
    "id": 1
}
```

Fonte

CHostInterface::update() em *ui/include/classes/api/services/CHostInterface.php*.

Item

Esta classe foi projetada para trabalhar com items.

Referências de objetos:

- Item
  - Cabeçalho HTTP
  - Campo de consulta HTTP
- Tag de item
- Pré-processamento de item

Métodos disponíveis:

- item.create - criar novos items
- item.delete - excluir items
- item.get - recuperar items
- item.update - atualizar items

Objeto item

Os seguintes objetos estão diretamente relacionados à API de `item`.

Item

> **Note:**
> Itens web não podem ser criados, atualizados ou excluídos diretamente via API do Zabbix.

O objeto item possui as seguintes propriedades.

Propriedade	Tipo	Descrição
itemid	ID	ID do item. Comportamento da propriedade: - *somente leitura* - *obrigatório* para operações de atualização

Propriedade	Tipo	Descrição
delay	string	Intervalo de atualização do item. Aceita segundos ou unidade de tempo com sufixo (por exemplo, 30s, 1m, 2h, 1d) e, opcionalmente, um ou mais intervalos personalizados, todos separados por ponto e vírgula. Os intervalos personalizados podem ser uma mistura de intervalos flexíveis e de agendamento. Aceita macros de usuário. Se usado, o valor deve ser uma única macro. Múltiplas macros ou macros misturadas com texto não são suportadas. Intervalos flexíveis podem ser escritos como duas macros separadas por uma barra (por exemplo, `{$FLEX_INTERVAL}/{$FLEX_PERIOD}`). Exemplo: `1h;wd1-5h9-18;{$Macro1}/1-7,00:00-24:00;0/6-7,12:00-24:00;{$Macr` Comportamento da propriedade: - *obrigatório* se `type` for definido como ”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), ou se `type` for definido como ”Zabbix agent (active)” (7) e `key_` não contiver ”mqtt.get”
hostid	ID	ID do host ou template ao qual o item pertence. Comportamento da propriedade: - *constante* - *obrigatório* para operações de criação
interfaceid	ID	ID da interface do host do item. Comportamento da propriedade: - *obrigatório* se o item pertencer ao host e `type` for definido como ”Zabbix agent”, ”IPMI agent”, ”JMX agent”, ”SNMP trap” ou ”SNMP agent” - *suportado* se o item pertencer ao host e `type` for definido como ”Simple check”, ”External check”, ”SSH agent”, ”TELNET agent” ou ”HTTP agent” - *somente leitura* para objetos descobertos
key_	string	Chave do item. Comportamento da propriedade: - *obrigatório* para operações de criação - *somente leitura* para objetos herdados ou descobertos
name	string	Nome do item. Suporta macros de usuário. Comportamento da propriedade: - *obrigatório* para operações de criação - *somente leitura* para objetos herdados ou descobertos
name_resolved	string	Nome do item com macros de usuário resolvidas. Comportamento da propriedade: - *somente leitura*

Propriedade	Tipo	Descrição
type	integer	Tipo do item. Valores possíveis: 0 - Zabbix agent; 2 - Zabbix trapper; 3 - Simple check; 5 - Zabbix internal; 7 - Zabbix agent (active); 9 - Web item; 10 - External check; 11 - Database monitor; 12 - IPMI agent; 13 - SSH agent; 14 - TELNET agent; 15 - Calculated; 16 - JMX agent; 17 - SNMP trap; 18 - Dependent item; 19 - HTTP agent; 20 - SNMP agent; 21 - Script; 22 - Browser. Comportamento da propriedade: - *obrigatório* para operações de criação - *somente leitura* para objetos herdados ou descobertos
url	string	String de URL. Suporta macros de usuário, {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {ITEM.ID}, {ITEM.KEY}. Comportamento da propriedade: - *obrigatório* se `type` for definido como "HTTP agent" - *somente leitura* para objetos herdados ou descobertos
value_type	integer	Tipo de informação do item. Valores possíveis: 0 - numeric float; 1 - character; 2 - log; 3 - numeric unsigned; 4 - text; 5 - binary. Comportamento da propriedade: - *obrigatório* para operações de criação - *somente leitura* para objetos herdados ou descobertos
allow_traps	integer	Permitir popular valor similar ao item trapper. 0 - *(padrão)* Não permitir aceitar dados recebidos; 1 - Permitir aceitar dados recebidos. Comportamento da propriedade: - *suportado* se `type` for definido como "HTTP agent" - *somente leitura* para objetos descobertos

Propriedade	Tipo	Descrição
authtype	integer	Método de autenticação. Valores possíveis se `type` for definido como ”SSH agent”: 0 - *(padrão)* senha; 1 - chave pública. Valores possíveis se `type` for definido como ”HTTP agent”: 0 - *(padrão)* nenhum; 1 - básico; 2 - NTLM; 3 - Kerberos; 4 - Digest. Comportamento da propriedade: - *suportado* se `type` for definido como ”SSH agent” ou ”HTTP agent” - *somente leitura* para objetos herdados (se `type` for definido como ”HTTP agent”) ou objetos descobertos
description	string	Descrição do item. Comportamento da propriedade: - *somente leitura* para objetos descobertos
error	string	Texto de erro se houver problemas ao atualizar o valor do item. Comportamento da propriedade: - *somente leitura*
flags	integer	Origem do item. Valores possíveis: 0 - um item simples; 4 - um item descoberto. Comportamento da propriedade: - *somente leitura*
follow_redirects	integer	Seguir redirecionamentos de resposta ao coletar dados. Valores possíveis: 0 - Não seguir redirecionamentos; 1 - *(padrão)* Seguir redirecionamentos. Comportamento da propriedade: - *suportado* se `type` for definido como ”HTTP agent” - *somente leitura* para objetos herdados ou descobertos
headers	array	Array de cabeçalhos que serão enviados ao realizar uma requisição HTTP. Comportamento da propriedade: - *suportado* se `type` for definido como ”HTTP agent” - *somente leitura* para objetos herdados ou descobertos
history	string	Unidade de tempo de quanto tempo os dados de histórico devem ser armazenados. Também aceita macro de usuário. Padrão: 31d. Comportamento da propriedade: - *somente leitura* para objetos descobertos
http_proxy	string	String de conexão do proxy HTTP(S). Comportamento da propriedade: - *suportado* se `type` for definido como ”HTTP agent” - *somente leitura* para objetos herdados ou descobertos

Propriedade	Tipo	Descrição
inventory_link	integer	ID do campo de inventário do host que é populado pelo item. Consulte a página de inventário do host para uma lista de campos de inventário de host suportados e seus IDs. Padrão: 0. Comportamento da propriedade: - *suportado* se `value_type` for definido como ”numeric float”, ”character”, ”numeric unsigned” ou ”text” - *somente leitura* para objetos descobertos
ipmi_sensor	string	Sensor IPMI. Comportamento da propriedade: - *obrigatório* se `type` for definido como ”IPMI agent” e `key_` não for definido como ”ipmi.get” - *suportado* se `type` for definido como ”IPMI agent” - *somente leitura* para objetos herdados ou descobertos
jmx_endpoint	string	String de conexão personalizada do JMX agent. Valor padrão: service:jmx:rmi:///jndi/rmi://{HOST.CONN}:{HOST.PORT}/jmxrmi Comportamento da propriedade: - *suportado* se `type` for definido como ”JMX agent” - *somente leitura* para objetos descobertos
lastclock	timestamp	Hora em que o valor do item foi atualizado pela última vez. Por padrão, apenas valores que se enquadram nas últimas 24 horas são exibidos. Você pode estender esse período de tempo alterando o valor do parâmetro *Período máximo de exibição do histórico* na seção de menu *Administração → Geral*. Comportamento da propriedade: - *somente leitura*
lastns	integer	Nanosegundos em que o valor do item foi atualizado pela última vez. Por padrão, apenas valores que se enquadram nas últimas 24 horas são exibidos. Você pode estender esse período de tempo alterando o valor do parâmetro *Período máximo de exibição do histórico* na seção de menu *Administração → Geral*. Comportamento da propriedade: - *somente leitura*
lastvalue	string	Último valor do item. Por padrão, apenas valores que se enquadram nas últimas 24 horas são exibidos. Você pode estender esse período de tempo alterando o valor do parâmetro *Período máximo de exibição do histórico* na seção de menu *Administração → Geral*. Comportamento da propriedade: - *somente leitura*
logtimefmt	string	Formato da hora nas entradas de log. Comportamento da propriedade: - *suportado* se `value_type` for definido como ”log” - *somente leitura* para objetos herdados ou descobertos

Propriedade	Tipo	Descrição
master_itemid	ID	ID do item mestre. É permitida recursão de até 3 itens dependentes e contagem máxima de itens dependentes igual a 29999. Comportamento da propriedade: - *obrigatório* se `type` for definido como ”Dependent item” - *somente leitura* para objetos herdados ou descobertos
output_format	integer	Se a resposta deve ser convertida para JSON. 0 - *(padrão)* Armazenar bruto; 1 - Converter para JSON. Comportamento da propriedade: - *suportado* se `type` for definido como ”HTTP agent” - *somente leitura* para objetos herdados ou descobertos
params	string	Parâmetros adicionais dependendo do tipo do item: - script executado para itens SSH agent e TELNET agent; - consulta SQL para itens database monitor; - fórmula para itens calculated; - o script para itens script e browser. Comportamento da propriedade: - *obrigatório* se `type` for definido como ”Database monitor”, ”SSH agent”, ”TELNET agent”, ”Calculated”, ”Script” ou ”Browser” - *somente leitura* para objetos herdados (se `type` for definido como ”Script” ou ”Browser”) ou objetos descobertos
parameters	object/array	Parâmetros adicionais se `type` for definido como ”Script” ou ”Browser”. Array de objetos com propriedades `name` e `value`, onde `name` deve ser único. Comportamento da propriedade: - *suportado* se `type` for definido como ”Script” ou ”Browser” - *somente leitura* para objetos herdados ou descobertos
password	string	Senha para autenticação. Comportamento da propriedade: - *obrigatório* se `type` for definido como ”JMX agent” e `username` estiver definido - *suportado* se `type` for definido como ”Simple check”, ”SSH agent”, ”TELNET agent”, ”Database monitor” ou ”HTTP agent” - *somente leitura* para objetos herdados (se `type` for definido como ”HTTP agent”) ou objetos descobertos
post_type	integer	Tipo de corpo de dados postados armazenados na propriedade `posts`. Valores possíveis: 0 - *(padrão)* Dados brutos; 2 - Dados JSON; 3 - Dados XML. Comportamento da propriedade: - *suportado* se `type` for definido como ”HTTP agent” - *somente leitura* para objetos herdados ou descobertos
posts	string	Dados do corpo da requisição HTTP(S). Comportamento da propriedade: - *obrigatório* se `type` for definido como ”HTTP agent” e `post_type` for definido como ”JSON data” ou ”XML data” - *suportado* se `type` for definido como ”HTTP agent” e `post_type` for definido como ”Raw data” - *somente leitura* para objetos herdados ou descobertos

Propriedade	Tipo	Descrição
prevvalue	string	Valor anterior do item. Por padrão, apenas valores que se enquadram nas últimas 24 horas são exibidos. Você pode estender esse período de tempo alterando o valor do parâmetro *Período máximo de exibição do histórico* na seção de menu *Administração → Geral*. Comportamento da propriedade: - *somente leitura*
privatekey	string	Nome do arquivo de chave privada. Comportamento da propriedade: - *obrigatório* se `type` for definido como ”SSH agent” e `authtype` for definido como ”public key” - *somente leitura* para objetos descobertos
publickey	string	Nome do arquivo de chave pública. Comportamento da propriedade: - *obrigatório* se `type` for definido como ”SSH agent” e `authtype` for definido como ”public key” - *somente leitura* para objetos descobertos
query_fields	array	Array de campos de consulta que serão enviados ao realizar uma requisição HTTP. Comportamento da propriedade: - *suportado* se `type` for definido como ”HTTP agent” - *somente leitura* para objetos herdados ou descobertos
request_method	integer	Tipo de método de requisição. Valores possíveis: 0 - *(padrão)* GET; 1 - POST; 2 - PUT; 3 - HEAD. Comportamento da propriedade: - *suportado* se `type` for definido como ”HTTP agent” - *somente leitura* para objetos herdados ou descobertos
retrieve_mode	integer	Qual parte da resposta deve ser armazenada. Valores possíveis se `request_method` for definido como ”GET”, ”POST” ou ”PUT”: 0 - *(padrão)* Corpo; 1 - Cabeçalhos; 2 - Corpo e cabeçalhos serão armazenados. Valores possíveis se `request_method` for definido como ”HEAD”: 1 - Cabeçalhos. Comportamento da propriedade: - *suportado* se `type` for definido como ”HTTP agent” - *somente leitura* para objetos herdados ou descobertos
snmp_oid	string	SNMP OID. Comportamento da propriedade: - *obrigatório* se `type` for definido como ”SNMP agent” - *somente leitura* para objetos herdados ou descobertos
ssl_cert_file	string	Caminho do arquivo de chave pública SSL. Comportamento da propriedade: - *suportado* se `type` for definido como ”HTTP agent” - *somente leitura* para objetos herdados ou descobertos

Propriedade	Tipo	Descrição
ssl_key_file	string	Caminho do arquivo de chave privada SSL. Comportamento da propriedade: - *suportado* se `type` for definido como ”HTTP agent” - *somente leitura* para objetos herdados ou descobertos
ssl_key_password	string	Senha para o arquivo de chave SSL. Comportamento da propriedade: - *suportado* se `type` for definido como ”HTTP agent” - *somente leitura* para objetos herdados ou descobertos
state	integer	Estado do item. Valores possíveis: 0 - *(padrão)* normal; 1 - não suportado. Comportamento da propriedade: - *somente leitura*
status	integer	Status do item. Valores possíveis: 0 - *(padrão)* item habilitado; 1 - item desabilitado.
status_codes	string	Intervalos de códigos de status HTTP necessários, separados por vírgulas. Também suporta macros de usuário como parte da lista separada por vírgulas. Exemplo: 200,200-{$M},{$M},200-400 Comportamento da propriedade: - *suportado* se `type` for definido como ”HTTP agent” - *somente leitura* para objetos herdados ou descobertos
templateid	ID	ID do item do template pai. *Dica*: Use a propriedade `hostid` para especificar o template ao qual o item pertence. Comportamento da propriedade: - *somente leitura*
timeout	string	Tempo limite da requisição de coleta de dados do item. Aceita segundos ou unidade de tempo com sufixo (por exemplo, 30s, 1m). Também aceita macros de usuário. Faixa de valores possíveis: 1-600s. Padrão: ”” - usar configurações do proxy/globais. Comportamento da propriedade: - *suportado* se `type` for definido como ”Zabbix agent” (0), ”Simple check” (3) e `key_` não começar com ”`vmware.`” e ”`icmpping`”, ”Zabbix agent (active)” (7), ”External check” (10), ”Database monitor” (11), ”SSH agent” (13), ”TELNET agent” (14), ”HTTP agent” (19), ”SNMP agent” (20) e `snmp_oid` começar com ”`walk[`” ou ”`get[`”, ”Script” (21), ”Browser” (22) - *somente leitura* para objetos herdados e descobertos

Propriedade	Tipo	Descrição
trapper_hosts	string	Hosts permitidos. Comportamento da propriedade: - *somente leitura* para objetos descobertos - *suportado* se `type` for definido como ”Zabbix trapper”, ou se `type` for definido como ”HTTP agent” e `allow_traps` for definido como ”Permitir aceitar dados recebidos”
trends	string	Unidade de tempo de quanto tempo os dados de tendências devem ser armazenados. Também aceita macro de usuário. Padrão: 365d. Comportamento da propriedade: - *suportado* se `value_type` for definido como ”numeric float” ou ”numeric unsigned” - *somente leitura* para objetos descobertos
units	string	Unidades do valor. Comportamento da propriedade: - *suportado* se `value_type` for definido como ”numeric float” ou ”numeric unsigned” - *somente leitura* para objetos herdados ou descobertos
username	string	Nome de usuário para autenticação. Comportamento da propriedade: - *obrigatório* se `type` for definido como ”SSH agent”, ”TELNET agent”, ou se `type` for definido como ”JMX agent” e `password` estiver definido - *suportado* se `type` for definido como ”Simple check”, ”Database monitor” ou ”HTTP agent” - *somente leitura* para objetos herdados (se `type` for definido como ”HTTP agent”) ou objetos descobertos
uuid	string	Identificador universal único, usado para vincular o item importado a itens já existentes. Gerado automaticamente, se não fornecido. Comportamento da propriedade: - *suportado* se o item pertencer a um template
valuemapid	ID	ID do mapa de valores associado. Comportamento da propriedade: - *suportado* se `value_type` for definido como ”numeric float”, ”character” ou ”numeric unsigned” - *somente leitura* para objetos herdados ou descobertos
verify_host	integer	Se deve validar que o nome do host para a conexão corresponde ao do certificado do host. Valores possíveis: 0 - *(padrão)* Não validar; 1 - Validar. Comportamento da propriedade: - *suportado* se `type` for definido como ”HTTP agent” - *somente leitura* para objetos herdados ou descobertos

Propriedade	Tipo	Descrição
verify_peer	integer	Se deve validar que o certificado do host é autêntico. Valores possíveis: 0 - *(padrão)* Não validar; 1 - Validar. Comportamento da propriedade: - *suportado* se `type` for definido como "HTTP agent" - *somente leitura* para objetos herdados ou descobertos

Cabeçalho HTTP

O objeto header possui as seguintes propriedades:

Propriedade	Tipo	Descrição
name	string	Nome do cabeçalho HTTP. Comportamento da propriedade: - *obrigatório*
value	string	Valor do cabeçalho. Comportamento da propriedade: - *obrigatório*

Campo de consulta HTTP

O objeto de campo de consulta define um nome e valor que é usado para especificar um parâmetro de URL. Ele possui as seguintes propriedades:

Propriedade	Tipo	Descrição
name	string	Nome do parâmetro. Comportamento da propriedade: - *obrigatório*
value	string	Valor do parâmetro. Comportamento da propriedade: - *obrigatório*

Tag do item

O objeto de tag do item possui as seguintes propriedades.

Propriedade	Tipo	Descrição
tag	string	Nome da tag do item. Comportamento da propriedade: - *obrigatório*
value	string	Valor da tag do item.

Pré-processamento de item

O objeto de pré-processamento de item possui as seguintes propriedades.

Propriedade	Tipo	Descrição
type	integer	O tipo de opção de pré-processamento. Valores possíveis: 1 - Multiplicador personalizado; 2 - Remover à direita; 3 - Remover à esquerda; 4 - Remover; 5 - Expressão regular; 6 - Booleano para decimal; 7 - Octal para decimal; 8 - Hexadecimal para decimal; 9 - Mudança simples; 10 - Mudança por segundo; 11 - XML XPath; 12 - JSONPath; 13 - Dentro do intervalo; 14 - Corresponde à expressão regular; 15 - Não corresponde à expressão regular; 16 - Verificar erro em JSON; 17 - Verificar erro em XML; 18 - Verificar erro usando expressão regular; 19 - Descartar inalterado; 20 - Descartar inalterado com heartbeat; 21 - JavaScript; 22 - Padrão Prometheus; 23 - Prometheus para JSON; 24 - CSV para JSON; 25 - Substituir; 26 - Verificar não suportado; 27 - XML para JSON; 28 - Valor SNMP walk; 29 - SNMP walk para JSON; 30 - Valor SNMP get. Comportamento da propriedade: - *obrigatório*
params	string	Parâmetros adicionais usados pela opção de pré-processamento. Múltiplos parâmetros são separados pelo caractere de nova linha (\n). Se `type` estiver definido como ”Verificar não suportado”, os parâmetros seguem a sintaxe `<scope>[\n<pattern>]`, onde *pattern* é uma expressão regular e *scope* é um dos: -1 - corresponder a qualquer erro; 0 - verificar se a mensagem de erro corresponde ao *pattern*; 1 - verificar se a mensagem de erro não corresponde ao *pattern*. Comportamento da propriedade: - *obrigatório* se `type` estiver definido como ”Multiplicador personalizado” (1), ”Remover à direita” (2), ”Remover à esquerda” (3), ”Remover” (4), ”Expressão regular” (5), ”XML XPath” (11), ”JSONPath” (12), ”Dentro do intervalo” (13), ”Corresponde à expressão regular” (14), ”Não corresponde à expressão regular” (15), ”Verificar erro em JSON” (16), ”Verificar erro em XML” (17), ”Verificar erro usando expressão regular” (18), ”Descartar inalterado com heartbeat” (20), ”JavaScript” (21), ”Padrão Prometheus” (22), ”Prometheus para JSON” (23), ”CSV para JSON” (24), ”Substituir” (25), ”Verificar não suportado” (26), ”Valor SNMP walk” (28), ”SNMP walk para JSON” (29) ou ”Valor SNMP get” (30)

Propriedade	Tipo	Descrição
error_handler	integer	Tipo de ação usada em caso de falha na etapa de pré-processamento. Valores possíveis: 0 - Mensagem de erro definida pelo Zabbix server; 1 - Descartar valor; 2 - Definir valor personalizado; 3 - Definir mensagem de erro personalizada. Valores possíveis se `type` estiver definido como ”Verificar não suportado”: 1 - Descartar valor; 2 - Definir valor personalizado; 3 - Definir mensagem de erro personalizada. Comportamento da propriedade: - *obrigatório* se `type` estiver definido como ”Multiplicador personalizado” (1), ”Expressão regular” (5), ”Booleano para decimal” (6), ”Octal para decimal” (7), ”Hexadecimal para decimal” (8), ”Mudança simples” (9), ”Mudança por segundo” (10), ”XML XPath” (11), ”JSONPath” (12), ”Dentro do intervalo” (13), ”Corresponde à expressão regular” (14), ”Não corresponde à expressão regular” (15), ”Verificar erro em JSON” (16), ”Verificar erro em XML” (17), ”Verificar erro usando expressão regular” (18), ”Padrão Prometheus” (22), ”Prometheus para JSON” (23), ”CSV para JSON” (24), ”Verificar não suportado” (26), ”XML para JSON” (27), ”Valor SNMP walk” (28), ”SNMP walk para JSON” (29) ou ”Valor SNMP get” (30)
error_handler_params	string	Parâmetros do manipulador de erro. Comportamento da propriedade: - *obrigatório* se `error_handler` estiver definido como ”Definir valor personalizado” ou ”Definir mensagem de erro personalizada”

Os seguintes parâmetros e manipuladores de erro são suportados para cada tipo de pré-processamento.

Tipo de pré-processamento	Nome	Parâmetro 1	Parâmetro 2	Parâmetro 3	Manipuladores de erro suportados
1	Multiplicador personalizado	número^1, 6^			0, 1, 2, 3
2	Remover à direita	lista de caracteres[2]			
3	Remover à esquerda	lista de caracteres[2]			
4	Remover	lista de caracteres[2]			
5	Expressão regular	padrão[3]	saída[2]		0, 1, 2, 3
6	Booleano para decimal				0, 1, 2, 3
7	Octal para decimal				0, 1, 2, 3

Tipo de pré-processamento	Nome	Parâmetro 1	Parâmetro 2	Parâmetro 3	Manipuladores de erro suportados
8	Hexadecimal para decimal				0, 1, 2, 3
9	Mudança simples				0, 1, 2, 3
10	Mudança por segundo				0, 1, 2, 3
11	XML XPath	caminho[4]			0, 1, 2, 3
12	JSONPath	caminho[4]			0, 1, 2, 3
13	Dentro do intervalo	min^1, 6^	max^1, 6^		0, 1, 2, 3
14	Corresponde à expressão regular	padrão[3]			0, 1, 2, 3
15	Não corresponde à expressão regular	padrão[3]			0, 1, 2, 3
16	Verificar erro em JSON	caminho[4]			0, 1, 2, 3
17	Verificar erro em XML	caminho[4]			0, 1, 2, 3
18	Verificar erro usando expressão regular	padrão[3]	saída[2]		0, 1, 2, 3
19	Descartar inalterado				
20	Descartar inalterado com heartbeat	segundos^5, 6^			
21	JavaScript	script[2]			
22	Padrão Prometheus	padrão^6, 7^	`value`, `label`, `function`	saída^8, 9^	0, 1, 2, 3

Tipo de pré-processamento	Nome	Parâmetro 1	Parâmetro 2	Parâmetro 3	Manipuladores de erro suportados
23	Prometheus para JSON	padrão^6, 7^			0, 1, 2, 3
24	CSV para JSON	caractere[2]	caractere[2]	0,1	0, 1, 2, 3
25	Substituir	string de busca[2]	substituição[2]		
26	Verificar não suportado	escopo[1]	padrão^3, 6^		1, 2, 3
27	XML para JSON				0, 1, 2, 3
28	Valor SNMP walk	OID[2]	Formato: 0 - Inalterado 1 - UTF-8 de Hex-STRING 2 - MAC de Hex-STRING 3 - Inteiro de BITS		0, 1, 2, 3
29	SNMP walk para JSON[10]	Nome do campo[2]	Prefixo OID[2]	Formato: 0 - Inalterado 1 - UTF-8 de Hex-STRING 2 - MAC de Hex-STRING 3 - Inteiro de BITS	0, 1, 2, 3
30	Valor SNMP get	Formato: 1 - UTF-8 de Hex-STRING 2 - MAC de Hex-STRING 3 - Inteiro de BITS			0, 1, 2, 3

[1] número de ponto flutuante (inteiros são convertidos implicitamente para valores float)
 [2] string
 [3] expressão regular
 [4] JSONPath ou XML XPath
 [5] inteiro positivo (com suporte a sufixos de tempo, por exemplo, 30s, 1m, 2h, 1d)
 [6] macro de usuário
 [7] padrão Prometheus seguindo a sintaxe: `<nome da métrica>{<nome do rótulo>="<valor do rótulo>", ...} == <valor>`. Cada componente do padrão Prometheus (métrica, nome do rótulo, valor do rótulo e valor da métrica) pode ser macro de usuário.
 [8] saída Prometheus seguindo a sintaxe: `<nome do rótulo>` (pode ser uma macro de usuário) se `label` for selecionado como o segundo parâmetro.
 [9] Uma das funções de agregação: `sum`, `min`, `max`, `avg`, `count` se `function` for selecionado como o segundo parâmetro.
 [10] Suporta múltiplos registros "Nome do campo,Prefixo OID,Formato" delimitados por um caractere de nova linha.

item.create

Descrição

`object item.create(object/array items)`

Este método permite criar novos items.

> **Note:**
> Items web não podem ser criados via API do Zabbix.

Note:
Este método está disponível apenas para os tipos de usuário *Admin* e *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função de usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object/array)` Items a serem criados.

Além das propriedades padrão do item, o método aceita os seguintes parâmetros.

Parâmetro	Tipo	Descrição
preprocessing	array	Opções de pré-processamento do item.
tags	array	Tags do item.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos items criados na propriedade `itemids`. A ordem dos IDs retornados corresponde à ordem dos items passados.

Exemplos

Criando um item

Crie um item numérico do agent Zabbix com 2 tags de item para monitorar o espaço livre em disco no host com ID "30074".

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "item.create",
    "params": {
        "name": "Espaço livre em disco em /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
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "24758"
        ]
    },
    "id": 1
}
```

Criando um item de inventário de host

Crie um item de agent Zabbix para preencher o campo de inventário "OS" do host.

Requisição:

```
{
    "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
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "24759"
        ]
    },
    "id": 1
}
```

Criando um item com pré-processamento

Crie um item usando multiplicador personalizado.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "item.create",
    "params": {
        "name": "Device uptime",
        "key_": "sysUpTime",
        "hostid": "10084",
        "type": 20,
        "snmp_oid": "SNMPv2-MIB::sysUpTime.0",
        "value_type": 1,
        "delay": "60s",
        "interfaceid": "83",
        "preprocessing": [
            {
                "type": 1,
                "params": "0.01",
                "error_handler": 1,
                "error_handler_params": ""
            }
        ]
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "44210"
```

```
        ]
    },
    "id": 1
}
```

Criando um item dependente

Crie um item dependente para o item mestre com ID 24759. Somente dependências no mesmo host são permitidas, portanto o item mestre e o dependente devem ter o mesmo hostid.

Requisição:

```
{
    "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
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "44211"
        ]
    },
    "id": 1
}
```

Criar item de agente HTTP

Crie um item de método de requisição POST com pré-processamento de resposta JSON.

Requisição:

```
{
    "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": "Exemplo de agente HTTP JSON",
        "value_type": 0,
        "output_format": 1,
        "preprocessing": [
            {
                "type": 12,
                "params": "$.random",
                "error_handler": 0,
                "error_handler_params": ""
            }
        ]
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "23865"
        ]
    },
    "id": 1
}
```

Criar item de script

Crie uma coleta de dados simples usando um item de script.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "item.create",
    "params": {
        "name": "Exemplo de script",
        "key_": "custom.script.item",
        "hostid": "12345",
        "type": 21,
        "value_type": 4,
        "params": "var request = new HttpRequest();\nreturn request.post(\"https://postman-echo.com/post\"
        "parameters": [
            {
                "name": "host",
                "value": "{HOST.CONN}"
            }
        ],
        "timeout": "6s",
        "delay": "30s"
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "23865"
        ]
    },
    "id": 1
```

```
}
```

Fonte

CItem::create() in *ui/include/classes/api/services/CItem.php*.

item.delete

Descrição

`object item.delete(array itemIds)`

Este método permite excluir items.

> **Note:**
> Items web não podem ser excluídos via API do Zabbix.

> **Note:**
> Este método está disponível apenas para os tipos de usuário *Admin* e *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função de usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(array)` IDs dos itens a serem excluídos.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos itens excluídos sob a propriedade `itemids`.

Exemplos

Excluindo múltiplos items

Exclua dois items.
 Items dependentes e protótipos de item são removidos automaticamente se o item mestre for excluído.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "item.delete",
    "params": [
        "22982",
        "22986"
    ],
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "22982",
            "22986"
        ]
    },
    "id": 1
}
```

Fonte

CItem::delete() em *ui/include/classes/api/services/CItem.php*.

item.get

Descrição

`integer/array item.get(object parameters)`

O método permite recuperar items de acordo com os parâmetros fornecidos.

> **Note:**
> Este método está disponível para usuários de qualquer tipo. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object)` Parâmetros que definem a saída desejada.

O método suporta os seguintes parâmetros.

Parâmetro	Tipo	Descrição
itemids	ID/array	Retorna apenas os items com os IDs fornecidos.
groupids	ID/array	Retorna apenas os items que pertencem aos hosts dos grupos fornecidos.
templateids	ID/array	Retorna apenas os items que pertencem aos templates fornecidos.
hostids	ID/array	Retorna apenas os items que pertencem aos hosts fornecidos.
proxyids	ID/array	Retorna apenas os items que são monitorados pelos proxies fornecidos.
interfaceids	ID/array	Retorna apenas os items que usam as interfaces de host fornecidas.
graphids	ID/array	Retorna apenas os items que são usados nos gráficos fornecidos.
triggerids	ID/array	Retorna apenas os items que são usados nos triggers fornecidos.
webitems	flag	Inclui items web no resultado.
inherited	boolean	Se definido como `true`, retorna apenas os items herdados de um template.
templated	boolean	Se definido como `true`, retorna apenas os items que pertencem a templates.
monitored	boolean	Se definido como `true`, retorna apenas os items habilitados que pertencem a hosts monitorados.
group	string	Retorna apenas os items que pertencem a um grupo com o nome fornecido.
host	string	Retorna apenas os items que pertencem a um host com o nome fornecido.
evaltype	integer	Método de avaliação da tag. Valores possíveis: 0 - *(padrão)* E/OU; 2 - OU.
tags	array	Retorna apenas os items com as tags fornecidas. Formato: `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`. Um array vazio retorna todos os items. Valores possíveis para operator: 0 - *(padrão)* Contém; 1 - Igual; 2 - Não contém; 3 - Diferente; 4 - Existe; 5 - Não existe.
with_triggers	boolean	Se definido como `true`, retorna apenas os items que são usados em triggers.
selectHosts	query	Retorna uma propriedade `hosts` com um array de hosts aos quais o item pertence.
selectInterfaces	query	Retorna uma propriedade `interfaces` com um array de interfaces de host usadas pelo item.
selectTriggers	query	Retorna uma propriedade `triggers` com os triggers nos quais o item é usado. Suporta `count`.

Parâmetro	Tipo	Descrição
selectGraphs	query	Retorna uma propriedade `graphs` com os gráficos que contêm o item. Suporta `count`.
selectDiscoveryRule	query	Retorna uma propriedade `discoveryRule` com a regra LLD que criou o item.
selectItemDiscovery	query	Retorna uma propriedade `itemDiscovery` com o objeto de descoberta de item. O objeto de descoberta de item vincula o item ao protótipo de item a partir do qual foi criado. Possui as seguintes propriedades: `itemdiscoveryid` - `(string)` ID da descoberta de item; `itemid` - `(string)` ID do item descoberto; `parent_itemid` - `(string)` ID do protótipo de item a partir do qual o item foi criado; `key_` - `(string)` chave do protótipo de item; `lastcheck` - `(timestamp)` hora em que o item foi descoberto pela última vez; `status` - `(int)` status da descoberta de item: 0 - *(padrão)* item descoberto, 1 - item não é mais descoberto; `ts_delete` - `(timestamp)` hora em que um item que não é mais descoberto será excluído; `ts_disable` - `(timestamp)` hora em que um item que não é mais descoberto será desabilitado; `disable_source` - `(int)` indicador se o item foi desabilitado por uma regra LLD ou manualmente: 0 - *(padrão)* desabilitado automaticamente, 1 - desabilitado por uma regra LLD.
selectPreprocessing	query	Retorna uma propriedade `preprocessing` com as opções de pré-processamento do item.
selectTags	query	Retorna as tags do item na propriedade `tags`.
selectValueMap	query	Retorna uma propriedade `valuemap` com o mapa de valores do item.
filter	object	Retorna apenas os resultados que correspondem exatamente ao filtro fornecido. Aceita um objeto, onde as chaves são nomes de propriedades e os valores são um único valor ou um array de valores para correspondência. Não suporta propriedades do tipo de dado `text` data type. Suporta propriedades adicionais: `host` - nome técnico do host ao qual o item pertence.
limitSelects	integer	Limita o número de registros retornados por subconsultas. Aplica-se às seguintes subconsultas: `selectGraphs` - os resultados serão ordenados por `name`; `selectTriggers` - os resultados serão ordenados por `description`.
sortfield	string/array	Ordena o resultado pelas propriedades fornecidas. Valores possíveis: `itemid`, `name`, `key_`, `delay`, `history`, `trends`, `type`, `status`.
countOutput	boolean	Esses parâmetros são descritos no comentário de referência.
editable	boolean	
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	

Parâmetro	Tipo	Descrição
sortorder	string/array	
startSearch	boolean	

Valores de retorno

`(integer/array)` Retorna:

- uma matriz de objetos;
- a contagem de objetos recuperados, se o parâmetro `countOutput` tiver sido usado.

Exemplos

Encontrando items por chave

Recupere todos os items usados em triggers para um ID de host específico que tenham a palavra ”system.cpu” na chave do item e ordene os resultados pelo nome.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "item.get",
    "params": {
        "output": "extend",
        "hostids": "10084",
        "with_triggers": true,
        "search": {
            "key_": "system.cpu"
        },
        "sortfield": "name"
    },
    "id": 1
}
```

Resposta:

```
{
    "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": "%",
            "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",
            "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": "",
            "state": "0",
            "error": "",
            "parameters": [],
            "lastclock": "0",
            "lastns": "0",
            "lastvalue": "0",
            "prevvalue": "0",
            "name_resolved": "CPU utilization"
        },
        {
            "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": "",
            "logtimefmt": "",
            "templateid": "42219",
            "valuemapid": "0",
            "params": "",
            "ipmi_sensor": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "flags": "0",
            "interfaceid": "1",
            "description": "",
            "inventory_link": "0",
            "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": "",
            "state": "0",
            "error": "",
            "parameters": [],
            "lastclock": "0",
            "lastns": "0",
            "lastvalue": "0",
            "prevvalue": "0",
            "name_resolved": "Load average (15m avg)"
        },
        {
            "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": "",
            "logtimefmt": "",
            "templateid": "42209",
            "valuemapid": "0",
            "params": "",
            "ipmi_sensor": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "flags": "0",
            "interfaceid": "1",
            "description": "",
            "inventory_link": "0",
            "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": "",
            "state": "0",
            "error": "",
            "parameters": [],
            "lastclock": "0",
            "lastns": "0",
            "lastvalue": "0",
            "prevvalue": "0",
            "name_resolved": "Load average (1m avg)"
        },
        {
            "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": "",
            "logtimefmt": "",
            "templateid": "42217",
            "valuemapid": "0",
            "params": "",
            "ipmi_sensor": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "flags": "0",
            "interfaceid": "1",
            "description": "",
            "inventory_link": "0",
            "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": "",
            "state": "0",
            "error": "",
            "parameters": [],
            "lastclock": "0",
            "lastns": "0",
            "lastvalue": "0",
            "prevvalue": "0",
            "name_resolved": "Load average (5m avg)"
        },
        {
            "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": "",
            "logtimefmt": "",
            "templateid": "42220",
            "valuemapid": "0",
            "params": "",
            "ipmi_sensor": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "flags": "0",
            "interfaceid": "1",
            "description": "",
            "inventory_link": "0",
            "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": "",
            "state": "0",
            "error": "",
            "parameters": [],
            "lastclock": "0",
            "lastns": "0",
            "lastvalue": "0",
            "prevvalue": "0",
            "name_resolved": "Number of CPUs"
        }
    ],
    "id": 1
}
```


Encontrando itens dependentes por chave

Recupere todos os itens dependentes do host com ID ”10116” que possuem a palavra ”apache” na chave.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "item.get",
    "params": {
        "output": "extend",
        "hostids": "10116",
        "search": {
            "key_": "apache"
        },
        "filter": {
            "type": 18
        }
    },
    "id": 1
}
```

Resposta:


```
{
    "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": "",
            "logtimefmt": "",
            "templateid": "0",
            "valuemapid": "0",
            "params": "",
            "ipmi_sensor": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "flags": "0",
            "interfaceid": "0",
            "description": "",
            "inventory_link": "0",
            "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": "",
            "state": "0",
            "error": "",
            "parameters": [],
            "lastclock": "0",
            "lastns": "0",
            "lastvalue": "0",
            "prevvalue": "0",
            "name_resolved": "Days"
        },
        {
            "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": "",
            "logtimefmt": "",
            "templateid": "0",
```

```
            "valuemapid": "0",
            "params": "",
            "ipmi_sensor": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "flags": "0",
            "interfaceid": "0",
            "description": "",
            "inventory_link": "0",
            "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": "",
            "state": "0",
            "error": "",
            "parameters": [],
            "lastclock": "0",
            "lastns": "0",
            "lastvalue": "0",
            "prevvalue": "0",
            "name_resolved": "Hours"
        }
    ],
    "id": 1
}
```


Encontrar item de agente HTTP

Encontre o item de agente HTTP com tipo de corpo post XML para um ID de host específico.

Requisição:


```
{
    "jsonrpc": "2.0",
    "method": "item.get",
    "params": {
        "hostids": "10255",
        "filter": {
            "type": 19,
            "post_type": 3
        }
    },
    "id": 1
```

```
}
```

Resposta:


```
{
    "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": "",
            "logtimefmt": "",
            "templateid": "0",
            "valuemapid": "0",
            "params": "",
            "ipmi_sensor": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "flags": "0",
            "interfaceid": "0",
            "description": "",
            "inventory_link": "0",
            "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": "",
            "state": "0",
```

```
            "error": "",
            "parameters": [],
            "lastclock": "0",
            "lastns": "0",
            "lastvalue": "",
            "prevvalue": "",
            "name_resolved": "template item"
        }
    ],
    "id": 1
}
```

Recuperando items com regras de pré-processamento

Recupere todos os items e suas regras de pré-processamento para um ID de host específico.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "item.get",
    "params": {
        "output": ["itemid", "name", "key_"],
        "selectPreprocessing": "extend",
        "hostids": "10254"
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemid": "23865",
        "name": "HTTP agent example JSON",
        "key_": "json",
        "preprocessing": [
            {
                "type": "12",
                "params": "$.random",
                "error_handler": "1",
                "error_handler_params": ""
            }
        ]
    },
    "id": 1
}
```

Veja também

- Regra de descoberta
- Gráfico
- Host
- Interface de host
- Trigger

Fonte

CItem::get() em *ui/include/classes/api/services/CItem.php*.

item.update

Descrição

`object item.update(object/array items)`

Este método permite atualizar items existentes.

> **Note:**
> Items web não podem ser atualizados via API do Zabbix.

> **Note:**
> Este método está disponível apenas para os tipos de usuário *Admin* e *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object/array)` Propriedades do item a serem atualizadas.

A propriedade `itemid` deve ser definida para cada item, todas as outras propriedades são opcionais. Somente as propriedades passadas serão atualizadas, todas as outras permanecerão inalteradas.

Além das propriedades padrão do item, o método aceita os seguintes parâmetros.

Parâmetro	Tipo	Descrição
preprocessing	array	Opções de pré-processamento do item para substituir as opções de pré-processamento atuais. Comportamento do parâmetro: - *somente leitura* para objetos herdados ou objetos descobertos
tags	array	Tags do item. Comportamento do parâmetro: - *somente leitura* para objetos descobertos

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos itens atualizados sob a propriedade `itemids`.

Exemplos

Habilitando um item

Habilite um item, ou seja, defina seu status como "0".

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "item.update",
    "params": {
        "itemid": "10092",
        "status": 0
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "10092"
        ]
    },
    "id": 1
}
```

Atualizar item dependente

Atualize o nome do item dependente e o ID do item mestre. Apenas dependências no mesmo host são permitidas, portanto, o item mestre e o item dependente devem ter o mesmo hostid.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "item.update",
    "params": {
        "name": "Dependent item updated name",
        "master_itemid": "25562",
        "itemid": "189019"
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "189019"
        ]
    },
    "id": 1
}
```

Atualizar item HTTP agent

Habilitar o recebimento de valores do item.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "item.update",
    "params": {
        "itemid": "23856",
        "allow_traps": 1
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "23856"
        ]
    },
    "id": 1
}
```

Atualizando um item com pré-processamento

Atualize um item com a regra de pré-processamento de item ”No intervalo”.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "item.update",
    "params": {
        "itemid": "23856",
        "preprocessing": [
            {
                "type": 13,
                "params": "\n100",
                "error_handler": 1,
```

```
                "error_handler_params": ""
            }
        ]
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "23856"
        ]
    },
    "id": 1
}
```

Atualizando um item de script

Atualize um item de script com um script diferente e remova parâmetros desnecessários que foram usados pelo script anterior.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "item.update",
    "params": {
        "itemid": "23865",
        "parameters": [],
        "params": "Zabbix.log(3, 'Log test');\nreturn 1;"
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "23865"
        ]
    },
    "id": 1
}
```

Fonte

CItem::update() em *ui/include/classes/api/services/CItem.php*.

Item de gráfico

Esta classe é projetada para trabalhar com itens de gráfico.

Referências de objeto:

- Item de gráfico

Métodos disponíveis:

- graphitem.get - recupera itens de gráfico

Objeto de item de gráfico

Os seguintes objetos estão diretamente relacionados à API `graphitem`.

Item de gráfico

> **Note:**
> Os itens de gráfico só podem ser modificados via a API `graph`.

O objeto de item de gráfico possui as seguintes propriedades.

Propriedade	Tipo	Descrição
gitemid	ID	ID do item de gráfico. Comportamento da propriedade: - *somente leitura*
color	string	Cor de desenho do item de gráfico como um código de cor hexadecimal. Comportamento da propriedade: - *obrigatório* para operações de criação
itemid	ID	ID do item. Comportamento da propriedade: - *obrigatório* para operações de criação
calc_fnc	integer	Valor do item que será exibido. Valores possíveis: 1 - valor mínimo; 2 - *(padrão)* valor médio; 4 - valor máximo; 7 - todos os valores; 9 - último valor, usado apenas por gráficos de pizza e gráficos explodidos.
drawtype	integer	Estilo de desenho do item de gráfico. Valores possíveis: 0 - *(padrão)* linha; 1 - região preenchida; 2 - linha em negrito; 3 - ponto; 4 - linha tracejada; 5 - linha em gradiente.
graphid	ID	ID do gráfico ao qual o item de gráfico pertence.
sortorder	integer	Posição do item no gráfico. Padrão: começa com ”0” e aumenta em um a cada entrada.
type	integer	Tipo de item de gráfico. Valores possíveis: 0 - *(padrão)* simples; 2 - soma do gráfico, usado apenas por gráficos de pizza e gráficos explodidos.
yaxisside	integer	Lado do gráfico onde a escala Y do item de gráfico será desenhada. Valores possíveis: 0 - *(padrão)* lado esquerdo; 1 - lado direito.

graphitem.get

Descrição

`integer/array graphitem.get(object parameters)`

O método permite recuperar itens de gráficos de acordo com os parâmetros fornecidos.

> **Note:**
> Este método está disponível para usuários de qualquer tipo. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object)` Parâmetros que definem a saída desejada.

O método suporta os seguintes parâmetros.

Parâmetro	Tipo	Descrição
graphids	ID/array	Retorna apenas os itens de gráfico que pertencem aos gráficos fornecidos.
itemids	ID/array	Retorna apenas os itens de gráfico com os IDs de item fornecidos.
type	integer	Retorna apenas os itens de gráfico com o tipo fornecido. Consulte a página do objeto de item de gráfico para obter uma lista dos tipos de itens de gráfico suportados.
selectGraphs	query	Retorna uma propriedade `graphs` com um array de gráficos aos quais o item pertence.
sortfield	string/array	Ordena o resultado pelas propriedades fornecidas. Valores possíveis: `gitemid`.
countOutput	boolean	Esses parâmetros são descritos no comentário de referência.
editable	boolean	
limit	integer	
output	query	
preservekeys	boolean	
sortorder	string/array	

Valores de retorno

`(integer/array)` Retorna:

- uma matriz de objetos;
- a contagem de objetos recuperados, se o parâmetro `countOutput` tiver sido usado.

Exemplos

Recuperando itens de gráfico de um gráfico

Recupere todos os itens de gráfico usados em um gráfico com informações adicionais sobre o item e o host.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "graphitem.get",
    "params": {
        "output": "extend",
        "graphids": "387"
    },
    "id": 1
}
```

Resposta:

```
{
    "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
}
```


Veja também

- Gráfico

Fonte

CGraphItem::get() em *ui/include/classes/api/services/CGraphItem.php*.

Log de auditoria

Esta classe é projetada para trabalhar com o log de auditoria.

Referências de objetos:

- Log de auditoria

Métodos disponíveis:

- auditlog.get - recupera registros do log de auditoria

Objeto de log de auditoria

Os seguintes objetos estão diretamente relacionados à API `auditlog`.

Log de auditoria

O objeto de log de auditoria contém informações sobre as ações do usuário. Ele possui as seguintes propriedades.

Propriedade	Tipo	Descrição
auditid	ID	ID da entrada do log de auditoria. Gerado usando o algoritmo CUID.
userid	ID	ID do usuário autor da entrada do log de auditoria.
username	string	Nome de usuário do autor da entrada do log de auditoria.
clock	timestamp	Timestamp de criação da entrada do log de auditoria.
ip	string	Endereço IP do autor da entrada do log de auditoria.

Propriedade	Tipo	Descrição
action	integer	Ação da entrada do log de auditoria. Valores possíveis: 0 - Adicionar; 1 - Atualizar; 2 - Excluir; 4 - Logout; 7 - Executar; 8 - Login; 9 - Falha no login; 10 - Limpar histórico; 11 - Atualizar configuração; 12 - Push.
resourcetype	integer	Tipo de recurso da entrada do log de auditoria. Valores possíveis: 0 - Usuário; 3 - Tipo de mídia; 4 - Host; 5 - Ação; 6 - Gráfico; 11 - Grupo de usuários; 13 - Trigger; 14 - Grupo de hosts; 15 - Item; 16 - Imagem; 17 - Mapa de valores; 18 - Serviço; 19 - Mapa; 22 - Cenário web; 23 - Regra de descoberta; 25 - Script; 26 - Proxy; 27 - Manutenção; 28 - Expressão regular; 29 - Macro; 30 - Template; 31 - Protótipo de trigger; 32 - Mapeamento de ícone; 33 - Dashboard; 34 - Correlação de eventos; 35 - Protótipo de gráfico; 36 - Protótipo de item; 37 - Protótipo de host; 38 - Autoregistro; 39 - Módulo; 40 - Configurações; 41 - Housekeeping; 42 - Autenticação; 43 - Dashboard de template; 44 - Função de usuário; 45 - Token de API; 46 - Relatório agendado; 47 - Nó de alta disponibilidade; 48 - SLA; 49 - Diretório de usuários; 50 - Grupo de templates; 51 - Conector; 52 - Regra LLD; 53 - Histórico.
resourceid	ID	Identificador do recurso da entrada do log de auditoria.

Propriedade	Tipo	Descrição
resource_cuid	ID	Identificador único do recurso da entrada do log de auditoria, gerado usando o algoritmo CUID.
resourcename	string	Nome legível do recurso da entrada do log de auditoria.
recordsetid	ID	ID do recordset da entrada do log de auditoria. Os registros de log de auditoria criados durante a mesma operação terão o mesmo ID de recordset. Gerado usando o algoritmo CUID.
details	text	Detalhes da entrada do log de auditoria. Os detalhes são armazenados como um objeto JSON, onde cada nome de propriedade é um caminho para a propriedade ou objeto aninhado em que a alteração ocorreu, e onde cada valor contém os dados (em formato de array) sobre a alteração nesta propriedade ou objeto aninhado. Possíveis formatos de valor: ["add"] - Objeto aninhado foi adicionado; ["add", "<valor>"] - A propriedade do objeto adicionado é igual a <valor>; ["update"] - Objeto aninhado foi atualizado; ["update", "<novo valor>", "<valor antigo>"] - A propriedade do objeto atualizado foi alterada de <valor antigo> para <novo valor>; ["delete"] - Objeto aninhado foi excluído.

auditlog.get

Descrição

`integer/array auditlog.get(object parameters)`

O método permite recuperar registros de log de auditoria de acordo com os parâmetros fornecidos.

> **Note:**
> Este método está disponível apenas para o tipo de usuário *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Consulte Funções de usuário para mais informações.

Parâmetros

`(object)` Parâmetros que definem a saída desejada.

O método suporta os seguintes parâmetros.

Parâmetro	Tipo	Descrição
auditids	ID/array	Retorna apenas logs de auditoria com os IDs fornecidos.
userids	ID/array	Retorna apenas logs de auditoria que foram criados pelos usuários fornecidos.
time_from	timestamp	Retorna apenas entradas de log de auditoria que foram criadas após ou no horário fornecido.
time_till	timestamp	Retorna apenas entradas de log de auditoria que foram criadas antes ou no horário fornecido.
sortfield	string/array	Ordena o resultado pelas propriedades fornecidas. Valores possíveis: `auditid`, `userid`, `clock`.
countOutput	boolean	Esses parâmetros são descritos no comentário de referência.
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valores de retorno

`(integer/array)` Retorna:

- um array de objetos;
- a contagem de objetos recuperados, se o parâmetro `countOutput` foi usado.

Exemplos

Recuperar o log de auditoria

Recupere os dois registros mais recentes do log de auditoria.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "auditlog.get",
    "params": {
        "output": "extend",
        "sortfield": "clock",
        "sortorder": "DESC",
        "limit": 2
    },
    "id": 1
}
```

Resposta:

```
{
    "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
}
```

Veja também

- objeto de log de auditoria

Fonte

CAuditLog::get() em *ui/include/classes/api/services/CAuditLog.php*.

Macro de usuário

Esta classe foi projetada para trabalhar com macros de usuário em nível de host e globais.

Referências de objetos:

- Macro global
- Macro de host

Métodos disponíveis:

- usermacro.create - criar novas macros de host
- usermacro.createglobal - criar novas macros globais
- usermacro.delete - excluir macros de host
- usermacro.deleteglobal - excluir macros globais
- usermacro.get - recuperar macros de host e globais
- usermacro.update - atualizar macros de host
- usermacro.updateglobal - atualizar macros globais

Objeto de macro de usuário

Os seguintes objetos estão diretamente relacionados à API `usermacro`.

Macro global

O objeto macro global possui as seguintes propriedades.

Propriedade	Tipo	Descrição
globalmacroid	ID	ID da macro global. Comportamento da propriedade: - *somente leitura* - *obrigatório* para operações de atualização
macro	string	String da macro. Comportamento da propriedade: - *obrigatório* para operações de criação
value	string	Valor da macro. Comportamento da propriedade: - *somente gravação* se `type` estiver definido como "Macro secreta" - *obrigatório* para operações de criação
type	integer	Tipo de macro. Valores possíveis: 0 - *(padrão)* Macro de texto; 1 - Macro secreta; 2 - Segredo do Vault.
description	string	Descrição da macro.

Macro de host

O objeto macro de host define uma macro disponível em um host, protótipo de host ou template. Possui as seguintes propriedades.

Propriedade	Tipo	Descrição
hostmacroid	ID	ID da macro de host. Comportamento da propriedade: - *somente leitura* - *obrigatório* para operações de atualização

Propriedade	Tipo	Descrição
hostid	ID	ID do host ao qual a macro pertence. Comportamento da propriedade: - *constante* - *obrigatório* para operações de criação
macro	string	String da macro. Comportamento da propriedade: - *obrigatório* para operações de criação
value	string	Valor da macro. Comportamento da propriedade: - *somente gravação* se `type` estiver definido como ”Macro secreta” - *obrigatório* para operações de criação
type	integer	Tipo de macro. Valores possíveis: 0 - *(padrão)* Macro de texto; 1 - Macro secreta; 2 - Segredo do Vault.
description	string	Descrição da macro.
automatic	integer	Define se a macro é controlada por regra de descoberta. Valores possíveis: 0 - *(padrão)* Macro é gerenciada pelo usuário; 1 - Macro é gerenciada por regra de descoberta. O usuário não tem permissão para criar macro automática. Para atualizar macro automática, ela deve ser convertida para manual.

usermacro.create

Descrição

`object usermacro.create(object/array hostMacros)`

Este método permite criar novas macros de host.

> **Note:**
> Este método está disponível apenas para os tipos de usuário *Admin* e *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object/array)` Macros de host a serem criadas.

O método aceita macros de host com as propriedades padrão de macro de host.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs das macros de host criadas sob a propriedade `hostmacroids`. A ordem dos IDs retornados corresponde à ordem das macros de host passadas.

Exemplos

Criando uma macro de host

Crie uma macro de host ”{$SNMP_COMMUNITY}” com o valor ”public” no host ”10198”.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "usermacro.create",
    "params": {
```

```
        "hostid": "10198",
        "macro": "{$SNMP_COMMUNITY}",
        "value": "public"
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostmacroids": [
            "11"
        ]
    },
    "id": 1
}
```

Fonte

CUserMacro::create() in *ui/include/classes/api/services/CUserMacro.php*.

usermacro.createglobal

Descrição

`object usermacro.createglobal(object/array globalMacros)`

Este método permite criar novas macros globais.

> **Note:**
> Este método está disponível apenas para o tipo de usuário *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object/array)` Macros globais a serem criadas.

O método aceita macros globais com as propriedades padrão de macro global.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs das macros globais criadas sob a propriedade `globalmacroids`. A ordem dos IDs retornados corresponde à ordem das macros globais passadas.

Exemplos

Criando uma macro global

Crie uma macro global "{$SNMP_COMMUNITY}" com valor "public".

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "usermacro.createglobal",
    "params":  {
        "macro": "{$SNMP_COMMUNITY}",
        "value": "public"
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "globalmacroids": [
```

```
            "6"
        ]
    },
    "id": 1
}
```

Fonte

CUserMacro::createGlobal() in *ui/include/classes/api/services/CUserMacro.php*.

usermacro.delete

Descrição

`object usermacro.delete(array hostMacroIds)`

Este método permite excluir macros de host.

> **Note:**
> Este método está disponível apenas para os tipos de usuário *Admin* e *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função de usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(array)` IDs das macros de hosts a serem deletadas.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs das macros de host excluídas na propriedade `hostmacroids`.

Exemplos

Excluindo várias macros de host

Exclua duas macros de host.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "usermacro.delete",
    "params": [
        "32",
        "11"
    ],
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostmacroids": [
            "32",
            "11"
        ]
    },
    "id": 1
}
```

Fonte

CUserMacro::delete() in *ui/include/classes/api/services/CUserMacro.php*.

usermacro.deleteglobal

Descrição

`object usermacro.deleteglobal(array globalMacroIds)`

Este método permite excluir macros globais.

> **Note:**
> Este método está disponível apenas para o tipo de usuário *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Consulte Funções de usuário para mais informações.

Parâmetros

`(array)` IDs das macros globais a serem excluídas.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs das macros globais excluídas na propriedade `globalmacroids`.

Exemplos

Excluindo múltiplas macros globais

Exclua duas macros globais.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "usermacro.deleteglobal",
    "params": [
        "32",
        "11"
    ],
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "globalmacroids": [
            "32",
            "11"
        ]
    },
    "id": 1
}
```

Fonte

CUserMacro::deleteGlobal() in *ui/include/classes/api/services/CUserMacro.php*.

usermacro.get

Descrição

`integer/array usermacro.get(object parameters)`

O método permite recuperar macros de host e globais de acordo com os parâmetros fornecidos.

> **Note:**
> Este método está disponível para usuários de qualquer tipo. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object)` Parâmetros que definem a saída desejada.

O método suporta os seguintes parâmetros.

Parâmetro	Tipo	Descrição
globalmacro	flag	Retorna macros globais em vez de macros de host.
globalmacroids	ID/array	Retorna apenas macros globais com os IDs fornecidos.
groupids	ID/array	Retorna apenas macros de host que pertencem a hosts ou templates dos grupos de hosts ou grupos de templates fornecidos.
hostids	ID/array	Retorna apenas macros que pertencem aos hosts ou templates fornecidos.
hostmacroids	ID/array	Retorna apenas macros de host com os IDs fornecidos.
inherited	boolean	Se definido como `true`, retorna apenas macros de usuário de protótipo de host herdadas de um template.
selectHostGroups	query	Retorna grupos de hosts aos quais a macro de host pertence na propriedade `hostgroups`. Usado apenas ao recuperar macros de host.
selectHosts	query	Retorna hosts aos quais a macro de host pertence na propriedade `hosts`. Usado apenas ao recuperar macros de host.
selectTemplateGroups	query	Retorna grupos de templates aos quais a macro de template pertence na propriedade `templategroups`. Usado apenas ao recuperar macros de template.
selectTemplates	query	Retorna templates aos quais a macro de host pertence na propriedade `templates`. Usado apenas ao recuperar macros de host.
sortfield	string/array	Ordena o resultado pelas propriedades fornecidas. Valores possíveis: `macro`.
countOutput	boolean	Esses parâmetros são descritos no comentário de referência.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	
selectGroups (obsoleto)	query	Este parâmetro está obsoleto, por favor use `selectHostGroups` ou `selectTemplateGroups` em seu lugar. Retorna grupos de hosts e grupos de templates aos quais a macro de host pertence na propriedade `groups`. Usado apenas ao recuperar macros de host.

Valores de retorno

`(integer/array)` Retorna:

- um array de objetos;
- a contagem de objetos recuperados, se o parâmetro `countOutput` tiver sido usado.

Exemplos

Recuperando macros de host para um host

Recupere todas as macros de host definidas para o host "10198".

Requisição:

```
{
    "jsonrpc": "2.0",
```

```
    "method": "usermacro.get",
    "params": {
        "output": "extend",
        "hostids": "10198"
    },
    "id": 1
}
```

Resposta:

```
{
    "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
}
```

Recuperando macros globais

Recupere todas as macros globais.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "usermacro.get",
    "params": {
        "output": "extend",
        "globalmacro": true
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "globalmacroid": "6",
            "macro": "{$SNMP_COMMUNITY}",
            "value": "public",
            "description": "",
            "type": "0"
        }
    ],
    "id": 1
}
```

Fonte

CUserMacro::get() in *ui/include/classes/api/services/CUserMacro.php*.

usermacro.update

Descrição

`object usermacro.update(object/array hostMacros)`

Este método permite atualizar macros de host existentes.

> **Note:**
> Este método está disponível apenas para os tipos de usuário *Admin* e *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object/array)` Propriedades de macro de host a serem atualizadas.

A propriedade `hostmacroid` deve ser definida para cada macro de host, todas as outras propriedades são opcionais. Apenas as propriedades passadas serão atualizadas, todas as outras permanecerão inalteradas.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs das macros de host atualizadas na propriedade `hostmacroids`.

Exemplos

Alterando o valor de uma macro de host

Altere o valor de uma macro de host para ”public”.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "usermacro.update",
    "params": {
        "hostmacroid": "1",
        "value": "public"
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostmacroids": [
            "1"
        ]
    },
    "id": 1
}
```

Alterar o valor da macro que foi criada por regra de descoberta

Converta a macro ”automática” criada pela regra de descoberta para ”manual” e altere seu valor para ”novo-valor”.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "usermacro.update",
    "params": {
        "hostmacroid": "1",
        "value": "novo-valor",
        "automatic": "0"
    },
```

```
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostmacroids": [
            "1"
        ]
    },
    "id": 1
}
```

Fonte

CUserMacro::update() in *ui/include/classes/api/services/CUserMacro.php*.

usermacro.updateglobal

Descrição

`object usermacro.updateglobal(object/array globalMacros)`

Este método permite atualizar macros globais existentes.

> **Note:**
> Este método está disponível apenas para o tipo de usuário *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função de usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object/array)` Propriedades de macro global a serem atualizadas.

A propriedade `globalmacroid` deve ser definida para cada macro global, todas as outras propriedades são opcionais. Somente as propriedades passadas serão atualizadas, todas as outras permanecerão inalteradas.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs das macros globais atualizadas na propriedade `globalmacroids`.

Exemplos

Alterando o valor de uma macro global

Altere o valor de uma macro global para "public".

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "usermacro.updateglobal",
    "params": {
        "globalmacroid": "1",
        "value": "public"
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "globalmacroids": [
            "1"
        ]
    },
```

```
    "id": 1
}
```

Fonte

CUserMacro::updateGlobal() in *ui/include/classes/api/services/CUserMacro.php*.

Manutenção

Esta classe foi projetada para trabalhar com manutenções.

Referências de objetos:

- Manutenção
- Período de tempo
- Tag de problema

Métodos disponíveis:

- maintenance.create - criar novas manutenções
- maintenance.delete - excluir manutenções
- maintenance.get - recuperar manutenções
- maintenance.update - atualizar manutenções

Objeto de manutenção

Os seguintes objetos estão diretamente relacionados à API `maintenance`.

Manutenção

O objeto de manutenção possui as seguintes propriedades.

Propriedade	Tipo	Descrição
maintenanceid	ID	ID da manutenção. Comportamento da propriedade: - *somente leitura* - *obrigatório* para operações de atualização
name	string	Nome da manutenção. Comportamento da propriedade: - *obrigatório* para operações de criação
active_since	timestamp	Hora em que a manutenção se torna ativa (inclusiva). O valor fornecido será arredondado para baixo para minutos. Comportamento da propriedade: - *obrigatório* para operações de criação
active_till	timestamp	Hora em que a manutenção deixa de estar ativa (exclusiva). O valor fornecido será arredondado para baixo para minutos. Comportamento da propriedade: - *obrigatório* para operações de criação
description	string	Descrição da manutenção.
maintenance_type	integer	Tipo de manutenção. Valores possíveis: 0 - *(padrão)* com coleta de dados; 1 - sem coleta de dados.

Propriedade	Tipo	Descrição
tags_evaltype	integer	Método de avaliação de tags de problema. Valores possíveis: 0 - *(padrão)* E/Ou; 2 - Ou.

Período de tempo

O objeto de período de tempo é usado para definir períodos em que a manutenção deve entrar em vigor. Ele possui as seguintes propriedades.

Propriedade	Tipo	Descrição
period	integer	Duração do período de manutenção em segundos. O valor fornecido será arredondado para baixo para minutos. Valores possíveis: 300 - 86399940. Padrão: 3600.
timeperiod_type	integer	Tipo de período de tempo. Valores possíveis: 0 - *(padrão)* apenas uma vez; 2 - diariamente; 3 - semanalmente; 4 - mensalmente.
start_date	timestamp	Data em que o período de manutenção deve entrar em vigor. O valor fornecido será arredondado para baixo para minutos. Padrão: data atual. Comportamento da propriedade: - *suportado* se `timeperiod_type` for definido como ”apenas uma vez”
start_time	integer	Hora do dia em que a manutenção começa em segundos. O valor fornecido será arredondado para baixo para minutos. Padrão: 0. Comportamento da propriedade: - *suportado* se `timeperiod_type` for definido como ”diariamente”, ”semanalmente” ou ”mensalmente”

Propriedade	Tipo	Descrição
every	integer	Para períodos diários e semanais, `every` define os intervalos de dias ou semanas em que a manutenção deve entrar em vigor. Valor padrão se `timeperiod_type` for definido como ”diariamente” ou ”semanalmente”: `1`. Para períodos mensais quando `day` está definido, a propriedade `every` define o dia do mês em que a manutenção deve entrar em vigor. Valor padrão se `timeperiod_type` for definido como ”mensalmente” e `day` estiver definido: `1`. Para períodos mensais quando `dayofweek` está definido, a propriedade `every` define a semana do mês em que a manutenção deve entrar em vigor. Valores possíveis se `timeperiod_type` for definido como ”mensalmente” e `dayofweek` estiver definido: 1 - *(padrão)* primeira semana; 2 - segunda semana; 3 - terceira semana; 4 - quarta semana; 5 - última semana. Comportamento da propriedade: - *suportado* se `timeperiod_type` for definido como ”diariamente”, ”semanalmente” ou ”mensalmente”
dayofweek	integer	Dias da semana em que a manutenção deve entrar em vigor. Valores possíveis de bitmap: 1 - Segunda-feira; 2 - Terça-feira; 4 - Quarta-feira; 8 - Quinta-feira; 16 - Sexta-feira; 32 - Sábado; 64 - Domingo. Este é um campo de máscara de bits; qualquer soma dos valores possíveis é aceitável (por exemplo, `21` para segunda, quarta e sexta-feira). Comportamento da propriedade: - *obrigatório* se `timeperiod_type` for definido como ”semanalmente” ou se `timeperiod_type` for definido como ”mensalmente” e `day` não estiver definido
day	integer	Dia do mês em que a manutenção deve entrar em vigor. Comportamento da propriedade: - *obrigatório* se `timeperiod_type` for definido como ”mensalmente” e `dayofweek` não estiver definido

Propriedade	Tipo	Descrição
month	integer	Meses em que a manutenção deve entrar em vigor. Valores possíveis de bitmap: 1 - Janeiro; 2 - Fevereiro; 4 - Março; 8 - Abril; 16 - Maio; 32 - Junho; 64 - Julho; 128 - Agosto; 256 - Setembro; 512 - Outubro; 1024 - Novembro; 2048 - Dezembro. Este é um campo de máscara de bits; qualquer soma dos valores possíveis é aceitável (por exemplo, 585 para janeiro, abril, julho e outubro). Comportamento da propriedade: - *obrigatório* se `timeperiod_type` for definido como ”mensalmente”

Tag de problema

O objeto de tag de problema é usado para definir quais problemas devem ser suprimidos quando a manutenção entrar em vigor. As tags só podem ser especificadas se o `maintenance_type` do objeto Maintenance estiver definido como ”com coleta de dados”. Possui as seguintes propriedades.

Propriedade	Tipo	Descrição
tag	string	Nome da tag do problema. Comportamento da propriedade: - *obrigatório*
operator	integer	Operador da condição. Valores possíveis: 0 - Igual; 2 - *(padrão)* Contém.
value	string	Valor da tag do problema.

maintenance.create

Descrição

`object maintenance.create(object/array maintenances)`

Este método permite criar novas manutenções.

> **Note:**
> Este método está disponível apenas para os tipos de usuário *Admin* e *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object/array)` Manutenções a serem criadas.

Além das propriedades padrão de manutenção, o método aceita os seguintes parâmetros.

Parâmetro	Tipo	Descrição
groups	object/array	Grupos de hosts que passarão por manutenção. Os grupos de hosts devem ter apenas a propriedade `groupid` definida. Comportamento do parâmetro: - *obrigatório* se `hosts` não estiver definido
hosts	object/array	Hosts que passarão por manutenção. Os hosts devem ter apenas a propriedade `hostid` definida. Comportamento do parâmetro: - *obrigatório* se `groups` não estiver definido
timeperiods	object/array	Períodos de tempo de manutenção. Comportamento do parâmetro: - *obrigatório*
tags	object/array	Tags de problema. Define quais problemas devem ser suprimidos. Se nenhuma tag for fornecida, todos os problemas ativos de manutenção do host serão suprimidos. Comportamento do parâmetro: - *suportado* se `maintenance_type` do objeto Maintenance estiver definido como "com coleta de dados"
groupids (obsoleto)	array	Este parâmetro está obsoleto, por favor use `groups` em seu lugar. IDs dos grupos de hosts que passarão por manutenção.
hostids (obsoleto)	array	Este parâmetro está obsoleto, por favor use `hosts` em seu lugar. IDs dos hosts que passarão por manutenção.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs das manutenções criadas na propriedade `maintenanceids`. A ordem dos IDs retornados corresponde à ordem da manutenção aprovada.

Exemplos

Criando uma manutenção

Crie uma manutenção com coleta de dados para o grupo de hosts com ID "2" e com tags de problema **service:mysqld** e **error**. Ela deve estar ativa de 17.03.2026 até 17.03.2027, entrar em vigor todo domingo às 18:00 e durar uma hora.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "maintenance.create",
    "params": {
        "name": "Sunday maintenance",
        "active_since": 1773720240,
        "active_till": 1805256240,
        "tags_evaltype": 0,
        "groups": [
            {"groupid": "2"}
        ],
        "timeperiods": [
            {
                "period": 3600,
                "timeperiod_type": 3,
                "start_time": 64800,
                "every": 1,
                "dayofweek": 64
            }
        ],
```

```
        "tags": [
            {
                "tag": "service",
                "operator": "0",
                "value": "mysqld"
            },
            {
                "tag": "error",
                "operator": "2",
                "value": ""
            }
        ]
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "maintenanceids": [
            "3"
        ]
    },
    "id": 1
}
```

Veja também

- Período de tempo

Fonte

CMaintenance::create() in *ui/include/classes/api/services/CMaintenance.php*.

maintenance.delete

Descrição

`object maintenance.delete(array maintenanceIds)`

Este método permite excluir períodos de manutenção.

> **Note:**
> Este método está disponível apenas para os tipos de usuário *Admin* e *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(array)` IDs dos períodos de manutenção a serem excluídos.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos períodos de manutenção excluídos na propriedade `maintenanceids`.

Exemplos

Excluindo múltiplos períodos de manutenção

Exclua dois períodos de manutenção.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "maintenance.delete",
    "params": [
        "3",
```

```
        "1"
    ],
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "maintenanceids": [
            "3",
            "1"
        ]
    },
    "id": 1
}
```

Fonte

CMaintenance::delete() in *ui/include/classes/api/services/CMaintenance.php*.

maintenance.get

Descrição

`integer/array maintenance.get(object parameters)`

O método permite recuperar manutenções de acordo com os parâmetros fornecidos.

Note:
Este método está disponível para usuários de qualquer tipo. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object)` Parâmetros que definem a saída desejada.

O método suporta os seguintes parâmetros.

Parâmetro	Tipo	Descrição
groupids	ID/array	Retorna apenas manutenções atribuídas aos grupos de hosts fornecidos.
hostids	ID/array	Retorna apenas manutenções atribuídas aos hosts fornecidos.
maintenanceids	ID/array	Retorna apenas manutenções com os IDs fornecidos.
selectHostGroups	query	Retorna uma propriedade `hostgroups` com os grupos de hosts atribuídos à manutenção.
selectHosts	query	Retorna uma propriedade `hosts` com os hosts atribuídos à manutenção.
selectTags	query	Retorna uma propriedade `tags` com as tags de problema da manutenção.
selectTimeperiods	query	Retorna uma propriedade `timeperiods` com os períodos de tempo da manutenção.
sortfield	string/array	Ordena o resultado pelas propriedades fornecidas. Valores possíveis: `maintenanceid`, `name`, `maintenance_type`, `active_since`, `active_till`.
countOutput	boolean	Esses parâmetros são descritos no comentário de referência.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	

Parâmetro	Tipo	Descrição
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	
selectGroups (obsoleto)	query	Este parâmetro está obsoleto, por favor, use `selectHostGroups` em seu lugar. Retorna uma propriedade `groups` com os grupos de hosts atribuídos à manutenção.

Valores de retorno

`(integer/array)` Retorna:

- um array de objetos;
- a contagem de objetos recuperados, se o parâmetro `countOutput` tiver sido usado.

Exemplos

Recuperando manutenções

Recupere todas as manutenções configuradas e os dados sobre os grupos de hosts atribuídos, períodos de tempo definidos e tags de problemas.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "maintenance.get",
    "params": {
        "output": "extend",
        "selectHostGroups": "extend",
        "selectTimeperiods": "extend",
        "selectTags": "extend"
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "maintenanceid": "3",
            "name": "Sunday maintenance",
            "maintenance_type": "0",
            "description": "",
            "active_since": "1358844540",
            "active_till": "1390466940",
            "tags_evaltype": "0",
            "hostgroups": [
                {
                    "groupid": "4",
                    "name": "Zabbix servers",
                    "flags": "0",
                    "uuid": "6f6799aa69e844b4b3918f779f2abf08"
                }
            ],
            "timeperiods": [
                {
                    "timeperiod_type": "3",
                    "every": "1",
                    "month": "0",
                    "dayofweek": "1",
                    "day": "0",
```

```
                    "start_time": "64800",
                    "period": "3600",
                    "start_date": "2147483647"
                }
            ],
            "tags": [
                {
                    "tag": "service",
                    "operator": "0",
                    "value": "mysqld",
                },
                {
                    "tag": "error",
                    "operator": "2",
                    "value": ""
                }
            ]
        }
    ],
    "id": 1
}
```

Veja também

- Host
- Grupo de hosts
- Período de tempo

Fonte

CMaintenance::get() em *ui/include/classes/api/services/CMaintenance.php*.

maintenance.update

Descrição

`object maintenance.update(object/array maintenances)`

Este método permite atualizar manutenções existentes.

> **Note:**
> Este método está disponível apenas para os tipos de usuário *Admin* e *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object/array)` Propriedades de manutenção a serem atualizadas.

A propriedade `maintenanceid` deve ser definida para cada manutenção, todas as outras propriedades são opcionais. Somente as propriedades passadas serão atualizadas, todas as outras permanecerão inalteradas.

Além das propriedades padrão de manutenção, o método aceita os seguintes parâmetros.

Parâmetro	Tipo	Descrição
groups	object/array	Grupos de hosts para substituir os grupos atuais. Os grupos de hosts devem ter apenas a propriedade `groupid` definida. Comportamento do parâmetro: - *obrigatório* se `hosts` não estiver definido

Parâmetro	Tipo	Descrição
hosts	object/array	Hosts para substituir os hosts atuais. Os hosts devem ter apenas a propriedade `hostid` definida. Comportamento do parâmetro: - *obrigatório* se `groups` não estiver definido
timeperiods	object/array	Períodos de tempo de manutenção para substituir os períodos atuais.
tags	object/array	Tags de problema para substituir as tags atuais. Comportamento do parâmetro: - *suportado* se `maintenance_type` do objeto Maintenance estiver definido como "com coleta de dados"
groupids (obsoleto)	array	Este parâmetro está obsoleto, por favor use `groups` em seu lugar. IDs dos grupos de hosts que entrarão em manutenção.
hostids (obsoleto)	array	Este parâmetro está obsoleto, por favor use `hosts` em seu lugar. IDs dos hosts que entrarão em manutenção.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs das manutenções atualizadas na propriedade `maintenanceids`.

Exemplos

Atribuindo hosts diferentes

Substitua os hosts atualmente atribuídos à manutenção por dois diferentes.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "maintenance.update",
    "params": {
        "maintenanceid": "3",
        "hosts": [
            {"hostid": "10085"},
            {"hostid": "10084"}
        ]
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "maintenanceids": [
            "3"
        ]
    },
    "id": 1
}
```

Veja também

- Período de tempo

Fonte

CMaintenance::update() in *ui/include/classes/api/services/CMaintenance.php*.

Map

Esta classe foi projetada para trabalhar com mapas.

Referências de objetos:

- Map
- Map element
  - Map element Host
  - Map element Host group
  - Map element Map
  - Map element Trigger
  - Map element Tag
  - Map element URL
- Map link
  - Map link trigger
  - Map link indicators
- Map URL
- Map user
- Map user group
- Map shapes
- Map lines

Métodos disponíveis:

- map.create - criar novos mapas
- map.delete - excluir mapas
- map.get - recuperar mapas
- map.update - atualizar mapas

Objeto de mapa

Os seguintes objetos estão diretamente relacionados à API de `map`.

Mapa

O objeto mapa possui as seguintes propriedades.

Propriedade	Tipo	Descrição
sysmapid	ID	ID do mapa. Comportamento da propriedade: - *somente leitura* - *obrigatório* para operações de atualização
height	integer	Altura do mapa em pixels. Comportamento da propriedade: - *obrigatório* para operações de criação
name	string	Nome do mapa. Comportamento da propriedade: - *obrigatório* para operações de criação
width	integer	Largura do mapa em pixels. Comportamento da propriedade: - *obrigatório* para operações de criação
backgroundid	ID	ID da imagem usada como fundo do mapa.
expand_macros	integer	Se macros serão expandidas nos rótulos ao configurar o mapa. Valores possíveis: 0 - *(padrão)* não expandir macros; 1 - expandir macros.

Propriedade	Tipo	Descrição
expandproblem	integer	Se o trigger de problema será exibido para elementos com um único problema. Valores possíveis: 0 - sempre exibir o número de problemas; 1 - *(padrão)* exibir o trigger de problema se houver apenas um problema.
grid_align	integer	Se o alinhamento à grade está habilitado. Valores possíveis: 0 - desabilitar alinhamento à grade; 1 - *(padrão)* habilitar alinhamento à grade.
grid_show	integer	Se a grade será exibida no mapa. Valores possíveis: 0 - não exibir a grade; 1 - *(padrão)* exibir a grade.
grid_size	integer	Tamanho da grade do mapa em pixels. Valores suportados: 20, 40, 50, 75 e 100. Padrão: 50.
highlight	integer	Se o destaque de ícones está habilitado. Valores possíveis: 0 - destaque desabilitado; 1 - *(padrão)* destaque habilitado.
iconmapid	ID	ID do mapa de ícones usado no mapa.
label_format	integer	Se rótulos avançados estão habilitados. Valores possíveis: 0 - *(padrão)* desabilitar rótulos avançados; 1 - habilitar rótulos avançados.
label_location	integer	Localização do rótulo do elemento do mapa. Valores possíveis: 0 - *(padrão)* inferior; 1 - esquerda; 2 - direita; 3 - superior.
label_string_host	string	Rótulo personalizado para elementos de host. Comportamento da propriedade: - *obrigatório* se `label_type_host` estiver definido como ”custom”
label_string_hostgroup	string	Rótulo personalizado para elementos de grupo de hosts. Comportamento da propriedade: - *obrigatório* se `label_type_hostgroup` estiver definido como ”custom”
label_string_image	string	Rótulo personalizado para elementos de imagem. Comportamento da propriedade: - *obrigatório* se `label_type_image` estiver definido como ”custom”
label_string_map	string	Rótulo personalizado para elementos de mapa. Comportamento da propriedade: - *obrigatório* se `label_type_map` estiver definido como ”custom”
label_string_trigger	string	Rótulo personalizado para elementos de trigger. Comportamento da propriedade: - *obrigatório* se `label_type_trigger` estiver definido como ”custom”

Propriedade	Tipo	Descrição
label_type	integer	Tipo de rótulo do elemento do mapa. Valores possíveis: 0 - rótulo; 1 - endereço IP; 2 - *(padrão)* nome do elemento; 3 - apenas status; 4 - nada.
label_type_host	integer	Tipo de rótulo para elementos de host. Valores possíveis: 0 - rótulo; 1 - endereço IP; 2 - *(padrão)* nome do elemento; 3 - apenas status; 4 - nada; 5 - personalizado.
label_type_hostgroup	integer	Tipo de rótulo para elementos de grupo de hosts. Valores possíveis: 0 - rótulo; 2 - *(padrão)* nome do elemento; 3 - apenas status; 4 - nada; 5 - personalizado.
label_type_image	integer	Tipo de rótulo para elementos de grupo de hosts. Valores possíveis: 0 - rótulo; 2 - *(padrão)* nome do elemento; 4 - nada; 5 - personalizado.
label_type_map	integer	Tipo de rótulo para elementos de mapa. Valores possíveis: 0 - rótulo; 2 - *(padrão)* nome do elemento; 3 - apenas status; 4 - nada; 5 - personalizado.
label_type_trigger	integer	Tipo de rótulo para elementos de trigger. Valores possíveis: 0 - rótulo; 2 - *(padrão)* nome do elemento; 3 - apenas status; 4 - nada; 5 - personalizado.
markelements	integer	Se os elementos do mapa que mudaram recentemente de status serão destacados. Valores possíveis: 0 - *(padrão)* não destacar elementos; 1 - destacar elementos.
severity_min	integer	Severidade mínima dos triggers que serão exibidos no mapa. Consulte a propriedade `severity` do trigger para uma lista de severidades de trigger suportadas.

Propriedade	Tipo	Descrição
show_unack	integer	Como os problemas devem ser exibidos. Valores possíveis: 0 - *(padrão)* exibir a contagem de todos os problemas; 1 - exibir apenas a contagem de problemas não reconhecidos; 2 - exibir a contagem de problemas reconhecidos e não reconhecidos separadamente.
userid	ID	ID do usuário que é o proprietário do mapa.
private	integer	Tipo de compartilhamento do mapa. Valores possíveis: 0 - mapa público; 1 - *(padrão)* mapa privado.
show_suppressed	integer	Se problemas suprimidos são exibidos. Valores possíveis: 0 - *(padrão)* ocultar problemas suprimidos; 1 - exibir problemas suprimidos.

Elemento do mapa

O objeto de elemento do mapa define um objeto exibido em um mapa. Ele possui as seguintes propriedades.

Propriedade	Tipo	Descrição
selementid	ID	ID do elemento do mapa. Comportamento da propriedade: - *somente leitura*
elements	array	Objeto de dados do Elemento. Comportamento da propriedade: - *obrigatório* se `elementtype` estiver definido como ”host”, ”mapa”, ”trigger” ou ”grupo de hosts”
elementtype	integer	Tipo de elemento do mapa. Valores possíveis: 0 - host; 1 - mapa; 2 - trigger; 3 - grupo de hosts; 4 - imagem. Comportamento da propriedade: - *obrigatório*
iconid_off	ID	ID da imagem usada para exibir o elemento no estado padrão. Comportamento da propriedade: - *obrigatório*
areatype	integer	Como os hosts do grupo de hosts devem ser exibidos. Valores possíveis: 0 - *(padrão)* o elemento do grupo de hosts ocupará todo o mapa; 1 - o elemento do grupo de hosts terá um tamanho fixo.
elementsubtype	integer	Como um elemento de grupo de hosts deve ser exibido em um mapa. Valores possíveis: 0 - *(padrão)* exibir o grupo de hosts como um único elemento; 1 - exibir cada host do grupo separadamente.

Propriedade	Tipo	Descrição
evaltype	integer	Condição de filtragem de tag do elemento do mapa método de avaliação. Valores possíveis: 0 - *(padrão)* E/Ou; 2 - Ou.
height	integer	Altura do elemento de grupo de hosts de tamanho fixo em pixels. Padrão: 200.
iconid_disabled	ID	ID da imagem usada para exibir elementos de mapa desabilitados. Comportamento da propriedade: - *suportado* se `elementtype` estiver definido como ”host”, ”mapa”, ”trigger” ou ”grupo de hosts”
iconid_maintenance	ID	ID da imagem usada para exibir elementos de mapa em manutenção. Comportamento da propriedade: - *suportado* se `elementtype` estiver definido como ”host”, ”mapa”, ”trigger” ou ”grupo de hosts”
iconid_on	ID	ID da imagem usada para exibir elementos de mapa com problemas. Comportamento da propriedade: - *suportado* se `elementtype` estiver definido como ”host”, ”mapa”, ”trigger” ou ”grupo de hosts”
label	string	Rótulo do elemento.
label_location	integer	Localização do rótulo do elemento do mapa. Valores possíveis: -1 - *(padrão)* localização padrão; 0 - inferior; 1 - esquerda; 2 - direita; 3 - superior.
permission	integer	Tipo de nível de permissão. Valores possíveis: -1 - nenhum; 2 - somente leitura; 3 - leitura-escrita.
sysmapid	ID	ID do mapa ao qual o elemento pertence. Comportamento da propriedade: - *somente leitura*
urls	array	URLs do elemento do mapa. O objeto de URL do elemento do mapa é descrito em detalhes abaixo.
use_iconmap	integer	Se o mapeamento de ícones deve ser usado para elementos de host. Valores possíveis: 0 - não usar mapeamento de ícones; 1 - *(padrão)* usar mapeamento de ícones.
viewtype	integer	Algoritmo de posicionamento do elemento do grupo de hosts. Valores possíveis: 0 - *(padrão)* grade.
width	integer	Largura do elemento de grupo de hosts de tamanho fixo em pixels. Padrão: 200.
x	integer	Coordenadas X do elemento em pixels. Padrão: 0.

Propriedade	Tipo	Descrição
y	integer	Coordenadas Y do elemento em pixels. Padrão: 0.

Elemento de mapa Host

O objeto de elemento de mapa Host define um elemento de host.

Propriedade	Tipo	Descrição
hostid	ID	ID do host.

Elemento de mapa Grupo de hosts

O objeto de elemento de mapa Grupo de hosts define um elemento de grupo de hosts.

Propriedade	Tipo	Descrição
groupid	ID	ID do grupo de hosts.

Mapa de elemento Mapa

O objeto de mapa de elemento Mapa define um elemento de mapa.

Propriedade	Tipo	Descrição
sysmapid	ID	ID do mapa.

Elemento de mapa Trigger

O objeto de elemento de mapa Trigger define um ou mais elementos de trigger.

Propriedade	Tipo	Descrição
triggerid	ID	ID do trigger.

Tag do elemento do mapa

O objeto Tag do elemento do mapa possui as seguintes propriedades.

Propriedade	Tipo	Descrição
tag	string	Nome da tag do elemento do mapa. Comportamento da propriedade: - *obrigatório*
operator	integer	Operador da condição da tag do elemento do mapa. Valores possíveis: 0 - *(padrão)* Contém; 1 - Igual; 2 - Não contém; 3 - Diferente; 4 - Existe; 5 - Não existe.
value	string	Valor da tag do elemento do mapa.

URL do elemento do mapa

O objeto de URL do elemento do mapa define um link clicável que estará disponível para um elemento de mapa específico. Ele possui as seguintes propriedades:

Propriedade	Tipo	Descrição
sysmapelementurlid	ID	ID da URL do elemento do mapa. Comportamento da propriedade: - *somente leitura*
name	string	Legenda do link. Comportamento da propriedade: - *obrigatório*
url	string	URL do link. Comportamento da propriedade: - *obrigatório*
selementid	ID	ID do elemento do mapa ao qual a URL pertence.

Link de mapa

O objeto de link de mapa define um link entre dois elementos do mapa. Ele possui as seguintes propriedades.

Propriedade	Tipo	Descrição
linkid	ID	ID do link do mapa. Comportamento da propriedade: - *somente leitura*
selementid1	ID	ID do primeiro elemento do mapa vinculado em uma extremidade. Comportamento da propriedade: - *obrigatório*
selementid2	ID	ID do primeiro elemento do mapa vinculado na outra extremidade. Comportamento da propriedade: - *obrigatório*
color	string	Cor da linha como um código de cor hexadecimal. Padrão: 000000.
drawtype	integer	Estilo de desenho da linha do link. Valores possíveis: 0 - *(padrão)* linha; 2 - linha em negrito; 3 - linha pontilhada; 4 - linha tracejada.
label	string	Rótulo do link.
linktriggers	array	Triggers de link de mapa a serem usadas como indicadores de status do link. O objeto de trigger de link de mapa é descrito em detalhes abaixo.
permission	integer	Tipo de nível de permissão. Valores possíveis: -1 - nenhum; 2 - somente leitura; 3 - leitura e gravação.
sysmapid	ID	ID do mapa ao qual o link pertence.

Trigger de link de mapa

O objeto de trigger de link de mapa define um indicador de status de link de mapa com base no estado de um trigger. Ele possui as seguintes propriedades:

Propriedade	Tipo	Descrição
linktriggerid	ID	ID do trigger de link de mapa. Comportamento da propriedade: - *somente leitura*
triggerid	ID	ID do trigger usado como indicador de link. Comportamento da propriedade: - *obrigatório*
color	string	Cor do indicador como um código de cor hexadecimal. Padrão: `DD0000`.
drawtype	integer	Estilo de desenho do indicador. Valores possíveis: 0 - *(padrão)* linha; 2 - linha em negrito; 3 - linha pontilhada; 4 - linha tracejada.
linkid	ID	ID do link de mapa ao qual o trigger de link pertence.

URL do mapa

O objeto de URL do mapa define um link clicável que estará disponível para todos os elementos de um tipo específico no mapa. Ele possui as seguintes propriedades:

Propriedade	Tipo	Descrição
sysmapurlid	ID	ID da URL do mapa. Comportamento da propriedade: - *somente leitura*
name	string	Legenda do link. Comportamento da propriedade: - *obrigatório*
url	string	URL do link. Comportamento da propriedade: - *obrigatório*
elementtype	integer	Tipo de elemento do mapa para o qual a URL estará disponível. Consulte a propriedade `type` do elemento do mapa para uma lista de tipos suportados. Padrão: 0.
sysmapid	ID	ID do mapa ao qual a URL pertence.

Usuário do mapa

Lista de permissões de mapa com base nos usuários. Possui as seguintes propriedades:

Propriedade	Tipo	Descrição
sysmapuserid	ID	ID do usuário do mapa. Comportamento da propriedade: - *somente leitura*
userid	ID	ID do usuário. Comportamento da propriedade: - *obrigatório*

Propriedade	Tipo	Descrição
permission	integer	Tipo de nível de permissão. Valores possíveis: 2 - somente leitura; 3 - leitura e gravação. Comportamento da propriedade: - *obrigatório*

Grupo de usuários do mapa

Lista de permissões de mapa com base em grupos de usuários. Possui as seguintes propriedades:

Propriedade	Tipo	Descrição
sysmapusrgrpid	ID	ID do grupo de usuários do mapa. Comportamento da propriedade: - *somente leitura*
usrgrpid	ID	ID do grupo de usuários. Comportamento da propriedade: - *obrigatório*
permission	integer	Tipo de nível de permissão. Valores possíveis: 2 - somente leitura; 3 - leitura e gravação. Comportamento da propriedade: - *obrigatório*

Formas do mapa

O objeto de forma do mapa define uma forma geométrica (com ou sem texto) exibida em um mapa. Possui as seguintes propriedades:

Propriedade	Tipo	Descrição
sysmap_shapeid	ID	ID do elemento de forma do mapa. Comportamento da propriedade: - *somente leitura*
type	integer	Tipo de elemento de forma do mapa. Valores possíveis: 0 - retângulo; 1 - elipse. A propriedade é obrigatória quando novas formas são criadas. Comportamento da propriedade: - *obrigatório*
x	integer	Coordenadas X da forma em pixels. Padrão: 0.
y	integer	Coordenadas Y da forma em pixels. Padrão: 0.

Propriedade	Tipo	Descrição
width	integer	Largura da forma em pixels. Padrão: 200.
height	integer	Altura da forma em pixels. Padrão: 200.
text	string	Texto da forma.
font	integer	Fonte do texto dentro da forma. Valores possíveis: 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 Padrão: 9.
font_size	integer	Tamanho da fonte em pixels. Padrão: 11.
font_color	string	Cor da fonte. Padrão: 000000.
text_halign	integer	Alinhamento horizontal do texto. Valores possíveis: 0 - centralizado; 1 - à esquerda; 2 - à direita. Padrão: 0.
text_valign	integer	Alinhamento vertical do texto. Valores possíveis: 0 - meio; 1 - topo; 2 - base. Padrão: 0.
border_type	integer	Tipo da borda. Valores possíveis: 0 - nenhuma; 1 - ——————; 2 - ┅; 3 - – – –. Padrão: 0.
border_width	integer	Largura da borda em pixels. Padrão: 0.
border_color	string	Cor da borda. Padrão: 000000.

Propriedade	Tipo	Descrição
background_color	string	Cor de fundo (cor de preenchimento). Padrão: `(vazio)`.
zindex	integer	Valor usado para ordenar todas as formas e linhas (z-index). Padrão: 0.

Linhas do mapa

O objeto de linha do mapa define uma linha exibida em um mapa. Ele possui as seguintes propriedades:

Propriedade	Tipo	Descrição
sysmap_shapeid	ID	ID do elemento de forma do mapa. Comportamento da propriedade: - *somente leitura*
x1	integer	Coordenadas X do ponto 1 da linha em pixels. Padrão: 0.
y1	integer	Coordenadas Y do ponto 1 da linha em pixels. Padrão: 0.
x2	integer	Coordenadas X do ponto 2 da linha em pixels. Padrão: 200.
y2	integer	Coordenadas Y do ponto 2 da linha em pixels. Padrão: 200.
line_type	integer	Tipo das linhas. Valores possíveis: 0 - nenhum; 1 - ————; 2 - ┅; 3 - − − −. Padrão: 0.
line_width	integer	Largura das linhas em pixels. Padrão: 0.
line_color	string	Cor da linha. Padrão: 000000.
zindex	integer	Valor usado para ordenar todas as formas e linhas (z-index). Padrão: 0.

map.create

Descrição

`object map.create(object/array maps)`

Este método permite criar novos mapas.

> **Note:**
> Este método está disponível para usuários de qualquer tipo. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

(object/array) Mapas a serem criados.

Além das propriedades padrão do mapa, o método aceita os seguintes parâmetros.

Parâmetro	Tipo	Descrição
links	array	Links do mapa a serem criados no mapa.
selements	array	Elementos do mapa a serem criados no mapa.
urls	array	URLs do mapa a serem criadas no mapa.
users	array	Compartilhamentos de usuário do mapa a serem criados no mapa.
userGroups	array	Compartilhamentos de grupo de usuários do mapa a serem criados no mapa.
shapes	array	Formas do mapa a serem criadas no mapa.
lines	array	Linhas do mapa a serem criadas no mapa.

Note:
Para criar links de mapa, você precisará definir um `selementid` de elemento de mapa para um valor arbitrário e, em seguida, usar esse valor para referenciar esse elemento nas propriedades `selementid1` ou `selementid2` dos links. Quando o elemento for criado, esse valor será substituído pelo ID correto gerado pelo Zabbix. Veja o exemplo.

Valores de retorno

(object) Retorna um objeto contendo os IDs dos mapas criados na propriedade `sysmapids`. A ordem dos IDs retornados corresponde à ordem dos mapas passados.

Exemplos

Criar um mapa vazio

Crie um mapa sem elementos.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "map.create",
    "params": {
        "name": "Map",
        "width": 600,
        "height": 600
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "sysmapids": [
            "8"
        ]
    },
    "id": 1
}
```

Criar um mapa de host

Crie um mapa com dois elementos de host e um link entre eles. Observe o uso de valores temporários "selementid1" e "selementid2" no objeto de link do mapa para se referir aos elementos do mapa.

Requisição:

```
{
    "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
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "sysmapids": [
            "9"
        ]
    },
    "id": 1
}
```

Criar um mapa de triggers

Crie um mapa com um elemento de trigger, que contém dois triggers.

Request:

```
{
    "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
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "sysmapids": [
            "10"
        ]
    },
    "id": 1
}
```

Compartilhamento de mapa

Crie um mapa com dois tipos de compartilhamento (usuário e grupo de usuários).

Requisição:

```
{
    "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
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "sysmapids": [
            "9"
        ]
    },
    "id": 1
}
```

Formas do mapa

Crie um mapa com o título do nome do mapa.

Requisição:

```
{
    "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
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "sysmapids": [
            "10"
        ]
    },
    "id": 1
}
```

Linhas do mapa

Crie uma linha de mapa.

Requisição:

```
{
    "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
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "sysmapids": [
            "11"
        ]
    },
```

```
    "id": 1
}
```

Veja também

- Elemento de mapa
- Link de mapa
- URL de mapa
- Usuário de mapa
- Grupo de usuários de mapa
- Forma de mapa
- Linha de mapa

Fonte

CMap::create() in *ui/include/classes/api/services/CMap.php*.

map.delete

Descrição

`object map.delete(array mapIds)`

Este método permite excluir mapas.

> **Note:**
> Este método está disponível para usuários de qualquer tipo. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(array)` IDs dos mapas a serem excluídos.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos mapas excluídos na propriedade 'sysmapids.

Exemplos

Excluir vários mapas

Exclua dois mapas.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "map.delete",
    "params": [
        "12",
        "34"
    ],
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "sysmapids": [
            "12",
            "34"
        ]
    },
    "id": 1
}
```

Fonte

CMap::delete() in *ui/include/classes/api/services/CMap.php*.

map.get

Descrição

`integer/array map.get(object parameters)`

O método permite recuperar mapas de acordo com os parâmetros fornecidos.

> **Note:**
> Este método está disponível para usuários de qualquer tipo. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object)` Parâmetros que definem a saída desejada.

O método suporta os seguintes parâmetros.

Parâmetro	Tipo	Descrição
sysmapids	ID/array	Retorna apenas mapas com os IDs fornecidos.
userids	ID/array	Retorna apenas mapas que pertencem aos IDs de usuário fornecidos.
expandUrls	flag	Adiciona URLs globais do mapa aos elementos correspondentes do mapa e expande macros em todas as URLs dos elementos do mapa.
selectIconMap	query	Retorna uma propriedade `iconmap` com o mapa de ícones usado no mapa.
selectLinks	query	Retorna uma propriedade `links` com os links do mapa entre os elementos.
selectSelements	query	Retorna uma propriedade `selements` com os elementos do mapa.
selectUrls	query	Retorna uma propriedade `urls` com as URLs do mapa.
selectUsers	query	Retorna uma propriedade `users` com os usuários com os quais o mapa é compartilhado.
selectUserGroups	query	Retorna uma propriedade `userGroups` com os grupos de usuários com os quais o mapa é compartilhado.
selectShapes	query	Retorna uma propriedade `shapes` com as formas do mapa.
selectLines	query	Retorna uma propriedade `lines` com as linhas do mapa.
sortfield	string/array	Classifica o resultado pelas propriedades fornecidas. Valores possíveis: `name`, `width`, `height`.
countOutput	boolean	Esses parâmetros são descritos no comentário de referência.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valores de retorno

`(integer/array)` Retorna:

- uma matriz de objetos;
- a contagem de objetos recuperados, se o parâmetro `countOutput` tiver sido usado.

Exemplos

Recuperar um mapa

Recupera todos os dados sobre o mapa "3".

Requisição:

```
{
    "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
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "selements": [
                {
                    "selementid": "10",
                    "sysmapid": "3",
                    "elementtype": "4",
                    "evaltype": "0",
                    "iconid_off": "1",
                    "iconid_on": "0",
                    "label": "Zabbix server",
                    "label_location": "3",
                    "x": "11",
                    "y": "141",
                    "iconid_disabled": "0",
                    "iconid_maintenance": "0",
                    "elementsubtype": "0",
                    "areatype": "0",
                    "width": "200",
                    "height": "200",
                    "tags": [
                        {
                            "tag": "service",
                            "value": "mysqld",
                            "operator": "0"
                        }
                    ],
                    "viewtype": "0",
                    "use_iconmap": "1",
                    "urls": [],
                    "elements": []
                },
                {
                    "selementid": "11",
                    "sysmapid": "3",
                    "elementtype": "4",
                    "evaltype": "0",
                    "iconid_off": "1",
                    "iconid_on": "0",
                    "label": "Web server",
                    "label_location": "3",
```

```
                "x": "211",
                "y": "191",
                "iconid_disabled": "0",
                "iconid_maintenance": "0",
                "elementsubtype": "0",
                "areatype": "0",
                "width": "200",
                "height": "200",
                "viewtype": "0",
                "use_iconmap": "1",
                "tags": [],
                "urls": [],
                "elements": []
            },
            {
                "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",
                "tags": [],
                "urls": [],
                "elements": [
                    {
                        "hostid": "10084"
                    }
                ]
            }
        ],
        "links": [
            {
                "linkid": "23",
                "sysmapid": "3",
                "selementid1": "10",
                "selementid2": "11",
                "drawtype": "0",
                "color": "00CC00",
                "label": "",
                "linktriggers": []
            }
        ],
        "users": [
            {
                "sysmapuserid": "1",
                "userid": "2",
                "permission": "2"
            }
        ],
        "userGroups": [
```

```
            {
                "sysmapusrgrpid": "1",
                "usrgrpid": "7",
                "permission": "2"
            }
        ],
        "shapes":[
            {
                "sysmap_shapeid":"1",
                "type":"0",
                "x":"0",
                "y":"0",
                "width":"680",
                "height":"15",
                "text":"{MAP.NAME}",
                "font":"9",
                "font_size":"11",
                "font_color":"000000",
                "text_halign":"0",
                "text_valign":"0",
                "border_type":"0",
                "border_width":"0",
                "border_color":"000000",
                "background_color":"",
                "zindex":"0"
            }
        ],
        "lines":[
            {
                "sysmap_shapeid":"2",
                "x1": 30,
                "y1": 10,
                "x2": 100,
                "y2": 50,
                "line_type": 1,
                "line_width": 10,
                "line_color": "009900",
                "zindex":"1"
            }
        ],
        "sysmapid": "3",
        "name": "Local network",
        "width": "400",
        "height": "400",
        "backgroundid": "0",
        "label_type": "2",
        "label_location": "3",
        "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
}
```

Veja também

- Mapa de ícones
- Elemento do mapa
- Link do mapa
- URL do mapa
- Usuário do mapa
- Grupo de usuários do mapa
- Formas do mapa
- Linhas do mapa

Fonte

CMap::get() in *ui/include/classes/api/services/CMap.php*.

map.update

Descrição

`object map.update(object/array maps)`

Este método permite atualizar mapas existentes.

> **Note:**
> Este método está disponível para usuários de qualquer tipo. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object/array)` Propriedades do mapa a serem atualizadas.

A propriedade `mapid` deve ser definida para cada mapa, todas as outras propriedades são opcionais. Somente as propriedades passadas serão atualizadas, todas as outras permanecerão inalteradas.

Além das propriedades padrão do mapa, o método aceita os seguintes parâmetros.

Parâmetro	Tipo	Descrição
links	array	Links do mapa para substituir os links existentes.
selements	array	Elementos do mapa para substituir os elementos existentes.
urls	array	URLs do mapa para substituir as URLs existentes.
users	array	Compartilhamentos de usuários do mapa para substituir os elementos existentes.
userGroups	array	Compartilhamentos de grupos de usuários do mapa para substituir os elementos existentes.
shapes	array	Formas do mapa para substituir as formas existentes.
lines	array	Linhas do mapa para substituir as linhas existentes.

Note:
Para criar links de mapa entre novos elementos do mapa, você precisará definir o `selementid` de um elemento para um valor arbitrário e então usar esse valor para referenciar esse elemento nas propriedades `selementid1` ou `selementid2` dos links. Quando o elemento for criado, esse valor será substituído pelo ID correto gerado pelo Zabbix. Veja o exemplo para map.create.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos mapas atualizados na propriedade `sysmapids`.

Exemplos

Redimensionar um mapa

Altere o tamanho do mapa para 1200x1200 pixels.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "map.update",
    "params": {
        "sysmapid": "8",
        "width": 1200,
        "height": 1200
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "sysmapids": [
            "8"
        ]
    },
    "id": 1
}
```

Alterar o proprietário do mapa

Disponível apenas para administradores e superadministradores.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "map.update",
    "params": {
        "sysmapid": "9",
        "userid": "1"
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "sysmapids": [
            "9"
        ]
    },
    "id": 1
}
```

Veja também

- Elemento de mapa
- Link de mapa
- URL de mapa
- Usuário de mapa
- Grupo de usuários de mapa
- Formas de mapa
- Linhas de mapa

Fonte

CMap::update() in *ui/include/classes/api/services/CMap.php*.

Mapa de valores

Esta classe foi projetada para trabalhar com mapas de valores.

Referências de objetos:

- Mapa de valores
  - Mapeamentos de valores

Métodos disponíveis:

- valuemap.create - criar novos mapas de valores
- valuemap.delete - excluir mapas de valores
- valuemap.get - recuperar mapas de valores
- valuemap.update - atualizar mapas de valores

Objeto de mapa de valores

Os seguintes objetos estão diretamente relacionados à API `valuemap`.

Mapa de valores

O objeto mapa de valores possui as seguintes propriedades.

Propriedade	Tipo	Descrição
valuemapid	ID	ID do mapa de valores. Comportamento da propriedade: - *somente leitura* - *obrigatório* para operações de atualização
hostid	ID	ID do host ou template ao qual o mapa de valores pertence. Comportamento da propriedade: - *constante* - *obrigatório* para operações de criação
name	string	Nome do mapa de valores. Comportamento da propriedade: - *obrigatório* para operações de criação
mappings	array	Mapeamentos de valores para o mapa de valores atual. O objeto de mapeamento é descrito em detalhes abaixo. Comportamento da propriedade: - *obrigatório* para operações de criação
uuid	string	Identificador único universal, usado para vincular mapas de valores importados aos já existentes. Gerado automaticamente, se não for fornecido. Comportamento da propriedade: - *suportado* se o mapa de valores pertencer a um template

Mapeamentos de valores

O objeto de mapeamentos de valores define os mapeamentos de valores do mapa de valores. Ele possui as seguintes propriedades.

Propriedade	Tipo	Descrição
type	integer	Tipo de correspondência do mapeamento. Valores possíveis: 0 - *(padrão)* o mapeamento será aplicado se o valor for igual; 1 - o mapeamento será aplicado se o valor for maior ou igual[1]; 2 - o mapeamento será aplicado se o valor for menor ou igual[1]; 3 - o mapeamento será aplicado se o valor estiver em um intervalo (os intervalos são inclusivos; vários intervalos, separados por vírgula, podem ser definidos)[1]; 4 - o mapeamento será aplicado se o valor corresponder a uma expressão regular[2]; 5 - se nenhuma correspondência for encontrada, o mapeamento não será aplicado e o valor padrão será usado. Se `type` for definido como "0", "1", "2", "3", "4", então `value` não pode estar vazio. Se `type` for definido como "5", então `value` deve estar vazio.
value	string	Valor original. Comportamento da propriedade: - *obrigatório* se `type` for definido como "1", "2", "3", "4" - *suportado* se `type` for definido como "5"
newvalue	string	Valor para o qual o valor original é mapeado. Comportamento da propriedade: - *obrigatório*

[1] suportado apenas para items com tipo de valor "numérico sem sinal", "numérico flutuante".
 [2] suportado apenas para items com tipo de valor "caractere".

valuemap.create

Descrição

`object valuemap.create(object/array valuemaps)`

Este método permite criar novos mapas de valores.

> **Note:**
> Este método está disponível apenas para o tipo de usuário *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object/array)` Mapas de valores a serem criados.

O método aceita mapas de valores com as propriedades padrão do mapa de valores.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos mapas de valores criados na propriedade `valuemapids`. A ordem dos IDs retornados corresponde à ordem dos mapas de valores passados.

Exemplos

Criando um mapa de valores

Crie um mapa de valores com dois mapeamentos.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "valuemap.create",
    "params": {
        "hostid": "50009",
        "name": "Service state",
        "mappings": [
            {
                "type": "1",
                "value": "1",
                "newvalue": "Up"
            },
            {
                "type": "5",
                "newvalue": "Down"
            }
        ]
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "valuemapids": [
            "1"
        ]
    },
    "id": 1
}
```

Fonte

CValueMap::create() in *ui/include/classes/api/services/CValueMap.php*.

valuemap.delete

Descrição

`object valuemap.delete(array valuemapids)`

Este método permite excluir mapas de valores.

> **Note:**
> Este método está disponível apenas para o tipo de usuário *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Consulte Funções de usuário para mais informações.

Parâmetros

`(array)` IDs dos valores de mapa a serem deletados.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos mapas de valores excluídos na propriedade `valuemapids`.

Exemplos

Excluindo vários mapas de valores

Exclua dois mapas de valores.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "valuemap.delete",
```

```
    "params": [
        "1",
        "2"
    ],
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "valuemapids": [
            "1",
            "2"
        ]
    },
    "id": 1
}
```

Fonte

CValueMap::delete() in *ui/include/classes/api/services/CValueMap.php*.

valuemap.get

Descrição

`integer/array valuemap.get(object parameters)`

O método permite recuperar mapas de valores de acordo com os parâmetros fornecidos.

> **Note:**
> Este método está disponível para usuários de qualquer tipo. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object)` Parâmetros que definem a saída desejada.

O método suporta os seguintes parâmetros.

Parâmetro	Tipo	Descrição
valuemapids	ID/array	Retornar apenas mapas de valores com os IDs fornecidos.
selectMappings	query	Retornar os mapeamentos de valores para o mapa de valores atual na propriedade `mappings`. Suporta `count`.
sortfield	string/array	Ordenar o resultado pelas propriedades fornecidas. Valores possíveis: `valuemapid`, `name`.
countOutput	boolean	Esses parâmetros são descritos no comentário de referência.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valores de retorno

`(integer/array)` Retorna:

- um array de objetos;
- a contagem de objetos recuperados, se o parâmetro `countOutput` foi usado.

Exemplos

Recuperando mapas de valores

Recupere todos os mapas de valores configurados.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "valuemap.get",
    "params": {
        "output": "extend"
    },
    "id": 1
}
```

Resposta:

```
{
    "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
}
```

Recupere um mapa de valores com seus mapeamentos.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "valuemap.get",
    "params": {
        "output": "extend",
        "selectMappings": "extend",
        "valuemapids": ["4"]
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "valuemapid": "4",
            "name": "APC Battery Replacement Status",
            "mappings": [
```

```
                {
                    "type": "0",
                    "value": "1",
                    "newvalue": "unknown"
                },
                {
                    "type": "0",
                    "value": "2",
                    "newvalue": "notInstalled"
                },
                {
                    "type": "0",
                    "value": "3",
                    "newvalue": "ok"
                },
                {
                    "type": "0",
                    "value": "4",
                    "newvalue": "failed"
                },
                {
                    "type": "0",
                    "value": "5",
                    "newvalue": "highTemperature"
                },
                {
                    "type": "0",
                    "value": "6",
                    "newvalue": "replaceImmediately"
                },
                {
                    "type": "0",
                    "value": "7",
                    "newvalue": "lowCapacity"
                }
            ]
        }
    ],
    "id": 1
}
```


Fonte

CValueMap::get() in *ui/include/classes/api/services/CValueMap.php*.

valuemap.update

Descrição

`object valuemap.update(object/array valuemaps)`

Este método permite atualizar mapas de valores existentes.

> **Note:**
> Este método está disponível apenas para o tipo de usuário *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função de usuário. Consulte Funções de usuário para mais informações.

Parâmetros

`(object/array)` Propriedades do mapa de valores a serem atualizadas.

A propriedade `valuemapid` deve ser definida para cada mapa de valores, todas as outras propriedades são opcionais. Somente as propriedades passadas serão atualizadas, todas as outras permanecerão inalteradas.

Valores de retorno

(objeto) Retorna um objete que contém os IDs de uma atualização de valor mapeia sob a propriedade `valuemapids`.

Exemplos

Alterando o nome do mapa de valores

Altere o nome do mapa de valores para ”Device status”.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "valuemap.update",
    "params": {
        "valuemapid": "2",
        "name": "Device status"
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "valuemapids": [
            "2"
        ]
    },
    "id": 1
}
```

Alterando mapeamentos para um mapa de valores.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "valuemap.update",
    "params": {
        "valuemapid": "2",
        "mappings": [
            {
                "type": "0",
                "value": "0",
                "newvalue": "Online"
            },
            {
                "type": "0",
                "value": "1",
                "newvalue": "Offline"
            }
        ]
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "valuemapids": [
            "2"
        ]
    },
    "id": 1
}
```

Fonte

CValueMap::update() in *ui/include/classes/api/services/CValueMap.php*.

Mapa de ícones

Esta classe foi projetada para trabalhar com mapas de ícones.

Referências de objetos:

- Mapa de ícones
- Mapeamento de ícones

Métodos disponíveis:

- iconmap.create - criar novos mapas de ícones
- iconmap.delete - excluir mapas de ícones
- iconmap.get - recuperar mapas de ícones
- iconmap.update - atualizar mapas de ícones

Objeto de mapa de ícones

Os seguintes objetos estão diretamente relacionados à API `iconmap`.

Mapa de ícones

O objeto mapa de ícones possui as seguintes propriedades.

Propriedade	Tipo	Descrição
iconmapid	ID	ID do mapa de ícones. Comportamento da propriedade: - *somente leitura* - *obrigatório* para operações de atualização
default_iconid	ID	ID do ícone padrão. Comportamento da propriedade: - *obrigatório* para operações de criação
name	string	Nome do mapa de ícones. Comportamento da propriedade: - *obrigatório* para operações de criação

Mapeamento de ícones

O objeto de mapeamento de ícones define um ícone específico a ser usado para hosts com um determinado valor de campo de inventário. Ele possui as seguintes propriedades.

Propriedade	Tipo	Descrição
iconid	ID	ID do ícone usado pelo mapeamento de ícones. Comportamento da propriedade: - *obrigatório*
expression	string	Expressão para comparar com o campo de inventário. Comportamento da propriedade: - *obrigatório*
inventory_link	integer	ID do campo de inventário do host. Consulte o objeto de inventário do host para obter uma lista de campos de inventário suportados. Comportamento da propriedade: - *obrigatório*

Propriedade	Tipo	Descrição
sortorder	integer	Posição do mapeamento de ícones no mapa de ícones. Comportamento da propriedade: - *somente leitura*

iconmap.create

Descrição

`object iconmap.create(object/array iconMaps)`

Este método permite criar novos mapas de ícones.

> **Note:**
> Este método está disponível apenas para o tipo de usuário *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função de usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object/array)` Mapas de ícones a serem criados.

Além das propriedades padrão do mapa de ícones, o método aceita os seguintes parâmetros.

Parâmetro	Tipo	Descrição
mappings	array	Mapeamentos de ícones a serem criados para o mapa de ícones. Comportamento do parâmetro: - *obrigatório*

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos mapas de ícones criados sob a propriedade `iconmapids`. A ordem dos IDs retornados corresponde a ordem dos mapas de ícones passados.

Exemplos

Criar um mapa de ícones

Crie um mapa de ícones para exibir hosts de diferentes tipos.

Requisição:

```
{
    "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
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "iconmapids": [
            "2"
        ]
    },
    "id": 1
}
```

Veja também

- Mapeamento de ícones

Fonte

CIconMap::create() em *ui/include/classes/api/services/CIconMap.php*.

iconmap.delete

Descrição

`object iconmap.delete(array iconMapIds)`

Este método permite excluir mapas de ícones.

> **Note:**
> Este método está disponível apenas para o tipo de usuário *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Consulte Funções de usuário para mais informações.

Parâmetros

`(array)` IDs dos mapas de ícones a serem excluídos.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos mapas de ícones excluídos sob a propriedade `iconmapids`.

Exemplos

Excluir vários mapas de ícones

Exclua dois mapas de ícones.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "iconmap.delete",
    "params": [
        "2",
        "5"
    ],
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "iconmapids": [
            "2",
            "5"
        ]
    },
    "id": 1
}
```

Fonte

CIconMap::delete() em *ui/include/classes/api/services/CIconMap.php*.

iconmap.get

Descrição

`integer/array iconmap.get(object parameters)`

O método permite recuperar mapas de ícones de acordo com os parâmetros fornecidos.

> **Note:**
> Este método está disponível para usuários de qualquer tipo. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object)` Parâmetros que definem a saída desejada.

O método suporta os seguintes parâmetros.

Parâmetro	Tipo	Descrição
iconmapids	ID/array	Retornar apenas mapas de ícones com os IDs fornecidos.
sysmapids	ID/array	Retornar apenas mapas de ícones que são usados nos mapas fornecidos.
selectMappings	query	Retornar uma propriedade `mappings` com os mapeamentos de ícones usados.
sortfield	string/array	Ordenar o resultado pelas propriedades fornecidas. Valores possíveis: `iconmapid`, `name`.
countOutput	boolean	Estes parâmetros são descritos no comentário de referência.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valores de retorno

`(integer/array)` Retorna:

- uma matriz de objetos;
- a contagem de objetos recuperados, se o parâmetro `countOutput` tiver sido usado.

Exemplos

Recuperar um mapa de ícones

Recupere todos os dados sobre o mapa de ícones "3".

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "iconmap.get",
    "params": {
        "iconmapids": "3",
        "output": "extend",
        "selectMappings": "extend"
    },
```

```
    "id": 1
}
```

Resposta:

```
{
    "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
}
```

Veja também

- Mapeamento de ícones

Fonte

CIconMap::get() em *ui/include/classes/api/services/CIconMap.php*.

iconmap.update

Descrição

`object iconmap.update(object/array iconMaps)`

Este método permite atualizar mapas de ícones existentes.

> **Note:**
> Este método está disponível apenas para o tipo de usuário *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função de usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object/array)` Propriedades do mapa de ícones a serem atualizadas.

A propriedade `iconmapid` deve ser definida para cada mapa de ícones, todas as outras propriedades são opcionais. Somente as propriedades passadas serão atualizadas, todas as outras permanecerão inalteradas.

Além das propriedades padrão do mapa de ícones, o método aceita os seguintes parâmetros.

Parâmetro	Tipo	Descrição
mappings	array	Mapeamentos de ícones para substituir os mapeamentos de ícones existentes.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos mapas de ícones atualizados sob a propriedade `iconmapids`.

Exemplos

Renomear mapa de ícones

Renomear um mapa de ícones para ”OS icons”.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "iconmap.update",
    "params": {
        "iconmapid": "1",
        "name": "OS icons"
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "iconmapids": [
            "1"
        ]
    },
    "id": 1
}
```

Veja também

- Mapeamento de ícones

Fonte

CIconMap::update() em *ui/include/classes/api/services/CIconMap.php*.

MFA

Esta classe foi projetada para trabalhar com métodos MFA (Autenticação Multifator).

Referências de objetos:

- MFA

Métodos disponíveis:

- mfa.create - criar novos métodos MFA
- mfa.delete - excluir métodos MFA
- mfa.get - recuperar métodos MFA
- mfa.update - atualizar métodos MFA

Objeto MFA

Os seguintes objetos estão diretamente relacionados à API `mfa`.

MFA

O objeto MFA (Autenticação Multifator) possui as seguintes propriedades.

Propriedade	Tipo	Descrição
mfaid	ID	ID do método MFA. Comportamento da propriedade: - *somente leitura* - *obrigatório* para operações de atualização
type	integer	Tipo do método MFA. Valores possíveis: 1 - TOTP (Senhas de uso único baseadas em tempo); 2 - Duo Universal Prompt.
name	string	Nome exclusivo do método MFA. Comportamento da propriedade: - *obrigatório* para operações de criação
hash_function	integer	Tipo de função hash para geração de códigos TOTP. Valores possíveis: 1 - SHA-1; 2 - SHA-256; 3 - SHA-512. Comportamento da propriedade: - *obrigatório* se `type` estiver definido como ”TOTP”
code_length	integer	Comprimento do código de verificação. Valores possíveis: 6 - 6 dígitos; 8 - 8 dígitos. Comportamento da propriedade: - *obrigatório* se `type` estiver definido como ”TOTP”
api_hostname	string	Nome do host da API fornecido pelo serviço de autenticação Duo. Comportamento da propriedade: - *obrigatório* se `type` estiver definido como ”Duo Universal Prompt”
clientid	string	ID do cliente fornecido pelo serviço de autenticação Duo. Comportamento da propriedade: - *obrigatório* se `type` estiver definido como ”Duo Universal Prompt”
client_secret	string	Segredo do cliente fornecido pelo serviço de autenticação Duo. Comportamento da propriedade: - *somente gravação* - *obrigatório* se `type` estiver definido como ”Duo Universal Prompt”

mfa.create

Descrição

`object mfa.create(object/array Métodos MFA)`

Este método permite criar novos métodos MFA.

> **Note:**
> Este método está disponível apenas para o tipo de usuário *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Consulte Funções de usuário para mais informações.

Parâmetros

`(object/array)` Métodos MFA a serem criados.

O método aceita métodos MFA com as propriedades padrão do método MFA.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos métodos MFA criados na propriedade `mfaids`. A ordem dos IDs retornados corresponde à ordem dos itens passados.

Exemplos

Criando métodos de MFA

Crie um método de MFA "Zabbix TOTP" utilizando senhas únicas baseadas em tempo (TOTP), com a função hash para gerar códigos TOTP definida como SHA-1 e o comprimento do código de verificação definido como 6 dígitos.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "mfa.create",
    "params": {
        "type": 1,
        "name": "Zabbix TOTP",
        "hash_function": 1,
        "code_length": 6
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "mfaids": [
            "1"
        ]
    },
    "id": 1
}
```

Fonte

CMfa::create() em *ui/include/classes/api/services/CMfa.php*.

mfa.delete

Descrição

`object mfa.delete(array mfaids)`

Este método permite excluir métodos MFA.

> **Note:**
> Este método está disponível apenas para o tipo de usuário *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Consulte Funções de usuário para mais informações.

Parâmetros

`(array)` IDs dos métodos MFA a serem excluídos.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos métodos MFA excluídos na propriedade `mfaids`.

Exemplos

Excluindo métodos MFA

Exclua um método MFA.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "mfa.delete",
    "params": [
        "2"
    ],
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "mfaids": [
            "2"
        ]
    },
    "id": 1
}
```

Fonte

CMfa::delete() em *ui/include/classes/api/services/CMfa.php*.

mfa.get

Descrição

`integer/array mfa.get(object parameters)`

O método permite recuperar métodos MFA de acordo com os parâmetros fornecidos.

> **Note:**
> Este método está disponível apenas para o tipo de usuário *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Consulte Funções de usuário para obter mais informações.

Parâmetros

`(object)` Parâmetros que definem a saída desejada.

O método suporta os seguintes parâmetros.

Parâmetro	Tipo	Descrição
mfaids	ID/array	Retorna apenas métodos MFA com os IDs fornecidos.
selectUsrgrps	query	Retorna uma propriedade `usrgrps` com grupos de usuários associados aos métodos MFA. Suporta `count`.
filter	object	Retorna apenas os resultados que correspondem exatamente ao filtro fornecido. Aceita um objeto, onde as chaves são nomes de propriedades e os valores são um único valor ou um array de valores para comparar. Suporta propriedades: `mfaid` - ID do método MFA; `type` - Tipo do método MFA.
sortfield	string/array	Ordena o resultado pelas propriedades fornecidas. Valores possíveis: `name`.

Parâmetro	Tipo	Descrição
search	object	Retorna resultados que correspondam ao padrão fornecido (case-insensitive). Aceita um objeto, onde as chaves são nomes de propriedades e os valores são strings a serem pesquisadas. Se nenhuma opção adicional for fornecida, será realizada uma busca `LIKE "%…%"`. Propriedades suportadas: `name`.
countOutput	boolean	Esses parâmetros são descritos no comentário de referência.
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valores de retorno

`(integer/array)` Retorna:

- um array de objetos;
- a contagem de objetos recuperados, se o parâmetro `countOutput` tiver sido usado.

Exemplos

Encontrando métodos MFA por nome

Recupere todos os métodos MFA com ”Zabbix” no nome.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "mfa.get",
    "params": {
        "output": "extend",
        "search": {
            "name": "Zabbix"
        }
    },
    "id": 1
}
```

Resposta:

```
{
    "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
}
```

Fonte

CMfa::get() em *ui/include/classes/api/services/CMfa.php*.

mfa.update

Descrição

`object mfa.update(object/array MFA methods)`

Este método permite atualizar métodos MFA existentes.

> **Note:**
> Este método está disponível apenas para o tipo de usuário *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Consulte Funções de usuário para mais informações.

Parâmetros

`(object/array)` Propriedades do método MFA a serem atualizadas.

A propriedade `mfaid` deve ser definida para cada item, todas as outras propriedades são opcionais. Somente as propriedades passadas serão atualizadas, todas as outras permanecerão inalteradas.

O método aceita métodos MFA com as propriedades padrão do método MFA.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos métodos MFA atualizados na propriedade `mfaids`.

Exemplos

Atualizando as propriedades do método

Atualize a função hash para gerar códigos TOTP e o comprimento do código de verificação para o método MFA "Zabbix TOTP" que utiliza senhas descartáveis baseadas em tempo (TOTP).

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "mfa.update",
    "params": {
        "mfaid": "1",
        "hash_function": 3,
        "code_length": 8
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "mfaids": [
            "1"
        ]
    },
    "id": 1
}
```

Fonte

CMfa::update() em *ui/include/classes/api/services/CMfa.php*.

Módulo

Esta classe foi projetada para trabalhar com módulos do frontend.

Referências de objetos:

- Module

Métodos disponíveis:

- module.create - instalar novos módulos
- module.delete - desinstalar módulos
- module.get - recuperar módulos
- module.update - atualizar módulos

Objeto module

Os seguintes objetos estão diretamente relacionados à API `module`.

Módulo

O objeto módulo possui as seguintes propriedades.

Propriedade	Tipo	Descrição
moduleid	ID	ID do módulo conforme armazenado no banco de dados. Comportamento da propriedade: - *somente leitura* - *obrigatório* para operações de atualização
id	string	ID único do módulo conforme definido por um desenvolvedor no arquivo manifest.json do módulo. Valores possíveis para módulos internos: veja a descrição da propriedade "type" em Widget do dashboard. Comportamento da propriedade: - *obrigatório* para operações de criação
relative_path	string	Caminho para o diretório do módulo relativo ao diretório do frontend do Zabbix. Valores possíveis: `widgets/*` - para módulos de widget internos; `modules/*` - para módulos de terceiros. Comportamento da propriedade: - *obrigatório* para operações de criação
status	integer	Se o módulo está habilitado ou desabilitado. Valores possíveis: 0 - *(padrão)* Desabilitado; 1 - Habilitado.
config	object	Configuração do módulo.

module.create

Descrição

`object module.create(object/array modules)`

Este método permite instalar novos módulos de frontend.

Note:
Este método está disponível apenas para o tipo de usuário *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Consulte Funções de usuário para mais informações.

Attention:
Os arquivos do módulo devem ser descompactados manualmente nos subdiretórios corretos, correspondendo à propriedade `relative_path` dos módulos.

Parâmetros

`(object/array)` Módulos a serem criados.

O método aceita módulos com as propriedades padrão do módulo.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos módulos instalados na propriedade `moduleids`. A ordem dos IDs retornados corresponde à ordem dos módulos passados.

Exemplos

Instalando um módulo

Instale um módulo com o status "Ativado".

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "module.create",
    "params": {
        "id": "example_module",
        "relative_path": "modules/example_module",
        "status": 1
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "moduleids": [
            "25"
        ]
    },
    "id": 1
}
```

Veja também

- Módulo
- Módulos do frontend

Fonte

CModule::create() em *ui/include/classes/api/services/CModule.php*.

module.delete

Descrição

`object module.delete(array moduleids)`

Este método permite desinstalar módulos.

Note:
Este método está disponível apenas para o tipo de usuário *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função de usuário. Consulte Funções de usuário para mais informações.

Attention:
Os arquivos do módulo devem ser removidos manualmente.

Parâmetros

`(array)` IDs dos módulos a serem desinstalados.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos módulos desinstalados na propriedade `moduleids`.

Exemplos

Desinstalando múltiplos módulos

Desinstale os módulos "27" e "28".

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "module.delete",
    "params": [
        "27",
        "28"
    ],
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "moduleids": [
            "27",
            "28"
        ]
    },
    "id": 1
}
```

Fonte

CModule::delete() em *ui/include/classes/api/services/CModule.php*.

module.get

Descrição

`integer/array module.get(object parameters)`

O método permite recuperar módulos de acordo com os parâmetros fornecidos.

Note:
Este método está disponível apenas para o tipo de usuário *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função de usuário. Consulte Funções de usuário para mais informações.

Parâmetros

`(object)` Parâmetros que definem a saída desejada.

O método suporta os seguintes parâmetros.

Parâmetro	Tipo	Descrição
moduleids	ID/array	Retorna apenas os módulos com os IDs fornecidos.
sortfield	string/array	Ordena o resultado pelas propriedades fornecidas. Valores possíveis: `moduleid`, `relative_path`.
countOutput	boolean	Estes parâmetros são descritos em Comentário de referência.
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valores de retorno

`(integer/array)` Retorna:

- um array de objetos;
- a contagem de objetos recuperados, se o parâmetro `countOutput` tiver sido usado.

Exemplos

Recuperando um módulo pelo ID

Recupere todos os dados sobre os módulos "1", "2" e "25".

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "module.get",
    "params": {
        "output": "extend",
        "moduleids": [
            "1",
            "2",
            "25"
        ]
    },
    "id": 1
}
```

Resposta:

```
{
    "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
}
```

Veja também

- Módulo
- Widget de dashboard
- Módulos do frontend

Fonte

CModule::get() em *ui/include/classes/api/services/CModule.php*.

module.update

Descrição

`object module.update(object/array modules)`

Este método permite atualizar módulos existentes.

> **Note:**
> Este método está disponível apenas para o tipo de usuário *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Consulte Funções de usuário para mais informações.

Parâmetros

`(object/array)` Propriedades do módulo a serem atualizadas.

A propriedade `moduleid` deve ser definida para cada módulo, todas as outras propriedades são opcionais. Apenas as propriedades especificadas serão atualizadas.

O método aceita módulos com as propriedades padrão do módulo.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos módulos atualizados na propriedade `moduleids`.

Exemplos

Desabilitando um módulo

Desabilite o módulo "25".

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "module.update",
    "params": {
        "moduleid": "25",
        "status": 0
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "moduleids": [
            "25"
```

```
        ]
    },
    "id": 1
}
```

Veja também

- Módulo
- Módulos do frontend

Fonte

CModule::update() em *ui/include/classes/api/services/CModule.php*.

Nó de alta disponibilidade

Esta classe foi projetada para trabalhar com nós de servidor que fazem parte de um cluster de alta disponibilidade ou uma instância de servidor independente.

Referências de objetos:

- Nó de alta disponibilidade

Métodos disponíveis:

- hanode.get - recuperar nós

Objeto de nó de alta disponibilidade

O objeto a seguir está relacionado à operação de um cluster de alta disponibilidade de servidores Zabbix.

Nó de alta disponibilidade

> **Note:**
> Os nós são criados pelo Zabbix server e não podem ser modificados via API.

O objeto Nó de alta disponibilidade possui as seguintes propriedades.

Propriedade	Tipo	Descrição
ha_nodeid	ID	ID do nó.
name	string	Nome atribuído ao nó, usando a entrada de configuração HANodeName do zabbix_server.conf. Vazio para um server executando em modo standalone.
address	string	IP ou nome DNS de onde o nó se conecta.
port	integer	Porta na qual o nó está em execução.
lastaccess	integer	Tempo de heartbeat, ou seja, tempo da última atualização do nó. Timestamp UTC.
status	integer	Estado do nó. Valores possíveis: 0 - standby; 1 - parado manualmente; 2 - indisponível; 3 - ativo.

hanode.get

Descrição

`integer/array hanode.get(object parameters)`

O método permite recuperar uma lista de nós de cluster de alta disponibilidade de acordo com os parâmetros fornecidos.

Note:
Este método está disponível apenas para tipos de usuário *Super admin*. Veja Funções de usuário para mais informações.

Parâmetros

`(object)` Parâmetros que definem a saída desejada.

O método suporta os seguintes parâmetros.

Parâmetro	Tipo	Descrição
ha_nodeids	ID/array	Retorna apenas os nodes com os IDs de node fornecidos.
filter	object	Retorna apenas os resultados que correspondem exatamente ao filtro fornecido. Aceita um objeto, onde as chaves são nomes de propriedades e os valores são um único valor ou um array de valores para comparar. Suporta as propriedades: `name`, `address`, `status`.
sortfield	string/array	Ordena o resultado pelas propriedades fornecidas. Valores possíveis: `name`, `lastaccess`, `status`.
countOutput	flag	Esses parâmetros são descritos no comentário de referência.
limit	integer	
output	query	
preservekeys	boolean	
sortorder	string/array	

Valores de retorno

`(integer/array)` Retorna:

- uma matriz de objetos;
- a contagem de objetos recuperados, se o parâmetro `countOutput` tiver sido usado.

Exemplos

Obter uma lista de nós ordenados por status

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "hanode.get",
    "params": {
        "preservekeys": true,
        "sortfield": "status",
        "sortorder": "DESC"
    },
    "id": 1
}
```

Resposta:

```
{
  "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
}
```


Obter uma lista de nós específicos por seus IDs

Requisição:


```
{
    "jsonrpc": "2.0",
    "method": "hanode.get",
    "params": {
        "ha_nodeids": ["ckuo7i1nw000e0sajwfttc1mp", "ckuo7i1nv000c0sajz85xcrtt"]
    },
    "id": 1
}
```


Resposta:


```
{
    "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
}
```


Obter uma lista de nós parados

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "hanode.get",
    "params": {
        "output": ["ha_nodeid", "address", "port"],
        "filter": {
            "status": 1
        }
    },
    "id": 1
}
```

Resposta:

```
{
    "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
}
```

Obter uma contagem de nós em espera

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "hanode.get",
    "params": {
        "countOutput": true,
        "filter": {
            "status": 0
        }
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": "3",
    "id": 1
}
```

Verificar o status dos nodes em endereços IP específicos

Requisição:

```
{
    "jsonrpc": "2.0",
```

```
    "method": "hanode.get",
    "params": {
        "output": ["name", "status"],
        "filter": {
            "address": ["192.168.1.7", "192.168.1.13"]
        }
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "name": "node3",
            "status": "0"
        },
        {
            "name": "node-active",
            "status": "3"
        }
    ],
    "id": 1
}
```

Fonte

CHaNode::get() em *ui/include/classes/api/services/CHaNode.php*.

Problem

Esta classe foi projetada para trabalhar com problemas.

Referências de objetos:

- Problem
  - Media type URL
- Problem tag

Métodos disponíveis:

- problem.get - recuperar problemas

Objeto problem

Os seguintes objetos estão diretamente relacionados à API `problem`.

Problema

> **Note:**
> Problemas são criados pelo Zabbix server e não podem ser modificados via API.

O objeto problema possui as seguintes propriedades.

Propriedade	Tipo	Descrição
eventid	ID	ID do evento do problema.
source	integer	Tipo do evento do problema. Valores possíveis: 0 - evento criado por um trigger; 3 - evento interno; 4 - evento criado na atualização do status do serviço.

Propriedade	Tipo	Descrição
object	integer	Tipo de objeto relacionado ao evento do problema. Valores possíveis se `source` for definido como "evento criado por um trigger": 0 - trigger. Valores possíveis se `source` for definido como "evento interno": 0 - trigger; 4 - item; 5 - regra LLD. Valores possíveis se `source` for definido como "evento criado na atualização do status do serviço": 6 - serviço.
objectid	ID	ID do objeto relacionado.
clock	timestamp	Hora em que o evento do problema foi criado.
ns	integer	Nanosegundos em que o evento do problema foi criado.
r_eventid	ID	ID do evento de recuperação.
r_clock	timestamp	Hora em que o evento de recuperação foi criado.
r_ns	integer	Nanosegundos em que o evento de recuperação foi criado.
cause_eventid	ID	ID do evento de causa.
correlationid	ID	ID da regra de correlação se este evento foi recuperado por uma regra de correlação global.
userid	ID	ID do usuário que fechou o problema (se o problema foi fechado manualmente).
name	string	Nome do problema resolvido.
acknowledged	integer	Estado de reconhecimento do problema. Valores possíveis: 0 - não reconhecido; 1 - reconhecido.
severity	integer	Gravidade atual do problema. Valores possíveis: 0 - não classificado; 1 - informação; 2 - aviso; 3 - médio; 4 - alto; 5 - desastre.
suppressed	integer	Se o problema está suprimido. Valores possíveis: 0 - problema está em estado normal; 1 - problema está suprimido.
opdata	string	Dados operacionais com macros expandidas.
urls	array	URLs de tipos de mídia ativas.

URL do tipo de mídia

O objeto URL do tipo de mídia possui as seguintes propriedades.

Propriedade	Tipo	Descrição
name	string	Nome da URL definida pelo tipo de mídia.
url	string	Valor da URL definida pelo tipo de mídia.

Os resultados conterão entradas apenas para tipos de mídia ativos com entrada de menu de evento habilitada. A macro usada nas propriedades será expandida, mas se uma das propriedades contiver uma macro não expandida, ambas as propriedades serão excluídas dos resultados. Para macros suportadas, consulte *Macros suportadas*.

Tag de problema

O objeto de tag de problema possui as seguintes propriedades.

Propriedade	Tipo	Descrição
tag	string	Nome da tag de problema.
value	string	Valor da tag de problema.

problem.get

Descrição

`integer/array problem.get(object parameters)`

O método permite recuperar problemas de acordo com os parâmetros fornecidos.

Este método serve para recuperar problemas não resolvidos. Também é possível, se especificado, recuperar adicionalmente problemas recentemente resolvidos. O período que determina o quão antigo é "recentemente" é definido em *Administração → Geral*. Problemas que foram resolvidos antes desse período não são mantidos na tabela de problemas. Para recuperar problemas que foram resolvidos há mais tempo, use o método `event.get`.

Attention:
Este método pode retornar problemas de uma entidade excluída se esses problemas ainda não tiverem sido removidos pelo housekeeper.

Note:
Este método está disponível para usuários de qualquer tipo. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object)` Parâmetros que definem a saída desejada.

O método suporta os seguintes parâmetros.

Parâmetro	Tipo	Descrição
eventids	ID/array	Retorna apenas problemas com os IDs fornecidos.
groupids	ID/array	Retorna apenas problemas criados por objetos que pertencem aos grupos de hosts fornecidos.
hostids	ID/array	Retorna apenas problemas criados por objetos que pertencem aos hosts fornecidos.
objectids	ID/array	Retorna apenas problemas criados pelos objetos fornecidos.
source	integer	Retorna apenas problemas com o tipo fornecido. Consulte a página do objeto de evento de problema para uma lista de tipos de eventos suportados. Padrão: 0 - problema criado por um trigger.
object	integer	Retorna apenas problemas criados por objetos do tipo fornecido. Consulte a página do objeto de evento de problema para uma lista de tipos de objetos suportados. Padrão: 0 - trigger.
acknowledged	boolean	Se definido como `true`, retorna apenas problemas reconhecidos.
action	integer	Retorna apenas problemas para os quais as ações de atualização de evento fornecidas foram executadas. Para múltiplas ações, use a soma de quaisquer valores de bitmap aceitáveis como máscara de bits (por exemplo, 34 para reconhecer e suprimir evento).
action_userids	ID/array	Retorna apenas problemas com os IDs fornecidos de usuários que executaram as ações de atualização de evento de problema.
suppressed	boolean	Se definido como `true`, retorna apenas problemas suprimidos.

Parâmetro	Tipo	Descrição
symptom	boolean	Se definido como `true`, retorna apenas eventos de problema do tipo sintoma.
severities	integer/array	Retorna apenas problemas com as gravidades de evento fornecidas. Aplica-se apenas se o objeto for trigger.
evaltype	integer	Método de avaliação da tag. Valores possíveis: 0 - *(padrão)* E/Ou; 2 - Ou.
tags	array	Retorna apenas problemas com as tags fornecidas. Formato: `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`. Um array vazio retorna todos os problemas. Valores possíveis para operator: 0 - *(padrão)* Contém; 1 - Igual; 2 - Não contém; 3 - Não é igual; 4 - Existe; 5 - Não existe.
recent	boolean	Se definido como `true`, retorna apenas problemas resolvidos recentemente (depende de `ok_period`).
eventid_from	string	Retorna apenas problemas com IDs maiores ou iguais ao ID fornecido.
eventid_till	string	Retorna apenas problemas com IDs menores ou iguais ao ID fornecido.
time_from	timestamp	Retorna apenas problemas que foram criados após ou no horário fornecido.
time_till	timestamp	Retorna apenas problemas que foram criados antes ou no horário fornecido.
selectAcknowledges	query	Retorna uma propriedade `acknowledges` com as atualizações do problema. As atualizações do problema são classificadas em ordem cronológica inversa. O objeto de atualização do problema possui as seguintes propriedades: `acknowledgeid` - `(ID)` ID da atualização; `userid` - `(ID)` ID do usuário que atualizou o evento; `eventid` - `(ID)` ID do evento atualizado; `clock` - `(timestamp)` horário em que o evento foi atualizado; `message` - `(string)` texto da mensagem; `action` - `(integer)` tipo de ação de atualização (veja `event.acknowledge`); `old_severity` - `(integer)` gravidade do evento antes desta ação de atualização; `new_severity` - `(integer)` gravidade do evento após esta ação de atualização; `suppress_until` - `(timestamp)` horário até o qual o evento será suprimido; `taskid` - `(ID)` ID da tarefa se o evento atual estiver passando por uma alteração de classificação. Suporta `count`.
selectTags	query	Retorna uma propriedade `tags` com as tags do problema. Formato de saída: `[{"tag": "<tag>", "value": "<value>"}, ...]`.
selectSuppressionData	query	Retorna uma propriedade `suppression_data` com a lista de manutenções ativas e supressões manuais: `maintenanceid` - `(ID)` ID da manutenção; `userid` - `(ID)` ID do usuário que suprimiu o problema; `suppress_until` - `(integer)` horário até o qual o problema será suprimido.

Parâmetro	Tipo	Descrição
filter	object	Retorna apenas os resultados que correspondem exatamente ao filtro fornecido. Aceita um objeto, onde as chaves são nomes de propriedades e os valores são um único valor ou um array de valores para comparar. Não suporta propriedades do tipo de dado `text` data type.
sortfield	string/array	Classifica o resultado pelas propriedades fornecidas. Valores possíveis: `eventid`.
countOutput	boolean	Estes parâmetros são descritos no comentário de referência.
editable	boolean	
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valores de retorno

`(integer/array)` Retorna:

- uma matriz de objetos;
- a contagem de objetos recuperados, se o parâmetro `countOutput` foi usado.

Exemplos

Recuperando eventos de problemas de trigger

Recupere eventos recentes do trigger ”15112”.

Requisição:

```
{
    "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
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "eventid": "1245463",
            "source": "0",
            "object": "0",
            "objectid": "15112",
            "clock": "1472457242",
```

```
            "ns": "209442442",
            "r_eventid": "1245468",
            "r_clock": "1472457285",
            "r_ns": "125644870",
            "correlationid": "0",
            "userid": "1",
            "name": "Zabbix agent on localhost is unreachable for 5 minutes",
            "acknowledged": "1",
            "severity": "3",
            "cause_eventid": "0",
            "opdata": "",
            "acknowledges": [
                {
                    "acknowledgeid": "14443",
                    "userid": "1",
                    "eventid": "1245463",
                    "clock": "1472457281",
                    "message": "problem solved",
                    "action": "6",
                    "old_severity": "0",
                    "new_severity": "0",
                    "suppress_until": "1472511600",
                    "taskid": "0"
                }
            ],
            "suppression_data": [
                {
                    "maintenanceid": "15",
                    "suppress_until": "1472511600",
                    "userid": "0"
                }
            ],
            "suppressed": "1",
            "tags": [
                {
                    "tag": "test-tag",
                    "value": "test-value"
                }
            ]
        }
    ],
    "id": 1
}
```


Recuperando problemas reconhecidos por um usuário específico

Recuperando problemas reconhecidos pelo usuário com ID=10

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "problem.get",
    "params": {
        "output": "extend",
        "action": 2,
        "action_userids": [10],
        "selectAcknowledges": ["userid", "action"],
        "sortfield": ["eventid"],
        "sortorder": "DESC"
    },
    "id": 1
}
```

Resposta:

```
{
    "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",
            "opdata": "",
            "acknowledges": [
                {
                    "userid": "10",
                    "action": "2"
                }
            ],
            "suppressed": "0"
        }
    ],
    "id": 1
}
```


Veja também

- Alerta
- Item
- Host
- Regra LLD
- Trigger

Fonte

CEvent::get() in *ui/include/classes/api/services/CProblem.php*.

Protótipo de gráfico

Esta classe foi projetada para trabalhar com protótipos de gráficos.

Referências de objetos:

- Protótipo de gráfico

Métodos disponíveis:

- graphprototype.create - cria novos protótipos de gráficos
- graphprototype.delete - exclui protótipos de gráficos
- graphprototype.get - recupera protótipos de gráficos
- graphprototype.update - atualiza protótipos de gráficos

Objeto de protótipo de gráfico

Os seguintes objetos estão diretamente relacionados à API `graphprototype`.

Protótipo de gráfico

O objeto protótipo de gráfico possui as seguintes propriedades.

Propriedade	Tipo	Descrição
graphid	ID	ID do protótipo de gráfico. Comportamento da propriedade: - *somente leitura* - *obrigatório* para operações de atualização
height	integer	Altura do protótipo de gráfico em pixels. Comportamento da propriedade: - *obrigatório* para operações de criação
name	string	Nome do protótipo de gráfico. Comportamento da propriedade: - *obrigatório* para operações de criação
width	integer	Largura do protótipo de gráfico em pixels. Comportamento da propriedade: - *obrigatório* para operações de criação
graphtype	integer	Tipo de layout do protótipo de gráfico. Valores possíveis: 0 - *(padrão)* normal; 1 - empilhado; 2 - pizza; 3 - pizza explodida.
percent_left	float	Percentil esquerdo. Padrão: 0.
percent_right	float	Percentil direito. Padrão: 0.
show_3d	integer	Se deve exibir gráficos de pizza e pizza explodida descobertos em 3D. Valores possíveis: 0 - *(padrão)* exibir em 2D; 1 - exibir em 3D.
show_legend	integer	Se deve exibir a legenda no gráfico descoberto. Valores possíveis: 0 - ocultar; 1 - *(padrão)* exibir.
show_work_period	integer	Se deve exibir o período de trabalho no gráfico descoberto. Valores possíveis: 0 - ocultar; 1 - *(padrão)* exibir.
templateid	ID	ID do protótipo de gráfico do template pai. Comportamento da propriedade: - *somente leitura*
yaxismax	float	O valor máximo fixo para o eixo Y.
yaxismin	float	O valor mínimo fixo para o eixo Y.
ymax_itemid	ID	ID do item que é usado como valor máximo para o eixo Y. Se um usuário não tiver acesso ao item especificado, o gráfico será renderizado como se `ymax_type` estivesse definido como ”calculado”.
ymax_type	integer	Método de cálculo do valor máximo para o eixo Y. Valores possíveis: 0 - *(padrão)* calculado; 1 - fixo; 2 - item.

Propriedade	Tipo	Descrição
ymin_itemid	ID	ID do item que é usado como valor mínimo para o eixo Y. Se um usuário não tiver acesso ao item especificado, o gráfico será renderizado como se `ymin_type` estivesse definido como "calculado".
ymin_type	integer	Método de cálculo do valor mínimo para o eixo Y. Valores possíveis: 0 - *(padrão)* calculado; 1 - fixo; 2 - item.
discover	integer	Status de descoberta do protótipo de gráfico. Valores possíveis: 0 - *(padrão)* novos gráficos serão descobertos; 1 - novos gráficos não serão descobertos e os gráficos existentes serão marcados como perdidos.
uuid	string	Identificador único universal, usado para vincular protótipos de gráficos importados aos já existentes. Gerado automaticamente, se não fornecido. Comportamento da propriedade: - *suportado* se o protótipo de gráfico pertencer a um template

graphprototype.create

Descrição

`object graphprototype.create(object/array graphPrototypes)`

Este método permite criar novos protótipos de gráficos.

> **Note:**
> Este método está disponível apenas para os tipos de usuário *Admin* e *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object/array)` Protótipos de gráficos a serem criados.

Além das propriedades padrão do protótipo de gráfico, o método aceita os seguintes parâmetros.

Parâmetro	Tipo	Descrição
gitems	array	Itens de gráfico a serem criados para os protótipos de gráficos. Os itens de gráfico podem referenciar tanto items quanto protótipos de item, mas pelo menos um protótipo de item deve estar presente. Comportamento do parâmetro: - *obrigatório*

Valores de retorno

`(object)` Retorna um objeto contendo os IDs do gráfico criado protótipos sob a propriedade `graphids`. A ordem dos IDs retornados corresponde à ordem dos protótipos de grafos passados.

Exemplos

Criando um protótipo de gráfico

Crie um protótipo de gráfico com dois items.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "graphprototype.create",
    "params": {
        "name": "Disk space usage {#FSNAME}",
        "width": 900,
        "height": 200,
        "gitems": [
            {
                "itemid": "22828",
                "color": "00AA00"
            },
            {
                "itemid": "22829",
                "color": "3333FF"
            }
        ]
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "graphids": [
            "652"
        ]
    },
    "id": 1
}
```

Veja também

- Item de gráfico

Fonte

CGraphPrototype::create() in *ui/include/classes/api/services/CGraphPrototype.php*.

graphprototype.delete

Descrição

`object graphprototype.delete(array graphPrototypeIds)`

Este método permite excluir protótipos de gráficos.

> **Note:**
> Este método está disponível apenas para os tipos de usuário *Admin* e *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Consulte Funções de usuário para mais informações.

Parâmetros

`(array)` IDs dos protótipos de gráfico a serem excluídos.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs do gráfico excluído protótipos sob a propriedade `graphids`.

Exemplos

Excluindo múltiplos protótipos de gráficos

Exclua dois protótipos de gráficos.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "graphprototype.delete",
    "params": [
        "652",
        "653"
    ],
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "graphids": [
            "652",
            "653"
        ]
    },
    "id": 1
}
```

Fonte

CGraphPrototype::delete() in *ui/include/classes/api/services/CGraphPrototype.php*.

graphprototype.get

Descrição

`integer/array graphprototype.get(object parameters)`

O método permite recuperar protótipos de gráficos de acordo com os parâmetros fornecidos.

Note:
Este método está disponível para usuários de qualquer tipo. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object)` Parâmetros que definem a saída desejada.

O método suporta os seguintes parâmetros.

Parâmetro	Tipo	Descrição
discoveryids	ID/array	Retorna apenas protótipos de gráficos que pertencem às regras de descoberta fornecidas.
graphids	ID/array	Retorna apenas protótipos de gráficos com os IDs fornecidos.
groupids	ID/array	Retorna apenas protótipos de gráficos que pertencem a hosts ou templates nos grupos de hosts ou grupos de templates fornecidos.
hostids	ID/array	Retorna apenas protótipos de gráficos que pertencem aos hosts fornecidos.
inherited	boolean	Se definido como `true`, retorna apenas protótipos de gráficos herdados de um template.
itemids	ID/array	Retorna apenas protótipos de gráficos que contêm os protótipos de item fornecidos.
templated	boolean	Se definido como `true`, retorna apenas protótipos de gráficos que pertencem a templates.
templateids	ID/array	Retorna apenas protótipos de gráficos que pertencem aos templates fornecidos.
selectDiscoveryRule	query	Retorna uma propriedade `discoveryRule` com a regra LLD à qual o protótipo de gráfico pertence.

Parâmetro	Tipo	Descrição
selectGraphItems	query	Retorna uma propriedade `gitems` com os itens de gráfico usados no protótipo de gráfico.
selectHostGroups	query	Retorna uma propriedade `hostgroups` com os grupos de hosts aos quais o protótipo de gráfico pertence.
selectHosts	query	Retorna uma propriedade `hosts` com os hosts aos quais o protótipo de gráfico pertence.
selectItems	query	Retorna uma propriedade `items` com os items e protótipos de item usados no protótipo de gráfico.
selectTemplateGroups	query	Retorna uma propriedade `templategroups` com os grupos de templates aos quais o protótipo de gráfico pertence.
selectTemplates	query	Retorna uma propriedade `templates` com os templates aos quais o protótipo de gráfico pertence.
filter	object	Retorna apenas os resultados que correspondem exatamente ao filtro fornecido. Aceita um objeto, onde as chaves são nomes de propriedades e os valores são um único valor ou um array de valores para comparar. Não suporta propriedades do tipo de dado `text` data type. Suporta propriedades adicionais: `host` - nome técnico do host ao qual o protótipo de gráfico pertence; `hostid` - ID do host ao qual o protótipo de gráfico pertence.
sortfield	string/array	Ordena o resultado pelas propriedades fornecidas. Valores possíveis: `graphid`, `name`, `graphtype`, `discovered`.
countOutput	boolean	Estes parâmetros são descritos no comentário de referência.
editable	boolean	
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	
selectGroups (deprecated)	query	Este parâmetro está obsoleto, por favor use `selectHostGroups` ou `selectTemplateGroups`. Retorna uma propriedade `groups` com os grupos de hosts e grupos de templates aos quais o protótipo de gráfico pertence.

Valores de retorno

`(integer/array)` Retorna:

- uma matriz de objetos;
- a contagem de objetos recuperados, se o parâmetro `countOutput` tiver sido usado.

Exemplos

Recuperando protótipos de gráficos de uma regra LLD

Recupere todos os protótipos de gráficos de uma regra LLD.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "graphprototype.get",
    "params": {
        "output": "extend",
        "discoveryids": "27426"
    },
```

```
    "id": 1
}
```

Resposta:

```
{
    "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",
            "discover": "0"
        }
    ],
    "id": 1
}
```

Veja também

- Regra de descoberta
- Item do gráfico
- Item
- Host
- Grupo de hosts
- Template
- Grupo de templates

Fonte

CGraphPrototype::get() in *ui/include/classes/api/services/CGraphPrototype.php*.

graphprototype.update

Descrição

`object graphprototype.update(object/array graphPrototypes)`

Este método permite atualizar protótipos de gráficos existentes.

> **Note:**
> Este método está disponível apenas para os tipos de usuário *Admin* e *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object/array)` Propriedades do protótipo de gráfico a serem atualizadas.

A propriedade `graphid` deve ser definida para cada protótipo de gráfico, todas as outras propriedades são opcionais. Somente as propriedades passadas serão atualizadas, todas as outras permanecerão inalteradas.

Além das propriedades padrão do protótipo de gráfico, o método aceita os seguintes parâmetros.

Parâmetro	Tipo	Descrição
gitems	array	Itens do gráfico para substituir os itens de gráfico existentes. Se um item de gráfico tiver a propriedade `gitemid` definida, ele será atualizado; caso contrário, um novo item de gráfico será criado.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs do gráfico atualizado protótipos sob a propriedade `graphids`.

Exemplos

Alterando o tamanho de um protótipo de gráfico

Altere o tamanho de um protótipo de gráfico para 1100 por 400 pixels.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "graphprototype.update",
    "params": {
        "graphid": "439",
        "width": 1100,
        "height": 400
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "graphids": [
            "439"
        ]
    },
    "id": 1
}
```

Fonte

CGraphPrototype::update() em *ui/include/classes/api/services/CGraphPrototype.php*.

Protótipo de host

Esta classe foi projetada para trabalhar com protótipos de host.

Referências de objeto:

- Protótipo de host
- Vínculo de grupo
- Protótipo de grupo
- Tag de protótipo de host
- Interface personalizada
    - Detalhes da interface personalizada

Métodos disponíveis:

- hostprototype.create - criar novos protótipos de host
- hostprototype.delete - excluir protótipos de host
- hostprototype.get - recuperar protótipos de host
- hostprototype.update - atualizar protótipos de host

Objeto de protótipo de host

Os seguintes objetos estão diretamente relacionados à API `hostprototype`.

Protótipo de host

O objeto protótipo de host possui as seguintes propriedades.

Propriedade	Tipo	Descrição
hostid	ID	ID do protótipo de host. Comportamento da propriedade: - *somente leitura* - *obrigatório* para operações de atualização
host	string	Nome técnico do protótipo de host. Comportamento da propriedade: - *obrigatório* para operações de criação - *somente leitura* para objetos herdados
name	string	Nome visível do protótipo de host. Padrão: valor da propriedade `host`. Comportamento da propriedade: - *somente leitura* para objetos herdados
status	integer	Status do protótipo de host. Valores possíveis: 0 - *(padrão)* host monitorado; 1 - host não monitorado.
inventory_mode	integer	Modo de preenchimento do inventário do host. Valores possíveis: -1 - *(padrão)* desabilitado; 0 - manual; 1 - automático.
templateid	ID	ID do protótipo de host do template pai. Comportamento da propriedade: - *somente leitura*
discover	integer	Status de descoberta do protótipo de host. Valores possíveis: 0 - *(padrão)* novos hosts serão descobertos; 1 - novos hosts não serão descobertos e hosts existentes serão marcados como perdidos.
custom_interfaces	integer	Fonte das interfaces personalizadas para hosts criados pelo protótipo de host. Valores possíveis: 0 - *(padrão)* herda interfaces do host pai; 1 - usa interfaces personalizadas dos protótipos de host. Comportamento da propriedade: - *somente leitura* para objetos herdados
uuid	string	Identificador único universal, usado para vincular protótipos de host importados a já existentes. Gerado automaticamente, se não fornecido. Comportamento da propriedade: - *suportado* se o protótipo de host pertencer a um template

Associação de grupo

O objeto de associação de grupo vincula um protótipo de host a um grupo de hosts. Ele possui as seguintes propriedades.

Propriedade	Tipo	Descrição
groupid	ID	ID do grupo de hosts. Comportamento da propriedade: - *obrigatório*

Protótipo de grupo

O objeto de protótipo de grupo define um grupo que será criado para um host descoberto. Ele possui as seguintes propriedades.

Propriedade	Tipo	Descrição
group_prototypeid	ID	ID do protótipo de grupo. Comportamento da propriedade: - *somente leitura*
name	string	Nome do protótipo de grupo. Comportamento da propriedade: - *obrigatório* para operações de criação

Tag de protótipo de host

O objeto de tag de protótipo de host possui as seguintes propriedades.

Propriedade	Tipo	Descrição
tag	string	Nome da tag do protótipo de host. Comportamento da propriedade: - *obrigatório*
value	string	Valor da tag do protótipo de host.

Interface personalizada

Interfaces personalizadas são suportadas se `custom_interfaces` do objeto protótipo de host estiver definido como "usar interfaces personalizadas dos protótipos de host". O objeto de interface personalizada possui as seguintes propriedades.

Propriedade	Tipo	Descrição
type	integer	Tipo de interface. Valores possíveis: 1 - Agent; 2 - SNMP; 3 - IPMI; 4 - JMX. Comportamento da propriedade: - *obrigatório*
useip	integer	Se a conexão deve ser feita via IP. Valores possíveis: 0 - conectar usando o nome DNS do host; 1 - conectar usando o endereço IP do host. Comportamento da propriedade: - *obrigatório*

Propriedade	Tipo	Descrição
ip	string	Endereço IP usado pela interface. Pode conter macros. Comportamento da propriedade: - *obrigatório* se `useip` estiver definido como ”conectar usando o endereço IP do host”
dns	string	Nome DNS usado pela interface. Pode conter macros. Comportamento da propriedade: - *obrigatório* se `useip` estiver definido como ”conectar usando o nome DNS do host”
port	string	Número da porta usada pela interface. Pode conter macros de usuário e LLD. Comportamento da propriedade: - *obrigatório*
main	integer	Se a interface é usada como padrão no host. Apenas uma interface de cada tipo pode ser definida como padrão em um host. Valores possíveis: 0 - não é padrão; 1 - padrão. Comportamento da propriedade: - *obrigatório*
details	object	Objeto adicional para detalhes da interface personalizada. Comportamento da propriedade: - *obrigatório* se `type` estiver definido como ”SNMP”

Detalhes da interface personalizada

O objeto de detalhes possui as seguintes propriedades.

Propriedade	Tipo	Descrição
version	integer	Versão da interface SNMP. Valores possíveis: 1 - SNMPv1; 2 - SNMPv2c; 3 - SNMPv3. Comportamento da propriedade: - *obrigatório*
bulk	integer	Se deve usar requisições SNMP bulk. Valores possíveis: 0 - não usar requisições bulk; 1 - *(padrão)* - usar requisições bulk.
community	string	Comunidade SNMP. Comportamento da propriedade: - *obrigatório* se `version` estiver definido como ”SNMPv1” ou ”SNMPv2c”

Propriedade	Tipo	Descrição
max_repetitions	integer	Valor máximo de repetição para requisições SNMP bulk nativas (GetBulkRequest-PDUs). Usado apenas para itens `discovery[]` e `walk[]` em SNMPv2 e v3. Padrão: 10.
securityname	string	Nome de segurança SNMPv3. Comportamento da propriedade: - *suportado* se `version` estiver definido como ”SNMPv3”
securitylevel	integer	Nível de segurança SNMPv3. Valores possíveis: 0 - *(padrão)* - noAuthNoPriv; 1 - authNoPriv; 2 - authPriv. Comportamento da propriedade: - *suportado* se `version` estiver definido como ”SNMPv3”
authpassphrase	string	Senha de autenticação SNMPv3. Comportamento da propriedade: - *suportado* se `version` estiver definido como ”SNMPv3” e `securitylevel` estiver definido como ”authNoPriv” ou ”authPriv”
privpassphrase	string	Senha de privacidade SNMPv3. Comportamento da propriedade: - *suportado* se `version` estiver definido como ”SNMPv3” e `securitylevel` estiver definido como ”authPriv”
authprotocol	integer	Protocolo de autenticação SNMPv3. Valores possíveis: 0 - *(padrão)* - MD5; 1 - SHA1; 2 - SHA224; 3 - SHA256; 4 - SHA384; 5 - SHA512. Comportamento da propriedade: - *suportado* se `version` estiver definido como ”SNMPv3” e `securitylevel` estiver definido como ”authNoPriv” ou ”authPriv”
privprotocol	integer	Protocolo de privacidade SNMPv3. Usado apenas por interfaces SNMPv3. Valores possíveis: 0 - *(padrão)* - DES; 1 - AES128; 2 - AES192; 3 - AES256; 4 - AES192C; 5 - AES256C. Comportamento da propriedade: - *suportado* se `version` estiver definido como ”SNMPv3” e `securitylevel` estiver definido como ”authPriv”
contextname	string	Nome do contexto SNMPv3. Comportamento da propriedade: - *suportado* se `version` estiver definido como ”SNMPv3”

hostprototype.create

Descrição

`object hostprototype.create(object/array hostPrototypes)`

Este método permite criar novos protótipos de host.

> **Note:**
> Este método está disponível apenas para os tipos de usuário *Admin* e *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função de usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object/array)` Protótipos de host a serem criados.

Além das propriedades padrão do protótipo de host, o método aceita os seguintes parâmetros.

Parâmetro	Tipo	Descrição
groupLinks	array	Vínculos de grupo a serem criados para o protótipo de host. Comportamento do parâmetro: - *obrigatório*
ruleid	ID	ID da regra LLD à qual o protótipo de host pertence. Comportamento do parâmetro: - *obrigatório*
groupPrototypes	array	Protótipos de grupo a serem criados para o protótipo de host.
macros	object/array	Macros de usuário a serem criadas para o protótipo de host.
tags	object/array	Tags do protótipo de host.
interfaces	object/array	Interfaces personalizadas do protótipo de host.
templates	object/array	Templates a serem vinculados ao protótipo de host. Os templates devem ter apenas a propriedade `templateid` definida.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs do host criado protótipos sob a propriedade `hostids`. A ordem dos IDs retornados corresponde à ordem dos protótipos de host passados.

Exemplos

Criando um protótipo de host

Crie um protótipo de host "{#VM.NAME}" na regra LLD "23542" com um protótipo de grupo "{#HV.NAME}", par de tags "datacenter": "{#DATACENTER.NAME}" e interface SNMPv2 personalizada 127.0.0.1:161 com comunidade {$SNMP_COMMUNITY}. Vincule-o ao grupo de hosts "2".

Requisição:

```
{
    "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
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10103"
        ]
    },
    "id": 1
}
```

Veja também

- Vínculo de grupo
- Protótipo de grupo
- Tag de protótipo de host
- Interface personalizada
- Macro de usuário

Fonte

CHostPrototype::create() em *ui/include/classes/api/services/CHostPrototype.php*.

hostprototype.delete

Descrição

`object hostprototype.delete(array hostPrototypeIds)`

Este método permite excluir protótipos de host.

> **Note:**
> Este método está disponível apenas para os tipos de usuário *Admin* e *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função de usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(array)` IDs dos protótipos de host a serem excluídos.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs do host excluído protótipos sob a propriedade `hostids`.

Exemplos

Excluindo múltiplos protótipos de host

Exclua dois protótipos de host.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "hostprototype.delete",
    "params": [
        "10103",
        "10105"
    ],
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10103",
            "10105"
        ]
    },
    "id": 1
}
```

Fonte

CHostPrototype::delete() in *ui/include/classes/api/services/CHostPrototype.php*.

hostprototype.get

Descrição

`integer/array hostprototype.get(object parameters)`

O método permite recuperar protótipos de host de acordo com os parâmetros fornecidos.

> **Note:**
> Este método está disponível para usuários de qualquer tipo. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object)` Parâmetros que definem a saída desejada.

O método suporta os seguintes parâmetros.

Parâmetro	Tipo	Descrição
hostids	ID/array	Retorna apenas protótipos de host com os IDs fornecidos.
discoveryids	ID/array	Retorna apenas protótipos de host que pertencem às regras LLD fornecidas.
inherited	boolean	Se definido como `true`, retorna apenas itens herdados de um template.
selectDiscoveryRule	query	Retorna uma propriedade `discoveryRule` com a regra LLD à qual o protótipo de host pertence.
selectInterfaces	query	Retorna uma propriedade `interfaces` com as interfaces personalizadas do protótipo de host.

Parâmetro	Tipo	Descrição
selectGroupLinks	query	Retorna uma propriedade `groupLinks` com os links de grupo do protótipo de host.
selectGroupPrototypes	query	Retorna uma propriedade `groupPrototypes` com os protótipos de grupo do protótipo de host.
selectMacros	query	Retorna uma propriedade `macros` com as macros do protótipo de host.
selectParentHost	query	Retorna uma propriedade `parentHost` com o host ao qual o protótipo de host pertence.
selectTags	query	Retorna uma propriedade `tags` com as tags do protótipo de host.
selectTemplates	query	Retorna uma propriedade `templates` com os templates vinculados ao protótipo de host. Suporta `count`.
sortfield	string/array	Classifica o resultado pelas propriedades fornecidas. Valores possíveis: `hostid`, `host`, `name`, `status`, `discovered`.
countOutput	boolean	Esses parâmetros são descritos no comentário de referência.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valores de retorno

`(integer/array)` Retorna:

- uma matriz de objetos;
- a contagem de objetos recuperados, se o parâmetro `countOutput` tiver sido usado.

Exemplos

Recuperando protótipos de host de uma regra LLD

Recupere todos os protótipos de host, seus links de grupo, protótipos de grupo e tags de uma regra LLD.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "hostprototype.get",
    "params": {
        "output": "extend",
        "selectInterfaces": "extend",
        "selectGroupLinks": "extend",
        "selectGroupPrototypes": "extend",
        "selectTags": "extend",
        "discoveryids": "23554"
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "hostid": "10092",
            "host": "{#HV.UUID}",
```

```
            "name": "{#HV.UUID}",
            "status": "0",
            "templateid": "0",
            "discover": "0",
            "custom_interfaces": "1",
            "inventory_mode": "-1",
            "groupLinks": [
                {
                    "group_prototypeid": "4",
                    "hostid": "10092",
                    "groupid": "7",
                    "templateid": "0"
                }
            ],
            "groupPrototypes": [
                {
                    "group_prototypeid": "7",
                    "hostid": "10092",
                    "name": "{#CLUSTER.NAME}",
                    "templateid": "0"
                }
            ],
            "tags": [
                {
                    "tag": "datacenter",
                    "value": "{#DATACENTER.NAME}"
                },
                {
                    "tag": "instance-type",
                    "value": "{#INSTANCE_TYPE}"
                }
            ],
            "interfaces": [
                {
                    "main": "1",
                    "type": "2",
                    "useip": "1",
                    "ip": "127.0.0.1",
                    "dns": "",
                    "port": "161",
                    "details": {
                        "version": "2",
                        "bulk": "1",
                        "community": "{$SNMP_COMMUNITY}",
                        "max_repetitions": "10"
                    }
                }
            ]
        }
    ],
    "id": 1
}
```


Veja também

- Vínculo de grupo
- Protótipo de grupo
- Macro de usuário

Fonte

CHostPrototype::get() in *ui/include/classes/api/services/CHostPrototype.php*.

hostprototype.update

Descrição

`object hostprototype.update(object/array hostPrototypes)`

Este método permite atualizar protótipos de host existentes.

> **Note:**
> Este método está disponível apenas para os tipos de usuário *Admin* e *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Consulte Funções de usuário para mais informações.

Parâmetros

`(object/array)` Propriedades do protótipo de host a serem atualizadas.

A propriedade `hostid` deve ser definida para cada protótipo de host, todas as outras propriedades são opcionais. Somente as propriedades passadas serão atualizadas, todas as outras permanecerão inalteradas.

Além das propriedades padrão do protótipo de host, o método aceita os seguintes parâmetros.

Parâmetro	Tipo	Descrição
groupLinks	array	Vínculos de grupo para substituir os vínculos de grupo atuais no protótipo de host. Comportamento do parâmetro: - *somente leitura* para objetos herdados
groupPrototypes	array	Protótipos de grupo para substituir os protótipos de grupo existentes no protótipo de host. Todos os protótipos de grupo que não estiverem listados na solicitação serão removidos. Comportamento do parâmetro: - *somente leitura* para objetos herdados
macros	object/array	Macros de usuário para substituir as macros de usuário atuais. Todas as macros que não estiverem listadas na solicitação serão removidas.
tags	object/array	Tags do protótipo de host para substituir as tags atuais. Todas as tags que não estiverem listadas na solicitação serão removidas. Comportamento do parâmetro: - *somente leitura* para objetos herdados
interfaces	object/array	Interfaces personalizadas do protótipo de host para substituir as interfaces atuais. O objeto de interface personalizada deve conter todos os seus parâmetros. Todas as interfaces que não estiverem listadas na solicitação serão removidas. Comportamento do parâmetro: - *suportado* se `custom_interfaces` do objeto protótipo de host estiver definido como "usar interfaces personalizadas dos protótipos de host" - *somente leitura* para objetos herdados
templates	object/array	Templates para substituir os templates atualmente vinculados. Os templates devem ter apenas a propriedade `templateid` definida.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs do host atualizado protótipos sob a propriedade `hostids`.

Exemplos

Desabilitando um protótipo de host

Desabilite um protótipo de host, ou seja, defina seu status como ”1”.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "hostprototype.update",
    "params": {
        "hostid": "10092",
        "status": 1
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10092"
        ]
    },
    "id": 1
}
```

Atualizando tags de protótipos de host

Substitua as tags do protótipo de host por novas.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "hostprototype.update",
    "params": {
        "hostid": "10092",
        "tags": [
            {
                "tag": "datacenter",
                "value": "{#DATACENTER.NAME}"
            },
            {
                "tag": "instance-type",
                "value": "{#INSTANCE_TYPE}"
            }
        ]
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10092"
        ]
    },
    "id": 1
}
```

Atualizando interfaces personalizadas de protótipos de host

Substitua as interfaces herdadas por interfaces personalizadas de protótipos de host.

Requisição:

```
{
    "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
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10092"
        ]
    },
    "id": 1
}
```

Veja também

- Vínculo de grupo
- Protótipo de grupo
- Tag de protótipo de host
- Interface personalizada
- Macro de usuário

Fonte

CHostPrototype::update() em *ui/include/classes/api/services/CHostPrototype.php*.

Protótipo de item

Esta classe foi projetada para trabalhar com protótipos de item.

Referências de objeto:

- Protótipo de item
  - Cabeçalho HTTP
  - Campo de consulta HTTP
- Tag de protótipo de item
- Pré-processamento de protótipo de item

Métodos disponíveis:

- itemprototype.create - criar novos protótipos de item

- itemprototype.delete - excluir protótipos de item
- itemprototype.get - recuperar protótipos de item
- itemprototype.update - atualizar protótipos de item

Objeto de protótipo de item

Os seguintes objetos estão diretamente relacionados à API `itemprototype`.

Protótipo de item

O objeto protótipo de item possui as seguintes propriedades.

Propriedade	Tipo	Descrição
itemid	ID	ID do protótipo de item. Comportamento da propriedade: - *somente leitura* - *obrigatório* para operações de atualização
delay	string	Intervalo de atualização do protótipo de item. Aceita segundos ou unidade de tempo com sufixo (por exemplo, 30s, 1m, 2h, 1d) e, opcionalmente, um ou mais intervalos personalizados, todos separados por ponto e vírgula. Os intervalos personalizados podem ser uma mistura de intervalos flexíveis e de agendamento. Aceita macros de usuário e macros LLD. Se usado, o valor deve ser uma única macro. Múltiplas macros ou macros misturadas com texto não são suportadas. Intervalos flexíveis podem ser escritos como duas macros separadas por uma barra (por exemplo, `{$FLEX_INTERVAL}/{$FLEX_PERIOD}`). Exemplo: `1h;wd1-5h9-18;{$Macro1}/1-7,00:00-24:00;0/6-7,12:00-24:00;{$Macr` Comportamento da propriedade: - *obrigatório* se `type` for definido como ”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), ou se `type` for definido como ”Zabbix agent (active)” (7) e `key_` não contiver ”mqtt.get”
hostid	ID	ID do host ao qual o protótipo de item pertence. Comportamento da propriedade: - *constante* - *obrigatório* para operações de criação
interfaceid	ID	ID da interface do host do protótipo de item. Comportamento da propriedade: - *obrigatório* se o protótipo de item pertencer ao host e `type` for definido como ”Zabbix agent”, ”IPMI agent”, ”JMX agent”, ”SNMP trap” ou ”SNMP agent” - *suportado* se o protótipo de item pertencer ao host e `type` for definido como ”Simple check”, ”External check”, ”SSH agent”, ”TELNET agent” ou ”HTTP agent”
key_	string	Chave do protótipo de item. Comportamento da propriedade: - *obrigatório* para operações de criação - *somente leitura* para objetos herdados

Propriedade	Tipo	Descrição
name	string	Nome do protótipo de item. Suporta macros de usuário. Comportamento da propriedade: - *obrigatório* para operações de criação - *somente leitura* para objetos herdados
type	integer	Tipo do protótipo de item. Valores possíveis: 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. Comportamento da propriedade: - *obrigatório* para operações de criação - *somente leitura* para objetos herdados
url	string	String de URL. Suporta macros LLD, macros de usuário, {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {ITEM.ID}, {ITEM.KEY}. Comportamento da propriedade: - *obrigatório* se `type` for definido como ”HTTP agent” - *somente leitura* para objetos herdados
value_type	integer	Tipo de informação do protótipo de item. Valores possíveis: 0 - numérico float; 1 - caractere; 2 - log; 3 - numérico sem sinal; 4 - texto; 5 - binário. Comportamento da propriedade: - *obrigatório* para operações de criação - *somente leitura* para objetos herdados
allow_traps	integer	Permitir popular valor de forma semelhante ao item trapper. 0 - *(padrão)* Não permitir aceitar dados recebidos; 1 - Permitir aceitar dados recebidos. Comportamento da propriedade: - *suportado* se `type` for definido como ”HTTP agent”

Propriedade	Tipo	Descrição
authtype	integer	Método de autenticação. Valores possíveis se `type` for definido como ”SSH agent”: 0 - *(padrão)* senha; 1 - chave pública. Valores possíveis se `type` for definido como ”HTTP agent”: 0 - *(padrão)* nenhum; 1 - básico; 2 - NTLM; 3 - Kerberos; 4 - Digest. Comportamento da propriedade: - *suportado* se `type` for definido como ”SSH agent” ou ”HTTP agent” - *somente leitura* para objetos herdados (se `type` for definido como ”HTTP agent”)
description	string	Descrição do protótipo de item.
follow_redirects	integer	Seguir redirecionamentos de resposta ao coletar dados. Valores possíveis: 0 - Não seguir redirecionamentos; 1 - *(padrão)* Seguir redirecionamentos. Comportamento da propriedade: - *suportado* se `type` for definido como ”HTTP agent” - *somente leitura* para objetos herdados
headers	array	Array de cabeçalhos que serão enviados ao executar uma requisição HTTP. Comportamento da propriedade: - *suportado* se `type` for definido como ”HTTP agent” - *somente leitura* para objetos herdados
history	string	Unidade de tempo de quanto tempo os dados de histórico devem ser armazenados. Também aceita macro de usuário e macro LLD. Padrão: 31d.
http_proxy	string	String de conexão do proxy HTTP(S). Comportamento da propriedade: - *suportado* se `type` for definido como ”HTTP agent” - *somente leitura* para objetos herdados
ipmi_sensor	string	Sensor IPMI. Comportamento da propriedade: - *obrigatório* se `type` for definido como ”IPMI agent” e `key_` não for definido como ”ipmi.get” - *suportado* se `type` for definido como ”IPMI agent” - *somente leitura* para objetos herdados
jmx_endpoint	string	String de conexão personalizada do JMX agent. Padrão: service:jmx:rmi:///jndi/rmi://{HOST.CONN}:{HOST.PORT}/jmxrmi Comportamento da propriedade: - *suportado* se `type` for definido como ”JMX agent”
logtimefmt	string	Formato da hora nas entradas de log. Comportamento da propriedade: - *suportado* se `value_type` for definido como ”log” - *somente leitura* para objetos herdados

Propriedade	Tipo	Descrição
master_itemid	ID	ID do item mestre. É permitida recursão de até 3 itens dependentes e protótipos de item e contagem máxima de itens dependentes e protótipos de item igual a 29999. Comportamento da propriedade: - *obrigatório* se `type` for definido como ”Dependent item” - *somente leitura* para objetos herdados
output_format	integer	Se a resposta deve ser convertida para JSON. Valores possíveis: 0 - *(padrão)* Armazenar bruto; 1 - Converter para JSON. Comportamento da propriedade: - *suportado* se `type` for definido como ”HTTP agent” - *somente leitura* para objetos herdados
params	string	Parâmetros adicionais dependendo do tipo do protótipo de item: - script executado para protótipos de item SSH agent e TELNET agent; - consulta SQL para protótipos de item database monitor; - fórmula para protótipos de item calculated; - o script para protótipos de item script e browser. Comportamento da propriedade: - *obrigatório* se `type` for definido como ”Database monitor”, ”SSH agent”, ”TELNET agent”, ”Calculated”, ”Script” ou ”Browser” - *somente leitura* para objetos herdados (se `type` for definido como ”Script” ou ”Browser”)
parameters	object/array	Parâmetros adicionais se `type` for definido como ”Script” ou ”Browser”. Array de objetos com propriedades `name` e `value`, onde `name` deve ser único. Comportamento da propriedade: - *suportado* se `type` for definido como ”Script” ou ”Browser” - *somente leitura* para objetos herdados
password	string	Senha para autenticação. Comportamento da propriedade: - *obrigatório* se `type` for definido como ”JMX agent” e `username` for definido - *suportado* se `type` for definido como ”Simple check”, ”SSH agent”, ”TELNET agent”, ”Database monitor” ou ”HTTP agent” - *somente leitura* para objetos herdados (se `type` for definido como ”HTTP agent”)
post_type	integer	Tipo de corpo de dados postados armazenados na propriedade `posts`. Valores possíveis: 0 - *(padrão)* Dados brutos. 2 - Dados JSON. 3 - Dados XML. Comportamento da propriedade: - *suportado* se `type` for definido como ”HTTP agent” - *somente leitura* para objetos herdados
posts	string	Dados do corpo da requisição HTTP(S). Comportamento da propriedade: - *obrigatório* se `type` for definido como ”HTTP agent” e `post_type` for definido como ”JSON data” ou ”XML data” - *suportado* se `type` for definido como ”HTTP agent” e `post_type` for definido como ”Raw data” - *somente leitura* para objetos herdados

Propriedade	Tipo	Descrição
privatekey	string	Nome do arquivo de chave privada. Comportamento da propriedade: - *obrigatório* se `type` for definido como ”SSH agent” e `authtype` for definido como ”public key”
publickey	string	Nome do arquivo de chave pública. Comportamento da propriedade: - *obrigatório* se `type` for definido como ”SSH agent” e `authtype` for definido como ”public key”
query_fields	array	Array de campos de consulta que serão enviados ao executar uma requisição HTTP. Comportamento da propriedade: - *suportado* se `type` for definido como ”HTTP agent” - *somente leitura* para objetos herdados
request_method	integer	Tipo de método de requisição. Valores possíveis: 0 - *(padrão)* GET; 1 - POST; 2 - PUT; 3 - HEAD. Comportamento da propriedade: - *suportado* se `type` for definido como ”HTTP agent” - *somente leitura* para objetos herdados
retrieve_mode	integer	Qual parte da resposta deve ser armazenada. Valores possíveis se `request_method` for definido como ”GET”, ”POST” ou ”PUT”: 0 - *(padrão)* Corpo; 1 - Cabeçalhos; 2 - Corpo e cabeçalhos serão armazenados. Valores possíveis se `request_method` for definido como ”HEAD”: 1 - Cabeçalhos. Comportamento da propriedade: - *suportado* se `type` for definido como ”HTTP agent” - *somente leitura* para objetos herdados
snmp_oid	string	SNMP OID. Comportamento da propriedade: - *obrigatório* se `type` for definido como ”SNMP agent” - *somente leitura* para objetos herdados
ssl_cert_file	string	Caminho do arquivo de chave pública SSL. Comportamento da propriedade: - *suportado* se `type` for definido como ”HTTP agent” - *somente leitura* para objetos herdados
ssl_key_file	string	Caminho do arquivo de chave privada SSL. Comportamento da propriedade: - *suportado* se `type` for definido como ”HTTP agent” - *somente leitura* para objetos herdados
ssl_key_password	string	Senha para o arquivo de chave SSL. Comportamento da propriedade: - *suportado* se `type` for definido como ”HTTP agent” - *somente leitura* para objetos herdados

Propriedade	Tipo	Descrição
status	integer	Status do protótipo de item. Valores possíveis: 0 - *(padrão)* protótipo de item habilitado; 1 - protótipo de item desabilitado; 3 - protótipo de item não suportado.
status_codes	string	Intervalos de códigos de status HTTP necessários, separados por vírgula. Também suporta macros de usuário ou macros LLD como parte da lista separada por vírgula. Exemplo: 200,200-{$M},{$M},200-400 Comportamento da propriedade: - *suportado* se `type` for definido como "HTTP agent" - *somente leitura* para objetos herdados
templateid	ID	ID do protótipo de item do template pai. Comportamento da propriedade: - *somente leitura*
timeout	string	Tempo limite da requisição de coleta de dados do item. Aceita segundos ou unidade de tempo com sufixo (por exemplo, 30s, 1m). Também aceita macros de usuário e macros LLD. Faixa de valores possíveis: 1-600s. Padrão: "" - usar configurações do proxy/global. Comportamento da propriedade: - *suportado* se `type` for definido como "Zabbix agent" (0), "Simple check" (3) e `key_` não começar com "`vmware.`" e "`icmpping`", "Zabbix agent (active)" (7), "External check" (10), "Database monitor" (11), "SSH agent" (13), "TELNET agent" (14), "HTTP agent" (19), "SNMP agent" (20) e `snmp_oid` começar com "`walk[`" ou "`get[`", "Script" (21), "Browser" (22) - *somente leitura* para objetos herdados
trapper_hosts	string	Hosts permitidos. Comportamento da propriedade: - *suportado* se `type` for definido como "Zabbix trapper", ou se `type` for definido como "HTTP agent" e `allow_traps` for definido como "Allow to accept incoming data"
trends	string	Unidade de tempo de quanto tempo os dados de tendências devem ser armazenados. Também aceita macro de usuário e macro LLD. Padrão: 365d. Comportamento da propriedade: - *suportado* se `value_type` for definido como "numeric float" ou "numeric unsigned"
units	string	Unidades de valor. Comportamento da propriedade: - *suportado* se `value_type` for definido como "numeric float" ou "numeric unsigned" - *somente leitura* para objetos herdados

Propriedade	Tipo	Descrição
username	string	Nome de usuário para autenticação. Comportamento da propriedade: - *obrigatório* se `type` for definido como ”SSH agent” ou ”TELNET agent”, ou se `type` for definido como ”JMX agent” e `password` for definido - *suportado* se `type` for definido como ”Simple check”, ”Database monitor” ou ”HTTP agent” - *somente leitura* para objetos herdados (se `type` for definido como ”HTTP agent”)
uuid	string	Identificador único universal, usado para vincular protótipos de item importados aos já existentes. Gerado automaticamente, se não fornecido. Comportamento da propriedade: - *suportado* se o protótipo de item pertencer a um template
valuemapid	ID	ID do mapa de valores associado. Comportamento da propriedade: - *suportado* se `value_type` for definido como ”numeric float”, ”character” ou ”numeric unsigned” - *somente leitura* para objetos herdados
verify_host	integer	Se deve validar que o nome do host para a conexão corresponde ao do certificado do host. Valores possíveis: 0 - *(padrão)* Não validar; 1 - Validar. Comportamento da propriedade: - *suportado* se `type` for definido como ”HTTP agent” - *somente leitura* para objetos herdados
verify_peer	integer	Se deve validar que o certificado do host é autêntico. Valores possíveis: 0 - *(padrão)* Não validar; 1 - Validar. Comportamento da propriedade: - *suportado* se `type` for definido como ”HTTP agent” - *somente leitura* para objetos herdados
discover	integer	Status de descoberta do protótipo de item. Valores possíveis: 0 - *(padrão)* novos itens serão descobertos; 1 - novos itens não serão descobertos e itens existentes serão marcados como perdidos.

Cabeçalho HTTP

O objeto header possui as seguintes propriedades:

Propriedade	Tipo	Descrição
name	string	Nome do cabeçalho HTTP. Comportamento da propriedade: - *obrigatório*
value	string	Valor do cabeçalho. Comportamento da propriedade: - *obrigatório*

Campo de consulta HTTP

O objeto de campo de consulta define um nome e valor que é usado para especificar um parâmetro de URL. Ele possui as seguintes propriedades:

Propriedade	Tipo	Descrição
name	string	Nome do parâmetro. Comportamento da propriedade: - *obrigatório*
value	string	Valor do parâmetro. Comportamento da propriedade: - *obrigatório*

Tag de protótipo de item

O objeto de tag de protótipo de item possui as seguintes propriedades.

Propriedade	Tipo	Descrição
tag	string	Nome da tag do protótipo de item. Comportamento da propriedade: - *obrigatório*
value	string	Valor da tag do protótipo de item.

Pré-processamento de protótipo de item

O objeto de pré-processamento de protótipo de item possui as seguintes propriedades.

Propriedade	Tipo	Descrição
type	integer	O tipo de opção de pré-processamento. Valores possíveis: 1 - Multiplicador personalizado; 2 - Remover à direita; 3 - Remover à esquerda; 4 - Remover; 5 - Expressão regular; 6 - Booleano para decimal; 7 - Octal para decimal; 8 - Hexadecimal para decimal; 9 - Alteração simples; 10 - Alteração por segundo; 11 - XML XPath; 12 - JSONPath; 13 - No intervalo; 14 - Corresponde à expressão regular; 15 - Não corresponde à expressão regular; 16 - Verificar erro em JSON; 17 - Verificar erro em XML; 18 - Verificar erro usando expressão regular; 19 - Descartar inalterado; 20 - Descartar inalterado com heartbeat; 21 - JavaScript; 22 - Padrão Prometheus; 23 - Prometheus para JSON; 24 - CSV para JSON; 25 - Substituir; 26 - Verificar não suportado; 27 - XML para JSON; 28 - Valor SNMP walk; 29 - SNMP walk para JSON; 30 - Valor SNMP get. Comportamento da propriedade: - *obrigatório*
params	string	Parâmetros adicionais usados pela opção de pré-processamento. Múltiplos parâmetros são separados pelo caractere de nova linha (\n). Se `type` estiver definido como ”Verificar não suportado”, os parâmetros seguem a sintaxe `<scope>[\n<pattern>]`, onde *pattern* é uma expressão regular e *scope* é um dos: -1 - corresponder a qualquer erro; 0 - verificar se a mensagem de erro corresponde ao *pattern*; 1 - verificar se a mensagem de erro não corresponde ao *pattern*. Comportamento da propriedade: - *obrigatório* se `type` estiver definido como ”Multiplicador personalizado” (1), ”Remover à direita” (2), ”Remover à esquerda” (3), ”Remover” (4), ”Expressão regular” (5), ”XML XPath” (11), ”JSONPath” (12), ”No intervalo” (13), ”Corresponde à expressão regular” (14), ”Não corresponde à expressão regular” (15), ”Verificar erro em JSON” (16), ”Verificar erro em XML” (17), ”Verificar erro usando expressão regular” (18), ”Descartar inalterado com heartbeat” (20), ”JavaScript” (21), ”Padrão Prometheus” (22), ”Prometheus para JSON” (23), ”CSV para JSON” (24), ”Substituir” (25), ”Verificar não suportado” (26), ”Valor SNMP walk” (28), ”SNMP walk para JSON” (29) ou ”Valor SNMP get” (30)

Propriedade	Tipo	Descrição
error_handler	integer	Tipo de ação usada em caso de falha na etapa de pré-processamento. Valores possíveis: 0 - Mensagem de erro definida pelo Zabbix server; 1 - Descartar valor; 2 - Definir valor personalizado; 3 - Definir mensagem de erro personalizada. Valores possíveis se `type` estiver definido como ”Verificar não suportado”: 1 - Descartar valor; 2 - Definir valor personalizado; 3 - Definir mensagem de erro personalizada. Comportamento da propriedade: - *obrigatório* se `type` estiver definido como ”Multiplicador personalizado” (1), ”Expressão regular” (5), ”Booleano para decimal” (6), ”Octal para decimal” (7), ”Hexadecimal para decimal” (8), ”Alteração simples” (9), ”Alteração por segundo” (10), ”XML XPath” (11), ”JSONPath” (12), ”No intervalo” (13), ”Corresponde à expressão regular” (14), ”Não corresponde à expressão regular” (15), ”Verificar erro em JSON” (16), ”Verificar erro em XML” (17), ”Verificar erro usando expressão regular” (18), ”Padrão Prometheus” (22), ”Prometheus para JSON” (23), ”CSV para JSON” (24), ”Verificar não suportado” (26), ”XML para JSON” (27), ”Valor SNMP walk” (28), ”SNMP walk para JSON” (29) ou ”Valor SNMP get” (30)
error_handler_params	string	Parâmetros do manipulador de erro. Comportamento da propriedade: - *obrigatório* se `error_handler` estiver definido como ”Definir valor personalizado” ou ”Definir mensagem de erro personalizada”

Os seguintes parâmetros e manipuladores de erro são suportados para cada tipo de pré-processamento.

Tipo de pré-processamento	Nome	Parâmetro 1	Parâmetro 2	Parâmetro 3	Manipuladores de erro suportados
1	Multiplicador personalizado	número^1, 6^			0, 1, 2, 3
2	Remover à direita	lista de caracteres[2]			
3	Remover à esquerda	lista de caracteres[2]			
4	Remover	lista de caracteres[2]			
5	Expressão regular	padrão[3]	saída[2]		0, 1, 2, 3
6	Booleano para decimal				0, 1, 2, 3
7	Octal para decimal				0, 1, 2, 3

Tipo de pré-processamento	Nome	Parâmetro 1	Parâmetro 2	Parâmetro 3	Manipuladores de erro suportados
8	Hexadecimal para decimal				0, 1, 2, 3
9	Alteração simples				0, 1, 2, 3
10	Alteração por segundo				0, 1, 2, 3
11	XML XPath	caminho[4]			0, 1, 2, 3
12	JSONPath	caminho[4]			0, 1, 2, 3
13	No intervalo	min^1, 6^	max^1, 6^		0, 1, 2, 3
14	Corresponde à expressão regular	padrão[3]			0, 1, 2, 3
15	Não corresponde à expressão regular	padrão[3]			0, 1, 2, 3
16	Verificar erro em JSON	caminho[4]			0, 1, 2, 3
17	Verificar erro em XML	caminho[4]			0, 1, 2, 3
18	Verificar erro usando expressão regular	padrão[3]	saída[2]		0, 1, 2, 3
19	Descartar inalterado				
20	Descartar inalterado com heartbeat	segundos^5, 6^			
21	JavaScript	script[2]			
22	Padrão Prometheus	padrão^6, 7^	`value`, `label`, `function`	saída^8, 9^	0, 1, 2, 3

Tipo de pré-processamento	Nome	Parâmetro 1	Parâmetro 2	Parâmetro 3	Manipuladores de erro suportados
23	Prometheus para JSON	padrão^6, 7^			0, 1, 2, 3
24	CSV para JSON	caractere[2]	caractere[2]	0,1	0, 1, 2, 3
25	Substituir	string de busca[2]	substituição[2]		
26	Verificar não suportado	escopo[1]	padrão^3, 6^		1, 2, 3
27	XML para JSON				0, 1, 2, 3
28	Valor SNMP walk	OID[2]	Formato: 0 - Inalterado 1 - UTF-8 de Hex-STRING 2 - MAC de Hex-STRING 3 - Inteiro de BITS		0, 1, 2, 3
29	SNMP walk para JSON[10]	Nome do campo[2]	Prefixo OID[2]	Formato: 0 - Inalterado 1 - UTF-8 de Hex-STRING 2 - MAC de Hex-STRING 3 - Inteiro de BITS	0, 1, 2, 3
30	Valor SNMP get	Formato: 1 - UTF-8 de Hex-STRING 2 - MAC de Hex-STRING 3 - Inteiro de BITS			0, 1, 2, 3

[1] número inteiro ou de ponto flutuante
 [2] string
 [3] expressão regular
 [4] JSONPath ou XML XPath
 [5] número inteiro positivo (com suporte a sufixos de tempo, por exemplo, 30s, 1m, 2h, 1d)
 [6] macro de usuário, macro LLD
 [7] padrão Prometheus seguindo a sintaxe: `<nome da métrica>{<nome do rótulo>="<valor do rótulo>", ...} == <valor>`. Cada componente do padrão Prometheus (métrica, nome do rótulo, valor do rótulo e valor da métrica) pode ser macro de usuário ou macro LLD.
 [8] saída Prometheus seguindo a sintaxe: `<nome do rótulo>` (pode ser uma macro de usuário ou uma macro LLD) se `label` for selecionado como o segundo parâmetro.
 [9] Uma das funções de agregação: `sum`, `min`, `max`, `avg`, `count` se `function` for selecionado como o segundo parâmetro.
 [10] Suporta vários registros "Nome do campo,Prefixo OID,Formato" delimitados por um caractere de nova linha.

itemprototype.create

Descrição

`object itemprototype.create(object/array itemPrototypes)`

Este método permite criar novos protótipos de item.

Note:
Este método está disponível apenas para os tipos de usuário *Admin* e *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object/array)` Prototipo de item a ser criado.

Além das propriedades padrão do protótipo de item, o método aceita os seguintes parâmetros.

Parâmetro	Tipo	Descrição
ruleid	ID	ID da regra LLD à qual o item pertence. Comportamento do parâmetro: - *obrigatório*
preprocessing	array	Opções de pré-processamento do protótipo de item.
tags	array	Tags do protótipo de item.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs do item criado protótipos sob a propriedade `itemids`. A ordem dos IDs retornados corresponde à ordem dos protótipos de itens passados.

Exemplos

Criando um protótipo de item

Crie um protótipo de item para monitorar o espaço livre em disco em um sistema de arquivos descoberto. Os itens descobertos devem ser itens numéricos do agent Zabbix atualizados a cada 30 segundos.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.create",
    "params": {
        "name": "Free disk space on {#FSNAME}",
        "key_": "vfs.fs.size[{#FSNAME},free]",
        "hostid": "10197",
        "ruleid": "27665",
        "type": 0,
        "value_type": 3,
        "interfaceid": "112",
        "delay": "30s"
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "27666"
        ]
    },
    "id": 1
}
```

Criando um protótipo de item com pré-processamento

Crie um item usando alteração por segundo e um multiplicador personalizado como um segundo passo.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.create",
    "params": {
        "name": "Incoming network traffic on {#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
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "44211"
        ]
    },
    "id": 1
}
```

Criando um protótipo de item dependente

Crie um protótipo de item dependente para o protótipo de item mestre com ID 44211. Somente dependências no mesmo host (template/regra de descoberta) são permitidas, portanto, o item mestre e o item dependente devem ter o mesmo hostid e ruleid.

Requisição:

```
{
    "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
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "44212"
```

```
        ]
    },
    "id": 1
}
```

Criar protótipo de item HTTP agent

Crie um protótipo de item com URL usando macro de usuário, campos de consulta e cabeçalhos personalizados.

Requisição:

```
{
    "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
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "28305"
        ]
    },
    "id": 1
}
```

Criar protótipo de item de script

Crie uma coleta de dados simples usando um protótipo de item de script.

Requisição:

```
{
    "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
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "23865"
        ]
    },
    "id": 1
}
```

Fonte

CItemPrototype::create() in *ui/include/classes/api/services/CItemPrototype.php*.

itemprototype.delete

Descrição

`object itemprototype.delete(array itemPrototypeIds)`

Este método permite excluir protótipos de item.

> **Note:**
> Este método está disponível apenas para os tipos de usuário *Admin* e *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(array)` IDs dos protótipos de itens a serem excluídos.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs do item excluído protótipos sob a propriedade `prototypeids`.

Exemplos

Excluindo múltiplos protótipos de item

Exclua dois protótipos de item.
 Protótipos de item dependentes são removidos automaticamente se o item mestre ou o protótipo de item for excluído.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.delete",
    "params": [
        "27352",
        "27356"
```

```
    ],
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "prototypeids": [
            "27352",
            "27356"
        ]
    },
    "id": 1
}
```

Fonte

CItemPrototype::delete() in *ui/include/classes/api/services/CItemPrototype.php*.

itemprototype.get

Descrição

`integer/array itemprototype.get(object parameters)`

O método permite recuperar protótipos de item de acordo com os parâmetros fornecidos.

> **Note:**
> Este método está disponível para usuários de qualquer tipo. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object)` Parâmetros que definem a saída desejada.

O método suporta os seguintes parâmetros.

Parâmetro	Tipo	Descrição
discoveryids	ID/array	Retornar apenas os protótipos de item que pertencem às regras LLD fornecidas.
graphids	ID/array	Retornar apenas os protótipos de item que são usados nos protótipos de gráfico fornecidos.
hostids	ID/array	Retornar apenas os protótipos de item que pertencem aos hosts fornecidos.
inherited	boolean	Se definido como `true`, retorna apenas os protótipos de item herdados de um template.
itemids	ID/array	Retornar apenas os protótipos de item com os IDs fornecidos.
monitored	boolean	Se definido como `true`, retorna apenas os protótipos de item habilitados que pertencem a hosts monitorados.
templated	boolean	Se definido como `true`, retorna apenas os protótipos de item que pertencem a templates.
templateids	ID/array	Retornar apenas os protótipos de item que pertencem aos templates fornecidos.
triggerids	ID/array	Retornar apenas os protótipos de item que são usados nos protótipos de trigger fornecidos.
selectDiscoveryRule	query	Retornar uma propriedade `discoveryRule` com a regra de descoberta de baixo nível à qual o protótipo de item pertence.
selectGraphs	query	Retornar uma propriedade `graphs` com os protótipos de gráfico em que o protótipo de item é usado. Suporta `count`.
selectHosts	query	Retornar uma propriedade `hosts` com um array de hosts aos quais o protótipo de item pertence.

Parâmetro	Tipo	Descrição
selectTags	query	Retornar as tags do protótipo de item na propriedade `tags`.
selectTriggers	query	Retornar uma propriedade `triggers` com os protótipos de trigger em que o protótipo de item é usado. Suporta `count`.
selectPreprocessing	query	Retornar uma propriedade `preprocessing` com as opções de pré-processamento do protótipo de item.
selectValueMap	query	Retornar uma propriedade `valuemap` com o mapa de valores do protótipo de item.
filter	object	Retornar apenas os resultados que correspondem exatamente ao filtro fornecido. Aceita um objeto, onde as chaves são nomes de propriedades e os valores são um único valor ou um array de valores para comparar. Não suporta propriedades do tipo de dado `text` data type. Suporta propriedades adicionais: `host` - nome técnico do host ao qual o protótipo de item pertence.
limitSelects	integer	Limita o número de registros retornados por subseleções. Aplica-se às seguintes subseleções: `selectGraphs` - os resultados serão classificados por `name`; `selectTriggers` - os resultados serão classificados por `description`.
sortfield	string/array	Classifica o resultado pelas propriedades fornecidas. Valores possíveis: `itemid`, `name`, `key_`, `delay`, `type`, `status`, `history`, `trends`, `discovered`.
countOutput	boolean	Esses parâmetros são descritos no comentário de referência.
editable	boolean	
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valores de retorno

`(integer/array)` Retorna:

- uma matriz de objetos;
- a contagem de objetos recuperados, se o parâmetro `countOutput` tiver sido usado.

Exemplos

Recuperando protótipos de item de uma regra LLD

Recupere todos os protótipos de item para um ID de regra LLD específico.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.get",
    "params": {
        "output": "extend",
        "discoveryids": "27426"
    },
```

```
    "id": 1
}
```

Resposta:


```
{
    "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",
            "logtimefmt": "",
            "templateid": "0",
            "valuemapid": "0",
            "params": "",
            "ipmi_sensor": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "interfaceid": "0",
            "description": "",
            "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": "",
            "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": "",
            "logtimefmt": "",
            "templateid": "0",
            "valuemapid": "0",
            "params": "",
            "ipmi_sensor": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "interfaceid": "0",
            "description": "The processor load is calculated as system CPU load divided by number of CPU c
            "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": "",
            "parameters": []
        }
    ],
    "id": 1
}
```


Encontrando um item dependente

Encontre um item dependente para um ID de item específico.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "item.get",
    "params": {
        "output": "extend",
        "filter": {
            "type": 18,
```

```
            "master_itemid": "25545"
        },
        "limit": "1"
    },
    "id": 1
}
```

Resposta:


```
{
    "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": "",
            "logtimefmt": "",
            "templateid": "0",
            "valuemapid": "0",
            "params": "",
            "ipmi_sensor": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "interfaceid": "0",
            "description": "",
            "evaltype": "0",
            "master_itemid": "25545",
            "jmx_endpoint": "",
            "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": "",
            "parameters": []
        }
```

```
    ],
    "id": 1
}
```

Encontrar protótipo de item HTTP agent

Encontre o protótipo de item HTTP agent com o método de requisição HEAD para um ID de host específico.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.get",
    "params": {
        "hostids": "10254",
        "filter": {
            "type": 19,
            "request_method": 3
        }
    },
    "id": 1
}
```

Resposta:

```
{
    "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": "",
            "logtimefmt": "",
            "templateid": "28255",
            "valuemapid": "0",
            "params": "",
            "ipmi_sensor": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "flags": "2",
            "interfaceid": "2",
            "description": "",
            "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": "",
            "parameters": []
        }
    ],
    "id": 1
}
```


Veja também

- Host
- Protótipo de gráfico
- Protótipo de trigger

Fonte

CItemPrototype::get() in *ui/include/classes/api/services/CItemPrototype.php*.

itemprototype.update

Descrição

`object itemprototype.update(object/array itemPrototypes)`

Este método permite atualizar protótipos de item existentes.

> **Note:**
> Este método está disponível apenas para os tipos de usuário *Admin* e *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object/array)` Propriedades do protótipo de item a serem atualizadas.

A propriedade `itemid` deve ser definida para cada protótipo de item, todas as outras propriedades são opcionais. Somente as propriedades passadas serão atualizadas, todas as outras permanecerão inalteradas.

Além das propriedades padrão do protótipo de item, o método aceita os seguintes parâmetros.

Parâmetro	Tipo	Descrição
preprocessing	array	Opções de pré-processamento do protótipo de item para substituir as opções de pré-processamento atuais. Comportamento do parâmetro: - *somente leitura* para objetos herdados
tags	array	Tags do protótipo de item.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos protótipos de itens atualizados na propriedade `itemids`.

Exemplos

Alterando a interface de um protótipo de item

Altere a interface do host que será usada pelos itens descobertos.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.update",
    "params": {
        "itemid": "27428",
        "interfaceid": "132"
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "27428"
        ]
    },
    "id": 1
}
```

Atualizar protótipo de item dependente

Atualiza o protótipo de item dependente com um novo ID de protótipo de item mestre. Apenas dependências no mesmo host (template/regra de descoberta) são permitidas, portanto, o item mestre e o item dependente devem ter o mesmo hostid e ruleid.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.update",
    "params": {
        "master_itemid": "25570",
        "itemid": "189030"
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "189030"
        ]
    },
    "id": 1
}
```

Atualizar protótipo de item de agente HTTP

Altere os campos de consulta e remova todos os cabeçalhos personalizados.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.update",
    "params": {
        "itemid":"28305",
        "query_fields": [
            {
                "name": "random",
                "value": "qwertyuiopasdfghjklzxcvbnm"
            }
```

```
        ],
        "headers": []
    }
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "28305"
        ]
    },
    "id": 1
}
```

Atualizando opções de pré-processamento de item

Atualize um protótipo de item com a regra de pré-processamento de item ”Multiplicador personalizado”.

Request:

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.update",
    "params": {
        "itemid": "44211",
        "preprocessing": [
            {
                "type": 1,
                "params": "4",
                "error_handler": 2,
                "error_handler_params": "5"
            }
        ]
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "44211"
        ]
    },
    "id": 1
}
```

Atualizando um protótipo de item de script

Atualize um protótipo de item de script com um script diferente e remova parâmetros desnecessários que foram usados pelo script anterior.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.update",
    "params": {
        "itemid": "23865",
        "parameters": [],
        "script": "Zabbix.log(3, 'Log test');\nreturn 1;"
    },
```

```
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "23865"
        ]
    },
    "id": 1
}
```

Fonte

CItemPrototype::update() in *ui/include/classes/api/services/CItemPrototype.php*.

Protótipo de trigger

Esta classe foi projetada para trabalhar com protótipos de trigger.

Referências de objetos:

- Protótipo de trigger
- Tag de protótipo de trigger

Métodos disponíveis:

- triggerprototype.create - criar novos protótipos de trigger
- triggerprototype.delete - excluir protótipos de trigger
- triggerprototype.get - recuperar protótipos de trigger
- triggerprototype.update - atualizar protótipos de trigger

Objeto de protótipo de trigger

Os seguintes objetos estão diretamente relacionados à API `triggerprototype`.

Protótipo de trigger

O objeto de protótipo de trigger possui as seguintes propriedades.

Propriedade	Tipo	Descrição
triggerid	ID	ID do protótipo de trigger. Comportamento da propriedade: - *somente leitura* - *obrigatório* para operações de atualização
description	string	Nome do protótipo de trigger. Comportamento da propriedade: - *obrigatório* para operações de criação
expression	string	Expressão do trigger. Deve conter pelo menos um protótipo de item. Protótipos de item com `value_type` definido como 5 (binário) não podem ser usados em expressões de trigger. Comportamento da propriedade: - *obrigatório* para operações de criação
event_name	string	Nome do evento gerado pelo trigger.
opdata	string	Dados operacionais.
comments	string	Comentários adicionais para o protótipo de trigger.

Propriedade	Tipo	Descrição
priority	integer	Severidade do protótipo de trigger. Valores possíveis: 0 - *(padrão)* não classificado; 1 - informação; 2 - aviso; 3 - médio; 4 - alto; 5 - desastre.
status	integer	Se o protótipo de trigger está habilitado ou desabilitado. Valores possíveis: 0 - *(padrão)* habilitado; 1 - desabilitado.
templateid	ID	ID do protótipo de trigger do template pai. Comportamento da propriedade: - *somente leitura*
type	integer	Se o protótipo de trigger pode gerar múltiplos eventos de problema. Valores possíveis: 0 - *(padrão)* não gerar múltiplos eventos; 1 - gerar múltiplos eventos.
url	string	URL associada ao protótipo de trigger.
url_name	string	Rótulo para a URL associada ao protótipo de trigger.
recovery_mode	integer	Modo de geração do evento OK. Valores possíveis: 0 - *(padrão)* Expressão; 1 - Expressão de recuperação; 2 - Nenhum.
recovery_expression	string	Expressão de recuperação do trigger. Deve conter pelo menos um protótipo de item. Protótipos de item com `value_type` definido como 5 (binário) não podem ser usados em expressões de recuperação de trigger.
correlation_mode	integer	Evento OK fecha. Valores possíveis: 0 - *(padrão)* Todos os problemas; 1 - Todos os problemas se os valores das tags coincidirem.
correlation_tag	string	Tag para correspondência.
manual_close	integer	Permitir fechamento manual. Valores possíveis: 0 - *(padrão)* Não; 1 - Sim.
discover	integer	Status de descoberta do protótipo de trigger. Valores possíveis: 0 - *(padrão)* novos triggers serão descobertos; 1 - novos triggers não serão descobertos e triggers existentes serão marcados como perdidos.
uuid	string	Identificador único universal, usado para vincular protótipos de trigger importados aos já existentes. Gerado automaticamente, se não fornecido. Comportamento da propriedade: - *suportado* se o protótipo de trigger pertencer a um template

Tag de protótipo de trigger

O objeto de tag de protótipo de trigger possui as seguintes propriedades.

Propriedade	Tipo	Descrição
tag	string	Nome da tag do protótipo de trigger. Comportamento da propriedade: - *obrigatório*
value	string	Valor da tag do protótipo de trigger.

triggerprototype.create

Descrição

`object triggerprototype.create(object/array triggerPrototypes)`

Este método permite criar novos protótipos de trigger.

Note:
Este método está disponível apenas para os tipos de usuário *Admin* e *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função de usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object/array)` Protótipos de trigger a serem criados.

Além das propriedades padrão do protótipo de trigger, o método aceita os seguintes parâmetros.

Parâmetro	Tipo	Descrição
dependencies	array	Triggers e protótipos de trigger dos quais o protótipo de trigger depende. Os triggers devem ter apenas a propriedade `triggerid` definida.
tags	array	Tags do protótipo de trigger.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos protótipos de trigger criados na propriedade `triggerids`. A ordem dos IDs retornados corresponde à ordem dos protótipos de triggers passados.

Exemplos

Criando um protótipo de trigger

Crie um protótipo de trigger para detectar quando um sistema de arquivos tem menos de 20% de espaço livre em disco.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "triggerprototype.create",
    "params": {
        "description": "Free disk space is less than 20% on volume {#FSNAME}",
        "expression": "last(/Zabbix server/vfs.fs.size[{#FSNAME},pfree])<20",
        "tags": [
            {
                "tag": "volume",
                "value": "{#FSNAME}"
            },
            {
                "tag": "type",
                "value": "{#FSTYPE}"
            }
        ]
    },
```

```
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "triggerids": [
            "17372"
        ]
    },
    "id": 1
}
```

Fonte

CTriggerPrototype::create() in *ui/include/classes/api/services/CTriggerPrototype.php*.

triggerprototype.delete

Descrição

`object triggerprototype.delete(array triggerPrototypeIds)`

Este método permite excluir protótipos de trigger.

> **Note:**
> Este método está disponível apenas para os tipos de usuário *Admin* e *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função de usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(array)` IDs dos protótipos de trigger a serem excluídos.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos protótipos de trigger excluídos na propriedade `triggerids`.

Exemplos

Excluindo múltiplos protótipos de trigger

Exclua dois protótipos de trigger.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "triggerprototype.delete",
    "params": [
        "12002",
        "12003"
    ],
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "triggerids": [
            "12002",
            "12003"
        ]
    },
    "id": 1
}
```

Fonte

CTriggerPrototype::delete() in *ui/include/classes/api/services/CTriggerPrototype.php*.

triggerprototype.get

Descrição

`integer/array triggerprototype.get(object parameters)`

O método permite recuperar protótipos de trigger de acordo com os parâmetros fornecidos.

> **Note:**
> Este método está disponível para usuários de qualquer tipo. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Consulte Funções de usuário para mais informações.

Parâmetros

`(object)` Parâmetros que definem a saída desejada.

O método suporta os seguintes parâmetros.

Parâmetro	Tipo	Descrição
active	flag	Retorna apenas protótipos de trigger habilitados que pertencem a hosts monitorados.
discoveryids	ID/array	Retorna apenas protótipos de trigger que pertencem às regras LLD fornecidas.
functions	string/array	Retorna apenas triggers que usam as funções fornecidas. Consulte a página Funções suportadas para obter uma lista de funções suportadas.
group	string	Retorna apenas protótipos de trigger que pertencem a hosts ou templates dos grupos de hosts ou grupos de templates com o nome fornecido.
groupids	ID/array	Retorna apenas protótipos de trigger que pertencem a hosts ou templates dos grupos de hosts ou grupos de templates fornecidos.
host	string	Retorna apenas protótipos de trigger que pertencem a hosts com o nome fornecido.
hostids	ID/array	Retorna apenas protótipos de trigger que pertencem aos hosts fornecidos.
inherited	boolean	Se definido como `true`, retorna apenas protótipos de trigger herdados de um template.
maintenance	boolean	Se definido como `true`, retorna apenas protótipos de trigger habilitados que pertencem a hosts em manutenção.
min_severity	integer	Retorna apenas protótipos de trigger com severidade maior ou igual à severidade fornecida.
monitored	flag	Retorna apenas protótipos de trigger habilitados que pertencem a hosts monitorados e contêm apenas items habilitados.
templated	boolean	Se definido como `true`, retorna apenas protótipos de trigger que pertencem a templates.
templateids	ID/array	Retorna apenas protótipos de trigger que pertencem aos templates fornecidos.
triggerids	ID/array	Retorna apenas protótipos de trigger com os IDs fornecidos.
expandExpression	flag	Expande funções e macros na expressão do trigger.
selectDependencies	query	Retorna protótipos de trigger e triggers dos quais o protótipo de trigger depende na propriedade `dependencies`.
selectDiscoveryRule	query	Retorna a regra LLD à qual o protótipo de trigger pertence na propriedade `discoveryRule`.

Parâmetro	Tipo	Descrição
selectFunctions	query	Retorna funções usadas no protótipo de trigger na propriedade `functions`. Os objetos de função representam as funções usadas na expressão do trigger e possuem as seguintes propriedades: `functionid` - `(ID)` ID da função; `itemid` - `(ID)` ID do item usado na função; `function` - `(string)` nome da função; `parameter` - `(string)` parâmetro passado para a função. O parâmetro de consulta é substituído pelo símbolo $ na string retornada.
selectHostGroups	query	Retorna os grupos de hosts aos quais o protótipo de trigger pertence na propriedade `hostgroups`.
selectHosts	query	Retorna os hosts aos quais o protótipo de trigger pertence na propriedade `hosts`.
selectItems	query	Retorna items e protótipos de item usados no protótipo de trigger na propriedade `items`.
selectTags	query	Retorna as tags do protótipo de trigger na propriedade `tags`.
selectTemplateGroups	query	Retorna os grupos de templates aos quais o protótipo de trigger pertence na propriedade `templategroups`.
filter	object	Retorna apenas os resultados que correspondem exatamente ao filtro fornecido. Aceita um objeto, onde as chaves são nomes de propriedades e os valores são um único valor ou um array de valores a serem correspondidos. Não suporta propriedades do tipo de dado `text` data type. Suporta propriedades adicionais: `host` - nome técnico do host ao qual o protótipo de trigger pertence; `hostid` - ID do host ao qual o protótipo de trigger pertence.
limitSelects	integer	Limita o número de registros retornados por subconsultas. Aplica-se às seguintes subconsultas: `selectHosts` - os resultados serão classificados por `host`.
sortfield	string/array	Classifica o resultado pelas propriedades fornecidas. Valores possíveis: `triggerid`, `description`, `status`, `priority`, `discovered`.
countOutput	boolean	Estes parâmetros são descritos no comentário de referência.
editable	boolean	
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	
selectGroups (obsoleto)	query	Este parâmetro está obsoleto, use `selectHostGroups` ou `selectTemplateGroups` em vez disso. Retorna os grupos de hosts e grupos de templates aos quais o protótipo de trigger pertence na propriedade `groups`.

Valores de retorno

`(integer/array)` Retorna:

- um array de objetos;
- a contagem de objetos recuperados, se o parâmetro `countOutput` tiver sido usado.

Exemplos

Recuperar protótipos de trigger de uma regra LLD

Recupere todos os protótipos de trigger e suas funções de uma regra LLD.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "triggerprototype.get",
    "params": {
        "output": "extend",
        "selectFunctions": "extend",
        "discoveryids": "22450"
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "triggerid": "13272",
            "expression": "{12598}<20",
            "description": "Free inodes is less than 20% on volume {#FSNAME}",
            "url": "",
            "status": "0",
            "value": "0",
            "priority": "2",
            "lastchange": "0",
            "comments": "",
            "error": "",
            "templateid": "0",
            "type": "0",
            "state": "0",
            "flags": "2",
            "recovery_mode": "0",
            "recovery_expression": "",
            "correlation_mode": "0",
            "correlation_tag": "",
            "manual_close": "0",
            "opdata": "",
            "discover": "0",
            "event_name": "",
            "uuid": "6ce467d05e8745409a177799bed34bb3",
            "url_name": "",
            "functions": [
                {
                    "functionid": "12598",
                    "itemid": "22454",
                    "parameter": "$",
                    "function": "last"
                }
            ]
        },
        {
            "triggerid": "13266",
            "expression": "{13500}<20",
            "description": "Free disk space is less than 20% on volume {#FSNAME}",
            "url": "",
            "status": "0",
            "value": "0",
            "priority": "2",
```

```
            "lastchange": "0",
            "comments": "",
            "error": "",
            "templateid": "0",
            "type": "0",
            "state": "0",
            "flags": "2",
            "recovery_mode": "0",
            "recovery_expression": "",
            "correlation_mode": "0",
            "correlation_tag": "",
            "manual_close": "0",
            "opdata": "",
            "discover": "0",
            "event_name": "",
            "uuid": "74a1fc62bfe24b7eabe4e244c70dc384",
            "url_name": "",
            "functions": [
                {
                    "functionid": "13500",
                    "itemid": "22686",
                    "parameter": "$",
                    "function": "last"
                }
            ]
        }
    ],
    "id": 1
}
```

Recuperando um protótipo de trigger específico com tags

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "triggerprototype.get",
    "params": {
        "output": [
            "triggerid",
            "description"
        ],
        "selectTags": "extend",
        "triggerids": [
            "17373"
        ]
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "triggerid": "17373",
            "description": "Free disk space is less than 20% on volume {#FSNAME}",
            "tags": [
                {
                    "tag": "volume",
                    "value": "{#FSNAME}"
                },
                {
                    "tag": "type",
```

```
                    "value": "{#FSTYPE}"
                }
            ]
        }
    ],
    "id": 1
}
```

Veja também

- Regra de descoberta
- Item
- Host
- Grupo de hosts
- Grupo de templates

Fonte

CTriggerPrototype::get() em *ui/include/classes/api/services/CTriggerPrototype.php*.

triggerprototype.update

Descrição

`object triggerprototype.update(object/array triggerPrototypes)`

Este método permite atualizar protótipos de trigger existentes.

> **Note:**
> Este método está disponível apenas para os tipos de usuário *Admin* e *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object/array)` Propriedades do trigger prototype a serem atualizadas.

A propriedade `triggerid` deve ser definida para cada trigger prototype, todas as outras propriedades são opcionais. Somente as propriedades passadas serão atualizadas, todas as outras permanecerão inalteradas.

Além das propriedades padrão do trigger prototype, o método aceita os seguintes parâmetros.

Parâmetro	Tipo	Descrição
dependencies	array	Triggers e trigger prototypes dos quais o trigger prototype depende. Os triggers devem ter apenas a propriedade `triggerid` definida.
tags	array	Tags do trigger prototype.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos protótipos de triggers atualizados na propriedade `triggerids`.

Exemplos

Habilitando um protótipo de trigger

Habilite um protótipo de trigger, ou seja, defina seu status como "0".

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "triggerprototype.update",
    "params": {
        "triggerid": "13938",
        "status": 0
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "triggerids": [
            "13938"
        ]
    },
    "id": 1
}
```

Substituindo tags de protótipos de trigger

Substitua as tags para um protótipo de trigger.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "triggerprototype.update",
    "params": {
        "triggerid": "17373",
        "tags": [
            {
                "tag": "volume",
                "value": "{#FSNAME}"
            },
            {
                "tag": "type",
                "value": "{#FSTYPE}"
            }
        ]
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "triggerids": [
            "17373"
        ]
    },
    "id": 1
}
```

Fonte

CTriggerPrototype::update() in *ui/include/classes/api/services/CTriggerPrototype.php*.

Proxy

Esta classe foi projetada para trabalhar com proxies.

Referências de objetos:

- Proxy

Métodos disponíveis:

- proxy.create - criar novos proxies
- proxy.delete - excluir proxies
- proxy.get - recuperar proxies
- proxy.update - atualizar proxies

Objeto proxy

Os seguintes objetos estão diretamente relacionados à API de `proxy`.

Proxy

O objeto proxy possui as seguintes propriedades.

Propriedade	Tipo	Descrição
proxyid	ID	ID do proxy. Comportamento da propriedade: - *somente leitura* - *obrigatório* para operações de atualização
name	string	Nome do proxy. Comportamento da propriedade: - *obrigatório* para operações de criação
proxy_groupid	ID	ID do grupo de proxy. 0 - o proxy não está atribuído a nenhum grupo de proxy.
local_address	string	Endereço para agents ativos. Endereço IP ou nome DNS para conectar. Comportamento da propriedade: - *obrigatório* se `proxy_groupid` não for 0
local_port	string	Número da porta local do proxy para conectar. Macros de usuário são suportadas. Padrão: 10051. Comportamento da propriedade: - *suportado* se `proxy_groupid` não for 0
operating_mode	integer	Tipo de proxy. Valores possíveis: 0 - proxy ativo; 1 - proxy passivo. Comportamento da propriedade: - *obrigatório* para operações de criação
description	text	Descrição do proxy.
lastaccess	timestamp	Hora em que o proxy se conectou pela última vez ao server. Comportamento da propriedade: - *somente leitura*
address	string	Endereço IP ou nome DNS para conectar. Macros de usuário são suportadas. Comportamento da propriedade: - *obrigatório* se o modo de operação do Zabbix proxy for passivo
port	string	Número da porta para conectar. Macros de usuário são suportadas. Padrão: 10051. Comportamento da propriedade: - *suportado* se o modo de operação do Zabbix proxy for passivo
allowed_addresses	string	Endereços IP ou nomes DNS separados por vírgula do Zabbix proxy ativo.

Propriedade	Tipo	Descrição
tls_connect	integer	Conexões para o host. Valores possíveis: 1 - *(padrão)* Sem criptografia; 2 - PSK; 4 - certificado.
tls_accept	integer	Conexões do host. Valores possíveis de bitmap: 1 - *(padrão)* Sem criptografia; 2 - PSK; 4 - certificado. Este é um campo de máscara de bits; qualquer soma dos valores possíveis de bitmap é aceitável (por exemplo, 6 para PSK e certificado).
tls_issuer	string	Emissor do certificado.
tls_subject	string	Assunto do certificado.
tls_psk_identity	string	Identidade PSK; deve ser pareada com apenas um PSK (entre autoregistration, hosts e proxies). Não inclua informações sensíveis na identidade PSK, pois ela é enviada sem criptografia pela rede para informar ao receptor qual PSK usar. Comportamento da propriedade: - *somente gravação* - *obrigatório* se `tls_connect` estiver definido como ”PSK” ou `tls_accept` contiver o bit ”PSK”
tls_psk	string	Chave pré-compartilhada (PSK); deve ter pelo menos 32 dígitos hexadecimais. Comportamento da propriedade: - *somente gravação* - *obrigatório* se `tls_connect` estiver definido como ”PSK” ou `tls_accept` contiver o bit ”PSK”
custom_timeouts	integer	Se deve substituir os timeouts globais de item no nível do proxy. Valores possíveis: 0 - *(padrão)* usar configurações globais; 1 - substituir timeouts.
timeout_zabbix_agent	string	Não gastar mais do que `timeout_*` segundos no processamento. Aceita segundos ou unidade de tempo com sufixo (por exemplo, 30s, 1m). Também aceita macros de usuário. Faixa de valores possíveis: 1-600s. Padrão: ””. Comportamento da propriedade: - *obrigatório* se `custom_timeouts` estiver definido como 1.
timeout_simple_check		
timeout_snmp_agent		
timeout_external_check		
timeout_db_monitor		
timeout_http_agent		
timeout_ssh_agent		
timeout_telnet_agent		
timeout_script		
timeout_browser		

Propriedade	Tipo	Descrição
version	integer	Versão do proxy. Número de versão do Zabbix em três partes, onde dois dígitos decimais são usados para cada parte, por exemplo, 60401 para a versão 6.4.1, 70002 para a versão 7.0.2, etc. 0 - Versão do proxy desconhecida. Comportamento da propriedade: - *somente leitura*
compatibility	integer	Versão do proxy em comparação com a versão do Zabbix server. Valores possíveis: 0 - Indefinido; 1 - Versão atual (proxy e server têm a mesma versão principal); 2 - Versão desatualizada (a versão do proxy é mais antiga que a versão do server, mas é parcialmente suportada); 3 - Versão não suportada (a versão do proxy é mais antiga que a versão anterior LTS do server ou a versão principal do server é mais antiga que a versão principal do proxy). Comportamento da propriedade: - *somente leitura*
state	integer	Estado do proxy. Valores possíveis: 0 - Desconhecido; 1 - Offline; 2 - Online. Comportamento da propriedade: - *somente leitura*

proxy.create

Descrição

`object proxy.create(object/array proxies)`

Este método permite criar novos proxies.

> **Note:**
> Este método está disponível apenas para o tipo de usuário *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object/array)` Proxies a serem criados.

Além das propriedades padrão do proxy, o método aceita os seguintes parâmetros.

Parâmetro	Tipo	Descrição
hosts	array	Hosts a serem monitorados pelo proxy. Se um host já estiver sendo monitorado por um proxy diferente, ele será reatribuído ao proxy atual. Os hosts devem ter apenas a propriedade `hostid` definida.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos proxies criados na propriedade `proxyids`. A ordem dos IDs retornados corresponde à ordem dos proxies passados.

Exemplos

Criar um proxy ativo

Crie um proxy de ação ”Proxy ativo” e atribua um host para ser monitorado por ele.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "proxy.create",
    "params": {
        "name": "Active proxy",
        "operating_mode": "0",
        "hosts": [
            {
                "hostid": "10279"
            }
        ]
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "proxyids": [
            "10280"
        ]
    },
    "id": 1
}
```

Criar um proxy passivo

Crie um proxy passivo ”Passive proxy” e atribua dois hosts para serem monitorados por ele.

Requisição:

```
{
    "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
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "proxyids": [
            "10284"
```

```
        ]
    },
    "id": 1
}
```

Criando um proxy e adicionando-o a um grupo de proxies

Crie um proxy ativo "Active proxy" e adicione-o ao grupo de proxies com ID "1".

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "proxy.create",
    "params": {
        "name": "Active proxy",
        "proxy_groupid": "1",
        "operating_mode": "0"
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "proxyids": [
            "5"
        ]
    },
    "id": 1
}
```

Veja também

- Host
- Grupo de proxy

Fonte

CProxy::create() in *ui/include/classes/api/services/CProxy.php*.

proxy.delete

Descrição

`object proxy.delete(array proxies)`

Este método permite excluir proxies.

> **Note:**
> Este método está disponível apenas para o tipo de usuário *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Consulte Funções de usuário para obter mais informações.

Parâmetros

`(array)` IDs de proxies a serem excluídos.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos proxies excluídos na propriedade `proxyids`.

Exemplos

Excluir múltiplos proxies

Exclua dois proxies.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "proxy.delete",
    "params": [
        "10286",
        "10285"
    ],
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "proxyids": [
            "10286",
            "10285"
        ]
    },
    "id": 1
}
```

Fonte

CProxy::delete() in *ui/include/classes/api/services/CProxy.php*.

proxy.get

Descrição

`integer/array proxy.get(object parameters)`

O método permite recuperar proxies de acordo com os parâmetros fornecidos.

Note:
Este método está disponível para usuários de qualquer tipo. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object)` Parâmetros que definem a saída desejada.

O método suporta os seguintes parâmetros.

Parâmetro	Tipo	Descrição
proxyids	ID/array	Retorna apenas proxies com os IDs fornecidos.
proxy_groupids	ID/array	Retorna apenas proxies que pertencem aos grupos de proxy fornecidos.
selectAssignedHosts	query	Retorna uma propriedade `assignedHosts` com os hosts atribuídos ao proxy. Suporta `count`.
selectHosts	query	Retorna uma propriedade `hosts` com os hosts monitorados pelo proxy. Suporta `count`.
selectProxyGroup	query	Retorna uma propriedade `proxyGroup` com o objeto do grupo de proxy.
sortfield	string/array	Ordena o resultado pelas propriedades fornecidas. Valores possíveis: `proxyid`, `name`, `operating_mode`.
countOutput	boolean	Esses parâmetros são descritos no comentário de referência.
editable	boolean	
excludeSearch	boolean	
filter	object	

Parâmetro	Tipo	Descrição
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valores de retorno

`(integer/array)` Retorna:

- um array de objetos;
- a contagem de objetos recuperados, se o parâmetro `countOutput` tiver sido usado.

Exemplos

Recuperar todos os proxies

Recupere todos os proxies configurados e suas interfaces.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "proxy.get",
    "params": {
        "output": "extend"
    },
    "id": 1
}
```

Resposta:

```
{
    "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
}
```


Veja também

- Host
- Grupo de proxy

Fonte

CProxy::get() in *ui/include/classes/api/services/CProxy.php*.

proxy.update

Descrição

`object proxy.update(object/array proxies)`

Este método permite atualizar proxies existentes.

> **Note:**
> Este método está disponível apenas para o tipo de usuário *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object/array)` Propriedades do proxy a serem atualizadas.

A propriedade `proxyid` deve ser definida para cada proxy, todas as outras propriedades são opcionais. Somente as propriedades passadas serão atualizadas, todas as outras permanecerão inalteradas.

Além das propriedades padrão do proxy, o método aceita os seguintes parâmetros.

Parâmetro	Tipo	Descrição
hosts	array	Hosts a serem monitorados pelo proxy. Se um host já for monitorado por um proxy diferente, ele será reatribuído ao proxy atual. Os hosts devem ter apenas a propriedade `hostid` definida.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos proxies atualizados na propriedade `proxyids`.

Exemplos

Alterar hosts monitorados por um proxy

Atualize o proxy para monitorar os dois hosts fornecidos.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "proxy.update",
    "params": {
        "proxyid": "10293",
        "hosts": [
            {
                "hostid": "10294"
            },
            {
                "hostid": "10295"
            }
        ]
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "proxyids": [
            "10293"
        ]
    },
    "id": 1
}
```

Alterar o status do proxy

Altere o proxy para um proxy ativo e renomeie-o para ”Active proxy”.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "proxy.update",
    "params": {
        "proxyid": "10293",
        "name": "Active proxy",
        "operating_mode": "0"
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "proxyids": [
            "10293"
        ]
    },
    "id": 1
}
```

Adicionar um proxy a um grupo de proxy

Atualize o proxy com ID ”5” e adicione-o ao grupo de proxy com ID ”1”.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "proxy.create",
    "params": {
        "proxyid": "5",
        "proxy_groupid": "1",
        "local_address": "127.0.0.1"
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "proxyids": [
            "5"
        ]
    },
    "id": 1
}
```

Veja também

- Host
- Grupo de proxy

Fonte

CProxy::update() in *ui/include/classes/api/services/CProxy.php*.

Regra de descoberta

Esta classe foi projetada para trabalhar com regras de descoberta de rede.

> **Note:**
> Esta API foi projetada para trabalhar com regras de descoberta de rede. Para regras de descoberta de baixo nível, consulte API de regra LLD.

Referências de objetos:

- Regra de descoberta

Métodos disponíveis:

- drule.create - criar novas regras de descoberta de rede
- drule.delete - excluir regras de descoberta de rede
- drule.get - recuperar regras de descoberta de rede
- drule.update - atualizar regras de descoberta de rede

Objeto de regra de descoberta

Os seguintes objetos estão diretamente relacionados à API `drule`.

Regra de descoberta

O objeto de regra de descoberta define uma regra de descoberta de rede. Ele possui as seguintes propriedades.

Propriedade	Tipo	Descrição
druleid	ID	ID da regra de descoberta. Comportamento da propriedade: - *somente leitura* - *obrigatório* para operações de atualização
iprange	string	Um ou vários intervalos de IP a serem verificados, separados por vírgulas. Consulte a seção de configuração de descoberta de rede para obter mais informações sobre os formatos suportados de intervalos de IP. Comportamento da propriedade: - *obrigatório* para operações de criação
name	string	Nome da regra de descoberta. Comportamento da propriedade: - *obrigatório* para operações de criação
delay	string	Intervalo de execução da regra de descoberta. Aceita segundos ou unidade de tempo com sufixo (por exemplo, 30s, 1m, 2h, 1d), ou uma macro de usuário. Padrão: 1h.
proxyid	ID	ID do proxy usado para descoberta.
status	integer	Se a regra de descoberta está habilitada. Valores possíveis: 0 - *(padrão)* habilitada; 1 - desabilitada.
concurrency_max	integer	Número máximo de verificações simultâneas por regra de descoberta. Valores possíveis: 0 - *(padrão)* número ilimitado de verificações; 1 - uma verificação; 2-999 - número personalizado de verificações.
error	string	Texto do erro caso tenha havido algum problema ao executar a regra de descoberta. Comportamento da propriedade: - *somente leitura*

drule.create

Descrição

`object drule.create(object/array discoveryRules)`

Este método permite criar novas regras de descoberta.

> **Note:**
> Este método está disponível apenas para os tipos de usuário *Admin* e *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Consulte Funções de usuário para mais informações.

Parâmetros

(object/array) Regras de descoberta a serem criadas.

Além das propriedades padrão da regra de descoberta, o método aceita os seguintes parâmetros.

Parâmetro	Tipo	Descrição
dchecks	array	Verificações de descoberta a serem criadas para a regra de descoberta. Comportamento do parâmetro: - *obrigatório*

Valores de retorno

(object) Retorna um objeto contendo os IDs das regras de descoberta sob a propriedade druleids. A ordem dos IDs retornados correspondem à ordem das regras de descoberta passadas.

Exemplos

Criar uma regra de descoberta

Crie uma regra de descoberta para encontrar máquinas executando o agent Zabbix na rede local. A regra deve usar uma única verificação do agent Zabbix na porta 10050.

Requisição:

```
{
    "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
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "druleids": [
            "6"
        ]
    },
    "id": 1
}
```

Veja também

- Discovery check

Fonte

CDRule::create() in *ui/include/classes/api/services/CDRule.php*.

drule.delete

Descrição

```
object drule.delete(array discoveryRuleIds)
```

Este método permite excluir regras de descoberta.

> **Note:**
> Este método está disponível apenas para os tipos de usuário *Admin* e *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(array)` IDs das regras de descoberta a serem excluídas.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs da descoberta excluída regras sob a propriedade `druleids`.

Exemplos

Excluir várias regras de descoberta

Exclua duas regras de descoberta.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "drule.delete",
    "params": [
        "4",
        "6"
    ],
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "druleids": [
            "4",
            "6"
        ]
    },
    "id": 1
}
```

Fonte

CDRule::delete() in *ui/include/classes/api/services/CDRule.php*.

drule.get

Descrição

`integer/array drule.get(object parameters)`

O método permite recuperar regras de descoberta de acordo com os parâmetros fornecidos.

> **Note:**
> Este método está disponível para usuários de qualquer tipo. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object)` Parâmetros que definem a saída desejada.

O método suporta os seguintes parâmetros.

Parâmetro	Tipo	Descrição
dhostids	ID/array	Retorna apenas as regras de descoberta que criaram os hosts descobertos fornecidos.
druleids	ID/array	Retorna apenas as regras de descoberta com os IDs fornecidos.
dserviceids	ID/array	Retorna apenas as regras de descoberta que criaram os serviços descobertos fornecidos.
selectDChecks	query	Retorna uma propriedade `dchecks` com as verificações de descoberta usadas pela regra de descoberta. Suporta `count`.
selectDHosts	query	Retorna uma propriedade `dhosts` com os hosts descobertos criados pela regra de descoberta. Suporta `count`.
limitSelects	integer	Limita o número de registros retornados por subconsultas. Aplica-se às seguintes subconsultas: `selectDChecks` - os resultados serão ordenados por `dcheckid`; `selectDHosts` - os resultados serão ordenados por `dhostsid`.
sortfield	string/array	Ordena o resultado pelas propriedades fornecidas. Valores possíveis: `druleid`, `name`.
countOutput	boolean	Estes parâmetros são descritos no comentário de referência.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valores de retorno

`(integer/array)` Retorna:

- uma matriz de objetos;
- a contagem de objetos recuperados, se o parâmetro `countOutput` tiver sido usado.

Exemplos

Recuperar todas as regras de descoberta

Recupera todas as regras de descoberta configuradas e as verificações de descoberta que elas usam.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "drule.get",
    "params": {
        "output": "extend",
        "selectDChecks": "extend"
    },
    "id": 1
}
```

Resposta:

```
{
    "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
}
```


Veja também

- Host descoberto
- Verificação de descoberta

Fonte

CDRule::get() em *ui/include/classes/api/services/CDRule.php*.

drule.update

Descrição

`object drule.update(object/array discoveryRules)`

Este método permite atualizar regras de descoberta existentes.

Note:
Este método está disponível apenas para os tipos de usuário *Admin* e *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object/array)` Propriedades da regra de descoberta a serem atualizadas.

A propriedade `druleid` deve ser definida para cada regra de descoberta, todas as outras propriedades são opcionais. Somente as propriedades passadas serão atualizadas, todas as outras permanecerão inalteradas.

Além das propriedades padrão da regra de descoberta, o método aceita os seguintes parâmetros.

Parâmetro	Tipo	Descrição
dchecks	array	Verificações de descoberta para substituir as verificações existentes.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs da descoberta atualizada regras sob a propriedade `druleids`.

Exemplos

Alterar o intervalo de IP de uma regra de descoberta

Altere o intervalo de IP de uma regra de descoberta para "192.168.2.1-255".

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "drule.update",
    "params": {
        "druleid": "6",
        "iprange": "192.168.2.1-255"
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "druleids": [
            "6"
        ]
    },
    "id": 1
}
```

Veja também

- Discovery check

Fonte

CDRule::update() em *ui/include/classes/api/services/CDRule.php*.

Regra de LLD

Esta classe foi projetada para trabalhar com regras de descoberta de baixo nível.

Referências de objetos:

- Regra de LLD
  - Cabeçalho HTTP
  - Campo de consulta HTTP
- Filtro de regra de LLD
  - Condição de filtro de regra de LLD
- Caminho de macro de LLD
- Pré-processamento de regra de LLD
- Substituições de regra de LLD
  - Filtro de substituição de regra de LLD
    - Condição de filtro de substituição de regra de LLD
  - Operação de substituição de regra de LLD
    - Status da operação de substituição de regra de LLD
    - Descoberta de operação de substituição de regra de LLD
    - Período de operação de substituição de regra de LLD
    - Histórico de operação de substituição de regra de LLD
    - Tendências de operação de substituição de regra de LLD
    - Gravidade da operação de substituição de regra de LLD
    - Tag de operação de substituição de regra de LLD
    - Template de operação de substituição de regra de LLD
    - Inventário de operação de substituição de regra de LLD

Métodos disponíveis:

- discoveryrule.copy - copiar regras de LLD
- discoveryrule.create - criar novas regras de LLD
- discoveryrule.delete - excluir regras de LLD
- discoveryrule.get - recuperar regras de LLD
- discoveryrule.update - atualizar regras de LLD

Objeto de regra de LLD

Os seguintes objetos estão diretamente relacionados à API `discoveryrule`.

Regra de LLD

O objeto de regra de descoberta de baixo nível possui as seguintes propriedades.

Propriedade	Tipo	Descrição
itemid	ID	ID da regra de LLD. Comportamento da propriedade: - *somente leitura* - *obrigatório* para operações de atualização
delay	string	Intervalo de atualização da regra de LLD. Aceita segundos ou unidade de tempo com sufixo (por exemplo, 30s, 1m, 2h, 1d) e, opcionalmente, um ou mais intervalos personalizados, todos separados por ponto e vírgula. Os intervalos personalizados podem ser uma mistura de intervalos flexíveis e de agendamento. Aceita macros de usuário. Se usado, o valor deve ser uma única macro. Múltiplas macros ou macros misturadas com texto não são suportadas. Intervalos flexíveis podem ser escritos como duas macros separadas por uma barra (por exemplo, `{$FLEX_INTERVAL}/{$FLEX_PERIOD}`). Exemplo: `1h;wd1-5h9-18;{$Macro1}/1-7,00:00-24:00;0/6-7,12:00-24:00;{$Macr` Comportamento da propriedade: - *obrigatório* se `type` for definido como ”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), ou se `type` for definido como ”Zabbix agent (active)” (7) e `key_` não contiver ”mqtt.get”
hostid	ID	ID do host ao qual a regra de LLD pertence. Comportamento da propriedade: - *constante* - *obrigatório* para operações de criação
interfaceid	ID	ID da interface do host da regra de LLD. Comportamento da propriedade: - *obrigatório* se a regra de LLD pertencer ao host e `type` for definido como ”Zabbix agent”, ”IPMI agent”, ”JMX agent” ou ”SNMP agent” - *suportado* se a regra de LLD pertencer ao host e `type` for definido como ”Simple check”, ”External check”, ”SSH agent”, ”TELNET agent” ou ”HTTP agent”
key_	string	Chave da regra de LLD. Comportamento da propriedade: - *obrigatório* para operações de criação - *somente leitura* para objetos herdados
name	string	Nome da regra de LLD. Comportamento da propriedade: - *obrigatório* para operações de criação - *somente leitura* para objetos herdados

Propriedade	Tipo	Descrição
type	integer	Tipo da regra de LLD. Valores possíveis: 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. Comportamento da propriedade: - *obrigatório* para operações de criação - *somente leitura* para objetos herdados
url	string	String de URL. Suporta macros de usuário, {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {ITEM.ID}, {ITEM.KEY}. Comportamento da propriedade: - *obrigatório* se `type` for definido como ”HTTP agent” - *somente leitura* para objetos herdados
allow_traps	integer	Permitir popular valor de forma semelhante ao item trapper. Valores possíveis: 0 - *(padrão)* Não permitir aceitar dados recebidos; 1 - Permitir aceitar dados recebidos. Comportamento da propriedade: - *suportado* se `type` for definido como ”HTTP agent”
authtype	integer	Método de autenticação. Valores possíveis se `type` for definido como ”SSH agent”: 0 - *(padrão)* senha; 1 - chave pública. Valores possíveis se `type` for definido como ”HTTP agent”: 0 - *(padrão)* nenhum; 1 - básico; 2 - NTLM; 3 - Kerberos; 4 - Digest. Comportamento da propriedade: - *suportado* se `type` for definido como ”SSH agent” ou ”HTTP agent” - *somente leitura* para objetos herdados (se `type` for definido como ”HTTP agent”)
description	string	Descrição da regra de LLD.
error	string	Texto de erro se houver problemas ao atualizar o valor da regra de LLD. Comportamento da propriedade: - *somente leitura*

Propriedade	Tipo	Descrição
follow_redirects	integer	Seguir redirecionamentos de resposta ao coletar dados. Valores possíveis: 0 - Não seguir redirecionamentos; 1 - *(padrão)* Seguir redirecionamentos. Comportamento da propriedade: - *suportado* se `type` for definido como ”HTTP agent” - *somente leitura* para objetos herdados
headers	array	Array de cabeçalhos que serão enviados ao executar uma requisição HTTP. Comportamento da propriedade: - *suportado* se `type` for definido como ”HTTP agent” - *somente leitura* para objetos herdados
http_proxy	string	String de conexão do proxy HTTP(S). Comportamento da propriedade: - *suportado* se `type` for definido como ”HTTP agent” - *somente leitura* para objetos herdados
ipmi_sensor	string	Sensor IPMI. Comportamento da propriedade: - *obrigatório* se `type` for definido como ”IPMI agent” e `key_` não for definido como ”ipmi.get” - *suportado* se `type` for definido como ”IPMI agent” - *somente leitura* para objetos herdados
jmx_endpoint	string	String de conexão personalizada do JMX agent. Padrão: service:jmx:rmi:///jndi/rmi://{HOST.CONN}:{HOST.PORT}/jmxrmi Comportamento da propriedade: - *suportado* se `type` for definido como ”JMX agent”
lifetime	string	Período de tempo após o qual os itens que não são mais descobertos serão excluídos. Aceita segundos, unidade de tempo com sufixo ou uma macro de usuário. Padrão: 7d.
lifetime_type	integer	Cenário para excluir recursos de LLD perdidos. Valores possíveis: 0 - *(padrão)* Excluir após atingir o limite de tempo de vida; 1 - Não excluir; 2 - Excluir imediatamente.
enabled_lifetime	string	Período de tempo após o qual os itens que não são mais descobertos serão desabilitados. Aceita segundos, unidade de tempo com sufixo ou uma macro de usuário. Padrão: 0.
enabled_lifetime_type	integer	Cenário para desabilitar recursos de LLD perdidos. Valores possíveis: 0 - Desabilitar após atingir o limite de tempo de vida; 1 - Não desabilitar; 2 - *(padrão)* Desabilitar imediatamente.

Propriedade	Tipo	Descrição
master_itemid	ID	ID do item mestre. É permitida recursão de até 3 itens dependentes e o número máximo de itens dependentes igual a 999. A regra de descoberta não pode ser item mestre para outra regra de descoberta. Comportamento da propriedade: - *obrigatório* se `type` for definido como ”Dependent item” - *somente leitura* para objetos herdados
output_format	integer	Se a resposta deve ser convertida para JSON. Valores possíveis: 0 - *(padrão)* Armazenar bruto; 1 - Converter para JSON. Comportamento da propriedade: - *suportado* se `type` for definido como ”HTTP agent” - *somente leitura* para objetos herdados
params	string	Parâmetros adicionais dependendo do tipo da regra de LLD: - script executado para regras de LLD SSH e Telnet; - consulta SQL para regras de LLD de monitoramento de banco de dados; - fórmula para regras de LLD calculadas; - o script para regras de LLD de script e browser. Comportamento da propriedade: - *obrigatório* se `type` for definido como ”Database monitor”, ”SSH agent”, ”TELNET agent”, ”Script” ou ”Browser” - *somente leitura* para objetos herdados (se `type` for definido como ”Script” ou ”Browser”)
parameters	object/array	Parâmetros adicionais se `type` for definido como ”Script” ou ”Browser”. Array de objetos com propriedades `name` e `value`, onde `name` deve ser único. Comportamento da propriedade: - *suportado* se `type` for definido como ”Script” ou ”Browser” - *somente leitura* para objetos herdados
password	string	Senha para autenticação. Comportamento da propriedade: - *obrigatório* se `type` for definido como ”JMX agent” e `username` for definido - *suportado* se `type` for definido como ”Simple check”, ”Database monitor”, ”SSH agent”, ”TELNET agent” ou ”HTTP agent” - *somente leitura* para objetos herdados (se `type` for definido como ”HTTP agent”)
post_type	integer	Tipo de corpo de dados postados armazenados na propriedade `posts`. Valores possíveis: 0 - *(padrão)* Dados brutos; 2 - Dados JSON; 3 - Dados XML. Comportamento da propriedade: - *suportado* se `type` for definido como ”HTTP agent” - *somente leitura* para objetos herdados

Propriedade	Tipo	Descrição
posts	string	Dados do corpo da requisição HTTP(S). Comportamento da propriedade: - *obrigatório* se `type` for definido como ”HTTP agent” e `post_type` for definido como ”JSON data” ou ”XML data” - *suportado* se `type` for definido como ”HTTP agent” e `post_type` for definido como ”Raw data” - *somente leitura* para objetos herdados
privatekey	string	Nome do arquivo de chave privada. Comportamento da propriedade: - *obrigatório* se `type` for definido como ”SSH agent” e `authtype` for definido como ”public key”
publickey	string	Nome do arquivo de chave pública. Comportamento da propriedade: - *obrigatório* se `type` for definido como ”SSH agent” e `authtype` for definido como ”public key”
query_fields	array	Array de campos de consulta que serão enviados ao executar uma requisição HTTP. Comportamento da propriedade: - *suportado* se `type` for definido como ”HTTP agent” - *somente leitura* para objetos herdados
request_method	integer	Tipo de método de requisição. Valores possíveis: 0 - *(padrão)* GET; 1 - POST; 2 - PUT; 3 - HEAD. Comportamento da propriedade: - *suportado* se `type` for definido como ”HTTP agent” - *somente leitura* para objetos herdados
retrieve_mode	integer	Qual parte da resposta deve ser armazenada. Valores possíveis se `request_method` for definido como ”GET”, ”POST” ou ”PUT”: 0 - *(padrão)* Corpo; 1 - Cabeçalhos; 2 - Corpo e cabeçalhos serão armazenados. Valores possíveis se `request_method` for definido como ”HEAD”: 1 - Cabeçalhos. Comportamento da propriedade: - *suportado* se `type` for definido como ”HTTP agent” - *somente leitura* para objetos herdados
snmp_oid	string	SNMP OID. Comportamento da propriedade: - *obrigatório* se `type` for definido como ”SNMP agent” - *somente leitura* para objetos herdados
ssl_cert_file	string	Caminho do arquivo de chave pública SSL. Comportamento da propriedade: - *suportado* se `type` for definido como ”HTTP agent” - *somente leitura* para objetos herdados

Propriedade	Tipo	Descrição
ssl_key_file	string	Caminho do arquivo de chave privada SSL. Comportamento da propriedade: - *suportado* se `type` for definido como ”HTTP agent” - *somente leitura* para objetos herdados
ssl_key_password	string	Senha para o arquivo de chave SSL. Comportamento da propriedade: - *suportado* se `type` for definido como ”HTTP agent” - *somente leitura* para objetos herdados
state	integer	Estado da regra de LLD. Valores possíveis: 0 - *(padrão)* normal; 1 - não suportado. Comportamento da propriedade: - *somente leitura*
status	integer	Status da regra de LLD. Valores possíveis: 0 - *(padrão)* regra de LLD habilitada; 1 - regra de LLD desabilitada.
status_codes	string	Intervalos de códigos de status HTTP necessários, separados por vírgulas. Também suporta macros de usuário como parte da lista separada por vírgulas. Exemplo: 200,200-{$M},{$M},200-400 Comportamento da propriedade: - *suportado* se `type` for definido como ”HTTP agent” - *somente leitura* para objetos herdados
templateid	ID	ID da regra de LLD da template pai. Comportamento da propriedade: - *somente leitura*
timeout	string	Tempo limite da requisição de coleta de dados do item. Aceita segundos ou unidade de tempo com sufixo (por exemplo, 30s, 1m). Também aceita macros de usuário. Faixa de valores possíveis: 1-600s. Padrão: ”” - usar configurações do proxy/globais. Comportamento da propriedade: - *suportado* se `type` for definido como ”Zabbix agent” (0), ”Simple check” (3) e `key_` não começar com ”`vmware.`” e ”`icmpping`”, ”Zabbix agent (active)” (7), ”External check” (10), ”Database monitor” (11), ”SSH agent” (13), ”TELNET agent” (14), ”HTTP agent” (19), ”SNMP agent” (20) e `snmp_oid` começar com ”`walk[`” ou ”`get[`”, ”Script” (21), ”Browser” (22) - *somente leitura* para objetos herdados
trapper_hosts	string	Hosts permitidos. Comportamento da propriedade: - *suportado* se `type` for definido como ”Zabbix trapper”, ou se `type` for definido como ”HTTP agent” e `allow_traps` for definido como ”Allow to accept incoming data”

Propriedade	Tipo	Descrição
username	string	Nome de usuário para autenticação. Comportamento da propriedade: - *obrigatório* se `type` for definido como ”SSH agent”, ”TELNET agent”, ou se `type` for definido como ”JMX agent” e `password` for definido - *suportado* se `type` for definido como ”Simple check”, ”Database monitor” ou ”HTTP agent” - *somente leitura* para objetos herdados (se `type` for definido como ”HTTP agent”)
uuid	string	Identificador único universal, usado para vincular regras de LLD importadas às já existentes. Gerado automaticamente, se não fornecido. Comportamento da propriedade: - *suportado* se a regra de LLD pertencer a uma template
verify_host	integer	Se deve validar que o nome do host para a conexão corresponde ao do certificado do host. Valores possíveis: 0 - *(padrão)* Não validar; 1 - Validar. Comportamento da propriedade: - *suportado* se `type` for definido como ”HTTP agent” - *somente leitura* para objetos herdados
verify_peer	integer	Se deve validar que o certificado do host é autêntico. Valores possíveis: 0 - *(padrão)* Não validar; 1 - Validar. Comportamento da propriedade: - *suportado* se `type` for definido como ”HTTP agent” - *somente leitura* para objetos herdados

Cabeçalho HTTP

O objeto header possui as seguintes propriedades:

Propriedade	Tipo	Descrição
name	string	Nome do cabeçalho HTTP. Comportamento da propriedade: - *obrigatório*
value	string	Valor do cabeçalho. Comportamento da propriedade: - *obrigatório*

Campo de consulta HTTP

O objeto de campo de consulta define um nome e valor que é usado para especificar um parâmetro de URL. Ele possui as seguintes propriedades:

Propriedade	Tipo	Descrição
name	string	Nome do parâmetro. Comportamento da propriedade: - *obrigatório*
value	string	Valor do parâmetro. Comportamento da propriedade: - *obrigatório*

Filtro de regra LLD

O objeto de filtro de regra LLD define um conjunto de condições que podem ser usadas para filtrar objetos descobertos. Ele possui as seguintes propriedades:

Propriedade	Tipo	Descrição
conditions	object/array	Conjunto de condições de filtro a serem usadas para filtrar resultados. As condições serão classificadas na ordem de sua colocação na fórmula. Comportamento da propriedade: - *obrigatório*
evaltype	integer	Método de avaliação da condição de filtro. Valores possíveis: 0 - E/OU; 1 - E; 2 - OU; 3 - Expressão personalizada. Comportamento da propriedade: - *obrigatório*
eval_formula	string	Expressão gerada que será usada para avaliar as condições do filtro. A expressão contém IDs que referenciam condições de filtro específicas pelo seu `formulaid`. O valor de `eval_formula` é igual ao valor de `formula` para filtros com uma expressão personalizada. Comportamento da propriedade: - *somente leitura*
formula	string	Expressão definida pelo usuário a ser usada para avaliar as condições de filtros com uma expressão personalizada. A expressão deve conter IDs que referenciem condições de filtro específicas pelo seu `formulaid`. Os IDs usados na expressão devem corresponder exatamente aos definidos nas condições do filtro: nenhuma condição pode permanecer não utilizada ou omitida. Comportamento da propriedade: - *obrigatório* se `evaltype` estiver definido como ”expressão personalizada”

Condição de filtro da regra LLD

O objeto de condição de filtro da regra LLD define uma verificação separada a ser realizada no valor de uma macro LLD. Ele possui as seguintes propriedades:

Propriedade	Tipo	Descrição
macro	string	Macro LLD para realizar a verificação. Comportamento da propriedade: - *obrigatório*
value	string	Valor a ser comparado. Comportamento da propriedade: - *obrigatório* se `operator` estiver definido como ”corresponde à expressão regular” ou ”não corresponde à expressão regular”
formulaid	string	ID exclusivo arbitrário usado para referenciar a condição a partir de uma expressão personalizada. Só pode conter letras maiúsculas. O ID deve ser definido pelo usuário ao modificar as condições do filtro, mas será gerado novamente ao solicitá-las posteriormente. Comportamento da propriedade: - *obrigatório* se `evaltype` do objeto de filtro da regra LLD estiver definido como ”expressão personalizada”
operator	integer	Operador da condição. Valores possíveis: 8 - *(padrão)* corresponde à expressão regular; 9 - não corresponde à expressão regular; 12 - existe; 13 - não existe.

Note:
Para entender melhor como usar filtros com vários tipos de expressões, consulte exemplos nas páginas dos métodos discoveryrule.get e discoveryrule.create.

Caminho da macro LLD

O caminho da macro LLD possui as seguintes propriedades:

Propriedade	Tipo	Descrição
lld_macro	string	Macro LLD. Comportamento da propriedade: - *obrigatório*
path	string	Seletor para o valor que será atribuído à macro correspondente. Comportamento da propriedade: - *obrigatório*

Pré-processamento de regra LLD

O objeto de pré-processamento de regra LLD possui as seguintes propriedades.

Propriedade	Tipo	Descrição
type	integer	O tipo de opção de pré-processamento. Valores possíveis: 5 - Expressão regular; 11 - XML XPath; 12 - JSONPath; 14 - Corresponde à expressão regular; 15 - Não corresponde à expressão regular; 16 - Verificar erro em JSON; 17 - Verificar erro em XML; 20 - Descartar inalterado com heartbeat; 21 - JavaScript; 23 - Prometheus para JSON; 24 - CSV para JSON; 25 - Substituir; 27 - XML para JSON; 28 - Valor SNMP walk; 29 - SNMP walk para JSON; 30 - Valor SNMP get. Comportamento da propriedade: - *obrigatório*
params	string	Parâmetros adicionais usados pela opção de pré-processamento. Vários parâmetros são separados pelo caractere de nova linha (\n). Comportamento da propriedade: - *obrigatório* se `type` for definido como ”Expressão regular” (5), ”XML XPath” (11), ”JSONPath” (12), ”Corresponde à expressão regular” (14), ”Não corresponde à expressão regular” (15), ”Verificar erro em JSON” (16), ”Verificar erro em XML” (17), ”Descartar inalterado com heartbeat” (20), ”JavaScript” (21), ”Prometheus para JSON” (23), ”CSV para JSON” (24), ”Substituir” (25), ”Valor SNMP walk” (28), ”SNMP walk para JSON” (29) ou ”Valor SNMP get” (30)
error_handler	integer	Tipo de ação usada em caso de falha na etapa de pré-processamento. Valores possíveis: 0 - Mensagem de erro definida pelo Zabbix server; 1 - Descartar valor; 2 - Definir valor personalizado; 3 - Definir mensagem de erro personalizada. Comportamento da propriedade: - *obrigatório* se `type` for definido como ”Expressão regular” (5), ”XML XPath” (11), ”JSONPath” (12), ”Corresponde à expressão regular” (14), ”Não corresponde à expressão regular” (15), ”Verificar erro em JSON” (16), ”Verificar erro em XML” (17), ”Prometheus para JSON” (23), ”CSV para JSON” (24), ”XML para JSON” (27), ”Valor SNMP walk” (28), ”SNMP walk para JSON” (29) ou ”Valor SNMP get” (30)
error_handler_params	string	Parâmetros do manipulador de erro. Comportamento da propriedade: - *obrigatório* se `error_handler` for definido como ”Definir valor personalizado” ou ”Definir mensagem de erro personalizada”

Os seguintes parâmetros e manipuladores de erro são suportados para cada tipo de pré-processamento.

Tipo de pré-processamento	Nome	Parâmetro 1	Parâmetro 2	Parâmetro 3	Manipuladores de erro suportados
5	Expressão regular	padrão[1]	saída[2]		0, 1, 2, 3
11	XML XPath	caminho[3]			0, 1, 2, 3
12	JSONPath	caminho[3]			0, 1, 2, 3
14	Corresponde à expressão regular	padrão[1]			0, 1, 2, 3
15	Não corresponde à expressão regular	padrão[1]			0, 1, 2, 3
16	Verificar erro em JSON	caminho[3]			0, 1, 2, 3
17	Verificar erro em XML	caminho[3]			0, 1, 2, 3
20	Descartar inalterado com heartbeat	segundos^4, 5^			
21	JavaScript	script[2]			
23	Prometheus para JSON	padrão^5, 6^			0, 1, 2, 3
24	CSV para JSON	caractere[2]	caractere[2]	0,1	0, 1, 2, 3
25	Substituir	string de busca[2]	substituição[2]		
27	XML para JSON				0, 1, 2, 3
28	Valor SNMP walk	OID[2]	Formato: 0 - Inalterado 1 - UTF-8 de Hex-STRING 2 - MAC de Hex-STRING 3 - Inteiro de BITS		0, 1, 2, 3

Tipo de pré-processamento	Nome	Parâmetro 1	Parâmetro 2	Parâmetro 3	Manipuladores de erro suportados
29	SNMP walk para JSON[7]	Nome do campo[2]	Prefixo OID[2]	Formato: 0 - Inalterado 1 - UTF-8 de Hex-STRING 2 - MAC de Hex-STRING 3 - Inteiro de BITS	0, 1, 2, 3
30	Valor SNMP get	Formato: 1 - UTF-8 de Hex-STRING 2 - MAC de Hex-STRING 3 - Inteiro de BITS			0, 1, 2, 3

[1] expressão regular
 [2] string
 [3] JSONPath ou XML XPath
 [4] inteiro positivo (com suporte a sufixos de tempo, por exemplo, 30s, 1m, 2h, 1d)
 [5] macro de usuário
 [6] padrão Prometheus seguindo a sintaxe: `<nome da métrica>{<nome do rótulo>="<valor do rótulo>", ...} == <valor>`. Cada componente do padrão Prometheus (métrica, nome do rótulo, valor do rótulo e valor da métrica) pode ser macro de usuário.
 [7] Suporta vários registros "Nome do campo,Prefixo OID,Formato" delimitados por um caractere de nova linha.

Substituições de regras de LLD

O objeto de substituições de regras de LLD define um conjunto de regras (filtros, condições e operações) que são usadas para substituir propriedades de diferentes objetos de protótipo. Possui as seguintes propriedades:

Propriedade	Tipo	Descrição
name	string	Nome único da substituição. Comportamento da propriedade: - *obrigatório*
step	integer	Número de ordem único da substituição. Comportamento da propriedade: - *obrigatório*
stop	integer	Parar o processamento das próximas substituições se corresponder. Valores possíveis: 0 - *(padrão)* não para o processamento das substituições; 1 - para o processamento das substituições se o filtro corresponder.
filter	object	Filtro de substituição.
operations	object/array	Operações de substituição.

Filtro de substituição da regra LLD

O objeto de filtro de substituição da regra LLD define um conjunto de condições que, se corresponderem ao objeto descoberto, a substituição é aplicada. Possui as seguintes propriedades:

Propriedade	Tipo	Descrição
conditions	object/array	Conjunto de condições de filtro de substituição a serem usadas para corresponder aos objetos descobertos. As condições serão classificadas na ordem de sua colocação na fórmula. Comportamento da propriedade: - *obrigatório*

Propriedade	Tipo	Descrição
evaltype	integer	Método de avaliação da condição do filtro de substituição. Valores possíveis: 0 - E/Ou; 1 - E; 2 - Ou; 3 - Expressão personalizada. Comportamento da propriedade: - *obrigatório*
eval_formula	string	Expressão gerada que será usada para avaliar as condições do filtro de substituição. A expressão contém IDs que referenciam condições específicas do filtro de substituição pelo seu `formulaid`. O valor de `eval_formula` é igual ao valor de `formula` para filtros com uma expressão personalizada. Comportamento da propriedade: - *somente leitura*
formula	string	Expressão definida pelo usuário a ser usada para avaliar as condições dos filtros de substituição com uma expressão personalizada. A expressão deve conter IDs que referenciem condições específicas do filtro de substituição pelo seu `formulaid`. Os IDs usados na expressão devem corresponder exatamente aos definidos nas condições do filtro de substituição: nenhuma condição pode permanecer não utilizada ou omitida. Comportamento da propriedade: - *obrigatório* se `evaltype` estiver definido como ”expressão personalizada”

Condição de filtro de substituição de regra LLD

O objeto de condição de filtro de substituição de regra LLD define uma verificação separada a ser realizada no valor de uma macro LLD. Possui as seguintes propriedades:

Propriedade	Tipo	Descrição
macro	string	Macro LLD para realizar a verificação. Comportamento da propriedade: - *obrigatório*
value	string	Valor para comparar. Comportamento da propriedade: - *obrigatório* se `operator` estiver definido como ”corresponde à expressão regular” ou ”não corresponde à expressão regular”
formulaid	string	ID exclusivo arbitrário usado para referenciar a condição a partir de uma expressão personalizada. Só pode conter letras maiúsculas. O ID deve ser definido pelo usuário ao modificar as condições do filtro, mas será gerado novamente ao solicitá-las posteriormente. Comportamento da propriedade: - *obrigatório* se `evaltype` do objeto de filtro de substituição de regra LLD estiver definido como ”expressão personalizada”
operator	integer	Operador de condição. Valores possíveis: 8 - *(padrão)* corresponde à expressão regular; 9 - não corresponde à expressão regular; 12 - existe; 13 - não existe.

Operação de substituição de regra LLD

A operação de substituição de regra LLD é uma combinação de condições e ações a serem realizadas no objeto protótipo. Possui as seguintes propriedades:

Propriedade	Tipo	Descrição
operationobject	integer	Tipo de objeto descoberto para realizar a ação. Valores possíveis: 0 - Protótipo de item; 1 - Protótipo de trigger; 2 - Protótipo de gráfico; 3 - Protótipo de host. Comportamento da propriedade: - *obrigatório*
operator	integer	Operador da condição de substituição. Valores possíveis: 0 - *(padrão)* igual; 1 - diferente; 2 - contém; 3 - não contém; 8 - corresponde; 9 - não corresponde.
value	string	Padrão para corresponder ao nome do protótipo de item, trigger, gráfico ou host, dependendo do objeto selecionado.
opstatus	object	Objeto de status da operação de substituição para objetos protótipo de item, trigger e host.
opdiscover	object	Objeto de status de descoberta da operação de substituição (todos os tipos de objeto).
opperiod	object	Objeto de período da operação de substituição (intervalo de atualização) para objeto protótipo de item.
ophistory	object	Objeto de histórico da operação de substituição para objeto protótipo de item.
optrends	object	Objeto de tendências da operação de substituição para objeto protótipo de item.
opseverity	object	Objeto de severidade da operação de substituição para objeto protótipo de trigger.
optag	object/array	Objeto de tag da operação de substituição para objetos protótipo de trigger e host.
optemplate	object/array	Objeto de template da operação de substituição para objeto protótipo de host.
opinventory	object	Objeto de inventário da operação de substituição para objeto protótipo de host.

Status da operação de substituição da regra LLD

Status da operação de substituição da regra LLD que é definida para o objeto descoberto. Ele possui as seguintes propriedades:

Propriedade	Tipo	Descrição
status	integer	Substitui o status para o objeto selecionado. Valores possíveis: 0 - Criar habilitado; 1 - Criar desabilitado. Comportamento da propriedade: - *obrigatório*

Operação de descoberta de substituição de regra LLD

Status de descoberta da operação de substituição de regra LLD que é definido para o objeto descoberto. Possui as seguintes propriedades:

Propriedade	Tipo	Descrição
discover	integer	Substitui o status de descoberta para o objeto selecionado. Valores possíveis: 0 - Sim, continue descobrindo os objetos; 1 - Não, novos objetos não serão descobertos e os existentes serão marcados como perdidos. Comportamento da propriedade: - *obrigatório*

Período de operação de substituição da regra LLD

O período de operação de substituição da regra LLD é um valor de intervalo de atualização que é definido para o item descoberto. Possui as seguintes propriedades:

Propriedade	Tipo	Descrição
delay	string	Substitui o intervalo de atualização do protótipo de item. Aceita segundos ou unidade de tempo com sufixo (por exemplo, 30s, 1m, 2h, 1d) e, opcionalmente, um ou mais intervalos personalizados, todos separados por ponto e vírgula. Os intervalos personalizados podem ser uma mistura de intervalos flexíveis e de agendamento. Aceita macros de usuário ou macros LLD. Se usadas, o valor deve ser uma única macro. Múltiplas macros ou macros misturadas com texto não são suportadas. Intervalos flexíveis podem ser escritos como duas macros separadas por uma barra (por exemplo, `{$FLEX_INTERVAL}/{$FLEX_PERIOD}`). Exemplo: `1h;wd1-5h9-18;{$Macro1}/1-7,00:00-24:00;0/6-7,12:00-24:00;{$Macr` Comportamento da propriedade: - *obrigatório*

Histórico de operação de substituição de regra LLD

Valor do histórico de operação de substituição de regra LLD que é definido para o item descoberto. Possui as seguintes propriedades:

Propriedade	Tipo	Descrição
history	string	Substitui o histórico do protótipo de item, que é uma unidade de tempo de quanto tempo os dados do histórico devem ser armazenados. Também aceita macro de usuário e macro LLD. Comportamento da propriedade: - *obrigatório*

Tendências de operação de substituição de regra LLD

Valor de tendências de operação de substituição de regra LLD que é definido para o item descoberto. Possui as seguintes propriedades:

Propriedade	Tipo	Descrição
trends	string	Substitui as tendências do protótipo de item, que é uma unidade de tempo de quanto tempo os dados de tendências devem ser armazenados. Também aceita macro de usuário e macro LLD. Comportamento da propriedade: - *obrigatório*

Gravidade da operação de substituição da regra LLD

Valor de gravidade da operação de substituição da regra LLD que é definido para o trigger descoberto. Possui as seguintes propriedades:

Propriedade	Tipo	Descrição
severity	integer	Substitui a gravidade do protótipo de trigger. Valores possíveis: 0 - *(padrão)* não classificado; 1 - informação; 2 - aviso; 3 - média; 4 - alta; 5 - desastre. Comportamento da propriedade: - *obrigatório*

Tag de operação de substituição de regra de LLD

O objeto de tag de operação de substituição de regra de LLD contém o nome e o valor da tag que são definidos para o objeto descoberto. Ele possui as seguintes propriedades:

Propriedade	Tipo	Descrição
tag	string	Novo nome da tag. Comportamento da propriedade: - *obrigatório*
value	string	Novo valor da tag.

Template de operação de sobrescrição de regra LLD

Objeto de template de operação de sobrescrição de regra LLD que está vinculado ao host descoberto. Ele possui as seguintes propriedades:

Propriedade	Tipo	Descrição
templateid	ID	Substitui o template dos templates vinculados ao protótipo de host. Comportamento da propriedade: - *obrigatório*

Inventário de operação de substituição de regra LLD

Valor do modo de inventário da operação de substituição da regra LLD que é definido para o host descoberto. Possui as seguintes propriedades:

Propriedade	Tipo	Descrição
inventory_mode	integer	Substitui o modo de inventário do protótipo de host. Valores possíveis: -1 - desabilitado; 0 - *(padrão)* manual; 1 - automático. Comportamento da propriedade: - *obrigatório*

discoveryrule.copy

Este método está obsoleto e será removido no futuro. Em vez disso, pode configurar regras LLD em modelos e aplicar esses modelos a outros modelos ou anfitriões, copiando efetivamente as regras LLD para os alvos especificados.

Descrição

`object discoveryrule.copy(object parameters)`

Este método permite copiar regras LLD com todos os protótipos para os hosts fornecidos.

Note:
Este método está disponível apenas para os tipos de usuário *Admin* e *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object)` Parâmetros que definem as regras de LLD a serem copiadas e os hosts de destino.

Parâmetro	Tipo	Descrição
discoveryids	array	IDs das regras de LLD a serem copiadas.
hostids	array	IDs dos hosts para os quais as regras de LLD serão copiadas.

Valores de retorno

`(boolean)` Retorna `true` se a cópia foi bem sucedida.

Exemplos

Copiar uma regra de LLD para vários hosts

Copie uma regra de LLD para dois hosts.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.copy",
    "params": {
        "discoveryids": [
            "27426"
        ],
        "hostids": [
            "10196",
            "10197"
        ]
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": true,
    "id": 1
}
```

Fonte

CDiscoveryRule::copy() em *ui/include/classes/api/services/CDiscoveryRule.php*.

discoveryrule.create

Descrição

`object discoveryrule.create(object/array lldRules)`

Este método permite criar novas regras de LLD.

> **Note:**
> Este método está disponível apenas para os tipos de usuário *Admin* e *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Consulte Funções de usuário para mais informações.

Parâmetros

`(object/array)` Regras de LLD a serem criadas.

Além das propriedades padrão da regra de LLD, o método aceita os seguintes parâmetros.

Parâmetro	Tipo	Descrição
filter	object	Filtro da regra de LLD para a regra de LLD.
preprocessing	object/array	Opções de pré-processamento da regra de LLD.
lld_macro_paths	object/array	Opções de lld_macro_path da regra de LLD.
overrides	object/array	Opções de substituições da regra de LLD.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs das regras LLD criadas na propriedade `itemids`. A ordem dos IDs retornados corresponde à ordem das regras LLD passadas.

Exemplos

Criando uma regra de LLD

Crie uma regra de LLD do agent Zabbix para descobrir sistemas de arquivos montados. Os items descobertos serão atualizados a cada 30 segundos.

Requisição:

```
{
    "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
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
```

```
        "itemids": [
            "27665"
        ]
    },
    "id": 1
}
```

Usando um filtro

Crie uma regra LLD com um conjunto de condições para filtrar os resultados. As condições serão agrupadas usando o operador lógico "and".

Requisição:

```
{
    "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
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "27665"
        ]
    },
    "id": 1
}
```

Criando uma regra LLD com caminhos de macro

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.create",
    "params": {
        "name": "LLD rule with LLD macro paths",
        "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
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "27665"
        ]
    },
    "id": 1
}
```

Usando um filtro de expressão personalizada

Crie uma regra LLD com um filtro que usará uma expressão personalizada para avaliar as condições. A regra LLD deve descobrir apenas objetos cujo valor da macro "{#MACRO1}" corresponda tanto à expressão regular "regex1" quanto à "regex2", e o valor de "{#MACRO2}" corresponda a "regex3" ou "regex4". Os IDs de fórmula "A", "B", "C" e "D" foram escolhidos arbitrariamente.

Requisição:

```
{
    "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
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "27665"
        ]
    },
    "id": 1
}
```

Usando campos de consulta e cabeçalhos personalizados

Crie uma regra de LLD com campos de consulta e cabeçalhos personalizados.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.create",
    "params": {
        "hostid": "10257",
        "interfaceid": "5",
        "type": 19,
        "name": "API HTTP agent",
        "key_": "api_discovery_rule",
        "delay": "5s",
        "url": "http://127.0.0.1?discoverer.php",
        "query_fields": [
            {
                "name": "mode",
                "value": "json"
            },
            {
                "name": "elements",
                "value": "2"
            }
        ],
        "headers": [
            {
                "name": "X-Type",
                "value": "api"
            },
            {
```

```
                "name": "Authorization",
                "value": "Bearer mF_A.B5f-2.1JcM"
            }
        ],
        "allow_traps": 1,
        "trapper_hosts": "127.0.0.1"
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "28336"
        ]
    },
    "id": 1
}
```

Criando uma regra LLD com pré-processamento

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.create",
    "params": {
        "name": "Discovery rule with preprocessing",
        "key_": "lld.with.preprocessing",
        "hostid": "10001",
        "ruleid": "27665",
        "type": 0,
        "delay": "60s",
        "interfaceid": "1155",
        "preprocessing": [
            {
                "type": 20,
                "params": "20",
                "error_handler": 0,
                "error_handler_params": ""
            }
        ]
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "44211"
        ]
    },
    "id": 1
}
```

Criando uma regra LLD com substituições

Requisição:

```
{
    "jsonrpc": "2.0",
```

```
    "method": "discoveryrule.create",
    "params": {
        "name": "Descobrir host de banco de dados",
        "key_": "lld.with.overrides",
        "hostid": "10001",
        "type": 0,
        "delay": "60s",
        "interfaceid": "1155",
        "overrides": [
            {
                "name": "Descobrir host MySQL",
                "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": "Host de banco de dados",
                        "opstatus": {
                            "status": "0"
                        },
                        "optemplate": [
                            {
                                "templateid": "10170"
                            }
                        ],
                        "optag": [
                            {
                                "tag": "database",
                                "value": "mysql"
                            }
                        ]
                    }
                ]
            },
            {
                "name": "Descobrir host PostgreSQL",
                "step": "2",
                "stop": "1",
                "filter": {
                    "evaltype": "0",
                    "conditions": [
                        {
                            "macro": "{#UNIT.NAME}",
                            "operator": "8",
                            "value": "^postgresql\\.service$"
                        }
```

```
                    ]
                },
                "operations": [
                    {
                        "operationobject": "3",
                        "operator": "2",
                        "value": "Host de banco de dados",
                        "opstatus": {
                            "status": "0"
                        },
                        "optemplate": [
                            {
                                "templateid": "10263"
                            }
                        ],
                        "optag": [
                            {
                                "tag": "database",
                                "value": "postgresql"
                            }
                        ]
                    }
                ]
            }
        ]
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "30980"
        ]
    },
    "id": 1
}
```

Criar regra de LLD de script

Crie uma coleta de dados simples usando uma regra de LLD de script.

Requisição:

```
{
    "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
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "23865"
        ]
    },
    "id": 1
}
```

Criar regra LLD com um período de tempo especificado para desativação e sem exclusão

Crie uma regra LLD com período de tempo personalizado para desativar a entidade após ela não ser mais descoberta, com a configuração de que nunca será excluída.

Requisição:

```
{
    "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
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "46864"
        ]
    },
    "id": 1
}
```

Veja também

- Filtro de regra de LLD
- Caminhos de macro de LLD
- Pré-processamento de regra de LLD

Fonte

CDiscoveryRule::create() in *ui/include/classes/api/services/CDiscoveryRule.php*.

discoveryrule.delete

Descrição

`object discoveryrule.delete(array lldRuleIds)`

Este método permite excluir regras de LLD.

> **Note:**
> Este método está disponível apenas para os tipos de usuário *Admin* e *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Consulte Funções de usuário para obter mais informações.

Parâmetros

`(array)` IDs das regras LLD a serem excluídas.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs das regras LLD excluídas na propriedade `ruleids`.

Exemplos

Excluindo várias regras de LLD

Exclua duas regras de LLD.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.delete",
    "params": [
        "27665",
        "27668"
    ],
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "ruleids": [
            "27665",
            "27668"
        ]
    },
    "id": 1
}
```

Fonte

CDiscoveryRule::delete() in *ui/include/classes/api/services/CDiscoveryRule.php*.

discoveryrule.get

Descrição

`integer/array discoveryrule.get(object parameters)`

O método permite recuperar regras LLD de acordo com os parâmetros fornecidos.

Note:
Este método está disponível para usuários de qualquer tipo. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object)` Parâmetros que definem a saída desejada.

O método suporta os seguintes parâmetros.

Parâmetro	Tipo	Descrição
itemids	ID/array	Retorna apenas regras LLD com os IDs fornecidos.
groupids	ID/array	Retorna apenas regras LLD que pertencem aos hosts dos grupos fornecidos.
hostids	ID/array	Retorna apenas regras LLD que pertencem aos hosts fornecidos.
inherited	boolean	Se definido como `true`, retorna apenas regras LLD herdadas de um template.
interfaceids	ID/array	Retorna apenas regras LLD que usam as interfaces de host fornecidas.

Parâmetro	Tipo	Descrição
monitored	boolean	Se definido como `true`, retorna apenas regras LLD habilitadas que pertencem a hosts monitorados.
templated	boolean	Se definido como `true`, retorna apenas regras LLD que pertencem a templates.
templateids	ID/array	Retorna apenas regras LLD que pertencem aos templates fornecidos.
selectFilter	query	Retorna uma propriedade `filter` com os dados do filtro usado pela regra LLD.
selectGraphs	query	Retorna uma propriedade `graphs` com protótipos de gráficos que pertencem à regra LLD. Suporta `count`.
selectHostPrototypes	query	Retorna uma propriedade `hostPrototypes` com protótipos de host que pertencem à regra LLD. Suporta `count`.
selectHosts	query	Retorna uma propriedade `hosts` com um array de hosts ao qual a regra LLD pertence.
selectItems	query	Retorna uma propriedade `items` com protótipos de item que pertencem à regra LLD. Suporta `count`.
selectTriggers	query	Retorna uma propriedade `triggers` com protótipos de trigger que pertencem à regra LLD. Suporta `count`.
selectLLDMacroPaths	query	Retorna uma propriedade `lld_macro_paths` com uma lista de macros LLD e caminhos para valores atribuídos a cada macro correspondente.
selectPreprocessing	query	Retorna uma propriedade `preprocessing` com opções de pré-processamento da regra LLD.
selectOverrides	query	Retorna uma propriedade `lld_rule_overrides` com uma lista de filtros de sobrescrita, condições e operações que são realizadas em objetos protótipos.
filter	object	Retorna apenas os resultados que correspondem exatamente ao filtro fornecido. Aceita um objeto, onde as chaves são nomes de propriedades e os valores são um único valor ou um array de valores para comparar. Não suporta propriedades do tipo de dado `text` data type. Suporta propriedades adicionais: `host` - nome técnico do host ao qual a regra LLD pertence.
limitSelects	integer	Limita o número de registros retornados por subconsultas. Aplica-se às seguintes subconsultas: `selectItems`, `selectGraphs`, `selectTriggers`.
sortfield	string/array	Ordena o resultado pelas propriedades fornecidas. Valores possíveis: `itemid`, `name`, `key_`, `delay`, `type`, `status`.
countOutput	boolean	Esses parâmetros são descritos no comentário de referência.
editable	boolean	
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valores de retorno

`(integer/array)` Retorna:

- uma série de objetos;
- a contagem de objetos recuperados, se o parâmetro `countOutput` tiver sido usado.

Exemplos

Recuperando regras de descoberta de um host

Recupere todas as regras de descoberta para um ID de host específico.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.get",
    "params": {
        "output": "extend",
        "hostids": "10202"
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "itemid": "27425",
            "type": "0",
            "snmp_oid": "",
            "hostid": "10202",
            "name": "Network interface discovery",
            "key_": "net.if.discovery",
            "delay": "1h",
            "status": "0",
            "trapper_hosts": "",
            "templateid": "22444",
            "valuemapid": "0",
            "params": "",
            "ipmi_sensor": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "interfaceid": "119",
            "description": "Discovery of network interfaces as defined in global regular expression \"Netw
            "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",
            "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": "Mounted filesystem discovery",
            "key_": "vfs.fs.discovery",
            "delay": "1h",
            "status": "0",
            "trapper_hosts": "",
            "templateid": "22450",
            "valuemapid": "0",
            "params": "",
            "ipmi_sensor": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "interfaceid": "119",
            "description": "Discovery of file systems of different types as defined in global regular expr
            "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",
            "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
}
```

Recuperando condições de filtro

Recupere o nome da regra LLD "24681" e suas condições de filtro. O filtro usa o tipo de avaliação "and", portanto, a propriedade `formula` está vazia e `eval_formula` é gerada automaticamente.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.get",
    "params": {
        "output": ["name"],
        "selectFilter": "extend",
        "itemids": ["24681"]
    },
    "id": 1
}
```

Resposta:

```
{
    "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
}
```

Recuperar regra de LLD por URL

Recupera a regra de LLD para o host pelo valor do campo URL da regra. Apenas a correspondência exata da string de URL definida para a regra de LLD é suportada.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.get",
    "params": {
        "hostids": "10257",
        "filter": {
            "type": 19,
            "url": "http://127.0.0.1/discoverer.php"
        }
    },
    "id": 1
}
```

Resposta:

```
{
    "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": "",
            "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",
            "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
}
```

Recuperar regra de LLD com substituições

Recupera uma regra de LLD que possui várias configurações de substituição.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.get",
    "params": {
        "output": ["name"],
        "itemids": "30980",
        "selectOverrides": ["name", "step", "stop", "filter", "operations"]
    },
    "id": 1
}
```

Resposta:

```
{
    "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
}
```

Veja também

- Protótipo de gráfico
- Host
- Protótipo de item
- Filtro de regra LLD
- Protótipo de trigger

Fonte

CDiscoveryRule::get() in *ui/include/classes/api/services/CDiscoveryRule.php*.

discoveryrule.update

Descrição

`object discoveryrule.update(object/array lldRules)`

Este método permite atualizar regras LLD existentes.

> **Note:**
> Este método está disponível apenas para os tipos de usuário *Admin* e *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object/array)` Propriedades da regra LLD a serem atualizadas.

A propriedade `itemid` deve ser definida para cada regra LLD, todas as outras propriedades são opcionais. Somente as propriedades passadas serão atualizadas, todas as outras permanecerão inalteradas.

Além das propriedades padrão da regra LLD, o método aceita os seguintes parâmetros.

Parâmetro	Tipo	Descrição
filter	object	Filtro da regra LLD para substituir o filtro existente.
preprocessing	object/array	Opções de pré-processamento da regra LLD para substituir as opções de pré-processamento existentes. Comportamento do parâmetro: - *somente leitura* para objetos herdados
lld_macro_paths	object/array	Opções de lld_macro_path da regra LLD para substituir as opções de lld_macro_path existentes. Comportamento do parâmetro: - *somente leitura* para objetos herdados

Parâmetro	Tipo	Descrição
overrides	object/array	Opções de substituições da regra LLD para substituir as opções de substituições existentes. Comportamento do parâmetro: - *somente leitura* para objetos herdados

Valores de retorno

`(object)` Retorna um objeto contendo os IDs das regras LLD atualizadas na propriedade `itemids`.

Exemplos

Adicionando um filtro a uma regra de LLD

Adicione um filtro para que o conteúdo da macro *{#FSTYPE}* corresponda à expressão regular *@File systems for discovery*.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.update",
    "params": {
        "itemid": "22450",
        "filter": {
            "evaltype": 1,
            "conditions": [
                {
                    "macro": "{#FSTYPE}",
                    "value": "@File systems for discovery"
                }
            ]
        }
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "22450"
        ]
    },
    "id": 1
}
```

Adicionando caminhos de macro LLD

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.update",
    "params": {
        "itemid": "22450",
        "lld_macro_paths": [
            {
                "lld_macro": "{#MACRO1}",
                "path": "$.json.path"
            }
        ]
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "22450"
        ]
    },
    "id": 1
}
```

Desabilitar trapping

Desabilitar trapping de LLD para a regra de descoberta.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.update",
    "params": {
        "itemid": "28336",
        "allow_traps": 0
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "28336"
        ]
    },
    "id": 1
}
```

Atualizando as opções de pré-processamento da regra LLD

Atualize uma regra LLD com a regra de pré-processamento ”JSONPath”.

Requisição:

```
{
    "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
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
```

```
            "44211"
        ]
    },
    "id": 1
}
```

Atualizando o script da regra de LLD

Atualize um script de regra de LLD com um script diferente e remova parâmetros desnecessários que foram usados pelo script anterior.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.update",
    "params": {
        "itemid": "23865",
        "parameters": [],
        "script": "Zabbix.log(3, 'Log test');\nreturn 1;"
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "23865"
        ]
    },
    "id": 1
}
```

Atualizando o tempo de vida da regra de LLD

Atualize a regra de LLD para desabilitar a entidade não descoberta após 12 horas e excluí-la após 7 dias.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.update",
    "params": {
        "itemid": "46864",
        "lifetime_type": 0,
        "lifetime": "7d",
        "enabled_lifetime_type": 0,
        "enabled_lifetime": "12h"
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "46864"
        ]
    },
    "id": 1
}
```

Fonte

CDiscoveryRule::update() in *ui/include/classes/api/services/CDiscoveryRule.php*.

Report

Esta classe foi projetada para trabalhar com relatórios agendados.

Referências de objetos:

- Report
- Users
- User groups

Métodos disponíveis:

- report.create - criar novos relatórios agendados
- report.delete - excluir relatórios agendados
- report.get - recuperar relatórios agendados
- report.update - atualizar relatórios agendados

Objeto report

Os seguintes objetos estão diretamente relacionados à API `report`.

Relatório

O objeto relatório possui as seguintes propriedades:

Propriedade	Tipo	Descrição
reportid	ID	ID do relatório. Comportamento da propriedade: - *somente leitura* - *obrigatório* para operações de atualização
userid	ID	ID do usuário que criou o relatório. Comportamento da propriedade: - *obrigatório* para operações de criação
name	string	Nome único do relatório. Comportamento da propriedade: - *obrigatório* para operações de criação
dashboardid	ID	ID do dashboard no qual o relatório é baseado. Comportamento da propriedade: - *obrigatório* para operações de criação
period	integer	Período para o qual o relatório será preparado. Valores possíveis: 0 - *(padrão)* dia anterior; 1 - semana anterior; 2 - mês anterior; 3 - ano anterior.
cycle	integer	Agendamento de repetição do período. Valores possíveis: 0 - *(padrão)* diariamente; 1 - semanalmente; 2 - mensalmente; 3 - anualmente.
start_time	integer	Hora do dia, em segundos, em que o relatório será preparado para envio. Padrão: 0.

Propriedade	Tipo	Descrição
weekdays	integer	Dias da semana para envio do relatório. Valores possíveis de bitmap: 1 - Segunda-feira; 2 - Terça-feira; 4 - Quarta-feira; 8 - Quinta-feira; 16 - Sexta-feira; 32 - Sábado; 64 - Domingo. Este é um campo de máscara de bits; qualquer soma dos valores possíveis é aceitável (por exemplo, 21 para segunda, quarta e sexta-feira). Padrão: 0. Comportamento da propriedade: - *obrigatório* se `cycle` for definido como ”weekly”.
active_since	string	Data de início. Valores possíveis: string vazia - *(padrão)* não especificado (armazenado como 0); data específica no formato AAAA-MM-DD (armazenado como timestamp do início do dia (00:00:00)).
active_till	string	Data de término. Valores possíveis: string vazia - *(padrão)* não especificado (armazenado como 0); data específica no formato AAAA-MM-DD (armazenado como timestamp do final do dia (23:59:59)).
subject	string	Assunto da mensagem do relatório.
message	string	Texto da mensagem do relatório.
status	integer	Se o relatório está habilitado ou desabilitado. Valores possíveis: 0 - Desabilitado; 1 - *(padrão)* Habilitado.
description	text	Descrição do relatório.
state	integer	Estado do relatório. Valores possíveis: 0 - *(padrão)* relatório ainda não processado; 1 - relatório foi gerado e enviado com sucesso para todos os destinatários; 2 - falha ao gerar o relatório; ”info” contém informações de erro; 3 - relatório foi gerado, mas o envio para alguns (ou todos) destinatários falhou; ”info” contém informações de erro. Comportamento da propriedade: - *somente leitura*
lastsent	timestamp	Timestamp Unix do último relatório enviado com sucesso. Comportamento da propriedade: - *somente leitura*
info	string	Descrição do erro ou informações adicionais. Comportamento da propriedade: - *somente leitura*

Usuários

O objeto users possui as seguintes propriedades:

Propriedade	Tipo	Descrição
userid	ID	ID do usuário para o qual o relatório será enviado. Comportamento da propriedade: - *obrigatório*
access_userid	ID	ID do usuário em nome do qual o relatório será gerado. 0 - *(padrão)* Gerar relatório pelo destinatário.
exclude	integer	Se deve excluir o usuário da lista de e-mails. Valores possíveis: 0 - *(padrão)* Incluir; 1 - Excluir.

Grupos de usuários

O objeto de grupos de usuários possui as seguintes propriedades:

Propriedade	Tipo	Descrição
usrgrpid	ID	ID do grupo de usuários para o qual o relatório será enviado. Comportamento da propriedade: - *obrigatório*
access_userid	ID	ID do usuário em nome do qual o relatório será gerado. 0 - *(padrão)* Gerar relatório pelo destinatário.

report.create

Descrição

`object report.create(object/array reports)`

Este método permite criar novos relatórios agendados.

Note:
Este método está disponível apenas para os tipos de usuário *Admin* e *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object/array)` Relatórios agendados a serem criados.

Além das propriedades padrão de relatórios agendados, o método aceita os seguintes parâmetros.

Parâmetro	Tipo	Descrição
users	object/array	Usuários para os quais enviar o relatório. Comportamento do parâmetro: - *obrigatório* se `user_groups` não estiver definido
user_groups	object/array	Grupos de usuários para os quais enviar o relatório. Comportamento do parâmetro: - *obrigatório* se `users` não estiver definido

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos relatórios programados criados na propriedade `reportids`. A ordem dos IDs retornados corresponde à ordem dos relatórios programados passados.

Exemplos

Criando um relatório agendado

Crie um relatório semanal que será preparado para a semana anterior toda segunda a sexta-feira às 12:00 de 2021-04-01 a 2021-08-31.

Requisição:

```
{
    "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
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "reportids": [
            "1"
        ]
    },
    "id": 1
}
```

Veja também

- Usuários
- Grupos de usuários

Fonte

CReport::create() in *ui/include/classes/api/services/CReport.php*.

report.delete

Descrição

`object report.delete(array reportids)`

Este método permite excluir relatórios agendados.

> **Note:**
> Este método está disponível apenas para os tipos de usuário *Admin* e *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função de usuário. Consulte Funções de usuário para mais informações.

Parâmetros

`(array)` IDs dos relatórios programados a serem excluídos.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos relatórios programados excluídos na propriedade `reportids`.

Exemplos

Excluindo vários relatórios agendados

Exclua dois relatórios agendados.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "report.delete",
    "params": [
        "1",
        "2"
    ],
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "reportids": [
            "1",
            "2"
        ]
    },
    "id": 1
}
```

Fonte

CReport::delete() in *ui/include/classes/api/services/CReport.php*.

report.get

Descrição

`integer/array report.get(object parameters)`

O método permite recuperar relatórios agendados de acordo com os parâmetros fornecidos.

> **Note:**
> Este método está disponível para usuários de qualquer tipo. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object)` Parâmetros que definem a saída desejada.

O método suporta os seguintes parâmetros.

Parâmetro	Tipo	Descrição
reportids	ID/array	Retorna apenas relatórios agendados com os IDs de relatório fornecidos.
expired	boolean	Se definido como `true`, retorna apenas relatórios agendados expirados.
selectUsers	query	Retorna uma propriedade `users` com os usuários para os quais o relatório está configurado para ser enviado.
selectUserGroups	query	Retorna uma propriedade `user_groups` com os grupos de usuários para os quais o relatório está configurado para ser enviado.
sortfield	string/array	Classifica o resultado pelas propriedades fornecidas. Valores possíveis: `reportid`, `name`, `status`.
countOutput	boolean	Esses parâmetros são descritos no comentário de referência.
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valores de retorno

`(integer/array)` Retorna:

- uma array de objetos;
- a contagem de objetos recuperados, se o parâmetro `countOutput` foi usado.

Exemplos

Recuperando dados do relatório

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "report.get",
    "params": [
        "output": "extend",
        "selectUsers": "extend",
        "selectUserGroups": "extend",
        "reportids": ["1", "2"]
    ],
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "reportid": "1",
            "userid": "1",
            "name": "Weekly report",
            "dashboardid": "1",
            "period": "1",
            "cycle": "1",
```

```
            "start_time": "43200",
            "weekdays": "31",
            "active_since": "2021-04-01",
            "active_till": "2021-08-31",
            "subject": "Weekly report",
            "message": "Report accompanying text",
            "status": "1",
            "description": "Report description",
            "state": "1",
            "lastsent": "1613563219",
            "info": "",
            "users": [
                {
                    "userid": "1",
                    "access_userid": "1",
                    "exclude": "0"
                },
                {
                    "userid": "2",
                    "access_userid": "0",
                    "exclude": "1"
                }
            ],
            "user_groups": [
                {
                    "usrgrpid": "7",
                    "access_userid": "0"
                }
            ]
        },
        {
            "reportid": "2",
            "userid": "1",
            "name": "Monthly report",
            "dashboardid": "2",
            "period": "2",
            "cycle": "2",
            "start_time": "0",
            "weekdays": "0",
            "active_since": "2021-05-01",
            "active_till": "",
            "subject": "Monthly report",
            "message": "Report accompanying text",
            "status": "1",
            "description": "",
            "state": "0",
            "lastsent": "0",
            "info": "",
            "users": [
                {
                    "userid": "1",
                    "access_userid": "1",
                    "exclude": "0"
                }
            ],
            "user_groups": []
        }
    ],
    "id": 1
}
```


Veja também

- Usuários
- Grupos de usuários

Fonte

CReport::get() in *ui/include/classes/api/services/CReport.php*.

report.update

Descrição

`object report.update(object/array reports)`

Este método permite atualizar relatórios agendados existentes.

> **Note:**
> Este método está disponível apenas para os tipos de usuário *Admin* e *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função de usuário. Consulte Funções de usuário para mais informações.

Parâmetros

`(object/array)` Propriedades do relatório agendado a serem atualizadas.

A propriedade `reportid` deve ser definida para cada relatório agendado, todas as outras propriedades são opcionais. Somente as propriedades passadas serão atualizadas, todas as outras permanecerão inalteradas.

Além das propriedades padrão do relatório agendado, o método aceita os seguintes parâmetros.

Parâmetro	Tipo	Descrição
users	object/array	Usuários para substituir os usuários atuais atribuídos ao relatório agendado. Comportamento do parâmetro: - *obrigatório* se `user_groups` não estiver definido
user_groups	object/array	Grupos de usuários para substituir os grupos de usuários atuais atribuídos ao relatório agendado. Comportamento do parâmetro: - *obrigatório* se `users` não estiver definido

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos relatórios programados atualizados na propriedade `reportids`.

Exemplos

Desabilitando relatório agendado

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "report.update",
    "params": {
        "reportid": "1",
        "status": "0"
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "reportids": [
            "1"
```

```
        ]
    },
    "id": 1
}
```

Veja também

- Usuários
- Grupos de usuários

Fonte

CReport::update() in *ui/include/classes/api/services/CReport.php*.

Script

Esta classe foi projetada para trabalhar com scripts.

Referências de objetos:

- Script
  - Parâmetros do webhook
- Debug
  - Entrada de log

Métodos disponíveis:

- script.create - criar novos scripts
- script.delete - excluir scripts
- script.execute - executar scripts
- script.get - recuperar scripts
- script.getscriptsbyevents - recuperar scripts para eventos
- script.getscriptsbyhosts - recuperar scripts para hosts
- script.update - atualizar scripts

Objeto script

Os seguintes objetos estão diretamente relacionados à API `script`.

Script

O objeto script possui as seguintes propriedades.

Propriedade	Tipo	Descrição
scriptid	ID	ID do script. Comportamento da propriedade: - *somente leitura* - *obrigatório* para operações de atualização
name	string	Nome do script. Comportamento da propriedade: - *obrigatório* para operações de criação

Propriedade	Tipo	Descrição
type	integer	Tipo de script. Valores possíveis se `scope` estiver definido como ”operação de ação”: 0 - Script; 1 - IPMI; 2 - SSH; 3 - TELNET; 5 - Webhook. Valores possíveis se `scope` estiver definido como ”ação manual de host” ou ”ação manual de evento”: 6 - URL. Comportamento da propriedade: - *obrigatório* para operações de criação
command	string	Comando a ser executado. Comportamento da propriedade: - *obrigatório* se `type` estiver definido como ”Script”, ”IPMI”, ”SSH”, ”TELNET” ou ”Webhook”
scope	integer	Escopo do script. Valores possíveis: 1 - operação de ação; 2 - ação manual de host; 4 - ação manual de evento. Comportamento da propriedade: - *obrigatório* para operações de criação
execute_on	integer	Onde executar o script. Valores possíveis: 0 - executar no Zabbix agent; 1 - executar no Zabbix server. É *suportado* apenas se a execução de scripts globais estiver habilitada no Zabbix server; 2 - *(padrão)* executar no Zabbix server ou proxy. Comportamento da propriedade: - *suportado* se `type` estiver definido como ”Script”
menu_path	string	Pastas separadas por barra que formam uma navegação tipo menu no frontend ao clicar no host ou evento. Comportamento da propriedade: - *suportado* se `scope` estiver definido como ”ação manual de host” ou ”ação manual de evento”
authtype	integer	Método de autenticação usado para o tipo de script SSH. Valores possíveis: 0 - senha; 1 - chave pública. Comportamento da propriedade: - *suportado* se `type` estiver definido como ”SSH”
username	string	Nome de usuário usado para autenticação. Comportamento da propriedade: - *obrigatório* se `type` estiver definido como ”SSH” ou ”TELNET”

Propriedade	Tipo	Descrição
password	string	Senha usada para scripts SSH com autenticação por senha e scripts TELNET. Comportamento da propriedade: - *suportado* se `type` estiver definido como ”SSH” e `authtype` estiver definido como ”senha”, ou `type` estiver definido como ”TELNET”
publickey	string	Nome do arquivo de chave pública usado para scripts SSH com autenticação por chave pública. Comportamento da propriedade: - *obrigatório* se `type` estiver definido como ”SSH” e `authtype` estiver definido como ”chave pública”
privatekey	string	Nome do arquivo de chave privada usado para scripts SSH com autenticação por chave pública. Comportamento da propriedade: - *obrigatório* se `type` estiver definido como ”SSH” e `authtype` estiver definido como ”chave pública”
port	string	Número da porta usada para scripts SSH e TELNET. Comportamento da propriedade: - *suportado* se `type` estiver definido como ”SSH” ou ”TELNET”
groupid	ID	ID do grupo de hosts no qual o script pode ser executado. Se definido como ”0”, o script estará disponível em todos os grupos de hosts. Padrão: 0.
usrgrpid	ID	ID do grupo de usuários que terá permissão para executar o script. Se definido como ”0”, o script estará disponível para todos os grupos de usuários. Padrão: 0.
host_access	integer	Permissões de host necessárias para executar o script. Valores possíveis: 2 - *(padrão)* leitura; 3 - gravação. Comportamento da propriedade: - *suportado* se `scope` estiver definido como ”ação manual de host” ou ”ação manual de evento”
confirmation	string	Texto do pop-up de confirmação. O pop-up aparecerá ao tentar executar o script a partir do frontend do Zabbix. Comportamento da propriedade: - *suportado* se `scope` estiver definido como ”ação manual de host” ou ”ação manual de evento”

Propriedade	Tipo	Descrição
timeout	string	Tempo limite de execução do script webhook em segundos. Sufixos de tempo são suportados (por exemplo, 30s, 1m). Valores possíveis: 1-60s. Padrão: 30s. Comportamento da propriedade: - *obrigatório* se `type` estiver definido como ”Webhook”
parameters	array	Array de parâmetros de entrada do webhook. Comportamento da propriedade: - *suportado* se `type` estiver definido como ”Webhook”
description	string	Descrição do script.
url	string	URL definida pelo usuário. Comportamento da propriedade: - *obrigatório* se `type` estiver definido como ”URL”
new_window	integer	Abrir URL em uma nova janela. Valores possíveis: 0 - Não; 1 - *(padrão)* Sim. Comportamento da propriedade: - *suportado* se `type` estiver definido como ”URL”
manualinput	integer	Indica se o script aceita entrada fornecida pelo usuário. Valores possíveis: 0 - *(padrão)* Desabilitado; 1 - Habilitado; Comportamento da propriedade: - *suportado* se `scope` estiver definido como ”ação manual de host” ou ”ação manual de evento”
manualinput_prompt	string	Texto do prompt de entrada manual. Comportamento da propriedade: - *obrigatório* se `manualinput` estiver definido como ”Habilitado”
manualinput_validator	string	Um campo de string de caracteres usado para validar a entrada fornecida pelo usuário. A string consiste em uma expressão regular ou um conjunto de valores separados por vírgulas. Comportamento da propriedade: - *obrigatório* se `manualinput` estiver definido como ”Habilitado”
manualinput_validator_type	integer	Determina o tipo de entrada do usuário esperada. Valores possíveis: 0 - *(padrão)* String. Indica que manualinput_validator deve ser tratado como uma expressão regular; 1 - Lista. Indica que manualinput_validator deve ser tratado como uma lista de valores possíveis separados por vírgulas. Comportamento da propriedade: - *suportado* se `manualinput` estiver definido como ”Habilitado”
manualinput_default_value	string	Valor padrão para preenchimento automático da entrada do usuário. Comportamento da propriedade: - *suportado* se `manualinput_validator_type` estiver definido como ”String”

Parâmetros do webhook

Os parâmetros passados para o script do webhook quando ele é chamado possuem as seguintes propriedades.

Propriedade	Tipo	Descrição
name	string	Nome do parâmetro. Comportamento da propriedade: - *obrigatório*
value	string	Valor do parâmetro. Suporta macros.

Depuração

Informações de depuração do script webhook executado. O objeto de depuração possui as seguintes propriedades.

Propriedade	Tipo	Descrição
logs	array	Array de entradas de log.
ms	string	Duração da execução do script em milissegundos.

Entrada de log

O objeto de entrada de log possui as seguintes propriedades.

Propriedade	Tipo	Descrição
level	integer	Nível de log.
ms	string	O tempo decorrido em milissegundos desde que o script foi executado antes da entrada de log ser adicionada.
message	string	Mensagem de log.

script.create

Descrição

`object script.create(object/array scripts)`

Este método permite criar novos scripts.

Note:
Este método está disponível apenas para o tipo de usuário *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(objeto/array)` Scripts para criar.

O método aceita scripts com as propriedades de script padrão.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos scripts criados na propriedade `scriptids`. A ordem dos IDs retornados corresponde à ordem dos scripts passados.

Exemplos

Crie um script de webhook

Crie um script de webhook que envie uma solicitação HTTP para um serviço externo.

Request:

```
{
    "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
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "scriptids": [
            "3"
        ]
    },
    "id": 1
}
```

Criar um script SSH

Crie um script SSH com autenticação por chave pública que pode ser executado em um host e possui um menu de contexto.

Requisição:

```
{
    "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
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
```

```
        "scriptids": [
            "5"
        ]
    },
    "id": 1
}
```

Criar um script personalizado

Crie um script personalizado que irá reiniciar um server. O script irá requerer acesso de escrita ao host e irá solicitar ao usuário uma entrada manual. Após o envio bem-sucedido da entrada, o script exibirá uma mensagem de confirmação no frontend.

Requisição:

```
{
    "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
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "scriptids": [
            "4"
        ]
    },
    "id": 1
}
```

Criar um script do tipo URL

Crie um script do tipo URL para escopo de host que permanece na mesma janela e possui texto de confirmação.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "script.create",
    "params": {
        "name": "URL script",
        "type": 6,
        "scope": 2,
        "url": "http://zabbix/ui/zabbix.php?action=host.edit&hostid={HOST.ID}",
        "confirmation": "Edit host {HOST.NAME}?",
        "new_window": 0
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
```

```
    "result": {
        "scriptids": [
            "56"
        ]
    },
    "id": 1
}
```

Crie um script do tipo URL com entrada manual

Crie um script do tipo URL para escopo de evento que abre em uma nova janela e possui entrada manual.

Requisição:

```
{
    "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
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "scriptids": [
            "57"
        ]
    },
    "id": 1
}
```

Fonte

CScript::create() in *ui/include/classes/api/services/CScript.php*.

script.delete

Descrição

`object script.delete(array scriptIds)`

Este método permite excluir scripts.

> **Note:**
> Este método está disponível apenas para o tipo de usuário *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(array)` IDs dos scripts a serem excluídos.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos scripts excluídos na propriedade `scriptids`.

Exemplos

Excluir vários scripts

Exclua dois scripts.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "script.delete",
    "params": [
        "3",
        "4"
    ],
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "scriptids": [
            "3",
            "4"
        ]
    },
    "id": 1
}
```

Fonte

CScript::delete() in *ui/include/classes/api/services/CScript.php*.

script.execute

Descrição

`object script.execute(object parameters)`

Este método permite executar um script em um host ou evento. Exceto para scripts do tipo URL. Estes não são executáveis.

> **Note:**
> Este método está disponível para usuários de qualquer tipo. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object)` Parâmetros contendo o ID do script a ser executado, o ID do host ou o ID do evento e o valor de manualinput.

Parâmetro	Tipo	Descrição
scriptid	ID	ID do script a ser executado. Comportamento do parâmetro: - *obrigatório*
hostid	ID	ID do host no qual executar o script. Comportamento do parâmetro: - *obrigatório* se `eventid` não estiver definido
eventid	ID	ID do evento no qual executar o script. Comportamento do parâmetro: - *obrigatório* se `hostid` não estiver definido
manualinput	string	Valor fornecido pelo usuário para executar o script, substituindo a macro {MANUALINPUT}.

Valores de retorno

(object) Retorna o resultado da execução do script.

Propriedade	Tipo	Descrição
response	string	Se o script foi executado com sucesso. Valor possível - success.
value	string	Saída do script.
debug	object	Contém um objeto debug se um script webhook for executado. Para outros tipos de script, contém um objeto vazio.

Exemplos

Execute um script de webhook

Execute um script de webhook que envia uma solicitação HTTP para um serviço externo.

Solicitação:

```
{
    "jsonrpc": "2.0",
    "method": "script.execute",
    "params": {
        "scriptid": "4",
        "hostid": "30079"
    },
    "id": 1
}
```

Resposta:

```
{
    "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
}
```

Execute um script personalizado

Execute um script ”ping” em um host.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "script.execute",
    "params": {
        "scriptid": "1",
        "hostid": "30079"
    },
    "id": 1
}
```

Resposta:

```
{
    "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
}
```


Executar um script personalizado com entrada manual

Executar um script "ping" com o comando "ping -c {MANUALINPUT} {HOST.CONN}; case $? in [01]) true;; *) false;; esac" em um host.

Requisição:


```
{
    "jsonrpc": "2.0",
    "method": "script.execute",
    "params": {
        "scriptid": "7",
        "hostid": "30079",
        "manualinput": "2"
    },
    "id": 1
}
```


Resposta:


```
{
    "jsonrpc": "2.0",
    "result": {
        "response": "success",
        "value": "PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.\n64 bytes from 127.0.0.1: icmp_seq=1 tt
        "debug": []
    },
    "id": 1
}
```


Fonte

CScript::execute() in *ui/include/classes/api/services/CScript.php*.

script.get

Descrição

`integer/array script.get(object parameters)`

O método permite recuperar scripts de acordo com os parâmetros fornecidos.

> **Note:**
> Este método está disponível para usuários de qualquer tipo. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object)` Parâmetros que definem a saída desejada.

O método suporta os seguintes parâmetros.

Parâmetro	Tipo	Descrição
groupids	ID/array	Retornar apenas scripts que podem ser executados nos grupos de hosts fornecidos.
hostids	ID/array	Retornar apenas scripts que podem ser executados nos hosts fornecidos.

Parâmetro	Tipo	Descrição
scriptids	ID/array	Retornar apenas scripts com os IDs fornecidos.
usrgrpids	ID/array	Retornar apenas scripts que podem ser executados por usuários nos grupos de usuários fornecidos.
selectHostGroups	query	Retornar uma propriedade `hostgroups` com os grupos de hosts nos quais o script pode ser executado.
selectHosts	query	Retornar uma propriedade `hosts` com os hosts nos quais o script pode ser executado.
selectActions	query	Retornar uma propriedade `actions` com as ações associadas ao script.
sortfield	string/array	Classificar o resultado pelas propriedades fornecidas. Valores possíveis: `scriptid`, `name`.
countOutput	boolean	Esses parâmetros são descritos no comentário de referência.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	
selectGroups (obsoleto)	query	Este parâmetro está obsoleto, por favor use `selectHostGroups` em seu lugar. Retornar uma propriedade `groups` com os grupos de hosts nos quais o script pode ser executado.

Valores de retorno

`(integer/array)` Retorna:

- uma array de objetos;
- a contagem de objetos recuperados, se o parâmetro `countOutput` foi usado.

Exemplos

Recuperar todos os scripts

Recupera todos os scripts configurados.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "script.get",
    "params": {
        "output": "extend"
    },
    "id": 1
}
```

Resposta:

```
{
    "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
}
```


Veja também

- Host

- Grupo de hosts

Fonte

CScript::get() in *ui/include/classes/api/services/CScript.php*.

script.getscriptsbyevents

Descrição

`object script.getscriptsbyevents(object parameters)`

Este método permite recuperar todos os scripts disponíveis no evento fornecido ou um script específico se o ID do script for fornecido. Quando manualinput é fornecido, ele substitui a macro {MANUALINPUT} pelo valor especificado.

> **Note:**
> Este método está disponível para usuários de qualquer tipo. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object/array)` O método aceita um objeto ou um array de objetos com os seguintes parâmetros.

Parâmetro	Tipo	Descrição
eventid	ID	ID do evento para retornar scripts. Deve ser único. Comportamento do parâmetro: - *obrigatório*
scriptid	ID	ID do script a ser retornado.
manualinput	string	Valor fornecido pelo usuário para o macro {MANUALINPUT}.

Valores de retorno

`(object)` Retorna um objeto com os IDs dos eventos como propriedades e arrays de scripts disponíveis como valores. Se o ID do script for fornecido, o valor associado será um array contendo o script específico.

> **Note:**
> O método irá expandir automaticamente as macros no texto de `confirmation`, no texto de `manualinput prompt` e na `url`.
> Se o parâmetro manualinput for fornecido, a macro {MANUALINPUT} será resolvida para o valor especificado.

Exemplos

Recuperar scripts por IDs de eventos

Recupere todos os scripts disponíveis para os eventos "632" e "614".

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "script.getscriptsbyevents",
    "params": [
      {
         "eventid":  "632"
      },
      {
         "eventid":  "614"
      }
    ],
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "632": [
            {
                "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": "4",
                "port": "",
                "authtype": "0",
                "username": "",
                "password": "",
                "publickey": "",
                "privatekey": "",
                "menu_path": "",
                "url": "",
                "new_window": "1",
                "manualinput": "0",
                "manualinput_prompt": "",
                "manualinput_validator_type": "0",
                "manualinput_validator": "",
                "manualinput_default_value": "",
                "parameters": []
            },
            {
                "scriptid": "1",
                "name": "Ping",
                "command": "/bin/ping -c 3 {HOST.CONN} 2>&1",
                "host_access": "2",
                "usrgrpid": "0",
                "groupid": "0",
                "description": "",
                "confirmation": "",
                "type": "0",
                "execute_on": "1",
                "timeout": "30s",
                "scope": "4",
                "port": "",
                "authtype": "0",
                "username": "",
                "password": "",
                "publickey": "",
                "privatekey": "",
                "menu_path": "",
                "url": "",
                "new_window": "1",
                "manualinput": "0",
                "manualinput_prompt": "",
                "manualinput_validator_type": "0",
                "manualinput_validator": "",
                "manualinput_default_value": "",
                "parameters": []
            },
```

```
        {
            "scriptid": "4",
            "name": "Open Zabbix page",
            "command": "",
            "host_access": "2",
            "usrgrpid": "0",
            "groupid": "0",
            "description": "",
            "confirmation": "Are you sure you want to open page *UNKNOWN*?",
            "type": "6",
            "execute_on": "2",
            "timeout": "30s",
            "scope": "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": "Zabbix page to open:",
            "manualinput_validator_type": "1",
            "manualinput_validator": "dashboard.view,discovery.view",
            "manualinput_default_value": "",
            "parameters": []
        },
        {
            "scriptid": "2",
            "name": "Traceroute",
            "command": "/usr/bin/traceroute {HOST.CONN} 2>&1",
            "host_access": "2",
            "usrgrpid": "0",
            "groupid": "0",
            "description": "",
            "confirmation": "",
            "type": "0",
            "execute_on": "1",
            "timeout": "30s",
            "scope": "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": "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": "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": "Open Zabbix page",
            "command": "",
            "host_access": "2",
            "usrgrpid": "0",
```

```
                "groupid": "0",
                "description": "",
                "confirmation": "Are you sure you want to open page *UNKNOWN*?",
                "type": "6",
                "execute_on": "2",
                "timeout": "30s",
                "scope": "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": "Zabbix page to open:",
                "manualinput_validator_type": "1",
                "manualinput_validator": "dashboard.view,discovery.view",
                "manualinput_default_value": "",
                "parameters": []
            },
            {
                "scriptid": "2",
                "name": "Traceroute",
                "command": "/usr/bin/traceroute {HOST.CONN} 2>&1",
                "host_access": "2",
                "usrgrpid": "0",
                "groupid": "0",
                "description": "",
                "confirmation": "",
                "type": "0",
                "execute_on": "1",
                "timeout": "30s",
                "scope": "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
}
```


Recuperar script específico com valor manualinput.

Recuperar script com ID ”4” no evento ”632” com valor manualinput ”dashboard.view”.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "script.getscriptsbyevents",
    "params": [
      {
        "eventid":  "632",
        "scriptid": "4",
        "manualinput": "dashboard.view"
      }
  ],
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
      "632": [
        {
          "scriptid": "4",
          "name": "Open Zabbix page",
          "command": "",
          "host_access": "2",
          "usrgrpid": "0",
          "groupid": "0",
          "description": "",
          "confirmation": "Are you sure you want to open page dashboard.view?",
          "type": "6",
          "execute_on": "2",
          "timeout": "30s",
          "scope": "4",
          "port": "",
          "authtype": "0",
          "username": "",
          "password": "",
          "publickey": "",
          "privatekey": "",
          "menu_path": "",
          "url": "http://localhost/ui/zabbix.php?action=dashboard.view",
          "new_window": "1",
          "manualinput": "1",
          "manualinput_prompt": "Zabbix page to open:",
          "manualinput_validator_type": "1",
          "manualinput_validator": "dashboard.view,discovery.view",
          "manualinput_default_value": "",
          "parameters": []
        }
      ]
    },
    "id": 1
}
```

Fonte

CScript::getScriptsByEvents() em *ui/include/classes/api/services/CScript.php*.

script.getscriptsbyhosts

Descrição

`object script.getscriptsbyhosts(object parameters)`

Este método permite recuperar todos os scripts disponíveis no host fornecido ou um script específico se o ID do script for fornecido. Quando manualinput é fornecido, ele substitui a macro {MANUALINPUT} pelo valor especificado.

Note:
Este método está disponível para usuários de qualquer tipo. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object/array)` O método aceita um objeto ou array de objetos com os seguintes parâmetros.

Parâmetro	Tipo	Descrição
hostid	ID	ID do host para retornar scripts. Deve ser único. Comportamento do parâmetro: - *obrigatório*
scriptid	ID	ID do script a ser retornado.
manualinput	string	Valor do macro {MANUALINPUT} fornecido pelo usuário.

Valores de retorno

`(object)` Retorna um objeto com os IDs dos hosts como propriedades e arrays de scripts disponíveis como valores. Se o ID do script for fornecido, o valor associado será um array contendo o script específico.

Note:
O método irá expandir automaticamente as macros no texto de `confirmation`, no texto de `manualinput prompt` e na `url`.
Se o parâmetro manualinput for fornecido, a macro {MANUALINPUT} será resolvida para o valor especificado.

Exemplos

Recuperar scripts por IDs de host

Recupere todos os scripts disponíveis nos hosts ”30079” e ”30073”.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "script.getscriptsbyhosts",
    "params": [
       {
          "hostid":  "30079"
       },
       {
          "hostid":  "30073"
       }
    ],
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "30079": [
            {
                "scriptid": "3",
                "name": "Detect operating system",
                "command": "sudo /usr/bin/nmap -O {HOST.CONN} 2>&1",
                "host_access": "2",
                "usrgrpid": "7",
                "groupid": "0",
                "description": "",
                "confirmation": "",
```

```
            "type": "0",
            "execute_on": "1",
            "timeout": "30s",
            "scope": "2",
            "port": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "menu_path": "",
            "url": "",
            "new_window": "1",
            "manualinput": "0",
            "manualinput_prompt": "",
            "manualinput_validator_type": "0",
            "manualinput_validator": "",
            "manualinput_default_value": "",
            "parameters": []
        },
        {
            "scriptid": "1",
            "name": "Ping",
            "command": "/bin/ping -c 3 {HOST.CONN} 2>&1",
            "host_access": "2",
            "usrgrpid": "0",
            "groupid": "0",
            "description": "",
            "confirmation": "",
            "type": "0",
            "execute_on": "1",
            "timeout": "30s",
            "scope": "2",
            "port": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "menu_path": "",
            "url": "",
            "new_window": "1",
            "manualinput": "0",
            "manualinput_prompt": "",
            "manualinput_validator_type": "0",
            "manualinput_validator": "",
            "manualinput_default_value": "",
            "parameters": []
        },
        {
            "scriptid": "4",
            "name": "Open Zabbix page",
            "command": "",
            "host_access": "2",
            "usrgrpid": "0",
            "groupid": "0",
            "description": "",
            "confirmation": "Are you sure you want to open page *UNKNOWN*?",
            "type": "6",
            "execute_on": "2",
            "timeout": "30s",
            "scope": "2",
```

```
            "port": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "menu_path": "",
            "url": "http://localhost/ui/zabbix.php?action=*UNKNOWN*",
            "new_window": "1",
            "manualinput": "0",
            "manualinput_prompt": "Zabbix page to open:",
            "manualinput_validator_type": "0",
            "manualinput_validator": "dashboard.view,discovery.view",
            "manualinput_default_value": "",
            "parameters": []
        },
        {
            "scriptid": "2",
            "name": "Traceroute",
            "command": "/usr/bin/traceroute {HOST.CONN} 2>&1",
            "host_access": "2",
            "usrgrpid": "0",
            "groupid": "0",
            "description": "",
            "confirmation": "",
            "type": "0",
            "execute_on": "1",
            "timeout": "30s",
            "scope": "2",
            "port": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "menu_path": "",
            "url": "",
            "new_window": "1",
            "manualinput": "0",
            "manualinput_prompt": "",
            "manualinput_validator_type": "0",
            "manualinput_validator": "",
            "manualinput_default_value": "",
            "parameters": []
        }
    ],
    "30073": [
        {
            "scriptid": "3",
            "name": "Detect operating system",
            "command": "sudo /usr/bin/nmap -O {HOST.CONN} 2>&1",
            "host_access": "2",
            "usrgrpid": "7",
            "groupid": "0",
            "description": "",
            "confirmation": "",
            "type": "0",
            "execute_on": "1",
            "timeout": "30s",
            "scope": "2",
            "port": "",
            "authtype": "0",
```

```
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "menu_path": "",
            "url": "",
            "new_window": "1",
            "manualinput": "0",
            "manualinput_prompt": "",
            "manualinput_validator_type": "0",
            "manualinput_validator": "",
            "manualinput_default_value": "",
            "parameters": []
        },
        {
            "scriptid": "1",
            "name": "Ping",
            "command": "/bin/ping -c 3 {HOST.CONN} 2>&1",
            "host_access": "2",
            "usrgrpid": "0",
            "groupid": "0",
            "description": "",
            "confirmation": "",
            "type": "0",
            "execute_on": "1",
            "timeout": "30s",
            "scope": "2",
            "port": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "menu_path": "",
            "url": "",
            "new_window": "1",
            "manualinput": "0",
            "manualinput_prompt": "",
            "manualinput_validator_type": "0",
            "manualinput_validator": "",
            "manualinput_default_value": "",
            "parameters": []
        },
        {
            "scriptid": "4",
            "name": "Open Zabbix page",
            "command": "",
            "host_access": "2",
            "usrgrpid": "0",
            "groupid": "0",
            "description": "",
            "confirmation": "Are you sure you want to open page *UNKNOWN*?",
            "type": "6",
            "execute_on": "2",
            "timeout": "30s",
            "scope": "2",
            "port": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
```

```
                "menu_path": "",
                "url": "http://localhost/ui/zabbix.php?action=*UNKNOWN*",
                "new_window": "1",
                "manualinput": "1",
                "manualinput_prompt": "Zabbix page to open:",
                "manualinput_validator_type": "1",
                "manualinput_validator": "dashboard.view,discovery.view",
                "manualinput_default_value": "",
                "parameters": []
            },
            {
                "scriptid": "2",
                "name": "Traceroute",
                "command": "/usr/bin/traceroute {HOST.CONN} 2>&1",
                "host_access": "2",
                "usrgrpid": "0",
                "groupid": "0",
                "description": "",
                "confirmation": "",
                "type": "0",
                "execute_on": "1",
                "timeout": "30s",
                "scope": "2",
                "port": "",
                "authtype": "0",
                "username": "",
                "password": "",
                "publickey": "",
                "privatekey": "",
                "menu_path": "",
                "url": "",
                "new_window": "1",
                "manualinput": "0",
                "manualinput_prompt": "",
                "manualinput_validator_type": "0",
                "manualinput_validator": "",
                "manualinput_default_value": "",
                "parameters": []
            }
        ]
    },
    "id": 1
}
```


Recuperar script específico com valor manualinput.

Recuperar script com ID ”4” no host ”30079” com valor manualinput ”dashboard.view”.

Requisição:


```
{
    "jsonrpc": "2.0",
    "method": "script.getscriptsbyhosts",
    "params": [
      {
        "hostid":  "30079",
        "scriptid": "4",
        "manualinput": "dashboard.view"
      }
  ],
    "id": 1
}
```


Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
      "30079": [
        {
          "scriptid": "4",
          "name": "Open Zabbix page",
          "command": "",
          "host_access": "2",
          "usrgrpid": "0",
          "groupid": "0",
          "description": "",
          "confirmation": "Are you sure you want to open page dashboard.view?",
          "type": "6",
          "execute_on": "2",
          "timeout": "30s",
          "scope": "2",
          "port": "",
          "authtype": "0",
          "username": "",
          "password": "",
          "publickey": "",
          "privatekey": "",
          "menu_path": "",
          "url": "http://localhost/ui/zabbix.php?action=dashboard.view",
          "new_window": "1",
          "manualinput": "1",
          "manualinput_prompt": "Zabbix page to open:",
          "manualinput_validator_type": "1",
          "manualinput_validator": "dashboard.view,discovery.view",
          "manualinput_default_value": "",
          "parameters": []
        }
      ]
    },
    "id": 1
}
```


Fonte

CScript::getScriptsByHosts() in *ui/include/classes/api/services/CScript.php*.

script.update

Descrição

`object script.update(object/array scripts)`

Este método permite atualizar scripts existentes.

> **Note:**
> Este método está disponível apenas para o tipo de usuário *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object/array)` Propriedades do script a ser atualizado.

A propriedade `scriptid` deve ser definida para cada script, todas as outras propriedades são opcionais. Apenas as propriedades passadas serão atualizadas, todas as outras permanecerão inalteradas. Uma exceção é a mudança da propriedade `type` de 5 (Webhook) para outra: a propriedade `parameters` será limpa.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos scripts atualizados na propriedade `scriptids`.

Exemplos

Alterar comando do script

Altere o comando do script para "/bin/ping -c 10 {HOST.CONN} 2>&1".

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "script.update",
    "params": {
        "scriptid": "1",
        "command": "/bin/ping -c 10 {HOST.CONN} 2>&1"
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "scriptids": [
            "1"
        ]
    },
    "id": 1
}
```

Alterar o comando do script e adicionar entrada manual

Altere o comando do script para "/bin/ping -c {MANUALINPUT} {HOST.CONN} 2>&1".

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "script.update",
    "params": {
      "scriptid": "1",
      "command": "/bin/ping -c {MANUALINPUT} {HOST.CONN} 2>&1",
      "manualinput": "1",
      "manualinput_prompt": "Specify the number of ICMP packets to send with the ping command",
      "manualinput_validator": "^(?:[1-9]|10)$",
      "manualinput_validator_type": "0",
      "manualinput_default_value": "10"
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "scriptids": [
            "1"
        ]
    },
    "id": 1
}
```

Fonte

CScript::update() in *ui/include/classes/api/services/CScript.php*.

Service

Esta classe foi projetada para trabalhar com serviços de infraestrutura de TI/negócios.

Referências de objeto:

- Service
- Status rule
- Service tag
- Service alarm
- Problem tag

Métodos disponíveis:

- service.create - criar novos serviços
- service.delete - excluir serviços
- service.get - recuperar serviços
- service.update - atualizar serviços

Objeto service

Os seguintes objetos estão diretamente relacionados à API `service`.

Serviço

O objeto serviço possui as seguintes propriedades.

Propriedade	Tipo	Descrição
serviceid	ID	ID do serviço. Comportamento da propriedade: - *somente leitura* - *obrigatório* para operações de atualização
algorithm	integer	Regra de cálculo de status. Aplicável apenas se existirem serviços filhos. Valores possíveis: 0 - definir status como OK; 1 - mais crítico se todos os filhos tiverem problemas; 2 - mais crítico dos serviços filhos.
name	string	Comportamento da propriedade: - *obrigatório* para operações de criação Nome do serviço.
sortorder	integer	Comportamento da propriedade: - *obrigatório* para operações de criação Posição do serviço usada para ordenação. Valores possíveis: 0-999.
weight	integer	Comportamento da propriedade: - *obrigatório* para operações de criação Peso do serviço. Valores possíveis: 0-1000000. Padrão: 0.

Propriedade	Tipo	Descrição
propagation_rule	integer	Regra de propagação de status. Valores possíveis: 0 - *(padrão)* propagar o status do serviço como está - sem alterações; 1 - aumentar o status propagado por um determinado `propagation_value` (de 1 a 5 severidades); 2 - diminuir o status propagado por um determinado `propagation_value` (de 1 a 5 severidades); 3 - ignorar este serviço - o status não é propagado para o serviço pai; 4 - definir status fixo do serviço usando um determinado `propagation_value`. Comportamento da propriedade: - *obrigatório* se `propagation_value` estiver definido
propagation_value	integer	Valor de propagação de status. Valores possíveis se `propagation_rule` estiver definido como "0" ou "3": 0 - Não classificado. Valores possíveis se `propagation_rule` estiver definido como "1" ou "2": 1 - Informação; 2 - Aviso; 3 - Médio; 4 - Alto; 5 - Desastre. Valores possíveis se `propagation_rule` estiver definido como "4": -1 - OK; 0 - Não classificado; 1 - Informação; 2 - Aviso; 3 - Médio; 4 - Alto; 5 - Desastre. Comportamento da propriedade: - *obrigatório* se `propagation_rule` estiver definido
status	integer	Se o serviço está em estado OK ou de problema. Se o serviço estiver em estado de problema, `status` é igual a: - a severidade do problema mais crítico; - o status mais alto de um serviço filho em estado de problema. Se o serviço estiver em estado OK, `status` é igual a: -1. Comportamento da propriedade: - *somente leitura*
description	string	Descrição do serviço.
uuid	string	Identificador único universal, usado para vincular serviços importados aos já existentes. Gerado automaticamente, se não fornecido.
created_at	integer	Timestamp Unix quando o serviço foi criado.
readonly	integer	Acesso ao serviço. Valores possíveis: 0 - Leitura e escrita; 1 - Somente leitura. Comportamento da propriedade: - *somente leitura*

Regra de status

O objeto de regra de status possui as seguintes propriedades.

Propriedade	Tipo	Descrição
type	integer	Condição para definir o status (Novo status). Valores possíveis: 0 - se pelo menos (N) serviços filhos tiverem status (Status) ou superior; 1 - se pelo menos (N%) dos serviços filhos tiverem status (Status) ou superior; 2 - se menos de (N) serviços filhos tiverem status (Status) ou inferior; 3 - se menos de (N%) dos serviços filhos tiverem status (Status) ou inferior; 4 - se o peso dos serviços filhos com status (Status) ou superior for pelo menos (W); 5 - se o peso dos serviços filhos com status (Status) ou superior for pelo menos (N%); 6 - se o peso dos serviços filhos com status (Status) ou inferior for menor que (W); 7 - se o peso dos serviços filhos com status (Status) ou inferior for menor que (N%). Onde: - N (W) é `limit_value`; - (Status) é `limit_status`; - (Novo status) é `new_status`. Comportamento da propriedade: - *obrigatório*
limit_value	integer	Valor limite. Valores possíveis: - para N e W: 1-100000; - para N%: 1-100. Comportamento da propriedade: - *obrigatório*
limit_status	integer	Status limite. Valores possíveis: -1 - OK; 0 - Não classificado; 1 - Informação; 2 - Aviso; 3 - Médio; 4 - Alto; 5 - Desastre. Comportamento da propriedade: - *obrigatório*

Propriedade	Tipo	Descrição
new_status	integer	Novo valor de status. Valores possíveis: 0 - Não classificado; 1 - Informação; 2 - Aviso; 3 - Médio; 4 - Alto; 5 - Desastre. Comportamento da propriedade: - *obrigatório*

Tag de serviço

O objeto de tag de serviço possui as seguintes propriedades.

Propriedade	Tipo	Descrição
tag	string	Nome da tag de serviço. Comportamento da propriedade: - *obrigatório*
value	string	Valor da tag de serviço.

Alarme de serviço

Note:
Alarmes de serviço não podem ser criados, atualizados ou excluídos diretamente via API do Zabbix.

Os objetos de alarme de serviço representam uma mudança de estado do serviço. Eles possuem as seguintes propriedades.

Propriedade	Tipo	Descrição
clock	timestamp	Momento em que a mudança de estado do serviço ocorreu.
value	integer	Status do serviço. Consulte a propriedade `status` do serviço para uma lista de valores possíveis.

Tag de problema

As tags de problema permitem vincular serviços a eventos de problema. O objeto de tag de problema possui as seguintes propriedades.

Propriedade	Tipo	Descrição
tag	string	Nome da tag de problema. Comportamento da propriedade: - *obrigatório*
operator	integer	Operador da condição de mapeamento. Valores possíveis: 0 - *(padrão)* Igual; 2 - Contém.
value	string	Valor da tag de problema.

service.create

Descrição

`object service.create(object/array services)`

Este método permite criar novos serviços.

> **Note:**
> Este método está disponível para usuários de qualquer tipo. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object/array)` serviços a serem criados.

Além das propriedades padrão do serviço, o método aceita os seguintes parâmetros.

Parâmetro	Tipo	Descrição
children	array	Serviços filhos a serem vinculados ao serviço. Os serviços filhos devem ter apenas a propriedade `serviceid` definida.
parents	array	Serviços pais a serem vinculados ao serviço. Os serviços pais devem ter apenas a propriedade `serviceid` definida.
tags	array	Tags de serviço a serem criadas para o serviço.
problem_tags	array	Tags de problema a serem criadas para o serviço.
status_rules	array	Regras de status a serem criadas para o serviço.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos serviços criados na propriedade `serviceids`. A ordem dos IDs retornados corresponde à ordem dos serviços passados

Exemplos

Criando um serviço

Crie um serviço que será alternado para o estado de problema, se pelo menos um filho tiver um problema.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "service.create",
    "params": {
        "name": "Server 1",
        "algorithm": 1,
        "sortorder": 1
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "serviceids": [
            "5"
        ]
    },
    "id": 1
}
```

Fonte

CService::create() in *ui/include/classes/api/services/CService.php*.

service.delete

Descrição

`object service.delete(array serviceIds)`

Este método permite excluir serviços.

> **Note:**
> Este método está disponível para usuários de qualquer tipo. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(array)` IDs dos serviços a serem excluídos.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos serviços excluídos na propriedade `serviceids`.

Exemplos

Excluindo vários serviços

Exclua dois serviços.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "service.delete",
    "params": [
        "4",
        "5"
    ],
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "serviceids": [
            "4",
            "5"
        ]
    },
    "id": 1
}
```

Fonte

CService::delete() in *ui/include/classes/api/services/CService.php*.

service.get

Descrição

`integer/array service.get(object parameters)`

O método permite recuperar serviços de acordo com os parâmetros fornecidos.

Note:
Este método está disponível para usuários de qualquer tipo. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object)` Parâmetros que definem a saída desejada.

O método suporta os seguintes parâmetros.

Parâmetro	Tipo	Descrição
serviceids	ID/array	Retorna apenas os serviços com os IDs fornecidos.
parentids	ID/array	Retorna apenas os serviços que estão vinculados aos serviços pai fornecidos.
deep_parentids	flag	Retorna todos os serviços filhos diretos e indiretos. Usado junto com `parentids`.
childids	ID/array	Retorna apenas os serviços que estão vinculados aos serviços filhos fornecidos.
evaltype	integer	Método de avaliação da tag. Valores possíveis: 0 - *(padrão)* E/OU; 2 - OU.
tags	object/array	Retorna apenas os serviços com as tags fornecidas. Formato: `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`. Um array vazio retorna todos os serviços. Valores possíveis para o operador: 0 - *(padrão)* Contém; 1 - Igual; 2 - Não contém; 3 - Não é igual; 4 - Existe; 5 - Não existe.
problem_tags	object/array	Retorna apenas os serviços com as tags de problema fornecidas. Formato: `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`. Um array vazio retorna todos os serviços. Valores possíveis para o operador: 0 - *(padrão)* Contém; 1 - Igual; 2 - Não contém; 3 - Não é igual; 4 - Existe; 5 - Não existe.
without_problem_tags	flag	Retorna apenas os serviços sem tags de problema.
slaids	ID/array	Retorna apenas os serviços que estão vinculados ao(s) SLA(s) específico(s).
selectChildren	query	Retorna uma propriedade `children` com os serviços filhos. Suporta `count`.
selectParents	query	Retorna uma propriedade `parents` com os serviços pai. Suporta `count`.
selectTags	query	Retorna uma propriedade `tags` com as tags do serviço. Suporta `count`.

Parâmetro	Tipo	Descrição
selectProblemEvents	query	Retorna uma propriedade `problem_events` com um array de objetos de eventos de problema. O objeto de evento de problema possui as seguintes propriedades: `eventid` - `(ID)` ID do evento; `severity` - `(string)` Gravidade atual do evento; `name` - `(string)` Nome resolvido do evento. Suporta `count`.
selectProblemTags	query	Retorna uma propriedade `problem_tags` com as tags de problema. Suporta `count`.
selectStatusRules	query	Retorna uma propriedade `status_rules` com as regras de status. Suporta `count`.
selectStatusTimeline	object/array	Retorna uma propriedade `status_timeline` contendo as mudanças de estado do serviço para os períodos fornecidos. Formato `[{"period_from": "<period_from>", "period_to": "<period_to>"}, ...]` - `period_from` sendo a data de início (inclusiva; timestamp inteiro) e `period_to` sendo a data de término (exclusiva; timestamp inteiro) para o período de interesse. Retorna um array de entradas contendo uma propriedade `start_value` e um array `alarms` para as mudanças de estado dentro dos períodos especificados.
sortfield	string/array	Ordena o resultado pelas propriedades fornecidas. Valores possíveis: `serviceid`, `name`, `status`, `sortorder`, `created_at`.
countOutput	boolean	Esses parâmetros são descritos no comentário de referência.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valores de retorno

`(integer/array)` Retorna:

- uma matriz de objetos;
- a contagem de objetos recuperados, se o parâmetro `countOutput` foi usado.

Exemplos

Recuperando todos os serviços

Recupere todos os dados sobre todos os serviços e suas dependências.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "service.get",
    "params": {
        "output": "extend",
```

```
        "selectChildren": "extend",
        "selectParents": ["serviceid", "name"]
    },
    "id": 1
}
```

Resposta:


```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "serviceid": "1",
            "name": "Zabbix cluster",
            "status": "-1",
            "algorithm": "2",
            "sortorder": "0",
            "weight": "0",
            "propagation_rule": "0",
            "propagation_value": "0",
            "description": "",
            "uuid": "8d77bd91b62347e4b79382912eb5df95",
            "created_at": "1761742392",
            "readonly": false,
            "parents": [],
            "children": [
                {
                    "serviceid": "2",
                    "name": "Zabbix server node 1",
                    "status": "-1",
                    "algorithm": "2",
                    "sortorder": "0",
                    "weight": "0",
                    "propagation_rule": "0",
                    "propagation_value": "0",
                    "description": "",
                    "uuid": "195911d26d7f4e218d6217079bcd5929",
                    "created_at": "1761742623",
                    "readonly": false
                },
                {
                    "serviceid": "3",
                    "name": "Zabbix server node 2",
                    "status": "-1",
                    "algorithm": "2",
                    "sortorder": "0",
                    "weight": "0",
                    "propagation_rule": "0",
                    "propagation_value": "0",
                    "description": "",
                    "uuid": "9fc659a30fe244f690dff25fc2a9db5c",
                    "created_at": "1761742654",
                    "readonly": false
                }
            ]
        },
        {
            "serviceid": "2",
            "name": "Zabbix server node 1",
            "status": "-1",
            "algorithm": "2",
            "sortorder": "0",
            "weight": "0",
```

```
            "propagation_rule": "0",
            "propagation_value": "0",
            "description": "",
            "uuid": "195911d26d7f4e218d6217079bcd5929",
            "created_at": "1761742623",
            "readonly": false,
            "parents": [
                {
                    "serviceid": "1",
                    "name": "Zabbix cluster"
                }
            ],
            "children": []
        },
        {
            "serviceid": "3",
            "name": "Zabbix server node 2",
            "status": "-1",
            "algorithm": "2",
            "sortorder": "0",
            "weight": "0",
            "propagation_rule": "0",
            "propagation_value": "0",
            "description": "",
            "uuid": "9fc659a30fe244f690dff25fc2a9db5c",
            "created_at": "1761742654",
            "readonly": false,
            "parents": [
                {
                    "serviceid": "1",
                    "name": "Zabbix cluster"
                }
            ],
            "children": []
        }
    ],
    "id": 1
}
```


Fonte

CService::get() in *ui/include/classes/api/services/CService.php*.

service.update

Descrição

`object service.update(object/array services)`

Este método permite atualizar serviços existentes.

> **Note:**
> Este método está disponível para usuários de qualquer tipo. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Consulte Funções de usuário para obter mais informações.

Parâmetros

`(object/array)` propriedades do serviço a serem atualizadas.

A propriedade `serviceid` deve ser definida para cada serviço, todas as outras propriedades são opcionais. Somente as propriedades passadas serão atualizadas, todas as outras permanecerão inalteradas.

Além das propriedades padrão do serviço, o método aceita os seguintes parâmetros.

Parâmetro	Tipo	Descrição
children	array	Serviços filhos para substituir os serviços filhos atuais. Os serviços filhos devem ter apenas a propriedade `serviceid` definida.
parents	array	Serviços pais para substituir os serviços pais atuais. Os serviços pais devem ter apenas a propriedade `serviceid` definida.
tags	array	Tags de serviço para substituir as tags de serviço atuais.
problem_tags	array	Tags de problema para substituir as tags de problema atuais.
status_rules	array	Regras de status para substituir as regras de status atuais.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos serviços atualizados na propriedade `serviceids`.

Exemplos

Definindo o pai para um serviço

Torne o serviço com ID ”3” o pai do serviço com ID ”5”.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "service.update",
    "params": {
        "serviceid": "5",
        "parents": [
            {
                "serviceid": "3"
            }
        ]
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "serviceids": [
            "5"
        ]
    },
    "id": 1
}
```

Adicionando um downtime agendado

Adicione um downtime para o serviço com ID ”4” agendado semanalmente de segunda-feira 22:00 até terça-feira 10:00.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "service.update",
    "params": {
        "serviceid": "4",
        "times": [
            {
                "type": "1",
                "ts_from": "165600",
                "ts_to": "201600"
            }
```

```
        ]
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "serviceids": [
            "4"
        ]
    },
    "id": 1
}
```

Fonte

CService::update() in *ui/include/classes/api/services/CService.php*.

Serviço descoberto

Esta classe é projetada para trabalhar com serviços descobertos.

Referências de objetos:

- Serviço descoberto

Métodos disponíveis:

- dservice.get - recuperar serviços descobertos

Objeto de serviço descoberto

Os seguintes objetos estão diretamente relacionados à API `dservice`.

Serviço descoberto

Note:
Serviços descobertos são criados pelo Zabbix server e não podem ser modificados via API.

O objeto de serviço descoberto contém informações sobre um serviço descoberto por uma regra de descoberta de rede em um host. Ele possui as seguintes propriedades.

Propriedade	Tipo	Descrição
dserviceid	ID	ID do serviço descoberto.
dcheckid	ID	ID da verificação de descoberta usada para detectar o serviço.
dhostid	ID	ID do host descoberto executando o serviço.
dns	string	DNS do host executando o serviço.
ip	string	Endereço IP do host executando o serviço.
lastdown	timestamp	Hora em que o serviço descoberto ficou indisponível pela última vez.
lastup	timestamp	Hora em que o serviço descoberto ficou disponível pela última vez.
port	integer	Número da porta do serviço.
status	integer	Status do serviço. Valores possíveis: 0 - serviço disponível; 1 - serviço indisponível.
value	string	Valor retornado pelo serviço ao executar uma verificação de descoberta do Zabbix agent, SNMPv1, SNMPv2 ou SNMPv3.

#####dservice.get {#manual-api-reference-dservice-get}

Descrição

`integer/array dservice.get(object parameters)`

O método permite recuperar serviços descobertos de acordo com os parâmetros fornecidos.

> **Note:**
> Este método está disponível para usuários de qualquer tipo. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object)` Parâmetros que definem a saída desejada.

O método suporta os seguintes parâmetros.

Parâmetro	Tipo	Descrição
dserviceids	ID/array	Retorna apenas os serviços descobertos com os IDs fornecidos.
dhostids	ID/array	Retorna apenas os serviços descobertos que pertencem aos hosts descobertos fornecidos.
dcheckids	ID/array	Retorna apenas os serviços descobertos que foram detectados pelas verificações de descoberta fornecidas.
druleids	ID/array	Retorna apenas os serviços descobertos que foram detectados pelas regras de descoberta fornecidas.
selectDRules	query	Retorna uma propriedade `drules` com um array das regras de descoberta que detectaram o serviço.
selectDHosts	query	Retorna uma propriedade `dhosts` com um array dos hosts descobertos aos quais o serviço pertence.
selectHosts	query	Retorna uma propriedade `hosts` com os hosts com o mesmo endereço IP e proxy que o serviço. Suporta `count`.
limitSelects	integer	Limita o número de registros retornados por subconsultas. Aplica-se às seguintes subconsultas: `selectHosts` - o resultado será ordenado por `hostid`.
sortfield	string/array	Ordena o resultado pelas propriedades fornecidas. Valores possíveis: `dserviceid`, `dhostid`, `ip`.
countOutput	boolean	Esses parâmetros são descritos no comentário de referência.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valores de retorno

`(integer/array)` Retorna:

- uma matriz de objetos;
- a contagem de objetos recuperados, se o parâmetro `countOutput` tiver sido usado.

Exemplos

Recuperar serviços descobertos em um host

Recupera todos os serviços descobertos detectados no host descoberto "11".

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "dservice.get",
    "params": {
        "output": "extend",
        "dhostids": "11"
    },
    "id": 1
}
```

Resposta:

```
{
    "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
}
```

Veja também

- Host descoberto
- Verificação de descoberta
- Host

Fonte

CDService::get() em *ui/include/classes/api/services/CDService.php*.

SLA

Esta classe foi projetada para trabalhar com objetos SLA (Acordo de Nível de Serviço) usados para estimar o desempenho da infraestrutura de TI e dos serviços de negócios.

Referências de objetos:

- SLA
- Agenda do SLA
- Tempo de inatividade excluído do SLA
- Tag de serviço do SLA

Métodos disponíveis:

- sla.create - criar novos SLAs
- sla.delete - excluir SLAs
- sla.get - recuperar SLAs
- sla.getsli - recuperar dados do Indicador de Nível de Serviço (SLI) para SLAs
- sla.update - atualizar SLAs

Objeto SLA

Os seguintes objetos estão diretamente relacionados à API `sla` (Acordo de Nível de Serviço).

SLA

O objeto SLA possui as seguintes propriedades.

Propriedade	Tipo	Descrição
slaid	ID	ID do SLA. Comportamento da propriedade: - *somente leitura* - *obrigatório* para operações de atualização
name	string	Nome do SLA. Comportamento da propriedade: - *obrigatório* para operações de criação
period	integer	Período de relatório do SLA. Valores possíveis: 0 - diário; 1 - semanal; 2 - mensal; 3 - trimestral; 4 - anual. Comportamento da propriedade: - *obrigatório* para operações de criação
slo	float	Objetivo de Nível de Serviço mínimo aceitável expresso em percentual. Se o Indicador de Nível de Serviço (SLI) cair abaixo, o SLA é considerado em estado de problema/não cumprido. Valores possíveis: 0-100 (até 4 casas decimais). Comportamento da propriedade: - *obrigatório* para operações de criação
effective_date	integer	Data de início de vigência do SLA. Valores possíveis: timestamp da data em UTC.
timezone	string	Fuso horário do relatório, por exemplo: `Europe/London`, `UTC`. Para a lista completa de fusos horários suportados, consulte a documentação do PHP. Comportamento da propriedade: - *obrigatório* para operações de criação
status	integer	Status do SLA. Valores possíveis: 0 - *(padrão)* SLA desabilitado; 1 - SLA habilitado.
description	string	Descrição do SLA.

Agenda de SLA

O objeto de agenda de SLA define períodos em que o(s) serviço(s) conectado(s) estão programados para estar em funcionamento. Ele possui as seguintes propriedades.

Propriedade	Tipo	Descrição
period_from	integer	Hora de início do período semanal recorrente (inclusive). Valores possíveis: número de segundos (contando a partir de domingo). Comportamento da propriedade: - *obrigatório*
period_to	integer	Hora de término do período semanal recorrente (exclusivo). Valores possíveis: número de segundos (contando a partir de domingo). Comportamento da propriedade: - *obrigatório*

SLA - tempo de inatividade excluído

O objeto de tempo de inatividade excluído define períodos em que o(s) serviço(s) conectado(s) estão programados para ficar fora de operação, sem afetar o SLI, por exemplo, durante uma manutenção planejada. Ele possui as seguintes propriedades.

Propriedade	Tipo	Descrição
name	string	Nome do tempo de inatividade excluído. Comportamento da propriedade: - *obrigatório*
period_from	integer	Hora de início do tempo de inatividade excluído (inclusivo). Valores possíveis: timestamp. Comportamento da propriedade: - *obrigatório*
period_to	integer	Hora de término do tempo de inatividade excluído (exclusivo). Valores possíveis: timestamp. Comportamento da propriedade: - *obrigatório*

Tag de serviço de SLA

O objeto de tag de serviço de SLA vincula os serviços a serem incluídos nos cálculos do SLA. Ele possui as seguintes propriedades.

Propriedade	Tipo	Descrição
tag	string	Nome da tag do serviço de SLA. Comportamento da propriedade: - *obrigatório*
operator	integer	Operador da tag do serviço de SLA. Valores possíveis: 0 - *(padrão)* Igual; 2 - Contém.
value	string	Valor da tag do serviço de SLA.

sla.create

Descrição

`object sla.create(object/array SLAs)`

Este método permite criar novos objetos de SLA.

> **Note:**
> Este método está disponível apenas para os tipos de usuário *Admin* e *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object/array)` Objetos de SLA a serem criados.

Além das propriedades padrão de SLA, o método aceita os seguintes parâmetros.

Parâmetro	Tipo	Descrição
service_tags	array	Tags de serviço de SLA a serem criadas para o SLA. Comportamento do parâmetro: - *obrigatório*
schedule	array	Agenda de SLA a ser criada para o SLA. Especificar um parâmetro vazio será interpretado como uma agenda 24x7. Padrão: agenda 24x7.
excluded_downtimes	array	Períodos de inatividade excluídos do SLA a serem criados para o SLA.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos SLAs criados na propriedade `slaids`. A ordem dos IDs retornados corresponde à ordem dos SLAs passados.

Exemplos

Criando um SLA

Instrua a criar uma entrada de SLA para: * acompanhar o tempo de atividade dos serviços relacionados ao SQL-engine; * agendamento personalizado de todos os dias da semana, excluindo a última hora do sábado; * uma data efetiva para o último dia do ano de 2022; * com 1 hora e 15 minutos de tempo de inatividade planejado começando à meia-noite do dia 4 de julho; * o cálculo do relatório semanal de SLA estará ativado; * o SLO mínimo aceitável será de 99,9995%.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "sla.create",
    "params": [
        {
            "name": "Database Uptime",
            "slo": "99.9995",
            "period": "1",
            "timezone": "America/Toronto",
            "description": "Provide excellent uptime for main database engines.",
            "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": "Software version upgrade rollout",
                    "period_from": "1648760400",
                    "period_to": "1648764900"
                }
            ]
        }
    ],
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "slaids": [
            "5"
        ]
    },
    "id": 1
}
```

Fonte

CSla::create() em *ui/include/classes/api/services/CSla.php*.

sla.delete

Descrição

`object sla.delete(array slaids)`

Este método permite excluir entradas de SLA.

> **Note:**
> Este método está disponível apenas para os tipos de usuário *Admin* e *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(array)` IDs dos SLAs a serem excluídos.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos SLAs excluídos na propriedade `slaids`.

Exemplos

Excluindo vários SLAs

Exclua duas entradas de SLA.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "sla.delete",
    "params": [
        "4",
```

```
        "5"
    ],
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "slaids": [
            "4",
            "5"
        ]
    },
    "id": 1
}
```

Fonte

CSla::delete() em *ui/include/classes/api/services/CSla.php*.

sla.get

Descrição

`integer/array sla.get(object parameters)`

O método permite recuperar objetos de SLA de acordo com os parâmetros fornecidos.

Note:
Este método está disponível para usuários de qualquer tipo. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object)` Parâmetros que definem a saída desejada.

O método suporta os seguintes parâmetros.

Parâmetro	Tipo	Descrição
slaids	ID/array	Retorna apenas SLAs com os IDs fornecidos.
serviceids	ID/array	Retorna apenas SLAs correspondentes aos serviços específicos.
selectSchedule	query	Retorna uma propriedade `schedule` com os agendamentos do SLA. Suporta `count`.
selectExcludedDowntimes	query	Retorna uma propriedade `excluded_downtimes` com os períodos de indisponibilidade excluídos do SLA. Suporta `count`.
selectServiceTags	query	Retorna uma propriedade `service_tags` com as tags de serviço do SLA. Suporta `count`.
sortfield	string/array	Ordena o resultado pelas propriedades fornecidas. Valores possíveis: `slaid`, `name`, `period`, `slo`, `effective_date`, `timezone`, `status`, `description`.
countOutput	boolean	Esses parâmetros são descritos no comentário de referência.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	

Parâmetro	Tipo	Descrição
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valores de retorno

`(integer/array)` Retorna:

- um array de objetos;
- a contagem de objetos recuperados, se o parâmetro `countOutput` tiver sido usado.

Exemplos

Recuperando todos os SLAs

Recupere todos os dados sobre todos os SLAs e suas propriedades.

Requisição:

```
{
    "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
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "1": {
            "slaid": "1",
            "name": "Database Uptime",
            "period": "1",
            "slo": "99.9995",
            "effective_date": "1672444800",
            "timezone": "America/Toronto",
            "status": "1",
            "description": "Provide excellent uptime for main SQL database engines.",
            "service_tags": [
                {
                    "tag": "database",
                    "operator": "0",
                    "value": "mysql"
                },
                {
                    "tag": "database",
                    "operator": "0",
                    "value": "postgresql"
                }
            ],
            "schedule": [
                {
                    "period_from": "0",
                    "period_to": "601200"
                }
```

```
            ],
            "excluded_downtimes": [
                {
                    "name": "Software version upgrade rollout",
                    "period_from": "1648760400",
                    "period_to": "1648764900"
                }
            ]
        }
    },
    "id": 1
}
```

Fonte

CSla:get() em *ui/include/classes/api/services/CSla.php*.

sla.getsli

Descrição

`object sla.getsli(object parameters)`

Este método permite calcular os dados do Indicador de Nível de Serviço (SLI) para um Acordo de Nível de Serviço (SLA).

Note:
Este método está disponível para usuários de qualquer tipo. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object)` Parâmetros contendo o ID do SLA, períodos de relatório e, opcionalmente, os IDs dos serviços - para calcular o SLI.

Parâmetro	Tipo	Descrição
slaid	ID	ID do SLA para retornar informações de disponibilidade. Comportamento do parâmetro: - *obrigatório*
period_from	timestamp	Timestamp inicial (inclusivo) para relatar o SLI. Valores possíveis: timestamp Unix.
period_to	timestamp	Timestamp final (inclusivo) para relatar o SLI. Valores possíveis: timestamp Unix.
periods	integer	Número de períodos a serem relatados. Valores possíveis: 1-100
serviceids	ID/array	IDs dos serviços para retornar o SLI.

Particionamento de períodos

A tabela a seguir demonstra o arranjo dos períodos retornados com base nas combinações dos parâmetros.

Note:
Os períodos retornados não precedem o primeiro período disponível (com base na data efetiva do SLA) e não excedem o período atual.

Parâmetros			Períodos retornados
period_from	**period_to**	**periods**	
-	-	-	Últimos 20 períodos, incluindo o atual.
-	-	N	Últimos N períodos.
-	especificado	-	Últimos 20 períodos antes de `period_to`.

Parâmetros			Períodos retornados
-	especificado	N	Últimos N períodos antes de `period_to`.
especificado	-	-	Primeiros 20 períodos a partir de `period_from`.
especificado	-	N	Primeiros N períodos a partir de `period_from`.
especificado	especificado	-	Até 100 períodos no intervalo especificado.
especificado	especificado	N	N períodos no intervalo especificado.

Valores de retorno

`(object)` Retorna os resultados do cálculo.

Propriedade	Tipo	Descrição
periods	array	Lista dos períodos reportados. Cada período reportado é representado como um objeto consistindo de: - `period_from` - `(timestamp)` Timestamp inicial (inclusivo) do período reportado. - `period_to` - `(timestamp)` Timestamp final (exclusivo) do período reportado. Os períodos são ordenados por `period_from`, com os períodos mais antigos aparecendo primeiro.
serviceids	array	Lista de IDs de serviços nos períodos reportados. A ordem de classificação da lista não está definida. Mesmo que o parâmetro `serviceids` tenha sido passado para o método `sla.getsli`.
sli	array	Dados SLI (como um **array bidimensional**) para cada período e serviço reportado. O índice da propriedade `periods` é usado como a **primeira** dimensão da propriedade `sli`. O índice da propriedade `serviceids` é usado como a **segunda** dimensão da propriedade `sli`.

Dados SLI

Os dados SLI retornados para cada período relatado e serviço consistem em:

Propriedade	Tipo	Descrição
uptime	integer	Quantidade de tempo que o serviço permaneceu em estado *OK* durante o tempo de atividade agendado, menos os períodos de inatividade excluídos.
downtime	integer	Quantidade de tempo que o serviço permaneceu em estado *não OK* durante o tempo de atividade agendado, menos os períodos de inatividade excluídos.
sli	float	SLI (porcentagem do tempo total de atividade), com base no uptime e downtime.
error_budget	integer	Orçamento de erro (em segundos), com base no SLI e no SLO.

Propriedade	Tipo	Descrição
excluded_downtimes	array	Array de períodos de inatividade excluídos neste período de relatório. Cada objeto conterá os seguintes parâmetros: - `name` - (`string`) Nome do período de inatividade excluído. - `period_from` - (`timestamp`) Timestamp inicial (inclusivo) do período de inatividade excluído. - `period_to` - (`timestamp`) Timestamp final (exclusivo) do período de inatividade excluído. Os períodos de inatividade excluídos são classificados por `period_from`, com os períodos mais antigos aparecendo primeiro.

Exemplos

Calculando o SLI para um SLA diário

Recupere os dados de SLI para os serviços com IDs ”1” e ”4” que estão vinculados ao SLA com ID ”1”. Recupere os dados para um único período até ”1761861599” (30 de outubro de 2025 23:59:59 GMT+0200). Como o período de relatório do SLA é diário, os dados de SLI são recuperados de ”1761775200” (30 de outubro de 2025 00:00:00 GMT+0200) até ”1761861600” (31 de outubro de 2025 00:00:00 GMT+0200).

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "sla.getsli",
    "params": {
        "slaid": "1",
        "serviceids": [
            1,
            4
        ],
        "periods": 1,
        "period_to": 1761861599,
    },
    "id": 1
}
```

Resposta:

```
{
    "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
}
```

Calculando SLI para um SLA mensal

Recupere os dados de SLI dos serviços com IDs ”50”, ”60” e ”70” que estão vinculados ao SLA com ID ”5”. Recupere os dados para três períodos a partir de ”1635724800” (01 de nov de 2021 00:00:00 UTC). Como o período de relatório do SLA é mensal, os dados de SLI são recuperados para os três meses seguintes:

- De ”1635724800” (01 de nov de 2021 00:00:00 UTC) a ”1638316800” (01 de dez de 2021 00:00:00 UTC)
- De ”1638316800” (01 de dez de 2021 00:00:00 UTC) a ”1640995200” (01 de jan de 2022 00:00:00 UTC)
- De ”1640995200” (01 de jan de 2022 00:00:00 UTC) a ”1643673600” (01 de fev de 2022 00:00:00 UTC)

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "sla.getsli",
    "params": {
        "slaid": "5",
        "serviceids": [
            50,
            60,
            70
        ],
        "periods": 3,
        "period_from": 1635724800
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "periods": [
            {
                "period_from": 1635724800,
                "period_to": 1638316800
            },
            {
                "period_from": 1638316800,
                "period_to": 1640995200
            },
            {
                "period_from": 1640995200,
                "period_to": 1643673600
            }
        ],
        "serviceids": [
```

```
            50,
            60,
            70
        ],
        "sli": [
            [
                {
                    "uptime": 1186212,
                    "downtime": 0,
                    "sli": 100,
                    "error_budget": 0,
                    "excluded_downtimes": [
                        {
                            "name": "Maintenance Nov 25 - Dec 01",
                            "period_from": 1637836212,
                            "period_to": 1638316800
                        }
                    ]
                },
                {
                    "uptime": 1186212,
                    "downtime": 0,
                    "sli": 100,
                    "error_budget": 0,
                    "excluded_downtimes": [
                        {
                            "name": "Maintenance Nov 25 - Dec 01",
                            "period_from": 1637836212,
                            "period_to": 1638316800
                        }
                    ]
                },
                {
                    "uptime": 1186212,
                    "downtime": 0,
                    "sli": 100,
                    "error_budget": 0,
                    "excluded_downtimes": [
                        {
                            "name": "Maintenance Nov 25 - Dec 01",
                            "period_from": 1637836212,
                            "period_to": 1638316800
                        }
                    ]
                }
            ],
            [
                {
                    "uptime": 1147548,
                    "downtime": 0,
                    "sli": 100,
                    "error_budget": 0,
                    "excluded_downtimes": [
                        {
                            "name": "Maintenance Dec 02 - Dec 10",
                            "period_from": 1638439200,
                            "period_to": 1639109652
                        }
                    ]
                },
                {
                    "uptime": 1147548,
```

```
                    "downtime": 0,
                    "sli": 100,
                    "error_budget": 0,
                    "excluded_downtimes": [
                        {
                            "name": "Maintenance Dec 02 - Dec 10",
                            "period_from": 1638439200,
                            "period_to": 1639109652
                        }
                    ]
                },
                {
                    "uptime": 1147548,
                    "downtime": 0,
                    "sli": 100,
                    "error_budget": 0,
                    "excluded_downtimes": [
                        {
                            "name": "Maintenance Dec 02 - Dec 10",
                            "period_from": 1638439200,
                            "period_to": 1639109652
                        }
                    ]
                }
            ],
            [
                {
                    "uptime": 1674000,
                    "downtime": 0,
                    "sli": 100,
                    "error_budget": 0,
                    "excluded_downtimes": []
                },
                {
                    "uptime": 1674000,
                    "downtime": 0,
                    "sli": 100,
                    "error_budget": 0,
                    "excluded_downtimes": []
                },
                {
                    "uptime": 1674000,
                    "downtime": 0,
                    "sli": 100,
                    "error_budget": 0,
                    "excluded_downtimes": []
                }
            ]
        ]
    },
    "id": 1
}
```


Fonte

CSla::getSli() em *ui/include/classes/api/services/CSla.php*

sla.update

Descrição

`object sla.update(object/array slaids)`

Este método permite atualizar entradas de SLA existentes.

Note:
Este método está disponível apenas para os tipos de usuário *Admin* e *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Consulte Funções de usuário para mais informações.

Parâmetros

`(object/array)` Propriedades do SLA a serem atualizadas.

A propriedade `slaid` deve ser definida para cada SLA, todas as outras propriedades são opcionais. Somente as propriedades passadas serão atualizadas, todas as outras permanecerão inalteradas.

Além das propriedades padrão do SLA, o método aceita os seguintes parâmetros.

Parâmetro	Tipo	Descrição
service_tags	array	Tags de serviço do SLA para substituir as tags de serviço do SLA atuais.
schedule	array	Agenda do SLA para substituir a atual. Especificar o parâmetro como vazio será interpretado como uma agenda 24x7.
excluded_downtimes	array	Períodos de inatividade excluídos do SLA para substituir os atuais.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos SLAs atualizados na propriedade `slaids`.

Exemplos

Atualizando tags de serviço

Faça com que o SLA com ID ”5” seja calculado em intervalos mensais para serviços relacionados a NoSQL, sem alterar sua programação ou períodos de inatividade excluídos; defina o SLO para 95%.

Requisição:

```
{
    "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
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "slaids": [
            "5"
```

```
        ]
    },
    "id": 1
}
```

Alterando o agendamento de um SLA

Altere o SLA com ID ”5” para um agendamento 24x7.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "sla.update",
    "params": {
        "slaid": "5",
        "schedule": []
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "slaids": [
            "5"
        ]
    },
    "id": 1
}
```

Alterando os períodos de indisponibilidade excluídos para um SLA

Adicione um período de indisponibilidade planejado de 4 horas para upgrade de RAM em 6 de abril de 2022, mantendo (precisa ser definido novamente) uma atualização de software previamente existente planejada para 4 de julho para o SLA com ID ”5”.

Requisição:

```
{
    "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
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "slaids": [
            "5"
```

```
        ]
    },
    "id": 1
}
```

Fonte

CSla::update() em *ui/include/classes/api/services/CSla.php*.

Task

Esta classe foi projetada para trabalhar com tarefas (como verificar items ou regras de descoberta de baixo nível sem recarregar a configuração).

Referências de objetos:

- Task
  - Objeto de solicitação 'Executar agora'
  - Objeto de solicitação 'Atualizar configuração do proxy'
  - Objeto de solicitação 'Informações de diagnóstico'
    * Objeto de solicitação de estatística
    * Objeto de resultado de estatística

Métodos disponíveis:

- task.create - criar novas tarefas
- task.get - recuperar tarefas

Objeto task

Os seguintes objetos estão diretamente relacionados à API `task`.

Task

O objeto task possui as seguintes propriedades:

Propriedade	Tipo	Descrição
taskid	ID	ID da task. Comportamento da propriedade: - *somente leitura*
type	integer	Tipo da task. Valores possíveis: 1 - Informação de diagnóstico; 2 - Atualizar configuração do proxy; 6 - Executar agora. Desde o Zabbix 7.0.19, usuários do tipo *Admin* e *User* podem criar tasks 'Executar agora'. Note que `task.get` sempre retorna "7" (Resumo da execução da task). Comportamento da propriedade: - *obrigatório*

Propriedade	Tipo	Descrição
status	integer	Status da task. Valores possíveis: 1 - nova task; 2 - task em andamento; 3 - task concluída; 4 - task expirada. Comportamento da propriedade: - *somente leitura*
clock	timestamp	Hora em que a task foi criada. Comportamento da propriedade: - *somente leitura*
ttl	integer	Tempo em segundos após o qual a task expira. Comportamento da propriedade: - *somente leitura*
proxyid	ID	ID do proxy sobre o qual a estatística de informação de diagnóstico é coletada. Comportamento da propriedade: - *suportado* se `type` estiver definido como ”Informação de diagnóstico” ou ”Atualizar configuração do proxy”
request	object	Objeto de requisição da task de acordo com o tipo da task: O objeto da task ’Executar agora’ é descrito em detalhes abaixo; O objeto da task ’Atualizar configuração do proxy’ é descrito em detalhes abaixo; O objeto da task ’Informação de diagnóstico’ é descrito em detalhes abaixo. Comportamento da propriedade: - *obrigatório*
result	object	Objeto de resultado da task de informação de diagnóstico. Pode conter NULL se o resultado ainda não estiver pronto. O objeto de resultado é descrito em detalhes abaixo. Comportamento da propriedade: - *somente leitura*

Objeto de solicitação ’Executar agora’

O objeto de solicitação da tarefa ’Executar agora’ possui as seguintes propriedades.

Propriedade	Tipo	Descrição
itemid	ID	ID do item e das regras de descoberta de baixo nível. Desde o Zabbix 7.0.19, usuários do tipo *Admin* e *User* podem ’Executar agora’ itens em hosts para os quais possuem permissão de *leitura-escrita* permission, ou permissão de *leitura* e a ação `invoke_execute_now` action habilitada para sua função. O mesmo se aplica a usuários do tipo *Admin* para regras de descoberta de baixo nível (LLD).

Objeto de solicitação ’Atualizar configuração do proxy’

O objeto de solicitação da tarefa ’Atualizar configuração do proxy’ possui as seguintes propriedades.

Propriedade	Tipo	Descrição
proxyids	array	IDs do proxy.

Objeto de solicitação de 'Informações de diagnóstico'

O objeto de solicitação de tarefa de informações de diagnóstico possui as seguintes propriedades. O objeto de solicitação de estatísticas para todos os tipos de propriedades é descrito em detalhes abaixo.

Propriedade	Tipo	Descrição
historycache	object	Solicitação de estatísticas do cache de histórico. Disponível no server e proxy.
valuecache	object	Solicitação de estatísticas do cache de items. Disponível no server.
preprocessing	object	Solicitação de estatísticas do gerenciador de pré-processamento. Disponível no server e proxy.
alerting	object	Solicitação de estatísticas do gerenciador de alertas. Disponível no server.
lld	object	Solicitação de estatísticas do gerenciador de LLD. Disponível no server.

Objeto de solicitação de estatísticas

O objeto de solicitação de estatísticas é usado para definir que tipo de informação deve ser coletada sobre os processos internos do server/proxy. Ele possui as seguintes propriedades.

Propriedade	Tipo	Descrição
stats	query	Propriedades do objeto de estatísticas a serem retornadas. A lista de campos disponíveis para cada tipo de informação estatística de diagnóstico está descrita em detalhes abaixo. Padrão: `extend` retornará todos os campos de estatísticas disponíveis.
top	object	Objeto para ordenar e limitar os valores de estatísticas retornados. A lista de campos disponíveis para cada tipo de informação estatística de diagnóstico está descrita em detalhes abaixo. Exemplo: { "source.alerts": 10 }

Lista de campos estatísticos disponíveis para cada tipo de solicitação de informação de diagnóstico

Os seguintes campos estatísticos podem ser solicitados para cada tipo de propriedade de solicitação de informação de diagnóstico.

Tipo de diagnóstico	Campos disponíveis	Descrição
historycache	items	Número de items em cache.
	values	Número de valores em cache.
	memory	Estatísticas de memória compartilhada (espaço livre, número de blocos usados, número de blocos livres, tamanho máximo do bloco livre).
	memory.data	Estatísticas de memória compartilhada do cache de dados históricos.
	memory.index	Estatísticas de memória compartilhada do cache de índice histórico.
valuecache	items	Número de items em cache.
	values	Número de valores em cache.
	memory	Estatísticas de memória compartilhada (espaço livre, número de blocos usados, número de blocos livres, tamanho máximo do bloco livre).
	mode	Modo do cache de valores.
preprocessing	values	Número de valores em fila.
	preproc.values	Número de valores em fila com etapas de pré-processamento.
alerting	alerts	Número de alertas em fila.
lld	rules	Número de regras em fila.
	values	Número de valores em fila.

Lista de campos de ordenação disponíveis para cada tipo de solicitação de informação de diagnóstico

Os seguintes campos estatísticos podem ser usados para ordenar e limitar as informações solicitadas.

Tipo de diagnóstico	Campos disponíveis	Tipo
historycache	values	integer
valuecache	values	integer
	request.values	integer
preprocessing	values	integer
alerting	media.alerts	integer
	source.alerts	integer
lld	values	integer

Objeto de resultado estatístico

O objeto de resultado estatístico é recuperado no campo `result` do objeto de tarefa.

Propriedade	Tipo	Descrição
status	integer	Status do resultado da tarefa. Valores possíveis: -1 - ocorreu um erro durante a execução da tarefa; 0 - resultado da tarefa foi criado. Comportamento da propriedade: - *somente leitura*
data	string/object	Resultados de acordo com o objeto de solicitação estatística de uma tarefa de informação de diagnóstico específica. Contém a mensagem de erro se ocorreu um erro durante a execução da tarefa.

task.create

Descrição

`object task.create(object/array tasks)`

Este método permite criar tarefas.

> **Note:**
> Este método está disponível para usuários de qualquer tipo (para usuários do tipo *Admin* e *User* desde o Zabbix 7.0.19). As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Consulte Funções de usuário para mais informações.

Parâmetros

`(object/array)` Uma tarefa a ser criada.

O método aceita tarefas com as propriedades padrão da tarefa.

Observe que tarefas 'Executar agora' podem ser criadas apenas para os seguintes tipos de items/regras de descoberta:

- Zabbix agent (passivo)
- Verificação simples
- SNMP agent (v1/v2/v3)
- Zabbix internal
- Verificação externa
- Monitor de banco de dados
- HTTP agent
- IPMI agent
- SSH agent
- TELNET agent
- JMX agent
- Calculado
- Item dependente

- Script
- Navegador

Se o item/regra de descoberta for do tipo ”Item dependente”, então seu item mestre também deve ser um dos tipos acima.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs das tarefas criadas na propriedade `taskids`. Uma tarefa é criada para cada item e regra de descoberta de baixo nível. A ordem dos IDs retornados corresponde à ordem dos `itemids` passados.

Exemplos

Criando uma tarefa

Crie uma tarefa ’Executar agora’ para um item e uma regra de descoberta de baixo nível.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "task.create",
    "params": [
        {
            "type": 6,
            "request": {
                "itemid": "10092"
            }
        },
        {
            "type": 6,
            "request": {
                "itemid": "10093"
            }
        }
    ],
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "taskids": [
            "1",
            "2"
        ]
    },
    "id": 1
}
```

Crie uma tarefa ’Atualizar configuração do proxy’ para dois proxies.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "task.create",
    "params": [
        {
            "type": 2,
            "request": {
                "proxyids": ["10459", "10460"]
            }
        }
    ],
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "taskids": [
            "1"
        ]
    },
    "id": 1
}
```

Crie uma tarefa 'Informações de diagnóstico'.

Requisição:

```
{
    "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
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "taskids": [
            "3"
        ]
    },
    "id": 1
}
```

Veja também

- Task
- Objeto de solicitação 'Executar agora'
- Objeto de solicitação 'Atualizar configuração do proxy'
- Objeto de solicitação 'Informações de diagnóstico'

Fonte

CTask::create() in *ui/include/classes/api/services/CTask.php*.

task.get

Descrição

`integer/array task.get(object parameters)`

O método permite recuperar tarefas de acordo com os parâmetros fornecidos.

> **Note:**
> Este método está disponível apenas para o tipo de usuário *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object)` Parâmetros que definem a saída desejada.

O método suporta os seguintes parâmetros.

Parâmetro	Tipo	Descrição
taskids	ID/array	Retornar apenas tarefas com os IDs fornecidos.
output	query	Esses parâmetros são descritos no comentário de referência.
preservekeys	boolean	

Valores de retorno

`(integer/array)` Retorna um array de objetos.

Exemplos

Recuperar tarefa por ID

Recupere todos os dados sobre a tarefa com o ID "1".

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "task.get",
    "params": {
        "output": "extend",
        "taskids": "1"
    },
    "id": 1
}
```

Resposta:

```
{
    "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
}
```


Veja também

- Task
- Objeto de solicitação 'Executar agora'
- Objeto de solicitação 'Atualizar configuração do proxy'
- Objeto de solicitação 'Informações de diagnóstico'

Fonte

CTask::get() in *ui/include/classes/api/services/CTask.php*.

Template

Esta classe foi projetada para trabalhar com templates.

Referências de objetos:

- Template
- Tag de template

Métodos disponíveis:

- template.create - criar novos templates
- template.delete - excluir templates
- template.get - recuperar templates
- template.massadd - adicionar objetos relacionados aos templates
- template.massremove - remover objetos relacionados dos templates
- template.massupdate - substituir ou remover objetos relacionados dos templates
- template.update - atualizar templates

Objeto template

Os seguintes objetos estão diretamente relacionados à API `template`.

Template

O objeto template possui as seguintes propriedades.

Propriedade	Tipo	Descrição
templateid	ID	ID do template. Comportamento da propriedade: - *somente leitura* - *obrigatório* para operações de atualização
host	string	Nome técnico do template. Comportamento da propriedade: - *obrigatório* para operações de criação
description	text	Descrição do template.
name	string	Nome visível do template. Padrão: valor da propriedade `host`.
uuid	string	Identificador universal único, usado para vincular templates importados aos já existentes. Gerado automaticamente, se não for fornecido.
vendor_name	string	Nome do fornecedor do template. Para operações de criação, tanto `vendor_name` quanto `vendor_version` devem ser definidos ou deixados em branco. Para operações de atualização, `vendor_version` pode ser deixado em branco se houver um valor no banco de dados.
vendor_version	string	Versão do fornecedor do template. Para operações de criação, tanto `vendor_name` quanto `vendor_version` devem ser definidos ou deixados em branco. Para operações de atualização, `vendor_name` pode ser deixado em branco se houver um valor no banco de dados.

Tag de template

O objeto de tag de template possui as seguintes propriedades.

Propriedade	Tipo	Descrição
tag	string	Nome da tag do template. Comportamento da propriedade: - *obrigatório*
value	string	Valor da tag do template.

template.create

Descrição

`object template.create(object/array templates)`

Este método permite criar novas templates.

Note:
Este método está disponível apenas para os tipos de usuário *Admin* e *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object/array)` Templates a serem criados.

Além das propriedades padrão do template, o método aceita os seguintes parâmetros.

Parâmetro	Tipo	Descrição
groups	object/array	Grupos de template aos quais o template será adicionado. Os grupos de template devem ter apenas a propriedade `groupid` definida. Comportamento do parâmetro: - *obrigatório*
tags	object/array	Tags de template.
templates	object/array	Templates a serem vinculados ao template. Os templates devem ter apenas a propriedade `templateid` definida.
macros	object/array	Macros de usuário a serem criadas para o template.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos templates criados na propriedade `templateids`. A ordem dos IDs retornados corresponde à ordem dos modelos passados.

Exemplos

Criando um template

Crie um template com tags e vincule dois templates a este template.

Requisição:

```
{
    "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
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "11117"
        ]
    },
    "id": 1
}
```

Fonte

CTemplate::create() in *ui/include/classes/api/services/CTemplate.php*.

template.delete

Descrição

`object template.delete(array templateIds)`

Este método permite excluir templates.

Excluir um template causará a exclusão de todas as entidades do template (items, triggers, gráficos, etc.). Para deixar as entidades do template nos hosts, mas excluir o template em si, primeiro desvincule o template dos hosts necessários usando um destes métodos: template.update, template.massupdate, host.update, host.massupdate.

> **Note:**
> Este método está disponível apenas para os tipos de usuário *Admin* e *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(array)` IDs dos templates a serem deletados.

Retorno de valores

`(object)` Retorna um objeto contendo os IDs dos templates deletados na propriedade `templateids`.

Exemplos

Excluindo múltiplos templates

Exclua dois templates.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "template.delete",
    "params": [
        "13",
        "32"
    ],
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "13",
            "32"
        ]
    },
    "id": 1
}
```

Fonte

CTemplate::delete() in *ui/include/classes/api/services/CTemplate.php*.

template.get

Descrição

`integer/array template.get(object parameters)`

O método permite recuperar templates de acordo com os parâmetros fornecidos.

Note:
Este método está disponível para usuários de qualquer tipo. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object)` Parâmetros que definem a saída desejada.

O método suporta os seguintes parâmetros.

Parâmetro	Tipo	Descrição
templateids	ID/array	Retorna apenas templates com os IDs de template fornecidos.
groupids	ID/array	Retorna apenas templates que pertencem aos grupos de templates fornecidos.
parentTemplateids	ID/array	Retorna apenas templates aos quais o template fornecido está vinculado.
hostids	ID/array	Retorna apenas templates que estão vinculados aos hosts/templates fornecidos.
graphids	ID/array	Retorna apenas templates que contêm os gráficos fornecidos.
itemids	ID/array	Retorna apenas templates que contêm os items fornecidos.
triggerids	ID/array	Retorna apenas templates que contêm os triggers fornecidos.
with_items	flag	Retorna apenas templates que possuem items.
with_triggers	flag	Retorna apenas templates que possuem triggers.
with_graphs	flag	Retorna apenas templates que possuem gráficos.
with_httptests	flag	Retorna apenas templates que possuem cenários web.
evaltype	integer	Método de avaliação da tag. Valores possíveis: 0 - *(padrão)* E/Ou; 2 - Ou.
tags	object/array	Retorna apenas templates com as tags fornecidas. Formato: `[{"tag": "<tag>", "value": "<valor>", "operator": "<operador>"}, ...]`. Um array vazio retorna todos os templates. Valores possíveis para operator: 0 - *(padrão)* Contém; 1 - Igual; 2 - Não contém; 3 - Diferente; 4 - Existe; 5 - Não existe.
selectTags	query	Retorna as tags do template na propriedade `tags`.
selectHosts	query	Retorna os hosts que estão vinculados ao template na propriedade `hosts`. Suporta `count`.
selectTemplateGroups	query	Retorna os grupos de templates aos quais o template pertence na propriedade `templategroups`.
selectTemplates	query	Retorna os templates aos quais o template fornecido está vinculado na propriedade `templates`. Suporta `count`.
selectParentTemplates	query	Retorna os templates que estão vinculados ao template fornecido na propriedade `parentTemplates`. Suporta `count`.
selectHttpTests	query	Retorna os cenários web do template na propriedade `httpTests`. Suporta `count`.
selectItems	query	Retorna os items do template na propriedade `items`. Suporta `count`.

Parâmetro	Tipo	Descrição
selectDiscoveries	query	Retorna as descobertas de baixo nível do template na propriedade `discoveries`. Suporta `count`.
selectTriggers	query	Retorna os triggers do template na propriedade `triggers`. Suporta `count`.
selectGraphs	query	Retorna os gráficos do template na propriedade `graphs`. Suporta `count`.
selectMacros	query	Retorna as macros do template na propriedade `macros`.
selectDashboards	query	Retorna os dashboards do template na propriedade `dashboards`. Suporta `count`.
selectValueMaps	query	Retorna uma propriedade `valuemaps` com os mapas de valores do template.
limitSelects	integer	Limita o número de registros retornados por subconsultas. Aplica-se às seguintes subconsultas: `selectTemplates` - os resultados serão ordenados por `name`; `selectHosts` - ordenados por `host`; `selectParentTemplates` - ordenados por `host`; `selectItems` - ordenados por `name`; `selectDiscoveries` - ordenados por `name`; `selectTriggers` - ordenados por `description`; `selectGraphs` - ordenados por `name`; `selectDashboards` - ordenados por `name`.
sortfield	string/array	Ordena o resultado pelas propriedades fornecidas. Valores possíveis: `hostid`, `host`, `name`, `status`.
countOutput	boolean	Esses parâmetros são descritos no comentário de referência.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	
selectGroups (obsoleto)	query	Este parâmetro está obsoleto, por favor use `selectTemplateGroups` em seu lugar. Retorna os grupos de templates aos quais o template pertence na propriedade `groups`.

Valores de retorno

`(integer/array)` Retorna ou:

- uma matriz de objetos;
- contagem de objetos recuperados, se o `countOutput` parâmetro tiver sido usado.

Exemplos

Recuperando templates pelo nome

Recupere todos os dados sobre dois templates chamados ”Linux” e ”Windows”.

Requisição:

```
{
    "jsonrpc": "2.0",
```

```
    "method": "template.get",
    "params": {
        "output": "extend",
        "filter": {
            "host": [
                "Linux",
                "Windows"
            ]
        }
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "proxyid": "0",
            "host": "Linux",
            "status": "3",
            "disable_until": "0",
            "error": "",
            "available": "0",
            "errors_from": "0",
            "lastaccess": "0",
            "ipmi_authtype": "0",
            "ipmi_privilege": "2",
            "ipmi_username": "",
            "ipmi_password": "",
            "ipmi_disable_until": "0",
            "ipmi_available": "0",
            "snmp_disable_until": "0",
            "snmp_available": "0",
            "maintenanceid": "0",
            "maintenance_status": "0",
            "maintenance_type": "0",
            "maintenance_from": "0",
            "ipmi_errors_from": "0",
            "snmp_errors_from": "0",
            "ipmi_error": "",
            "snmp_error": "",
            "jmx_disable_until": "0",
            "jmx_available": "0",
            "jmx_errors_from": "0",
            "jmx_error": "",
            "name": "Linux",
            "flags": "0",
            "templateid": "10001",
            "description": "",
            "tls_connect": "1",
            "tls_accept": "1",
            "tls_issuer": "",
            "tls_subject": "",
            "tls_psk_identity": "",
            "tls_psk": "",
            "uuid": "282ffe33afc74cccaf1524d9aa9dc502"
        },
        {
            "proxyid": "0",
            "host": "Windows",
            "status": "3",
```

```
            "disable_until": "0",
            "error": "",
            "available": "0",
            "errors_from": "0",
            "lastaccess": "0",
            "ipmi_authtype": "0",
            "ipmi_privilege": "2",
            "ipmi_username": "",
            "ipmi_password": "",
            "ipmi_disable_until": "0",
            "ipmi_available": "0",
            "snmp_disable_until": "0",
            "snmp_available": "0",
            "maintenanceid": "0",
            "maintenance_status": "0",
            "maintenance_type": "0",
            "maintenance_from": "0",
            "ipmi_errors_from": "0",
            "snmp_errors_from": "0",
            "ipmi_error": "",
            "snmp_error": "",
            "jmx_disable_until": "0",
            "jmx_available": "0",
            "jmx_errors_from": "0",
            "jmx_error": "",
            "name": "Windows",
            "flags": "0",
            "templateid": "10081",
            "description": "",
            "tls_connect": "1",
            "tls_accept": "1",
            "tls_issuer": "",
            "tls_subject": "",
            "tls_psk_identity": "",
            "tls_psk": "",
            "uuid": "522d17e1834049be879287b7c0518e5d"
        }
    ],
    "id": 1
}
```


Recuperando grupos de templates

Recupere os grupos de templates dos quais o template ”Linux by Zabbix agent” é membro.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "template.get",
    "params": {
        "output": ["hostid"],
        "selectTemplateGroups": "extend",
        "filter": {
            "host": [
                "Linux by Zabbix agent"
            ]
        }
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "templateid": "10001",
            "templategroups": [
                {
                    "groupid": "10",
                    "name": "Templates/Operating systems",
                    "uuid": "846977d1dfed4968bc5f8bdb363285bc"
                }
            ]
        }
    ],
    "id": 1
}
```

Recuperando hosts por template

Recupere hosts que tenham o template ”10001” (*Linux por Zabbix agent*) vinculado a eles.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "template.get",
    "params": {
        "output": "templateid",
        "templateids": "10001",
        "selectHosts": ["hostid", "name"]
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "templateid": "10001",
            "hosts": [
                {
                    "hostid": "10084",
                    "name": "Zabbix server"
                },
                {
                    "hostid": "10603",
                    "name": "Host 1"
                },
                {
                    "hostid": "10604",
                    "name": "Host 2"
                }
            ]
        }
    ],
    "id": 1
}
```

Pesquisando por tags de template

Recupere templates que possuem a tag ”host-name” igual a ”{HOST.NAME}”.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "template.get",
    "params": {
        "output": ["hostid"],
        "selectTags": "extend",
        "evaltype": 0,
        "tags": [
            {
                "tag": "host-name",
                "value": "{HOST.NAME}",
                "operator": 1
            }
        ]
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "hostid": "10402",
            "tags": [
                {
                    "tag": "host-name",
                    "value": "{HOST.NAME}"
                }
            ]
        }
    ],
    "id": 1
}
```

Veja também

- Grupo de templates
- Template
- Macro de usuário
- Interface de host

Fonte

CTemplate::get() in *ui/include/classes/api/services/CTemplate.php*.

template.massadd

Descrição

`object template.massadd(object parameters)`

Este método permite adicionar simultaneamente vários objetos relacionados aos templates fornecidos.

> **Note:**
> Este método está disponível apenas para os tipos de usuário *Admin* e *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object)` Parâmetros contendo os IDs dos templates a serem atualizados e os objetos a serem adicionados aos templates.

O método aceita os seguintes parâmetros.

Parâmetro	Tipo	Descrição
templates	object/array	Templates a serem atualizados. Os templates devem ter apenas a propriedade `templateid` definida. Comportamento do parâmetro: - *obrigatório*
groups	object/array	Grupos de template para adicionar os templates fornecidos. Os grupos de template devem ter apenas a propriedade `groupid` definida.
macros	object/array	Macros de usuário a serem criadas para os templates fornecidos.
templates_link	object/array	Templates para vincular aos templates fornecidos. Os templates devem ter apenas a propriedade `templateid` definida.

Retorno de valores

`(object)` Retorna um objeto que contenha os IDs dos templates atualizados na propriedade `templateids`.

Exemplos

Vincular um grupo a templates

Adicione o grupo de template ”2” a dois templates.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "template.massadd",
    "params": {
        "templates": [
            {
                "templateid": "10085"
            },
            {
                "templateid": "10086"
            }
        ],
        "groups": [
            {
                "groupid": "2"
            }
        ]
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "10085",
            "10086"
        ]
    },
    "id": 1
}
```

Vincular dois templates a um template

Vincule os templates ”10106” e ”10104” ao template ”10073”.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "template.massadd",
    "params": {
        "templates": [
            {
                "templateid": "10073"
            }
        ],
        "templates_link": [
            {
                "templateid": "10106"
            },
            {
                "templateid": "10104"
            }
        ]
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "10073"
        ]
    },
    "id": 1
}
```

Veja também

- template.update
- Host
- Grupo de template
- Macro de usuário

Fonte

CTemplate::massAdd() in *ui/include/classes/api/services/CTemplate.php*.

template.massremove

Descrição

`object template.massremove(object parameters)`

Este método permite remover objetos relacionados de vários templates.

Note:
Este método está disponível apenas para os tipos de usuário *Admin* e *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função de usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object)` Parâmetros contendo os IDs dos templates a serem atualizados e os objetos que devem ser removidos.

Parâmetro	Tipo	Descrição
templateids	ID/array	IDs dos templates a serem atualizados. Comportamento do parâmetro: - *obrigatório*

Parâmetro	Tipo	Descrição
groupids	ID/array	IDs dos grupos de templates dos quais remover os templates fornecidos.
macros	string/array	IDs das macros de usuário a serem excluídas dos templates fornecidos.
templateids_clear	ID/array	IDs dos templates a serem desvinculados e limpos dos templates fornecidos (upstream).
templateids_link	ID/array	IDs dos templates a serem desvinculados dos templates fornecidos (upstream).

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos modelos atualizados na propriedade `templateids`.

Exemplos

Removendo templates de um grupo

Remove dois templates do grupo ”2”.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "template.massremove",
    "params": {
        "templateids": [
            "10085",
            "10086"
        ],
        "groupids": "2"
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "10085",
            "10086"
        ]
    },
    "id": 1
}
```

Desvinculando templates de um host

Desvincule os templates ”10106” e ”10104” do template ”10085”.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "template.massremove",
    "params": {
        "templateids": "10085",
        "templateids_link": [
            "10106",
            "10104"
        ]
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "10085"
        ]
    },
    "id": 1
}
```

Veja também

- template.update
- Macro de usuário

Fonte

CTemplate::massRemove() in *ui/include/classes/api/services/CTemplate.php*.

template.massupdate

Descrição

`object template.massupdate(object parameters)`

Este método permite substituir ou remover simultaneamente objetos relacionados e atualizar propriedades em vários templates.

> **Note:**
> Este método está disponível apenas para os tipos de usuário *Admin* e *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função de usuário. Consulte Funções de usuário para mais informações.

Parâmetros

`(object)` Parâmetros contendo os IDs dos templates a serem atualizados e os objetos a serem substituídos nos templates.

O método aceita os seguintes parâmetros.

Parâmetro	Tipo	Descrição
templates	object/array	Templates a serem atualizados. Os templates devem ter apenas a propriedade `templateid` definida. Comportamento do parâmetro: - *obrigatório*
groups	object/array	Grupos de template para substituir os grupos de template atuais aos quais os templates pertencem. Os grupos de template devem ter apenas a propriedade `groupid` definida.
macros	object/array	Macros de usuário para substituir todas as macros de usuário atuais nos templates fornecidos.
templates_clear	object/array	Templates para desvincular e limpar dos templates fornecidos. Os templates devem ter apenas a propriedade `templateid` definida.
templates_link	object/array	Templates para substituir os templates atualmente vinculados. Os templates devem ter apenas a propriedade `templateid` definida.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos modelos atualizados na propriedade `templateids`.

Exemplos

Desvinculando um template

Desvincule e limpe o template ”10091” dos templates fornecidos.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "template.massupdate",
    "params": {
        "templates": [
            {
                "templateid": "10085"
            },
            {
                "templateid": "10086"
            }
        ],
        "templates_clear": [
            {
                "templateid": "10091"
            }
        ]
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "10085",
            "10086"
        ]
    },
    "id": 1
}
```

Substituindo macros de usuário

Substitua todas as macros de usuário pela macro de usuário fornecida em vários templates.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "template.massupdate",
    "params": {
        "templates": [
            {
                "templateid": "10074"
            },
            {
                "templateid": "10075"
            },
            {
                "templateid": "10076"
            },
            {
                "templateid": "10077"
            }
        ],
        "macros": [
            {
                "macro": "{$AGENT.TIMEOUT}",
                "value": "5m",
                "description": "Timeout após o qual o agent é considerado indisponível. Funciona apenas pa
```

```
            }
        ]
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "10074",
            "10075",
            "10076",
            "10077"
        ]
    },
    "id": 1
}
```

Veja também

- template.update
- template.massadd
- Grupo de template
- Macro de usuário

Fonte

CTemplate::massUpdate() in *ui/include/classes/api/services/CTemplate.php*.

template.update

Descrição

`object template.update(object/array templates)`

Este método permite atualizar templates existentes.

> **Note:**
> Este método está disponível apenas para os tipos de usuário *Admin* e *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função de usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object/array)` Propriedades do template a serem atualizadas.

A propriedade `templateid` deve ser definida para cada template, todas as outras propriedades são opcionais. Somente as propriedades fornecidas serão atualizadas, todas as outras permanecerão inalteradas.

Além das propriedades padrão do template, o método aceita os seguintes parâmetros.

Parâmetro	Tipo	Descrição
groups	object/array	Grupos de templates para substituir os grupos de templates atuais aos quais os templates pertencem. Os grupos de templates devem ter apenas a propriedade `groupid` definida.
tags	object/array	Tags de template para substituir as tags de template atuais.
macros	object/array	Macros de usuário para substituir as macros de usuário atuais nos templates fornecidos.
templates	object/array	Templates para substituir os templates atualmente vinculados. Templates que não forem passados serão apenas desvinculados. Os templates devem ter apenas a propriedade `templateid` definida.

Parâmetro	Tipo	Descrição
templates_clear	object/array	Templates para desvincular e limpar dos templates fornecidos. Os templates devem ter apenas a propriedade `templateid` definida.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos templates atualizados na propriedade `templateids`.

Exemplos

Alterando as propriedades padrão do template

Altere o nome técnico do template para ”Linux by Zabbix agent Custom”, o nome visível para ”My template” e atualize a descrição do template.

Request:

```
{
    "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
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "10086"
        ]
    },
    "id": 1
}
```

Atualizando grupos de templates

Substitua todos os grupos de templates do template fornecido por um diferente.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "template.update",
    "params": {
        "templateid": "10086",
        "groups": [
            {
                "groupid": "24"
            }
        ]
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
```

```
            "10086"
        ]
    },
    "id": 1
}
```

Atualizando tags de template

Substitua todas as tags de template por uma diferente.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "template.update",
    "params": {
        "templateid": "10086",
        "tags": [
            {
                "tag": "host-name",
                "value": "{HOST.NAME}"
            }
        ]
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "10086"
        ]
    },
    "id": 1
}
```

Atualizando macros de template

Substitua todas as macros de template por uma diferente.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "template.update",
    "params": {
        "templateid": "10086",
        "macros": [
            {
                "macro": "{$MY_MACRO}",
                "value": "new_value"
            }
        ]
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "10086"
        ]
```

```
    },
    "id": 1
}
```

Atualizando os templates vinculados ao template

Desvincula (sem limpar) todos os templates do template fornecido e vincula um diferente a ele.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "template.update",
    "params": {
        "templateid": "10086",
        "templates": [
            {
                "templateid": "10087"
            }
        ]
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "10086"
        ]
    },
    "id": 1
}
```

Limpando templates vinculados ao template

Desvincula e limpa o template fornecido de um template vinculado específico.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "template.update",
    "params": {
        "templateid": "10086",
        "templates_clear": [
            {
                "templateid": "10087"
            }
        ]
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "10086"
        ]
    },
    "id": 1
}
```

Fonte

CTemplate::update() in *ui/include/classes/api/services/CTemplate.php*.

Template dashboard

Esta classe foi projetada para trabalhar com dashboards de template.

Referências de objetos:

- Dashboard de template
- Página do dashboard de template
  - Widget do dashboard de template
    - Campo do widget do dashboard de template

Métodos disponíveis:

- templatedashboard.create - criar novos dashboards de template
- templatedashboard.delete - excluir dashboards de template
- templatedashboard.get - recuperar dashboards de template
- templatedashboard.update - atualizar dashboards de template

Objeto de dashboard de template

Os seguintes objetos estão diretamente relacionados à API `templatedashboard`.

Dashboard de template

O objeto dashboard de template possui as seguintes propriedades.

Propriedade	Tipo	Descrição
dashboardid	ID	ID do dashboard de template. Comportamento da propriedade: - *somente leitura* - *obrigatório* para operações de atualização
name	string	Nome do dashboard de template. Comportamento da propriedade: - *obrigatório* para operações de criação
templateid	ID	ID do template ao qual o dashboard pertence. Comportamento da propriedade: - *constante* - *obrigatório* para operações de criação
display_period	integer	Período padrão de exibição da página (em segundos). Valores possíveis: 10, 30, 60, 120, 600, 1800, 3600. Padrão: 30.
auto_start	integer	Iniciar apresentação de slides automaticamente. Valores possíveis: 0 - não iniciar apresentação de slides automaticamente; 1 - *(padrão)* iniciar apresentação de slides automaticamente.
uuid	string	Identificador universal único, usado para vincular dashboards de template importados aos já existentes. Gerado automaticamente, se não fornecido.

Página do dashboard do template

O objeto de página do dashboard do template possui as seguintes propriedades.

Propriedade	Tipo	Descrição
dashboard_pageid	ID	ID da página do dashboard. Comportamento da propriedade: - *somente leitura*
name	string	Nome da página do dashboard. Padrão: string vazia.
display_period	integer	Período de exibição da página do dashboard (em segundos). Valores possíveis: 0, 10, 30, 60, 120, 600, 1800, 3600. Padrão: 0 (usará o período de exibição padrão da página).
widgets	array	Array de objetos widget do dashboard do template.

Widget de dashboard de template

O objeto widget de dashboard de template possui as seguintes propriedades.

Propriedade	Tipo	Descrição
widgetid	ID	ID do widget de dashboard. Comportamento da propriedade: - *somente leitura*
type	string	Tipo do widget de dashboard. Valores possíveis: actionlog - Log de ação; clock - Relógio; *(obsoleto)* dataover - Visão geral dos dados; discovery - Status de descoberta; favgraphs - Gráficos favoritos; favmaps - Mapas favoritos; gauge - Medidor; graph - Gráfico (clássico); graphprototype - Protótipo de gráfico; honeycomb - Favo de mel; hostavail - Disponibilidade do host; hostnavigator - Navegador de host; itemnavigator - Navegador de item; item - Valor do item; map - Mapa; navtree - Árvore de navegação de mapa; piechart - Gráfico de pizza; plaintext - Texto simples; problemhosts - Hosts com problemas; problems - Problemas; problemsbysv - Problemas por severidade; slareport - Relatório de SLA; svggraph - Gráfico; systeminfo - Informações do sistema; tophosts - Principais hosts; toptriggers - Principais triggers; trigover - Visão geral de triggers; url - URL; web - Monitoramento web. Comportamento da propriedade: - *obrigatório*
name	string	Nome personalizado do widget.

Propriedade	Tipo	Descrição
x	integer	Posição horizontal a partir do lado esquerdo do dashboard. Os valores possíveis variam de 0 a 71.
y	integer	Posição vertical a partir do topo do dashboard. Os valores possíveis variam de 0 a 63.
width	integer	A largura do widget. Os valores possíveis variam de 1 a 72.
height	integer	A altura do widget. Os valores possíveis variam de 1 a 64.
view_mode	integer	O modo de visualização do widget. Valores possíveis: 0 - *(padrão)* visualização padrão do widget; 1 - com cabeçalho oculto;
fields	array	Array de objetos campo de widget de dashboard de template.

Campo de widget de dashboard de template

O objeto de campo de widget de dashboard de template possui as seguintes propriedades.

Propriedade	Tipo	Descrição
type	integer	Tipo do campo do widget. Valores possíveis: 0 - Inteiro; 1 - String; 4 - Item; 5 - Protótipo de item; 6 - Gráfico; 7 - Protótipo de gráfico; 8 - Mapa; 9 - Serviço; 10 - SLA; 11 - Usuário; 12 - Ação; 13 - Tipo de mídia. Comportamento da propriedade: - *obrigatório*
name	string	Nome do campo do widget. Valores possíveis: veja Campos de widget de dashboard. Observe que alguns parâmetros relacionados ao host (por exemplo, *Grupos de hosts*, *Excluir grupos de hosts* e *Hosts* no widget *Problemas*, *Grupos de hosts* no widget *Disponibilidade do host*, etc.) não estão disponíveis ao configurar o widget em um dashboard de template. Isso ocorre porque dashboards de template exibem dados apenas do host ao qual o template está vinculado. Comportamento da propriedade: - *obrigatório*

Propriedade	Tipo	Descrição
value	mixed	Valor do campo do widget dependendo do tipo. Valores possíveis: veja Campos de widget de dashboard. Observe que alguns parâmetros relacionados ao host (por exemplo, *Grupos de hosts*, *Excluir grupos de hosts* e *Hosts* no widget *Problemas*, *Grupos de hosts* no widget *Disponibilidade do host*, etc.) não estão disponíveis ao configurar o widget em um dashboard de template. Isso ocorre porque dashboards de template exibem dados apenas do host ao qual o template está vinculado. Comportamento da propriedade: - *obrigatório*

templatedashboard.create

Descrição

`object templatedashboard.create(object/array templateDashboards)`

Este método permite criar novos dashboards de template.

Note:
Este método está disponível apenas para os tipos de usuário *Admin* e *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object/array)` Dashboards de template a serem criados.

Além das propriedades padrão do dashboard de template, o método aceita os seguintes parâmetros.

Parâmetro	Tipo	Descrição
pages	array	Páginas do dashboard de template a serem criadas para o dashboard. As páginas do dashboard serão ordenadas na mesma ordem especificada. Comportamento do parâmetro: - *obrigatório*

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos painéis de modelo criados na propriedade `dashboardids`. A ordem dos IDs retornados corresponde à ordem dos painéis de modelo passados

Exemplos

Criando um dashboard de template

Crie um dashboard de template chamado ”Graphs” com um widget de Gráfico em uma única página do dashboard.

Requisição:

```
{
    "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
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "32"
        ]
    },
    "id": 1
}
```

Veja também

- Página do dashboard do template
- Widget do dashboard do template
- Campo do widget do dashboard do template

Fonte

CTemplateDashboard::create() in *ui/include/classes/api/services/CTemplateDashboard.php*.

templatedashboard.delete

Descrição

`object templatedashboard.delete(array templateDashboardIds)`

Este método permite excluir dashboards de template.

> **Note:**
> Este método está disponível apenas para os tipos de usuário *Admin* e *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função de usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(array)` IDs de templates de dashboard para deletar.

Retorno de valores

`(object)` Retorna um objeto contendo os IDs dos templates de dashboards deletados na propriedade `dashboardids`.

Exemplos

Excluindo vários painéis de template

Exclua dois painéis de template.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "templatedashboard.delete",
    "params": [
        "45",
        "46"
    ],
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "45",
            "46"
        ]
    },
    "id": 1
}
```

Fonte

CTemplateDashboard::delete() in *ui/include/classes/api/services/CTemplateDashboard.php*.

templatedashboard.get

Descrição

`integer/array templatedashboard.get(object parameters)`

O método permite recuperar dashboards de template de acordo com os parâmetros fornecidos.

Note:
Este método está disponível para usuários de qualquer tipo. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object)` Parâmetros que definem a saída desejada.

O método suporta os seguintes parâmetros.

Parâmetro	Tipo	Descrição
dashboardids	ID/array	Retorna apenas dashboards de template com os IDs fornecidos.
templateids	ID/array	Retorna apenas dashboards de template que pertencem aos templates fornecidos.
selectPages	query	Retorna uma propriedade `pages` com as páginas do dashboard de template, corretamente ordenadas.
sortfield	string/array	Ordena o resultado pelas propriedades fornecidas. Valores possíveis: `dashboardid`, `name`.
countOutput	boolean	Esses parâmetros são descritos no comentário de referência.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	

Parâmetro	Tipo	Descrição
sortorder	string/array	
startSearch	boolean	

Valores de retorno

`(inteiro/array)` Retorna:

- uma matriz de objetos;
- a contagem de objetos recuperados, se o parâmetro `countOutput` tiver sido usado.

Exemplos

Recuperando dashboards de template

Recupera todos os dashboards de template com widgets para um template especificado.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "templatedashboard.get",
    "params": {
        "output": "extend",
        "selectPages": "extend",
        "templateids": "10001"
    },
    "id": 1
}
```

Resposta:

```
{
    "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
}
```

Veja também

- Página do dashboard do template
- Widget do dashboard do template
- Campo do widget do dashboard do template

Fonte

CTemplateDashboard::get() in *ui/include/classes/api/services/CTemplateDashboard.php*.

templatedashboard.update

Descrição

`object templatedashboard.update(object/array templateDashboards)`

Este método permite atualizar dashboards de template existentes.

> **Note:**
> Este método está disponível apenas para os tipos de usuário *Admin* e *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Consulte Funções de usuário para mais informações.

Parâmetros

`(object/array)` Propriedades do dashboard do template a serem atualizadas.

A propriedade `dashboardid` deve ser especificada para cada dashboard, todas as outras propriedades são opcionais. Somente as propriedades especificadas serão atualizadas.

Além das propriedades padrão do dashboard do template, o método aceita os seguintes parâmetros.

Parâmetro	Tipo	Descrição
pages	array	Páginas do dashboard do template para substituir as páginas do dashboard existentes. As páginas do dashboard são atualizadas pela propriedade `dashboard_pageid`. Novas páginas do dashboard serão criadas para objetos sem a propriedade `dashboard_pageid` e as páginas do dashboard existentes serão excluídas se não forem reutilizadas. As páginas do dashboard serão ordenadas na mesma ordem especificada. Somente as propriedades especificadas das páginas do dashboard serão atualizadas. Pelo menos um objeto de página do dashboard é obrigatório para a propriedade `pages`.

Retorno de valores

`(object)` Retorna um objeto contendo os IDs dos templates de dashboards atualizados na propriedade `dashboardids`.

Exemplos

Renomeando um dashboard de template

Renomeie um dashboard de template para ”Performance graphs”.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "templatedashboard.update",
    "params": {
        "dashboardid": "23",
        "name": "Performance graphs"
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "23"
        ]
    },
    "id": 1
}
```

Atualizando páginas de dashboard de template

Renomeie a primeira página do dashboard, substitua os widgets na segunda página do dashboard e adicione uma nova página como a terceira. Exclua todas as outras páginas do dashboard.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "templatedashboard.update",
    "params": {
        "dashboardid": "2",
        "pages": [
            {
                "dashboard_pageid": 1,
                "name": "Renamed Page"
            },
            {
                "dashboard_pageid": 2,
                "widgets": [
                    {
                        "type": "clock",
                        "x": 0,
                        "y": 0,
                        "width": 12,
                        "height": 3
                    }
                ]
            },
            {
                "display_period": 60
            }
        ]
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
```

```
            "2"
        ]
    },
    "id": 1
}
```

Veja também

- Widget de dashboard de template
- Campo de widget de dashboard de template

Source

CTemplateDashboard::update() in *ui/include/classes/api/services/CTemplateDashboard.php*.

Tipo de mídia

Esta classe é projetada para trabalhar com tipos de mídia.

Referências de objetos:

- Tipo de mídia
  - Parâmetros do webhook
  - Parâmetros do script
- Modelo de mensagem

Métodos disponíveis:

- mediatype.create - criar novos tipos de mídia
- mediatype.delete - excluir tipos de mídia
- mediatype.get - recuperar tipos de mídia
- mediatype.update - atualizar tipos de mídia

Objeto media type

Os seguintes objetos estão diretamente relacionados à API `mediatype`.

Tipo de mídia

O objeto tipo de mídia possui as seguintes propriedades.

Propriedade	Tipo	Descrição
mediatypeid	ID	ID do tipo de mídia. Comportamento da propriedade: - *somente leitura* - *obrigatório* para operações de atualização
name	string	Nome do tipo de mídia. Comportamento da propriedade: - *obrigatório* para operações de criação
type	integer	Transporte utilizado pelo tipo de mídia. Valores possíveis: 0 - E-mail; 1 - Script; 2 - SMS; 4 - Webhook. Comportamento da propriedade: - *obrigatório* para operações de criação

Propriedade	Tipo	Descrição
exec_path	string	Nome do arquivo de script (por exemplo, notification.sh) que está localizado no diretório especificado no parâmetro de configuração do server AlertScriptsPath. Comportamento da propriedade: - *obrigatório* se `type` estiver definido como ”Script”
gsm_modem	string	Nome do dispositivo serial do modem GSM. Comportamento da propriedade: - *obrigatório* se `type` estiver definido como ”SMS”
passwd	string	Senha de autenticação. Comportamento da propriedade: - *suportado* se `smtp_authentication` estiver definido como ”Senha normal”
provider	integer	Provedor de e-mail. Valores possíveis: 0 - *(padrão)* SMTP genérico; 1 - Gmail; 2 - Gmail relay; 3 - Office365; 4 - Office365 relay.
smtp_email	string	Endereço de e-mail a partir do qual as notificações serão enviadas. Comportamento da propriedade: - *obrigatório* se `type` estiver definido como ”E-mail”
smtp_helo	string	SMTP HELO. Comportamento da propriedade: - *suportado* se `type` estiver definido como ”E-mail”
smtp_server	string	Servidor SMTP. Comportamento da propriedade: - *obrigatório* se `type` estiver definido como ”E-mail”
smtp_port	integer	Porta do servidor SMTP para conectar. Padrão: 25. Comportamento da propriedade: - *suportado* se `type` estiver definido como ”E-mail”
smtp_security	integer	Nível de segurança da conexão SMTP a ser usado. Valores possíveis: 0 - *(padrão)* Nenhum; 1 - STARTTLS; 2 - SSL/TLS. Comportamento da propriedade: - *suportado* se `type` estiver definido como ”E-mail”
smtp_verify_host	integer	Verificar host SSL para SMTP. Valores possíveis: 0 - *(padrão)* Não; 1 - Sim. Comportamento da propriedade: - *suportado* se `smtp_security` estiver definido como ”STARTTLS” ou ”SSL/TLS”

Propriedade	Tipo	Descrição
smtp_verify_peer	integer	Verificar peer SSL para SMTP. Valores possíveis: 0 - *(padrão)* Não; 1 - Sim. Comportamento da propriedade: - *suportado* se `smtp_security` estiver definido como ”STARTTLS” ou ”SSL/TLS”
smtp_authentication	integer	Método de autenticação SMTP a ser usado. Valores possíveis: 0 - *(padrão)* Nenhum; 1 - Senha normal. Comportamento da propriedade: - *suportado* se `type` estiver definido como ”E-mail”
status	integer	Se o tipo de mídia está habilitado. Valores possíveis: 0 - *(padrão)* Habilitado; 1 - Desabilitado.
username	string	Nome de usuário. Comportamento da propriedade: - *suportado* se `smtp_authentication` estiver definido como ”Senha normal”
maxsessions	integer	O número máximo de alertas que podem ser processados em paralelo. Valores possíveis se `type` estiver definido como ”SMS”: 1. Valores possíveis se `type` estiver definido como ”E-mail”, ”Script” ou ”Webhook”: 0-100. Padrão: 1.
maxattempts	integer	O número máximo de tentativas para enviar um alerta. Valores possíveis: 1-100. Padrão: 3.
attempt_interval	string	O intervalo entre as tentativas de repetição. Aceita segundos e unidade de tempo com sufixo. Valores possíveis: 0-1h. Padrão: 10s.
content_type (obsoleto)	integer	Este parâmetro está obsoleto, por favor use `message_format` em seu lugar. Formato da mensagem. Valores possíveis: 0 - Texto simples; 1 - *(padrão)* HTML. Comportamento da propriedade: - *suportado* se `type` estiver definido como ”E-mail”

Propriedade	Tipo	Descrição
message_format	integer	Formato da mensagem. Valores possíveis: 0 - Texto simples; 1 - *(padrão)* HTML. Comportamento da propriedade: - *suportado* se `type` estiver definido como ”E-mail”
script	text	Corpo do script do webhook (JavaScript). Comportamento da propriedade: - *obrigatório* se `type` estiver definido como ”Webhook”
timeout	string	Timeout do script do webhook. Aceita segundos e unidade de tempo com sufixo. Valores possíveis: 1-60s. Padrão: 30s. Comportamento da propriedade: - *suportado* se `type` estiver definido como ”Webhook”
process_tags	integer	Processar valores de propriedades JSON na resposta do script do webhook como tags. Essas tags são adicionadas a quaisquer tags de problema existentes. Valores possíveis: 0 - *(padrão)* Ignorar resposta do script do webhook; 1 - Processar resposta do script do webhook como tags. Comportamento da propriedade: - *suportado* se `type` estiver definido como ”Webhook”
show_event_menu	integer	Incluir uma entrada no menu de evento que vincula a uma URL personalizada. Também adiciona a propriedade `urls` à saída de problem.get e event.get. Valores possíveis: 0 - *(padrão)* Não incluir entrada de menu de evento ou propriedade `urls`; 1 - Incluir entrada de menu de evento e propriedade `urls`. Comportamento da propriedade: - *suportado* se `type` estiver definido como ”Webhook”
event_menu_url	string	URL usada na entrada do menu de evento e na propriedade `urls` retornada por problem.get e event.get. Comportamento da propriedade: - *suportado* se `show_event_menu` estiver definido como ”Incluir entrada de menu de evento e propriedade `urls`”
event_menu_name	string	Nome usado para a entrada do menu de evento e na propriedade `urls` retornada por problem.get e event.get. Comportamento da propriedade: - *suportado* se `show_event_menu` estiver definido como ”Incluir entrada de menu de evento e propriedade `urls`”
parameters	array	Parâmetros de webhook ou script. Comportamento da propriedade: - *suportado* se `type` estiver definido como ”Webhook” ou ”Script”
description	text	Descrição do tipo de mídia.

Parâmetros do webhook

Os parâmetros do webhook possuem as seguintes propriedades.

Propriedade	Tipo	Descrição
name	string	Nome do parâmetro. Comportamento da propriedade: - *obrigatório*
value	string	Valor do parâmetro, suporta macros. As macros suportadas estão descritas na página Macros suportadas.

Parâmetros do script

Os parâmetros do script possuem as seguintes propriedades.

Propriedade	Tipo	Descrição
sortorder	integer	A ordem em que os valores dos parâmetros serão passados para o script como argumentos de linha de comando, começando com 0 como o primeiro. Comportamento da propriedade: - *obrigatório*
value	string	Valor do parâmetro, suporta macros. As macros suportadas estão descritas na página Macros suportadas.

Modelo de mensagem

O objeto de modelo de mensagem define um modelo que será usado como uma mensagem padrão para operações de ação para enviar uma notificação. Ele possui as seguintes propriedades.

Propriedade	Tipo	Descrição
eventsource	integer	Fonte do evento. Valores possíveis: 0 - Triggers; 1 - Descoberta; 2 - Autoregistro; 3 - Interno; 4 - Serviços. Comportamento da propriedade: - *obrigatório*
recovery	integer	Modo de operação. Valores possíveis: 0 - Operações; 1 - Operações de recuperação; 2 - Operações de atualização. Comportamento da propriedade: - *obrigatório*
subject	string	Assunto da mensagem.
message	string	Texto da mensagem.

mediatype.create

Descrição

`object mediatype.create(object/array mediaTypes)`

Este método permite criar novos tipos de mídia.

Note:
Este método está disponível apenas para o tipo de usuário *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object/array)` Tipos de mídia a serem criados.

Além das propriedades padrão do tipo de mídia, o método aceita os seguintes parâmetros.

Parâmetro	Tipo	Descrição
message_templates	array	Modelos de mensagem a serem criados para o tipo de mídia.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos tipos de mídia criados na propriedade `mediatypeids`. A ordem dos IDs retornados corresponde à ordem dos tipos de mídia passados

Exemplos

Criando um tipo de mídia de e-mail

Crie um novo tipo de mídia de e-mail com uma porta SMTP personalizada e modelos de mensagem.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "mediatype.create",
    "params": {
        "type": "0",
        "name": "Email",
        "smtp_server": "mail.example.com",
        "smtp_helo": "example.com",
        "smtp_email": "zabbix@example.com",
        "smtp_port": "587",
        "message_format": "1",
        "message_templates": [
            {
                "eventsource": "0",
                "recovery": "0",
                "subject": "Problem: {EVENT.NAME}",
                "message": "Problem \"{EVENT.NAME}\" on host \"{HOST.NAME}\" started at {EVENT.TIME}."
            },
            {
                "eventsource": "0",
                "recovery": "1",
                "subject": "Resolved in {EVENT.DURATION}: {EVENT.NAME}",
                "message": "Problem \"{EVENT.NAME}\" on host \"{HOST.NAME}\" has been resolved at {EVENT.R
            },
            {
                "eventsource": "0",
                "recovery": "2",
                "subject": "Updated problem in {EVENT.AGE}: {EVENT.NAME}",
                "message": "{USER.FULLNAME} {EVENT.UPDATE.ACTION} problem \"{EVENT.NAME}\" on host \"{HOST
            }
        ]
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
```

```
        "mediatypeids": [
            "7"
        ]
    },
    "id": 1
}
```

Criando um tipo de mídia de script

Crie um novo tipo de mídia de script com um valor personalizado para o número de tentativas e o intervalo entre elas.

Requisição:

```
{
    "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
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "mediatypeids": [
            "8"
        ]
    },
    "id": 1
}
```

Criando um tipo de mídia webhook

Crie um novo tipo de mídia webhook.

Requisição:

```
{
    "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
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "mediatypeids": [
            "9"
        ]
    },
    "id": 1
}
```

Fonte

CMediaType::create() in *ui/include/classes/api/services/CMediaType.php*.

mediatype.delete

Descrição

`object mediatype.delete(array mediaTypeIds)`

Este método permite excluir tipos de mídia.

> **Note:**
> Este método está disponível apenas para o tipo de usuário *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(array)` IDs dos tipos de mídia a serem excluídos.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos tipos de mídia excluídos na propriedade `mediatypeids`.

Exemplos

Excluindo múltiplos tipos de mídia

Exclua dois tipos de mídia.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "mediatype.delete",
    "params": [
```

```
            "3",
            "5"
        ],
        "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "mediatypeids": [
            "3",
            "5"
        ]
    },
    "id": 1
}
```

Fonte

CMediaType::delete() in *ui/include/classes/api/services/CMediaType.php*.

mediatype.get

Descrição

`integer/array mediatype.get(object parameters)`

O método permite recuperar tipos de mídia de acordo com os parâmetros fornecidos.

Note:
Este método está disponível para usuários de qualquer tipo (para usuários do tipo *Usuário* desde o Zabbix 7.0.6). As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object)` Parâmetros que definem a saída desejada.

O método suporta os seguintes parâmetros.

Note:
Ao solicitar informações relacionadas ao usuário dos tipos de mídia, usuários do tipo *Admin* (desde o Zabbix 7.0.4) e usuários do tipo *User* (desde o Zabbix 7.0.9) podem recuperar apenas dados sobre seu próprio usuário. Para um exemplo, consulte Recuperando tipos de mídia como *Admin*.

Parâmetro	Tipo	Descrição
mediatypeids	ID/array	Retorna apenas os tipos de mídia com os IDs fornecidos.
mediaids	ID/array	Retorna apenas os tipos de mídia usados pela mídia fornecida.
userids	ID/array	Retorna apenas os tipos de mídia usados pelos usuários fornecidos.
selectActions	query	Retorna uma propriedade `actions` com as ações que usam o tipo de mídia.
selectMessageTemplates	query	Retorna uma propriedade `message_templates` com um array de mensagens do tipo de mídia. Comportamento do parâmetro: - *suportado* para usuários do tipo *Super admin* (desde o Zabbix 7.0.4)
selectUsers	query	Retorna uma propriedade `users` com os usuários que usam o tipo de mídia. Consulte `user.get` para restrições baseadas no tipo de usuário.

Parâmetro	Tipo	Descrição
sortfield	string/array	Classifica o resultado pelas propriedades fornecidas. Valores possíveis: `mediatypeid`.
filter	object	Retorna apenas os resultados que correspondem exatamente ao filtro fornecido. Aceita um objeto, onde as chaves são nomes de propriedades e os valores são um único valor ou um array de valores para comparar. Não suporta propriedades do tipo de dado `text` data type.
output	query	Possíveis propriedades do objeto Media type para usuários do tipo *Admin* (desde o Zabbix 7.0.4) e usuários do tipo *User* (desde o Zabbix 7.0.6): `mediatypeid`, `name`, `type`, `status`, `maxattempts`. Propriedades do objeto Media type a serem retornadas. Usuários do tipo *Admin* (desde o Zabbix 7.0.4) e usuários do tipo *User* (desde o Zabbix 7.0.6) podem recuperar apenas as seguintes propriedades: `mediatypeid`, `name`, `type`, `status`, `maxattempts`, `description`. Para um exemplo, consulte Recuperando tipos de mídia como *Admin*. Padrão: `extend`.
search	object	Retorna resultados que correspondam ao padrão fornecido (case-insensitive). Aceita um objeto, onde as chaves são nomes de propriedades e os valores são strings a serem pesquisadas. Se nenhuma opção adicional for fornecida, isso executará uma pesquisa `LIKE "%…%"`. Suporta apenas propriedades dos tipos de dado `string` e `text` data type. Possíveis propriedades do objeto Media type para usuários do tipo *Admin* (desde o Zabbix 7.0.4) e usuários do tipo *User* (desde o Zabbix 7.0.6): `name`, `description`.
countOutput	boolean	Esses parâmetros são descritos no comentário de referência.
editable	boolean	
excludeSearch	boolean	
limit	integer	
preservekeys	boolean	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valores de retorno

`(integer/array)` Retorna:

- uma matriz de objetos;
- a contagem de objetos recuperados, se o parâmetro `countOutput` foi usado.

Exemplos

Recuperando tipos de mídia

Recupere todos os tipos de mídia configurados. O exemplo a seguir retorna dois tipos de mídia:

- tipo de mídia de e-mail;
- tipo de mídia SMS.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "mediatype.get",
    "params": {
        "output": "extend",
        "selectMessageTemplates": "extend"
    },
    "id": 1
}
```

Resposta:

```
{
    "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
}
```

Recuperando tipos de mídia como *Admin*

Como um usuário do tipo *Admin*, recupere todos os tipos de mídia que estão habilitados, com os usuários que utilizam esses tipos de mídia. O exemplo a seguir retorna dois tipos de mídia:

- tipo de mídia de e-mail com um usuário (desde o Zabbix 7.0.4, apenas o próprio usuário do tipo *Admin*);
- tipo de mídia SMS sem usuários.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "mediatype.get",
    "params": {
        "output": "extend",
        "filter": {
            "status": 0
        },
        "selectUsers": "extend"
    },
    "id": 1
}
```

Resposta:

```
{
    "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
}
```


Recuperar tipos de mídia de script e webhook

O exemplo a seguir retorna três tipos de mídia:

- tipo de mídia de script com parâmetros;
- tipo de mídia de script sem parâmetros;
- tipo de mídia webhook com parâmetros.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "mediatype.get",
    "params": {
        "output": ["mediatypeid", "name", "parameters"],
        "filter": {
            "type": [1, 4]
        }
    },
    "id": 1
}
```

Resposta:


```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "mediatypeid": "10",
            "name": "Script with parameters",
            "parameters": [
                {
                    "sortorder": "0",
                    "value": "{ALERT.SENDTO}"
                },
                {
                    "sortorder": "1",
                    "value": "{EVENT.NAME}"
                },
                {
                    "sortorder": "2",
                    "value": "{ALERT.MESSAGE}"
```

```
                },
                {
                    "sortorder": "3",
                    "value": "Zabbix alert"
                }
            ]
        },
        {
            "mediatypeid": "13",
            "name": "Script without parameters",
            "parameters": []
        },
        {
            "mediatypeid": "11",
            "name": "Webhook",
            "parameters": [
                {
                    "name": "alert_message",
                    "value": "{ALERT.MESSAGE}"
                },
                {
                    "name": "event_update_message",
                    "value": "{EVENT.UPDATE.MESSAGE}"
                },
                {
                    "name": "host_name",
                    "value": "{HOST.NAME}"
                },
                {
                    "name": "trigger_description",
                    "value": "{TRIGGER.DESCRIPTION}"
                },
                {
                    "name": "trigger_id",
                    "value": "{TRIGGER.ID}"
                },
                {
                    "name": "alert_source",
                    "value": "Zabbix"
                }
            ]
        }
    ],
    "id": 1
}
```


Veja também

- User

Fonte

CMediaType::get() in *ui/include/classes/api/services/CMediaType.php*.

mediatype.update

Descrição

`object mediatype.update(object/array mediaTypes)`

Este método permite atualizar tipos de mídia existentes.

Note:
Este método está disponível apenas para o tipo de usuário *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função de usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object/array)` Propriedades do tipo de mídia a serem atualizadas.

A propriedade `mediatypeid` deve ser definida para cada tipo de mídia, todas as outras propriedades são opcionais. Somente as propriedades passadas serão atualizadas, todas as outras permanecerão inalteradas.

Além das propriedades padrão do tipo de mídia, o método aceita os seguintes parâmetros.

Parâmetro	Tipo	Descrição
message_templates	array	Modelos de mensagem para substituir os modelos de mensagem atuais.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos tipos de mídia atualizados na propriedade `mediatypeids`.

Exemplos

Habilitando um tipo de mídia

Habilite um tipo de mídia, ou seja, defina seu status como "0".

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "mediatype.update",
    "params": {
        "mediatypeid": "6",
        "status": "0"
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "mediatypeids": [
            "6"
        ]
    },
    "id": 1
}
```

Fonte

CMediaType::update() in *ui/include/classes/api/services/CMediaType.php*.

Token

Esta classe foi projetada para trabalhar com tokens.

Referências de objetos:

- Token

Métodos disponíveis:

- token.create - criar novos tokens
- token.delete - excluir tokens
- token.get - recuperar tokens

- token.update - atualizar tokens
- token.generate - gerar tokens

Objeto token

Os seguintes objetos estão diretamente relacionados à API `token`.

Token

O objeto token possui as seguintes propriedades.

Propriedade	Tipo	Descrição
tokenid	ID	ID do token. Comportamento da propriedade: - *somente leitura* - *obrigatório* para operações de atualização
name	string	Nome do token. Comportamento da propriedade: - *obrigatório* para operações de criação
description	text	Descrição do token.
userid	ID	ID do usuário ao qual o token foi atribuído. Padrão: *usuário atual*. Comportamento da propriedade: - *constante*
lastaccess	timestamp	Data e hora mais recente em que o token foi autenticado. ”0” se o token nunca foi autenticado. Comportamento da propriedade: - *somente leitura*
status	integer	Status do token. Valores possíveis: 0 - *(padrão)* token habilitado; 1 - token desabilitado.
expires_at	timestamp	Data e hora de expiração do token. ”0” para tokens que nunca expiram.
created_at	timestamp	Data e hora de criação do token. Comportamento da propriedade: - *somente leitura*
creator_userid	ID	ID do usuário que criou o token. Comportamento da propriedade: - *somente leitura*

token.create

Descrição

`object token.create(object/array tokens)`

Este método permite criar novos tokens.

> **Note:**
> A permissão *Gerenciar tokens de API* permission é necessária para a função de usuário gerenciar tokens de outros usuários.

Attention:
Um token criado por este método também precisa ser gerado antes de ser utilizável.

Parâmetros

`(objeto/array)` Tokens para criar.

O método aceita tokens com as propriedades de token padrão.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos tokens criados na propriedade `tokenids`. A ordem dos IDs retornados corresponde à ordem dos tokens passados.

Exemplos

Criar um token

Crie um token habilitado que nunca expira e autentica o usuário de ID 2.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "token.create",
    "params": {
        "name": "Your token",
        "userid": "2"
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "tokenids": [
            "188"
        ]
    },
    "id": 1
}
```

Crie um token desabilitado que expira em 21 de janeiro de 2021. Este token autenticará o usuário atual.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "token.create",
    "params": {
        "name": "Your token",
        "status": "1",
        "expires_at": "1611238072"
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "tokenids": [
            "189"
        ]
    },
    "id": 1
}
```

Fonte

CToken::create() in *ui/include/classes/api/services/CToken.php*.

token.delete

Descrição

`object token.delete(array tokenids)`

Este método permite excluir tokens.

> **Note:**
> A permissão *Gerenciar tokens de API* permission é necessária para que a função de usuário gerencie tokens de outros usuários.

Parâmetros

`(array)` IDs dos tokens a serem excluídos.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos tokens excluídos na propriedade `tokenids`.

Exemplos

Excluir vários tokens

Excluir dois tokens.

Request:

```
{
    "jsonrpc": "2.0",
    "method": "token.delete",
    "params": [
        "188",
        "192"
    ],
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result" (resultado): {
        "tokenids": [
            "188",
            "192"
        ]
    },
    "id": 1
}
```

Fonte

CToken::delete() in *ui/include/classes/api/services/CToken.php*.

token.generate

Descrição

`object token.generate(array tokenids)`

Este método permite gerar tokens.

Note:
A permissão *Gerenciar tokens de API* permission é necessária para que a função de usuário gerencie tokens de outros usuários.

Attention:
Um token pode ser gerado por este método somente se ele tiver sido criado.

Parâmetros

`(array)` IDs dos tokens a serem gerados.

Valores de retorno

`(array)` Retorna um array de objetos contendo o ID do token gerado na propriedade `tokenid` e a string de autorização gerada na propriedade `token`.

Propriedade	Tipo	Descrição
tokenid	ID	ID do token.
token	string	A string de autorização gerada para este token.

Exemplos

Gerar múltiplos tokens

Gere dois tokens.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "token.generate",
    "params": [
        "1",
        "2"
    ],
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "tokenid": "1",
            "token": "bbcfce79a2d95037502f7e9a534906d3466c9a1484beb6ea0f4e7be28e8b8ce2"
        },
        {
            "tokenid": "2",
            "token": "fa1258a83d518eabd87698a96bd7f07e5a6ae8aeb8463cae33d50b91dd21bd6d"
        }
    ],
    "id": 1
}
```

Fonte

CToken::generate() in *ui/include/classes/api/services/CToken.php*.

token.get

Descrição

`integer/array token.get(object parameters)`

O método permite recuperar tokens de acordo com os parâmetros fornecidos.

Note:
Apenas o tipo de usuário *Super admin* tem permissão para visualizar tokens para outros usuários.

Parâmetros

`(object)` Parâmetros que definem a saída desejada.

O método suporta os seguintes parâmetros.

Parâmetro	Tipo	Descrição
tokenids	ID/array	Retornar apenas tokens com os IDs fornecidos.
userids	ID/array	Retornar apenas tokens criados para os usuários fornecidos.
token	string	Retornar apenas tokens criados para o *Auth token* fornecido.
valid_at	timestamp	Retornar apenas tokens que são válidos (não expirados) na data e hora fornecidas.
expired_at	timestamp	Retornar apenas tokens que estão expirados (não válidos) na data e hora fornecidas.
sortfield	string/array	Ordenar o resultado pelas propriedades fornecidas. Valores possíveis: `tokenid`, `name`, `lastaccess`, `status`, `expires_at`, `created_at`.
countOutput	boolean	Esses parâmetros são descritos no comentário de referência.
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valores de retorno

`(integer/array)` Retorna:

- uma matriz de objetos;
- a contagem de objetos recuperados, se o parâmetro `countOutput` foi usado.

Exemplos

Recuperar um token

Recupere todos os dados do token com o ID "2".

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "token.get",
    "params": {
        "output": "extend",
        "tokenids": "2"
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "tokenid": "1",
```

```
            "name": "The Token",
            "description": "",
            "userid": "1",
            "lastaccess": "0",
            "status": "0",
            "expires_at": "1609406220",
            "created_at": "1611239454",
            "creator_userid": "1"
        }
    ],
    "id": 1
}
```


Fonte

CToken::get() in *ui/include/classes/api/services/CToken.php*.

token.update

Descrição

`object token.update(object/array tokens)`

Este método permite atualizar tokens existentes.

> **Note:**
> A permissão *Gerenciar tokens de API* permission é necessária para a função de usuário gerenciar tokens de outros usuários.

Parâmetros

`(object/array)` Propriedades do token a serem atualizadas.

A propriedade `tokenid` deve ser definida para cada token, todas as outras propriedades são opcionais. Apenas as propriedades passadas serão atualizadas, todas as outras permanecerão inalteradas.

O método aceita tokens com as propriedades de token padrão.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos tokens atualizados na propriedade `tokenids`.

Exemplos

Remover expiração do token

Remove a data de expiração do token.

Requisição:


```
{
    "jsonrpc": "2.0",
    "method": "token.update",
    "params": {
        "tokenid": "2",
        "expires_at": "0"
    },
    "id": 1
}
```


Resposta:


```
{
    "jsonrpc": "2.0",
    "result": {
        "tokenids": [
            "2"
        ]
    },
```

```
    "id": 1
}
```

Fonte

CToken::update() in *ui/include/classes/api/services/CToken.php*.

Trend

Esta classe foi projetada para trabalhar com dados de tendência.

Referências de objetos:

- Tendência de ponto flutuante
- Tendência de inteiro

Métodos disponíveis:

- trend.get - recuperar tendências

Objeto de tendência

Os seguintes objetos estão diretamente relacionados à API `trend`.

Note:
Os objetos de tendência diferem dependendo do tipo de informação do item. Eles são criados pelo Zabbix server e não podem ser modificados via API.

Tendência de ponto flutuante

O objeto de tendência de ponto flutuante possui as seguintes propriedades.

Propriedade	Tipo	Descrição
clock	timestamp	Timestamp de uma hora para a qual o valor foi calculado. Por exemplo, o timestamp ”04:00:00” significa valores calculados para o período ”04:00:00-04:59:59”.
itemid	ID	ID do item relacionado.
num	integer	Número de valores que estavam disponíveis para a hora.
value_min	float	Valor mínimo por hora.
value_avg	float	Valor médio por hora.
value_max	float	Valor máximo por hora.

Tendência de inteiro

O objeto de tendência de inteiro possui as seguintes propriedades.

Propriedade	Tipo	Descrição
clock	timestamp	Timestamp de uma hora para a qual o valor foi calculado. Por exemplo, o timestamp ”04:00:00” significa valores calculados para o período ”04:00:00-04:59:59”.
itemid	ID	ID do item relacionado.
num	integer	Número de valores que estavam disponíveis para a hora.
value_min	integer	Valor mínimo por hora.
value_avg	integer	Valor médio por hora.
value_max	integer	Valor máximo por hora.

trend.get

Descrição

`integer/array trend.get(object parameters)`

O método permite recuperar dados de tendência de acordo com os parâmetros fornecidos.

Note:
Este método está disponível para usuários de qualquer tipo. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Consulte Funções de usuário para mais informações.

Parâmetros

`(object)` Parâmetros que definem a saída desejada.

O método suporta os seguintes parâmetros.

Parâmetro	Tipo	Descrição
itemids	ID/array	Retorna apenas tendências com os IDs de item fornecidos.
time_from	timestamp	Retorna apenas valores que foram coletados após ou no horário fornecido.
time_till	timestamp	Retorna apenas valores que foram coletados antes ou no horário fornecido.
countOutput	boolean	Conta o número de objetos recuperados.
limit	integer	Limita a quantidade de objetos recuperados.
output	query	Define as propriedades do objeto Trend a serem retornadas.

Valores de retorno

`(integer/array)` Retorna:

- uma matriz de objetos;
- a contagem de objetos recuperados, se o parâmetro `countOutput` foi usado.

Exemplos

Recuperando dados de tendência do item

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "trend.get",
    "params": {
        "output": [
            "itemid",
            "clock",
            "num",
            "value_min",
            "value_avg",
            "value_max"
        ],
        "itemids": [
            "23715"
        ],
        "limit": "1"
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "itemid": "23715",
            "clock": "1446199200",
            "num": "60",
            "value_min": "0.165",
            "value_avg": "0.2168",
            "value_max": "0.35"
```

```
        }
    ],
    "id": 1
}
```

Fonte

CTrend::get() in *ui/include/classes/api/services/CTrend.php*.

Trigger

Esta classe foi projetada para trabalhar com triggers.

Referências de objetos:

- Trigger
- Tag de trigger

Métodos disponíveis:

- trigger.create - criar novas triggers
- trigger.delete - excluir triggers
- trigger.get - recuperar triggers
- trigger.update - atualizar triggers

Objeto trigger

Os seguintes objetos estão diretamente relacionados à API de `trigger`.

Trigger

O objeto trigger possui as seguintes propriedades.

Propriedade	Tipo	Descrição
triggerid	ID	ID do trigger. Comportamento da propriedade: - *somente leitura* - *obrigatório* para operações de atualização
description	string	Nome do trigger. Comportamento da propriedade: - *obrigatório* para operações de criação
expression	string	Expressão do trigger. Itens com `value_type` definido como 5 (binário) não podem ser usados em expressões de trigger. Comportamento da propriedade: - *obrigatório* para operações de criação
event_name	string	Nome do evento gerado pelo trigger.
opdata	string	Dados operacionais.
comments	string	Descrição adicional do trigger.
error	string	Texto do erro caso tenha havido algum problema ao atualizar o estado do trigger. Comportamento da propriedade: - *somente leitura*

Propriedade	Tipo	Descrição
flags	integer	Origem do trigger. Valores possíveis: 0 - *(padrão)* um trigger simples; 4 - um trigger descoberto. Comportamento da propriedade: - *somente leitura*
lastchange	timestamp	Hora em que o trigger mudou de estado pela última vez. Comportamento da propriedade: - *somente leitura*
priority	integer	Severidade do trigger. Valores possíveis: 0 - *(padrão)* não classificado; 1 - informação; 2 - aviso; 3 - média; 4 - alta; 5 - desastre.
state	integer	Estado do trigger. Valores possíveis: 0 - *(padrão)* o estado do trigger está atualizado; 1 - o estado atual do trigger é desconhecido. Comportamento da propriedade: - *somente leitura*
status	integer	Se o trigger está habilitado ou desabilitado. Valores possíveis: 0 - *(padrão)* habilitado; 1 - desabilitado.
templateid	ID	ID do trigger do template pai. Comportamento da propriedade: - *somente leitura*
type	integer	Se o trigger pode gerar múltiplos eventos de problema. Valores possíveis: 0 - *(padrão)* não gerar múltiplos eventos; 1 - gerar múltiplos eventos.
url	string	URL associada ao trigger.
url_name	string	Rótulo para a URL associada ao trigger.
value	integer	Se o trigger está em estado OK ou de problema. Valores possíveis: 0 - *(padrão)* OK; 1 - problema. Comportamento da propriedade: - *somente leitura*
recovery_mode	integer	Modo de geração de evento OK. Valores possíveis: 0 - *(padrão)* Expressão; 1 - Expressão de recuperação; 2 - Nenhum.

Propriedade	Tipo	Descrição
recovery_expression	string	Expressão de recuperação do trigger. Itens com `value_type` definido como 5 (binário) não podem ser usados em expressões de recuperação de trigger.
correlation_mode	integer	Fechamento de eventos OK. Valores possíveis: 0 - *(padrão)* Todos os problemas; 1 - Todos os problemas se os valores das tags coincidirem.
correlation_tag	string	Tag para correspondência.
manual_close	integer	Permitir fechamento manual. Valores possíveis: 0 - *(padrão)* Não; 1 - Sim.
uuid	string	Identificador único universal, usado para vincular triggers importados aos já existentes. Gerado automaticamente, se não fornecido. Comportamento da propriedade: - *suportado* se o trigger pertencer a um template

Tag de trigger

O objeto de tag de trigger possui as seguintes propriedades.

Propriedade	Tipo	Descrição
tag	string	Nome da tag de trigger. Comportamento da propriedade: - *obrigatório*
value	string	Valor da tag de trigger.

trigger.create

Descrição

`object trigger.create(object/array triggers)`

Este método permite criar novos triggers.

Note:
Este método está disponível apenas para os tipos de usuário *Admin* e *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object/array)` Triggers a serem criados.

Além das propriedades padrão do trigger, o método aceita os seguintes parâmetros.

Parâmetro	Tipo	Descrição
dependencies	array	Triggers das quais o trigger depende. Os triggers devem ter apenas a propriedade `triggerid` definida.
tags	array	Tags do trigger.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs das triggers criados na propriedade `triggerids`. A ordem dos IDs retornados corresponde à ordem das triggers passadas.

Exemplos

Criando um trigger

Crie dois triggers, cada um dependendo de outro trigger.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "trigger.create",
    "params": [
        {
            "description": "Processor load is too high on {HOST.NAME}",
            "expression": "last(/Linux server/system.cpu.load[percpu,avg1])>5",
            "dependencies": [
                {
                    "triggerid": "17367"
                }
            ]
        },
        {
            "description": "Service has stopped",
            "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
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "triggerids": [
            "17369",
            "17370"
        ]
    },
    "id": 1
}
```

Fonte

CTrigger::create() in *ui/include/classes/api/services/CTrigger.php*.

trigger.delete

Descrição

`object trigger.delete(array triggerIds)`

Este método permite excluir triggers.

> **Note:**
> Este método está disponível apenas para os tipos de usuário *Admin* e *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(array)` IDs das trigger a serem excluídas.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs das triggers excluídas na propriedade `triggerids`.

Exemplos

Excluir múltiplos triggers

Exclua dois triggers.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "trigger.delete",
    "params": [
        "12002",
        "12003"
    ],
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "triggerids": [
            "12002",
            "12003"
        ]
    },
    "id": 1
}
```

Fonte

CTrigger::delete() in *ui/include/classes/api/services/CTrigger.php*.

trigger.get

Descrição

`integer/array trigger.get(object parameters)`

O método permite recuperar triggers de acordo com os parâmetros fornecidos.

> **Note:**
> Este método está disponível para usuários de qualquer tipo. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Consulte Funções de usuário para mais informações.

Parâmetros

`(object)` Parâmetros que definem a saída desejada.

O método suporta os seguintes parâmetros.

Parâmetro	Tipo	Descrição
triggerids	ID/array	Retorna apenas triggers com os IDs fornecidos.

Parâmetro	Tipo	Descrição
groupids	ID/array	Retorna apenas triggers que pertencem a hosts ou templates dos grupos de hosts ou grupos de templates fornecidos.
templateids	ID/array	Retorna apenas triggers que pertencem aos templates fornecidos.
hostids	ID/array	Retorna apenas triggers que pertencem aos hosts fornecidos.
itemids	ID/array	Retorna apenas triggers que contêm os itens fornecidos.
functions	string/array	Retorna apenas triggers que usam as funções fornecidas. Consulte a página de funções suportadas para obter uma lista de funções suportadas.
group	string	Retorna apenas triggers que pertencem a hosts ou templates do grupo de hosts ou grupo de templates com o nome fornecido.
host	string	Retorna apenas triggers que pertencem ao host com o nome técnico fornecido.
inherited	boolean	Se definido como `true`, retorna apenas triggers herdados de um template.
templated	boolean	Se definido como `true`, retorna apenas triggers que pertencem a templates.
dependent	boolean	Se definido como `true`, retorna apenas triggers que possuem dependências. Se definido como `false`, retorna apenas triggers que não possuem dependências.
monitored	flag	Retorna apenas triggers habilitados que pertencem a hosts monitorados e contêm apenas itens habilitados.
active	flag	Retorna apenas triggers habilitados que pertencem a hosts monitorados.
maintenance	boolean	Se definido como `true`, retorna apenas triggers habilitados que pertencem a hosts em manutenção.
withUnacknowledgedEvents	flag	Retorna apenas triggers que possuem eventos não reconhecidos.
withAcknowledgedEvents	flag	Retorna apenas triggers com todos os eventos reconhecidos.
withLastEventUnacknowledged	flag	Retorna apenas triggers com o último evento não reconhecido.
skipDependent	flag	Ignora triggers em estado de problema que são dependentes de outros triggers. Observe que os outros triggers são ignorados se estiverem desabilitados, tiverem itens desabilitados ou hosts de itens desabilitados.
lastChangeSince	timestamp	Retorna apenas triggers que mudaram seu estado após o tempo fornecido.
lastChangeTill	timestamp	Retorna apenas triggers que mudaram seu estado antes do tempo fornecido.
only_true	flag	Retorna apenas triggers que recentemente estiveram em estado de problema.
min_severity	integer	Retorna apenas triggers com severidade maior ou igual à severidade fornecida.
evaltype	integer	Método de avaliação de tag. Valores possíveis: 0 - *(padrão)* E/Ou; 2 - Ou.
tags	array	Retorna apenas triggers com as tags fornecidas. Formato: `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`. Um array vazio retorna todos os triggers. Valores possíveis para operator: 0 - *(padrão)* Contém; 1 - Igual; 2 - Não contém; 3 - Não é igual; 4 - Existe; 5 - Não existe.
expandComment	flag	Expande macros na descrição do trigger.
expandDescription	flag	Expande macros no nome do trigger.
expandExpression	flag	Expande funções e macros na expressão do trigger.

Parâmetro	Tipo	Descrição
selectHostGroups	query	Retorna os grupos de hosts aos quais o trigger pertence na propriedade `hostgroups`.
selectHosts	query	Retorna os hosts aos quais o trigger pertence na propriedade `hosts`.
selectItems	query	Retorna os itens contidos pelo trigger na propriedade `items`.
selectFunctions	query	Retorna funções usadas no trigger na propriedade `functions`. Os objetos de função representam as funções usadas na expressão do trigger e possuem as seguintes propriedades: `functionid` - `(ID)` ID da função; `itemid` - `(ID)` ID do item usado na função; `function` - `(string)` nome da função; `parameter` - `(string)` parâmetro passado para a função. O parâmetro de consulta é substituído pelo símbolo $ na string retornada.
selectDependencies	query	Retorna triggers dos quais o trigger depende na propriedade `dependencies`.
selectDiscoveryRule	query	Retorna a regra de descoberta de baixo nível que criou o trigger na propriedade `discoveryRule`.
selectLastEvent	query	Retorna o último evento significativo do trigger na propriedade `lastEvent`.
selectTags	query	Retorna as tags do trigger na propriedade `tags`.
selectTemplateGroups	query	Retorna os grupos de templates aos quais o trigger pertence na propriedade `templategroups`.
selectTriggerDiscovery	query	Retorna o objeto de descoberta de trigger na propriedade `triggerDiscovery`. Os objetos de descoberta de trigger vinculam o trigger a um protótipo de trigger do qual ele foi criado. Possui as seguintes propriedades: `parent_triggerid` - `(ID)` ID do protótipo de trigger do qual o trigger foi criado; `status` - `(int)` status da descoberta do trigger: 0 - *(padrão)* trigger foi descoberto, 1 - trigger não é mais descoberto; `ts_delete` - `(timestamp)` hora em que um trigger que não é mais descoberto será excluído; `ts_disable` - `(timestamp)` hora em que um trigger que não é mais descoberto será desabilitado; `disable_source` - `(int)` indicador de se o trigger foi desabilitado por uma regra LLD ou manualmente: 0 - *(padrão)* desabilitado automaticamente, 1 - desabilitado por uma regra LLD.
filter	object	Retorna apenas os resultados que correspondem exatamente ao filtro fornecido. Aceita um objeto, onde as chaves são nomes de propriedades e os valores são um único valor ou um array de valores para corresponder. Não suporta propriedades do tipo de dado `text` data type. Suporta propriedades adicionais: `host` - nome técnico do host ao qual o trigger pertence; `hostid` - ID do host ao qual o trigger pertence.
limitSelects	integer	Limita o número de registros retornados por subconsultas. Aplica-se às seguintes subconsultas: `selectHosts` - os resultados serão classificados por `host`.
sortfield	string/array	Classifica o resultado pelas propriedades fornecidas. Valores possíveis: `triggerid`, `description`, `status`, `priority`, `lastchange`, `hostname`.
countOutput	boolean	Estes parâmetros são descritos no comentário de referência.
editable	boolean	

Parâmetro	Tipo	Descrição
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	
selectGroups (obsoleto)	query	Este parâmetro está obsoleto, use `selectHostGroups` ou `selectTemplateGroups` em vez disso. Retorna os grupos de hosts e grupos de templates aos quais o trigger pertence na propriedade `groups`.

Valores de retorno

`(integer/array)` Retorna:

- um array de objetos;
- a contagem de objetos recuperados, se o parâmetro `countOutput` tiver sido usado.

Exemplos

Recuperando dados pelo ID do trigger

Recupere todos os dados e as funções usadas no trigger "14062".

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "trigger.get",
    "params": {
        "triggerids": "14062",
        "output": "extend",
        "selectFunctions": "extend"
    },
    "id": 1
}
```

Resposta:

```
{
    "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",
                    "parameter": "$",
                    "function": "last"
                }
            ]
        }
    ],
    "id": 1
}
```

Recuperando triggers em estado de problema

Recupere o ID, nome e gravidade de todos os triggers em estado de problema e classifique-os por gravidade em ordem decrescente.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "trigger.get",
    "params": {
        "output": [
            "triggerid",
            "description",
            "priority"
        ],
        "filter": {
            "value": 1
        },
        "sortfield": "priority",
        "sortorder": "DESC"
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "triggerid": "13907",
            "description": "Zabbix self-monitoring processes < 100% busy",
            "priority": "4"
        },
        {
            "triggerid": "13824",
            "description": "Zabbix discoverer processes more than 75% busy",
            "priority": "3"
        }
    ],
    "id": 1
}
```

Recuperando um trigger específico com tags

Recupere um trigger específico com tags.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "trigger.get",
    "params": {
        "output": [
            "triggerid",
            "description"
        ],
        "selectTags": "extend",
        "triggerids": [
            "17578"
        ]
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "triggerid": "17370",
            "description": "Service status",
            "tags": [
                {
                    "tag": "service",
                    "value": "{{ITEM.VALUE}.regsub(\"Service (.*) has stopped\", \"\\1\")}"
                },
                {
                    "tag": "error",
                    "value": ""
                }
            ]
        }
    ],
    "id": 1
}
```

Veja também

- Regra de descoberta
- Item
- Host
- Grupo de hosts
- Grupo de templates

Fonte

CTrigger::get() in *ui/include/classes/api/services/CTrigger.php*.

trigger.update

Descrição

`object trigger.update(object/array triggers)`

Este método permite atualizar triggers existentes.

> **Note:**
> Este método está disponível apenas para os tipos de usuário *Admin* e *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object/array)` Propriedades do trigger a serem atualizadas.

A propriedade `triggerid` deve ser definida para cada trigger, todas as outras propriedades são opcionais. Somente as propriedades passadas serão atualizadas, todas as outras permanecerão inalteradas.

Além das propriedades padrão do trigger, o método aceita os seguintes parâmetros.

Parâmetro	Tipo	Descrição
dependencies	array	Triggers dos quais o trigger depende. Os triggers devem ter apenas a propriedade `triggerid` definida.
tags	array	Tags do trigger.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs das triggers atualizados na propriedade `triggerids`.

Exemplos

Habilitando um trigger

Habilite um trigger, ou seja, defina seu status como "0".

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "trigger.update",
    "params": {
        "triggerid": "13938",
        "status": 0
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "triggerids": [
            "13938"
        ]
    },
    "id": 1
}
```

Substituindo tags de triggers

Substitua tags para trigger.

Requisição:

```
{
    "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
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "triggerids": [
            "13938"
        ]
    },
    "id": 1
}
```

Substituindo dependências

Substitua as dependências para o trigger.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "trigger.update",
    "params": {
        "triggerid": "22713",
        "dependencies": [
            {
                "triggerid": "22712"
            },
            {
                "triggerid": "22772"
            }
        ]
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "triggerids": [
            "22713"
        ]
    },
    "id": 1
}
```

Fonte

CTrigger::update() in *ui/include/classes/api/services/CTrigger.php*.

Usuário

Esta classe foi projetada para trabalhar com usuários.

Referências de objetos:

- User
- Media

Métodos disponíveis:

- user.checkauthentication - verifica e prolonga sessões de usuário
- user.create - cria novos usuários
- user.delete - exclui usuários

- user.get - recupera usuários
- user.login - faz login na API
- user.logout - faz logout da API
- user.provision - provisiona usuários LDAP
- user.resettotp - redefine segredos TOTP do usuário
- user.unblock - desbloqueia usuários
- user.update - atualiza usuários

Objeto user

Os seguintes objetos estão diretamente relacionados à API `user`.

Usuário

O objeto usuário possui as seguintes propriedades.

Propriedade	Tipo	Descrição
userid	ID	ID do usuário. Comportamento da propriedade: - *somente leitura* - *obrigatório* para operações de atualização
username	string	Nome do usuário. Comportamento da propriedade: - *obrigatório* para operações de criação - *somente leitura* para usuários provisionados se o usuário estiver vinculado a um diretório de usuários (`userdirectoryid` definido para um valor válido diferente de ”0”), e o status de provisionamento do diretório de usuários estiver habilitado (`provision_status` do objeto Diretório de Usuários definido como ”1”), e o status de autenticação de todo o provisionamento LDAP ou SAML estiver habilitado (`ldap_jit_status` do objeto Autenticação definido como ”Habilitado para IdPs LDAP configurados” ou `saml_jit_status` do objeto Autenticação definido como ”Habilitado para IdPs SAML configurados”)
passwd	string	Senha do usuário. O valor deste parâmetro pode ser uma string vazia se o usuário estiver vinculado a um diretório de usuários. Comportamento da propriedade: - *somente gravação*
roleid	ID	ID da função do usuário. Observe que usuários sem função podem fazer login no Zabbix apenas usando autenticação LDAP ou SAML, desde que suas informações LDAP/SAML correspondam aos mapeamentos de grupos de usuários configurados no Zabbix.
attempt_clock	timestamp	Hora da última tentativa de login malsucedida. Comportamento da propriedade: - *somente leitura*
attempt_failed	integer	Contagem recente de tentativas de login malsucedidas. Comportamento da propriedade: - *somente leitura*
attempt_ip	string	Endereço IP de onde veio a última tentativa de login malsucedida. Comportamento da propriedade: - *somente leitura*

Propriedade	Tipo	Descrição
autologin	integer	Se o login automático está habilitado. Valores possíveis: 0 - *(padrão)* login automático desabilitado; 1 - login automático habilitado.
autologout	string	Tempo de vida da sessão do usuário. Aceita segundos e unidade de tempo com sufixo. Se definido como 0s, a sessão nunca expirará. Padrão: 15m.
lang	string	Código do idioma do usuário, por exemplo, `pt_BR`. Padrão: `default` - padrão do sistema.
name	string	Nome do usuário.
provisioned	integer	Se o usuário foi provisionado. Valores possíveis: 0 - não provisionado; 1 - provisionado. Esta propriedade é suportada desde o Zabbix 7.0.9. Comportamento da propriedade: - *somente leitura*
refresh	string	Período de atualização automática. Aceita segundos ou unidade de tempo com sufixo (por exemplo, 30s, 90s, 1m, 1h). Padrão: 30s.
rows_per_page	integer	Quantidade de linhas de objetos para mostrar por página. Padrão: 50.
surname	string	Sobrenome do usuário.
theme	string	Tema do usuário. Valores possíveis: `default` - *(padrão)* padrão do sistema; `blue-theme` - Azul; `dark-theme` - Escuro; `hc-light` - Alto contraste claro; `hc-dark` - Alto contraste escuro.
ts_provisioned	timestamp	Hora em que a última operação de provisionamento foi realizada. Comportamento da propriedade: - *somente leitura* - *suportado* para usuários do tipo *Super admin* (desde o Zabbix 7.0.9)
url	string	URL da página para redirecionar o usuário após o login.
userdirectoryid	ID	ID do diretório de usuários ao qual o usuário está vinculado. Usado para provisionamento (criação ou atualização), bem como para fazer login de um usuário vinculado a um diretório de usuários. Para operações de login, o valor desta propriedade terá prioridade sobre a propriedade `userdirectoryid` dos grupos de usuários aos quais o usuário pertence. Padrão: 0. Comportamento da propriedade: - *somente leitura* - *suportado* para usuários do tipo *Super admin* (desde o Zabbix 7.0.9)

Propriedade	Tipo	Descrição
timezone	string	Fuso horário do usuário, por exemplo, `Europe/London`, `UTC`. Padrão: `default` - padrão do sistema. Para a lista completa de fusos horários suportados, consulte a documentação do PHP.

Mídia

O objeto mídia possui as seguintes propriedades.

Propriedade	Tipo	Descrição
mediaid	ID	ID da mídia do usuário. Comportamento da propriedade: - *somente leitura*
mediatypeid	ID	ID do tipo de mídia usado pela mídia do usuário. Comportamento da propriedade: - *obrigatório*
sendto	string/array	Endereço, nome de usuário ou outro identificador do destinatário. Se o `type` do Tipo de mídia estiver definido como "Email", os valores são representados como array. Para outros tipos de Tipos de mídia, o valor é representado como string. Comportamento da propriedade: - *obrigatório*
active	integer	Se a mídia está habilitada. Valores possíveis: 0 - *(padrão)* habilitada; 1 - desabilitada.
severity	integer	Severidades de trigger para enviar notificações. Valores possíveis de bitmap: 1 - Não classificado; 2 - Informação; 4 - Aviso; 8 - Médio; 16 - Alto; 32 - Desastre. Este é um campo de máscara de bits; qualquer soma dos valores possíveis de bitmap é aceitável (por exemplo, 48 para Médio, Alto e Desastre). Padrão: 63.
period	string	Horário em que as notificações podem ser enviadas como um período de tempo ou macros de usuário separadas por ponto e vírgula. Padrão: 1-7,00:00-24:00.
provisioned	integer	Se o usuário foi provisionado. Valores possíveis: 0 - não provisionado; 1 - provisionado. Esta propriedade é suportada desde o Zabbix 7.0.9.

Propriedade	Tipo	Descrição
userdirectory_mediaid	ID	ID de mapeamento de mídia do diretório de usuários para mídia provisionada. Comportamento da propriedade: - *somente leitura* - *suportado* para usuários do tipo *Super admin* (desde o Zabbix 7.0.9)

user.checkAuthentication

Descrição

`object user.checkAuthentication`

Este método verifica e prolonga a sessão do usuário.

Attention:
Chamar o método `user.checkAuthentication` usando o parâmetro `sessionid` prolonga a sessão do usuário por padrão.

Parâmetros

O método aceita os seguintes parâmetros.

Parâmetro	Tipo	Descrição
extend	boolean	Se a sessão do usuário deve ser prolongada. Valor padrão: ”true”. Definir o valor como ”false” permite verificar a sessão do usuário sem prolongá-la.
sessionid	string	Comportamento do parâmetro: - *suportado* se `sessionid` estiver definido Token de autenticação do usuário.
token	string	Comportamento do parâmetro: - *obrigatório* se `token` não estiver definido Token de API do usuário.
		Comportamento do parâmetro: - *obrigatório* se `sessionid` não estiver definido

Valores de retorno

`(object)` Retorna um objeto contendo informações sobre o usuário.

Além das propriedades padrão do usuário, as seguintes informações são retornadas.

Propriedade	Tipo	Descrição
auth_type	integer	Autenticação padrão para o usuário. Consulte a propriedade `authentication_type` do objeto Authentication para uma lista de valores possíveis.
debug_mode	integer	Se o modo de depuração está habilitado ou desabilitado para o usuário. Consulte a propriedade `debug_mode` do objeto User group para uma lista de valores possíveis.
deprovisioned	boolean	Se o usuário pertence a um grupo de usuários desprovisionados.

Propriedade	Tipo	Descrição
gui_access	string	Método de autenticação do usuário no frontend. Consulte a propriedade `gui_access` do objeto User group para uma lista de valores possíveis.
secret	string	String aleatória de 32 caracteres. É gerada no login do usuário. A propriedade `secret` não é retornada se a sessão do usuário for verificada usando um token de API.
sessionid	string	Token de autenticação, que deve ser usado nas próximas requisições da API. A propriedade `sessionid` não é retornada se a sessão do usuário for verificada usando um token de API.
type	integer	Tipo de usuário. Consulte a propriedade `type` do objeto Role para uma lista de valores possíveis.
userip	string	Endereço IP do usuário.

Exemplos

Verificar autenticação usando o token de autenticação

Verifique e prolongue uma sessão de usuário usando o token de autenticação do usuário e retorne informações adicionais sobre o usuário.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "user.checkAuthentication",
    "params": {
        "sessionid": "673b8ba11562a35da902c66cf5c23fa2"
    },
    "id": 1
}
```

Resposta:

```
{
    "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
}
```

Verificar autenticação usando o token da API

Verifique uma sessão de usuário usando o token da API do usuário e retorne informações adicionais sobre o usuário.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "user.checkAuthentication",
    "params": {
        "token": "00aff470e07c12d707e50d98cfe39edef9e6ec349c14728dbdfbc8ddc5ea3eae"
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "userid": "1",
        "username": "Admin",
        "name": "Zabbix",
        "surname": "Administrator",
        "url": "",
        "autologin": "1",
        "autologout": "0",
        "lang": "ru_RU",
        "refresh": "0",
        "theme": "default",
        "attempt_failed": "0",
        "attempt_ip": "127.0.0.1",
        "attempt_clock": "1355919338",
        "rows_per_page": "50",
        "timezone": "Europe/Riga",
        "roleid": "3",
        "userdirectoryid": "0",
        "ts_provisioned": "0",
        "debug_mode": 0,
        "deprovisioned": false,
        "gui_access": "1",
        "mfaid": 0,
        "auth_type": 0,
        "type": 3,
        "userip": "127.0.0.1"
    },
    "id": 1
}
```

Fonte

CUser::checkAuthentication() em *ui/include/classes/api/services/CUser.php*.

user.create

Descrição

`object user.create(object/array users)`

Este método permite criar novos usuários.

> **Note:**
> Este método está disponível apenas para o tipo de usuário *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função de usuário. Veja Funções de usuário para mais informações.

> **Note:**
> A força da senha do usuário é validada de acordo com as regras de política de senha definidas pela API de Autenticação. Veja API de Autenticação para mais informações.

Parâmetros

`(object/array)` Usuários a serem criados.

Além das propriedades padrão do usuário, o método aceita os seguintes parâmetros.

Parâmetro	Tipo	Descrição
usrgrps	array	Grupos de usuários aos quais o usuário será adicionado. Os grupos de usuários devem ter apenas a propriedade `usrgrpid` definida.
medias	array	Mídias do usuário a serem criadas.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos usuários criados na propriedade `userids`. A ordem dos IDs retornados corresponde à ordem dos usuários passados.

Exemplos

Criando um usuário

Crie um novo usuário, adicione-o a um grupo de usuários e crie uma nova mídia para ele.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "user.create",
    "params": {
        "username": "John",
        "passwd": "Doe123",
        "roleid": "5",
        "usrgrps": [
            {
                "usrgrpid": "7"
            }
        ],
        "medias": [
            {
                "mediatypeid": "1",
                "sendto": [
                    "support@company.com"
                ],
                "active": 0,
                "severity": 63,
                "period": "1-7,00:00-24:00"
            }
        ]
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "userids": [
            "12"
        ]
    },
    "id": 1
}
```

Veja também

- Autenticação
- Mídia
- Grupo de usuários
- Função

Fonte

CUser::create() in *ui/include/classes/api/services/CUser.php*.

user.delete

Descrição

`object user.delete(array users)`

Este método permite excluir usuários.

> **Note:**
> Este método está disponível apenas para o tipo de usuário *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(array)` IDs de usuários a serem excluídos.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos usuários excluídos na propriedade `userids`.

Exemplos

Excluindo vários usuários

Exclua dois usuários.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "user.delete",
    "params": [
        "1",
        "5"
    ],
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "userids": [
            "1",
            "5"
        ]
    },
```

```
    "id": 1
}
```

Fonte

CUser::delete() in *ui/include/classes/api/services/CUser.php*.

user.get

Descrição

`integer/array user.get(object parameters)`

O método permite recuperar usuários de acordo com os parâmetros fornecidos.

> **Note:**
> Este método está disponível para usuários de qualquer tipo. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object)` Parâmetros que definem a saída desejada.

O método suporta os seguintes parâmetros.

> **Note:**
> A partir do Zabbix 7.0.9, ao solicitar informações de mídia, permissões ou função do usuário, usuários do tipo *Admin* e *User* podem recuperar dados apenas sobre seu próprio usuário. Para um exemplo, consulte Recuperando usuários como *Admin*.

Parâmetro	Tipo	Descrição
mediaids	ID/array	Retorna apenas usuários que utilizam a mídia informada.
mediatypeids	ID/array	Retorna apenas usuários que utilizam os tipos de mídia informados.
userids	ID/array	Retorna apenas usuários com os IDs informados.
usrgrpids	ID/array	Retorna apenas usuários que pertencem aos grupos de usuários informados.
getAccess	flag	Adiciona informações adicionais sobre as permissões do usuário. Adiciona as seguintes propriedades para cada usuário: `gui_access` - `(integer)` método de autenticação do usuário no frontend. Consulte a propriedade `gui_access` do objeto grupo de usuários para uma lista de valores possíveis. `debug_mode` - `(integer)` indica se o debug está habilitado para o usuário. Valores possíveis: 0 - debug desabilitado, 1 - debug habilitado. `users_status` - `(integer)` indica se o usuário está desabilitado. Valores possíveis: 0 - usuário habilitado, 1 - usuário desabilitado.
selectMedias	query	Retorna as mídias utilizadas pelo usuário na propriedade `medias`.
selectMediatypes	query	Retorna os tipos de mídia utilizados pelo usuário na propriedade `mediatypes`. Consulte `mediatype.get` para restrições baseadas no tipo de usuário.
selectUsrgrps	query	Retorna os grupos de usuários aos quais o usuário pertence na propriedade `usrgrps`. Consulte `usergroup.get` para restrições baseadas no tipo de usuário.
selectRole	query	Retorna a função do usuário na propriedade `role`.

Parâmetro	Tipo	Descrição
filter	object	Retorna apenas os resultados que correspondem exatamente ao filtro informado. Aceita um objeto, onde as chaves são nomes de propriedades e os valores são um único valor ou um array de valores para correspondência. Não suporta propriedades do tipo de dado `text` data type. Propriedades possíveis do objeto User para usuários do tipo *Admin* e *User* ao solicitar dados de usuários em seu grupo de usuários (desde o Zabbix 7.0.9): `userid`, `name`, `surname`, `username`.
output	query	Propriedades do objeto User a serem retornadas. Desde o Zabbix 7.0.9, usuários do tipo *Admin* e *User* podem recuperar apenas as seguintes propriedades: - Para seu próprio usuário: `userid`, `attempt_clock`, `attempt_failed`, `attempt_ip`, `autologin`, `autologout`, `lang`, `name`, `provisioned`, `refresh`, `roleid`, `rows_per_page`, `surname`, `theme`, `timezone`, `url`, `username`. - Para usuários em seu grupo de usuários: `userid`, `name`, `surname`, `username`. Padrão: `extend`.
search	object	Retorna resultados que correspondam ao padrão informado (case-insensitive). Aceita um objeto, onde as chaves são nomes de propriedades e os valores são strings a serem pesquisadas. Se nenhuma opção adicional for informada, será realizada uma busca `LIKE "%...%"`. Suporta apenas propriedades dos tipos de dado `string` e `text` data type. Propriedades possíveis do objeto User para usuários do tipo *Admin* e *User* ao solicitar dados de usuários em seu grupo de usuários (desde o Zabbix 7.0.9): `name`, `surname`, `username`.
sortfield	string/array	Ordena o resultado pelas propriedades informadas. Valores possíveis: `userid`, `username`.
countOutput	boolean	Esses parâmetros estão descritos no comentário de referência.
editable	boolean	
excludeSearch	boolean	
limit	integer	
preservekeys	boolean	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valores de retorno

`(integer/array)` Retorna:

- uma array de objetos;
- a contagem de objetos recuperados, se o parâmetro `countOutput` foi usado.

Exemplos

Recuperando usuários

Recupere todos os usuários configurados.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "user.get",
    "params": {
        "output": "extend"
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "userid": "1",
            "username": "Admin",
            "name": "Zabbix",
            "surname": "Administrator",
            "url": "",
            "autologin": "1",
            "autologout": "0",
            "lang": "en_US",
            "refresh": "0s",
            "theme": "default",
            "attempt_failed": "0",
            "attempt_ip": "",
            "attempt_clock": "0",
            "rows_per_page": "50",
            "timezone": "default",
            "roleid": "3",
            "userdirectoryid": "0",
            "ts_provisioned": "0"
        },
        {
            "userid": "2",
            "username": "guest",
            "name": "",
            "surname": "",
            "url": "",
            "autologin": "0",
            "autologout": "15m",
            "lang": "default",
            "refresh": "30s",
            "theme": "default",
            "attempt_failed": "0",
            "attempt_ip": "",
            "attempt_clock": "0",
            "rows_per_page": "50",
            "timezone": "default",
            "roleid": "4",
            "userdirectoryid": "0",
            "ts_provisioned": "0"
        },
        {
            "userid": "3",
            "username": "user",
            "name": "Zabbix",
            "surname": "User",
            "url": "",
            "autologin": "0",
            "autologout": "0",
            "lang": "ru_RU",
```

```
            "refresh": "15s",
            "theme": "dark-theme",
            "attempt_failed": "0",
            "attempt_ip": "",
            "attempt_clock": "0",
            "rows_per_page": "100",
            "timezone": "default",
            "roleid": "1",
            "userdirectoryid": "0",
            "ts_provisioned": "0"
        }
    ],
    "id": 1
}
```

Recuperando usuários como *Admin*

Como um usuário do tipo *Admin*, recupere dados detalhados sobre seu próprio usuário e dados limitados para usuários em seu grupo de usuários.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "user.get",
    "params": {
        "output": "extend",
        "getAccess": true,
        "selectMedias": "extend",
        "selectMediatypes": "extend",
        "selectUsrgrps": "extend",
        "selectRole": "extend"
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "userid": "1",
            "username": "Admin",
            "name": "Zabbix",
            "surname": "Administrator",
            "usrgrps": [
                {
                    "usrgrpid": "7",
                    "name": "Zabbix administrators",
                    "gui_access": "0",
                    "users_status": "0",
                    "debug_mode": "0",
                    "mfa_status": "0"
                }
            ]
        },
        {
            "userid": "3",
            "username": "database-admin",
            "name": "John",
            "surname": "Doe",
            "url": "",
            "autologin": "0",
            "autologout": "0",
```

```
            "lang": "default",
            "refresh": "30s",
            "theme": "default",
            "attempt_failed": "0",
            "attempt_ip": "",
            "attempt_clock": "0",
            "rows_per_page": "50",
            "timezone": "default",
            "roleid": "2",
            "provisioned": "0",
            "gui_access": "0",
            "debug_mode": "0",
            "users_status": "0",
            "usrgrps": [
                {
                    "usrgrpid": "7",
                    "name": "Zabbix administrators",
                    "gui_access": "0",
                    "users_status": "0",
                    "debug_mode": "0",
                    "mfa_status": "0"
                }
            ],
            "medias": [
                {
                    "mediaid": "2",
                    "mediatypeid": "1",
                    "sendto": [
                        "john.doe@example.com"
                    ],
                    "active": "0",
                    "severity": "63",
                    "period": "1-7,00:00-24:00",
                    "provisioned": 0
                }
            ],
            "mediatypes": [
                {
                    "mediatypeid": "1",
                    "type": "0",
                    "name": "Email",
                    "status": "0",
                    "description": "",
                    "maxattempts": "3"
                }
            ],
            "role": {
                "roleid": "2",
                "name": "Admin role",
                "type": "2",
                "readonly": "0"
            }
        }
    ],
    "id": 1
}
```


Recuperando dados do usuário

Recupere os dados de um usuário com ID ”12”.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "user.get",
    "params": {
        "output": ["userid", "username"],
        "selectRole": "extend",
        "userids": "12"
    },
    "id": 1
}
```

Resposta:

```
{
  "jsonrpc": "2.0",
  "result": [
      {
          "userid": "12",
          "username": "John",
          "role": {
              "roleid": "5",
              "name": "Operator",
              "type": "1",
              "readonly": "0"
          }
      }
  ],
  "id": 1
}
```

Veja também

- Mídia
- Tipo de mídia
- Grupo de usuários
- Função

Fonte

CUser::get() in *ui/include/classes/api/services/CUser.php*.

user.login

Descrição

`string/object user.login(object parameters)`

Este método permite fazer login na API e gerar um token de autenticação.

Warning:
Ao usar este método, você também precisa executar user.logout para evitar a geração de um grande número de registros de sessões abertas.

Attention:
Este método está disponível apenas para usuários não autenticados que não pertencem a nenhum grupo de usuários com autenticação multifator habilitada. Este método deve ser chamado sem o parâmetro `auth` na solicitação JSON-RPC.

Parâmetros

`(object)` Parâmetros contendo o nome de usuário e a senha.

O método aceita os seguintes parâmetros.

Parâmetro	Tipo	Descrição
password	string	Senha do usuário. Comportamento do parâmetro: - *obrigatório*
username	string	Nome do usuário. Comportamento do parâmetro: - *obrigatório*
userData	flag	Retorna informações sobre o usuário autenticado.

Valores de retorno

`(string/object)` Se o parâmetro `userData` for usado, retorna um objeto contendo informações sobre o usuário autenticado.

Além das propriedades padrão do usuário, as seguintes informações são retornadas:

Propriedade	Tipo	Descrição
auth_type	integer	Autenticação padrão para o usuário. Consulte a propriedade `authentication_type` do objeto Authentication para uma lista de valores possíveis.
debug_mode	integer	Se o modo de depuração está ativado ou desativado para o usuário. Consulte a propriedade `debug_mode` do objeto User group para uma lista de valores possíveis.
deprovisioned	boolean	Se o usuário pertence a um grupo de usuários desprovisionados.
gui_access	string	Método de autenticação do usuário no frontend. Consulte a propriedade `gui_access` do objeto User group para uma lista de valores possíveis.
mfaid	integer	ID do método MFA a ser usado para o usuário durante o login. Retorna "0" se o MFA estiver desativado globalmente ou para todos os grupos de usuários aos quais o usuário pertence.
secret	string	String aleatória de 32 caracteres. É gerada no login do usuário.
sessionid	string	Token de autenticação, que deve ser usado nas próximas requisições da API.
type	integer	Tipo de usuário. Consulte a propriedade `type` do objeto Role para uma lista de valores possíveis.
userip	string	Endereço IP do usuário.

Note:
Se um usuário foi autenticado com sucesso após uma ou mais tentativas falhas, o método retornará os valores atuais para as propriedades `attempt_clock`, `attempt_failed` e `attempt_ip` e em seguida irá redefini-las.

Se o parâmetro `userData` não for usado, o método retorna um token de autenticação.

Note:
O token de autenticação gerado deve ser memorizado e usado no parâmetro `auth` das próximas requisições JSON-RPC. Ele também é necessário ao usar autenticação HTTP.

Exemplos

Autenticando um usuário

Autentica um usuário.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "user.login",
    "params": {
        "username": "Admin",
        "password": "zabbix"
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": "0424bd59b807674191e7d77572075f33",
    "id": 1
}
```

Solicitando informações do usuário autenticado

Autentica e retorna informações adicionais sobre o usuário.

Solicitação:

```
{
    "jsonrpc": "2.0",
    "method": "user.login",
    "params": {
        "username": "Admin",
        "password": "zabbix",
        "userData": true
    },
    "id": 1
}
```

Resposta:

```
{
    "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
}
```

Veja também

- user.logout

Fonte

CUser::login() in *ui/include/classes/api/services/CUser.php*.

user.logout

Descrição

`string/object user.logout(array)`

Este método permite sair da API e invalida o token de autenticação atual.

> **Note:**
> Este método está disponível para usuários de qualquer tipo. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(array)` O método aceita um array vazio.

Valores de retorno

`(boolean)` Retorna `true` se o usuário foi desconectado com sucesso.

Exemplos

Logout

Faz logout da API.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "user.logout",
    "params": [],
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": true,
    "id": 1
}
```

Veja também

- user.login

Fonte

CUser::login() in *ui/include/classes/api/services/CUser.php*.

user.provision

Descrição

`object user.provision(object/array users)`

Este método permite provisionar usuários LDAP.

Note:
Este método está disponível apenas para o tipo de usuário *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função de usuário. Consulte Funções de usuário para obter mais informações.

Parâmetros

`(array)` IDs dos usuários a serem provisionados.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos usuários provisionados na propriedade `userids`.

Exemplos

Provisionando múltiplos usuários

Provisionar dois usuários.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "user.provision",
    "params": [
        "1",
        "5"
    ],
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "userids": [
            "1",
            "5"
        ]
    },
    "id": 1
}
```

Fonte

CUser::provision() em *ui/include/classes/api/services/CUser.php*.

user.resettotp

Descrição

`object user.resettotp(object/array users)`

Este método permite redefinir os segredos TOTP do usuário.

Note:
Este método está disponível apenas para o tipo de usuário *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Consulte Funções de usuário para obter mais informações.

Parâmetros

`(array)` IDs dos usuários para os quais redefinir os segredos TOTP.

Note:
As sessões de usuário para os usuários especificados também serão excluídas (exceto para o usuário que está enviando a solicitação).

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos usuários para os quais os segredos TOTP foram redefinidos, sob a propriedade `userids`.

Exemplos

Redefinindo segredos TOTP para vários usuários

Redefina os segredos TOTP para dois usuários.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "user.resettotp",
    "params": [
        "1",
        "5"
    ],
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "userids": [
            "1",
            "5"
        ]
    },
    "id": 1
}
```

Veja também

- Objeto MFA

Fonte

CUser::resettotp() em *ui/include/classes/api/services/CUser.php*.

user.unblock

Descrição

`object user.unblock(array userids)`

Este método permite desbloquear usuários.

> **Note:**
> Este método está disponível apenas para o tipo de usuário *Super admin*. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(array)` IDs dos usuários a serem desbloqueados.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos usuários desbloqueados na propriedade `userids`.

Exemplos

Desbloqueando vários usuários

Desbloqueie dois usuários.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "user.unblock",
    "params": [
        "1",
        "5"
    ],
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "userids": [
            "1",
            "5"
        ]
    },
    "id": 1
}
```

Fonte

CUser::unblock() in *ui/include/classes/api/services/CUser.php*.

user.update

Descrição

`object user.update(object/array users)`

Este método permite atualizar usuários existentes.

Note:
Este método está disponível para usuários de qualquer tipo. As permissões para chamar o método podem ser revogadas nas configurações de função de usuário. Veja Funções de usuário para mais informações.

Note:
A força da senha do usuário é validada de acordo com as regras de política de senha definidas pela API de Autenticação. Veja API de Autenticação para mais informações.

Parâmetros

`(object/array)` Propriedades do usuário a serem atualizadas.

A propriedade `userid` deve ser definida para cada usuário, todas as outras propriedades são opcionais. Somente as propriedades passadas serão atualizadas, todas as outras permanecerão inalteradas.

Além das propriedades padrão do usuário, o método aceita os seguintes parâmetros.

Parâmetro	Tipo	Descrição
current_passwd	string	Senha atual do usuário. O valor deste parâmetro pode ser uma string vazia se o usuário estiver vinculado a um diretório de usuários. Comportamento do parâmetro: - *somente escrita* - *obrigatório* se `passwd` do objeto User estiver definido e o usuário alterar sua própria senha

Parâmetro	Tipo	Descrição
usrgrps	array	Grupos de usuários para substituir os grupos de usuários existentes. Os grupos de usuários devem ter apenas a propriedade `usrgrpid` definida.
medias	array	Mídias do usuário para substituir as mídias existentes não provisionadas. As mídias provisionadas podem ser omitidas ao atualizar as mídias.

Valores de retorno

`(object)` Retorna um objeto contendo os IDs dos usuários atualizados na propriedade `userids`.

Exemplos

Renomeando um usuário

Renomeie um usuário para John Doe.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "user.update",
    "params": {
        "userid": "1",
        "name": "John",
        "surname": "Doe"
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "userids": [
            "1"
        ]
    },
    "id": 1
}
```

Alterando a função do usuário

Altere a função de um usuário.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "user.update",
    "params": {
        "userid": "12",
        "roleid": "6"
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "userids": [
            "12"
        ]
```

```
    },
    "id": 1
}
```

Veja também

- Autenticação

Fonte

CUser::update() in *ui/include/classes/api/services/CUser.php*.

Verificação de descoberta

Esta classe foi projetada para trabalhar com verificações de descoberta.

Referências de objetos:

- Verificação de descoberta

Métodos disponíveis:

- dcheck.get - recuperar verificações de descoberta

Objeto de verificação de descoberta

Os seguintes objetos estão diretamente relacionados à API `dcheck`.

Verificação de descoberta

O objeto de verificação de descoberta define uma verificação específica realizada por uma regra de descoberta de rede. Ele possui as seguintes propriedades.

Propriedade	Tipo	Descrição
dcheckid	ID	ID da verificação de descoberta.
druleid	ID	ID da regra de descoberta à qual a verificação pertence.
key_	string	Chave do item (se `type` estiver definido como ”Zabbix agent”) ou OID SNMP (se `type` estiver definido como ”SNMPv1 agent”, ”SNMPv2 agent” ou ”SNMPv3 agent”). Comportamento da propriedade: - *obrigatório* se `type` estiver definido como ”Zabbix agent”, ”SNMPv1 agent”, ”SNMPv2 agent” ou ”SNMPv3 agent”
ports	string	Um ou vários intervalos de portas a serem verificados, separados por vírgulas. Padrão: 0. Comportamento da propriedade: - *suportado* se `type` estiver definido como ”SSH” (0), ”LDAP” (1), ”SMTP” (2), ”FTP” (3), ”HTTP” (4), ”POP” (5), ”NNTP” (6), ”IMAP” (7), ”TCP” (8), ”Zabbix agent” (9), ”SNMPv1 agent” (10), ”SNMPv2 agent” (11), ”SNMPv3 agent” (13), ”HTTPS” (14) ou ”Telnet” (15)
snmp_community	string	Comunidade SNMP. Comportamento da propriedade: - *obrigatório* se `type` estiver definido como ”SNMPv1 agent” ou ”SNMPv2 agent”
snmpv3_authpassphrase	string	Senha de autenticação. Comportamento da propriedade: - *suportado* se `type` estiver definido como ”SNMPv3 agent” e `snmpv3_securitylevel` estiver definido como ”authNoPriv” ou ”authPriv”

Propriedade	Tipo	Descrição
snmpv3_authprotocol	integer	Protocolo de autenticação. Valores possíveis: 0 - *(padrão)* MD5; 1 - SHA1; 2 - SHA224; 3 - SHA256; 4 - SHA384; 5 - SHA512. Comportamento da propriedade: - *suportado* se `type` estiver definido como ”SNMPv3 agent” e `snmpv3_securitylevel` estiver definido como ”authNoPriv” ou ”authPriv”
snmpv3_contextname	string	Nome do contexto SNMPv3. Comportamento da propriedade: - *suportado* se `type` estiver definido como ”SNMPv3 agent”
snmpv3_privpassphrase	string	Senha de privacidade. Comportamento da propriedade: - *suportado* se `type` estiver definido como ”SNMPv3 agent” e `snmpv3_securitylevel` estiver definido como ”authPriv”
snmpv3_privprotocol	integer	Protocolo de privacidade. Valores possíveis: 0 - *(padrão)* DES; 1 - AES128; 2 - AES192; 3 - AES256; 4 - AES192C; 5 - AES256C. Comportamento da propriedade: - *suportado* se `type` estiver definido como ”SNMPv3 agent” e `snmpv3_securitylevel` estiver definido como ”authPriv”
snmpv3_securitylevel	string	Nível de segurança. Valores possíveis: 0 - noAuthNoPriv; 1 - authNoPriv; 2 - authPriv. Comportamento da propriedade: - *suportado* se `type` estiver definido como ”SNMPv3 agent”
snmpv3_securityname	string	Nome de segurança. Comportamento da propriedade: - *suportado* se `type` estiver definido como ”SNMPv3 agent”

Propriedade	Tipo	Descrição
type	integer	Tipo de verificação. Valores possíveis: 0 - SSH; 1 - LDAP; 2 - SMTP; 3 - FTP; 4 - HTTP; 5 - POP; 6 - NNTP; 7 - IMAP; 8 - TCP; 9 - Zabbix agent; 10 - SNMPv1 agent; 11 - SNMPv2 agent; 12 - ICMP ping; 13 - SNMPv3 agent; 14 - HTTPS; 15 - Telnet. Comportamento da propriedade: - *obrigatório*
uniq	integer	Se deve usar esta verificação como critério de exclusividade do dispositivo. Apenas uma verificação exclusiva pode ser configurada para uma regra de descoberta. Valores possíveis: 0 - *(padrão)* não usar esta verificação como critério de exclusividade; 1 - usar esta verificação como critério de exclusividade. Comportamento da propriedade: - *suportado* se `type` estiver definido como ”Zabbix agent”, ”SNMPv1 agent”, ”SNMPv2 agent” ou ”SNMPv3 agent”
host_source	integer	Fonte para o nome do host. Valores possíveis: 1 - *(padrão)* DNS; 2 - IP; 3 - valor de descoberta desta verificação.
name_source	integer	Fonte para o nome visível. Valores possíveis: 0 - *(padrão)* não especificado; 1 - DNS; 2 - IP; 3 - valor de descoberta desta verificação.
allow_redirect	integer	Permitir situação em que o alvo do ICMP ping responde de um endereço IP diferente. Valores possíveis: 0 - *(padrão)* tratar respostas redirecionadas como se o host de destino estivesse inativo (falha); 1 - tratar respostas redirecionadas como se o host de destino estivesse ativo (sucesso). Comportamento da propriedade: - *suportado* se `type` estiver definido como ”ICMP ping”

#####dcheck.get {#manual-api-reference-dcheck-get}

Descrição

`integer/array dcheck.get(object parameters)`

O método permite recuperar verificações de descoberta de acordo com os parâmetros fornecidos.

> **Note:**
> Este método está disponível para usuários de qualquer tipo. As permissões para chamar o método podem ser revogadas nas configurações de função do usuário. Veja Funções de usuário para mais informações.

Parâmetros

`(object)` Parâmetros que definem a saída desejada.

O método suporta os seguintes parâmetros.

Parâmetro	Tipo	Descrição
dcheckids	ID/array	Retorna apenas verificações de descoberta com os IDs fornecidos.
druleids	ID/array	Retorna apenas verificações de descoberta que pertencem às regras de descoberta fornecidas.
dserviceids	ID/array	Retorna apenas verificações de descoberta que detectaram os serviços descobertos fornecidos.
selectDRules	query	Retorna regras de descoberta relacionadas às verificações de descoberta.
sortfield	string/array	Ordena o resultado pelas propriedades fornecidas. Valores possíveis: `dcheckid`, `druleid`.
countOutput	boolean	Esses parâmetros são descritos no comentário de referência.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Valores de retorno

`(integer/array)` Retorna:

- uma matriz de objetos;
- a contagem de objetos recuperados, se o parâmetro `countOutput` tiver sido usado.

Exemplos

Recuperar verificações de descoberta para uma regra de descoberta

Recupere todas as verificações de descoberta usadas pela regra de descoberta ”6”.

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "dcheck.get",
    "params": {
        "output": "extend",
        "dcheckids": "6"
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "dcheckid": "6",
            "druleid": "4",
            "type": "3",
            "key_": "",
            "snmp_community": "",
            "ports": "21",
            "snmpv3_securityname": "",
            "snmpv3_securitylevel": "0",
            "snmpv3_authpassphrase": "",
            "snmpv3_privpassphrase": "",
            "uniq": "0",
            "snmpv3_authprotocol": "0",
            "snmpv3_privprotocol": "0",
            "snmpv3_contextname": "",
            "host_source": "1",
            "name_source": "0",
            "allow_redirect": "0"
        }
    ],
    "id": 1
}
```


Fonte

CDCheck::get() em *ui/include/classes/api/services/CDCheck.php*.

Apêndice 1. Comentário de referência

Notação Tipos de dados

A API do Zabbix suporta os seguintes tipos de dados como entrada:

Tipo	Descrição
ID	Um identificador exclusivo usado para referenciar uma entidade.
boolean	Um valor booleano (`true` ou `false`).
flag	Um valor que é considerado como `true` se passado e não for igual a `null`; caso contrário, o valor é considerado como `false`.
integer	Um número inteiro.
float	Um número de ponto flutuante.
string	Uma string de texto.
text	Uma string de texto mais longa.
timestamp	Um timestamp Unix.
array	Uma sequência ordenada de valores (um array simples).
object	Um array associativo.
query	Um valor que define os dados a serem retornados. O valor pode ser definido como um array de nomes de propriedades (para retornar apenas propriedades específicas) ou como um dos valores predefinidos: `extend` - retorna todas as propriedades do objeto; `count` - retorna o número de registros recuperados, suportado apenas por alguns subselects.

Attention:
A API do Zabbix sempre retorna valores apenas como strings ou arrays.

Comportamento da propriedade

Algumas das propriedades do objeto são marcadas com rótulos curtos para descrever seu comportamento. Os seguintes rótulos são usados:

- ***somente leitura*** - o valor da propriedade é definido automaticamente e não pode ser definido ou alterado pelo usuário, mesmo em algumas condições específicas (por exemplo, *somente leitura* para objetos herdados ou objetos descobertos);
- ***somente gravação*** - o valor da propriedade pode ser definido, mas não pode ser acessado posteriormente;
- ***constante*** - o valor da propriedade pode ser definido ao criar um objeto, mas não pode ser alterado posteriormente;
- ***suportado*** - o valor da propriedade não é obrigatório, mas pode ser definido em algumas condições específicas (por exemplo, *suportado* se `type` for definido como ”Verificação simples”, ”Verificação externa”, ”Agente SSH”, ”Agente TELNET” ou ”Agente HTTP”); observe, no entanto, que as propriedades *suportadas* ainda podem ser definidas para seus valores padrão, independentemente das condições;
- ***obrigatório*** - o valor da propriedade é obrigatório para todas as operações (exceto operações de obtenção) ou em algumas condições específicas (por exemplo, *obrigatório* para operações de criação; *obrigatório* se `operationtype` for definido como ”script global” e `opcommand_hst` não estiver definido).

Note:
Para operações de atualização, uma propriedade é considerada como ”definida” ao defini-la durante a operação de atualização.

As propriedades que não são marcadas com rótulos são opcionais.

Comportamento do parâmetro

Alguns dos parâmetros de operação são marcados com rótulos curtos para descrever seu comportamento para a operação. Os seguintes rótulos são usados:

- ***somente leitura*** - o valor do parâmetro é definido automaticamente e não pode ser definido ou alterado pelo usuário, mesmo em algumas condições específicas (por exemplo, *somente leitura* para objetos herdados ou objetos descobertos);
- ***somente gravação*** - o valor do parâmetro pode ser definido, mas não pode ser acessado depois;
- ***suportado*** - o valor do parâmetro não precisa ser definido, mas pode ser definido em algumas condições específicas (por exemplo, *suportado* se `operating_mode` do objeto Proxy estiver definido como ”proxy passivo”); observe, no entanto, que os parâmetros *suportados* ainda podem ser definidos para seus valores padrão, independentemente das condições;
- ***obrigatório*** - o valor do parâmetro deve ser definido.

Os parâmetros que não estão marcados com rótulos são opcionais.

Valor de ID reservado ”0” O valor de ID reservado ”0” pode ser usado para filtrar elementos e remover objetos referenciados. Por exemplo, para remover um proxy referenciado de um host, proxyid deve ser definido como 0 (”proxyid”: ”0”) ou para filtrar hosts monitorados pela opção server, proxyids deve ser definido como 0 (”proxyids”: ”0”).

Parâmetros comuns do método ”get” Os seguintes parâmetros são suportados por todos os métodos `get`:

Parâmetro	Tipo	Descrição
countOutput	boolean	Retorna o número de registros no resultado em vez dos dados reais.
editable	boolean	Se definido como `true`, retorna apenas objetos para os quais o usuário tem permissões de escrita. Padrão: `false`.
excludeSearch	boolean	Retorna resultados que não correspondem aos critérios fornecidos no parâmetro `search`.
filter	object	Retorna apenas os resultados que correspondem exatamente ao filtro fornecido. Aceita um objeto, onde as chaves são nomes de propriedades (por exemplo, propriedades do objeto Host em `host.get`, propriedades do objeto Item em `item.get`, etc.), e os valores são um único valor ou um array de valores para comparar. Não suporta propriedades do tipo de dado `text` data type. Observe que alguns métodos têm funcionalidades específicas para este parâmetro, que são descritas na página do método (por exemplo, o parâmetro `filter` em host.get também suporta propriedades da interface do Host).
limit	integer	Limita o número de registros retornados.

Parâmetro	Tipo	Descrição
output	query	Propriedades do objeto a serem retornadas. Observe que o ID do objeto (ou seja, `hostid`, `itemid`, etc.) é sempre incluído na resposta, mesmo que não seja especificado no parâmetro `output`. Padrão: `extend`.
preservekeys	boolean	Usa os IDs como chaves no array resultante.
search	object	Retorna resultados que correspondem ao padrão fornecido (case-insensitive). Aceita um objeto, onde as chaves são nomes de propriedades (por exemplo, propriedades do objeto Host em `host.get`, propriedades do objeto Item em `item.get`, etc.), e os valores são strings a serem pesquisadas. Se nenhuma opção adicional for fornecida, será realizada uma pesquisa `LIKE "%…%"`. Suporta apenas propriedades dos tipos de dado `string` e `text` data type. Observe que alguns métodos têm funcionalidades específicas para este parâmetro, que são descritas na página do método (por exemplo, o parâmetro `search` em host.get também suporta propriedades da interface do Host).
searchByAny	boolean	Se definido como `true`, retorna resultados que correspondem a qualquer um dos critérios fornecidos nos parâmetros `filter` ou `search` em vez de todos eles. Padrão: `false`.
searchWildcardsEnabled	boolean	Se definido como `true`, habilita o uso de "*" como caractere curinga no parâmetro `search`. Padrão: `false`.
sortfield	string/array	Ordena o resultado pelas propriedades fornecidas. Consulte a descrição do método get da API específica para uma lista de propriedades que podem ser usadas para ordenação. Macros não são expandidas antes da ordenação. Se nenhum valor for especificado, os dados serão retornados sem ordenação.
sortorder	string/array	Ordem da ordenação. Se um array for passado, cada valor será associado à propriedade correspondente fornecida no parâmetro `sortfield`. Valores possíveis: `ASC` - *(padrão)* ascendente; `DESC` - descendente.
startSearch	boolean	O parâmetro `search` irá comparar o início dos campos, ou seja, realizará uma pesquisa `LIKE "…%"`. Ignorado se `searchWildcardsEnabled` estiver definido como `true`.

Exemplos Verificação de permissão do usuário

O usuário tem permissão para gravar em hosts cujos nomes começam com "MySQL" ou "Linux"?

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "countOutput": true,
        "search": {
            "host": ["MySQL", "Linux"]
        },
        "editable": true,
        "startSearch": true,
        "searchByAny": true
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": "0",
    "id": 1
}
```

Note:
Resultado zero significa que não há hosts com permissões de leitura/gravação.

Contagem de não correspondências

Conte o número de hosts cujos nomes não contêm a substring "ubuntu"

Requisição:

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "countOutput": true,
        "search": {
            "host": "ubuntu"
        },
        "excludeSearch": true
    },
    "id": 1
}
```

Resposta:

```
{
    "jsonrpc": "2.0",
    "result": "44",
    "id": 1
}
```

Pesquisando hosts usando curingas

Encontre hosts cujo nome contenha a palavra "server" e que tenham portas de interface "10050" ou "10071". Classifique o resultado pelo nome do host em ordem decrescente e limite a 5 hosts.

Requisição:

```
{
    "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
}
```

Resposta:

```
{
    "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
}
```


Pesquisando hosts usando curingas com ”preservekeys”

Se você adicionar o parâmetro ”preservekeys” à solicitação anterior, o resultado será retornado como um array associativo, onde as chaves são os IDs dos objetos.

Requisição:

```
{
    "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
}
```

Resposta:

```
{
    "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
}
```

Apêndice 2. Alterações de 6.4 para 7.0

Alterações incompatíveis com versões anteriores alert

ZBXNEXT-6974 `alert.get`: Adicionada validação estrita dos parâmetros do método.

authentication

ZBXNEXT-8880 As propriedades `http_auth_enabled`, `http_login_form`, `http_strip_domains`, `http_case_sensitive` estarão disponíveis apenas se `$ALLOW_HTTP_AUTH` estiver habilitado no arquivo de configuração do frontend (*zabbix.conf.php*).

connector

ZBXNEXT-8735 Adicionada nova propriedade `item_value_type`, que é suportada se `data_type` estiver definido como "Valores de item" (0).
 ZBXNEXT-8735 Adicionada nova propriedade `attempt_interval`, que é suportada se `max_attempts` for maior que 1.

dashboard

ZBXNEXT-8316, ZBXNEXT-9193, ZBX-24488, ZBX-24490 Renomeado o tipo de widget de dashboard de `plaintext` para `itemhistory`, substituídos os campos do widget de dashboard `itemids.0`, `style`, `show_as_html` por `columns.0.itemid`, `layout`, `columns.0.display` e adicionados novos campos de widget de dashboard.
 ZBXNEXT-8496 Substituídos os campos do widget de dashboard `columns.0.timeshift`, `columns.0.aggregate_interval` por `columns.0.time_period._reference`, `columns.0.time_period.from`, `columns.0.time_period.from` no widget `tophosts`.
 ZBXNEXT-2299 Substituído o campo do widget de dashboard `unacknowledged` por dois novos campos `acknowledgement_status` e `acknowledged_by_me` no widget `problems`.
 ZBXNEXT-8245 Removido o campo do widget de dashboard `adv_conf` nos widgets `clock` e `item`.
 ZBXNEXT-8145 Alterada a nomenclatura dos campos do widget de dashboard: campos de dados complexos renomeados de `str.str.index1.index2` para `str.index1.str.index2` (por exemplo, `thresholds.0.threshold.1`, `ds.0.hosts.1`); campos que referenciam objetos do banco de dados renomeados de `str` para `str.index1` (por exemplo, `itemid.0`, `severities.0`).
 ZBXNEXT-8145 Substituído o campo do widget de dashboard `filter_widget_reference` por `sysmapid._reference` e removido o campo `source_type` no widget `map`.
 ZBXNEXT-8145 Substituído o campo do widget de dashboard `dynamic` por `override_hostid._reference` nos widgets `gauge`, `graph`, `graphprototype`, `item`, `plaintext` e `url`.
 ZBXNEXT-8145 Substituídos os campos do widget de dashboard `graph_time` por `time_period._reference`, `time_from` por `time_period.from`, `time_to` por `time_period.to` no widget `svggraph`.
 ZBXNEXT-9044 Alterado o intervalo de valores dos parâmetros do widget de dashboard `x` (de 0-23 para 0-71) e `y` (de 0-62 para 0-63), bem como `width` (de 1-24 para 1-72) e `height` (de 2-32 para 1-64).

discoveryrule

ZBXNEXT-7578 `discoveryrule.get`, `discoveryrule.create`, `discoveryrule.update`: As propriedades `headers` e `query_fields` mudaram de objeto indexado por nome para array de objetos com propriedades separadas `name` e `value`.

event

ZBXNEXT-6974 `event.get`: Adicionada validação estrita dos parâmetros do método.

host

ZBXNEXT-8758 A propriedade `proxyid` agora é *obrigatória* se `monitored_by` estiver definido como "Proxy"; as solicitações da API usando `proxyid` agora também devem usar `monitored_by`.
 ZBXNEXT-5878 `host.create`, `host.update`, `host.massAdd`, `host.massUpdate`, `host.massRemove`: Adicionada validação estrita dos parâmetros `groups`, `macros`, `tags`, `templates`.
 ZBXNEXT-8500 `host.get`: Removido o suporte para o parâmetro `proxy_hosts`.
 ZBXNEXT-8500 Propriedade `proxy_hostid` renomeada para `proxyid`.

hostgroup

ZBXNEXT-8702 `hostgroup.get`: Parâmetros renomeados de `selectHostPrototype` para `selectHostPrototypes`, `selectDiscoveryRule` para `selectDiscoveryRules`, `selectGroupDiscovery` para `selectGroupDiscoveries`; agora cada parâmetro retornará um array de objetos em vez de um objeto.
 ZBXNEXT-8702 `hostgroup.get`: Propriedades de saída

renomeadas de hostPrototype para hostPrototypes, discoveryRule para discoveryRules, groupDiscovery para groupDiscoveries.
 ZBXNEXT-8702 hostgroup.get: Removido o suporte para a propriedade groupid no parâmetro selectGroupDiscoveries (anteriormente - selectGroupDiscovery).

item

ZBXNEXT-7726 A propriedade params agora é *obrigatória* para etapas de pré-processamento do tipo "Verificar valor não suportado".
 ZBXNEXT-7578 item.get, item.create, item.update: As propriedades headers e query_fields mudaram de objeto indexado por nome para array de objetos com propriedades separadas name e value.

protótipo de item

ZBXNEXT-7726 A propriedade params agora é *obrigatória* para etapas de pré-processamento do tipo "Verificar valor não suportado".
 ZBXNEXT-7578 itemprototype.get, itemprototype.create, itemprototype.update: As propriedades headers e query_fields mudaram de objeto indexado por nome para array de objetos com propriedades separadas name e value.

problem

ZBXNEXT-6974 problem.get: Adicionada validação estrita dos parâmetros do método.

proxy

ZBXNEXT-8500 Removidas as propriedades de objeto interface e auto_compress.
 ZBXNEXT-8500 Adicionadas as propriedades address e port para proxies Zabbix passivos.
 ZBXNEXT-8500 Renomeadas as propriedades host para name, status para operating_mode, proxy_address para allowed_addresses.
 ZBXNEXT-8500 Alterados os valores da propriedade operating_mode (anteriormente - status) (0 - proxy ativo, 1 - proxy passivo).
 ZBXNEXT-8500 proxy.get: Removido o suporte para o parâmetro selectInterface.
 ZBXNEXT-8500 proxy.create e proxy.update: Removido o suporte para o parâmetro interface.

script

ZBXNEXT-8880 script.create e script.update: O valor do parâmetro execute_on "1" (executar no Zabbix server) será suportado apenas se a execução de scripts globais estiver habilitada no Zabbix server.
 ZBXNEXT-8121 script.getscriptsbyhosts: O método não aceita mais um array de IDs de host. Agora aceita um objeto com os seguintes parâmetros: hostid, scriptid, manualinput.
 ZBXNEXT-8121 script.getscriptsbyevents: O método não aceita mais um array de IDs de evento. Agora aceita um objeto com os seguintes parâmetros: eventid, scriptid, manualinput.

tarefa

ZBXNEXT-8500 Renomeada a propriedade proxy_hostid para proxyid.

templatedashboard

ZBXNEXT-9044 Alterado o intervalo de valores dos parâmetros do widget do dashboard x (de 0-23 para 0-71) e y (de 0-62 para 0-63), bem como width (de 1-24 para 1-72) e height (de 2-32 para 1-64).

user

ZBXNEXT-8760 user.update: A propriedade userdirectoryid para usuário foi tornada somente leitura.
 ZBXNEXT-8760 user.get: Adicionada nova propriedade somente leitura userdirectory_mediaid para mídia criada por ação de provisionamento.

userdirectory

ZBXNEXT-8760 userdirectory.create, userdirectory.update: Adicionadas novas propriedades aos mapeamentos de mídia - active, severity, period.

usergroup

ZBXNEXT-8760 usergroup.update: Adicionada restrição nas alterações de usuários do grupo para usuários provisionados.

Outras alterações e correções de bugs action

ZBXNEXT-6524 Adicionado suporte para dois novos valores na propriedade operationtype (13 - Adicionar tags de host, 14 - Remover tags de host) e nova propriedade optag para dois tipos de ação eventsource (1 - Descoberta, 2 - Autoregistro) disponíveis apenas na propriedade operations.
 ZBX-21850 action.get: As condições de filtro serão classificadas na ordem de sua colocação na fórmula.

auditlog

ZBXNEXT-8541 Adicionada nova ação de entrada audit log (12 - Push) e tipo de recurso (53 - History).

authentication

ZBXNEXT-6876 Adicionadas novas propriedades `mfa_status` e `mfaid`.

mfa

ZBXNEXT-6876 Adicionada nova API MFA com métodos `mfa.create`, `mfa.update`, `mfa.get`, `mfa.delete`.

dashboard

ZBXNEXT-8956 Adicionados novos campos de widget de dashboard `groupids._reference`, `hostids._reference`, `itemid._reference`, `graphid._reference`, `reference` em múltiplos widgets.
 ZBXNEXT-9057 Adicionado novo campo de widget de dashboard `show_software_update_check_details` no widget `systeminfo`.
 ZBXNEXT-8686 Adicionado novo tipo de widget de dashboard `itemnavigator`.
 ZBXNEXT-8685 Adicionado novo tipo de widget de dashboard `hostnavigator`.
 ZBXNEXT-8683 Adicionado novo tipo de widget de dashboard `honeycomb`.
 ZBXNEXT-8907 Adicionado novo campo de widget de dashboard `legend_lines_mode` no widget `graph`.
 ZBXNEXT-8496 Adicionados novos campos de widget de dashboard `aggregate_function`, `history`, `time_period._reference`, `time_period.from`, `time_period.from` no widget `item`.
 ZBXNEXT-7736 Adicionado novo valor de campo de widget de dashboard `interface_type` (5 - Zabbix Agent(active checks)) no widget `hostavail`.
 ZBXNEXT-7736 Adicionado novo campo de widget de dashboard `only_totals` no widget `hostavail`.
 ZBXNEXT-7687 Adicionado novo campo de widget de dashboard `maintenance` no widget `tophosts`.
 ZBXNEXT-6974 Adicionado novo tipo de widget de dashboard `toptriggers`.
 ZBXNEXT-743 Adicionado novo tipo de widget de dashboard `gauge`.
 ZBXNEXT-8331, ZBXNEXT-8145, ZBXNEXT-8908, ZBXNEXT-8907 Adicionado novo tipo de widget de dashboard `piechart`.
 ZBXNEXT-8331 Adicionado novo campo de widget de dashboard `legend_aggregation` no widget `graph`.
 ZBXNEXT-8145 Adicionado novo campo de widget de dashboard `reference` nos widgets `svggraph`, `graph` e `graphprototype`.
 ZBXNEXT-8145 Adicionados novos campos de widget de dashboard `time_period._reference`, `time_period.from`, `time_period.to` nos widgets `actionlog`, `graph`, `graphprototype` e `toptriggers`.

dcheck

ZBXNEXT-8079 Adicionada nova propriedade `allow_redirect`.

discoveryrule

ZBXNEXT-9150 Adicionado novo tipo de regra LLD `type` (22 - Browser).
 ZBXNEXT-8645 Adicionado novo tipo de pré-processamento de regra LLD "SNMP get value" (30).
 ZBXNEXT-8645 A propriedade `timeout` agora também é suportada se o `type` do item for definido como "SNMP agent" (20) e `snmp_oid` começar com "`get[`".
 ZBXNEXT-1096 A propriedade `timeout` agora é suportada se o `type` do item for definido como "Zabbix agent" (0), "Simple check" (3) e `key_` não começar com "`vmware.`" e "`icmpping`", "Zabbix agent (active)" (7), "External check" (10), "Database monitor" (11), "SSH agent" (13), "TELNET agent" (14), "SNMP agent" (20) e `snmp_oid` começar com "`walk[`".
 ZBXNEXT-7726 Adicionado suporte para o tipo de pré-processamento de regra LLD "Matches regular expression" (14).
 ZBXNEXT-6986 Método `discoveryrule.copy` obsoleto.
 ZBXNEXT-7578 Agora é possível armazenar mais dados para a propriedade `query_fields`, ter entradas repetidas de `header` e `query_fields`.
 ZBXNEXT-2020 Adicionadas novas propriedades `lifetime_type`, `enabled_lifetime` e `enabled_lifetime_type`.
 ZBXNEXT-2020 Valor padrão do parâmetro `lifetime` alterado de 30d para 7d.
 ZBX-21850 `discoveryrule.get`: As condições de filtro serão classificadas na ordem de sua colocação na fórmula.

drule

ZBXNEXT-2732 Adicionada nova propriedade `concurrency_max`.
 ZBXNEXT-8500 Renomeado `proxy_hostid` para `proxyid`

event

ZBXNEXT-6974 `event.get`: Adicionados novos parâmetros `selectAcknowledges`, `selectAlerts`, `trigger_severities` e `groupBy`.
 ZBXNEXT-6974 `event.get`: Parâmetros obsoletos `select_acknowledges` e `select_alerts`.
 ZBXNEXT-2299, ZBX-23240 `event.get`: Adicionados novos parâmetros `action` e `action_userids`.

gráfico

ZBXNEXT-2020 `graph.get`: O método agora também oferece suporte à propriedade `status` se o parâmetro `selectGraphDiscovery` for usado.

history

ZBXNEXT-9193 `history.get`: O parâmetro `sortfield` agora suporta `ns`.
 ZBXNEXT-8541 Adicionado novo método `history.push`.

host

ZBXNEXT-8758 Adicionadas novas propriedades `monitored_by` e `proxy_groupid`.
 ZBXNEXT-8758 Adicionada nova propriedade somente leitura `assigned_proxyid`.
 ZBXNEXT-8758 `host.get`: Adicionado novo parâmetro `proxy_groupids`.
 ZBXNEXT-2020 `host.get`: O método agora também suporta as propriedades `status`, `ts_disable` e `disable_source` se o parâmetro `selectHostDiscovery` for usado.

hostgroup

ZBXNEXT-2020 `hostgroup.get`: O método agora também oferece suporte à propriedade `status` se o parâmetro `selectGroupDiscoverie` for usado.

hostprototype

ZBX-23284 `hostprototype.get`, `hostprototype.update`: retornado o suporte para a propriedade `group_prototypeid` de objetos de protótipo de grupo.

item

ZBXNEXT-9150 Adicionado novo `type` de item (22 - Browser).
 ZBXNEXT-7460 Adicionada nova propriedade somente leitura `name_resolved`.
 ZBXNEXT-8645 Adicionado novo tipo de pré-processamento de item "SNMP get value" (30).
 ZBXNEXT-8645 A propriedade `timeout` agora também é suportada se o `type` do item for definido como "SNMP agent" (20) e `snmp_oid` começar com "`get[`".
 ZBXNEXT-1096 A propriedade `timeout` agora é suportada se o `type` do item for definido como "Zabbix agent" (0), "Simple check" (3) e `key_` não começar com "`vmware.`" e "`icmpping`", "Zabbix agent (active)" (7), "External check" (10), "Database monitor" (11), "SSH agent" (13), "TELNET agent" (14), "SNMP agent" (20) e `snmp_oid` começar com "`walk[`".
 ZBXNEXT-7726 Adicionados os componentes *scope* e *pattern* aos parâmetros para etapas do tipo "Check for not supported value".
 ZBXNEXT-7578 Agora é possível armazenar mais dados para a propriedade `query_fields`, ter entradas repetidas de `header` e `query_fields`.
 ZBXNEXT-2020 `item.get`: O método agora também suporta as propriedades `status`, `ts_disable` e `disable_source` se o parâmetro `selectItemDiscovery` for usado.

protótipo de item

ZBXNEXT-9150 Adicionado novo tipo de protótipo de item `type` (22 - Browser).
 ZBXNEXT-8645 Adicionado novo tipo de pré-processamento de protótipo de item "SNMP get value" (30).
 ZBXNEXT-8645 A propriedade `timeout` agora também é suportada se o `type` do item for definido como "SNMP agent" (20) e `snmp_oid` começar com "`get[`".
 ZBXNEXT-1096 A propriedade `timeout` agora é suportada se o `type` do item for definido como "Zabbix agent" (0), "Simple check" (3) e `key_` não começar com "`vmware.`" e "`icmpping`", "Zabbix agent (active)" (7), "External check" (10), "Database monitor" (11), "SSH agent" (13), "TELNET agent" (14), "SNMP agent" (20) e `snmp_oid` começar com "`walk[`".
 ZBXNEXT-7726 Adicionados os componentes *scope* e *pattern* aos parâmetros para etapas do tipo "Check for not supported value".
 ZBXNEXT-7578 Agora é possível armazenar mais dados para a propriedade `query_fields`, ter entradas repetidas de `header` e `query_fields`.

mediatype

ZBXNEXT-4165 `mediatype.get`: Adicionado novo parâmetro `selectActions`.
 ZBXNEXT-9138 `mediatype.get`, `mediatype.create`, `mediatype.update`: Adicionada nova propriedade `message_format`.
 ZBXNEXT-9138 `mediatype.get`, `mediatype.create`, `mediatype.update`: Propriedade `content_type` obsoleta.

correlation

ZBX-21850 `correlation.get`: As condições de filtro serão classificadas na ordem de sua colocação na fórmula.

problem

ZBXNEXT-2299, ZBX-23240 `problem.get`: Adicionados novos parâmetros `action` e `action_userids`.

proxy

ZBXNEXT-9150 Adicionada nova propriedade `timeout_browser`.
 ZBXNEXT-8758 Adicionada nova propriedade somente leitura `state`.
 ZBXNEXT-8758 `proxy.get`: Adicionados novos parâmetros `proxy_groupids`, `selectAssignedHosts` e `selectProxyGroup`.
 ZBXNEXT-8758 `proxy.get`: O parâmetro `selectHosts` agora suporta `count`.
 ZBXNEXT-1096 Adicionadas novas propriedades `custom_timeouts`, `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`.
 ZBXNEXT-8500 Adicionadas propriedades `address` e `port` para proxies Zabbix passivos.

proxygroup

ZBXNEXT-8758 Adicionada nova API proxygroup.

role

ZBXNEXT-8758 Adicionado novo elemento de UI `administration.proxy_groups`.

script

ZBXNEXT-8121 Adicionadas novas propriedades `manualinput`, `manualinput_prompt`, `manualinput_validator`, `manualinput_valida` `manualinput_default_value`.
 ZBXNEXT-8121 `script.execute`: Adicionado novo parâmetro `manualinput`.

configurações

ZBXNEXT-9150 Adicionada nova propriedade `timeout_browser`.
 ZBXNEXT-8837 Adicionada nova propriedade `auditlog_mode`.
 ZBXNEXT-1096 Adicionadas novas propriedades `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`.

templatedashboard

ZBXNEXT-8686 Adicionado novo tipo de widget de dashboard de template `itemnavigator`.
 ZBXNEXT-8685 Adicionado novo tipo de widget de dashboard de template `hostnavigator`.
 ZBXNEXT-8683 Adicionado novo tipo de widget de dashboard de template `honeycomb`.
 ZBXNEXT-6974 Adicionado novo tipo de widget de dashboard de template `toptriggers`.
 ZBXNEXT-743 Adicionado novo tipo de widget de dashboard de template `gauge`.
 ZBXNEXT-8086 Adicionados novos tipos de widget de dashboard de template `actionlog`, `dataover`, `discovery`, `favgraphs`, `favmaps`, `hostavail`, `map`, `navtree`, `problemhosts`, `problems`, `problemsbysv`, `slareport`, `svggraph`, `systeminfo`, `tophosts`, `trigover`, `web`.
 ZBXNEXT-8086 Adicionados novos tipos de campo de widget de dashboard de template (8 - Mapa, 9 - Serviço, 10 - SLA, 11 - Usuário, 12 - Ação, 13 - Tipo de mídia).
 ZBXNEXT-8331 Adicionado novo tipo de widget de dashboard de template `piechart`.

gatilho

ZBXNEXT-2020 `trigger.get`: O método agora também oferece suporte às propriedades `status`, `ts_disable` e `disable_source` se o parâmetro `selectTriggerDiscovery` for usado.

user

ZBXNEXT-6876 Adicionado novo método `user.resettotp`.
 ZBXNEXT-6876 `user.login`: O método agora também retorna `mfaid` se o parâmetro `userData` for usado.

usergroup

ZBXNEXT-6876 Adicionadas novas propriedades `mfa_status` e `mfaid`.
 ZBXNEXT-6876 `usergroup.get`: Adicionados novos parâmetros `mfa_status` e `mfaids`.

Apêndice 3. Mudanças na versão 7.0

7.0.19 dashboard

ZBXNEXT-10143 Adicionado novo campo de widget de dashboard `highlight_row` no widget problems.

tarefa

ZBX-26202 `task.create`: usuários dos tipos *Admin* e *User* agora podem criar tarefas 'Executar agora'.

7.0.14 dashboard

ZBX-25929 Removidos os campos do widget do dashboard `date_size`, `time_size`, `tzone_size` no widget `clock`.

7.0.11 API

ZBX-22952 Adicionado suporte para aceitar cabeçalhos de forma case-insensitive (por exemplo, `authorization`, `Authorization` e `AUTHORIZATION` são tratados da mesma forma).

7.0.10 alert

ZBX-26023 `alert.get`: usuários dos tipos *Admin* e *User* agora podem recuperar dados de alerta do tipo "message" (0) sobre usuários em seu grupo de usuários.

drule

ZBX-23760 `drule.update`: Não é mais possível excluir uma verificação de regra de descoberta que é usada em uma condição de ação. Nesse caso, uma mensagem de erro é retornada em vez de desabilitar automaticamente a ação.

host

ZBX-23760 `host.delete`: Não é mais possível excluir um host que é usado em uma condição ou operação de ação. Nesses casos, uma mensagem de erro é retornada em vez de desabilitar automaticamente a ação.

hostgroup

ZBX-23760 `hostgroup.delete`: Não é mais possível excluir um grupo de hosts que está sendo usado em uma condição ou operação de ação. Nesses casos, uma mensagem de erro é retornada em vez de desabilitar automaticamente a ação.

template

ZBX-23760 `template.delete`: Não é mais possível excluir um template que está sendo usado em uma condição ou operação de ação. Nesses casos, uma mensagem de erro é retornada em vez de desabilitar automaticamente a ação.

trigger

ZBX-23760 `trigger.delete`: Não é mais possível excluir um trigger que é usado em uma condição de ação. Nesse caso, uma mensagem de erro é retornada em vez de desabilitar automaticamente a ação.

7.0.9 alert

ZBX-26258 `alert.get`: O parâmetro `selectUsers` agora retorna dados com base nas restrições adicionadas ao método `user.get`.
 ZBX-26258 `alert.get`: Usuários dos tipos *Admin* e *User* agora podem recuperar dados de alerta do tipo "message" (0) apenas sobre seu próprio usuário.

mediatype

ZBX-26258 `mediatype.get`: O parâmetro `selectUsers` agora retorna dados com base nas restrições adicionadas ao método `user.get`.

role

ZBX-26258 `role.get`: O parâmetro `selectUsers` agora retorna dados com base nas restrições adicionadas ao método `user.get`.

user

ZBX-26258 Adicionada nova propriedade `provisioned` ao objeto User e ao objeto user Media.
 ZBX-26258 As propriedades `ts_provisioned`, `userdirectoryid` e `userdirectory_mediaid` agora são suportadas apenas para usuários do tipo *Super admin*.
 ZBX-26258 `user.get`: Ao solicitar informações de mídia do usuário, permissões ou informações de função, usuários dos tipos *Admin* e *User* podem recuperar dados apenas sobre seu próprio usuário.
 ZBX-26258 `user.get`: Usuários dos tipos *Admin* e *User* agora podem recuperar apenas as seguintes propriedades do objeto User para seu próprio usuário: `userid`, `attempt_clock`, `attempt_failed`, `attempt_ip`, `autologin`, `autologout`, `lang`, `name`, `provisioned`, `refresh`, `roleid`, `rows_per_page`, `surname`, `theme`, `timezone`, `url`, `username`. Para usuários em seu grupo de usuários: `userid`, `name`, `surname`, `username`.

usergroup

ZBX-26258 As propriedades `mfaid` e `userdirectoryid` agora são suportadas apenas para usuários do tipo *Super admin*.
 ZBX-26258 `usergroup.get`: O parâmetro `mfaids` agora é suportado apenas para usuários do tipo *Super admin*.
 ZBX-26258 `usergroup.get`: O parâmetro `selectUsers` agora retorna dados com base nas restrições adicionadas ao método `user.get`.

7.0.7 API

ZBX-25732 Adicionado suporte para o cabeçalho de requisição HTTP `Authorization` e para o método de requisição HTTP `OPTIONS` em requisições de origem cruzada (CORS).

7.0.6 alert

ZBXNEXT-9404 `alert.get`: usuários do tipo *Usuário* agora podem recuperar as seguintes propriedades do objeto Media type com o parâmetro `selectMediatypes`: `mediatypeid`, `name`, `type`, `status`, `maxattempts`, `description`.

mediatype

ZBXNEXT-9404 `mediatype.get`: usuários do tipo *Usuário* agora podem recuperar as seguintes propriedades do objeto Media type: `mediatypeid`, `name`, `type`, `status`, `maxattempts`, `description`.

user

ZBXNEXT-9404 `user.get`: usuários do tipo *Usuário* agora podem recuperar as seguintes propriedades do objeto Media type com o parâmetro `selectMediatypes`: `mediatypeid`, `name`, `type`, `status`, `maxattempts`, `description`.

7.0.4 alert

ZBX-25385 `alert.get`: usuários do tipo *Admin* agora podem recuperar apenas as seguintes propriedades do objeto Media type com o parâmetro `selectMediatypes`: `mediatypeid`, `name`, `type`, `status`, `maxattempts`, `description`.

mediatype

ZBX-25385 `mediatype.get`: O parâmetro `selectMessageTemplates` agora é suportado apenas para usuários do tipo *Super admin*.
 ZBX-25385 `mediatype.get`: Usuários do tipo *Admin* agora podem recuperar apenas as seguintes propriedades do objeto Media type: `mediatypeid`, `name`, `type`, `status`, `maxattempts`, `description`.
 ZBX-25385 `mediatype.get`: Ao solicitar informações relacionadas ao usuário dos tipos de mídia, usuários do tipo *Admin* agora podem recuperar apenas dados sobre seu próprio usuário.

user

ZBX-25385 `user.get`: usuários do tipo *Admin* agora podem recuperar apenas as seguintes propriedades do objeto Media type com o parâmetro `selectMediatypes`: `mediatypeid`, `name`, `type`, `status`, `maxattempts`, `description`.

7.0.1 dashboard

ZBXNEXT-9215 Adicionado novo campo de widget de dashboard `ds.0.itemids.0._reference` nos widgets `svggraph` e `piechart`.
 ZBXNEXT-8331 Adicionado novo campo de widget de dashboard `stroke` no widget `piechart`.

21 Extensões

Visão geral Embora o Zabbix ofereça uma grande variedade de recursos, sempre há espaço para funcionalidades adicionais. As extensões são uma maneira conveniente de modificar e aprimorar os recursos de monitoramento do Zabbix sem alterar seu código-fonte.

Você pode ampliar a funcionalidade do Zabbix usando as opções de extensão incorporadas (itens trapper, parâmetros de usuário, etc.) ou usando ou criando extensões personalizadas (módulos carregáveis, plugins, etc.).

Esta seção fornece uma visão geral com referências a todas as opções de extensão do Zabbix.

Coleta de dados com comandos personalizados Itens trapper

Itens trapper são itens que aceitam dados recebidos em vez de consultá-los. Itens trapper são úteis para enviar dados específicos para o server ou proxy Zabbix, por exemplo, dados periódicos de disponibilidade e desempenho no caso de scripts de usuário de longa execução. O envio de dados para o server ou proxy Zabbix é possível usando o utilitário Zabbix sender ou o protocolo do Zabbix sender. O envio de dados para o server Zabbix também é possível usando o método de API `history.push`.

Verificações externas

Uma verificação externa é um item para execução de verificações executando um executável, por exemplo, um shell script ou um binário.

As verificações externas são executadas pelo servidor Zabbix ou pelo proxy (quando o host é monitorado pelo proxy), e não requerem um agente em execução no host que está sendo monitorado.

Parâmetros do usuário

Um parâmetro de usuário é um comando definido pelo usuário (associado a uma chave definida pelo usuário) que, quando executado, pode recuperar os dados necessários do host em que o Zabbix Agent está sendo executado. Os parâmetros do usuário são úteis para configurar itens do agente ou do agente 2 que não estão predefinidos no Zabbix.

Itens do agent Zabbix `system.run[]`

O item do agent Zabbix `system.run[]` é um item para um comando definido pelo usuário (associado a uma chave predefinida `system.run[]`, por exemplo, `system.run[myscript.sh]`) que pode ser executado no host onde o agent Zabbix está em execução.

Nota: os itens `system.run[]` estão desabilitados por padrão e, se usados, devem ser habilitados (permitidos) e definidos no arquivo de configuração do agent Zabbix ou agent 2 (parâmetro de configuração `AllowKey`).

Comandos definidos pelo usuário em itens como verificações externas, parâmetros do usuário e itens do Zabbix Agent `system.run[]` são executados pelo usuário do sistema operacional utilizado para executar os componentes do Zabbix. Para executar esses comandos, esse usuário deve ter as permissões necessárias.

Items de agente HTTP

O item agente HTTP é um item para executar solicitações de dados via HTTP/HTTPS. Os itens de agente HTTP são úteis para enviar solicitações para endpoints HTTP para recuperar dados de serviços como *Elasticsearch* e *OpenWeatherMap*, para verificar o status da API do Zabbix ou o status do servidor web Apache ou Nginx, etc. Os itens de agente HTTP (com trapping habilitado) também podem funcionar como itens trapper.

Itens de script

Um item de script é um item para execução de código JavaScript definido pelo usuário que recupera dados por HTTP/HTTPS. Os itens de script são úteis quando a funcionalidade fornecida pelos itens do agente HTTP não é suficiente. Por exemplo, em cenários de coleta de dados exigentes que requerem várias etapas ou lógica complexa, um item de script pode ser configurado para fazer uma chamada HTTP e, em seguida, processar os dados recebidos, e, em seguida, passar o valor transformado para uma segunda chamada HTTP.

::: dica de nota Os itens de agente HTTP e os itens de script são suportados pelo Zabbix Server e pelo proxy, e não requerem um agente em execução no host que está sendo monitorado.

Extensões avançadas Módulos carregáveis

Os módulos carregáveis, escritos em C, são uma opção versátil versátil e voltada para o desempenho para estender a funcionalidade dos componentes do Zabbix (servidor, proxy, agente) em plataformas UNIX. Um módulo carregável é basicamente uma biblioteca compartilhada usada pelo daemon do Zabbix e carregada na inicialização. A biblioteca deve conter determinadas funções, para que um processo do Zabbix possa detectar que o arquivo é, de fato, um módulo que pode ser carregado e com o qual pode trabalhar.

Os módulos carregáveis têm vários benefícios, inclusive a capacidade de adicionar novas métricas ou implementar qualquer outra lógica (por exemplo, Zabbix history data export), ótimo desempenho e a opção de desenvolver, usar e compartilhar a funcionalidade que eles fornecem. Isso contribui para uma manutenção sem problemas e ajuda a fornecer novas funcionalidades de forma mais fácil e independente da base de código principal do Zabbix.

Os módulos carregáveis são especialmente úteis em uma configuração de monitoramento complexa. Ao monitorar sistemas incorporados, ter um grande número de parâmetros monitorados ou scripts pesados com lógica complexa ou longo tempo de inicialização, extensões como parâmetros de usuário, itens do agente Zabbix `system.run[]` e verificações externas terão um impacto no desempenho. Os módulos carregáveis oferecem uma maneira de estender a funcionalidade do Zabbix sem sacrificar o desempenho.

Plugins

Plugins estendem as capacidades de monitoramento do Zabbix agent 2. Escritos na linguagem de programação Go, eles fornecem uma alternativa aos módulos carregáveis (escritos em C).

Um plugin é um pacote Go que define a estrutura e implementa uma ou várias interfaces de plugin (*Exporter*, *Collector*, *Configurator*, *Runner*, *Watcher*). Dois tipos de plugins do Zabbix agent 2 são suportados:

- Plugins embutidos (suportados desde o Zabbix 4.4.0)
- Plugins carregáveis (suportados desde o Zabbix 6.0.0)

Veja a lista de plugins fornecidos pelo Zabbix.

Para instruções e tutoriais sobre como criar seus próprios plugins, consulte o Centro de desenvolvedores.

Personalização de alertas Webhooks

Um webhook é um tipo de mídia do Zabbix que fornece uma opção para estender os recursos de alerta do Zabbix para softwares externos, como sistemas de helpdesk, chats ou messengers. Da mesma forma que os itens de script, os webhooks são úteis para fazer chamadas HTTP usando código JavaScript personalizado, por exemplo, para enviar notificações para diferentes plataformas, como Microsoft Teams, Discord e Jira. Também é possível retornar alguns dados (por exemplo, sobre tíquetes de helpdesk criados) que são exibidos no Zabbix.

Os webhooks existentes estão disponíveis no [repositório Git] do Zabbix (https://git.zabbix.com/projects/ZBX/repos/zabbix/browse/templates/n
Para o desenvolvimento de webhooks personalizados, consulte Diretrizes de desenvolvimento de webhooks.

Scripts de alerta

Um script de alerta é um tipo de mídia do Zabbix que fornece uma opção para criar uma maneira alternativa (script) de lidar com os alertas do Zabbix. Os scripts de alerta são úteis se você não estiver satisfeito com os tipos de mídia existentes para o envio de alertas no Zabbix.

Personalização de front-end Temas personalizados

É possível alterar a aparência visual do frontend do Zabbix usando temas personalizados. Veja as instruções para criar e aplicar seus próprios temas.

Módulos de front-end

Os módulos front-end oferecem uma opção para ampliar a funcionalidade do front-end do Zabbix adicionando módulos de terceiros ou desenvolvendo seus próprios módulos. Com os módulos de front-end, você pode adicionar novos itens de menu, suas respectivas visualizações, ações, etc.

Scripts globais Um script global é um conjunto de comandos definido pelo usuário que pode ser executado em um alvo de monitoramento (pelo interpretador shell (/bin/sh)), dependendo do escopo configurado e das permissões do usuário. Os scripts globais podem ser configurados para as seguintes ações:

- Ação operação
- Ação manual de host
- Ação manual de evento

Os scripts globais são úteis em muitos casos. Por exemplo, se configurado para operações de ação ou ações manuais do host, você pode usar scripts globais para executar automática ou manualmente [comandos remotos] (/manual/config/notifications/action/operation/remote_command) como reiniciar um aplicativo (servidor da Web, middleware, CRM etc.) ou liberar espaço em disco (remover arquivos antigos, limpar `/tmp`, etc.). Ou, outro exemplo, se configurado para ações de eventos manuais, você pode usar scripts globais para gerenciar tíquetes de problemas em sistemas externos.

Os scripts globais podem ser executados pelo Zabbix Server, proxy ou agente.

Os comandos definidos pelo usuário são executados a partir do usuário do sistema operacional utilizado para executar os componentes do Zabbix. Para executar esses comandos, esse usuário deve ter as permissões necessárias.

API do Zabbix a API do Zabbix é uma API baseada em HTTP que faz parte do front-end do Zabbix. Com a API do Zabbix, você pode realizar qualquer uma das seguintes operações:

- Recuperar e modificar programaticamente a configuração do Zabbix.
- Importar e exportar a configuração do Zabbix.
- Acessar dados históricos e de tendências do Zabbix.
- Configurar aplicativos para trabalhar com o Zabbix.
- Integrar o Zabbix com software de terceiros.
- Automatizar tarefas de rotina.

A API do Zabbix consiste em uma multiplicidade de métodos que são nominalmente agrupados em APIs separadas. Cada método executa uma tarefa específica. Para conhecer os métodos disponíveis, bem como uma visão geral das funções fornecidas pela API do Zabbix, consulte API do Zabbix Referência do método.

1 Módulos carregáveis

Visão geral

Módulos carregáveis oferecem uma opção orientada para desempenho para estender a funcionalidade do Zabbix.

Você pode estender a funcionalidade do Zabbix de várias maneiras, por exemplo, com parâmetros de usuário, verificações externas e itens do agent Zabbix system.run. Eles funcionam muito bem, mas têm uma grande desvantagem, que é o fork(). O Zabbix precisa criar um novo processo toda vez que lida com uma métrica de usuário, o que não é bom para o desempenho. Normalmente não é um grande problema, no entanto, pode ser uma questão séria ao monitorar sistemas embarcados, ter um grande número de parâmetros monitorados ou scripts pesados com lógica complexa ou longo tempo de inicialização.

O suporte a módulos carregáveis oferece maneiras de estender o agent, server e proxy do Zabbix sem sacrificar o desempenho.

Um módulo carregável é basicamente uma biblioteca compartilhada usada pelo daemon do Zabbix e carregada na inicialização. A biblioteca deve conter certas funções, para que um processo do Zabbix possa detectar que o arquivo é realmente um módulo que pode carregar e trabalhar.

Módulos carregáveis têm vários benefícios. Ótimo desempenho e capacidade de implementar qualquer lógica são muito importantes, mas talvez a vantagem mais importante seja a capacidade de desenvolver, usar e compartilhar módulos do Zabbix. Isso contribui para uma manutenção sem problemas e ajuda a entregar novas funcionalidades de forma mais fácil e independente do código base principal do Zabbix.

O licenciamento de módulos e a distribuição em forma binária são regidos pela licença AGPL-3.0 (os módulos são vinculados ao Zabbix em tempo de execução e usam os headers do Zabbix; todo o código do Zabbix é licenciado sob AGPL-3.0 desde o Zabbix 7.0). A compatibilidade binária não é garantida pelo Zabbix.

A estabilidade da API do módulo é garantida durante um ciclo de release LTS (Long Term Support) do Zabbix. A estabilidade da API do Zabbix não é garantida (tecnicamente é possível chamar funções internas do Zabbix a partir de um módulo, mas não há garantia de que tais módulos funcionarão).

API do módulo

Para que uma biblioteca compartilhada seja tratada como um módulo Zabbix, ela deve implementar e exportar várias funções. Atualmente, existem seis funções na API do módulo Zabbix, das quais apenas uma é obrigatória e as outras cinco são opcionais.

Interface obrigatória

A única função obrigatória é **zbx_module_api_version()**:

```
int zbx_module_api_version(void);
```

Esta função deve retornar a versão da API implementada por este módulo e, para que o módulo seja carregado, esta versão deve corresponder à versão da API do módulo suportada pelo Zabbix. A versão da API do módulo suportada pelo Zabbix é

ZBX_MODULE_API_VERSION. Portanto, esta função deve retornar esta constante. A antiga constante ZBX_MODULE_API_VERSION_ONE usada para este propósito agora está definida como igual a ZBX_MODULE_API_VERSION para preservar a compatibilidade do código-fonte, mas seu uso não é recomendado.

Interface opcional

As funções opcionais são **zbx_module_init()**, **zbx_module_item_list()**, **zbx_module_item_timeout()**, **zbx_module_history_write_cbs()** e **zbx_module_uninit()**:

```
int zbx_module_init(void);
```

Esta função deve realizar a inicialização necessária para o módulo (se houver). Se for bem-sucedida, deve retornar ZBX_MODULE_OK. Caso contrário, deve retornar ZBX_MODULE_FAIL. Neste último caso, o Zabbix não será iniciado.

```
ZBX_METRIC  *zbx_module_item_list(void);
```

Esta função deve retornar uma lista de items suportados pelo módulo. Cada item é definido em uma estrutura ZBX_METRIC, veja a seção abaixo para detalhes. A lista é terminada por uma estrutura ZBX_METRIC com o campo ”key” igual a NULL.

```
void    zbx_module_item_timeout(int timeout);
```

Se o módulo exportar **zbx_module_item_list()**, então esta função é usada pelo Zabbix para especificar as configurações de timeout no arquivo de configuração do Zabbix que as verificações de item implementadas pelo módulo devem obedecer. Aqui, o parâmetro ”timeout” está em segundos.

```
ZBX_HISTORY_WRITE_CBS   zbx_module_history_write_cbs(void);
```

Esta função deve retornar funções de callback que o servidor Zabbix usará para exportar o histórico de diferentes tipos de dados. As funções de callback são fornecidas como campos da estrutura ZBX_HISTORY_WRITE_CBS, os campos podem ser NULL se o módulo não estiver interessado no histórico de determinado tipo.

```
int zbx_module_uninit(void);
```

Esta função deve realizar a desinicialização necessária (se houver), como liberar recursos alocados, fechar descritores de arquivos, etc.

Todas as funções são chamadas uma vez na inicialização do Zabbix quando o módulo é carregado, com exceção de zbx_module_uninit(), que é chamada uma vez no desligamento do Zabbix quando o módulo é descarregado.

Definindo items

Cada item é definido em uma estrutura ZBX_METRIC:

```
typedef struct
{
    char        *key;
    unsigned    flags;
    int     (*function)();
    char        *test_param;
}
ZBX_METRIC;
```

Aqui, **key** é a chave do item (por exemplo, ”dummy.random”), **flags** é CF_HAVEPARAMS ou 0 (dependendo se o item aceita parâmetros ou não), **function** é uma função C que implementa o item (por exemplo, ”zbx_module_dummy_random”), e **test_param** é a lista de parâmetros a ser usada quando o agent Zabbix é iniciado com o parâmetro ”-p” (por exemplo, ”1,1000”, pode ser NULL). Uma definição de exemplo pode ser assim:

```
static ZBX_METRIC keys[] =
{
    { "dummy.random", CF_HAVEPARAMS, zbx_module_dummy_random, "1,1000" },
    { NULL }
}
```

Cada função que implementa um item deve aceitar dois parâmetros ponteiros, o primeiro do tipo AGENT_REQUEST e o segundo do tipo AGENT_RESULT:

```
int zbx_module_dummy_random(AGENT_REQUEST *request, AGENT_RESULT *result)
{
    ...

    SET_UI64_RESULT(result, from + rand() % (to - from + 1));
```

```
    return SYSINFO_RET_OK;
}
```

Essas funções devem retornar SYSINFO_RET_OK, se o valor do item foi obtido com sucesso. Caso contrário, devem retornar SYSINFO_RET_FAIL. Veja o exemplo do módulo "dummy" abaixo para detalhes sobre como obter informações de AGENT_REQUEST e como definir informações em AGENT_RESULT.

Fornecendo callbacks de exportação de histórico

Attention:
A exportação de histórico via módulo não é mais suportada pelo Zabbix proxy.

O módulo pode especificar funções para exportar dados de histórico por tipo: Numérico (float), Numérico (unsigned), Caractere, Texto e Log:

```
typedef struct
{
    void    (*history_float_cb)(const ZBX_HISTORY_FLOAT *history, int history_num);
    void    (*history_integer_cb)(const ZBX_HISTORY_INTEGER *history, int history_num);
    void    (*history_string_cb)(const ZBX_HISTORY_STRING *history, int history_num);
    void    (*history_text_cb)(const ZBX_HISTORY_TEXT *history, int history_num);
    void    (*history_log_cb)(const ZBX_HISTORY_LOG *history, int history_num);
}
ZBX_HISTORY_WRITE_CBS;
```

Cada uma delas deve receber o array "history" de "history_num" elementos como argumentos. Dependendo do tipo de dado de histórico a ser exportado, "history" é um array das seguintes estruturas, respectivamente:

```
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;
```

Os callbacks serão usados pelos processos history syncer do Zabbix server no final do procedimento de sincronização de histórico, após os dados serem gravados no banco de dados do Zabbix e salvos no value cache.

> **Attention:**
> Em caso de erro interno no módulo de exportação de histórico, recomenda-se que o módulo seja escrito de forma que não bloqueie todo o monitoramento até se recuperar, mas descarte os dados e permita que o Zabbix server continue em execução.

Compilando módulos

Atualmente, os módulos devem ser compilados dentro da árvore de código-fonte do Zabbix, porque a API do módulo depende de algumas estruturas de dados que são definidas nos headers do Zabbix.

O header mais importante para módulos carregáveis é **include/module.h**, que define essas estruturas de dados. Outros headers de sistema necessários que ajudam **include/module.h** a funcionar corretamente são **stdlib.h** e **stdint.h**.

Com essas informações em mente, tudo está pronto para o módulo ser compilado. O módulo deve incluir **stdlib.h**, **stdint.h** e **module.h**, e o script de compilação deve garantir que esses arquivos estejam no caminho de inclusão. Veja o exemplo do módulo ”dummy” abaixo para mais detalhes.

Outro header útil é **include/zbxcommon.h**, que define a função **zabbix_log()**, que pode ser usada para fins de log e depuração.

Parâmetros de configuração

O agent, server e proxy do Zabbix suportam dois parâmetros para lidar com módulos:

- LoadModulePath – caminho completo para o local dos módulos carregáveis
- LoadModule – módulo(s) a serem carregados na inicialização. Os módulos devem estar localizados em um diretório especificado por LoadModulePath ou o caminho deve preceder o nome do módulo. Se o caminho anterior for absoluto (começar com ’/’) então LoadModulePath será ignorado. É permitido incluir vários parâmetros LoadModule.

Por exemplo, para estender o agent do Zabbix, poderíamos adicionar os seguintes parâmetros:

```
LoadModulePath=/usr/local/lib/zabbix/agent/
LoadModule=mariadb.so
LoadModule=apache.so
LoadModule=kernel.so
LoadModule=/usr/local/lib/zabbix/dummy.so
```

Ao iniciar o agent, ele carregará os módulos mariadb.so, apache.so e kernel.so do diretório /usr/local/lib/zabbix/agent, enquanto dummy.so será carregado de /usr/local/lib/zabbix. O agent não conseguirá iniciar se um módulo estiver ausente, em caso de permissões incorretas ou se uma biblioteca compartilhada não for um módulo do Zabbix.

Configuração do frontend

Módulos carregáveis são suportados pelo agent, server e proxy do Zabbix. Portanto, o tipo de item no frontend do Zabbix depende de onde o módulo é carregado. Se o módulo for carregado no agent, então o tipo de item deve ser ”Zabbix agent” ou ”Zabbix agent (active)”. Se o módulo for carregado no server ou proxy, então o tipo de item deve ser ”Simple check”.

A exportação de histórico através de módulos do Zabbix não precisa de nenhuma configuração no frontend. Se o módulo for carregado com sucesso pelo server e fornecer a função **zbx_module_history_write_cbs()** que retorna pelo menos uma função de callback diferente de NULL, então a exportação de histórico será habilitada automaticamente.

Módulo de exemplo

O Zabbix inclui um módulo de exemplo escrito em linguagem C. O módulo está localizado em 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
```

O módulo está bem documentado e pode ser usado como um template para seus próprios módulos.

Após executar ./configure na raiz do código-fonte do Zabbix, conforme descrito acima, basta executar **make** para compilar o **dummy.so**.

```
/*
** Zabbix
** Copyright (C) 2001-2020 Zabbix SIA
**
** This program is free software; you can redistribute it and/or modify
** it under the terms of the GNU General Public License as published by
** the Free Software Foundation; either version 2 of the License, or
** (at your option) any later version.
**
** This program is distributed in the hope that it will be useful,
** but WITHOUT ANY WARRANTY; without even the implied warranty of
** MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
** GNU General Public License for more details.
**
** You should have received a copy of the GNU General Public License
** along with this program; if not, write to the Free Software
** Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
**/

###include <stdlib.h>
###include <string.h>
###include <time.h>
###include <stdint.h>

###include "module.h"

/* the variable keeps timeout setting for item processing */
static int  item_timeout = 0;

/* module SHOULD define internal functions as static and use a naming pattern different from Zabbix intern
/* symbols (zbx_*) and loadable module API functions (zbx_module_*) to avoid conflicts
static int  dummy_ping(AGENT_REQUEST *request, AGENT_RESULT *result);
static int  dummy_echo(AGENT_REQUEST *request, AGENT_RESULT *result);
static int  dummy_random(AGENT_REQUEST *request, AGENT_RESULT *result);

static ZBX_METRIC keys[] =
/*  KEY         FLAG        FUNCTION    TEST PARAMETERS */
{
    {"dummy.ping",      0,      dummy_ping, NULL},
    {"dummy.echo",      CF_HAVEPARAMS,  dummy_echo, "a message"},
    {"dummy.random",    CF_HAVEPARAMS,  dummy_random,   "1,1000"},
    {NULL}
};

/******************************************************************************
 *                                                                            *
 * Function: zbx_module_api_version                                           *
 *                                                                            *
 * Purpose: returns version number of the module interface                    *
 *                                                                            *
 * Return value: ZBX_MODULE_API_VERSION - version of module.h module is       *
 *               compiled with, in order to load module successfully Zabbix   *
 *               MUST be compiled with the same version of this header file   *
 *                                                                            *
```

```
 ******************************************************************************/
int zbx_module_api_version(void)
{
    return ZBX_MODULE_API_VERSION;
}

/******************************************************************************
 *                                                                            *
 * Function: zbx_module_item_timeout                                          *
 *                                                                            *
 * Purpose: set timeout value for processing of items                         *
 *                                                                            *
 * Parameters: timeout - timeout in seconds, 0 - no timeout set               *
 *                                                                            *
 ******************************************************************************/
void    zbx_module_item_timeout(int timeout)
{
    item_timeout = timeout;
}

/******************************************************************************
 *                                                                            *
 * Function: zbx_module_item_list                                             *
 *                                                                            *
 * Purpose: returns list of item keys supported by the module                 *
 *                                                                            *
 * Return value: list of item keys                                            *
 *                                                                            *
 ******************************************************************************/
ZBX_METRIC  *zbx_module_item_list(void)
{
    return keys;
}

static int  dummy_ping(AGENT_REQUEST *request, AGENT_RESULT *result)
{
    SET_UI64_RESULT(result, 1);

    return SYSINFO_RET_OK;
}

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;
}
```

O módulo exporta três novos items:

- `dummy.ping` - sempre retorna '1'
- `dummy.echo[param1]` - retorna o primeiro parâmetro como está, por exemplo, `dummy.echo[ABC]` retornará ABC
- `dummy.random[param1, param2]` - retorna um número aleatório dentro do intervalo de param1-param2, por exemplo, `dummy.random[1,1000000]`

Limitações

O suporte a módulos carregáveis está implementado apenas para a plataforma Unix. Isso significa que não funciona para agents Windows.

Em alguns casos, um módulo pode precisar ler parâmetros de configuração relacionados ao módulo a partir do *zabbix_agentd.conf*. Isso não é suportado atualmente. Se você precisar que seu módulo use alguns parâmetros de configuração, provavelmente deverá implementar a análise de um arquivo de configuração específico do módulo.

2 Plugins

Visão geral

Plugins estendem as capacidades de monitoramento do Zabbix agent 2. Escritos na linguagem de programação Go, eles fornecem uma alternativa aos módulos carregáveis (escritos em C) e outros métodos para estender a funcionalidade do Zabbix.

Esses recursos são específicos do agent 2 e seus plugins:

- Intervalos agendados e flexíveis para verificações passivas e ativas
- Gerenciamento de fila de tarefas em relação ao agendamento e concorrência de tarefas
- Timeouts no nível do plugin
- Verificação de compatibilidade do Zabbix agent 2 e seus plugins na inicialização

Todas as métricas suportadas pelo Zabbix agent 2 são coletadas por plugins. Esta página descreve os princípios de configuração de plugins do ponto de vista do usuário e lista tanto os plugins internos quanto os carregáveis fornecidos pelo Zabbix.

Note:
Desde o Zabbix 6.0, os plugins não precisam mais ser incorporados ao agent 2. Você pode adicioná-los como plugins carregáveis, simplificando assim a criação de plugins para novas métricas. Para instruções sobre como criar seus próprios plugins, consulte o Centro de desenvolvedores e o Plugin de exemplo para o Zabbix agent 2.

Configurando plugins

Esta seção fornece princípios comuns de configuração de plugins e melhores práticas.

Todos os plugins são configurados usando o parâmetro *Plugins.**, que pode fazer parte do arquivo de configuração do Zabbix agent 2 ou do arquivo de configuração próprio do plugin. Se um plugin usar um arquivo de configuração separado, o caminho para esse arquivo deve ser especificado no parâmetro Include do arquivo de configuração do Zabbix agent 2.

Um parâmetro típico de plugin tem a seguinte estrutura:

Plugins.<PluginName>.<Parameter>=<Value>

Além disso, existem dois grupos específicos de parâmetros:

- *Plugins.<PluginName>.Default.<Parameter>=<Value>* usado para definir valores padrão de parâmetros.
- *Plugins.<PluginName>.<SessionName>.<Parameter>=<Value>* usado para definir conjuntos separados de parâmetros para diferentes alvos de monitoramento por meio de sessões nomeadas.

Todos os nomes de parâmetros devem obedecer aos seguintes requisitos:

- recomenda-se capitalizar os nomes dos seus plugins
- o parâmetro deve ser capitalizado
- caracteres especiais não são permitidos
- o aninhamento não é limitado por um nível máximo
- o número de parâmetros não é limitado

Por exemplo, para executar cheques ativos que não possuem *Scheduling* intervalo de atualização imediatamente após o reinício do agent apenas para o plugin Uptime, defina `Plugins.Uptime.System.ForceActiveChecksOnStart=1` no arquivo de configuração. Da mesma forma, para definir um limite personalizado para cheques concorrentes para o plugin CPU, defina o `Plugins.CPU.System.Capacity=N` no arquivo de configuração.

Valores padrão

Você pode definir valores padrão para os parâmetros relacionados à conexão (URI, nome de usuário, senha, etc.) no arquivo de configuração no formato:

```
Plugins.<PluginName>.Default.<Parameter>=<Value>
```

Por exemplo, `Plugins.Mysql.Default.Username=zabbix`, `Plugins.MongoDB.Default.Uri=tcp://127.0.0.1:27017`, etc.

Se um valor para esse parâmetro não for fornecido em uma chave de item ou nos parâmetros da sessão nomeada, o plugin usará o valor padrão. Se um parâmetro padrão também não estiver definido, os valores padrão codificados serão usados.

> **Note:**
> Se uma chave de item não tiver nenhum parâmetro, o agent Zabbix 2 tentará coletar a métrica usando os valores definidos na seção de parâmetros padrão.

Sessões nomeadas

Sessões nomeadas representam um nível adicional de parâmetros de plugin e podem ser usadas para especificar conjuntos separados de parâmetros de autenticação para cada uma das instâncias monitoradas. Cada parâmetro de sessão nomeada deve ter a seguinte estrutura:

```
Plugins.<PluginName>.Sessions.<SessionName>.<Parameter>=<Value>
```

Um nome de sessão pode ser usado como um parâmetro de chave de item connString em vez de especificar um URI, nome de usuário e/ou senha separadamente.

Nas chaves de item, o primeiro parâmetro pode ser um connString ou um URI. Se o primeiro parâmetro da chave não corresponder a nenhum nome de sessão, ele será tratado como um URI. Observe que passar credenciais de URI incorporadas na chave do item não é suportado, use parâmetros de sessão nomeada em vez disso.

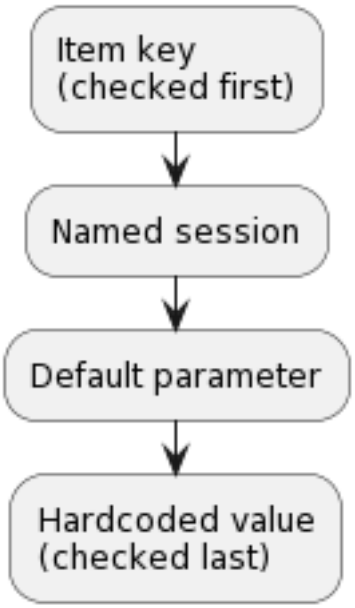
A lista de parâmetros de sessão nomeada disponíveis depende do plugin.

É possível substituir os parâmetros da sessão especificando novos valores nos parâmetros da chave do item (veja o exemplo).

Se um parâmetro não for definido para a sessão nomeada, o Zabbix agent 2 usará o valor definido no parâmetro padrão do plugin.

Prioridade dos parâmetros

Os plugins do Zabbix agent 2 procuram pelos valores dos parâmetros relacionados à conexão na seguinte ordem:



1. O primeiro parâmetro da chave do item é comparado aos nomes das sessões. Se nenhuma correspondência for encontrada, ele será tratado como um valor real; neste caso, a etapa 3 será ignorada. Se uma correspondência for encontrada, o valor do parâmetro (geralmente, um URI) deve ser definido na sessão nomeada.
2. Outros parâmetros serão obtidos da chave do item, se definidos.
3. Se um parâmetro da chave do item (por exemplo, senha) estiver vazio, o plugin procurará o parâmetro correspondente na sessão nomeada.
4. Se o parâmetro da sessão também não for especificado, será usado o valor definido no parâmetro padrão correspondente.
5. Se tudo mais falhar, o plugin usará o valor padrão codificado.

Exemplo 1

Monitoramento de duas instâncias “MySQL1” e “MySQL2”.

Parâmetros de configuração:

```
Plugins.Mysql.Sessions.MySQL1.Uri=tcp://127.0.0.1:3306
Plugins.Mysql.Sessions.MySQL1.User=mysql1_user
Plugins.Mysql.Sessions.MySQL1.Password=unique_password
Plugins.Mysql.Sessions.MySQL2.Uri=tcp://192.0.2.0:3306
Plugins.Mysql.Sessions.MySQL2.User=mysql2_user
Plugins.Mysql.Sessions.MySQL2.Password=different_password
```

Como resultado desta configuração, cada nome de sessão pode ser usado como um connString em uma chave de item, por exemplo, `mysql.ping[MySQL1]` ou `mysql.ping[MySQL2]`.

Exemplo 2

Fornecendo alguns dos parâmetros na chave do item.

Parâmetros de configuração:

```
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
```

Chave do item: `pgsql.ping[session1,new_username,,postgres]`

Como resultado desta configuração, o agent irá se conectar ao PostgreSQL usando os seguintes parâmetros:

- URI do parâmetro de sessão: `192.0.2.234:5432`
- Nome de usuário da chave do item: `new_username`
- Senha do parâmetro de sessão (já que foi omitida na chave do item): `session_password`
- Nome do banco de dados da chave do item: `postgres`

Exemplo 3

Coletando uma métrica usando parâmetros de configuração padrão.

Parâmetros de configuração:

```
Plugins.PostgreSQL.Default.Uri=tcp://192.0.2.234:5432
Plugins.PostgreSQL.Default.User=zabbix
Plugins.PostgreSQL.Default.Password=password
```

Chave do item: `pgsql.ping[,,,postgres]`

Como resultado desta configuração, o agent irá se conectar ao PostgreSQL usando os parâmetros:

- URI padrão: `192.0.2.234:5432`
- Nome de usuário padrão: `zabbix`
- Senha padrão: `password`
- Nome do banco de dados da chave do item: `postgres`

Conexões

Alguns plugins suportam a coleta de métricas de várias instâncias simultaneamente. Tanto instâncias locais quanto remotas podem ser monitoradas. Conexões TCP e Unix-socket são suportadas.

É recomendável configurar os plugins para manter as conexões com as instâncias em estado aberto. Os benefícios são a redução da congestão de rede, latência e uso de CPU e memória devido ao menor número de conexões. A biblioteca cliente cuida disso.

> **Note:**
> O período de tempo pelo qual as conexões não utilizadas devem permanecer abertas pode ser determinado pelo parâmetro *Plugins.<PluginName>.KeepAlive*. Exemplo: *Plugins.Memcached.KeepAlive*

Plugins internos

Plugins internos do Zabbix agent 2 estão disponíveis por padrão. Clique no nome do plugin para acessar o repositório do plugin com informações adicionais.

Nome do plugin	Descrição	Chaves de item suportadas	Comentários
Agent	Métricas do Zabbix agent em uso.	agent.hostname, agent.ping, agent.version	As chaves suportadas possuem os mesmos parâmetros das chaves do Zabbix agent.
Ceph	Monitoramento do Ceph.	ceph.df.details, ceph.osd.stats, ceph.osd.discovery, ceph.osd.dump, ceph.ping, ceph.pool.discovery, ceph.status	
CPU	Monitoramento da CPU do sistema (número de CPUs/núcleos, CPUs descobertas, percentual de utilização).	system.cpu.discovery, system.cpu.num, system.cpu.util	As chaves suportadas possuem os mesmos parâmetros das chaves do Zabbix agent.
Docker	Monitoramento de containers Docker.	docker.container_info, docker.container_stats, docker.containers, docker.containers.discovery, docker.data_usage, docker.images, docker.images.discovery, docker.info, docker.ping	Veja também: Parâmetros de configuração
File	Coleta de métricas de arquivos.	vfs.file.cksum, vfs.file.contents, vfs.file.exists, vfs.file.md5sum, vfs.file.regexp, vfs.file.regmatch, vfs.file.size, vfs.file.time	As chaves suportadas possuem os mesmos parâmetros das chaves do Zabbix agent.
Kernel	Monitoramento do kernel.	kernel.maxfiles, kernel.maxproc	As chaves suportadas possuem os mesmos parâmetros das chaves do Zabbix agent.
Log	Monitoramento de arquivos de log.	log, log.count, logrt, logrt.count	As chaves suportadas possuem os mesmos parâmetros das chaves do Zabbix agent. Veja também: Parâmetros de configuração do plugin (Unix/Windows)
Memcached	Monitoramento do servidor Memcached.	memcached.ping, memcached.stats	
Modbus	Leitura de dados Modbus.	modbus.get	As chaves suportadas possuem os mesmos parâmetros das chaves do Zabbix agent.
MQTT	Recebe valores publicados de tópicos MQTT.	mqtt.get	Para configurar uma conexão criptografada com o broker MQTT, especifique os parâmetros TLS no arquivo de configuração do agent como uma sessão nomeada ou parâmetros padrão. Atualmente, os parâmetros TLS não podem ser passados como parâmetros de chave de item.
MySQL	Monitoramento do MySQL e seus forks.	mysql.custom.query, mysql.db.discovery, mysql.db.size, mysql.get_status_variables, mysql.ping, mysql.replication.discovery, mysql.replication.get_slave_status, mysql.version	Para configurar uma conexão criptografada com o banco de dados, especifique os parâmetros TLS no arquivo de configuração do agent como uma sessão nomeada ou parâmetros padrão. Atualmente, os parâmetros TLS não podem ser passados como parâmetros de chave de item.

Nome do plugin	Descrição	Chaves de item suportadas	Comentários
NetIf	Monitoramento de interfaces de rede.	net.if.collisions, net.if.discovery, net.if.in, net.if.out, net.if.total	As chaves suportadas possuem os mesmos parâmetros das chaves do Zabbix agent.
Oracle	Monitoramento do Oracle Database.	oracle.diskgroups.stats, oracle.diskgroups.discovery, oracle.archive.info, oracle.archive.discovery, oracle.cdb.info, oracle.custom.query, oracle.datafiles.stats, oracle.db.discovery, oracle.fra.stats, oracle.instance.info, oracle.pdb.info, oracle.pdb.discovery, oracle.pga.stats, oracle.ping, oracle.proc.stats, oracle.redolog.info, oracle.sga.stats, oracle.sessions.stats, oracle.sys.metrics, oracle.sys.params, oracle.ts.stats, oracle.ts.discovery, oracle.user.info, oracle.version	Instale o Oracle Instant Client antes de usar o plugin.
Proc	Percentual de utilização de CPU por processo.	proc.cpu.util	A chave suportada possui os mesmos parâmetros da chave do Zabbix agent.
Redis	Monitoramento do servidor Redis.	redis.config, redis.info, redis.ping, redis.slowlog.count	Para configurar uma conexão criptografada com o Redis, especifique os parâmetros TLS no arquivo de configuração do agent como uma sessão nomeada ou parâmetros padrão. Os parâmetros TLS não podem ser passados como parâmetros de chave de item. Observe que uma configuração TLS incorreta ou inválida pode impedir o Zabbix agent 2 de iniciar, portanto, verifique os arquivos de certificado, permissões e caminhos antes de habilitar o TLS.

Smart	Monitoramento S.M.A.R.T.	smart.attribute.discovery, smart.disk.discovery, smart.disk.get	A versão mínima exigida do smartctl é 7.1. Direitos de acesso sudo/root ao smartctl são necessários para o usuário que executa o Zabbix agent 2. O plugin utiliza apenas os seguintes comandos:
			`/usr/sbin/smartctl -a *`
			`/usr/sbin/smartctl --scan *`
			`/usr/sbin/smartctl -j -V`
			
			As chaves suportadas podem ser usadas com o Zabbix agent 2 apenas em Linux/Windows, tanto como checagem passiva quanto ativa.
			Veja também:
			Parâmetros de configuração

SW	Listagem de pacotes instalados.	system.sw.packages, system.sw.packages.get	As chaves suportadas possuem os mesmos parâmetros da chave do Zabbix agent.

Nome do plugin	Descrição	Chaves de item suportadas	Comentários
Swap	Tamanho do espaço swap em bytes/porcentagem.	system.swap.size	A chave suportada possui os mesmos parâmetros da chave do Zabbix agent.
SystemRun	Executa o comando especificado.	system.run	A chave suportada possui os mesmos parâmetros da chave do Zabbix agent. Veja também: Parâmetros de configuração do plugin (Unix/Windows)
Systemd	Monitoramento de serviços systemd.	systemd.unit.discovery, systemd.unit.get, systemd.unit.info	
TCP	Verificação de disponibilidade de conexão TCP.	net.tcp.port	A chave suportada possui os mesmos parâmetros da chave do Zabbix agent.
UDP	Monitoramento da disponibilidade e desempenho de serviços UDP.	net.udp.service, net.udp.service.perf	As chaves suportadas possuem os mesmos parâmetros das chaves do Zabbix agent.
Uname	Recuperação de informações sobre o sistema.	system.hostname, system.sw.arch, system.uname	As chaves suportadas possuem os mesmos parâmetros das chaves do Zabbix agent.
Uptime	Coleta de métricas de tempo de atividade do sistema.	system.uptime	A chave suportada possui os mesmos parâmetros da chave do Zabbix agent.
VFSDev	Coleta de métricas VFS.	vfs.dev.discovery, vfs.dev.read, vfs.dev.write	As chaves suportadas possuem os mesmos parâmetros das chaves do Zabbix agent.
WebCertificate	Monitoramento de certificados TLS/SSL de sites.	web.certificate.get	
WebPage	Monitoramento de páginas web.	web.page.get, web.page.perf, web.page.regexp	As chaves suportadas possuem os mesmos parâmetros das chaves do Zabbix agent.
ZabbixAsync	Coleta assíncrona de métricas.	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	As chaves suportadas possuem os mesmos parâmetros das chaves do Zabbix agent.
ZabbixStats	Métricas internas do Zabbix server/proxy ou número de itens atrasados na fila.	zabbix.stats	As chaves suportadas possuem os mesmos parâmetros das chaves do Zabbix agent.

Nome do plugin	Descrição	Chaves de item suportadas	Comentários
ZabbixSync	Coleta síncrona de métricas.	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.	As chaves suportadas possuem os mesmos parâmetros das chaves do Zabbix agent.

Plugins carregáveis

Os plugins carregáveis do agent 2 do Zabbix não estão disponíveis por padrão e precisam ser instalados separadamente:

- No Linux, você pode usar pacotes (por exemplo, instale o Ember+ no Ubuntu com `apt install zabbix-agent2-plugin-ember-pl` ou compilar plugins
- No Windows, você pode instalar a partir do MSI ou compilar plugins

Attention:
Antes de instalar um plugin, verifique o arquivo README do plugin. Ele pode conter requisitos específicos e instruções de instalação.

Clique no nome do plugin para acessar o repositório do plugin, que contém o arquivo README do plugin com informações adicionais.

Nome do plugin	Descrição	Chaves de item suportadas	Comentários
Ember+	Monitoramento do Ember+.	ember.get	Veja também Parâmetros de configuração do plugin Ember+.
MongoDB	Monitoramento de servidores e clusters MongoDB (banco de dados distribuído baseado em documentos).	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	Para configurar conexões criptografadas com o banco de dados, especifique os parâmetros TLS no arquivo de configuração do agent como parâmetros de sessão nomeada. Atualmente, os parâmetros TLS não podem ser passados como parâmetros de chave de item. Veja também Parâmetros de configuração do plugin MongoDB.

Nome do plugin	Descrição	Chaves de item suportadas	Comentários
MSSQL	Monitoramento do banco de dados MSSQL.	mssql.availability.group.get, mssql.custom.query, mssql.db.get, mssql.job.status.get, mssql.last.backup.get, mssql.local.db.get, mssql.mirroring.get, mssql.nonlocal.db.get, mssql.perfcounter.get, mssql.ping, mssql.quorum.get, mssql.quorum.member.get, mssql.replica.get, mssql.version	Para configurar a conexão criptografada com o banco de dados, especifique os parâmetros TLS no arquivo de configuração do agent como parâmetros de sessão nomeada ou padrão. Atualmente, os parâmetros TLS não podem ser passados como parâmetros de chave de item. Veja também Parâmetros de configuração do plugin MSSQL.
PostgreSQL	Monitoramento do PostgreSQL e seus forks.	pgsql.autovacuum.count, pgsql.archive, pgsql.bgwriter, pgsql.cache.hit, pgsql.connections, pgsql.custom.query, pgsql.dbstat, pgsql.dbstat.sum, pgsql.db.age, pgsql.db.bloating_tables, pgsql.db.discovery, pgsql.db.size, pgsql.locks, pgsql.oldest.xid, pgsql.ping, pgsql.queries, pgsql.replication.count, pgsql.replication.process, pgsql.replication.process.discovery, pgsql.replication.recovery_role, pgsql.replication.status, pgsql.replication_lag.b, pgsql.replication_lag.sec, pgsql.uptime, pgsql.version, pgsql.wal.stat	Para configurar conexões criptografadas com o banco de dados, especifique os parâmetros TLS no arquivo de configuração do agent como parâmetros de sessão nomeada ou padrão. Atualmente, os parâmetros TLS não podem ser passados como parâmetros de chave de item. Veja também Parâmetros de configuração do plugin PostgreSQL.

Note:
Plugins carregáveis, quando lançados com:
 - `-V --version` - imprime a versão do plugin e informações de licença;
 - `-h --help` - imprime informações de ajuda.

3 Módulos do frontend

Visão geral

É possível aprimorar a funcionalidade do frontend do Zabbix adicionando módulos de terceiros ou desenvolvendo seus próprios módulos sem a necessidade de alterar o código-fonte do Zabbix.

Observe que o código do módulo será executado com os mesmos privilégios do código-fonte do Zabbix. Isso significa:

- módulos de terceiros podem ser prejudiciais. Você deve confiar nos módulos que está instalando;
- Erros no código de um módulo de terceiros podem travar o frontend. Se isso acontecer, basta remover o código do módulo do frontend. Assim que você recarregar o frontend do Zabbix, verá uma nota dizendo que alguns módulos estão ausentes.

Vá para Administração de módulos (em *Administração* → *Geral* → *Módulos*) e clique em *Verificar diretório* novamente para remover módulos inexistentes do banco de dados.

Instalação

Por favor, leia sempre o manual de instalação de um determinado módulo. Recomenda-se instalar novos módulos um por um para identificar falhas facilmente.

Antes de instalar um módulo:

- Certifique-se de que você fez o download do módulo de uma fonte confiável. A instalação de código malicioso pode levar a consequências como perda de dados
- Diferentes versões do mesmo módulo (mesmo ID) podem ser instaladas em paralelo, mas apenas uma única versão pode ser habilitada por vez

Etapas para instalar um módulo:

- Descompacte seu módulo em sua própria pasta dentro da pasta `modules` do frontend do Zabbix
- Certifique-se de que a pasta do seu módulo contenha pelo menos o arquivo manifest.json
- Navegue até a Administração de módulos e clique no botão *Procurar diretório*
- O novo módulo aparecerá na lista junto com sua versão, autor, descrição e status
- Habilite o módulo clicando em seu status

Solução de problemas:

Problema	Solução
O módulo não apareceu na lista	Se o seu módulo não apareceu na lista, certifique-se de que o manifest.json existe em modules/seu-modulo/. Se não existir, você pode ter descompactado o módulo no diretório errado. Se existir, o módulo pode não ser compatível com a sua versão do Zabbix. Além disso, verifique se o usuário do servidor web tem pelo menos acesso de leitura e pesquisa (`r-x`) à pasta do módulo e a todos os subdiretórios, e acesso de leitura (`r--`) a todos os arquivos dentro dela.
O frontend travou	O código do módulo não é compatível com a versão atual do Zabbix ou com a configuração do servidor. Por favor, exclua os arquivos do módulo e recarregue o frontend. Você verá um aviso de que alguns módulos estão ausentes. Vá para Administração de módulos e clique novamente em *Procurar diretório* para remover módulos inexistentes do banco de dados.
Mensagem de erro sobre namespace, ID ou ações idênticas aparece	O novo módulo tentou registrar um namespace, ID ou ações que já estão registrados por outros módulos habilitados. Desabilite o módulo conflitante (mencionado na mensagem de erro) antes de habilitar o novo.
Mensagens de erro técnico aparecem	Relate os erros ao desenvolvedor do módulo.

Desenvolvendo módulos

Para obter informações sobre o desenvolvimento de módulos personalizados, consulte o Centro de desenvolvedores.

22 Apêndices

Use a barra lateral para acessar o conteúdo da seção Apêndices.

1 Instalação e configuração

Por favor, use a barra lateral para acessar o conteúdo desta seção.

1 Criação do banco de dados

Visão geral

Um banco de dados Zabbix deve ser criado durante a instalação do server ou proxy Zabbix.

Esta seção fornece instruções para criar um banco de dados Zabbix. Um conjunto separado de instruções está disponível para cada banco de dados suportado.

> **Note:**
> Para melhorar a segurança do banco de dados criando funções/usuários de banco de dados com privilégios mínimos, consulte as melhores práticas de criação de banco de dados para cada banco de dados suportado:

> - MySQL/MariaDB
> - PostgreSQL/TimescaleDB
> - Oracle
>
> Para configurar conexões TLS seguras, consulte Conexão segura com o banco de dados.

UTF-8 é a única codificação suportada pelo Zabbix. Sabe-se que funciona sem falhas de segurança. Os usuários devem estar cientes de que existem problemas de segurança conhecidos ao usar algumas das outras codificações. Para alternar para UTF-8, consulte Reparando o conjunto de caracteres e a ordenação do banco de dados Zabbix. Veja também Limites de filtragem com ordenações utf8mb4.

> **Note:**
> Se estiver instalando a partir do repositório Git do Zabbix, você precisa executar o seguinte comando antes de prosseguir para as próximas etapas:

 `make dbschema`

MySQL/MariaDB

Os conjuntos de caracteres utf8 (também conhecido como utf8mb3) e utf8mb4 são suportados (com as colações utf8_bin e utf8mb4_bin, respectivamente) para que o Zabbix server/proxy funcione corretamente com o banco de dados MySQL. Recomenda-se usar utf8mb4 para novas instalações.

Triggers determinísticos precisam ser criados durante a importação do schema. No MySQL e MariaDB, isso requer que `GLOBAL log_bin_trust_function_creators = 1` seja definido se o log binário estiver habilitado e não houver privilégios de superusuário e `log_bin_trust_function_creators = 1` não estiver definido no arquivo de configuração do MySQL.

Se você estiver instalando a partir dos **pacotes** do Zabbix, siga as instruções para sua plataforma.

Se você estiver instalando o Zabbix a partir do código-fonte:

- Crie e configure um banco de dados e um usuário.

```
mysql -uroot -p<senha>

mysql> create database zabbix character set utf8mb4 collate utf8mb4_bin;
mysql> create user 'zabbix'@'localhost' identified by '<senha>';
mysql> grant all privileges on zabbix.* to 'zabbix'@'localhost';
mysql> SET GLOBAL log_bin_trust_function_creators = 1;
mysql> quit;
```

- Importe os dados para o banco de dados e defina o conjunto de caracteres utf8mb4 como padrão (assumindo que você está no diretório raiz do código-fonte do Zabbix). Para um banco de dados de Zabbix proxy, apenas o `schema.sql` deve ser importado (não importando images.sql nem data.sql).

```
cd database/mysql
mysql -uzabbix -p<senha> zabbix < schema.sql
#### pare aqui se estiver criando o banco de dados para o Zabbix proxy
mysql -uzabbix -p<senha> zabbix < images.sql
mysql -uzabbix -p<senha> --default-character-set=utf8mb4 zabbix < data.sql
```

`log_bin_trust_function_creators` pode ser desabilitado após o schema ter sido importado com sucesso:

```
mysql -uroot -p<senha>

mysql> SET GLOBAL log_bin_trust_function_creators = 0;
mysql> quit;
```

PostgreSQL

Você precisa ter um usuário de banco de dados com permissões para criar objetos de banco de dados.

Se você estiver instalando a partir dos **pacotes** do Zabbix, siga as instruções para sua plataforma.

Se você estiver instalando o Zabbix a partir do código-fonte:

- Crie um usuário de banco de dados.

O seguinte comando shell criará o usuário `zabbix`. Especifique uma senha quando solicitado e repita a senha (observe que, primeiro, pode ser solicitada a senha do sudo):

```
sudo -u postgres createuser --pwprompt zabbix
```

- Crie um banco de dados.

O seguinte comando shell criará o banco de dados `zabbix` (último parâmetro) com o usuário criado anteriormente como proprietário (`-O zabbix`).

```
sudo -u postgres createdb -O zabbix -E Unicode -T template0 zabbix
```

- Importe o esquema inicial e os dados (assumindo que você está no diretório raiz do código-fonte do Zabbix). Para um banco de dados de proxy do Zabbix, apenas o `schema.sql` deve ser importado (não importando images.sql nem data.sql).

```
cd database/postgresql
cat schema.sql | sudo -u zabbix psql zabbix
#### pare aqui se estiver criando o banco de dados para o proxy do Zabbix
cat images.sql | sudo -u zabbix psql zabbix
cat data.sql | sudo -u zabbix psql zabbix
```

> **Attention:**
> Os comandos acima são fornecidos como exemplo e funcionarão na maioria das instalações GNU/Linux. Você pode usar comandos diferentes dependendo de como seu sistema/banco de dados está configurado, por exemplo:

 `psql -U <username>`

 Se você tiver algum problema ao configurar o banco de dados, consulte o administrador do seu banco de dados.

TimescaleDB

As instruções para criar e configurar o TimescaleDB são fornecidas em uma seção separada.

Oracle

As instruções para criar e configurar o banco de dados Oracle são fornecidas em uma seção separada.

SQLite

O uso do SQLite é suportado apenas para o **Zabbix proxy**!

O banco de dados será criado automaticamente se não existir.

Retorne à seção de instalação.

2 Reparando o conjunto de caracteres e agrupamento do banco de dados Zabbix

MySQL/MariaDB

Historicamente, o MySQL e seus derivados usavam 'utf8' como um alias para utf8mb3 - a implementação própria do MySQL de 3 bytes do padrão UTF8, que é de 4 bytes. A partir do MySQL 8.0.28 e MariaDB 10.6.1, o conjunto de caracteres 'utf8mb3' está obsoleto e, em algum momento, seu suporte será removido enquanto 'utf8' se tornará uma referência para 'utf8mb4'. Desde o Zabbix 6.0, 'utf8mb4' é suportado. Para evitar problemas futuros, é altamente recomendado usar 'utf8mb4'. Outra vantagem de mudar para 'utf8mb4' é o suporte a caracteres Unicode suplementares.

> **Warning:**
> Como versões anteriores ao Zabbix 6.0 não reconhecem utf8mb4, certifique-se de primeiro atualizar o Zabbix server e o esquema do banco de dados para 6.0.x ou posterior antes de executar a conversão para utf8mb4.

1. Verifique o conjunto de caracteres e a collation do banco de dados.

Por exemplo:

```
mysql> SELECT @@character_set_database, @@collation_database;
+--------------------------+----------------------+
```

```
| @@character_set_database | @@collation_database |
+--------------------------+----------------------+
| latin2                   | latin2 _general_ci   |
+--------------------------+----------------------+
```

Ou:

```
mysql> SELECT @@character_set_database, @@collation_database;
+--------------------------+----------------------+
| @@character_set_database | @@collation_database |
+--------------------------+----------------------+
| utf8                     | utf8_bin             |
+--------------------------+----------------------+
```

Como podemos ver, o conjunto de caracteres aqui não é 'utf8mb4' e a collation não é 'utf8mb4_bin', então precisamos corrigi-los.

2. Pare o Zabbix.

3. Crie uma cópia de backup do banco de dados!

4. Corrija o conjunto de caracteres e a collation no nível do banco de dados:

```
alter database <your DB name> character set utf8mb4 collate utf8mb4_bin;
```

Valores corrigidos:

```
mysql> SELECT @@character_set_database, @@collation_database;
+--------------------------+----------------------+
| @@character_set_database | @@collation_database |
+--------------------------+----------------------+
| utf8mb4                  | utf8mb4_bin          |
+--------------------------+----------------------+
```

5. Carregue o script para corrigir o conjunto de caracteres e a collation no nível de tabela e coluna:

```
mysql <your DB name> < utf8mb4_convert.sql
```

6. Execute o script:

```
               SET @ZABBIX_DATABASE = '<your DB name>';
Se MariaDB →  set innodb_strict_mode = OFF;
               CALL zbx_convert_utf8();
Se MariaDB →  set innodb_strict_mode = ON;
               drop procedure zbx_convert_utf8;
```

Observe que espera-se que 'utf8mb4' consuma um pouco mais de espaço em disco.

7. Se não houver erros, você pode querer criar uma cópia de backup do banco de dados com o banco de dados corrigido.

8. Inicie o Zabbix.

3 Atualização do banco de dados para chaves primárias

Visão geral

Esta seção fornece instruções para atualizar manualmente as tabelas em instalações existentes para chaves primárias.

A atualização para chaves primárias otimiza a forma como os dados são indexados e acessados, o que pode acelerar as consultas e economizar espaço. Também melhora o gerenciamento e a sincronização de dados em ambientes clusterizados, ajudando na escalabilidade e garantindo que o sistema permaneça confiável mesmo que alguns servidores falhem.

> **Attention:**
> As instruções fornecidas nesta página são destinadas a usuários avançados e podem precisar ser ajustadas para sua configuração específica. A atualização para chaves primárias pode ser demorada e consumir muitos recursos. Certifique-se de que haja espaço livre em disco suficiente; dependendo do tamanho do seu banco de dados e dos dados armazenados, o processo pode exigir até 2,5 vezes o espaço atualmente utilizado pelas tabelas de histórico.

Chaves primárias são usadas para todas as tabelas em novas instalações desde o Zabbix 6.0.

Não há atualização automática do banco de dados para chaves primárias; no entanto, instalações existentes podem ser atualizadas manualmente **após** a atualização do Zabbix server para a versão 6.0 ou superior.

Attention:
Desde o Zabbix 7.0, a atualização das tabelas para chaves primárias também atualiza as tabelas para usar tipos de dados de dupla precisão.

 Se você possui o Zabbix 7.0, as tabelas já usam dupla precisão. No entanto, as instruções desta página ainda podem ser seguidas para atualizar as tabelas para chaves primárias sem afetar as tabelas que já usam dupla precisão.

 Se você possui o Zabbix 6.4 (ou anterior), considere atualizar as tabelas para dupla precisão primeiro. Para mais informações, consulte Atualizando para valores numéricos de faixa estendida.

As instruções estão disponíveis para:

- MySQL
- PostgreSQL
- TimescaleDB
- Oracle

Notas importantes

Para realizar a atualização do banco de dados:

1. Pare o Zabbix server.

Parar o Zabbix server durante a atualização é fortemente recomendado. No entanto, se for absolutamente necessário, você pode realizar a atualização enquanto o server estiver em execução (apenas para MySQL, MariaDB e PostgreSQL sem TimescaleDB).

2. Faça backup do seu banco de dados.

Se estiver usando PostgreSQL ou TimescaleDB, consulte problemas conhecidos para restaurar um backup do PostgreSQL ou TimescaleDB criado no Zabbix 7.0.0-7.0.4.

3. Instale o pacote zabbix-sql-scripts mais recente compatível com sua versão do Zabbix (por exemplo, para RHEL: `dnf install zabbix-sql-scripts`).
4. Execute os scripts para o seu banco de dados.
5. Inicie o Zabbix server.

Warning:
Execute os scripts apenas para o banco de dados do server. O proxy não se beneficiará desta atualização.

Se o banco de dados usar partições, entre em contato com o administrador do banco de dados ou com o Suporte Zabbix para obter ajuda.

Os arquivos CSV podem ser removidos após uma atualização bem-sucedida para chaves primárias.

Opcionalmente, o frontend do Zabbix pode ser colocado em modo de manutenção.

MySQL

A exportação e importação devem ser realizadas em tmux/screen para garantir que a sessão não seja encerrada.

Veja também: Notas importantes

MySQL 8.0+ com mysqlsh

Este método pode ser usado com um Zabbix server em execução, mas é recomendável parar o server durante a atualização. O MySQL Shell (*mysqlsh*) deve estar instalado e ser capaz de se conectar ao banco de dados.

- Faça login no console do MySQL como root (recomendado) ou como qualquer usuário com privilégios de FILE.
- Inicie o MySQL com a variável local_infile habilitada.
- Renomeie as tabelas antigas e crie novas tabelas executando `history_upgrade_prepare.sql`.

```
mysql -uzabbix -p<senha> zabbix < /usr/share/zabbix-sql-scripts/mysql/option-patches/history_upgrade_prepa
```

- Exporte e importe os dados.

Conecte-se via mysqlsh. Se estiver usando uma conexão via socket, pode ser necessário especificar o caminho.

```
sudo mysqlsh -uroot -S /run/mysqld/mysqld.sock --no-password -Dzabbix
```

Altere para o modo JavaScript usando:

```
\js
```

Em seguida, execute o código abaixo (CSVPATH pode ser alterado conforme necessário):

```
CSVPATH="/var/lib/mysql-files";

util.exportTable("history_old", CSVPATH + "/history.csv", { dialect: "csv" });
util.importTable(CSVPATH + "/history.csv", {"dialect": "csv", "table": "history" });

util.exportTable("history_uint_old", CSVPATH + "/history_uint.csv", { dialect: "csv" });
util.importTable(CSVPATH + "/history_uint.csv", {"dialect": "csv", "table": "history_uint" });

util.exportTable("history_str_old", CSVPATH + "/history_str.csv", { dialect: "csv" });
util.importTable(CSVPATH + "/history_str.csv", {"dialect": "csv", "table": "history_str" });

util.exportTable("history_log_old", CSVPATH + "/history_log.csv", { dialect: "csv" });
util.importTable(CSVPATH + "/history_log.csv", {"dialect": "csv", "table": "history_log" });

util.exportTable("history_text_old", CSVPATH + "/history_text.csv", { dialect: "csv" });
util.importTable(CSVPATH + "/history_text.csv", {"dialect": "csv", "table": "history_text" });
```

Se você receber a mensagem "JavaScript is not supported", sua instalação do MySQL Shell não possui suporte a JS. Nesse caso, instale o pacote oficial do MySQL Shell da Oracle (ou compile a partir do código-fonte) para que o modo JavaScript seja habilitado.

- Siga as instruções pós-migração para remover as tabelas antigas.

MariaDB/MySQL 8.0+ sem mysqlsh

Este método de atualização leva mais tempo e deve ser usado apenas se uma atualização com *mysqlsh* não for possível.

Atualização de tabelas

- Faça login no console do MySQL como root (recomendado) ou qualquer usuário com privilégios de FILE.
- Se estiver realizando a migração com um servidor Zabbix em execução, inicie o MySQL com a variável local_infile habilitada.
- Renomeie as tabelas antigas e crie novas tabelas executando `history_upgrade_prepare.sql`:

```
mysql -uzabbix -p<senha> zabbix < /usr/share/zabbix-sql-scripts/mysql/option-patches/history_upgrade_prepa
```

Migração com o servidor parado

`max_execution_time` (no MySQL) ou `max_statement_time` (no MariaDB) deve ser desabilitado antes de migrar os dados para evitar timeout durante a migração.

Para MySQL:

```
SET @@max_execution_time=0;
```

Para 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;
```

Siga as instruções pós-migração para remover as tabelas antigas.

Migração com o servidor em execução

Verifique para quais caminhos a importação/exportação está habilitada:

```
mysql> SELECT @@secure_file_priv;
+-----------------------+
| @@secure_file_priv    |
+-----------------------+
| /var/lib/mysql-files/ |
+-----------------------+
```

Se o valor de *secure_file_priv* for um caminho para um diretório, a exportação/importação será realizada para arquivos nesse diretório. Nesse caso, edite os caminhos dos arquivos nas consultas conforme necessário ou defina o valor de *secure_file_priv* como uma string vazia durante o tempo de atualização.

Se o valor de *secure_file_priv* estiver vazio, a exportação/importação pode ser realizada de qualquer local.

Se o valor de *secure_file_priv* for NULL, defina-o para o caminho que contém os dados da tabela exportada ('/var/lib/mysql-files/' no exemplo acima).

Para mais informações, consulte a documentação do MySQL ou a documentação do MariaDB.

`max_execution_time` (no MySQL) ou `max_statement_time` (no MariaDB) deve ser desabilitado antes de exportar os dados para evitar timeout durante a exportação.

Para MySQL:

```
SET @@max_execution_time=0;
```

Para 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
```

Siga as instruções pós-migração para remover as tabelas antigas.

PostgreSQL

A exportação e importação devem ser realizadas em tmux/screen para garantir que a sessão não seja encerrada. Para instalações com TimescaleDB, pule esta seção e prossiga para PostgreSQL + TimescaleDB.

Veja também: Notas importantes

Atualização de tabelas

- Renomeie as tabelas usando `history_upgrade_prepare.sql`:

```
sudo -u zabbix psql zabbix < /usr/share/zabbix-sql-scripts/postgresql/option-patches/history_upgrade_prepa
```

Migração com o server parado

- Exporte o histórico atual, importe para a tabela temporária, em seguida, insira os dados nas novas tabelas ignorando duplicatas:

```
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;
```

Veja dicas para melhorar o desempenho do INSERT: PostgreSQL: Bulk Loading Huge Amounts of Data, Checkpoint Distance and Amount of WAL.

- Siga as instruções pós-migração para remover as tabelas antigas.

Migração com o servidor em execução

- Exporte o histórico atual, importe-o para a tabela temporária, em seguida, insira os dados nas novas tabelas ignorando duplicatas:

```
\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;
```

- Siga as instruções pós-migração para remover as tabelas antigas.

PostgreSQL + TimescaleDB

A exportação e importação devem ser realizadas em tmux/screen para garantir que a sessão não seja encerrada. O servidor Zabbix deve estar desligado durante a atualização.

Veja também: Notas importantes

- Renomeie as tabelas usando `history_upgrade_prepare.sql`.
  - Se a compactação estiver habilitada (na instalação padrão), execute o script de `/usr/share/zabbix-sql-scripts/postgres`

```
cat /usr/share/zabbix-sql-scripts/postgresql/timescaledb/option-patches/with-compression/history_
```

- Se a compactação estiver desabilitada, execute o script de `/usr/share/zabbix-sql-scripts/postgresql/timescaledb/`

```
cat /usr/share/zabbix-sql-scripts/postgresql/timescaledb/option-patches/without-compression/histo
```

- Execute os scripts de migração do hypertable do TimescaleDB com base nas configurações de compactação:
  - Se a compactação estiver habilitada (na instalação padrão), execute os scripts de `/usr/share/zabbix-sql-scripts/postgre`

```
cat /usr/share/zabbix-sql-scripts/postgresql/timescaledb/option-patches/with-compression/history_
cat /usr/share/zabbix-sql-scripts/postgresql/timescaledb/option-patches/with-compression/history_
cat /usr/share/zabbix-sql-scripts/postgresql/timescaledb/option-patches/with-compression/history_
cat /usr/share/zabbix-sql-scripts/postgresql/timescaledb/option-patches/with-compression/history_
cat /usr/share/zabbix-sql-scripts/postgresql/timescaledb/option-patches/with-compression/history_
cat /usr/share/zabbix-sql-scripts/postgresql/timescaledb/option-patches/with-compression/trends_u
```

  - Se a compactação estiver desabilitada, execute os scripts de `/usr/share/zabbix-sql-scripts/postgresql/timescaledb`

```
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
```

Veja também: Dicas para melhorar o desempenho do INSERT.

- Siga as instruções pós-migração para remover as tabelas antigas.

Oracle

> **Attention:**
> O suporte ao Oracle DB está obsoleto desde o Zabbix 7.0.

A exportação e importação devem ser realizadas em tmux/screen para garantir que a sessão não seja encerrada. O Zabbix server deve estar desligado durante a atualização.

Veja também: Notas importantes

Atualização de tabelas

- Instale o Oracle Data Pump (disponível no pacote Instant Client Tools).

Consulte a documentação do Oracle Data Pump para dicas de desempenho.

- Renomeie as tabelas usando `history_upgrade_prepare.sql`.

```
cd /path/to/zabbix-sources/database/oracle/option-patches
sqlplus zabbix/password@oracle_host/service
sqlplus> @history_upgrade_prepare.sql
```

Migração em lote das tabelas de histórico

- Prepare os diretórios para o Data Pump.

O Data Pump deve ter permissões de leitura e gravação nesses diretórios.

Exemplo:

```
mkdir -pv /export/history
chown -R oracle:oracle /export
```

- Crie um objeto de diretório e conceda permissões de leitura e gravação a esse objeto para o usuário usado para autenticação no Zabbix ('zabbix' no exemplo abaixo). Sob o papel *sysdba*, execute:

```
create directory history as '/export/history';
grant read,write on directory history to zabbix;
```

- Exporte as tabelas. Substitua N pela contagem de threads desejada.

```
expdp zabbix/password@oracle_host/service \
    DIRECTORY=history \
    TABLES=history_old,history_uint_old,history_str_old,history_log_old,history_text_old \
    PARALLEL=N
```

- Importe as tabelas. Substitua N pela contagem de threads desejada.

```
impdp zabbix/password@oracle_host/service \
    DIRECTORY=history \
    TABLES=history_uint_old \
 REMAP_TABLE=history_old:history,history_uint_old:history_uint,history_str_old:history_str,history_log_old
    data_options=SKIP_CONSTRAINT_ERRORS table_exists_action=APPEND  PARALLEL=N CONTENT=data_only
```

- Siga as instruções pós-migração para remover as tabelas antigas.

Migração individual das tabelas de histórico

- Prepare diretórios para o Data Pump para cada tabela de histórico. O Data Pump deve ter permissões de leitura e gravação nesses diretórios.

Exemplo:

```
mkdir -pv /export/history /export/history_uint /export/history_str /export/history_log /export/history_tex
chown -R oracle:oracle /export
```

- Crie um objeto de diretório e conceda permissões de leitura e gravação a este objeto para o usuário usado para autenticação do Zabbix ('zabbix' no exemplo abaixo). Sob o papel *sysdba*, execute:

```
create directory history as '/export/history';
grant read,write on directory history to zabbix;

create directory history_uint as '/export/history_uint';
grant read,write on directory history_uint to zabbix;

create directory history_str as '/export/history_str';
grant read,write on directory history_str to zabbix;

create directory history_log as '/export/history_log';
grant read,write on directory history_log to zabbix;

create directory history_text as '/export/history_text';
grant read,write on directory history_text to zabbix;
```

- Exporte e importe cada tabela. Substitua N pela contagem de threads desejada.

```
expdp zabbix/password@oracle_host:1521/xe DIRECTORY=history TABLES=history_old PARALLEL=N

impdp zabbix/password@oracle_host:1521/xe DIRECTORY=history TABLES=history_old REMAP_TABLE=history_old:his

expdp zabbix/password@oracle_host:1521/xe DIRECTORY=history_uint TABLES=history_uint_old PARALLEL=N

impdp zabbix/password@oracle_host:1521/xe DIRECTORY=history_uint TABLES=history_uint_old REMAP_TABLE=histo

expdp zabbix/password@oracle_host:1521/xe DIRECTORY=history_str TABLES=history_str_old PARALLEL=N

impdp zabbix/password@oracle_host:1521/xe DIRECTORY=history_str TABLES=history_str_old REMAP_TABLE=history

expdp zabbix/password@oracle_host:1521/xe DIRECTORY=history_log TABLES=history_log_old PARALLEL=N

impdp zabbix/password@oracle_host:1521/xe DIRECTORY=history_log TABLES=history_log_old REMAP_TABLE=history

expdp zabbix/password@oracle_host:1521/xe DIRECTORY=history_text TABLES=history_text_old PARALLEL=N

impdp zabbix/password@oracle_host:1521/xe DIRECTORY=history_text TABLES=history_text_old REMAP_TABLE=histo
```

- Siga as instruções pós-migração para remover as tabelas antigas.

Pós-migração

Para todos os bancos de dados, uma vez que a migração seja concluída, faça o seguinte:

- Verifique se tudo está funcionando como esperado.
- Exclua as tabelas antigas:

```
DROP TABLE history_old;
DROP TABLE history_uint_old;
DROP TABLE history_str_old;
DROP TABLE history_log_old;
DROP TABLE history_text_old;
```

- Para o TimescaleDB, exclua também a seguinte tabela antiga:

```
DROP TABLE trends_old;
```

Veja também

- Preparando a tabela auditlog para particionamento

4 Preparando a tabela auditlog para particionamento

Visão geral

Alguns bancos de dados (por exemplo, MySQL) exigem que a coluna de particionamento faça parte da restrição exclusiva da tabela. Portanto, para particionar a tabela `auditlog` por tempo, a chave primária deve ser alterada de `auditid` para uma chave composta `auditid` + `clock`.

Esta seção fornece instruções para alterar a chave primária da tabela `auditlog`.

> **Attention:**
> As instruções fornecidas nesta página são destinadas a usuários avançados. Observe que essas instruções podem precisar ser ajustadas para sua configuração específica. Alterar a chave primária também pode ser incompatível com patches de atualização futuros, portanto, pode ser necessário lidar manualmente com futuras atualizações.

 Alterar a chave primária pode ser uma operação que consome muitos recursos e leva muito tempo, dependendo do tamanho da tabela `auditlog`. Recomenda-se parar o Zabbix server e alternar o Zabbix frontend para o modo de manutenção durante o tempo da alteração. No entanto, se absolutamente necessário, existe uma maneira de alterar a chave primária sem tempo de inatividade (veja abaixo).

Particionar a tabela `auditlog` pode melhorar, por exemplo, a limpeza em instalações grandes. Embora o housekeeping do Zabbix atualmente não possa tirar proveito de tabelas particionadas (exceto para TimescaleDB), você pode desabilitar o housekeeping do Zabbix e excluir partições usando scripts.

Desde o Zabbix 7.0, a tabela `auditlog` para TimescaleDB foi convertida em uma hypertable, o que permite que o housekeeper exclua dados por chunks. Para atualizar a tabela `auditlog` existente para uma hypertable, consulte Atualizando o schema do TimescaleDB.

MySQL

Notas importantes sobre a reconstrução de índices

O MySQL reconstrói automaticamente os índices para a chave primária durante a operação `ALTER TABLE`. No entanto, é altamente recomendável também reconstruir manualmente os índices com a instrução `OPTIMIZE TABLE` para garantir o desempenho ideal do banco de dados.

A reconstrução de índices pode exigir temporariamente tanto espaço em disco adicional quanto a própria tabela utiliza. Para obter o tamanho atual dos dados e dos índices, você pode executar as seguintes instruções:

```
ANALYZE TABLE auditlog;
SHOW TABLE STATUS LIKE 'auditlog';
```

Se o espaço disponível em disco for uma preocupação, siga as instruções de Alterando a chave primária sem tempo de inatividade. Outras opções também estão disponíveis:

- Aumentar o parâmetro do MySQL `sort_buffer_size` pode ajudar a reduzir o uso de espaço em disco ao reconstruir índices manualmente. No entanto, modificar esta variável pode impactar o uso geral de memória do banco de dados.
- Considere liberar espaço excluindo dados potencialmente desnecessários.
- Considere diminuir o parâmetro *Período de armazenamento de dados* do housekeeper antes de executar o housekeeper.

Alterando a chave primária com downtime

1. Remova a chave primária atual da tabela `auditlog` e adicione a nova chave primária.

```
ALTER TABLE auditlog DROP PRIMARY KEY, ADD PRIMARY KEY (auditid, clock);
```

2. Reconstrua os índices (opcional, mas altamente recomendado, veja Notas importantes sobre a reconstrução de índices).

```
OPTIMIZE TABLE auditlog;
```

Alterando a chave primária sem tempo de inatividade

O método manual de alteração da chave primária é descrito aqui. Como alternativa, você pode usar o pt-online-schema-change toolkit da Percona. Esta ferramenta executa as seguintes ações automaticamente, além de minimizar o espaço utilizado para alterar a tabela `auditlog`.

1. Crie uma nova tabela com a nova chave primária e crie os índices.

```
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. Troque as tabelas.

```
RENAME TABLE auditlog TO auditlog_old, auditlog_new TO auditlog;
```

3. Copie os dados da tabela antiga para a nova tabela.

```
INSERT INTO auditlog SELECT * FROM auditlog_old;
```

Isso pode ser feito em partes (várias instruções `INSERT INTO` com cláusulas `WHERE clock` conforme necessário) para evitar uso excessivo de recursos.

4. Exclua a tabela antiga.

```
DROP TABLE auditlog_old;
```

PostgreSQL

Notas importantes sobre a reconstrução de índices

O PostgreSQL reconstrói automaticamente os índices para a chave primária durante a operação `ALTER TABLE`. No entanto, é altamente recomendável também reconstruir manualmente os índices com a instrução `REINDEX TABLE CONCURRENTLY` para garantir o desempenho ideal do banco de dados.

A reconstrução de índices pode exigir temporariamente até três vezes o espaço em disco atualmente usado pelos índices. Para obter o tamanho atual dos índices, você pode executar a seguinte consulta:

```
SELECT pg_size_pretty(pg_indexes_size('auditlog'));
```

Se o espaço em disco disponível for uma preocupação, siga as instruções de Alterando a chave primária sem tempo de inatividade. Outras opções também estão disponíveis:

- Aumentar o parâmetro do PostgreSQL `maintenance_work_mem` pode ajudar a reduzir o uso de espaço em disco ao reconstruir manualmente os índices. No entanto, modificar esta variável pode impactar o uso geral de memória do banco de dados.
- Se você tiver outro disco ou tablespace com mais espaço disponível, pode considerar alterar o local de armazenamento temporário para a reconstrução do índice. Você pode definir o parâmetro do PostgreSQL `temp_tablespaces` para especificar um tablespace diferente para objetos temporários.
- Considere liberar espaço excluindo dados potencialmente desnecessários.
- Considere diminuir o parâmetro *Período de armazenamento de dados* housekeeper antes de executar o housekeeper.

Alterando a chave primária com tempo de inatividade

1. Remova a chave primária atual da tabela auditlog e adicione a nova chave primária.

```
ALTER TABLE auditlog DROP CONSTRAINT auditlog_pkey;
ALTER TABLE auditlog ADD PRIMARY KEY (auditid,clock);
```

2. Reconstrua os índices (opcional, mas altamente recomendado, veja Notas importantes sobre a reconstrução de índices).

```
REINDEX TABLE CONCURRENTLY auditlog;
```

Alterando a chave primária sem tempo de inatividade

O método manual de alteração da chave primária é descrito aqui. Como alternativa, a extensão pg_repack pode ser considerada para criar uma nova tabela, copiar os dados e trocar as tabelas.

1. Crie uma nova tabela com a nova chave primária e crie os índices.

```
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. Troque as tabelas.

```
ALTER TABLE auditlog RENAME TO auditlog_old;
ALTER TABLE auditlog_new RENAME TO auditlog;
```

3. Copie os dados da tabela antiga para a nova tabela.

```
INSERT INTO auditlog SELECT * FROM auditlog_old;
```

Isso pode ser feito em partes (várias instruções INSERT INTO com cláusulas WHERE clock conforme necessário) para evitar uso excessivo de recursos.

4. Exclua a tabela antiga.

```
DROP TABLE auditlog_old;
```

Oracle

Notas importantes sobre a reconstrução de índices

O Oracle reconstrói automaticamente os índices para a chave primária durante a operação ALTER TABLE. No entanto, é altamente recomendável também reconstruir manualmente os índices com as instruções ALTER INDEX <index> REBUILD PARALLEL para garantir o desempenho ideal do banco de dados.

A reconstrução de índices pode exigir temporariamente um espaço significativo em disco.

Se o espaço disponível em disco for uma preocupação, siga as instruções em Alterando a chave primária sem tempo de inatividade. Outras opções também estão disponíveis:

- Aumentar o parâmetro Oracle SORT_AREA_SIZE pode ajudar a reduzir o uso de espaço em disco ao reconstruir índices manualmente. No entanto, modificar essa variável impactará o uso geral de memória do banco de dados.
- Você pode definir o grau de paralelismo usando a cláusula PARALLEL, por exemplo: ALTER INDEX auditlog_1 REBUILD PARALLEL 4
- Considere liberar espaço excluindo dados potencialmente desnecessários.
- Considere diminuir o parâmetro *Período de armazenamento de dados* housekeeper antes de executar o housekeeper.

Alterando a chave primária com downtime

1. Recupere o nome da restrição.

```
SELECT CONSTRAINT_NAME FROM all_constraints WHERE TABLE_NAME = 'AUDITLOG' AND CONSTRAINT_TYPE = 'P';
```

2. Remova a chave primária atual da tabela auditlog e adicione a nova chave primária.

```
ALTER TABLE auditlog DROP CONSTRAINT <constraint_name>;
ALTER TABLE auditlog ADD CONSTRAINT auditlog_pk PRIMARY KEY (auditid, clock);
```

3. Reconstrua os índices (opcional, mas altamente recomendado, veja Notas importantes sobre reconstrução de índices).

3.1. Obtenha os nomes dos índices.

```
SELECT index_name FROM user_indexes WHERE table_name='AUDITLOG';
```

3.2. Reconstrua cada índice.

```
ALTER INDEX auditlog_pk REBUILD PARALLEL;
ALTER INDEX auditlog_1 REBUILD PARALLEL;
ALTER INDEX auditlog_2 REBUILD PARALLEL;
ALTER INDEX auditlog_3 REBUILD PARALLEL;
```

Alterando a chave primária sem tempo de inatividade

1. Crie uma nova tabela com a nova chave primária e crie índices.

```
CREATE TABLE auditlog_new (
  auditid              nvarchar2(25)                          ,
  userid               number(20)                             NULL,
  username             nvarchar2(100)  DEFAULT ''             ,
  clock                number(10)      DEFAULT '0'            NOT NULL,
  ip                   nvarchar2(39)   DEFAULT ''             ,
  action               number(10)      DEFAULT '0'            NOT NULL,
  resourcetype         number(10)      DEFAULT '0'            NOT NULL,
  resourceid           number(20)                             NULL,
  resource_cuid        nvarchar2(25)                          ,
  resourcename         nvarchar2(255)  DEFAULT ''             ,
  recordsetid          nvarchar2(25)                          ,
  details              nclob           DEFAULT ''             ,
  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. Troque as tabelas.

```
ALTER TABLE auditlog RENAME TO auditlog_old;
ALTER TABLE auditlog_new RENAME TO auditlog;
```

3. Copie os dados da tabela antiga para a nova tabela.

```
INSERT INTO auditlog SELECT * FROM auditlog_old;
```

Isso pode ser feito em partes (várias instruções `INSERT INTO` com cláusulas `WHERE clock` conforme necessário) para evitar o uso excessivo de recursos.

4. Exclua a tabela antiga.

```
DROP TABLE auditlog_old;
```

Veja também

- Atualização do banco de dados para chaves primárias

5 Conexão segura com o banco de dados

Visão geral

Esta seção fornece etapas de configuração do Zabbix e exemplos de configuração para conexões TLS seguras entre:

Banco de dados	Componentes do Zabbix
MySQL	Zabbix frontend, Zabbix server, Zabbix proxy
PostgreSQL	Zabbix frontend, Zabbix server, Zabbix proxy

Para configurar a criptografia de conexão dentro do SGBD, consulte a documentação oficial do fornecedor para obter detalhes:

- MySQL: servidores de banco de dados de replicação de origem e réplica.
- MySQL: servidores de banco de dados de replicação de grupo, etc.
- PostgreSQL opções de criptografia.

Todos os exemplos são baseados nas versões GA do MySQL CE (8.0) e PgSQL (13) disponíveis através dos repositórios oficiais usando CentOS 8.

Requisitos

O seguinte é necessário para configurar a criptografia:

- Sistema operacional suportado pelo desenvolvedor com OpenSSL >=1.1.X ou alternativa.

Note:
Recomenda-se evitar sistemas operacionais em status de fim de vida, especialmente no caso de novas instalações

- Mecanismo de banco de dados (RDBMS) instalado e mantido a partir do repositório oficial fornecido pelo desenvolvedor. Os sistemas operacionais geralmente são fornecidos com versões desatualizadas do software de banco de dados para as quais o suporte à criptografia não é implementado, por exemplo, sistemas baseados em RHEL 7 e PostgreSQL 9.2, MariaDB 5.5 sem suporte à criptografia.

Terminologia

Definir esta opção obriga o uso de conexão TLS com o banco de dados a partir do Zabbix server/proxy e frontend para o banco de dados:

- `required` - conectar usando TLS como modo de transporte sem verificações de identidade
- `verify_ca` - conectar usando TLS e verificar o certificado
- `verify_full` - conectar usando TLS, verificar o certificado e verificar se a identidade do banco de dados (CN) especificada por DBHost corresponde ao seu certificado

Configuração do Zabbix

Frontend para o banco de dados

Uma conexão segura com o banco de dados pode ser configurada durante a instalação do frontend:

- Marque a caixa de seleção *Criptografia TLS do banco de dados* na etapa Configurar conexão com o BD para habilitar a criptografia de transporte.
- Marque a caixa de seleção *Verificar certificado do banco de dados* que aparece quando o campo *Criptografia TLS* está marcado para habilitar a criptografia com certificados.

Note:
Para MySQL, a caixa de seleção *Criptografia TLS do banco de dados* é desabilitada se o *Host do banco de dados* estiver definido como localhost, porque a conexão que usa um arquivo de soquete (no Unix) ou memória compartilhada (no Windows) não pode ser criptografada.
 Para PostgreSQL, a caixa de seleção *Criptografia TLS* é desabilitada se o valor do campo *Host do banco de dados* começar com uma barra ou o campo estiver vazio.

Os seguintes parâmetros ficam disponíveis no modo de criptografia TLS com certificados (se ambas as caixas de seleção estiverem marcadas):

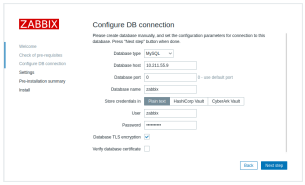
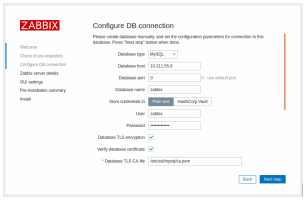
Parâmetro	Descrição
Arquivo CA TLS do banco de dados	Especifique o caminho completo para um arquivo de autoridade certificadora (CA) TLS válido.
Arquivo de chave TLS do banco de dados	Especifique o caminho completo para um arquivo de chave TLS válido.
Arquivo de certificado TLS do banco de dados	Especifique o caminho completo para um arquivo de certificado TLS válido.

Parâmetro	Descrição
Verificação do host do banco de dados	Marque esta caixa de seleção para ativar a verificação do host. Desabilitado para MYSQL, pois a biblioteca PHP MySQL não permite pular a etapa de validação do certificado do par.
Lista de cifras TLS do banco de dados	Especifique uma lista personalizada de cifras válidas. O formato da lista de cifras deve estar em conformidade com o padrão OpenSSL. Disponível apenas para MySQL.

Attention:
Os parâmetros TLS devem apontar para arquivos válidos. Se apontarem para arquivos inexistentes ou inválidos, isso resultará em erro de autorização.
 Se os arquivos de certificado forem graváveis, o frontend gera um aviso no relatório Informações do sistema de que "Os arquivos de certificado TLS devem ser somente leitura." (exibido apenas se o usuário PHP for o proprietário do certificado).
Certificados protegidos por senha não são suportados.

Casos de uso

O frontend do Zabbix usa a interface GUI para definir as opções possíveis: `required`, `verify_ca`, `verify_full`. Especifique as opções necessárias na etapa do assistente de instalação *Configurar conexões com o BD*. Essas opções são mapeadas para o arquivo de configuração (zabbix.conf.php) da seguinte maneira:

Configurações GUI	Arquivo de configuração	Descrição	Resultado
	... // Usado para conexão TLS. $DB['ENCRYPTION'] = true; $DB['KEY_FILE'] = ''; $DB['CERT_FILE'] = ''; $DB['CA_FILE'] = ''; $DB['VERIFY_HOST'] = false; $DB['CIPHER_LIST'] = ''; ...	Marque *Criptografia TLS do banco de dados* Deixe *Verificar certificado do banco de dados* desmarcado	Habilita o modo `required`.
	... $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. Marque *Criptografia TLS do banco de dados* e *Verificar certificado do banco de dados* 2. Especifique o caminho para *Arquivo CA TLS do banco de dados*	Habilita o modo `verify_ca`.

Configurações GUI	Arquivo de configuração	Descrição	Resultado

	...
	// Usado para conexão TLS com lista de cifras estritamente definida.
	$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>';
	...
	
	Ou:
	
	...
	// Usado para conexão TLS sem lista de cifras definida - selecionada pelo servidor MySQL
	$DB['ENCRYPTION'] = true;
	$DB['KEY_FILE'] = '<key_file_path>';
	$DB['CERT_FILE'] = '<key_file_path>';
	$DB['CA_FILE'] = '<key_file_path>';
	$DB['VERIFY_HOST'] = true;
	$DB['CIPHER_LIST'] = '';
	...
		1. Marque *Criptografia TLS do banco de dados* e *Verificar certificado do banco de dados*
		2. Especifique o caminho para *Arquivo de chave TLS do banco de dados*
		3. Especifique o caminho para *Arquivo CA TLS do banco de dados*
		4. Especifique o caminho para *Arquivo de certificado TLS do banco de dados*
		5. Especifique *Lista de cifras TLS do banco de dados* (opcional)
			Habilita o modo `verify_full` para MySQL.

	... $DB['ENCRYPTION'] = true; $DB['KEY_FILE'] = '<key_file_path>'; $DB['CERT_FILE'] = '<key_file_path>'; $DB['CA_FILE'] = '<key_file_path>'; $DB['VERIFY_HOST'] = true; $DB['CIPHER_LIST'] = ' '; ...	1. Marque *Criptografia TLS do banco de dados* e *Verificar certificado do banco de dados* 2. Especifique o caminho para *Arquivo de chave TLS do banco de dados* 3. Especifique o caminho para *Arquivo CA TLS do banco de dados* 4. Especifique o caminho para *Arquivo de certificado TLS do banco de dados* 5. Marque *Verificação do host do banco de dados*	Habilita o modo `verify_full` para PostgreSQL.

Veja também: Exemplos de configuração de criptografia para MySQL, Exemplos de configuração de criptografia para PostgreSQL.

Configuração do server/proxy do Zabbix

Conexões seguras com o banco de dados podem ser configuradas com os respectivos parâmetros no arquivo de configuração do server e/ou do proxy do Zabbix.

Configuração	Resultado
Nenhuma	Conexão com o banco de dados sem criptografia.
1. Defina `DBTLSConnect=required`	O server/proxy faz uma conexão TLS com o banco de dados. Uma conexão não criptografada não é permitida.
1. Defina `DBTLSConnect=verify_ca` 2. Defina `DBTLSCAFile` - especifique o arquivo da autoridade certificadora TLS	O server/proxy faz uma conexão TLS com o banco de dados após verificar o certificado do banco de dados.
1. Defina `DBTLSConnect=verify_full` 2. Defina `DBTLSCAFile` - especifique o arquivo da autoridade certificadora TLS	O server/proxy faz uma conexão TLS com o banco de dados após verificar o certificado do banco de dados e a identidade do host do banco de dados.
1. Defina `DBTLSCAFile` - especifique o arquivo da autoridade certificadora TLS 2. Defina `DBTLSCertFile` - especifique o arquivo do certificado de chave pública do cliente 3. Defina `DBTLSKeyFile` - especifique o arquivo da chave privada do cliente	O server/proxy fornece um certificado de cliente ao conectar-se ao banco de dados.
1. Defina `DBTLSCipher` - a lista de cifras de criptografia que o cliente permite para conexões usando protocolos TLS até o TLS 1.2 ou `DBTLSCipher13` - a lista de cifras de criptografia que o cliente permite para conexões usando o protocolo TLS 1.3	(MySQL) A conexão TLS é feita usando uma cifra da lista fornecida. (PostgreSQL) Definir esta opção será considerado um erro.

1 Configuração de criptografia do MySQL

Visão geral

Esta seção fornece vários exemplos de configuração de criptografia para CentOS 8.2 e MySQL 8.0.30 e pode ser usada como um guia rápido para criptografar a conexão com o banco de dados.

Attention:
Se o host do MySQL estiver definido como localhost, as opções de criptografia não estarão disponíveis. Neste caso, uma conexão entre o frontend do Zabbix e o banco de dados usa um arquivo de soquete (no Unix) ou memória compartilhada (no Windows) e não pode ser criptografada.

Note:
A lista de combinações de criptografia não se limita às listadas nesta página. Existem muitas outras combinações disponíveis.

Pré-requisitos

Instale o banco de dados MySQL a partir do repositório oficial.

Consulte a documentação do MySQL para obter detalhes sobre como usar o repositório do MySQL.

O servidor MySQL está pronto para aceitar conexões seguras usando um certificado autoassinado.

Para ver quais usuários estão usando uma conexão criptografada, execute a seguinte consulta (Performance Schema deve estar ativado):

```
mysql> SELECT sbt.variable_value AS tls_version, t2.variable_value AS cipher, processlist_user AS user, pr
        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;
```

Criptografia apenas de transporte

Configuração do MySQL

Versões modernas do banco de dados já estão prontas para uso no modo de criptografia `required`. Um certificado do lado do servidor será criado após a configuração e inicialização.

Crie usuários e funções para os principais componentes:

> **Attention:**
> Para versões do MySQL 8.4+, `caching_sha2_password` deve ser usado em vez de `mysql_native_password`.

```
mysql> CREATE USER
 'zbx_srv'@'%' IDENTIFIED WITH mysql_native_password BY '<strong_password>',
 'zbx_web'@'%' IDENTIFIED WITH mysql_native_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'@'%';
```

Observe que o protocolo X.509 não é usado para verificar a identidade, mas o usuário está configurado para usar apenas conexões criptografadas. Consulte a documentação do MySQL para mais detalhes sobre a configuração de usuários.

Execute para verificar a conexão (a conexão via socket não pode ser usada para testar conexões seguras):

```
mysql -u zbx_srv -p -h 10.211.55.9 --ssl-mode=REQUIRED
```

Verifique o status atual e as suítes de criptografia disponíveis:

```
mysql> status
--------------
mysql Ver 8.0.21 for Linux on x86_64 (MySQL Community Server - GPL)

Connection id: 62
Current database:
Current user: zbx_srv@bfdb.local
SSL: Cipher in use is TLS_AES_256_GCM_SHA384

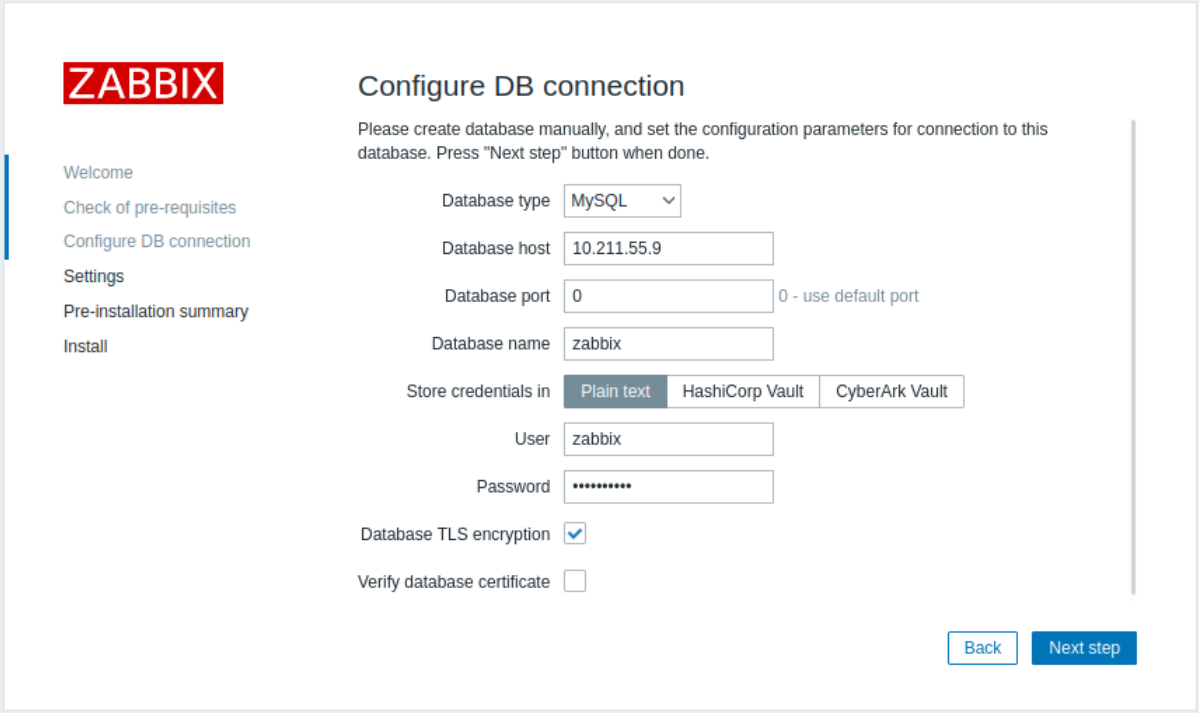

mysql> SHOW SESSION STATUS LIKE 'Ssl_cipher_list'\G;
*************************** 1. row ***************************
Variable_name: Ssl_cipher_list
Value: TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256:TLS_AES_128_CCM_SHA256:E
1 row in set (0.00 sec)

ERROR:
No query specified
```

Frontend

Para habilitar a criptografia apenas de transporte para conexões entre o frontend do Zabbix e o banco de dados:

- Marque *Criptografia TLS do banco de dados*
- Deixe *Verificar certificado do banco de dados* desmarcado



Server

Para habilitar a criptografia apenas para transporte nas conexões entre o server e o banco de dados, configure */etc/zabbix/zabbix_server.conf*:

```
...
DBHost=10.211.55.9
DBName=zabbix
DBUser=zbx_srv
DBPassword=<strong_password>
DBTLSConnect=required
...
```

Criptografia com verificação de autoridade certificadora

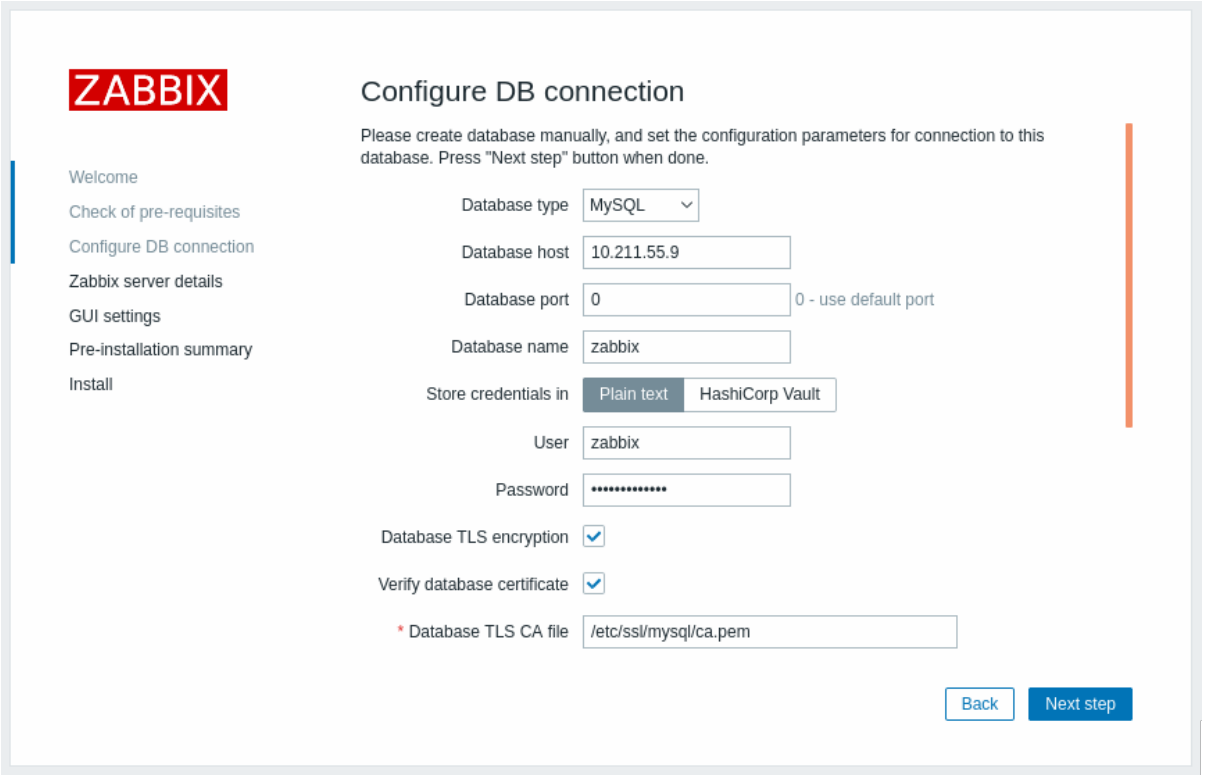
Copie o CA do MySQL necessário para o servidor frontend do Zabbix, atribua as permissões adequadas para permitir que o servidor web leia este arquivo.

Note:
Este modo não funciona no SLES 12 e RHEL 7 devido a bibliotecas MySQL antigas.

Frontend

Para habilitar a criptografia com verificação de certificado para conexões entre o frontend do Zabbix e o banco de dados:

- Marque *Criptografia TLS do banco de dados* e *Verificar certificado do banco de dados*
- Especifique o caminho para o arquivo CA TLS do banco de dados



Alternativamente, isso pode ser definido em */etc/zabbix/web/zabbix.conf.php*:

```
...
$DB['ENCRYPTION'] = true;
$DB['KEY_FILE'] = '';
$DB['CERT_FILE'] = '';
$DB['CA_FILE'] = '/etc/ssl/mysql/ca.pem';
$DB['VERIFY_HOST'] = false;
$DB['CIPHER_LIST'] = '';
...
```

Solucione problemas usando a ferramenta de linha de comando para verificar se a conexão é possível para o usuário necessário:

```
mysql -u zbx_web -p -h 10.211.55.9 --ssl-mode=REQUIRED --ssl-ca=/var/lib/mysql/ca.pem
```

Server

Para habilitar a criptografia com verificação de certificado para conexões entre o Zabbix server e o banco de dados, configure */etc/zabbix/zabbix_server.conf*:

```
...
DBHost=10.211.55.9
DBName=zabbix
DBUser=zbx_srv
DBPassword=<strong_password>
DBTLSConnect=verify_ca
DBTLSCAFile=/etc/ssl/mysql/ca.pem
...
```

Criptografia com verificação completa

Configuração do MySQL

Defina a opção de configuração do servidor MySQL CE (*/etc/my.cnf.d/server-tls.cnf*) para:

```
[mysqld]
...
# neste exemplo, as chaves estão localizadas no diretório datadir do MySQL CE
ssl_ca=ca.pem
ssl_cert=server-cert.pem
ssl_key=server-key.pem

require_secure_transport=ON
```

```
tls_version=TLSv1.3
...
```

As chaves para o servidor MySQL CE e o cliente (frontend do Zabbix) devem ser criadas manualmente de acordo com a documentação do MySQL CE: Criando certificados e chaves SSL e RSA usando o MySQL ou Criando certificados e chaves SSL usando o openssl

Attention:
O certificado do servidor MySQL deve conter o campo Common Name definido para o nome FQDN, pois o frontend do Zabbix usará o nome DNS para se comunicar com o banco de dados ou o endereço IP do host do banco de dados.

Crie o usuário MySQL:

Attention:
Para versões do MySQL 8.4+, `caching_sha2_password` deve ser usado em vez de `mysql_native_password`.

```
mysql> CREATE USER
  'zbx_srv'@'%' IDENTIFIED WITH mysql_native_password BY '<strong_password>',
  'zbx_web'@'%' IDENTIFIED WITH mysql_native_password BY '<strong_password>'
  REQUIRE X509
  PASSWORD HISTORY 5;
```

Verifique se é possível fazer login com esse usuário:

```
mysql -u zbx_web -p -h 10.211.55.9 --ssl-mode=VERIFY_IDENTITY --ssl-ca=/var/lib/mysql/ca.pem --ssl-cert=/v
```

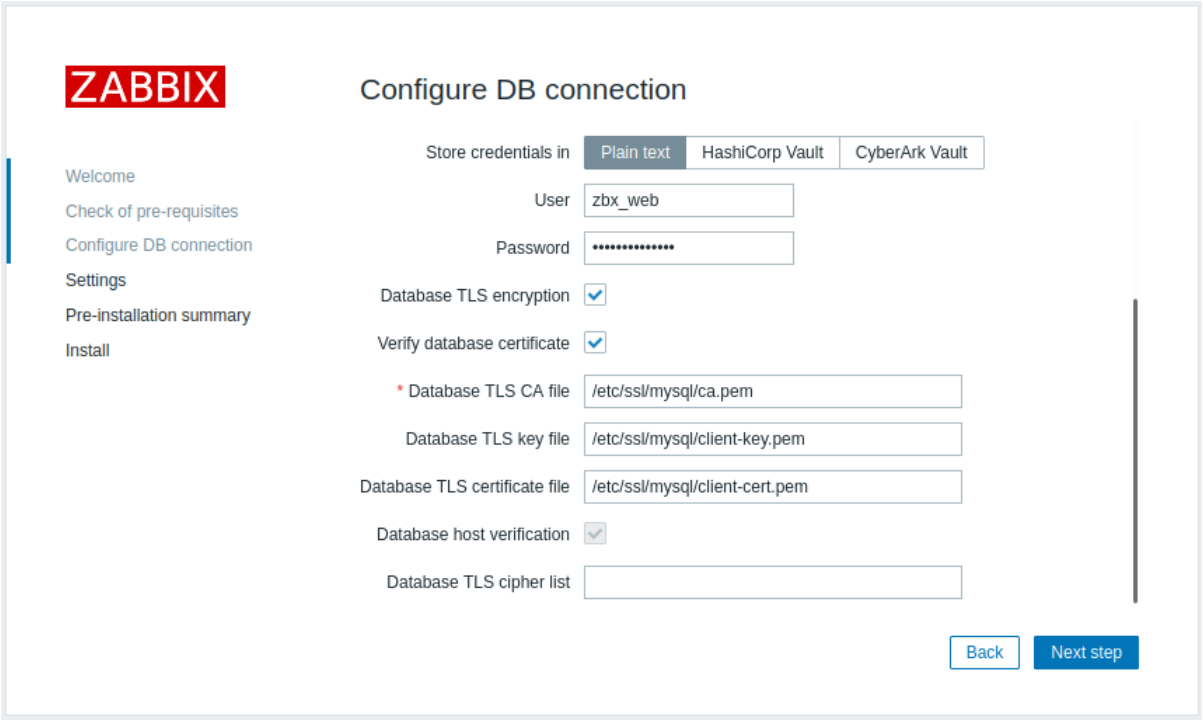
Frontend

Para habilitar a criptografia com verificação completa para conexões entre o frontend do Zabbix e o banco de dados:

- Marque *Database TLS encryption* e *Verify database certificate*
- Especifique o caminho para *Database TLS key file*
- Especifique o caminho para *Database TLS CA file*
- Especifique o caminho para *Database TLS certificate file*

Observe que *Database host verification* está marcada e desabilitada - esta etapa não pode ser ignorada para MySQL.

Warning:
Se o campo *Database TLS cipher list* for deixado em branco, os cifradores comuns permitidos tanto pelo frontend (cliente) quanto pelo servidor serão habilitados. Alternativamente, os cifradores podem ser definidos explicitamente, em conformidade com os requisitos de configuração de cifradores.



Alternativamente, isso pode ser definido em */etc/zabbix/web/zabbix.conf.php*:

```
...
// Usado para conexão TLS com lista de cifradores estritamente definida.
$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
...
// ou
...
// Usado para conexão TLS sem lista de cifradores definida - selecionada pelo servidor MySQL
$DB['ENCRYPTION'] = true;
$DB['KEY_FILE'] = '/etc/ssl/mysql/client-key.pem';
$DB['CERT_FILE'] = '/etc/ssl/mysql/client-cert.pem';
$DB['CA_FILE'] = '/etc/ssl/mysql/ca.pem';
$DB['VERIFY_HOST'] = true;
$DB['CIPHER_LIST'] = '';
...
```

Server

Para habilitar a criptografia com verificação completa para conexões entre o Zabbix server e o banco de dados, configure */etc/zabbix/zabbix_server.conf*:

```
...
DBHost=10.211.55.9
DBName=zabbix
DBUser=zbx_srv
DBPassword=<strong_password>
DBTLSConnect=verify_full
DBTLSCAFile=/etc/ssl/mysql/ca.pem
DBTLSCertFile=/etc/ssl/mysql/client-cert.pem
DBTLSKeyFile=/etc/ssl/mysql/client-key.pem
...
```

2 Configuração de criptografia do PostgreSQL

Visão geral

Esta seção fornece vários exemplos de configuração de criptografia para CentOS 8.2 e PostgreSQL 13.

> **Note:**
> A conexão entre o frontend do Zabbix e o PostgreSQL não pode ser criptografada (os parâmetros na interface gráfica estão desabilitados) se o valor do campo *Database host* começar com uma barra ou se o campo estiver vazio.

Pré-requisitos

Instale o banco de dados PostgreSQL usando o repositório oficial.

O PostgreSQL não está configurado para aceitar conexões TLS por padrão. Siga as instruções da documentação do PostgreSQL para preparação de certificado com postgresql.conf e também para controle de acesso de usuário através do ph_hba.conf.

Por padrão, o socket do PostgreSQL está vinculado ao localhost, para as conexões remotas de rede permitirem a escuta na interface de rede real.

As configurações do PostgreSQL para todos os modos podem ser assim:

/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'
...
```

Para controle de acesso ajuste */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

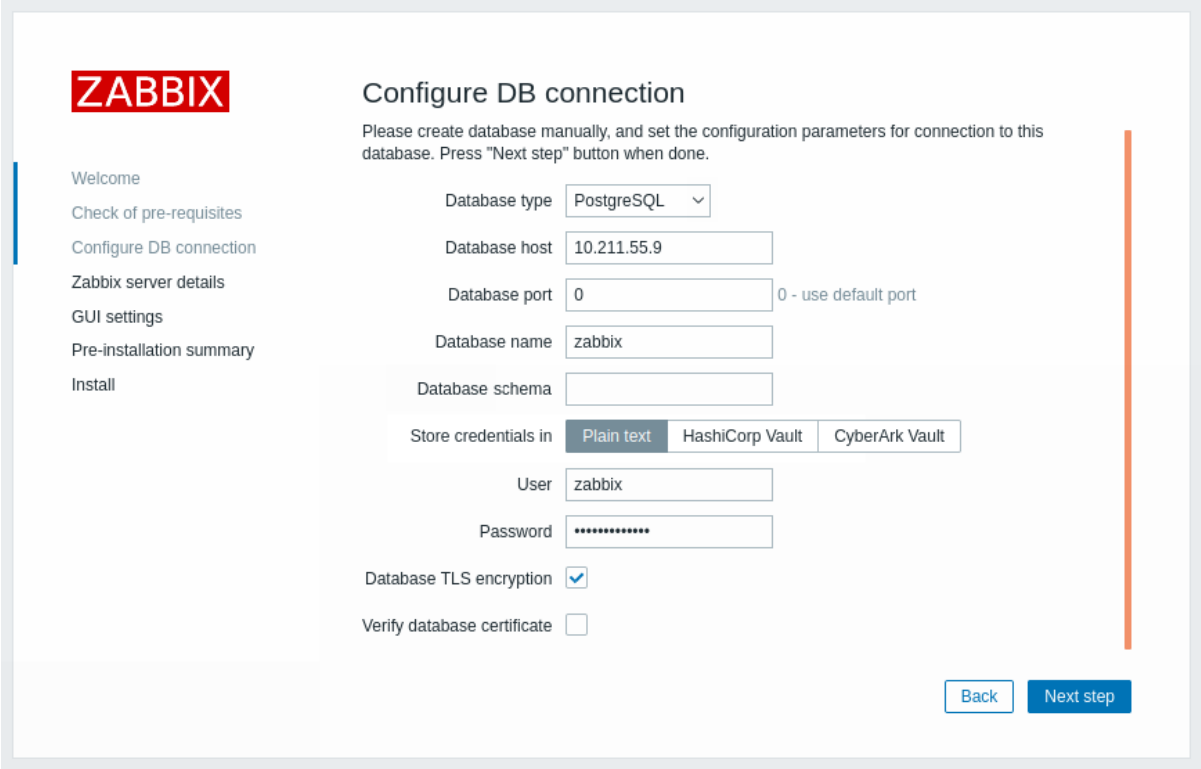
### verify full
hostssl all all 0.0.0.0/0 md5 clientcert=verify-full
...
```

Criptografia apenas de transporte

Frontend

Para habilitar a criptografia apenas de transporte para conexões entre o frontend do Zabbix e o banco de dados:

- Marque *Criptografia TLS do banco de dados*
- Deixe *Verificar certificado do banco de dados* desmarcado



Server

Para habilitar a criptografia apenas para transporte nas conexões entre o server e o banco de dados, configure */etc/zabbix/zabbix_server.conf*:

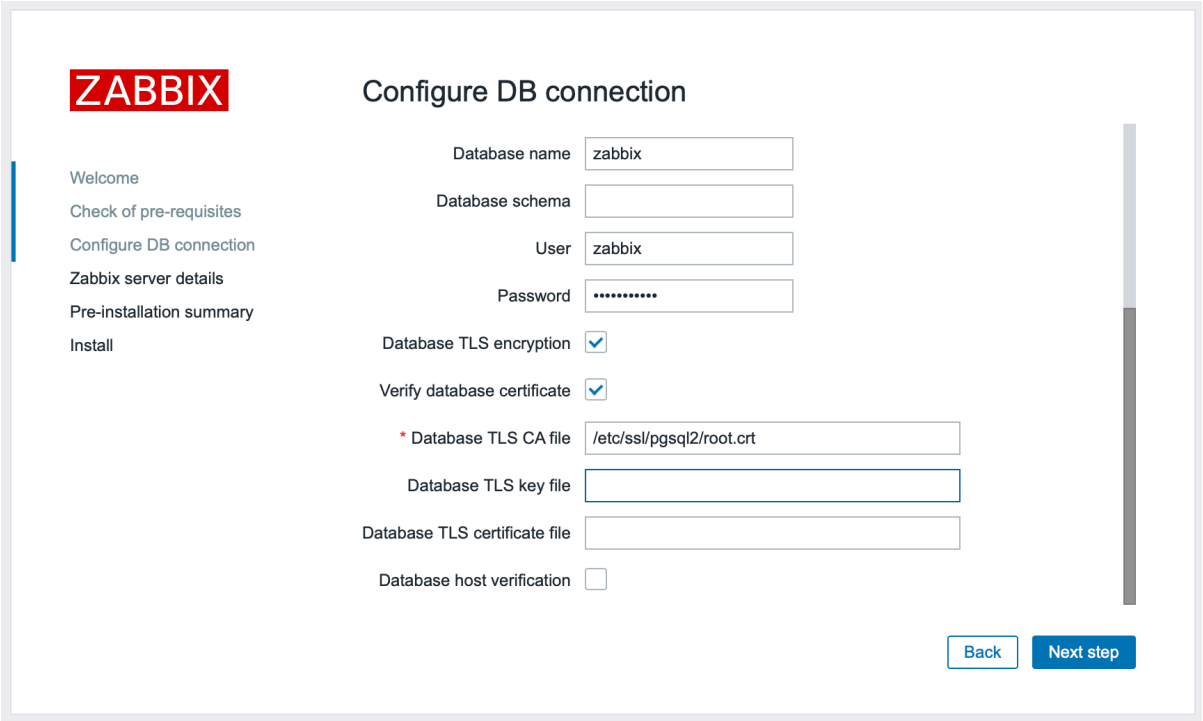
```
...
DBHost=10.211.55.9
DBName=zabbix
DBUser=zbx_srv
DBPassword=<strong_password>
DBTLSConnect=required
...
```

Criptografia com verificação de autoridade de certificação

Frontend

Para habilitar a criptografia com verificação de autoridade certificadora para conexões entre o frontend do Zabbix e o banco de dados:

- Marque *Criptografia TLS do banco de dados* e *Verificar certificado do banco de dados*
- Especifique o caminho para o *Arquivo CA TLS do banco de dados*



Alternativamente, isso pode ser definido em */etc/zabbix/web/zabbix.conf.php:*

```
...
$DB['ENCRYPTION'] = true;
$DB['KEY_FILE'] = '';
$DB['CERT_FILE'] = '';
$DB['CA_FILE'] = '/etc/ssl/pgsql/root.crt';
$DB['VERIFY_HOST'] = false;
$DB['CIPHER_LIST'] = '';
...
```

Server

Para habilitar a criptografia com verificação de certificado para conexões entre o Zabbix server e o banco de dados, configure */etc/zabbix/zabbix_server.conf:*

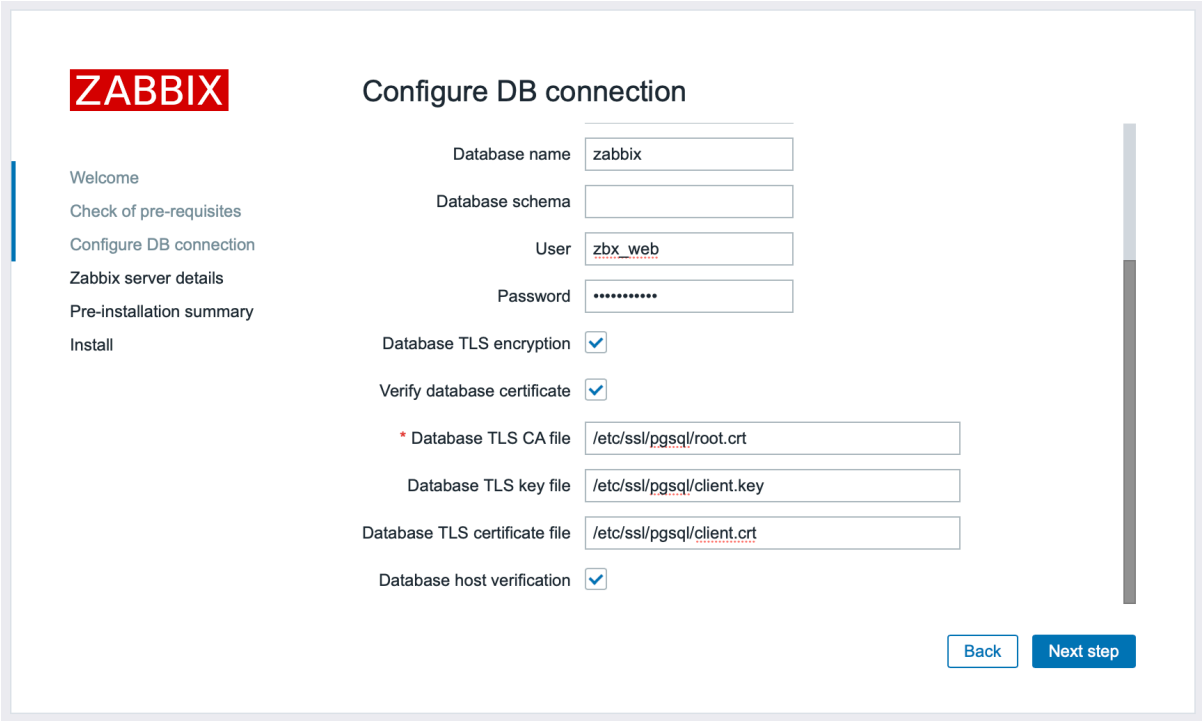
```
...
DBHost=10.211.55.9
DBName=zabbix
DBUser=zbx_srv
DBPassword=<strong_password>
DBTLSConnect=verify_ca
DBTLSCAFile=/etc/ssl/pgsql/root.crt
...
```

Criptografia com verificação completa

Frontend

Para habilitar a criptografia com certificado e verificação de identidade do host do banco de dados para conexões entre o frontend do Zabbix e o banco de dados:

- Marque *Criptografia TLS do banco de dados* e *Verificar certificado do banco de dados*
- Especifique o caminho para o *Arquivo de chave TLS do banco de dados*
- Especifique o caminho para o *Arquivo CA TLS do banco de dados*
- Especifique o caminho para o *Arquivo de certificado TLS do banco de dados*
- Marque *Verificação do host do banco de dados*



Alternativamente, isso pode ser definido em */etc/zabbix/web/zabbix.conf.php:*

```
$DB['ENCRYPTION'] = true;
$DB['KEY_FILE'] = '';
$DB['CERT_FILE'] = '';
$DB['CA_FILE'] = '/etc/ssl/pgsql/root.crt';
$DB['VERIFY_HOST'] = true;
$DB['CIPHER_LIST'] = '';
...
```

Server

Para habilitar a criptografia com verificação de certificado e identidade do host do banco de dados para conexões entre o Zabbix server e o banco de dados, configure */etc/zabbix/zabbix_server.conf*:

```
...
DBHost=10.211.55.9
DBName=zabbix
DBUser=zbx_srv
DBPassword=<strong_password>
DBTLSConnect=verify_full
DBTLSCAFile=/etc/ssl/pgsql/root.crt
DBTLSCertFile=/etc/ssl/pgsql/client.crt
DBTLSKeyFile=/etc/ssl/pgsql/client.key
...
```

6 Configuração do TimescaleDB

Visão geral

O Zabbix suporta o TimescaleDB, uma solução de banco de dados baseada em PostgreSQL que particiona automaticamente os dados em blocos baseados em tempo para oferecer melhor desempenho em larga escala.

Warning:
Atualmente, o TimescaleDB não é suportado pelo proxy Zabbix.

As instruções nesta página podem ser usadas para os seguintes cenários:

- Criar um banco de dados TimescaleDB ou migrar de tabelas PostgreSQL existentes para TimescaleDB (consulte Configuração).
- Atualizar um esquema de banco de dados TimescaleDB existente ao atualizar o Zabbix (consulte Atualizando o esquema do TimescaleDB).

Configuração

Pré-requisitos: Extensão TimescaleDB de uma versão suportada instalada no servidor de banco de dados. Para instruções de instalação, consulte a documentação do TimescaleDB.

> **Warning:**
> Antes de instalar o TimescaleDB, instale uma versão suportada do PostgreSQL a partir do repositório oficial do PostgreSQL.

Habilite a extensão TimescaleDB para o banco de dados específico executando:

```
echo "CREATE EXTENSION IF NOT EXISTS timescaledb CASCADE;" | sudo -u postgres psql zabbix
```

A execução deste comando requer privilégios de administrador do banco de dados.

> **Note:**
> Se você usar um schema de banco de dados diferente de 'public', será necessário adicionar uma cláusula SCHEMA ao comando acima. Exemplo:
 `echo "CREATE EXTENSION IF NOT EXISTS timescaledb SCHEMA yourschema CASCADE;" | sudo -u postgres psql zabbix`

Em seguida, execute o script `postgresql/timescaledb/schema.sql`. Para novas instalações, o script deve ser executado após o banco de dados PostgreSQL regular ter sido criado com o schema/dados iniciais (consulte criação do banco de dados).

```
cat /usr/share/zabbix-sql-scripts/postgresql/timescaledb/schema.sql | sudo -u zabbix psql zabbix
```

> **Attention:**
> Por favor, ignore as mensagens de aviso informando que as melhores práticas não estão sendo seguidas ao executar o script `schema.sql` em versões do TimescaleDB 2.9.0 ou superiores. Independentemente deste aviso, a configuração será concluída com sucesso.

A migração dos dados existentes de histórico, tendências e log de auditoria pode levar muito tempo. O Zabbix server e o frontend devem estar desligados durante o período de migração.

O script `schema.sql` define os seguintes parâmetros de limpeza:

- Substituir período de histórico do item
- Substituir período de tendência do item

Para usar a limpeza particionada para histórico e tendências, ambas as opções devem ser ativadas. Também é possível ativar a substituição individualmente, apenas para histórico ou apenas para tendências.

O script `postgresql/timescaledb/schema.sql` define dois parâmetros adicionais:

- Habilitar compressão
- Comprimir registros com mais de 7 dias

Para remover com sucesso os dados compactados pelo housekeeper, as opções *Substituir o período de histórico do item* e *Substituir o período de tendências do item* devem estar ativadas. Se a substituição estiver desativada e as tabelas tiverem blocos compactados, o housekeeper não removerá os dados dessas tabelas e avisos sobre configuração incorreta serão exibidos nas seções *Housekeeping* e *Informações do sistema*.

Todos esses parâmetros podem ser alterados em *Administração* > *Limpeza* após a instalação.

> **Note:**
> Você pode querer executar a ferramenta timescaledb-tune fornecida pelo TimescaleDB para otimizar os parâmetros de configuração do PostgreSQL em seu `postgresql.conf`.

Atualizando o schema do TimescaleDB

Ao atualizar o Zabbix para uma versão que contém novos hypertables do TimescaleDB, o Zabbix server não configura automaticamente esses hypertables (por exemplo, ao atualizar do Zabbix 6.4 para o 7.0.3, pois as versões 7.0.0 e 7.0.2 introduziram novos hypertables).

Para configurar novos hypertables do TimescaleDB, siga estas etapas:

1. Inicie o Zabbix server; isso atualizará o banco de dados existente.
2. Verifique no arquivo de log do server se a atualização do banco de dados foi concluída; quando concluída, pare o Zabbix server. Observe que o server registra um aviso se tentar habilitar a compactação para uma tabela que não é um hypertable.

3. Execute o script `postgresql/timescaledb/schema.sql`; isso configura os novos hypertables do TimescaleDB. Observe que, desde o Zabbix 7.0.0, a localização e o nome do script foram alterados de `postgresql/timescaledb.sql` para `postgresql/timescaledb/schema.sql`.

Attention:
Ignore as mensagens de aviso informando que as melhores práticas não estão sendo seguidas ao executar o script `schema.sql` no TimescaleDB versão 2.9.0 ou superior. Independentemente desse aviso, a configuração será concluída com sucesso.

Compressão TimescaleDB

A compressão nativa do TimescaleDB é suportada para todas as tabelas do Zabbix que são hypertables do TimescaleDB. Durante a atualização ou migração para o TimescaleDB, a compressão inicial das tabelas grandes pode levar muito tempo.

Observe que a compressão é suportada sob a licença Timescale Community "timescale" e não é suportada sob a licença Apache 2.0 "apache". Se o Zabbix detectar que a compressão não é suportada, uma mensagem de aviso será registrada no log do Zabbix server e os usuários não poderão habilitar a compressão no frontend.

Note:
Recomenda-se que os usuários se familiarizem com a compressão na documentação do TimescaleDB antes de usar a compressão.

Observe que existem certas limitações impostas pela compressão, especificamente:

- Modificações em chunks comprimidos (inserções, exclusões, atualizações) não são permitidas
- Alterações de esquema para tabelas comprimidas não são permitidas.

As configurações de compressão podem ser alteradas no bloco *Compressão de histórico, tendências e log de auditoria* em *Administração > Limpeza* na seção do frontend do Zabbix.

Parâmetro	Padrão	Comentários
Habilitar compressão	Habilitado	Marcar ou desmarcar a caixa de seleção não ativa/desativa a compressão imediatamente. Como a compressão é tratada pelo Housekeeper, as alterações entrarão em vigor em até 2 vezes o número de horas definido em `HousekeepingFrequency` (definido em zabbix_server.conf) Após desabilitar a compressão, novos chunks que entrarem no período de compressão não serão comprimidos. No entanto, todos os dados previamente comprimidos permanecerão comprimidos. Para descomprimir chunks previamente comprimidos, siga as instruções na documentação do TimescaleDB.
Comprimir registros mais antigos que	7d	Ao atualizar de versões mais antigas do Zabbix com suporte ao TimescaleDB, a compressão não será habilitada por padrão. Este parâmetro não pode ser inferior a 7 dias. Devido à imutabilidade dos chunks comprimidos, todos os dados atrasados (por exemplo, dados atrasados por um proxy) que sejam mais antigos que esse valor serão descartados.

Para melhor desempenho na atualização de tendências, você pode querer reduzir o "chunk_time_interval" para as tabelas `trends` e `trends_uint` de 30 dias para 7 dias ou menos, dependendo de quantos items usam tendências. O objetivo desta configuração é aderir às melhores práticas do TimescaleDB e garantir que o tamanho do chunk permaneça dentro dos recursos disponíveis do sistema.

7 Configuração do Elasticsearch

Attention:
O suporte ao Elasticsearch é experimental!

O Zabbix suporta o armazenamento de dados históricos por meio do Elasticsearch em vez de um banco de dados. Os usuários podem escolher o local de armazenamento dos dados históricos entre um banco de dados compatível e o Elasticsearch. O procedimento de configuração descrito nesta seção é aplicável à versão 7.X do Elasticsearch. Caso uma versão anterior ou posterior do Elasticsearch seja usada, algumas funcionalidades podem não funcionar conforme o esperado.

Warning:
1. Se todos os dados históricos forem armazenados no Elasticsearch, as tendências **não** são calculadas nem armazenadas no banco de dados. Sem tendências calculadas e armazenadas, o período de armazenamento do histórico pode precisar ser estendido.
 2. Quando o Elasticsearch é utilizado, as consultas de intervalo que recuperam valores do banco de dados são limitadas pelo timestamp do período de armazenamento dos dados.

Configuração

Para garantir a comunicação adequada entre todos os elementos envolvidos, certifique-se de os parâmetros do arquivo de configuração do servidor e do arquivo de configuração do frontend são configurado corretamente.

Servidor e frontend Zabbix

Rascunho do arquivo de configuração do servidor Zabbix com parâmetros a serem atualizados:

```
### Opção: HistoryStorageURL
# URL de armazenamento de histórico HTTP[S].
#
# Obrigatório: não
# Predefinição:
# HistoryStorageURL=
### Opção: HistoryStorageTypes
# Lista separada por vírgulas de tipos de valores a serem enviados para o armazenamento de histórico.
#
# Obrigatório: não
# Predefinição:
# HistoryStorageTypes=uint,dbl,str,log,texto
```

Valores de parâmetro de exemplo para preencher o arquivo de configuração do servidor Zabbix com:

```
HistoryStorageURL=http://test.elasticsearch.lan:9200
HistoryStorageTypes=str,log,texto
```

Esta configuração força o Zabbix Server a armazenar valores de histórico de tipos numéricos no banco de dados correspondente e dados de histórico textual em Pesquisa elástica.

O Elasticsearch oferece suporte aos seguintes tipos de itens:

```
uint,dbl,str,log,texto
```

Explicação do tipo de item suportado:

Tipo de valor de item	Tabela de banco de dados	Tipo de pesquisa elástica
Numérico (sem sinal)	história_uint	uint
Numérico (flutuante)	histórico	dbl
Personagem	história_str	str
Log	histórico_log	log
Texto	história_texto	texto

Rascunho do arquivo de configuração de frontend do Zabbix (`conf/zabbix.conf.php`) com parâmetros a serem atualizados:

```
// URL do Elasticsearch (pode ser string se o mesmo URL for usado para todos os tipos).
$HISTORY['url'] = [
      'uint' => 'http://localhost:9200',
      'texto' => 'http://localhost:9200'
];
// Tipos de valor armazenados no Elasticsearch.
$HISTORY['types'] = ['uint', 'text'];
```

Valores de parâmetro de exemplo para preencher o arquivo de configuração do frontend do Zabbix com:

```
$HISTORY['url'] = 'http://test.elasticsearch.lan:9200';
$HISTORY['types'] = ['str', 'text', 'log'];
```

Esta configuração força a armazenar o histórico de `Text`, `Character` e `Log` valores no Elasticsearch.

Também é necessário tornar `$HISTORY` global em `conf/zabbix.conf.php` para certifique-se de que tudo está funcionando corretamente (consulte `conf/zabbix.conf.php.example` para saber como fazer isso):

```
// Arquivo de configuração da interface gráfica do Zabbix.
global $DB, $HISTÓRICO;
```

Instalando o Elasticsearch e criando o mapeamento

Os dois últimos passos para fazer tudo funcionar são instalar o Elasticsearch e criar o processo de mapeamento.

Para instalar o Elasticsearch, consulte o guia de instalação do Elasticsearch.

Note:
O mapeamento é uma estrutura de dados no Elasticsearch (semelhante a uma tabela em um banco de dados). O mapeamento para todos os tipos de dados de histórico está disponível aqui: `database/elasticsearch/elasticsearch.map`.

Warning:
A criação do mapeamento é obrigatória. Algumas funcionalidades não funcionarão se o mapeamento não for criado de acordo com as instruções.

Para criar o mapeamento para o tipo `text`, envie a seguinte solicitação para o Elasticsearch:

```
curl -X PUT \
 http://your-elasticsearch.here: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"
         }
      }
   }
}'
```

Uma solicitação semelhante deve ser executada para a criação do mapeamento dos valores de histórico `Character` e `Log`, com a devida correção do tipo.

Note:
Para trabalhar com o Elasticsearch, consulte os Requisitos para informações adicionais.

> **Note:**
> O Housekeeper não exclui nenhum dado do Elasticsearch.

Armazenando dados do histórico em vários índices baseados em data

Esta seção descreve as etapas adicionais necessárias para trabalhar com pipelines e ingerir nós.

Para começar, você deve criar modelos para índices.

O exemplo a seguir mostra uma solicitação para criar o modelo uint:

```
curl -X PUT \
 http://your-elasticsearch.here:9200/_template/uint_template \
 -H 'content-type:application/json' \
 -d'{
   "index_patterns": [
      "uint*"
   ],
   "definições": {
      "índice": {
         "number_of_replicas": 1,
         "number_of_shards": 5
      }
   },
   "mapeamentos": {
      "propriedades": {
         "item": {
            "tipo": "longo"
         },
         "relógio": {
            "format": "epoch_second",
            "tipo": "data"
         },
         "valor": {
            "tipo": "longo"
         }
      }
   }
}'
```

Para criar outros modelos, o usuário deve alterar a URL (a última parte é a nome do modelo), altere o campo `"index_patterns"` para corresponder ao nome do índice e para definir um mapeamento válido, que pode ser obtido de `banco de dados/elasticsearch/elas`

Por exemplo, o comando a seguir pode ser usado para criar um modelo para índice de texto:

```
curl -X PUT \
 http://your-elasticsearch.here:9200/_template/text_template \
 -H 'content-type:application/json' \
 -d'{
   "index_patterns": [
      "texto*"
   ],
   "definições": {
      "índice": {
         "number_of_replicas": 1,
         "number_of_shards": 5
      }
   },
   "mapeamentos": {
      "propriedades": {
         "item": {
            "tipo": "longo"
         },
         "relógio": {
            "format": "epoch_second",
```

```
            "tipo": "data"
         },
         "valor": {
            "Campos": {
               "analisado": {
                  "índice": verdadeiro,
                  "tipo": "texto",
                  "analisador": "padrão"
               }
            },
            "índice": falso,
            "tipo": "texto"
         }
      }
   }
}'
```

Isso é necessário para permitir que o Elasticsearch defina um mapeamento válido para índices criado automaticamente. Em seguida, é necessário criar o pipeline definição. Pipeline é algum tipo de pré-processamento de dados antes colocando dados em índices. O comando a seguir pode ser usado para criar pipeline para o índice uint:

```
curl -X PUT \
 http://your-elasticsearch.here:9200/_ingest/pipeline/uint-pipeline \
 -H 'content-type:application/json' \
 -d'{
   "description": "nomeação diária do índice uint",
   "processadores": [
      {
         "date_index_name": {
            "campo": "relógio",
            "date_formats": [
               "UNIX"
            ],
            "index_name_prefix": "uint-",
            "date_rounding": "d"
         }
      }
   ]
}'
```

O usuário pode alterar o parâmetro de arredondamento (”date_rounding”) para definir um período específico de rotação do índice. Para criar outros pipelines, o usuário deve altere a URL (a última parte é o nome do pipeline) e altere campo ”index_name_prefix” para corresponder ao nome do índice.

Veja também Elasticsearch documentação.

Além disso, armazenar dados históricos em vários índices baseados em datas deve também ser habilitado no novo parâmetro na configuração do servidor Zabbix:

```
### Opção: HistoryStorageDateIndex
# Habilite o pré-processamento de valores de histórico no armazenamento de histórico para armazenar valore
# 0 - desabilitar
#1 - habilitar
#
# Obrigatório: não
# Predefinição:
# HistoryStorageDateIndex=0
```

Solução de problemas

As etapas a seguir podem ajudá-lo a solucionar problemas com a configuração do Elasticsearch:

1. Verifique se o mapeamento está correto (solicitação GET para a URL do índice necessário, como `http://localhost:9200/uint`).
2. Verifique se os shards não estão em estado de falha (reiniciar o Elasticsearch pode ajudar).
3. Verifique a configuração do Elasticsearch. A configuração deve permitir o acesso a partir do host do frontend do Zabbix e do host do server do Zabbix.
4. Verifique os logs do Elasticsearch.

5. `LogSlowQueries` pode ser usado para verificar consultas lentas no banco de dados do Elasticsearch.

Se você ainda estiver enfrentando problemas com sua instalação, crie um relatório de bug com todas as informações desta lista (mapeamento, logs de erro, configuração, versão, etc.)

8 Notas específicas de distribuição sobre a configuração do Nginx para o Zabbix

RHEL

No RHEL 8, 9 e 10, o Nginx está incluído no AppStream—não são necessários repositórios extras. Basta habilitar e instalar com:

```
dnf module list nginx
dnf -y install nginx
```

No RHEL 7, o Nginx pode ser instalado a partir do repositório CentOS 7 Extras (se estiver usando CentOS) via:

```
yum -y install nginx
```

Se preferir as versões mais recentes do upstream, configure o repositório oficial do NGINX em `/etc/yum.repos.d/nginx.repo`:

```
cat > /etc/yum.repos.d/nginx.repo << 'EOF'
[nginx-stable]
name=nginx stable repo
baseurl=https://nginx.org/packages/rhel/$releasever/$basearch/
gpgcheck=1
enabled=1
gpgkey=https://nginx.org/keys/nginx_signing.key
EOF
dnf -y install nginx
```

SLES 15

No SUSE Linux Enterprise Server 15, você precisa configurar o `php-fpm` (o caminho para o arquivo de configuração pode variar um pouco dependendo do service pack):

```
cp /etc/php8/fpm/php-fpm.conf{.default,}
cp /etc/php8/fpm/php-fpm.d/www.conf{.default,}
sed -i 's/user = nobody/user = wwwrun/; s/group = nobody/group = www/' /etc/php8/fpm/php-fpm.d/www.conf
```

9 Executando o agent como root

Desde o Zabbix **5.0.0**, o arquivo de serviço systemd para o agent Zabbix nos pacotes oficiais inclui explicitamente diretivas para `User` e `Group`. Ambos são definidos como `zabbix`.

Não é mais possível configurar qual usuário o agent Zabbix será executado através do arquivo `zabbix_agentd.conf`, porque o agent irá ignorar essa configuração e será executado como o usuário especificado no arquivo de serviço systemd. Para executar o agent Zabbix como root, você precisa fazer as modificações descritas abaixo.

Zabbix agent

Para substituir o usuário e grupo padrão do Zabbix agent, execute:

```
systemctl edit zabbix-agent
```

Em seguida, adicione o seguinte conteúdo:

```
[Service]
User=root
Group=root
```

Recarregue os daemons e reinicie o serviço zabbix-agent:

```
systemctl daemon-reload
systemctl restart zabbix-agent
```

Para o **Zabbix agent**, isso reativa a funcionalidade de configurar o usuário no arquivo `zabbix_agentd.conf`. Agora você precisa definir os parâmetros de configuração `User=root` e `AllowRoot=1` no arquivo de configuração do agent.

Zabbix agent 2

Para substituir o usuário e grupo padrão do Zabbix agent 2, execute:

```
systemctl edit zabbix-agent2
```

Em seguida, adicione o seguinte conteúdo:

```
[Service]
User=root
Group=root
```

Recarregue os daemons e reinicie o serviço zabbix-agent2:

```
systemctl daemon-reload
systemctl restart zabbix-agent2
```

Para o **Zabbix agent2**, isso determina completamente o usuário com o qual ele será executado. Nenhuma modificação adicional é necessária.

10 Agent Zabbix no Microsoft Windows

Configurando o agent

Ambas as gerações dos agents Zabbix são executadas como um serviço do Windows. Para o Zabbix agent 2, substitua *agentd* por *agent2* nas instruções abaixo.

Você pode executar uma única instância do agent Zabbix ou várias instâncias do agent em um host Microsoft Windows. Uma única instância pode usar:

- o arquivo de configuração padrão, localizado no mesmo diretório que o binário do agent;
- um arquivo de configuração especificado na linha de comando.

No caso de múltiplas instâncias, cada instância do agent deve ter seu próprio arquivo de configuração (uma das instâncias pode usar o arquivo de configuração padrão).

Um arquivo de configuração de exemplo está disponível no arquivo-fonte do Zabbix como:

- `conf/zabbix_agentd.conf` para o Zabbix agent;
- `conf/zabbix_agent2.conf` para o Zabbix agent2.

Se você deseja instalar o agent/agent 2 do Zabbix para Windows como um serviço a partir de um arquivo compactado sem especificar explicitamente o arquivo de configuração, então, antes de instalar o agent:

- `conf/zabbix_agentd.conf` deve ser copiado manualmente para o diretório onde o zabbix_agentd.exe será instalado;
- `conf/zabbix_agent2.conf` e o diretório `conf/zabbix_agent2.d` devem ser copiados manualmente para o diretório onde o zabbix_agent2.exe será instalado.

Consulte as opções do arquivo de configuração para obter detalhes sobre como configurar o agent Zabbix para Windows.

Parâmetro Hostname

Para executar cheques ativos em um host, o agent Zabbix precisa ter o hostname definido. Além disso, o valor do hostname definido no lado do agent deve corresponder exatamente ao "Nome do host" configurado para o host no frontend.

O valor do hostname no lado do agent pode ser definido pelo parâmetro **Hostname** ou **HostnameItem** no arquivo de configuração do agent - ou os valores padrão são usados se nenhum desses parâmetros for especificado.

O valor padrão para o parâmetro **HostnameItem** é o valor retornado pela chave do agent "system.hostname". Para Windows, retorna o resultado da função gethostname(), que consulta os provedores de namespace para determinar o nome do host local. Se nenhum provedor de namespace responder, o nome NetBIOS é retornado.

O valor padrão para **Hostname** é o valor retornado pelo parâmetro HostnameItem. Portanto, na prática, se ambos os parâmetros não forem especificados, o hostname real será o nome NetBIOS do host; o agent Zabbix usará o nome NetBIOS do host para recuperar a lista de cheques ativos do server Zabbix e enviar os resultados para ele.

A chave "system.hostname" suporta dois parâmetros opcionais - *type* e *transform*.

Type determina o tipo de nome que o item deve retornar:

- *netbios* (padrão) - retorna o nome NetBIOS do host, que é limitado a 15 símbolos e está apenas em MAIÚSCULAS;
- *host* - sensível a maiúsculas e minúsculas, retorna o nome completo e real do host Windows (sem domínio);
- *shorthost* - retorna a parte do hostname antes do primeiro ponto. Retornará a string completa se o nome não contiver um ponto.
- *fqdn* - retorna o Fully Qualified Domain Name (sem o ponto final).

Transform permite especificar uma regra de transformação adicional para o hostname:

- *none* (padrão) - usa o caso original das letras;
- *lower* - converte o texto para minúsculas.

Portanto, para simplificar a configuração do arquivo zabbix_agentd.conf e torná-lo unificado, três abordagens diferentes podem ser usadas:

1. Deixe os parâmetros **Hostname** ou **HostnameItem** indefinidos e o agent Zabbix usará o nome NetBIOS do host como hostname.
2. Deixe o parâmetro **Hostname** indefinido e defina **HostnameItem** assim:
 HostnameItem=system.hostname[host] - para que o agent Zabbix use o nome completo e real (sensível a maiúsculas e minúsculas) do host Windows como hostname.
 HostnameItem=system.hostname[shorthost,lower] - para que o agent Zabbix use apenas a parte do hostname antes do primeiro ponto, convertida em minúsculas.
 HostnameItem=system.hostname[fqdn] - para que o agent Zabbix use o Fully Qualified Domain Name como hostname.

O nome do host também é usado como parte do nome do serviço do Windows, que é usado para instalar, iniciar, parar e desinstalar o serviço do Windows. Por exemplo, se o arquivo de configuração do agent Zabbix especificar `Hostname=Windows_db_server`, então o agent será instalado como um serviço do Windows "`Zabbix Agent [Windows_db_server]`". Portanto, para ter um nome de serviço do Windows diferente para cada instância do agent Zabbix, cada instância deve usar um nome de host diferente.

Instalando o agent como serviço do Windows

Antes de instalar o agent, copie manualmente o arquivo `conf/zabbix_agentd.conf` para o diretório onde o zabbix_agentd.exe será instalado.

Para instalar uma única instância do agent Zabbix com o arquivo de configuração padrão:

```
zabbix_agentd.exe --install
```

> **Attention:**
> Em um sistema de 64 bits, é necessária uma versão de agent Zabbix de 64 bits para que todas as verificações relacionadas à execução de processos de 64 bits funcionem corretamente.

Se você deseja usar um arquivo de configuração diferente do padrão, deve usar o seguinte comando para instalar o serviço:

```
zabbix_agentd.exe --config <seu_arquivo_de_configuração> --install
```

Um caminho completo para o arquivo de configuração deve ser especificado.

Múltiplas instâncias do agent Zabbix podem ser instaladas como serviços assim:

```
zabbix_agentd.exe --config <arquivo_de_configuração_para_instância_1> --install --multiple-agents
zabbix_agentd.exe --config <arquivo_de_configuração_para_instância_2> --install --multiple-agents
...
zabbix_agentd.exe --config <arquivo_de_configuração_para_instância_N> --install --multiple-agents
```

O serviço instalado agora deve estar visível no Painel de Controle.

Iniciando o agente

Para iniciar o serviço do agente você pode usar o painel de controle ou faze-lo a partir da linha de comando.

Iniciar uma instância simples do Zabbix Agent com arquivo padrão de configuração:

zabbix_agentd.exe --start

Iniciar uma instância simples do Zabbix Agent com arquivo de configuração específico:

zabbix_agentd.exe --config <your_configuration_file> --start

Iniciar múltiplas instâncias do Zabbix Agent com arquivo de configuração específicos:

zabbix_agentd.exe --config <configuration_file_for_this_instance> --start --multiple-agents

Parando o agente

Para parar o serviço do agente você pode usar o painel de controle ou faze-lo a partir da linha de comando.

Parar uma instância simples do Zabbix Agent com arquivo padrão de configuração:

zabbix_agentd.exe --stop

Parar uma instância simples do Zabbix Agent com arquivo de configuração específico:

zabbix_agentd.exe --config <your_configuration_file> --stop

Parar múltiplas instâncias do Zabbix Agent com arquivo de configuração específicos:

zabbix_agentd.exe --config <configuration_file_for_this_instance> --stop --multiple-agents

Desinstalar o serviço do agente

Para desinstalar uma instância simples do Zabbix Agent com arquivo padrão de configuração:

zabbix_agentd.exe --uninstall

Para desinstalar uma instância simples do Zabbix Agent com arquivo de configuração específico:

zabbix_agentd.exe --config <your_configuration_file> --uninstall

Para desinstalar múltiplas instâncias do Zabbix Agent com arquivo de configuração específicos:

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

Limitações

O agent Zabbix para Windows não suporta configurações não padrão do Windows onde as CPUs são distribuídas de forma não uniforme entre os nós NUMA. Se as CPUs lógicas forem distribuídas de forma não uniforme, as métricas de desempenho da CPU podem não estar disponíveis para algumas CPUs. Por exemplo, se houver 72 CPUs lógicas com 2 nós NUMA, ambos os nós devem ter 36 CPUs cada.

11 Configuração do SAML com o Microsoft Entra ID

Visão geral

Esta seção fornece diretrizes para configurar o login único e o provisionamento de usuários no Zabbix a partir do Microsoft Entra ID (anteriormente Microsoft Azure Active Directory) usando autenticação SAML 2.0.

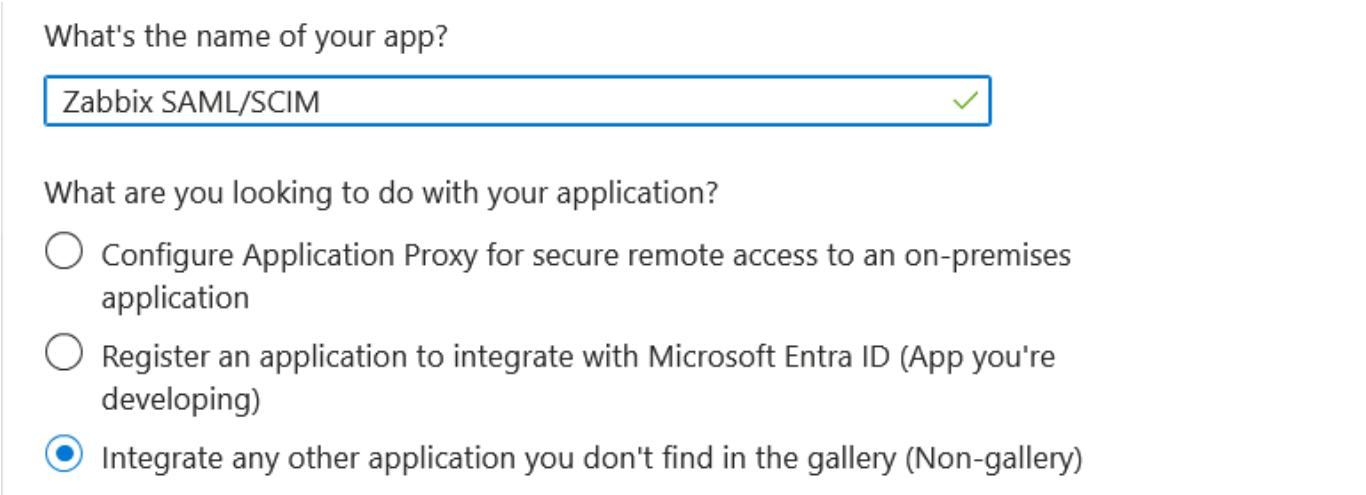
Configuração do Microsoft Entra ID

Criando uma aplicação

1. Faça login no Microsoft Entra admin center em Microsoft Entra ID. Para fins de teste, você pode criar uma conta gratuita de avaliação no Microsoft Entra ID.

2. No Microsoft Entra admin center selecione *Aplicativos -> Aplicativos corporativos -> Novo aplicativo -> Criar seu próprio aplicativo*.

3. Adicione o nome do seu aplicativo e selecione a opção *Integrar qualquer outro aplicativo...*. Depois disso, clique em *Criar*.

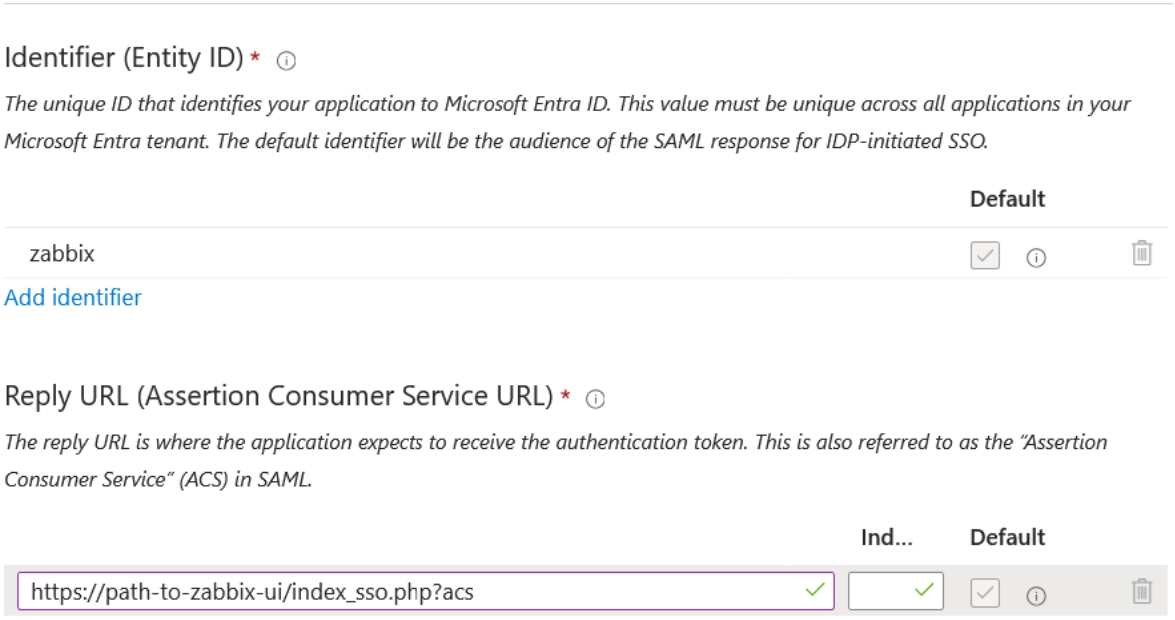


Configurando o single sign-on

1. Na página do seu aplicativo, vá para *Configurar single sign-on* e clique em *Começar*. Em seguida, selecione *SAML*.

2. Edite a *Configuração básica do SAML*:

- Em *Identificador (ID da entidade)* defina um nome exclusivo para identificar seu aplicativo no Microsoft Entra ID, por exemplo, `zabbix`;
- Em *URL de resposta (URL do serviço de consumidor de asserção)* defina o endpoint de single sign-on do Zabbix: `https://<caminho-para-o-zabbix-ui>/index_sso.php?acs`:



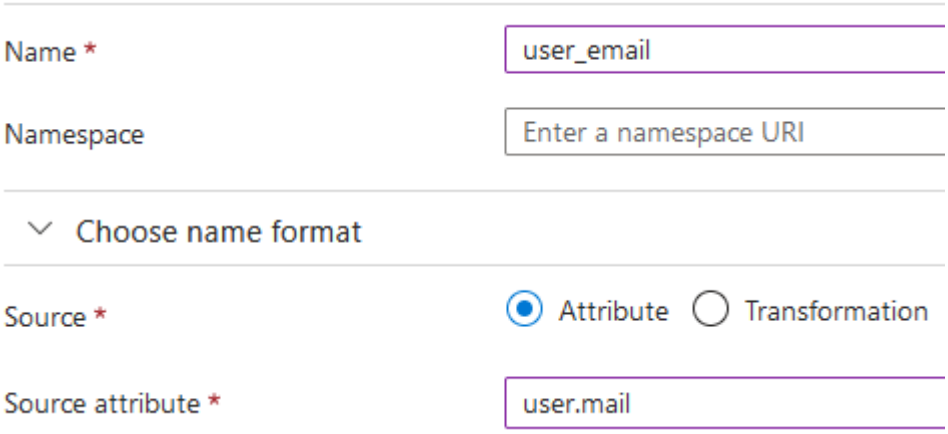
Observe que "https" é obrigatório. Para que isso funcione com o Zabbix, é necessário adicionar ao arquivo `conf/zabbix.conf.php` a seguinte linha:

```
$SSO['SETTINGS'] = ['use_proxy_headers' => true];
```

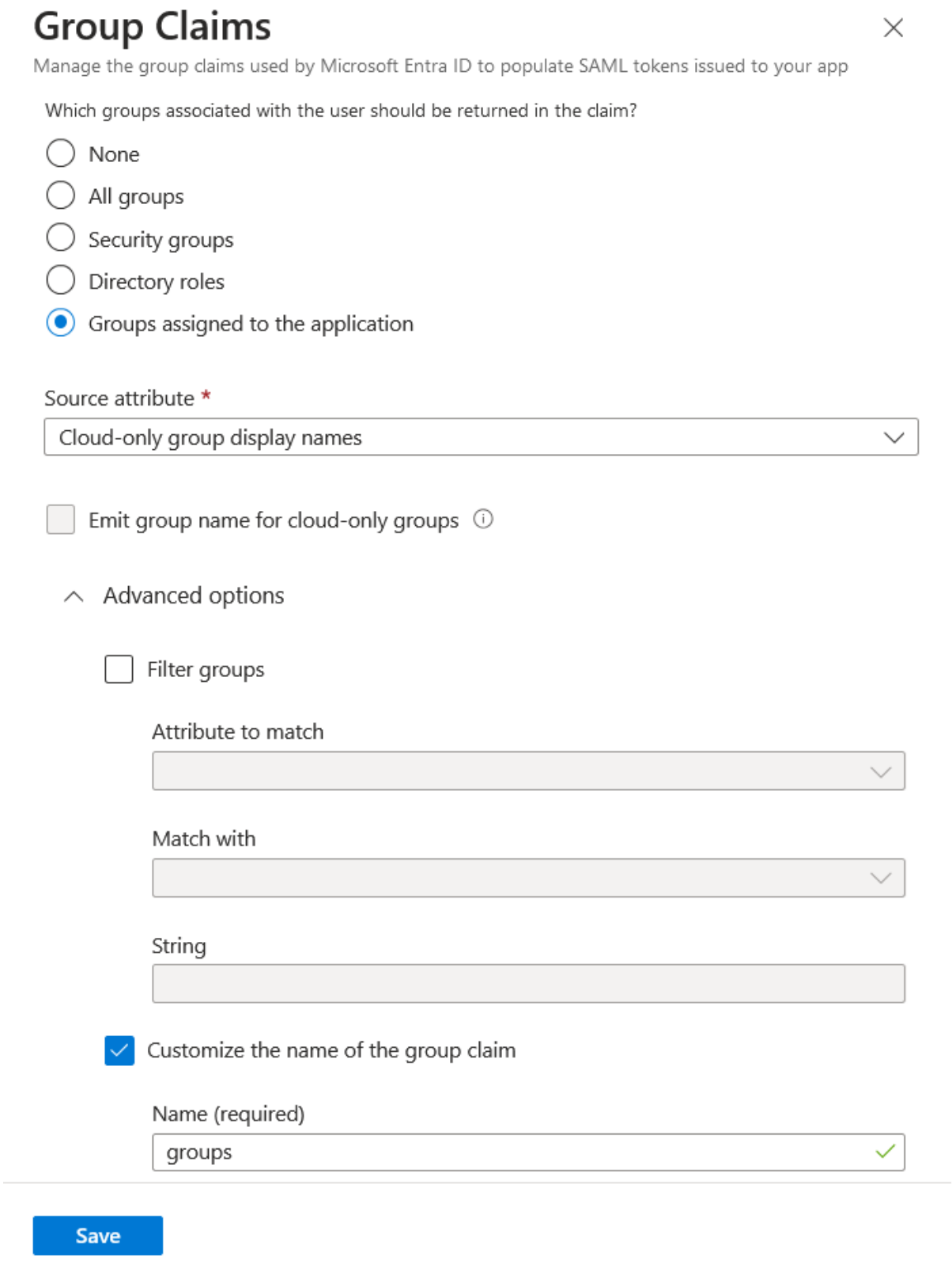
3. Edite *Atributos e declarações*. Você deve adicionar todos os atributos que deseja passar para o Zabbix (user_name, user_lastname, user_email, user_mobile, groups).

Os nomes dos atributos são arbitrários. Podem ser usados nomes de atributos diferentes, no entanto, é necessário que correspondam ao valor do campo respectivo nas configurações SAML do Zabbix.

- Clique em *Adicionar nova declaração* para adicionar um atributo:



- Clique em *Adicionar uma declaração de grupo* para adicionar um atributo para passar grupos para o Zabbix:



É importante nesta declaração que os nomes dos grupos (em vez dos IDs dos grupos) sejam passados para o Zabbix pelo *Atributo de origem* selecionado. Caso contrário, o provisionamento de usuários JIT não funcionará corretamente.

4. Em *Certificados SAML* faça o download do certificado Base64 fornecido pelo Entra ID e coloque-o em `conf/certs` da instalação do frontend do Zabbix.

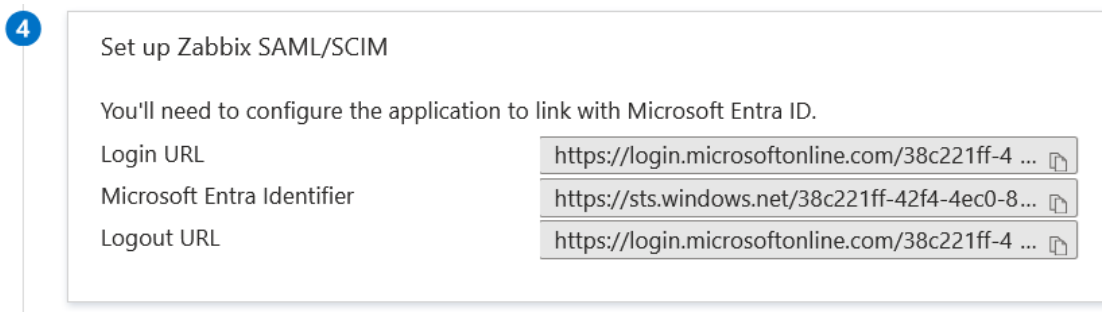
Defina as permissões 644 executando:

```
chmod 644 entra.cer
```

Certifique-se de que o arquivo `conf/zabbix.conf.php` contenha a linha:

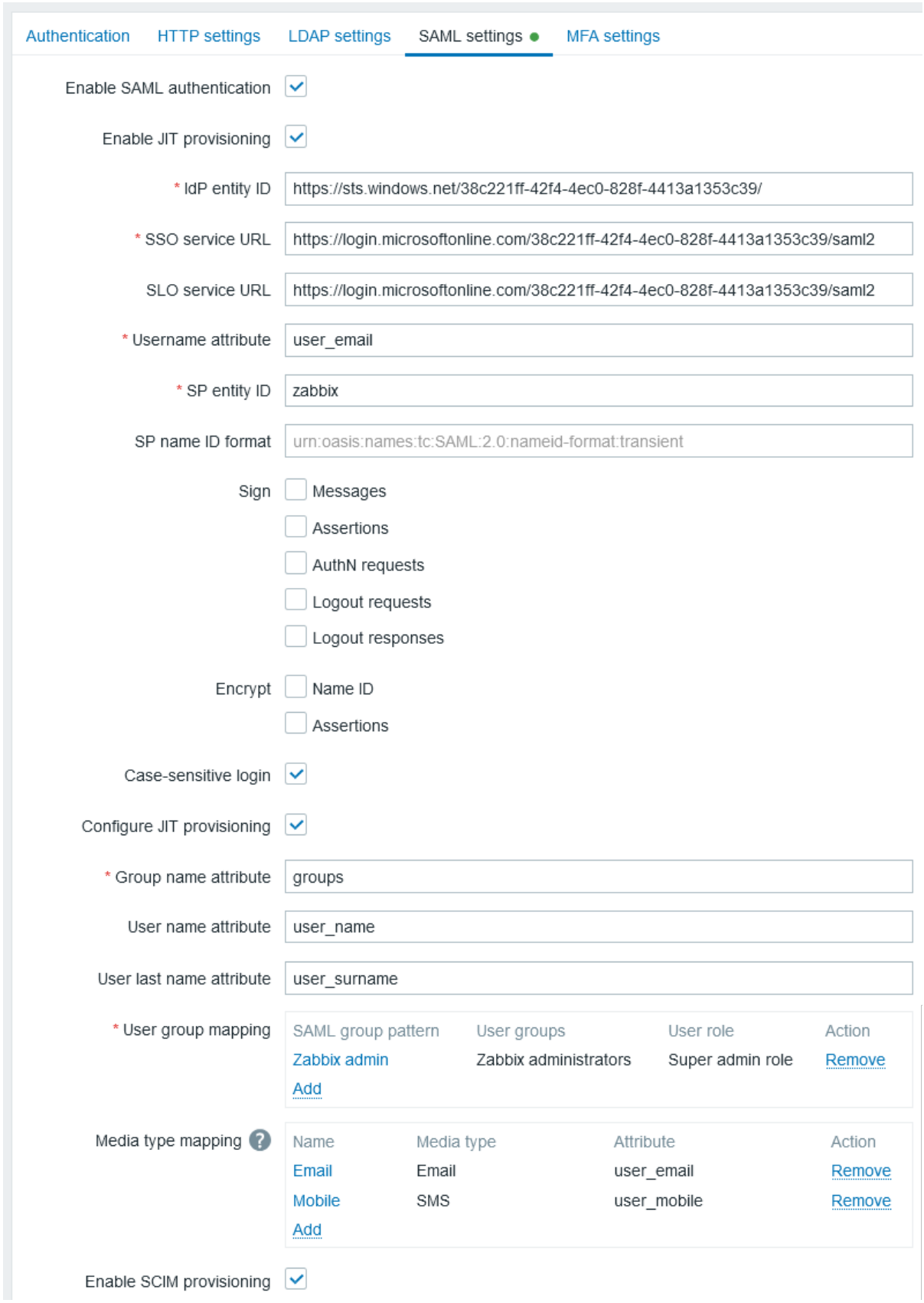
```
$SSO['IDP_CERT'] = 'conf/certs/entra.cer';
```

5. Use os valores de *Configurar <nome do seu aplicativo>* no Entra ID para configurar a autenticação SAML do Zabbix (veja a próxima seção):



Configuração do Zabbix

1. No Zabbix, vá para as configurações SAML e preencha as opções de configuração com base na configuração do Entra ID:



Campo do Zabbix	Campo de configuração no Entra ID	Valor de exemplo
IdP entity ID	Identificador do Microsoft Entra	
SSO service URL	URL de login	
SLO service URL	URL de logout	
SP entity ID	Identificador (Entity ID)	
Username attribute	Atributo personalizado (claim)	`user_email`
Group name attribute	Atributo personalizado (claim)	`groups`
User name attribute	Atributo personalizado (claim)	`user_name`

Campo do Zabbix	Campo de configuração no Entra ID	Valor de exemplo
User last name attribute	Atributo personalizado (claim)	`user_lastname`

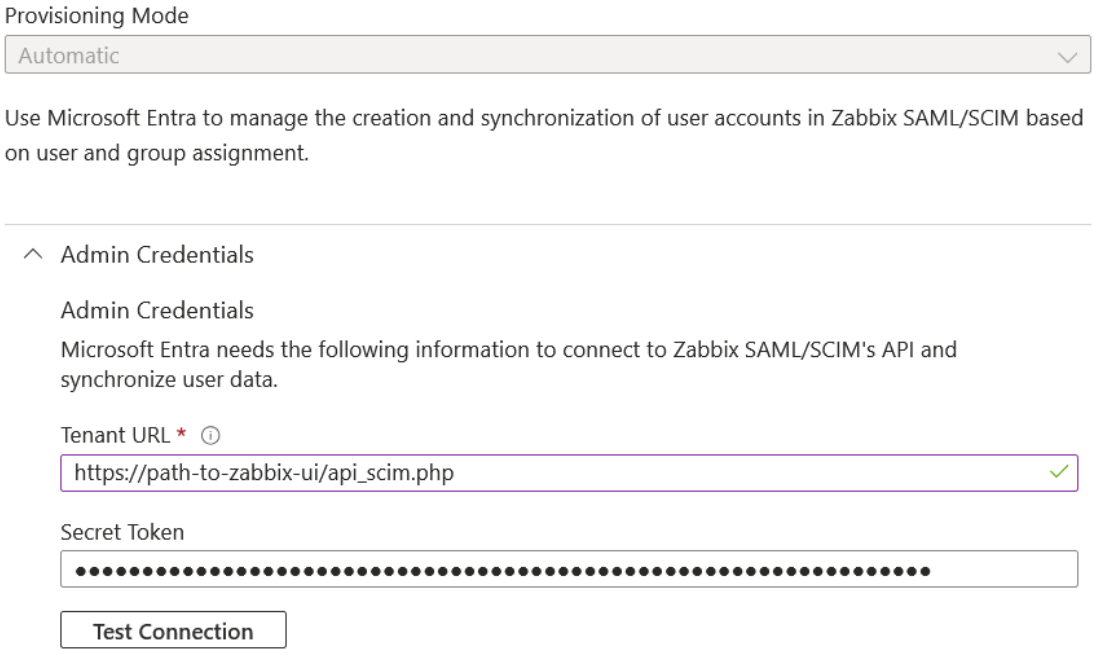
Também é necessário configurar o mapeamento de grupos de usuários. O mapeamento de mídia é opcional.

Clique em *Atualizar* para salvar essas configurações.

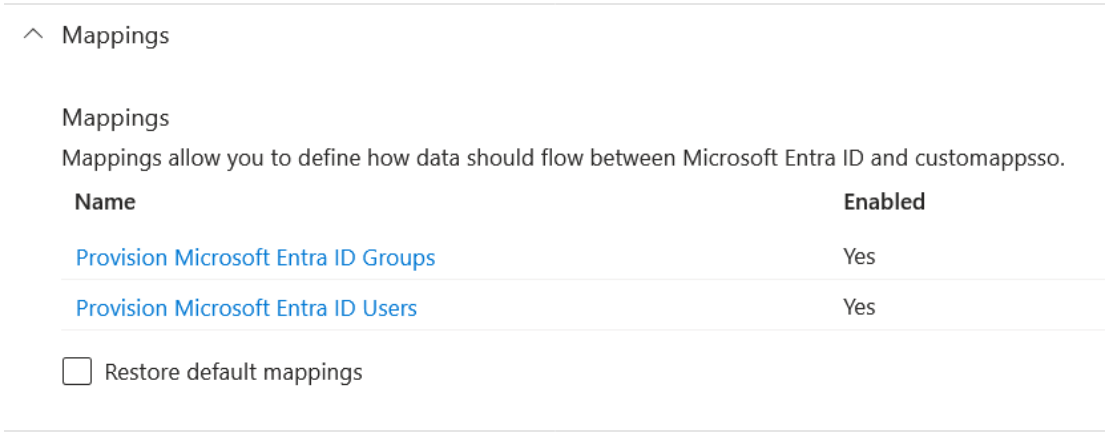
Provisionamento de usuários SCIM

1. Na página do seu aplicativo Entra ID, no menu principal, abra a página Provisionamento. Clique em *Introdução* e, em seguida, selecione o modo de provisionamento Automático:

- Em *URL do locatário*, defina o seguinte valor: `https://<path-to-zabbix-ui>/api_scim.php`
- Em *Token secreto*, insira um token da API do Zabbix com permissões de Super admin.
- Clique em *Testar conexão* para verificar se a conexão foi estabelecida.

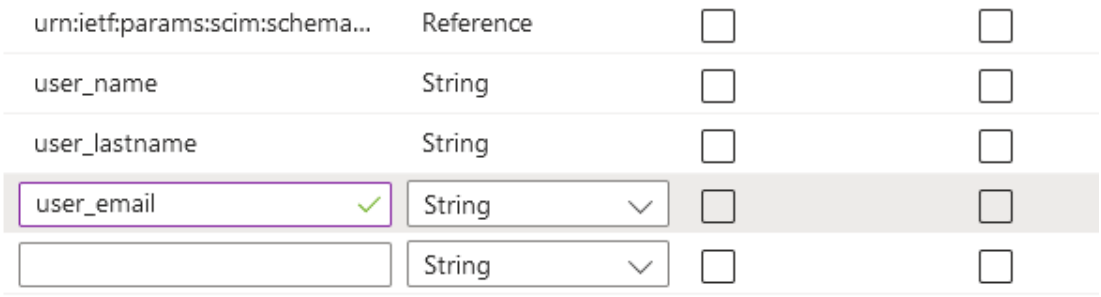


2. Agora você pode adicionar todos os atributos que serão passados com o SCIM para o Zabbix. Para isso, clique em *Mapeamentos* e depois em *Provisionar usuários do Microsoft Entra ID*.



Na parte inferior da lista de Mapeamento de Atributos, habilite *Mostrar opções avançadas* e, em seguida, clique em *Editar lista de atributos para customappsso*.

Na parte inferior da lista de atributos, adicione seus próprios atributos com o tipo 'String':



Salve a lista.

3. Agora você pode adicionar mapeamentos para os atributos adicionados. Na parte inferior da lista de Mapeamento de Atributos, clique em *Adicionar novo mapeamento* e crie os mapeamentos conforme mostrado abaixo:

Quando todos os mapeamentos forem adicionados, salve a lista de mapeamentos.

4. Como pré-requisito para o provisionamento de usuários no Zabbix, você deve ter usuários e grupos configurados no Entra ID.

Para isso, acesse o *Centro de administração do Microsoft Entra* e adicione usuários/grupos nas respectivas páginas de Usuários e Grupos.

5. Quando os usuários e grupos forem criados no Entra AD, você pode ir ao menu *Usuários e grupos* do seu aplicativo e adicioná-los ao aplicativo.

6. Vá para o menu *Provisionamento* do seu aplicativo e clique em *Iniciar provisionamento* para que os usuários sejam provisionados no Zabbix.

Observe que a solicitação PATCH de Usuários no Entra ID não suporta alterações em mídias.

Assinatura de solicitação de autenticação

É possível configurar o Entra ID para validar a assinatura de solicitações de autenticação assinadas.

Para que isso funcione, crie chaves pública/privada:

```
openssl req -x509 -newkey rsa:4096 -keyout /usr/share/zabbix/conf/certs/request-sign.key -out /usr/share/z
```

Atribua permissões:

```
chown apache /usr/share/zabbix/conf/certs/request-sign.key
chmod 400 /usr/share/zabbix/conf/certs/request-sign.key
```

Atualize a configuração do frontend do Zabbix adicionando:

```
$SSO['SP_KEY'] = 'conf/certs/request-sign.key';
$SSO['SP_CERT'] = 'conf/certs/request-sign.crt';
```

Solução de problemas

Problemas de autenticação podem ocorrer com navegadores MicroSoft Edge quando um usuário, tentando fazer login no Zabbix via SAML, já está logado com o perfil do MicroSoft Edge. Como sinal desse problema, o usuário pode conseguir fazer login no Zabbix usando o MicroSoft Edge no modo privado.

Para evitar problemas de autenticação nesse caso, pode ser necessário definir `requestedAuthnContext` como ”false” no arquivo de configuração do frontend do Zabbix (*zabbix.conf.php*).

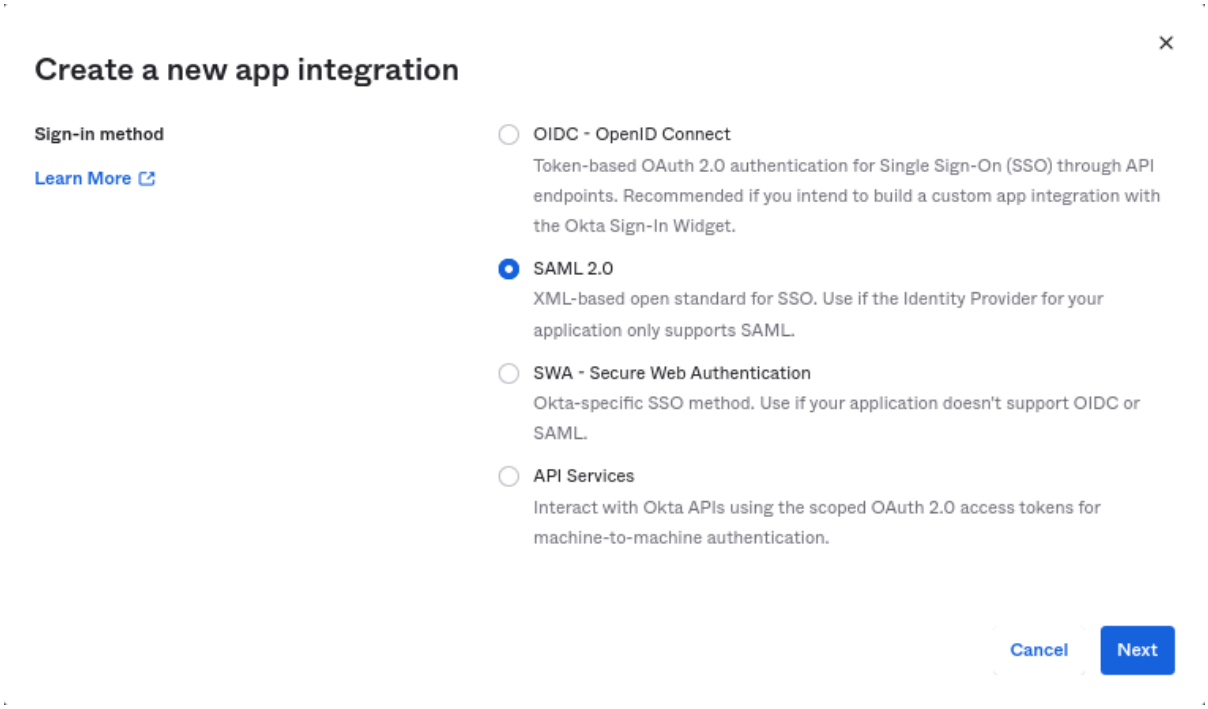
```
$SSO['SETTINGS'] = [
    'security' => [
        'requestedAuthnContext' => false
    ]
];
```

12 Configuração do SAML com Okta

Esta seção fornece diretrizes para configurar o Okta para habilitar a autenticação SAML 2.0 e o provisionamento de usuários para o Zabbix.

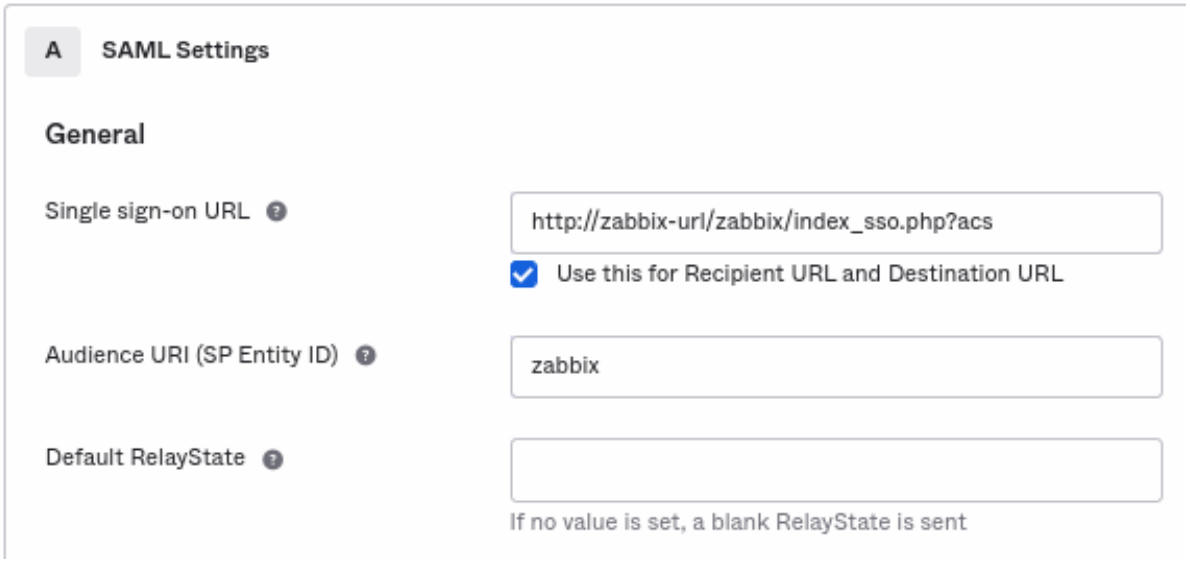
Configuração do Okta

1. Acesse https://developer.okta.com/signup/ e registre-se/entre em sua conta.
2. Na interface web do Okta, navegue até *Applications* → *Applications*.
3. Clique em *Create App Integration*.

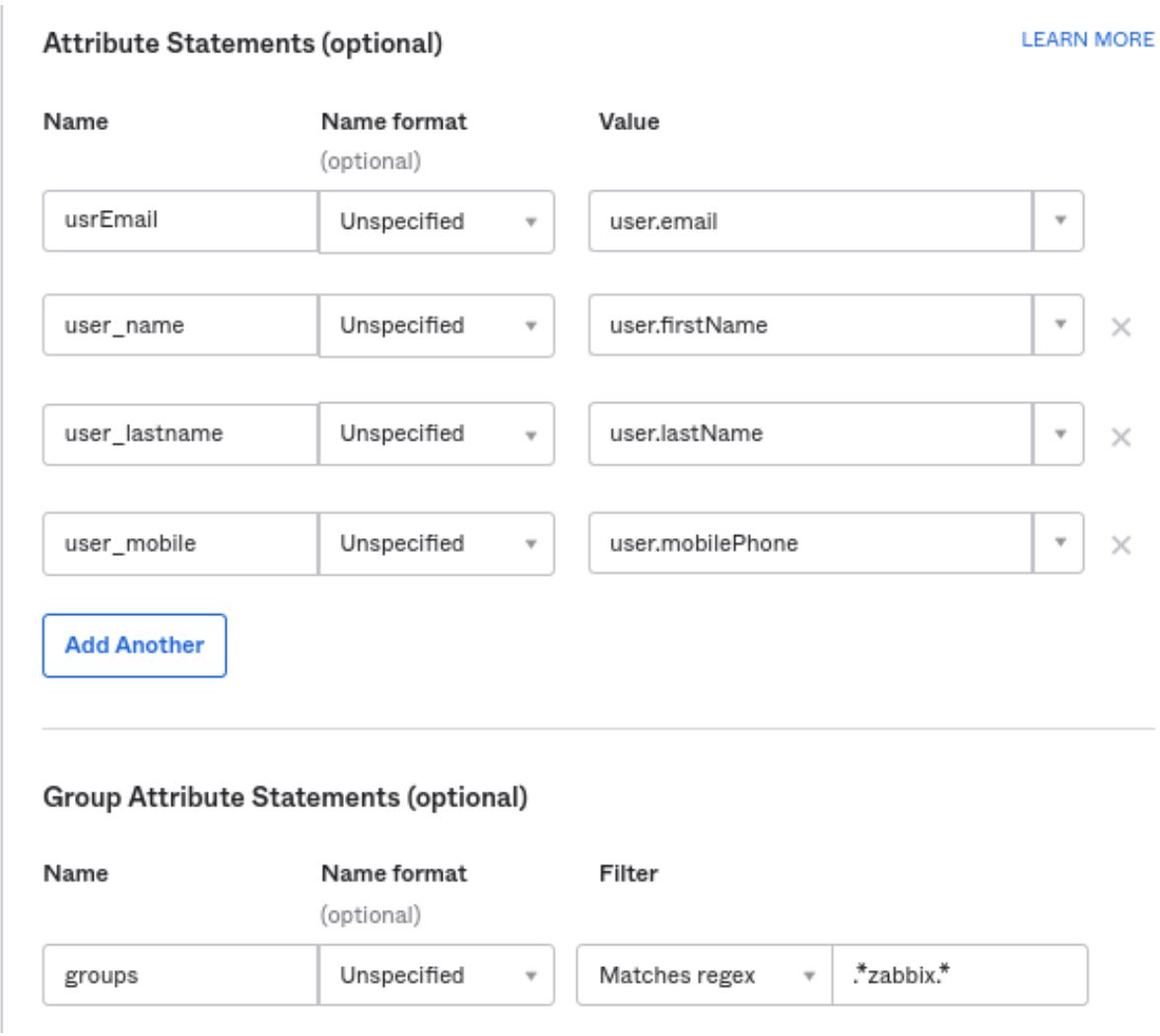


Selecione ”SAML 2.0” como método de login e clique em *Next*.

4. Em configurações gerais, preencha o nome do aplicativo e clique em *Next*.
5. Na configuração SAML, insira os valores fornecidos abaixo e clique em *Next*.



- Em **General** adicione:
  - *Single sign-on URL*: `http://<your-zabbix-url>/zabbix/index_sso.php?acs<br>` Observe o uso de "http", e não "https", para que o parâmetro `acs` não seja cortado na requisição. A caixa *Use this for Recipient URL and Destination URL* também deve ser marcada.
  - *Audience URI (SP Entity ID)*: `zabbix<br>` Observe que este valor será usado dentro da asserção SAML como um identificador exclusivo do provedor de serviço (se não corresponder, a operação será rejeitada). É possível especificar uma URL ou qualquer string de dados neste campo.
  - *Default RelayState*:
 Deixe este campo em branco; se um redirecionamento personalizado for necessário, ele pode ser adicionado no Zabbix em *Users* → *Users*.
  - Preencha outros campos de acordo com suas preferências.
- Em **Attribute Statements/Group Attribute Statements** adicione:



Essas declarações de atributos são inseridas nas asserções SAML compartilhadas com o Zabbix.

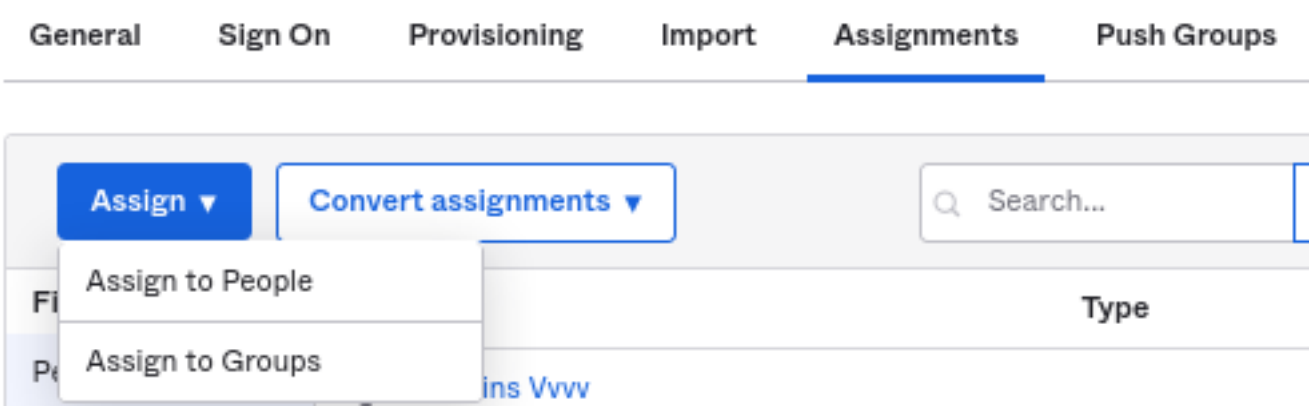
Os nomes dos atributos usados aqui são exemplos arbitrários. Você pode usar nomes de atributos diferentes, no entanto, é necessário que eles correspondam ao valor do campo respectivo nas configurações SAML do Zabbix.

Se você deseja configurar o login SAML no Zabbix *sem* provisionamento de usuário JIT, apenas o atributo de e-mail é necessário.

Note:
Se planeja usar uma conexão criptografada, gere os certificados de criptografia privados e públicos e faça o upload do certificado público para o Okta. O formulário de upload do certificado aparece quando *Assertion Encryption* está definido como "Encrypted" (clique em *Show Advanced Settings* para encontrar este parâmetro).

6. Na próxima aba, selecione "I'm a software vendor. I'd like to integrate my app with Okta" e pressione "Finish".

7. Navegue até a aba "Assignments" do aplicativo recém-criado e clique no botão *Assign*, depois selecione "Assign to People" no menu suspenso.



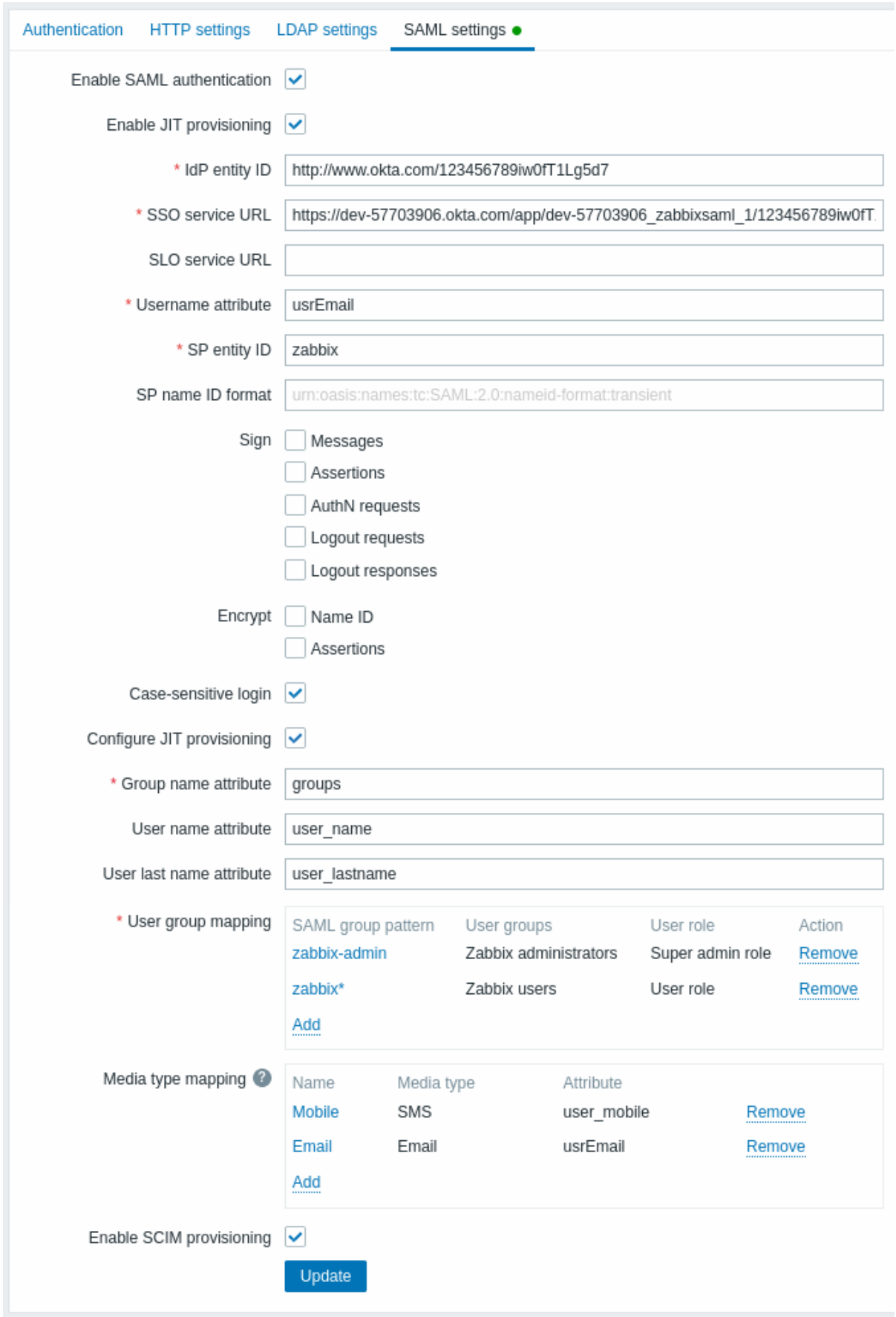
8. Em um pop-up que aparecer, atribua o aplicativo às pessoas que usarão o SAML 2.0 para autenticar com o Zabbix e clique em *Save and go back*.

9. Navegue até a aba "Sign On" e clique no botão *View Setup Instructions*.

As **instruções** de configuração serão abertas em uma nova aba; mantenha esta aba aberta enquanto configura o Zabbix.

Configuração do Zabbix

1. No Zabbix, acesse as configurações SAML e preencha as opções de configuração com base nas instruções de configuração do Okta:



Campo do Zabbix	Campo de configuração no Okta	Valor de exemplo
IdP entity ID	Identity Provider Issuer	
SSO service URL	Identity Provider Single Sign-On URL	
Username attribute	Attribute name	`usrEmail`
SP entity ID	Audience URI	`zabbix`
Group name attribute	Attribute name	`groups`

Campo do Zabbix	Campo de configuração no Okta	Valor de exemplo
User name attribute	Attribute name	`user_name`
User last name attribute	Attribute name	`user_lastname`

Também é necessário configurar o mapeamento de grupo de usuários e de mídia.

2. Baixe o certificado fornecido nas instruções de configuração SAML do Okta na pasta *ui/conf/certs* como idp.crt.

Defina as permissões 644 para ele executando:

```
chmod 644 idp.crt
```

3. Se *Assertion Encryption* foi definido como "Encrypted" no Okta, a caixa de seleção "Assertions" do parâmetro *Encrypt* também deve ser marcada no Zabbix.

4. Pressione o botão "Atualizar" para salvar essas configurações.

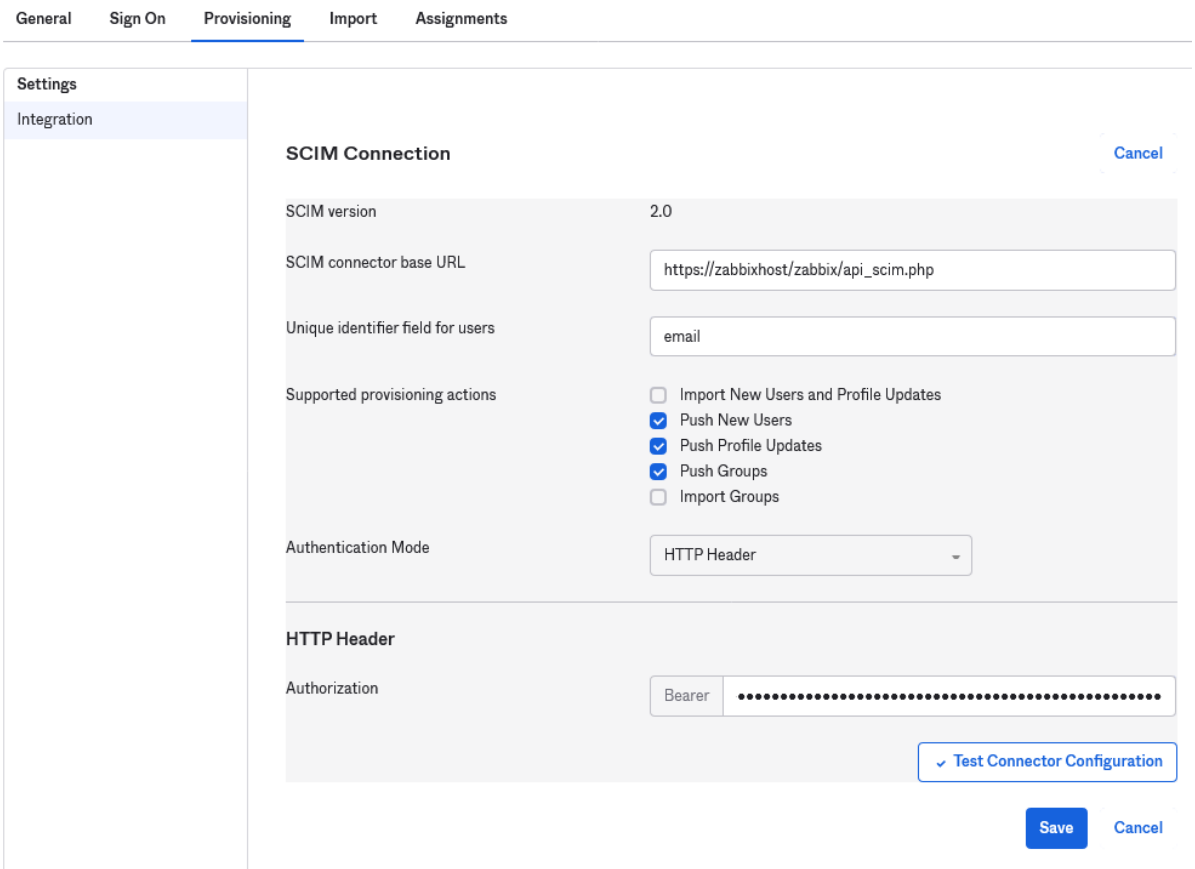
Provisionamento SCIM

1. Para ativar o provisionamento SCIM, vá para "General" -> "App Settings" do aplicativo no Okta.

Marque a caixa *Enable SCIM provisioning*. Como resultado, uma nova aba *Provisioning* aparecerá.

2. Vá para a aba "Provisioning" para configurar uma conexão SCIM:

- Em *SCIM connector base URL* especifique o caminho para o frontend do Zabbix e adicione `api_scim.php` ao final, ou seja:
 `https://<your-zabbix-url>/zabbix/api_scim.php`
- *Unique identifier field for users*: `email`
- *Authentication mode*: `HTTP header`
- Em *Authorization* insira um token de API válido com direitos de Super admin



Attention:
Se você estiver enfrentando problemas de autenticação, veja Encaminhamento do cabeçalho de autorização.

3. Clique em *Test Connector Configuration* para testar a conexão. Se tudo estiver correto, uma mensagem de sucesso será exibida.

4. Em "Provisioning" -> "To App", certifique-se de marcar as seguintes caixas:

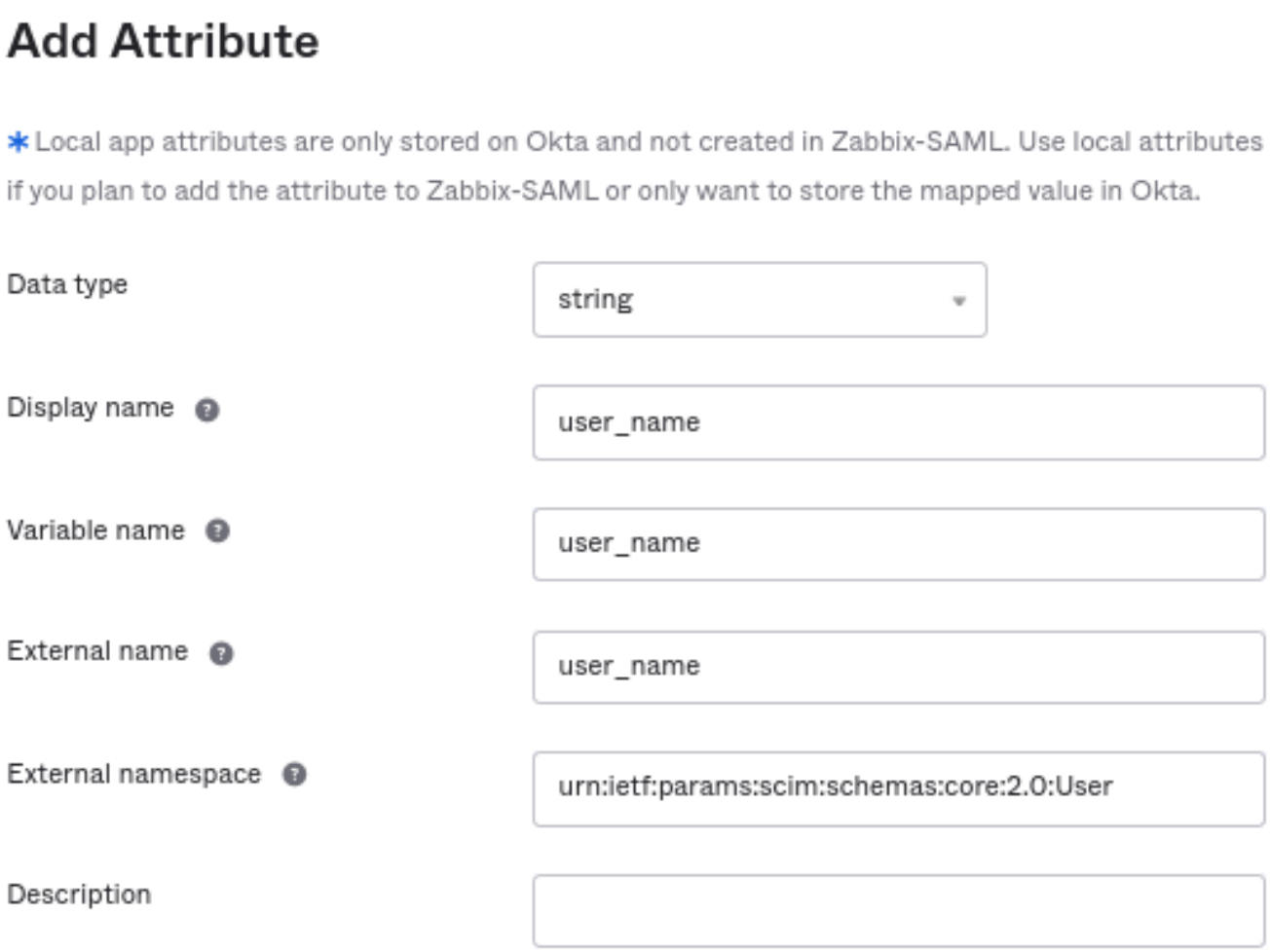
- Create Users

- Update User Attributes
- Deactivate Users

Isso garantirá que esses tipos de requisições sejam enviados para o Zabbix.

5. Certifique-se de que todos os atributos definidos no SAML estejam definidos no SCIM. Você pode acessar o editor de perfil do seu aplicativo em ”Provisioning” -> ”To App”, clicando em *Go to Profile Editor*.

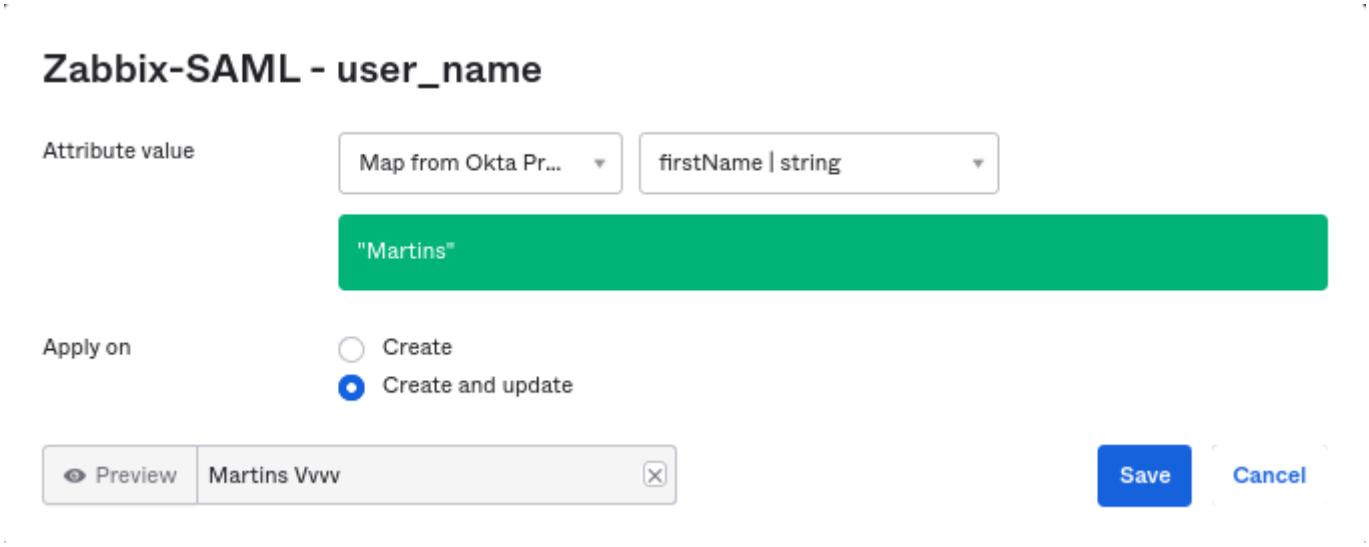
Clique em *Add Attribute*. Preencha os valores para *Display name*, *Variable name*, *External name* com o nome do atributo SAML, por exemplo, `user_name`.



External namespace deve ser o mesmo do esquema de usuário: `urn:ietf:params:scim:schemas:core:2.0:User`

6. Vá para ”Provisioning” -> ”To App” -> ”Attribute Mappings” do seu aplicativo. Clique em *Show Unmapped Attributes* na parte inferior. Os atributos recém-adicionados aparecerão.

7. Mapeie cada atributo adicionado.



8. Adicione usuários na aba ”Assignments”. Os usuários precisam ser previamente adicionados em *Directory -> People*. Todas essas atribuições serão enviadas como requisições para o Zabbix.

9. Adicione grupos na aba ”Push Groups”. O padrão de mapeamento de grupos de usuários nas configurações SAML do Zabbix deve corresponder a um grupo especificado aqui. Se não houver correspondência, o usuário não poderá ser criado no Zabbix.

As informações sobre os membros do grupo são enviadas toda vez que alguma alteração é feita.

13 Configuração do SAML com OneLogin

Visão geral

Esta seção fornece diretrizes para configurar o login único e o provisionamento de usuários no Zabbix a partir do OneLogin usando autenticação SAML 2.0.

Configuração do OneLogin

Criando o aplicativo

1. Faça login na sua conta no OneLogin. Para fins de teste, você pode criar uma conta de desenvolvedor gratuita no OneLogin.

2. Na interface web do OneLogin, navegue até *Applications* → *Applications*.

3. Clique em ”Add App” e procure pelo aplicativo apropriado. As diretrizes nesta página são baseadas no exemplo do aplicativo *SCIM Provisioner with SAML (SCIM v2 Enterprise, full SAML)*.

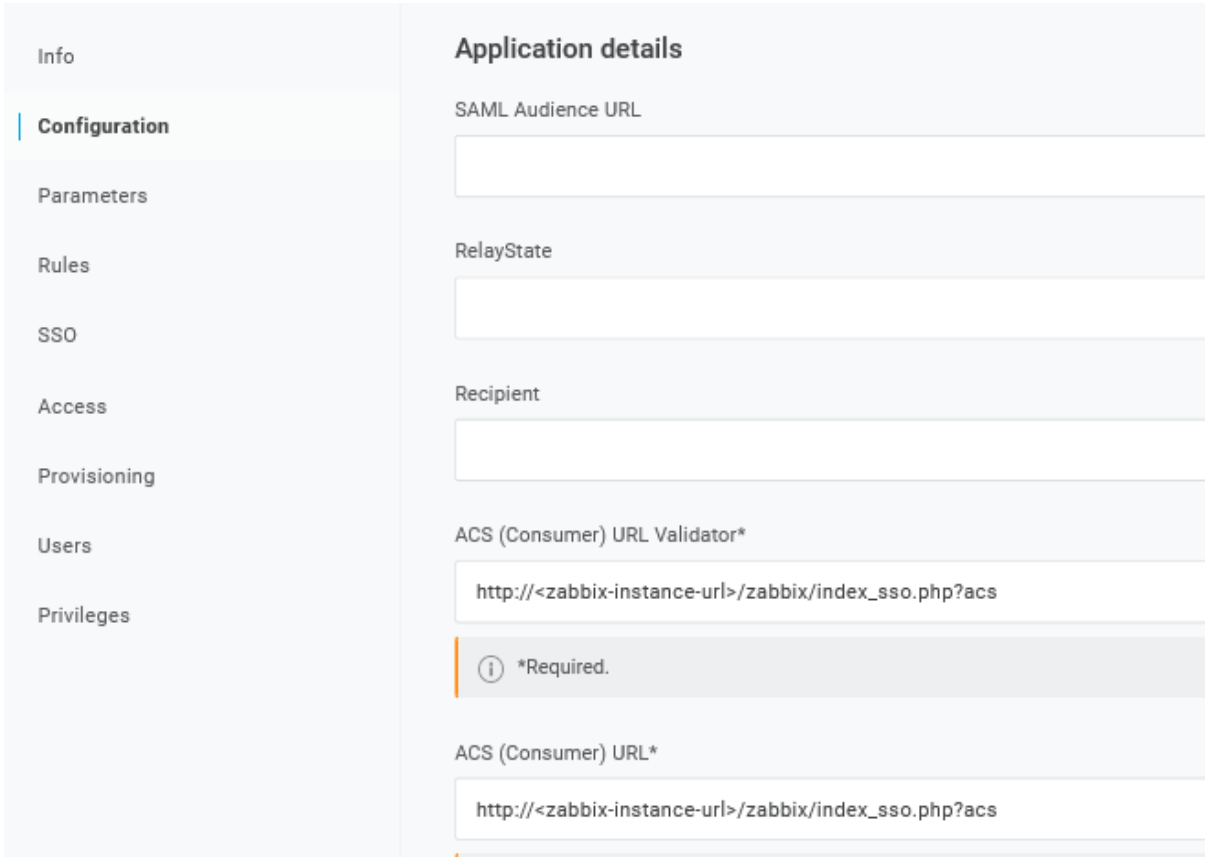
4. Para começar, você pode querer personalizar o nome de exibição do seu aplicativo. Você também pode adicionar o ícone e os detalhes do aplicativo. Depois disso, clique em *Save*.

Configurando o provisionamento SSO/SCIM

1. Em *Configuration -> Application details*, defina o endpoint de single sign-on do Zabbix `http://<zabbix-instance-url>/zabbix/inde` como valor dos seguintes campos:

- *ACS (Consumer) URL Validator*
- *ACS (Consumer) URL*

Observe o uso de ”http”, e não ”https”, para que o parâmetro `acs` não seja cortado na requisição.



Também é possível usar "https". Para que isso funcione com o Zabbix, é necessário adicionar ao arquivo `conf/zabbix.conf.php` a seguinte linha:

```
$SSO['SETTINGS'] = ['use_proxy_headers' => true];
```

Deixe as outras opções com seus valores padrão.

2. Em *Configuration -> API connection*, defina os seguintes valores:

- *SCIM Base URL*: `https://<zabbix-instance-url>/zabbix/api_scim.php`
- *SCIM JSON Template*: deve conter todos os atributos personalizados que você deseja passar para o Zabbix via SCIM, como `user_name`, `user_lastname`, `user_email` e `user_mobile`:

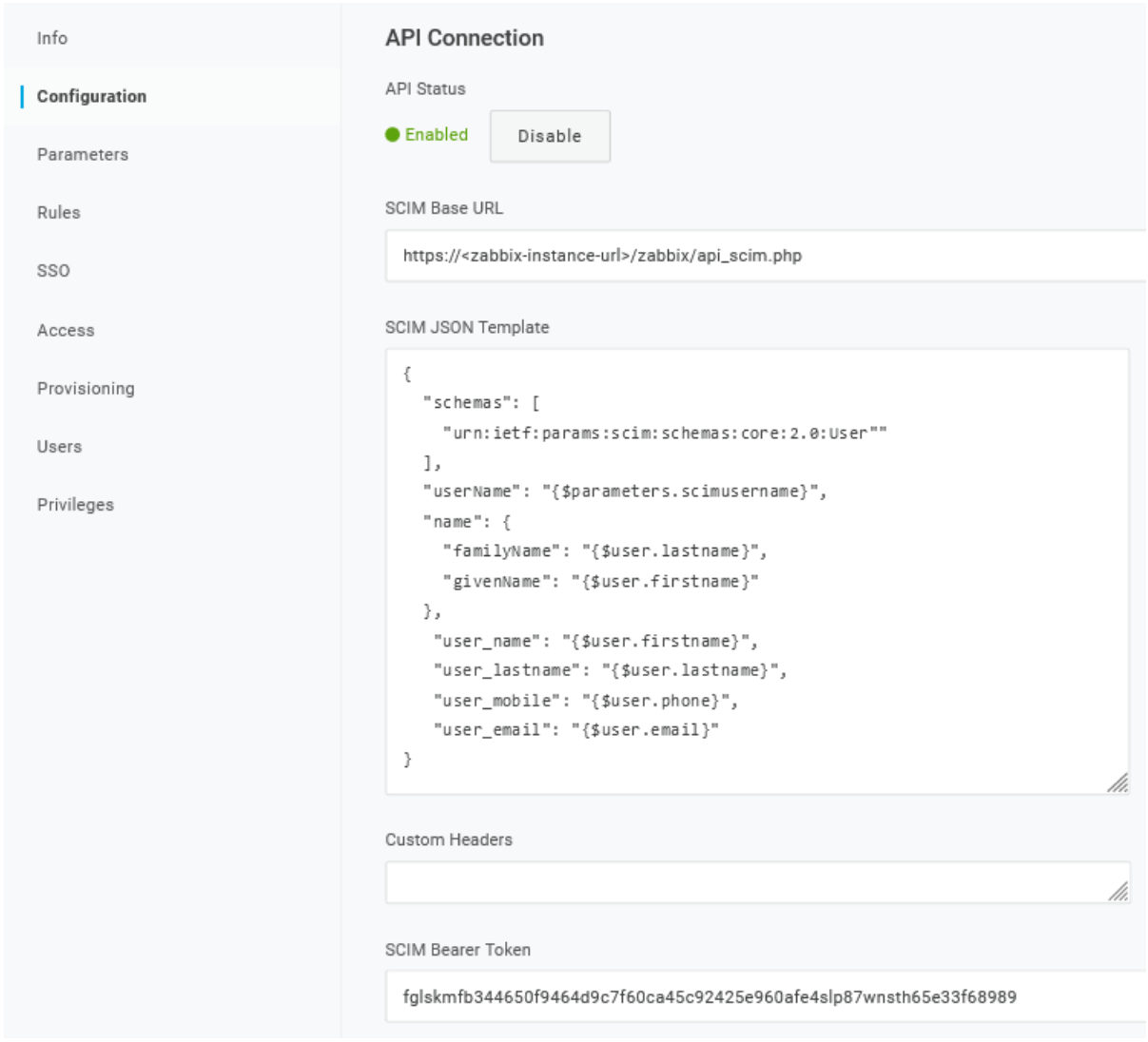
```
{
  "schemas": [
    "urn:ietf:params:scim:schemas:core:2.0:User"
  ],
  "userName": "{$parameters.scimusername}",
  "name": {
    "familyName": "{$user.lastname}",
    "givenName": "{$user.firstname}"
  },
   "user_name": "{$user.firstname}",
   "user_lastname": "{$user.lastname}",
   "user_mobile": "{$user.phone}",
   "user_email": "{$user.email}"
}
```

Os nomes dos atributos são arbitrários. Podem ser usados nomes de atributos diferentes, porém, é necessário que correspondam ao respectivo valor do campo nas configurações SAML do Zabbix.

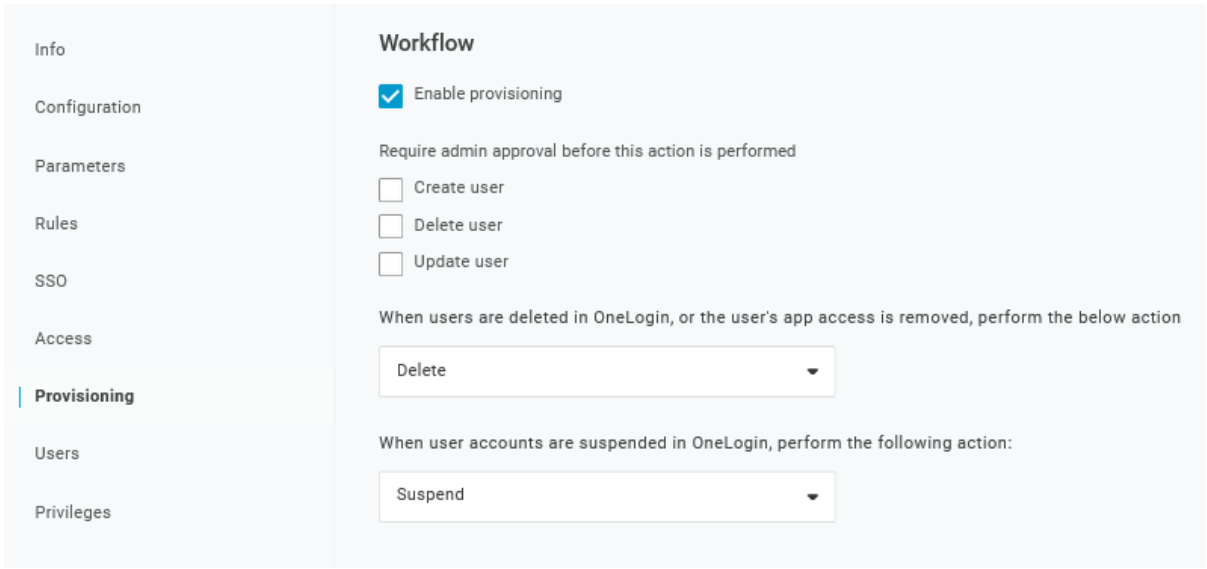
Observe que, para que o provisionamento de usuários funcione, o OneLogin precisa receber em resposta um atributo 'name' com 'givenName' e 'familyName', mesmo que não seja exigido pelo provedor de serviço. Portanto, é necessário especificar isso no schema na parte de configuração do aplicativo.

- *SCIM Bearer Token*: insira um token de API do Zabbix com permissões de Super admin.

Clique em *Enable* para ativar a conexão.

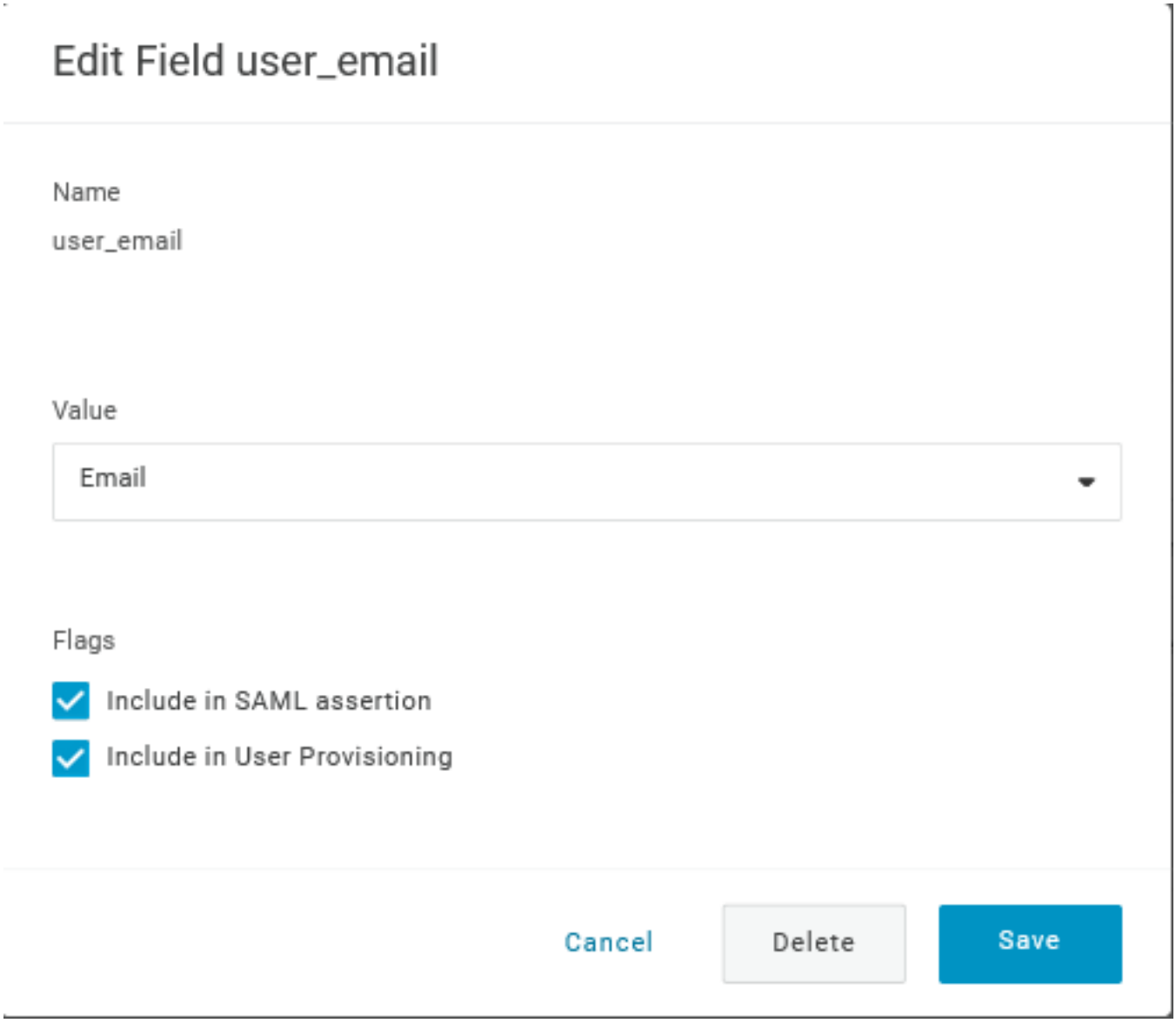


3. Na página *Provisioning*, habilite a opção Provisioning:



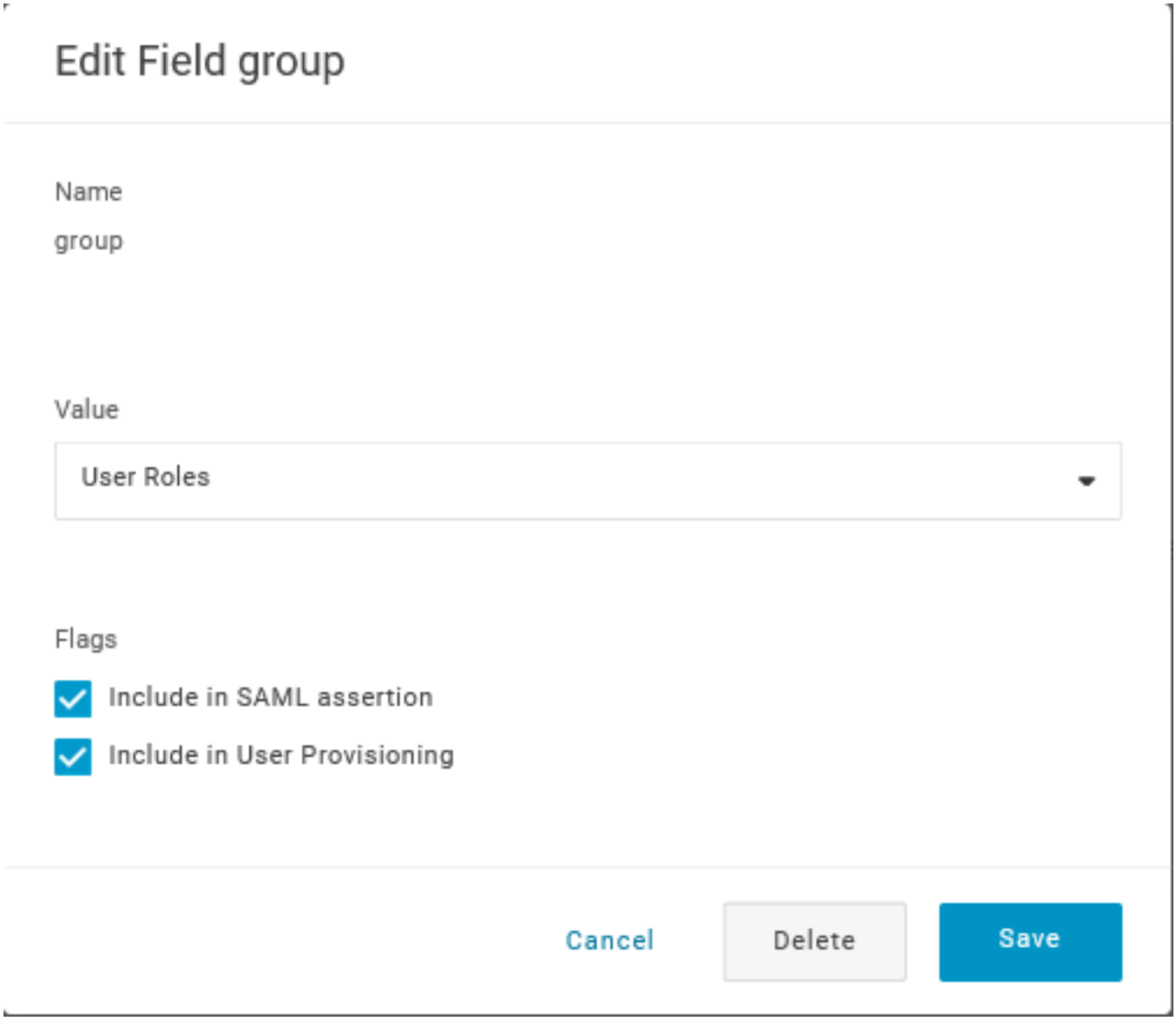
4. A página *Parameters* contém uma lista de parâmetros padrão:

- Certifique-se de que o 'scimusername' corresponda ao valor de login do usuário no OneLogin (por exemplo, e-mail);
- Marque a opção *Include in User Provisioning* para o parâmetro 'Groups';
- Clique em "+" para criar os parâmetros personalizados necessários para as asserções SAML e o provisionamento de usuários, como `user_name`, `user_lastname`, `user_email` e `user_mobile`:

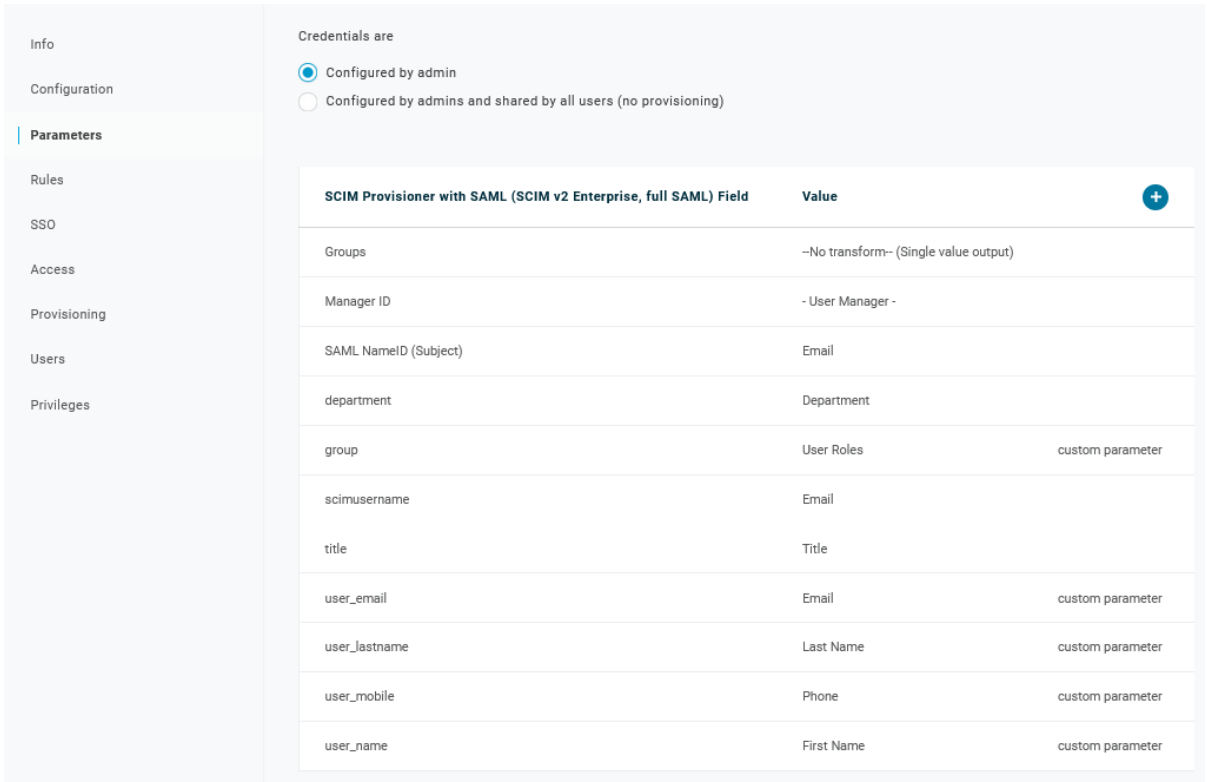


Ao adicionar um parâmetro, certifique-se de marcar as opções *Include in SAML assertion* e *Include in User Provisioning*.

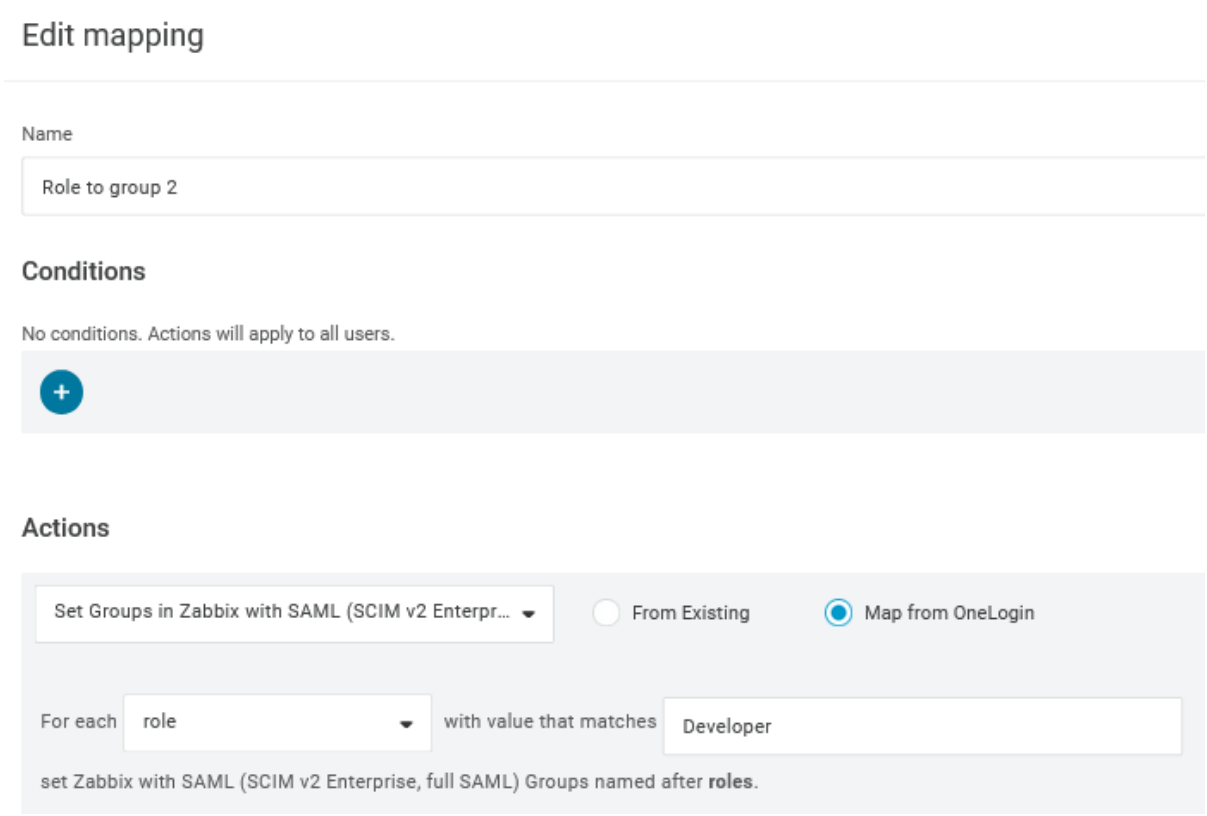
- Adicione um parâmetro 'group' que corresponda aos papéis de usuário no OneLogin. Os papéis de usuário serão passados como uma string, separados por ponto e vírgula `;`. Os papéis de usuário do OneLogin serão usados para criar grupos de usuários no Zabbix:



Verifique a lista de parâmetros:



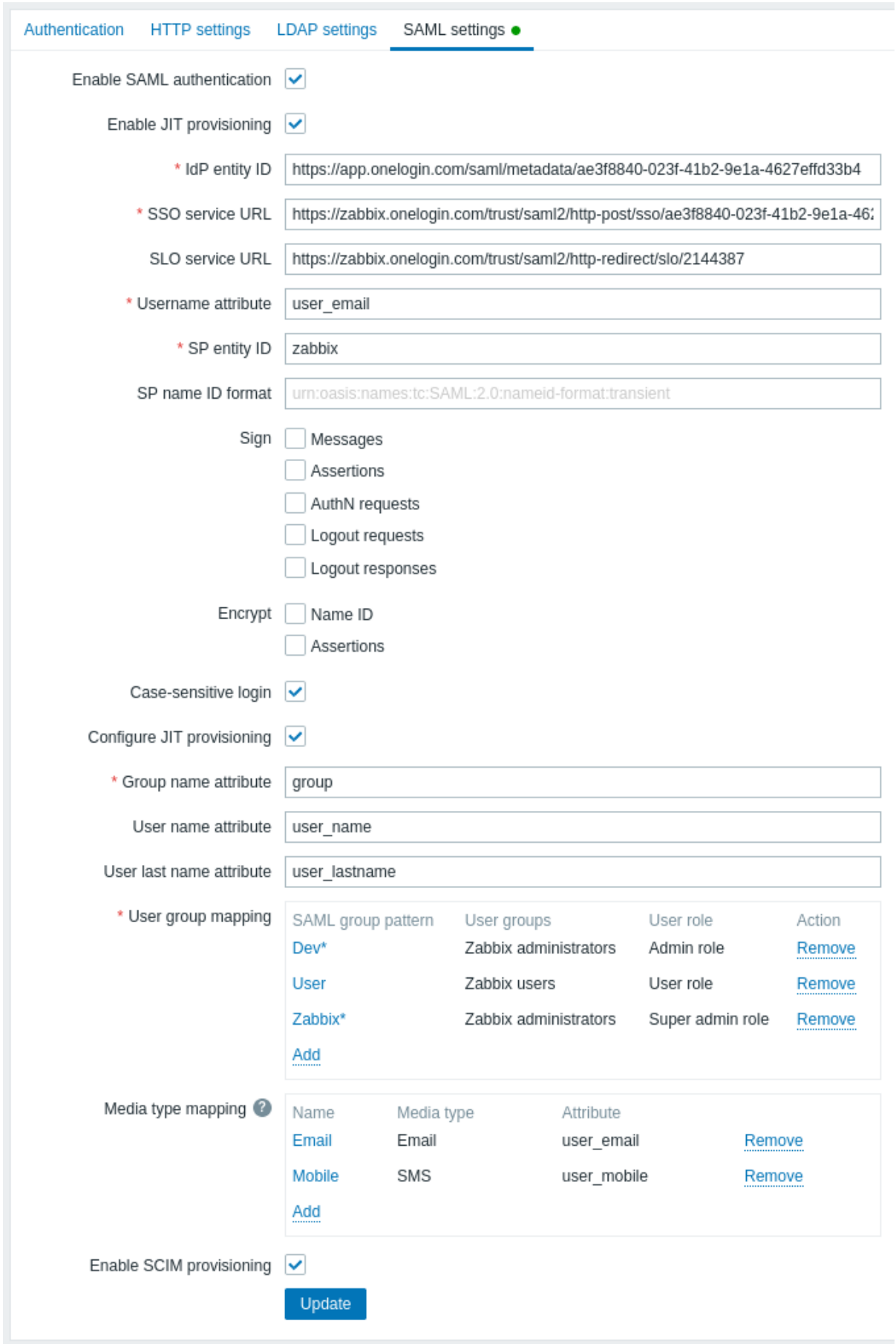
5. Na página *Rules*, crie mapeamentos de papéis de usuário para o parâmetro padrão Groups.



Você pode usar uma expressão regular para passar papéis específicos como grupos. Os nomes dos papéis não devem conter `;`, pois o OneLogin o utiliza como separador ao enviar um atributo com vários papéis.

Configuração do Zabbix

1. No Zabbix, acesse as configurações SAML e preencha as opções de configuração com base na configuração do OneLogin:



Campo do Zabbix	Campo de configuração no OneLogin	Valor de exemplo
IdP entity ID	Issuer URL (veja a aba SSO do seu aplicativo no OneLogin)	
SSO service URL	SAML 2.0 Endpoint (HTTP) (veja a aba SSO do seu aplicativo no OneLogin)	

Campo do Zabbix	Campo de configuração no OneLogin	Valor de exemplo
SLO service URL	SLO Endpoint (HTTP) (veja a aba SSO do seu aplicativo no OneLogin)	
Username attribute	Parâmetro personalizado	`user_email`
Group name attribute	Parâmetro personalizado	`group`
User name attribute	Parâmetro personalizado	`user_name`
User last name attribute	Parâmetro personalizado	`user_lastname`

Também é necessário configurar o mapeamento de grupos de usuários. O mapeamento de mídia é opcional. Clique em *Atualizar* para salvar essas configurações.

2. Baixe o certificado fornecido pelo OneLogin e coloque-o em `conf/certs` da instalação do frontend do Zabbix, como idp.crt.

Defina as permissões 644 executando:

```
chmod 644 idp.crt
```

Você pode acessar o download do certificado no OneLogin em *Applications* -> *SSO* -> clique em *View details* abaixo do certificado atual.

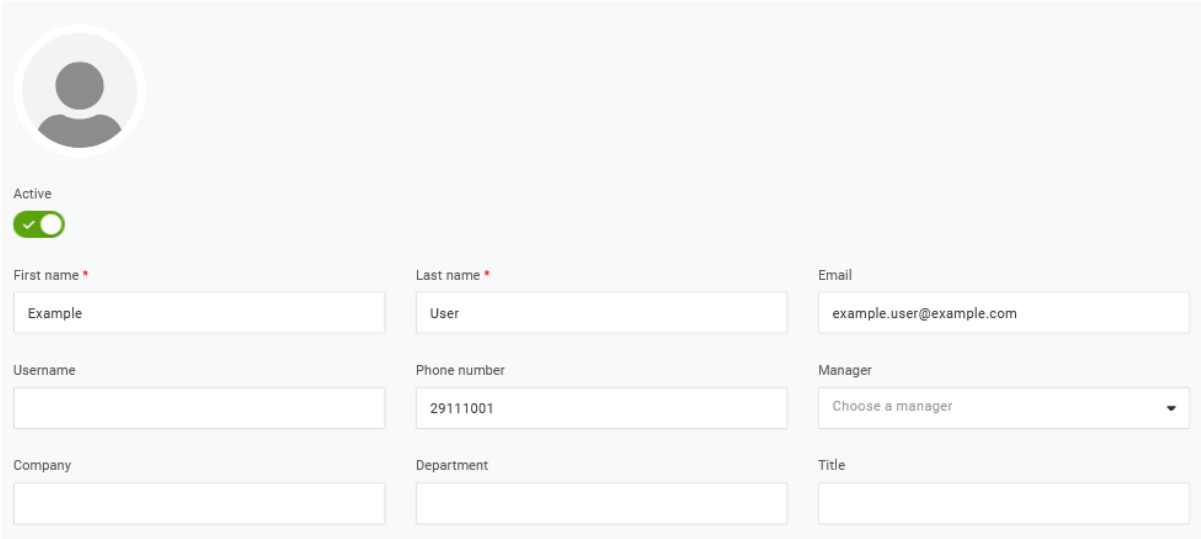
É possível usar um nome e local de certificado diferente. Nesse caso, certifique-se de adicionar ao `conf/zabbix.conf.php` a seguinte linha:

```
$SSO['IDP_CERT'] = 'path/to/certname.crt';
```

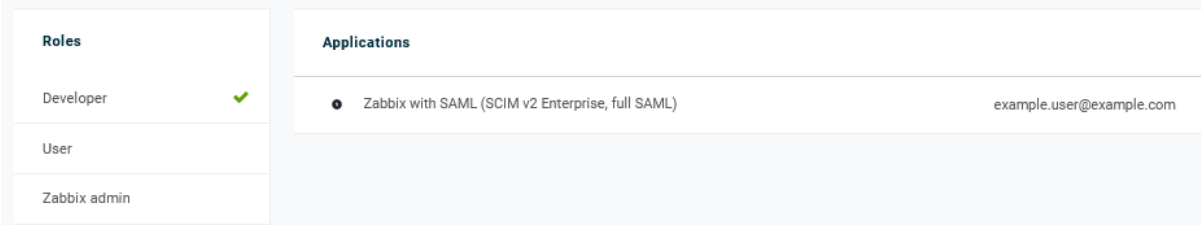
Provisionamento de usuários SCIM

Com o provisionamento de usuários ativado, agora é possível adicionar/atualizar usuários e seus papéis no OneLogin e tê-los imediatamente provisionados no Zabbix.

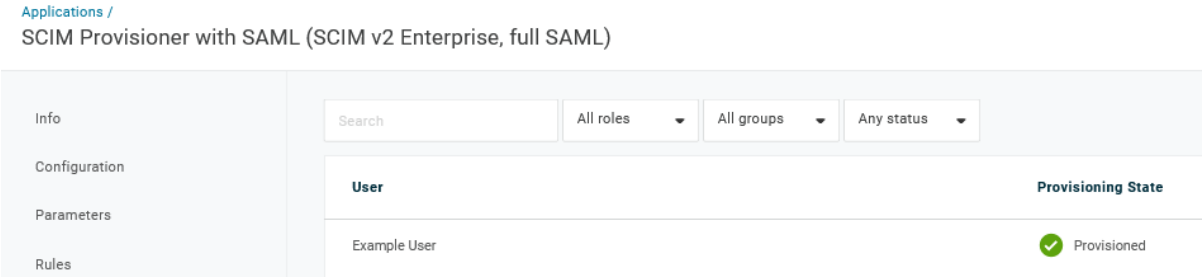
Por exemplo, você pode criar um novo usuário:



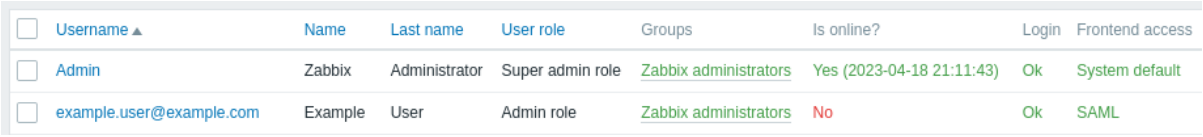
Adicione-o a um papel de usuário e ao aplicativo que irá provisionar o usuário:



Ao salvar o usuário, ele será provisionado no Zabbix. Em Aplicativo -> Usuários, você pode verificar o status de provisionamento dos usuários atuais do aplicativo:



Se for provisionado com sucesso, o usuário poderá ser visto na lista de usuários do Zabbix.



Username ▲	Name	Last name	User role	Groups	Is online?	Login	Frontend access
Admin	Zabbix	Administrator	Super admin role	Zabbix administrators	Yes (2023-04-18 21:11:43)	Ok	System default
example.user@example.com	Example	User	Admin role	Zabbix administrators	No	Ok	SAML

14 Configuração do banco de dados Oracle

Visão geral

Esta seção contém instruções para criar o banco de dados Oracle e configurar as conexões entre o banco de dados e o Zabbix server, proxy e frontend.

Attention:
O suporte ao Oracle DB está obsoleto desde o Zabbix 7.0.

Criação do banco de dados

Assumimos que um usuário de banco de dados *zabbix* com a senha *password* existe e tem permissões para criar objetos de banco de dados no serviço ORCL localizado no servidor de banco de dados Oracle *host*. O Zabbix requer um conjunto de caracteres Unicode e um conjunto de caracteres nacionais UTF8. Verifique as configurações atuais:

```
sqlplus> select parameter,value from v$nls_parameters where parameter='NLS_CHARACTERSET' or parameter='NLS
```

Agora prepare o banco de dados:

```
cd /path/to/zabbix-sources/database/oracle
sqlplus zabbix/password@oracle_host/ORCL
sqlplus> @schema.sql
# pare aqui se estiver criando o banco de dados para o Zabbix proxy
sqlplus> @images.sql
sqlplus> @data.sql
```

Note:
Por favor, defina o parâmetro de inicialização CURSOR_SHARING=FORCE para melhor desempenho.

Configuração da conexão

O Zabbix suporta dois tipos de identificadores de conexão (métodos de conexão):

- Easy Connect
- Net Service Name

Os parâmetros de configuração de conexão para o Zabbix server e o Zabbix proxy podem ser definidos nos arquivos de configuração. Os parâmetros importantes para o server e o proxy são *DBHost*, *DBUser*, *DBName* e *DBPassword*. Os mesmos parâmetros são importantes para o frontend: *$DB["SERVER"]*, *$DB["PORT"]*, *$DB["DATABASE"]*, *$DB["USER"]*, *$DB["PASSWORD"]*.

O Zabbix usa a seguinte sintaxe de string de conexão:

```
{DBUser/DBPassword[@<connect_identifier>]}
```

<connect_identifier> pode ser especificado tanto na forma de "Net Service Name" quanto de "Easy Connect".

```
@[[//]Host[:Port]/<service_name> | <net_service_name>]
```

Easy Connect

O Easy Connect usa os seguintes parâmetros para se conectar ao banco de dados:

- *Host* - o nome do host ou endereço IP do computador do servidor de banco de dados (parâmetro DBHost no arquivo de configuração).
- *Port* - a porta de escuta no servidor de banco de dados (parâmetro DBPort no arquivo de configuração; se não for definida, a porta padrão 1521 será usada).
- <service_name> - o nome do serviço do banco de dados que você deseja acessar (parâmetro DBName no arquivo de configuração).

Exemplo

Parâmetros do banco de dados definidos no arquivo de configuração do server ou proxy (zabbix_server.conf e zabbix_proxy.conf):

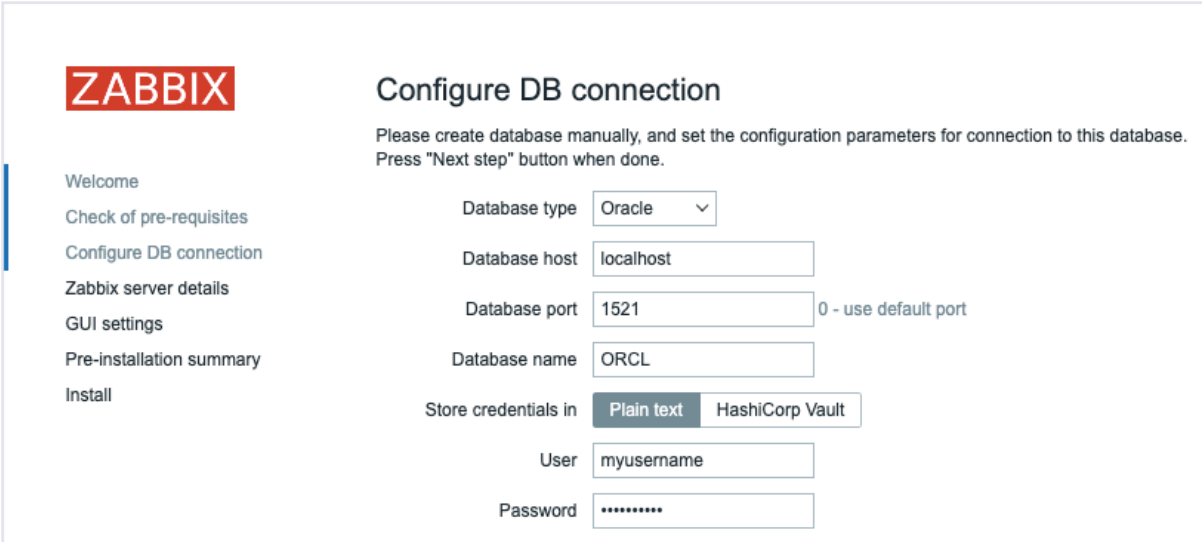
```
DBHost=localhost
DBPort=1521
DBUser=myusername
DBName=ORCL
DBPassword=mypassword
```

String de conexão usada pelo Zabbix para estabelecer a conexão:

```
DBUser/DBPassword@DBHost:DBPort/DBName
```

Durante a instalação do frontend do Zabbix, defina os parâmetros correspondentes na etapa *Configure DB connection* do assistente de instalação:

- Database host: localhost
- Database port: 1521
- Database name: ORCL
- User: myusername
- Password: mypassword



Como alternativa, esses parâmetros podem ser definidos no arquivo de configuração do frontend (zabbix.conf.php):

```
$DB["TYPE"]                     = 'ORACLE';
$DB["SERVER"]                   = 'localhost';
$DB["PORT"]             = '1521';
$DB["DATABASE"]                 = 'ORCL';
$DB["USER"]                     = 'myusername';
$DB["PASSWORD"]                 = 'mypassword';
```

Nome do serviço de rede

É possível conectar ao Oracle usando o nome do serviço de rede.

<net_service_name> é um nome simples para um serviço que resolve para um descritor de conexão.

Para usar o nome do serviço para criar uma conexão, este nome de serviço deve ser definido no arquivo tnsnames.ora localizado tanto no servidor de banco de dados quanto nos sistemas clientes. A maneira mais fácil de garantir que a conexão será bem-sucedida é definir a localização do arquivo tnsnames.ora na variável de ambiente TNS_ADMIN. O local padrão do arquivo tnsnames.ora é:

```
$ORACLE_HOME/network/admin/
```

Um exemplo simples de arquivo tnsnames.ora:

```
ORCL =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = localhost)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = ORCL)
    )
  )
```

Para definir os parâmetros de configuração para o método de conexão "Nome do Serviço de Rede", use uma das seguintes opções:

- Defina um parâmetro DBHost vazio e defina DBName normalmente:

```
DBHost=
DBName=ORCL
```

- Defina ambos os parâmetros e deixe ambos vazios:

```
DBHost=
DBName=
```

No segundo caso, a variável de ambiente TWO_TASK deve ser definida. Ela especifica o serviço Oracle remoto padrão (nome do serviço). Quando esta variável é definida, o conector conecta-se ao banco de dados especificado usando um listener Oracle que aceita solicitações de conexão. Esta variável é para uso apenas em Linux e UNIX. Use a variável de ambiente LOCAL para Microsoft Windows.

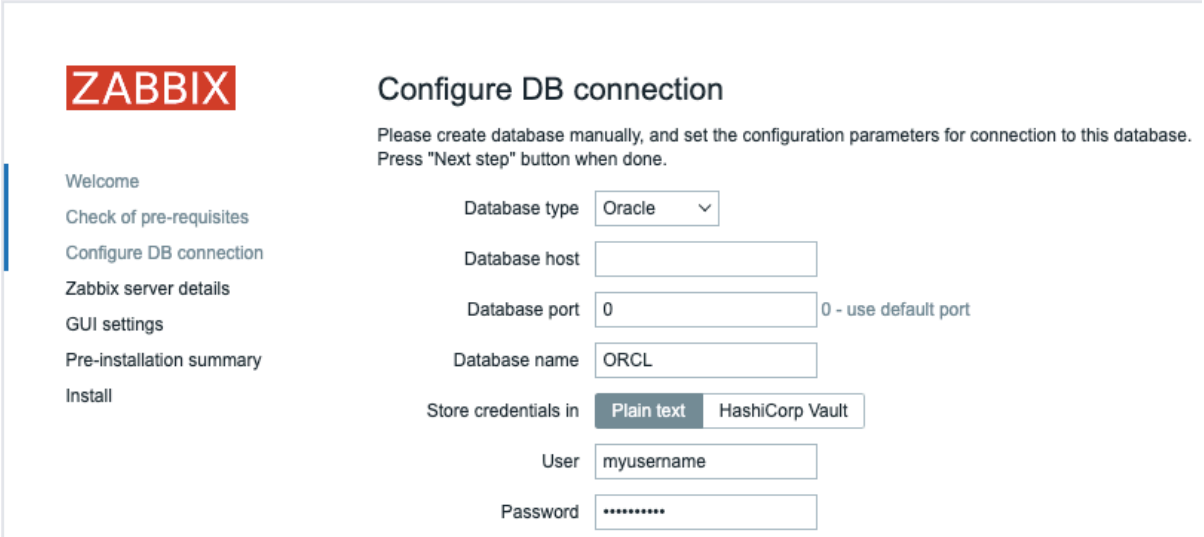
Exemplo

Conecte-se a um banco de dados usando o Nome do Serviço de Rede definido como ORCL e a porta padrão. Parâmetros do banco de dados definidos no arquivo de configuração do server ou proxy (zabbix_server.conf e zabbix_proxy.conf):

```
DBHost=
#DBPort=
DBUser=myusername
DBName=ORCL
DBPassword=mypassword
```

Durante a instalação do frontend do Zabbix, defina os parâmetros correspondentes na etapa *Configure DB connection* do assistente de configuração:

- Database host:
- Database port: 0
- Database name: ORCL
- User: myusername
- Password: mypassword



Alternativamente, esses parâmetros podem ser definidos no arquivo de configuração do frontend (zabbix.conf.php):

```
$DB["TYPE"]                     = 'ORACLE';
$DB["SERVER"]                   = '';
$DB["PORT"]             = '0';
$DB["DATABASE"]                 = 'ORCL';
```

```
$DB["USER"]                     = 'myusername';
$DB["PASSWORD"]                 = 'mypassword';
```

String de conexão usada pelo Zabbix para estabelecer a conexão:

`DBUser/DBPassword@ORCL`

Problemas conhecidos

Para melhorar o desempenho, você pode converter os tipos de campo de *nclob* para *nvarchar2*, consulte problemas conhecidos.

15 Atualizando para valores numéricos de faixa estendida

Visão geral

Desde o Zabbix 5.0, o tipo de dado numérico (float) suporta precisão de aproximadamente 15 dígitos e intervalo de aproximadamente -1.79E+308 a 1.79E+308. O antigo tipo numérico foi desabilitado, mas ainda suportado. A partir do Zabbix 6.4, ele foi depreciado e, a partir do Zabbix 7.0, foi totalmente removido.

Se sua instalação não foi atualizada para usar valores numéricos de intervalo estendido, o seguinte aviso é exibido em *Relatórios* → *Informações do sistema*: "Tabelas de histórico do banco de dados atualizadas: Não. O suporte ao antigo tipo numérico está depreciado. Por favor, atualize para valores numéricos de intervalo estendido".

Ao atualizar para o Zabbix 7.0, os seguintes bancos de dados são atualizados **automaticamente**:

- MySQL
- PostgreSQL/TimescaleDB (sem compressão)
- Oracle

No entanto, para Oracle, versões antigas do MySQL e instalações grandes, recomenda-se atualizar o tipo de dado manualmente antes de atualizar para o Zabbix 7.0. Para instruções, consulte MySQL, PostgreSQL/TimescaleDB (sem compressão), Oracle.

Se estiver usando TimescaleDB com compressão, o banco de dados deve ser atualizado **manualmente** antes de atualizar para o Zabbix 7.0. Para instruções, consulte TimescaleDB (com compressão).

> **Attention:**
> A partir do Zabbix 7.0, a atualização manual das tabelas para chaves primárias também atualiza as tabelas para usar tipos de dados de dupla precisão. Para instruções sobre como atualizar para chaves primárias, consulte Atualização do banco de dados para chaves primárias.

MySQL, PostgreSQL/TimescaleDB (sem compressão), Oracle

O patch de atualização altera as colunas de dados das tabelas de histórico e tendências, que geralmente contêm muitos dados; por esse motivo, a atualização pode levar algum tempo para ser concluída. A estimativa exata não pode ser prevista e depende do desempenho do servidor, da configuração e da versão do sistema de gerenciamento de banco de dados. Portanto, recomenda-se primeiro testar o patch fora do ambiente de produção. Por exemplo, com MySQL 8.0 e MariaDB 10.5 (configuração padrão), sabe-se que o patch é executado instantaneamente para tabelas grandes devido ao algoritmo eficiente e ao fato de que anteriormente o mesmo tipo double era usado, mas com precisão limitada, o que significa que os próprios dados não precisam ser modificados.

Para atualizar o MySQL, PostgreSQL/TimescaleDB (sem compressão) ou Oracle antes de atualizar para o Zabbix 7.0:

1. Pare o Zabbix server.
2. Faça backup do seu banco de dados.
3. Execute o patch apropriado (arquivo SQL) para o seu banco de dados Zabbix 6.4:

> **Warning:**
> Execute os scripts apenas para o banco de dados do server. O proxy não se beneficiará desta atualização.

- `/usr/share/zabbix-sql-scripts/mysql/double.sql`
- `/usr/share/zabbix-sql-scripts/postgresql/double.sql` (para PostgreSQL/TimescaleDB)
- `/usr/share/zabbix-sql-scripts/oracle/double.sql`

Se estiver usando fontes, consulte o patch do MySQL, PostgreSQL/TimescaleDB ou Oracle para o Zabbix 6.4 no repositório GIT do Zabbix.

4. Inicie o Zabbix server.
5. Atualize ou defina o valor do parâmetro `$DB['DOUBLE_IEEE754']` como `true` em `/ui/conf/zabbix.conf.php`.

TimescaleDB (com compressão)

Para atualizar o TimescaleDB (com compressão) antes de atualizar para o Zabbix 7.0:

1. Desabilite a compressão do TimescaleDB no frontend do Zabbix.

2. Recarregue o cache de configuração do Zabbix server usando o controle em tempo de execução: `zabbix_server -R config_cache_reload`

3. Pare o Zabbix server.

4. Faça backup do seu banco de dados.

5. Execute o patch apropriado (arquivo SQL) para o TimescaleDB:

   - `/usr/share/zabbix-sql-scripts/postgresql/double.sql`

Se estiver usando os fontes, veja o patch PostgreSQL/TimescaleDB para o Zabbix 6.4 no repositório GIT do Zabbix.

6. Inicie o Zabbix server.

7. Atualize ou defina o valor do parâmetro `$DB['DOUBLE_IEEE754']` para `true` em `/ui/conf/zabbix.conf.php`.

16 Configurando relatórios agendados

Visão geral

Esta seção fornece instruções sobre como instalar o serviço web do Zabbix e configurar o Zabbix para permitir a geração de relatórios agendados.

Instalação

Um novo processo do serviço web do Zabbix e o navegador Google Chrome devem ser instalados para permitir a geração de relatórios agendados. O serviço web pode ser instalado na mesma máquina onde o Zabbix server está instalado ou em uma máquina diferente. O navegador Google Chrome deve ser instalado na mesma máquina onde o serviço web está instalado.

O pacote oficial zabbix-web-service está disponível no repositório Zabbix. O navegador Google Chrome não está incluído nesses pacotes e deve ser instalado separadamente.

Para compilar o serviço web do Zabbix a partir do código-fonte, consulte Instalando o serviço web do Zabbix.

Após a instalação, execute o zabbix_web_service na máquina onde o serviço web está instalado:

```
zabbix_web_service
```

Configuração

Para garantir a comunicação adequada entre todos os elementos envolvidos, certifique-se de que o arquivo de configuração do server e os parâmetros de configuração do frontend estejam devidamente configurados.

Zabbix server

Os seguintes parâmetros no arquivo de configuração do Zabbix server precisam ser atualizados: *WebServiceURL* e *StartReportWriters*.

WebServiceURL

Este parâmetro é necessário para habilitar a comunicação com o serviço web. A URL deve estar no formato `http[s]://host:port/report`.

- Por padrão, o serviço web escuta na porta 10053. Uma porta diferente pode ser especificada no arquivo de configuração do serviço web.
- Especificar o caminho `/report` é obrigatório (o caminho é fixo e não pode ser alterado).

Exemplo:

```
WebServiceURL=http://localhost:10053/report
```

> **Attention:**
> É altamente recomendável configurar a criptografia entre o Zabbix server e o Zabbix web service usando certificados. Por padrão, os dados transmitidos entre o Zabbix server e o Zabbix web service não são criptografados, o que pode levar a acessos não autorizados.

StartReportWriters

Este parâmetro determina quantos processos de geração de relatórios devem ser iniciados. Se não for definido ou for igual a 0, a geração de relatórios é desabilitada. Com base no número e na frequência dos relatórios necessários, é possível habilitar de 1 a 100 processos de geração de relatórios.

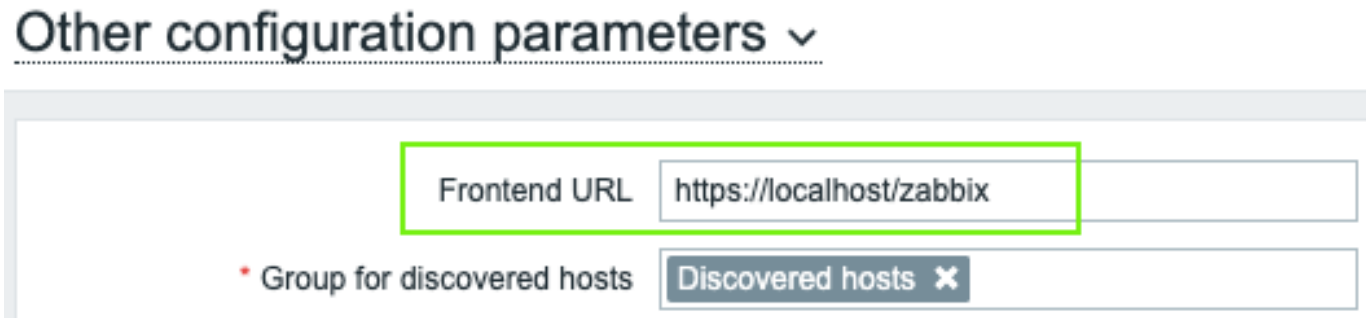
Exemplo:

```
StartReportWriters=3
```

Frontend do Zabbix

O parâmetro *Frontend URL* deve ser definido para habilitar a comunicação entre o frontend do Zabbix e o serviço web do Zabbix:

- Acesse a seção de menu do frontend *Administração* > *Geral* > *Outros*
- Especifique a URL completa da interface web do Zabbix no parâmetro *Frontend URL*



Note:
Após a conclusão do procedimento de configuração, você pode configurar e enviar um relatório de teste para garantir que tudo está funcionando corretamente.

17 Idiomas adicionais do frontend

Visão geral

Para usar qualquer outro idioma que não o inglês na interface web do Zabbix, sua localidade deve ser instalada no servidor web. Além disso, o PHP A extensão gettext é necessária para que as traduções funcionem.

Instalando localidades

Para listar todos os idiomas instalados, execute:

```
localidade -a
```

Se alguns idiomas necessários não estiverem listados, abra o */etc/locale.gen* e descomente os códigos de idioma necessários. Desde o Zabbix usa a codificação UTF-8, você precisa selecionar localidades com o conjunto de caracteres UTF-8.

Agora, execute:

```
locale-gen
```

Reinicie o servidor web.

As localidades agora devem ser instaladas. Pode ser necessário recarregar o Zabbix página de frontend no navegador usando Ctrl + F5 para que novos idiomas apareçam.

Instalando o Zabbix

Se instalar o Zabbix diretamente do Zabbix git repositório, arquivos de tradução devem ser gerados manualmente. Para gerar a tradução arquivos, execute:

```
fazer gettext
locale/make_mo.sh
```

Este passo não é necessário ao instalar o Zabbix a partir de pacotes ou fontes arquivos tar.gz.

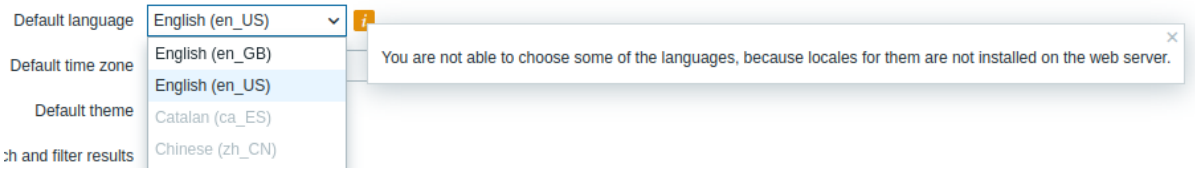
Selecionando um idioma

Existem várias maneiras de selecionar um idioma na interface web do Zabbix:

- Ao instalar a interface web - no assistente de instalação do frontend. O idioma selecionado será definido como padrão do sistema.

- Após a instalação, o idioma padrão do sistema pode ser alterado na seção de menu *Administração→Geral→GUI* (menu section).
- O idioma para um usuário específico pode ser alterado no perfil do usuário.

Se um locale para um idioma não estiver instalado na máquina, esse idioma ficará desabilitado no seletor de idiomas do Zabbix. Um ícone laranja é exibido ao lado do seletor de idiomas se pelo menos um locale estiver ausente. Ao pressionar este ícone, a seguinte mensagem será exibida: ”Você não pode escolher alguns dos idiomas porque os locales para eles não estão instalados no servidor web.”



2 Configuração do processo

Use a barra lateral para acessar o conteúdo desta seção.

1 servidor Zabbix

Visão geral

Os parâmetros suportados pelo arquivo de configuração do Zabbix server (zabbix_server.conf) estão listados nesta seção.

Os parâmetros são listados sem informações adicionais. Clique no parâmetro para ver os detalhes completos.

Parâmetro	Descrição
AlertScriptsPath	O local dos scripts de alerta personalizados.
AllowRoot	Permite que o server seja executado como ’root’.
AllowSoftwareUpdateCheck	Permite que o Zabbix UI receba informações sobre atualizações de software do zabbix.com.
AllowUnsupportedDBVersions	Permite que o server funcione com versões de banco de dados não suportadas.
CacheSize	O tamanho do cache de configuração.
CacheUpdateFrequency	Este parâmetro determina com que frequência o Zabbix executará a atualização do cache de configuração em segundos.
DBHost	O nome do host do banco de dados.
DBName	O nome do banco de dados.
DBPassword	A senha do banco de dados.
DBPort	A porta do banco de dados quando não estiver usando o socket local.
DBSchema	O nome do schema do banco de dados. Usado para PostgreSQL.
DBSocket	O caminho para o arquivo de socket do MySQL.
DBUser	O usuário do banco de dados.
DBTLSConnect	Definir esta opção para o valor especificado força o uso de uma conexão TLS com o banco de dados.
DBTLSCAFile	O caminho completo de um arquivo contendo os certificados da(s) CA(s) de nível superior para verificação do certificado do banco de dados.
DBTLSCertFile	O caminho completo de um arquivo contendo o certificado do Zabbix server para autenticação no banco de dados.
DBTLSKeyFile	O caminho completo de um arquivo contendo a chave privada para autenticação no banco de dados.
DBTLSCipher	A lista de cifras de criptografia que o Zabbix server permite para protocolos TLS até o TLS v1.2. Suportado apenas para MySQL.
DBTLSCipher13	A lista de cifras de criptografia que o Zabbix server permite para o protocolo TLS v1.3. Suportado apenas para MySQL, a partir da versão 8.0.16.
DebugLevel	Especifica o nível de debug.
EnableGlobalScripts	Habilita scripts globais no Zabbix server.
ExportDir	O diretório para exportação em tempo real de eventos, histórico e tendências em formato JSON delimitado por nova linha. Se definido, habilita a exportação em tempo real.
ExportFileSize	O tamanho máximo por arquivo de exportação em bytes.
ExportType	A lista de tipos de entidades separados por vírgula (eventos, histórico, tendências) para exportação em tempo real (todos os tipos por padrão).

Parâmetro	Descrição
ExternalScripts	O local dos scripts externos.
Fping6Location	O local do fping6.
FpingLocation	O local do fping.
HANodeName	O nome do nó do cluster de alta disponibilidade.
HistoryCacheSize	O tamanho do cache de histórico.
HistoryIndexCacheSize	O tamanho do cache de índice de histórico.
HistoryStorageDateIndex	Habilita o pré-processamento de valores de histórico no armazenamento de histórico para armazenar valores em índices diferentes com base na data.
HistoryStorageURL	A URL HTTP[S] do armazenamento de histórico.
HistoryStorageTypes	Uma lista separada por vírgulas de tipos de valores a serem enviados para o armazenamento de histórico.
HousekeepingFrequency	Este parâmetro determina com que frequência o Zabbix executará o procedimento de housekeeping em horas.
Include	Você pode incluir arquivos individuais ou todos os arquivos em um diretório no arquivo de configuração.
JavaGateway	O endereço IP (ou nome do host) do Zabbix Java gateway.
JavaGatewayPort	A porta que o Zabbix Java gateway escuta.
ListenBacklog	O número máximo de conexões pendentes na fila TCP.
ListenIP	Uma lista de endereços IP separados por vírgula que o trapper deve escutar.
ListenPort	A porta de escuta para o trapper.
LoadModule	O módulo a ser carregado na inicialização do server.
LoadModulePath	O caminho completo para o local dos módulos do server.
LogFile	O nome do arquivo de log.
LogFileSize	O tamanho máximo do arquivo de log.
LogSlowQueries	Determina quanto tempo uma consulta ao banco de dados pode levar antes de ser registrada em milissegundos.
LogType	O tipo de saída do log.
MaxConcurrentChecksPerPoller	O número máximo de verificações assíncronas que podem ser executadas de uma vez por cada poller HTTP agent, poller agent ou poller SNMP.
MaxHousekeeperDelete	Não mais do que 'MaxHousekeeperDelete' linhas (correspondentes a [tablename], [field], [value]) serão excluídas por uma tarefa em um ciclo de housekeeping.
NodeAddress	O IP ou nome do host com porta opcional para substituir como o frontend deve se conectar ao server.
PidFile	O nome do arquivo PID.
ProblemHousekeepingFrequency	Determina com que frequência o Zabbix excluirá problemas de triggers excluídos.
ProxyConfigFrequency	Determina com que frequência o Zabbix server envia dados de configuração para um Zabbix proxy.
ProxyDataFrequency	Determina com que frequência o Zabbix server solicita dados históricos de um Zabbix proxy.
ServiceManagerSyncFrequency	Determina com que frequência o Zabbix sincronizará a configuração de um gerenciador de serviços.
SMSDevices	Uma lista de arquivos de modem separados por vírgula permitidos para uso pelo Zabbix server.
SNMPTrapperFile	O arquivo temporário usado para passar dados do daemon de trap SNMP para o server.
SocketDir	O diretório para armazenar os sockets IPC usados pelos serviços internos do Zabbix.
SourceIP	O endereço IP de origem.
SSHKeyLocation	O local das chaves públicas e privadas para verificações e ações SSH.
SSLCertLocation	O local dos arquivos de certificado de cliente SSL para autenticação de cliente.
SSLKeyLocation	O local dos arquivos de chave privada SSL para autenticação de cliente.
SSLCALocation	Substitui o local dos arquivos da autoridade certificadora (CA) para verificação do certificado do server SSL.
StartAgentPollers	O número de instâncias pré-forked de agent pollers assíncronos do Zabbix.
StartAlerters	O número de instâncias pré-forked de alerters.
StartBrowserPollers	O número de instâncias pré-forked de browser item pollers.
StartConnectors	O número de instâncias pré-forked de workers de conector.
StartDBSyncers	O número de instâncias pré-forked de history syncers.
StartDiscoverers	O número de instâncias pré-forked de discovery workers.
StartEscalators	O número de instâncias pré-forked de escalators.
StartHistoryPollers	O número de instâncias pré-forked de history pollers.
StartHTTPAgentPollers	O número de instâncias pré-forked de HTTP agent pollers assíncronos.
StartHTTPPollers	O número de instâncias pré-forked de HTTP pollers.
StartIPMIPollers	O número de instâncias pré-forked de IPMI pollers.
StartJavaPollers	O número de instâncias pré-forked de Java pollers.

Parâmetro	Descrição
StartLLDProcessors	O número de instâncias pré-forked de workers de descoberta de baixo nível (LLD).
StartODBCPollers	O número de instâncias pré-forked de ODBC pollers.
StartPingers	O número de instâncias pré-forked de ICMP pingers.
StartPollersUnreachable	O número de instâncias pré-forked de pollers para hosts inacessíveis (incluindo IPMI e Java).
StartPollers	O número de instâncias pré-forked de pollers.
StartPreprocessors	O número de instâncias pré-iniciadas de workers de pré-processamento.
StartProxyPollers	O número de instâncias pré-forked de pollers para proxies passivos.
StartReportWriters	O número de instâncias pré-forked de report writers.
StartSNMPPollers	O número de instâncias pré-forked de SNMP pollers assíncronos.
StartSNMPTrapper	Se definido como 1, um processo SNMP trapper será iniciado.
StartTimers	O número de instâncias pré-forked de timers.
StartTrappers	O número de instâncias pré-forked de trappers.
StartVMwareCollectors	O número de instâncias pré-forked de VMware collector.
StatsAllowedIP	Uma lista de endereços IP separados por vírgula, opcionalmente na notação CIDR, ou nomes DNS de instâncias externas do Zabbix. A solicitação de estatísticas será aceita apenas dos endereços listados aqui.
Timeout	Especifica quanto tempo esperar (em segundos) para estabelecer conexão e trocar dados com Zabbix proxy, agent, serviço web e para verificações SNMP (exceto itens SNMP `walk[OID]` e `get[OID]`).
TLSCAFile	O caminho completo de um arquivo contendo os certificados da(s) CA(s) de nível superior para verificação do certificado do peer, usado para comunicações criptografadas entre componentes do Zabbix.
TLSCertFile	O caminho completo de um arquivo contendo o certificado do server ou cadeia de certificados, usado para comunicações criptografadas entre componentes do Zabbix.
TLSCipherAll	A string de prioridade do GnuTLS ou a string de cifra do OpenSSL (TLS 1.2). Substitui os critérios padrão de seleção de ciphersuite para criptografia baseada em certificado e PSK.
TLSCipherAll13	A string de cifra para OpenSSL 1.1.1 ou mais recente em TLS 1.3. Substitui os critérios padrão de seleção de ciphersuite para criptografia baseada em certificado e PSK.
TLSCipherCert	A string de prioridade do GnuTLS ou a string de cifra do OpenSSL (TLS 1.2). Substitui os critérios padrão de seleção de ciphersuite para criptografia baseada em certificado.
TLSCipherCert13	A string de cifra para OpenSSL 1.1.1 ou mais recente em TLS 1.3. Substitui os critérios padrão de seleção de ciphersuite para criptografia baseada em certificado.
TLSCipherPSK	A string de prioridade do GnuTLS ou a string de cifra do OpenSSL (TLS 1.2). Substitui os critérios padrão de seleção de ciphersuite para criptografia baseada em PSK.
TLSCipherPSK13	A string de cifra para OpenSSL 1.1.1 ou mais recente em TLS 1.3. Substitui os critérios padrão de seleção de ciphersuite para criptografia baseada em PSK.
TLSCRLFile	O caminho completo de um arquivo contendo certificados revogados. Este parâmetro é usado para comunicações criptografadas entre componentes do Zabbix.
TLSKeyFile	O caminho completo de um arquivo contendo a chave privada do server, usado para comunicações criptografadas entre componentes do Zabbix.
TmpDir	O diretório temporário.
TrapperTimeout	Especifica o timeout em segundos para: - recuperação de dados históricos do Zabbix proxy; - envio de dados de configuração para o Zabbix proxy; - execução de script global ou execução de comando remoto no Zabbix server.
TrendCacheSize	O tamanho do cache de tendências.
TrendFunctionCacheSize	O tamanho do cache de função de tendências.
UnavailableDelay	Determina com que frequência o host é verificado quanto à disponibilidade durante o período de indisponibilidade.
UnreachableDelay	Determina com que frequência o host é verificado quanto à disponibilidade durante o período de inacessibilidade.
UnreachablePeriod	Determina após quantos segundos de inacessibilidade um host é tratado como indisponível.
User	Remove privilégios para um usuário específico e existente no sistema.
ValueCacheSize	O tamanho do cache de valores históricos.
Vault	Especifica o provedor de vault.
VaultDBPath	Especifica um local de onde as credenciais do banco de dados devem ser recuperadas por chaves.
VaultPrefix	Prefixo personalizado para o caminho ou consulta do vault.
VaultTLSCertFile	O nome do arquivo de certificado SSL usado para autenticação de cliente.
VaultTLSKeyFile	O nome do arquivo de chave privada SSL usado para autenticação de cliente.
VaultToken	O token de autenticação do HashiCorp vault.

Parâmetro	Descrição
VaultURL	A URL HTTP[S] do vault server.
VMwareCacheSize	O tamanho da memória compartilhada para armazenar dados do VMware.
VMwareFrequency	O atraso em segundos entre a coleta de dados de um único serviço VMware.
VMwarePerfFrequency	O atraso em segundos entre a recuperação de estatísticas de contadores de desempenho de um único serviço VMware.
VMwareTimeout	O número máximo de segundos que um vmware collector aguardará por uma resposta do serviço VMware.
WebDriverURL	URL HTTP[S] da interface WebDriver.
WebServiceURL	URL HTTP[S] para o Zabbix web service no formato <host:port>/report.

Todos os parâmetros são não obrigatórios, a menos que explicitamente declarado que o parâmetro é obrigatório.

Observe que:

- Os valores padrão refletem os padrões do daemon, não os valores nos arquivos de configuração fornecidos;
- O Zabbix suporta arquivos de configuração apenas em codificação UTF-8 sem BOM;
- Comentários iniciados com "#" são suportados apenas no início da linha.

Detalhes do parâmetro

AlertScriptsPath

O local dos scripts de alerta personalizados (depende da variável de instalação *datadir* em tempo de compilação).

Padrão: `/usr/local/share/zabbix/alertscripts`

AllowRoot

Permite que o server seja executado como 'root'. Se desabilitado e o server for iniciado por 'root', o server tentará alternar para o usuário 'zabbix'. Não tem efeito se iniciado sob um usuário comum.

Padrão: 0
 Valores: 0 - não permitir; 1 - permitir

AllowSoftwareUpdateCheck

Permitir que a interface do Zabbix receba informações sobre atualizações de software do zabbix.com.

Padrão: 1
 Valores: 0 - não permitir; 1 - permitir

AllowUnsupportedDBVersions

Permite que o servidor trabalhe com versões de banco de dados sem suporte.

Padrão: 0
 Valores: 0 - não permitir; 1 - permitir

CacheSize

O tamanho do cache de configuração, em bytes. O tamanho da memória compartilhada para armazenar dados de host, item e trigger.

Padrão: 32M
 Intervalo: 128K-64G

CacheUpdateFrequency

Este parâmetro determina com que frequência o Zabbix irá realizar a atualização do cache de configuração em segundos. Veja também as opções de controle em tempo de execução.

Padrão: 10
 Intervalo: 1-3600

DBHost

O nome do host do banco de dados.
Com MySQL, `localhost` ou string vazia resulta no uso de um socket. Com PostgreSQL, se definido como string vazia, o socket padrão do domínio Unix é usado; se definido como um caminho (por exemplo, /var/run/pgbouncer), o socket do domínio Unix especificado é usado. Com Oracle, string vazia resulta no uso do método de conexão Net Service Name; nesse caso, considere usar a variável de ambiente TNS_ADMIN para especificar o diretório do arquivo tnsnames.ora.

Padrão: `localhost`

DBName

O nome do banco de dados.
Com Oracle, se o método de conexão Net Service Name for usado, especifique o nome do serviço em tnsnames.ora ou defina-o como uma cadeia de caracteres vazia; defina a variável de ambiente TWO_TASK se DBName for definido como uma cadeia de caracteres vazia.

Obrigatório: Sim

DBPassword

A senha do banco de dados. Comente esta linha se nenhuma senha for usada.

DBPort

A porta do banco de dados quando não estiver usando o socket de domínio Unix padrão.[3]
Com Oracle, se o método de conexão Net Service Name for usado, este parâmetro será ignorado; o número da porta do arquivo tnsnames.ora será usado em vez disso.

Padrão para MySQL: 3306

Padrão para PostgreSQL: 5432

Intervalo: 1024-65535

DBSchema

O nome do schema do banco de dados. Usado para PostgreSQL.

DBSocket

O caminho para o arquivo de socket do MySQL.[3]

DBUser

O usuário do banco de dados.

DBTLSConnect

Definir essa opção com os seguintes valores obriga o uso de uma conexão TLS com o banco de dados:
required - conecta-se usando TLS
verify_ca - conecta-se usando TLS e verifica o certificado
verify_full - conecta-se usando TLS, verifica o certificado e verifica se a identidade do banco de dados especificada pelo DBHost corresponde ao seu certificado

No MySQL, a partir da versão 5.7.11, e no PostgreSQL, os seguintes valores são suportados: `required`, `verify_ca`, `verify_full`.
Com o MariaDB, a partir da versão 10.2.6, os valores `required` e `verify_full` são suportados.
Por padrão, não é definida nenhuma opção e o comportamento depende da configuração do banco de dados.

DBTLSCAFile

O nome completo do caminho de um arquivo que contém os certificados da(s) CA(s) de nível superior para verificação do certificado do banco de dados.

Obrigatório: não (sim, se DBTLSConnect estiver definido como *verify_ca* ou *verify_full*)

DBTLSCertFile

O nome completo do caminho de um arquivo que contém o certificado do servidor Zabbix para autenticação no banco de dados.

DBTLSKeyFile

O nome completo do caminho de um arquivo que contém a chave privada para autenticação no banco de dados.

DBTLSCipher

A lista de cifras de criptografia que o Zabbix server permite para protocolos TLS até o TLS v1.2. Suportado apenas para MySQL.

DBTLSCipher13

A lista de conjuntos de cifras de criptografia que o Zabbix server permite para o protocolo TLS v1.3. Suportado apenas para MySQL, a partir da versão 8.0.16.

DebugLevel

Especifica o nível de debug:
0 - informações básicas sobre o início e parada dos processos do Zabbix
1 - informações críticas;
2 - informações de erro;
3 - avisos;
4 - para depuração (gera muitas informações);
5 - depuração estendida (gera ainda mais informações).
Veja também as opções de controle em tempo de execução.

Padrão: 3
 Intervalo: 0-5

EnableGlobalScripts

Habilita scripts globais no Zabbix server.
 Nota: A execução de scripts globais está habilitada por padrão. No entanto, para novas instalações a partir do Zabbix 7.0, EnableGlobalScripts é explicitamente definido como 0 (desabilitado).

Padrão: 1
 Valores: 0 - desabilitar; 1 - habilitar

ExportDir

O diretório para exportação em tempo real de eventos, histórico e tendências no formato JSON delimitado por nova linha. Se definido, ativa a exportação em tempo real.

ExportFileSize

O tamanho máximo por arquivo de exportação em bytes. Usado para rotação se `ExportDir` estiver definido.

Padrão: `1G`
 Intervalo: 1M-1G

ExportType

A lista de tipos de entidades separados por vírgula (events, history, trends) para exportação em tempo real (todos os tipos por padrão). Válido apenas se ExportDir estiver definido.
Observação: se ExportType for especificado, mas ExportDir não, isso é um erro de configuração e o server não será iniciado.

Exemplo para exportação de history e trends:

```
ExportType=history,trends
```

Exemplo para exportação apenas de events:

```
ExportType=events
```

ExternalScripts

O local dos scripts externos (depende da variável de instalação de tempo de compilação `datadir`).

Padrão: `/usr/local/share/zabbix/externalscripts`

Fping6Location

O local do fping6. Certifique-se de que o binário fping6 tenha propriedade de root e a flag SUID definida. Deixe vazio (”Fping6Location=”) se sua ferramenta fping for capaz de processar endereços IPv6.

Padrão: `/usr/sbin/fping6`

FpingLocation

O local do fping. Certifique-se de que o binário fping tenha propriedade root e o sinalizador SUID definido.

Padrão: `/usr/sbin/fping`

HANodeName

O nome do nó do cluster de alta disponibilidade. Quando vazio, o server está funcionando em modo standalone e um nó com nome vazio é criado.

HistoryCacheSize

O tamanho do cache de histórico, em bytes. O tamanho da memória compartilhada para armazenar dados de histórico.

Padrão: `16M`
 Intervalo: 128K-2G

HistoryIndexCacheSize

O tamanho do cache de índice de histórico, em bytes. O tamanho da memória compartilhada para indexar os dados de histórico armazenados no cache de histórico. O tamanho do cache de índice precisa de aproximadamente 100 bytes para armazenar um item.

Padrão: `4M`
 Intervalo: 128K-2G

HistoryStorageDateIndex

Habilita o pré-processamento de valores de histórico no armazenamento de histórico para armazenar valores em índices diferentes com base na data.

Padrão: `0`
 Valores: 0 - desativar; 1 - ativar

HistoryStorageURL

A URL HTTP[S] de armazenamento do histórico. Este parâmetro é usado para a configuração do Elasticsearch.

HistoryStorageTypes

Uma lista separada por vírgulas de tipos de valores a serem enviados para o armazenamento de histórico. Este parâmetro é usado para a configuração do Elasticsearch.

Padrão: `uint,dbl,str,log,text`

HousekeepingFrequency

Este parâmetro determina com que frequência o Zabbix executará o procedimento de housekeeping em horas. Housekeeping é a remoção de informações desatualizadas do banco de dados.
Nota: Para evitar que o housekeeper fique sobrecarregado (por exemplo, quando os períodos de histórico e tendências são muito reduzidos), não mais do que 4 vezes o número de horas definido em HousekeepingFrequency de informações desatualizadas são excluídas em um ciclo de housekeeping, para cada item. Assim, se HousekeepingFrequency for 1, não mais do que 4 horas de informações desatualizadas (começando pela entrada mais antiga) serão excluídas por ciclo.
Nota: Para reduzir a carga na inicialização do server, o housekeeping é adiado por 30 minutos após o início do server. Assim, se HousekeepingFrequency for 1, o primeiro procedimento de housekeeping após o início do server será executado após 30 minutos e será repetido com um atraso de uma hora a partir de então.
É possível desabilitar o housekeeping automático definindo HousekeepingFrequency como 0. Neste caso, o procedimento de housekeeping só pode ser iniciado pela opção de controle em tempo de execução *housekeeper_execute* e o período de informações desatualizadas excluídas em um ciclo de housekeeping é 4 vezes o período desde o último ciclo de housekeeping, mas não menos que 4 horas e não mais que 4 dias.
Veja também as opções de controle em tempo de execução.

Padrão: 1
 Intervalo: 0-24

Include

Você pode incluir arquivos individuais ou todos os arquivos em um diretório no arquivo de configuração. Para incluir apenas arquivos relevantes no diretório especificado, o caractere curinga asterisco é suportado para correspondência de padrões. Veja notas especiais sobre limitações.

Exemplo:

```
Include=/absolute/path/to/config/files/*.conf
```

JavaGateway

O endereço IP (ou nome do host) do Zabbix Java gateway. Necessário apenas se os pollers Java forem iniciados.

JavaGatewayPort

A porta em que o gateway Java do Zabbix escuta.

Padrão: 10052
 Intervalo: 1024-32767

ListenBacklog

O número máximo de conexões pendentes na fila TCP.
O valor padrão é uma constante codificada, que depende do sistema.
O valor máximo suportado depende do sistema; valores muito altos podem ser silenciosamente truncados para o ”máximo especificado pela implementação”.

Padrão: SOMAXCONN
 Intervalo: 0 - INT_MAX

ListenIP

Uma lista de endereços IP delimitados por vírgulas nos quais o trapper deve escutar.
O trapper escutará em todas as interfaces de rede se esse parâmetro estiver ausente.

Padrão: 0.0.0.0

ListenPort

A porta de escuta do trapper.

Padrão: 10051
 Intervalo: 1024-32767

LoadModule

O módulo a ser carregado na inicialização do server. Os módulos são usados para estender a funcionalidade do server. O módulo deve estar localizado no diretório especificado por LoadModulePath ou o caminho deve preceder o nome do módulo. Se o caminho anterior for absoluto (começar com ’/’) então LoadModulePath será ignorado.
Formatos:
LoadModule=<module.so>
LoadModule
permitido incluir vários parâmetros LoadModule.

LoadModulePath

O caminho completo para o local dos módulos do server. O padrão depende das opções de compilação.

LogFile

O nome do arquivo de registro.

Obrigatório: Sim, se LogType estiver definido como *file*; caso contrário, não

LogFileSize

O tamanho máximo do arquivo de registro em MB.
0 - desativa a rotação automática do registro.
Nota: Se o limite de tamanho do arquivo de log for atingido e a rotação de arquivos falhar, por qualquer motivo, o arquivo de log existente será truncado e iniciado novamente.

Padrão: 1
 Intervalo: 0-1024
 Obrigatório: Sim, se LogType estiver definido como *file*; caso contrário, não

LogSlowQueries

Determina o tempo que uma consulta ao banco de dados pode levar antes de ser registrada em milissegundos.
0 - não registrar consultas lentas.
Essa opção é ativada a partir de DebugLevel=3.

Padrão: 0
 Intervalo: 0-3600000

LogType

O tipo de saída de registro:
file - grava o registro no arquivo especificado pelo parâmetro LogFile;
system - grava o registro no syslog;
console - grava o registro na saída padrão.

Padrão: `file`

MaxConcurrentChecksPerPoller

O número máximo de verificações assíncronas que podem ser executadas de uma vez por cada poller HTTP agent, poller agent ou poller SNMP. Veja StartHTTPAgentPollers, StartAgentPollers e StartSNMPPollers.

Padrão: 1000
 Intervalo: 1-1000

MaxHousekeeperDelete

Não serão excluídas mais do que 'MaxHousekeeperDelete' linhas (correspondentes a [tablename], [field], [value]) por tarefa em um ciclo de housekeeping.
Se definido como 0, nenhum limite será usado. Neste caso, você deve saber o que está fazendo para não sobrecarregar o banco de dados.[2]
Este parâmetro se aplica apenas à exclusão de dados deixados para trás por items já excluídos.

Padrão: 5000
 Intervalo: 0-1000000

NodeAddress

IP ou nome do host com porta opcional para substituir como o frontend deve se conectar ao server.
Formato: <address>[:<port>]

Se o IP ou nome do host não for definido, o valor de ListenIP será usado. Se ListenIP não for definido, o valor `localhost` será usado.
Se a porta não for definida, o valor de ListenPort será usado. Se ListenPort não for definido, o valor 10051 será usado.

Esta opção pode ser substituída pelo endereço especificado na configuração do frontend.

Veja também: parâmetro HANodeName; Habilitando alta disponibilidade.

Padrão: `localhost:10051`

PidFile

Nome do arquivo PID.

Padrão: `/tmp/zabbix_server.pid`

ProblemHousekeepingFrequency

Determina a frequência com que o Zabbix excluirá os problemas de triggers excluídos, em segundos.

Padrão: 60
 Faixa: 1-3600

ProxyConfigFrequency

Determina com que frequência o Zabbix server envia dados de configuração para um Zabbix proxy em segundos. Usado apenas para proxies em modo passivo.

Padrão: 10
 Intervalo: 1-604800

ProxyDataFrequency

Determina com que frequência o Zabbix server solicita dados históricos de um Zabbix proxy em segundos. Usado apenas para proxies no modo passivo.

Padrão: 1
 Intervalo: 1-3600

ServiceManagerSyncFrequency

Determina a frequência com que o Zabbix sincronizará a configuração de um gerenciador de serviços em segundos.

Padrão: 60
 Faixa: 1-3600

SMSDevices

Uma lista de arquivos de modem separados por vírgulas permitidos para uso pelo Zabbix server. Este parâmetro é suportado desde o Zabbix 7.0.0.
O envio de SMS não é possível se este parâmetro não estiver definido.

Exemplo:

```
SMSDevices=/dev/ttyUSB0,/dev/ttyUSB1
```

SNMPTrapperFile

Arquivo temporário usado para passar dados do daemon de interceptação SNMP para o servidor.
Deve ser o mesmo que em zabbix_trap_receiver.pl ou no arquivo de configuração SNMPTT.

Padrão: `/tmp/zabbix_traps.tmp`

SocketDir

Diretório para armazenar os soquetes IPC usados pelos serviços internos do Zabbix.

Padrão: `/tmp`

SourceIP

Endereço IP de origem para:

- conexões de saída para o proxy Zabbix e o agent Zabbix;
- conexões sem agent (VMware, SSH, JMX, SNMP, Telnet e verificações simples);
- conexões do agente HTTP;
- solicitações HTTP JavaScript de item de script;
- solicitações HTTP JavaScript de pré-processamento;
- envio de e-mails de notificação (conexões com o servidor SMTP);
- notificações webhook (conexões HTTP JavaScript);
- conexões com o Vault

SSHKeyLocation

Localização das chaves públicas e privadas para verificações e ações de SSH.

SSLCertLocation

Localização dos arquivos de certificado do cliente SSL para autenticação do cliente.
Esse parâmetro é usado somente no monitoramento da Web.

SSLKeyLocation

Localização dos arquivos de chave privada SSL para autenticação do cliente.
Esse parâmetro é usado somente no monitoramento da Web.

SSLCALocation

Substitui o local dos arquivos da autoridade certificadora (CA) para verificação do certificado SSL do servidor. Se não definido, o diretório do sistema será usado.
Observe que o valor deste parâmetro será definido como a opção libcurl CURLOPT_CAPATH. Para versões do libcurl anteriores à 7.42.0, isso só terá efeito se o libcurl tiver sido compilado para usar o OpenSSL. Para mais informações, consulte a página da web do cURL.
Este parâmetro é usado no monitoramento web e na autenticação SMTP.

StartAgentPollers

O número de instâncias pré-forked dos pollers de agent do Zabbix. Veja MaxConcurrentChecksPerPoller.

Padrão: 1
 Intervalo: 0-1000

StartAlerters

O número de instâncias pré-forked de alerters.

Padrão: 3
 Intervalo: 1-100

StartBrowserPollers

O número de instâncias pré-forked de pollers de item de navegador.

Padrão: 1
 Intervalo: 0-1000

StartConnectors

O número de instâncias pré-forked de conector workers. O processo de gerenciamento de conector é iniciado automaticamente quando um conector worker é iniciado.

Padrão: 0
 Intervalo: 0-1000

StartDBSyncers

O número de instâncias pré-forked de history syncers.
Nota: Tenha cuidado ao alterar este valor, aumentá-lo pode causar mais problemas do que benefícios. Em termos gerais, o valor padrão deve ser suficiente para lidar com até 4000 NVPS.

Padrão: 4
 Intervalo: 1-100

StartDiscoverers

O número de instâncias pré-forked de discovery workers[1].

Padrão: 5
 Intervalo: 0-1000

StartEscalators

O número de instâncias pré-forked de escalators.

Padrão: 1
 Intervalo: 1-100

StartHistoryPollers

O número de instâncias pré-forked de history pollers.
Necessário apenas para verificações calculadas.

Padrão: 5
 Intervalo: 0-1000

StartHTTPAgentPollers

O número de instâncias pré-forked de HTTP agent pollers. Veja MaxConcurrentChecksPerPoller.

Padrão: 1
 Intervalo: 0-1000

StartHTTPPollers

O número de instâncias pré-forked de HTTP pollers[1].

Padrão: 1
 Intervalo: 0-1000

StartIPMIPollers

O número de instâncias pré-forked de IPMI pollers.

Padrão: 0
 Intervalo: 0-1000

StartJavaPollers

O número de instâncias pré-forked de Java pollers[1].

Padrão: 0
 Intervalo: 0-1000

StartLLDProcessors

O número de instâncias pré-criadas de workers de descoberta de baixo nível (LLD)[1].
O processo de gerenciamento de LLD é iniciado automaticamente quando um worker de LLD é iniciado.

Padrão: 2
 Intervalo: 1-100

StartODBCPollers

O número de instâncias pré-forked de ODBC pollers[1].

Padrão: 1
 Intervalo: 0-1000

StartPingers

O número de instâncias pré-criadas de ICMP pingers[1].

Padrão: 1
 Intervalo: 0-1000

StartPollersUnreachable

O número de instâncias pré-forked de pollers para hosts inalcançáveis (incluindo IPMI e Java)[1].
Pelo menos um poller para hosts inalcançáveis deve estar em execução se pollers regulares, IPMI ou Java forem iniciados.

Padrão: 1
 Intervalo: 0-1000

StartPollers

O número de instâncias pré-forked de pollers[1].

Padrão: 5
 Intervalo: 0-1000

StartPreprocessors

O número de instâncias pré-iniciadas de threads de worker de pré-processamento deve ser definido para não menos do que a contagem de núcleos de CPU disponíveis. Mais workers devem ser definidos se o pré-processamento não for limitado pela CPU e tiver muitas solicitações de rede.

Padrão: 16 (antes do Zabbix 7.0.6: 3)
 Intervalo: 1-1000

StartProxyPollers

O número de instâncias pré-forked de pollers para proxies passivos[1].

Padrão: 1
 Intervalo: 0-250

StartReportWriters

O número de instâncias pré-forked de report writers.
Se definido como 0, a geração de relatórios agendada é desabilitada.
O processo do gerenciador de relatórios é iniciado automaticamente quando um report writer é iniciado.

Padrão: 0
 Intervalo: 0-100

StartSNMPPollers

O número de instâncias pré-forked de SNMP pollers. Veja MaxConcurrentChecksPerPoller.

Padrão: 1
 Intervalo: 0-1000

StartSNMPTrapper

Se definido como 1, um processo SNMP trapper será iniciado.

Padrão: 0
 Intervalo: 0-1

StartTimers

O número de instâncias pré-forked de timers.
Timers processam períodos de manutenção.

Padrão: 1
 Intervalo: 1-1000

StartTrappers

O número de instâncias pré-criadas de trappers[1].
Trappers aceitam conexões recebidas do Zabbix sender, agents ativos e proxies ativos.

Padrão: 5
 Intervalo: 0-1000

StartVMwareCollectors

O número de instâncias pré-forked do coletor VMware.

Padrão: 0
 Intervalo: 0-250

StatsAllowedIP

Uma lista de endereços IP separados por vírgula, opcionalmente na notação CIDR, ou nomes DNS de instâncias Zabbix externas. A solicitação de estatísticas será aceita apenas dos endereços listados aqui. Se este parâmetro não for definido, nenhuma solicitação de estatísticas será aceita.
Se o suporte a IPv6 estiver habilitado, então '127.0.0.1', '::127.0.0.1', '::ffff:127.0.0.1' são tratados igualmente e '::/0' permitirá qualquer endereço IPv4 ou IPv6. '0.0.0.0/0' pode ser usado para permitir qualquer endereço IPv4.

Exemplo:

```
StatsAllowedIP=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com
```

Timeout

Especifica quanto tempo esperar (em segundos) para estabelecer conexão e trocar dados com o proxy, agent, serviço web do Zabbix e verificações SNMP legadas (número único de OID ou string).

Este parâmetro define a duração para várias operações de comunicação:

- execução de comandos remotos no agent Zabbix
- execução de comandos SSH/Telnet
- solicitações ao serviço web do Zabbix
- tempo limite de comunicação para solicitação de teste de tipo de mídia e o método `history.push`
- reagendamento de items quando uma interface IPMI se torna indisponível
- envio de resposta ao proxy Zabbix ao falhar na troca de dados devido a problemas de direitos ou criptografia
- prazo para sockets IPC assíncronos e opções de controle em tempo de execução
- conexões JMX
- obtenção de estatísticas de um proxy ou server Zabbix remoto
- envio de respostas para o frontend Zabbix

- solicitações DNS do poller assíncrono
- resposta para heartbeat de verificação ativa
- recuperação de dados (valores) do agent Zabbix de agents ativos
- recuperação de dados do Zabbix sender
- resposta quando o trapper falha ao processar uma solicitação
- envio da lista de verificações ativas para o agent Zabbix

Este timeout **não** será usado para aquelas verificações que possuem configurações de timeout flexível configuradas no frontend (em nível global, proxy ou por item). Por exemplo, items SNMP `walk[OID]` e `get[OID]` usam o timeout configurado no frontend; verificações SNMP legadas ainda usam o valor de timeout do server.

Padrão: 3
 Intervalo: 1-30

TLSCAFile

O nome completo do caminho de um arquivo que contém os certificados da(s) CA(s) de nível superior para verificação de certificados de pares, usados para comunicações criptografadas entre os componentes do Zabbix.

TLSCertFile

O nome completo do caminho de um arquivo que contém o certificado do servidor ou a cadeia de certificados, usado para comunicações criptografadas entre os componentes do Zabbix.

TLSCipherAll

A string de prioridade do GnuTLS ou a string de cifra do OpenSSL (TLS 1.2). Substitui os critérios padrão de seleção de ciphersuite para criptografia baseada em certificado e PSK.

Exemplo:

```
TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSCipherAll13

A string de cifra para OpenSSL 1.1.1 ou mais recente em TLS 1.3. Substitui os critérios padrão de seleção de ciphersuite para criptografia baseada em certificado e PSK.

Exemplo para GnuTLS:

```
NONE:+VERS-TLS1.2:+ECDHE-RSA:+RSA:+ECDHE-PSK:+PSK:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL
```

Exemplo para OpenSSL:

```
EECDH+aRSA+AES128:RSA+aRSA+AES128:kECDHEPSK+AES128:kPSK+AES128
```

TLSCipherCert

A string de prioridade do GnuTLS ou a string de cifra do OpenSSL (TLS 1.2). Substitui os critérios padrão de seleção de ciphersuite para criptografia baseada em certificado.

Exemplo para GnuTLS:

```
NONE:+VERS-TLS1.2:+ECDHE-RSA:+RSA:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL:+COMP-NULL:+SIG
```

Exemplo para OpenSSL:

```
EECDH+aRSA+AES128:RSA+aRSA+AES128
```

TLSCipherCert13

A string de cifra para OpenSSL 1.1.1 ou mais recente em TLS 1.3. Substitui os critérios padrão de seleção de ciphersuite para criptografia baseada em certificado.

TLSCipherPSK

A string de prioridade do GnuTLS ou a string de cifra do OpenSSL (TLS 1.2). Substitui os critérios padrão de seleção de ciphersuite para criptografia baseada em PSK.

Exemplo para GnuTLS:

```
NONE:+VERS-TLS1.2:+ECDHE-PSK:+PSK:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL:+COMP-NULL:+SIG
```

Exemplo para OpenSSL:

```
kECDHEPSK+AES128:kPSK+AES128
```

TLSCipherPSK13

A string de cifra para OpenSSL 1.1.1 ou mais recente em TLS 1.3. Substitui os critérios padrão de seleção de ciphersuite para criptografia baseada em PSK.

Exemplo:

```
TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSCRLFile

O caminho completo de um arquivo contendo certificados revogados. Este parâmetro é usado para comunicações criptografadas entre os componentes do Zabbix.

TLSKeyFile

O nome completo do caminho de um arquivo que contém a chave privada do servidor, usada para comunicações criptografadas entre os componentes do Zabbix.

TmpDir

O diretório temporário.

Padrão: `/tmp`

TrapperTimeout

Especifica o tempo limite em segundos para:

- recuperação de dados históricos do proxy Zabbix;
- envio de dados de configuração para o proxy Zabbix;
- execução de script global ou execução de comando remoto no server Zabbix.

Padrão: 300
 Intervalo: 1-300

TrendCacheSize

O tamanho do cache de tendências, em bytes.
O tamanho da memória compartilhada para armazenar dados de tendências.

Padrão: 4M
 Intervalo: 128K-2G

TrendFunctionCacheSize

O tamanho do cache da função de tendência, em bytes.
O tamanho da memória compartilhada para armazenar em cache os dados calculados da função de tendência.

Padrão: 4M
 Intervalo: 128K-2G

UnavailableDelay

Determina com que frequência o host é verificado quanto à disponibilidade durante o período de indisponibilidade, em segundos.

Padrão: 60
 Intervalo: 1-3600

UnreachableDelay

Determina com que frequência o host é verificado quanto à disponibilidade durante o período de inacessibilidade, em segundos.

Padrão: 15
 Intervalo: 1-3600

UnreachablePeriod

Determina após quantos segundos de inacessibilidade trata um host como indisponível.

Padrão: 45
 Intervalo: 1-3600

Usuário

Reduz os privilégios de um usuário específico existente no sistema.
Só terá efeito se for executado como ”root” e se AllowRoot estiver desativado.

Padrão: `zabbix`

ValueCacheSize

O tamanho do cache de valores do histórico, em bytes.
O tamanho da memória compartilhada para armazenar em cache as solicitações de dados do histórico do item.
A configuração como 0 desativa o cache de valores (não recomendado).
Quando o cache de valores esgota a memória compartilhada, uma mensagem de aviso é gravada no log do servidor a cada 5 minutos.

Padrão: 8M
 Intervalo: 0,128K-64G

Cofre

Especifica o provedor do cofre:
HashiCorp - HashiCorp KV Secrets Engine versão 2
CyberArk - CyberArk Central Credential Provider
Deve corresponder ao provedor do cofre definido no frontend.

Padrão: `HashiCorp`

VaultDBPath

Caminho ou consulta do Vault, dependendo do Vault, de onde as credenciais para o banco de dados serão recuperadas pelas chaves.

As chaves usadas para HashiCorp são 'password' e 'username'.

Exemplo de caminho com VaultPrefix=/v1/secret/data/zabbix/:

```
database
```

Exemplo de caminho sem VaultPrefix:

```
secret/zabbix/database
```

As chaves usadas para CyberArk são 'Content' e 'UserName'.

Exemplo:

```
AppID=zabbix_server&Query=Safe=passwordSafe;Object=zabbix_server_database
```

Esta opção só pode ser usada se DBUser e DBPassword não estiverem especificados.

VaultPrefix

Um prefixo personalizado para o caminho ou consulta do Vault, dependendo do Vault. Os padrões mais adequados serão usados se não especificado.
Observe que 'data' é automaticamente anexado após o ponto de montagem para HashiCorp se VaultPrefix não for especificado.

Exemplo de prefixo para Hashicorp:

```
v1/secret/data/zabbix/
```

Exemplo de prefixo para Cyberark:

```
/AIMWebService/api/Accounts?
```

VaultTLSCertFile

O nome do arquivo de certificado SSL usado para autenticação do cliente.
 O arquivo de certificado deve estar no formato PEM1.
 Se o arquivo de certificado também contiver a chave privada, deixe o campo do arquivo de chave SSL vazio.
 O diretório que contém este arquivo é especificado pelo parâmetro de configuração SSLCertLocation.
Esta opção pode ser omitida, mas é recomendada para o cofre CyberArkCCP.

VaultTLSKeyFile

O nome do arquivo de chave privada SSL usado para autenticação do cliente.
 O arquivo de chave privada deve estar no formato PEM1.
 O diretório que contém esse arquivo é especificado pelo parâmetro de configuração SSLKeyLocation.
Essa opção pode ser omitida, mas é recomendada para o cofre do CyberArkCCP.

VaultToken

O token de autenticação do HashiCorp Vault que deveria ter sido gerado exclusivamente para o servidor Zabbix com permissão somente de leitura para os caminhos especificados em Vault macros e permissão somente de leitura para o caminho especificado no parâmetro de configuração opcional VaultDBPath.
É um erro se VaultToken e a variável de ambiente VAULT_TOKEN forem definidas ao mesmo tempo.

Obrigatório: Sim, se o Vault estiver definido como *HashiCorp*; caso contrário, não

VaultURL

A URL HTTP[S] do servidor Vault. O diretório de certificados CA do sistema será usado se SSLCALocation não for especificado.

Padrão: `https://127.0.0.1:8200`

VMwareCacheSize

O tamanho da memória compartilhada para armazenar dados do VMware.
Uma verificação interna do VMware zabbix[vmware,buffer,...] pode ser usada para monitorar o uso do cache do VMware (consulte Internal checks).
Observe que a memória compartilhada não é alocada se não houver instâncias do coletor do VMware configuradas para iniciar.

Padrão: `8M`
 Intervalo: 256K-2G

VMwareFrequency

O atraso, em segundos, entre a coleta de dados de um único serviço da VMware.
Esse atraso deve ser definido como o menor intervalo de atualização de qualquer item de monitoramento da VMware.

Padrão: 60
 Intervalo: 10-86400

VMwarePerfFrequency

O atraso em segundos entre a recuperação das estatísticas do contador de desempenho de um único serviço VMware. Esse atraso deve ser definido para o menor intervalo de atualização de qualquer item de monitoramento VMware que use contadores de desempenho VMware.

Padrão: 60
 Intervalo: 10-86400

VMwareTimeout

O número máximo de segundos que um coletor vmware aguardará por uma resposta do serviço VMware (hipervisor vCenter ou ESX).

Padrão: 10
 Intervalo: 1-300

WebServiceURL

A URL HTTP[S] para o serviço web do Zabbix no formato `http[s]://host:port/report`.

Exemplo:

```
WebServiceURL=http://localhost:10053/report
```

Observação: o esquema (`http://`) pode ser omitido apenas para conexões sem TLS (HTTP); se o TLS estiver configurado, `https://` deve ser usado.

WebDriverURL

URL HTTP[S] da interface WebDriver.

Exemplo (usado com o Selenium WebDriver standalone server):

```
WebDriverURL=http://localhost:4444
```

Notas de rodapé

[1] Observe que muitos processos de coleta de dados (pollers, unreachable pollers, ODBC pollers, HTTP pollers, Java pollers, pingers, trappers, proxypollers) juntamente com o IPMI manager, SNMP trapper, preprocessing workers e discovery workers podem esgotar o limite de descritores de arquivos por processo para o preprocessing manager.

Warning:
Esgotar o limite de descritores de arquivos fará com que o Zabbix server pare, normalmente logo após a inicialização, mas às vezes pode demorar mais. Para evitar esses problemas, revise o arquivo de configuração do Zabbix server para otimizar o número de verificações e processos simultâneos. Além disso, se necessário, certifique-se de que o limite de descritores de arquivos esteja definido suficientemente alto, verificando e ajustando os limites do sistema.

[2] Quando muitos items são excluídos, isso aumenta a carga no banco de dados, porque o housekeeper precisará remover todos os dados de histórico que esses items possuíam. Por exemplo, se precisarmos remover apenas 1 protótipo de item do template, mas esse template está vinculado a 50 hosts e para cada host o protótipo é expandido para 100 items reais, 5000 items no total terão que ser removidos (1*50*100). Se 500 for definido para MaxHousekeeperDelete (MaxHousekeeperDelete=500), o processo housekeeper terá que remover até 2500000 valores (5000*500) dos items excluídos das tabelas de histórico e tendências em um ciclo.

[3] DBSocket e DBPort são mutuamente exclusivos na configuração do server. Especifique apenas um ou deixe ambos indefinidos.

2 Zabbix proxy

Visão geral

Os parâmetros suportados pelo arquivo de configuração do Zabbix proxy (zabbix_proxy.conf) estão listados nesta seção.

Os parâmetros são listados sem informações adicionais. Clique no parâmetro para ver os detalhes completos.

Parâmetro	Descrição
AllowRoot	Permite que o proxy seja executado como 'root'.

Parâmetro	Descrição
AllowUnsupportedDBVersions	Permite que o proxy funcione com versões de banco de dados não suportadas.
CacheSize	O tamanho do cache de configuração.
ConfigFrequency	Este parâmetro está **obsoleto** (use ProxyConfigFrequency em vez disso). Com que frequência o proxy recupera dados de configuração do Zabbix server em segundos.
DataSenderFrequency	O proxy enviará os dados coletados para o server a cada N segundos.
DBHost	O nome do host do banco de dados.
DBName	O nome do banco de dados ou o caminho para o arquivo do banco de dados para SQLite3.
DBPassword	A senha do banco de dados.
DBPort	A porta do banco de dados quando não estiver usando o socket local.
DBSchema	O nome do schema do banco de dados. Usado para PostgreSQL.
DBSocket	O caminho para o arquivo de socket do MySQL.
DBUser	O usuário do banco de dados.
DBTLSConnect	Definir esta opção para o valor especificado força o uso de uma conexão TLS com o banco de dados.
DBTLSCAFile	O caminho completo de um arquivo contendo os certificados da(s) CA(s) de nível superior para verificação do certificado do banco de dados.
DBTLSCertFile	O caminho completo de um arquivo contendo o certificado do Zabbix proxy para autenticação no banco de dados.
DBTLSKeyFile	O caminho completo de um arquivo contendo a chave privada para autenticação no banco de dados.
DBTLSCipher	A lista de cifras de criptografia que o Zabbix proxy permite para protocolos TLS até o TLS v1.2. Suportado apenas para MySQL.
DBTLSCipher13	A lista de cifras de criptografia que o Zabbix proxy permite para o protocolo TLS v1.3. Suportado apenas para MySQL, a partir da versão 8.0.16.
DebugLevel	O nível de debug.
EnableRemoteCommands	Se comandos remotos do Zabbix server são permitidos.
ExternalScripts	O local dos scripts externos.
Fping6Location	O local do fping6.
FpingLocation	O local do fping.
HistoryCacheSize	O tamanho do cache de histórico.
HistoryIndexCacheSize	O tamanho do cache de índice de histórico.
Hostname	Um nome de proxy exclusivo e sensível a maiúsculas e minúsculas.
HostnameItem	O item usado para definir o Hostname se ele estiver indefinido.
HousekeepingFrequency	Com que frequência o Zabbix executará o procedimento de housekeeping em horas.
Include	Você pode incluir arquivos individuais ou todos os arquivos em um diretório no arquivo de configuração.
JavaGateway	O endereço IP (ou nome do host) do Zabbix Java gateway.
JavaGatewayPort	A porta que o Zabbix Java gateway escuta.
ListenBacklog	O número máximo de conexões pendentes na fila TCP.
ListenIP	Uma lista de endereços IP separados por vírgula nos quais o trapper deve escutar.
ListenPort	A porta de escuta para o trapper.
LoadModule	O módulo a ser carregado na inicialização do proxy.
LoadModulePath	O caminho completo para o local dos módulos do proxy.
LogFile	O nome do arquivo de log.
LogFileSize	O tamanho máximo do arquivo de log.
LogRemoteCommands	Habilita o registro de comandos shell executados como avisos.
LogSlowQueries	Por quanto tempo uma consulta ao banco de dados pode demorar antes de ser registrada.
LogType	O tipo de saída do log.
MaxConcurrentChecksPerPoller	O número máximo de verificações assíncronas que podem ser executadas de uma vez por cada HTTP agent poller, agent poller ou SNMP poller.
PidFile	O nome do arquivo PID.
ProxyBufferMode	Especifica o mecanismo de armazenamento de dados de histórico, descoberta e autoregistro (disco/memória/híbrido).
ProxyConfigFrequency	Com que frequência o proxy recupera dados de configuração do Zabbix server em segundos.
ProxyLocalBuffer	O proxy manterá os dados localmente por N horas, mesmo que os dados já tenham sido sincronizados com o server.
ProxyMemoryBufferAge	A idade máxima dos dados no buffer de memória do proxy em segundos.
ProxyMemoryBufferSize	O tamanho do cache de memória compartilhada para dados de histórico, descoberta e autoregistro coletados.
ProxyMode	O modo de operação do proxy (ativo/passivo).
ProxyOfflineBuffer	O proxy manterá os dados por N horas em caso de falta de conectividade com o Zabbix server.

Parâmetro	Descrição
Server	Se ProxyMode estiver definido como modo ativo: endereço IP ou nome DNS do Zabbix server (endereço:porta) ou cluster (endereço:porta;endereço2:porta) para obter dados de configuração e enviar dados. Se ProxyMode estiver definido como modo passivo: lista de endereços IP separados por vírgula, opcionalmente na notação CIDR, ou nomes DNS do Zabbix server.
SNMPTrapperFile	O arquivo temporário usado para passar dados do daemon SNMP trap para o proxy.
SocketDir	O diretório para armazenar os sockets IPC usados pelos serviços internos do Zabbix.
SourceIP	O endereço IP de origem.
SSHKeyLocation	O local das chaves públicas e privadas para verificações e ações SSH.
SSLCertLocation	O local dos arquivos de certificado do cliente SSL para autenticação do cliente.
SSLKeyLocation	O local dos arquivos de chave privada SSL para autenticação do cliente.
SSLCALocation	Substitui o local dos arquivos da autoridade certificadora (CA) para verificação do certificado do server SSL.
StartAgentPollers	O número de instâncias pré-iniciadas de agent pollers assíncronos do Zabbix.
StartBrowserPollers	O número de instâncias pré-iniciadas de browser item pollers.
StartDBSyncers	O número de instâncias pré-iniciadas de history syncers.
StartDiscoverers	O número de instâncias pré-iniciadas de discovery workers.
StartHTTPAgentPollers	O número de instâncias pré-iniciadas de HTTP agent pollers assíncronos.
StartHTTPPollers	O número de instâncias pré-iniciadas de HTTP pollers.
StartIPMIPollers	O número de instâncias pré-iniciadas de IPMI pollers.
StartJavaPollers	O número de instâncias pré-iniciadas de Java pollers.
StartODBCPollers	O número de instâncias pré-iniciadas de ODBC pollers.
StartPingers	O número de instâncias pré-iniciadas de ICMP pingers.
StartPollersUnreachable	O número de instâncias pré-iniciadas de pollers para hosts inacessíveis (incluindo IPMI e Java).
StartPollers	O número de instâncias pré-iniciadas de pollers.
StartPreprocessors	O número de instâncias pré-iniciadas de preprocessing workers.
StartSNMPPollers	O número de instâncias pré-iniciadas de SNMP pollers assíncronos.
StartSNMPTrapper	Se definido como 1, um processo SNMP trapper será iniciado.
StartTrappers	O número de instâncias pré-iniciadas de trappers.
StartVMwareCollectors	O número de instâncias pré-iniciadas de VMware collector.
StatsAllowedIP	Uma lista de endereços IP separados por vírgula, opcionalmente na notação CIDR, ou nomes DNS de instâncias externas do Zabbix. A solicitação de estatísticas será aceita apenas dos endereços listados aqui.
Timeout	Especifica por quanto tempo aguardar (em segundos) para estabelecer conexão e trocar dados com o Zabbix proxy, agent, serviço web e para verificações SNMP (exceto itens SNMP `walk[OID]` e `get[OID]`).
TLSAccept	Quais conexões de entrada aceitar do Zabbix server.
TLSCAFile	O caminho completo de um arquivo contendo os certificados da(s) CA(s) de nível superior para verificação do certificado do peer, usado para comunicações criptografadas entre componentes do Zabbix.
TLSCertFile	O caminho completo de um arquivo contendo o certificado do server ou cadeia de certificados, usado para comunicações criptografadas entre componentes do Zabbix.
TLSCipherAll	A string de prioridade do GnuTLS ou a string de cifra do OpenSSL (TLS 1.2). Substitui os critérios padrão de seleção de ciphersuite para criptografia baseada em certificado e PSK.
TLSCipherAll13	A string de cifra para OpenSSL 1.1.1 ou mais recente em TLS 1.3. Substitui os critérios padrão de seleção de ciphersuite para criptografia baseada em certificado e PSK.
TLSCipherCert	A string de prioridade do GnuTLS ou a string de cifra do OpenSSL (TLS 1.2). Substitui os critérios padrão de seleção de ciphersuite para criptografia baseada em certificado.
TLSCipherCert13	A string de cifra para OpenSSL 1.1.1 ou mais recente em TLS 1.3. Substitui os critérios padrão de seleção de ciphersuite para criptografia baseada em certificado.
TLSCipherPSK	A string de prioridade do GnuTLS ou a string de cifra do OpenSSL (TLS 1.2). Substitui os critérios padrão de seleção de ciphersuite para criptografia baseada em PSK.
TLSCipherPSK13	A string de cifra para OpenSSL 1.1.1 ou mais recente em TLS 1.3. Substitui os critérios padrão de seleção de ciphersuite para criptografia baseada em PSK.
TLSConnect	Como o proxy deve se conectar ao Zabbix server.
TLSCRLFile	O caminho completo de um arquivo contendo certificados revogados. Este parâmetro é usado para comunicações criptografadas entre componentes do Zabbix.
TLSKeyFile	O caminho completo de um arquivo contendo a chave privada do proxy, usado para comunicações criptografadas entre componentes do Zabbix.
TLSPSKFile	O caminho completo de um arquivo contendo a chave pré-compartilhada do proxy, usado para comunicações criptografadas com o Zabbix server.

Parâmetro	Descrição
TLSPSKIdentity	A string de identidade da chave pré-compartilhada, usada para comunicações criptografadas com o Zabbix server.
TLSServerCertIssuer	O emissor do certificado do server permitido.
TLSServerCertSubject	O assunto do certificado do server permitido.
TmpDir	O diretório temporário.
TrapperTimeout	Especifica o tempo limite em segundos para: - recuperação de dados de configuração do Zabbix server; - execução de script global ou execução de comando remoto.
UnavailableDelay	Com que frequência um host é verificado quanto à disponibilidade durante o período de indisponibilidade.
UnreachableDelay	Com que frequência um host é verificado quanto à disponibilidade durante o período de inacessibilidade.
UnreachablePeriod	Após quantos segundos de inacessibilidade tratar o host como indisponível.
User	Reduz privilégios para um usuário específico e existente no sistema.
Vault	O provedor de vault.
VaultDBPath	O local de onde as credenciais do banco de dados devem ser recuperadas por chaves.
VaultPrefix	Prefixo personalizado para o caminho ou consulta do vault.
VaultTLSCertFile	O nome do arquivo de certificado SSL usado para autenticação do cliente.
VaultTLSKeyFile	O nome do arquivo de chave privada SSL usado para autenticação do cliente.
VaultToken	O token de autenticação do HashiCorp vault.
VaultURL	A URL HTTP[S] do vault server.
VMwareCacheSize	O tamanho da memória compartilhada para armazenar dados do VMware.
VMwareFrequency	O atraso em segundos entre a coleta de dados de um único serviço VMware.
VMwarePerfFrequency	O atraso em segundos entre a recuperação de estatísticas do contador de desempenho de um único serviço VMware.
VMwareTimeout	O número máximo de segundos que um vmware collector aguardará por uma resposta do serviço VMware.
WebDriverURL	A URL HTTP[S] da interface WebDriver.

Todos os parâmetros não são obrigatórios, a menos que seja declarado explicitamente que o parâmetro é obrigatório.

Observe que:

- Os valores padrão refletem os padrões do daemon, não os valores nos arquivos de configuração fornecidos.
- O Zabbix suporta arquivos de configuração apenas na codificação UTF-8 sem BOM.
- Comentários iniciados com ”#” são suportados apenas no início da linha.

Detalhes do parâmetro

AllowRoot

Permite que o proxy seja executado como ’root’. Se desabilitado e o proxy for iniciado por ’root’, o proxy tentará alternar para o usuário ’zabbix’. Não tem efeito se iniciado sob um usuário comum.

Padrão: 0
 Valores: 0 - não permitir; 1 - permitir

AllowUnsupportedDBVersions

Permite que o proxy trabalhe com versões de banco de dados sem suporte.

Padrão: 0
 Valores: 0 - não permitir; 1 - permitir

CacheSize

O tamanho do cache de configuração, em bytes. O tamanho da memória compartilhada para armazenar dados de host e item.

Padrão: 32M
 Intervalo: 128K-64G

ConfigFrequency

Este parâmetro está **obsoleto** (use ProxyConfigFrequency em seu lugar).
Com que frequência o proxy recupera os dados de configuração do servidor Zabbix em segundos.
Parâmetro de proxy ativo. Ignorado para proxies passivos (veja o parâmetro ProxyMode).

Padrão: 3600
 Intervalo: 1-604800

DataSenderFrequency

O proxy enviará os dados coletados para o server a cada N segundos. Observe que um proxy ativo ainda consultará o server Zabbix a cada segundo para tarefas de comando remoto.
Parâmetro de proxy ativo. Ignorado para proxies passivos (veja o parâmetro ProxyMode).

Padrão: 1
 Intervalo: 1-3600

DBHost

O nome do host do banco de dados.
Com MySQL, `localhost` ou uma string vazia resulta no uso de um socket. Com PostgreSQL, apenas uma string vazia resulta em uma tentativa de usar o socket. Com Oracle, uma string vazia resulta no uso do método de conexão Net Service Name; nesse caso, considere usar a variável de ambiente TNS_ADMIN para especificar o diretório do arquivo tnsnames.ora.

Padrão: `localhost`

DBName

O nome do banco de dados ou o caminho para o arquivo de banco de dados para SQLite3 (a arquitetura multi-processo do Zabbix não permite o uso de [in-memory database] (https://www.sqlite.org/inmemorydb.html), por exemplo, `:memory:`, `file::memory:?cache=shared` ou `file:memdb1?mode=memory&cache=shared`).
Warning: Não tente usar o mesmo banco de dados que o servidor Zabbix está usando.
Com o Oracle, se o método de conexão Net Service Name for usado, especifique o nome do serviço em tnsnames.ora ou defina-o como uma cadeia de caracteres vazia; defina a variável de ambiente TWO_TASK se DBName for definido como uma cadeia de caracteres vazia.

Obrigatório: Sim

DBPassword

A senha do banco de dados. Comente esta linha se nenhuma senha for usada. Ignorado para SQLite.

DBPort

A porta do banco de dados quando não estiver usando o socket local.[1] Ignorado para SQLite.
Com Oracle, se o método de conexão Net Service Name for usado, este parâmetro será ignorado; o número da porta do arquivo tnsnames.ora será usado em vez disso.

Padrão para MySQL: 3306

Padrão para PostgreSQL: 5432

Intervalo: 1024-65535

DBSchema

O nome do schema do banco de dados. Usado para PostgreSQL.

DBSocket

O caminho para o arquivo de socket do MySQL.[1]
A porta do banco de dados quando não estiver usando o socket local. Ignorado para SQLite.

Padrão: 3306

DBUser

O usuário do banco de dados. Ignorado para SQLite.

DBTLSConnect

A configuração dessa opção força o uso da conexão TLS com o banco de dados:
required - conecta-se usando TLS
verify_ca - conecta-se usando TLS e verifica o certificado
verify_full - conecta-se usando TLS, verifica o certificado e verifica se a identidade do banco de dados especificada pelo DBHost corresponde ao seu certificado
No MySQL a partir da versão 5.7.11 e no PostgreSQL, há suporte para os seguintes valores: ”required”, ”verify”, ”verify_full”.
No MariaDB, a partir da versão 10.2.6, há suporte para os valores ”required” e ”verify_full”.
Por padrão, nenhuma opção é definida e o comportamento depende da configuração do banco de dados.

DBTLSCAFile

O nome completo do caminho de um arquivo que contém os certificados da(s) CA(s) de nível superior para verificação do certificado do banco de dados.

Obrigatório: não (sim, se DBTLSConnect estiver definido como *verify_ca* ou *verify_full*)

DBTLSCertFile

O nome completo do caminho de um arquivo que contém o certificado de proxy do Zabbix para autenticação no banco de dados.

DBTLSKeyFile

O nome completo do caminho de um arquivo que contém a chave privada para autenticação no banco de dados.

DBTLSCipher

A lista de cifras de criptografia que o Zabbix proxy permite para protocolos TLS até o TLS v1.2. Suportado apenas para MySQL.

DBTLSCipher13

A lista de conjuntos de cifras de criptografia que o Zabbix proxy permite para o protocolo TLS v1.3. Suportado apenas para MySQL, a partir da versão 8.0.16.

DebugLevel

Especifique o nível de depuração:
0 - informações básicas sobre a inicialização e a interrupção dos processos do Zabbix
1 - informações críticas;
2 - informações sobre erros;
3 - avisos;
4 - para depuração (produz muitas informações);
5 - depuração estendida (produz ainda mais informações).

Padrão: 3
 Intervalo: 0-5

EnableRemoteCommands

Se os comandos remotos do servidor Zabbix são permitidos.

Padrão: 0
 Valores: 0 - não permitido; 1 - permitido

ExternalScripts

O local dos scripts externos (depende da variável de instalação de tempo de compilação `datadir`).

Padrão: `/usr/local/share/zabbix/externalscripts`

Fping6Location

O local do fping6. Certifique-se de que o binário fping6 tenha propriedade root e o sinalizador SUID definido. Deixe vazio (”Fping6Location=”) se o seu utilitário fping for capaz de processar endereços IPv6.

Padrão: `/usr/sbin/fping6`

FpingLocation

O local do fping. Certifique-se de que o binário fping tenha propriedade root e a flag SUID definida.

Padrão: `/usr/sbin/fping`

HistoryCacheSize

O tamanho do cache de histórico, em bytes. O tamanho da memória compartilhada para armazenar dados de histórico.

Padrão: `16M`
 Intervalo: 128K-2G

HistoryIndexCacheSize

O tamanho do cache de índice de histórico, em bytes. O tamanho da memória compartilhada para indexar os dados de histórico armazenados no cache de histórico. O tamanho do cache de índice precisa de aproximadamente 100 bytes para armazenar em cache um item.

Padrão: `4M`
 Intervalo: 128K-2G

Hostname

Um nome de proxy exclusivo e sensível a maiúsculas e minúsculas. Certifique-se de que o nome do proxy seja conhecido pelo servidor.
Caracteres permitidos: alfanuméricos, ’.’, ’ ’, ’_’ e ’-’. Comprimento máximo: 128

Padrão: Definido por HostnameItem

HostnameItem

O item usado para definir o Hostname se ele não estiver definido (isso será executado no proxy da mesma forma que em um agent). Ignorado se Hostname estiver definido.
Não suporta UserParameters, contadores de desempenho ou aliases, mas suporta system.run[].

Padrão: system.hostname

HousekeepingFrequency

Com que frequência o Zabbix executará o procedimento de housekeeping (em horas). Housekeeping é a remoção de informações desatualizadas do banco de dados.
Nota: Para reduzir a carga no início do proxy, o housekeeping é adiado por 30 minutos após o início do proxy. Assim, se HousekeepingFrequency for 1, o primeiro procedimento de housekeeping após o início do proxy

será executado após 30 minutos e será repetido a cada hora depois disso.
É possível desabilitar o housekeeping automático definindo HousekeepingFrequency como 0. Neste caso, o procedimento de housekeeping só pode ser iniciado pela opção de controle em tempo de execução *housekeeper_execute*.
Veja também as opções de controle em tempo de execução.

Padrão: 1
 Intervalo: 0-24

Include

Você pode incluir arquivos individuais ou todos os arquivos em um diretório no arquivo de configuração.
Para incluir apenas arquivos relevantes no diretório especificado, o caractere curinga asterisco é suportado para correspondência de padrões.
Veja notas especiais sobre limitações.

Exemplo:

```
Include=/absolute/path/to/config/files/*.conf
```

JavaGateway

O endereço IP (ou nome do host) do Zabbix Java gateway. Só é necessário se os pollers Java forem iniciados.

JavaGatewayPort

A porta em que o gateway Java do Zabbix escuta.

Padrão: 10052
 Intervalo: 1024-32767

ListenBacklog

O número máximo de conexões pendentes na fila TCP.
O valor padrão é uma constante codificada, que depende do sistema.
O valor máximo suportado depende do sistema; valores muito altos podem ser silenciosamente truncados para o ”máximo especificado pela implementação”.

Padrão: SOMAXCONN
 Intervalo: 0 - INT_MAX

ListenIP

Uma lista de endereços IP delimitados por vírgulas nos quais o trapper deve escutar.
O trapper escutará em todas as interfaces de rede se esse parâmetro estiver ausente.

Padrão: 0.0.0.0

ListenPort

A porta de escuta do trapper.

Padrão: 10051
 Intervalo: 1024-32767

LoadModule

O módulo a ser carregado na inicialização do proxy. Os módulos são usados para estender a funcionalidade do proxy. O módulo deve estar localizado no diretório especificado por LoadModulePath ou o caminho deve preceder o nome do módulo. Se o caminho anterior for absoluto (começar com ’/’) então LoadModulePath será ignorado.
Formatos:
LoadModule=<module.so>
LoadModule permitido incluir vários parâmetros LoadModule.

LoadModulePath

O caminho completo para o local dos módulos do proxy. O padrão depende das opções de compilação.

LogFile

O nome do arquivo de registro.

Obrigatório: Sim, se LogType estiver definido como *file*; caso contrário, não

LogFileSize

O tamanho máximo de um arquivo de log em MB.
0 - desativa a rotação automática de logs.
Nota: Se o limite de tamanho do arquivo de log for atingido e a rotação do arquivo falhar, por qualquer motivo, o arquivo de log existente será truncado e iniciado novamente.

Padrão: 1
 Intervalo: 0-1024

LogRemoteCommands

Ative o registro em log dos comandos do shell executados como avisos.

Padrão: 0
 Valores: 0 - desativado, 1 - ativado

LogType

O tipo da saída de registro:
file - grava o registro no arquivo especificado pelo parâmetro LogFile;
system - grava o registro no syslog;
console - grava o registro na saída padrão.

Padrão: `file`

LogSlowQueries

Quanto tempo uma consulta ao banco de dados pode levar antes de ser registrada (em milissegundos).
0 - não registrar consultas lentas.
Essa opção é ativada a partir de DebugLevel=3.

Padrão: 0
 Intervalo: 0-3600000

MaxConcurrentChecksPerPoller

O número máximo de verificações assíncronas que podem ser executadas de uma vez por cada poller HTTP agent, agent poller ou SNMP poller. Veja StartHTTPAgentPollers, StartAgentPollers e StartSNMPPollers.

Padrão: 1000
 Intervalo: 1-1000

PidFile

O nome do arquivo PID.

Padrão: `/tmp/zabbix_proxy.pid`

ProxyBufferMode

Especifica o mecanismo de armazenamento de dados de histórico, descoberta de rede e auto-registro: *disk* - os dados são armazenados no banco de dados e enviados a partir do banco de dados; *memory* - os dados são armazenados na memória e enviados a partir da memória. Se o buffer ficar sem memória, os dados antigos serão descartados. Ao desligar, o buffer é descartado. *hybrid* - o buffer do proxy normalmente funciona como no modo de memória até que fique sem memória ou o registro mais antigo exceda a idade configurada. Se isso acontecer, o buffer é gravado no banco de dados e funciona como no modo de disco até que todos os dados tenham sido enviados e ele volte a funcionar com a memória novamente. Ao desligar, o buffer de memória é gravado no banco de dados.

Veja também: Buffer de memória do proxy.

Padrão: `disk`
 Valores: disk; memory; hybrid

ProxyConfigFrequency

Com que frequência o proxy recupera os dados de configuração do Zabbix server em segundos.
Parâmetro de proxy ativo. Ignorado para proxies passivos (veja o parâmetro ProxyMode).

Padrão: 10
 Intervalo: 1-604800

ProxyLocalBuffer

O proxy manterá os dados localmente por N horas, mesmo que os dados já tenham sido sincronizados com o servidor.
Esse parâmetro pode ser usado se os dados locais forem usados por aplicativos de terceiros.

Padrão: 0
 Faixa: 0-720

ProxyMemoryBufferAge

A idade máxima dos dados no buffer de memória do proxy, em segundos. Quando ativado (diferente de zero) e os registros no buffer de memória do proxy são mais antigos, isso força o buffer do proxy a mudar para o modo de banco de dados até que todos os registros sejam enviados para o server. Este parâmetro deve ser menor ou igual ao parâmetro ProxyOfflineBuffer.

Padrão: 0
 Intervalo: 0;600-864000

ProxyMemoryBufferSize

O tamanho do cache de memória compartilhada para dados de histórico, descoberta e autorregistro coletados, em bytes. Se ativado (diferente de zero), o proxy manterá os dados de histórico, descoberta e autorregistro na memória, a menos que o cache esteja cheio ou os registros armazenados sejam mais antigos do que o definido em ProxyMemoryBufferAge. Este parâmetro não pode ser usado junto com o parâmetro ProxyLocalBuffer.

Padrão: 0
 Intervalo: 0;128K-2G

ProxyMode

O modo de operação do proxy.
0 - proxy no modo ativo
1 - proxy no modo passivo
Observe que dados de configuração (sensíveis) do proxy podem ficar disponíveis para partes que tenham acesso à porta trapper do Zabbix server ao usar um proxy ativo. Isso é possível porque qualquer pessoa pode se passar por um proxy ativo e solicitar dados de configuração; a autenticação não ocorre.

Padrão: 0
 Intervalo: 0-1

ProxyOfflineBuffer

O proxy manterá os dados por N horas em caso de falta de conectividade com o servidor Zabbix.
Os dados mais antigos serão perdidos.

Padrão: 1
 Faixa: 1-720

Server

Se ProxyMode estiver definido como *modo ativo*:
Endereço IP ou nome DNS do Zabbix server (endereço:porta) ou cluster (endereço:porta;endereço2:porta) para obter dados de configuração e enviar dados.
Se a porta não for especificada, a porta padrão será usada.
Os nós do cluster devem ser separados por ponto e vírgula.

Se ProxyMode estiver definido como *modo passivo*:
Lista de endereços IP separados por vírgula, opcionalmente na notação CIDR, ou nomes DNS do Zabbix server. As conexões recebidas serão aceitas apenas dos endereços listados aqui. Se o suporte a IPv6 estiver habilitado, então '127.0.0.1', '::127.0.0.1', '::ffff:127.0.0.1' são tratados igualmente.
'::/0' permitirá qualquer endereço IPv4 ou IPv6. '0.0.0.0/0' pode ser usado para permitir qualquer endereço IPv4.

Exemplo:

```
Server=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com
```

Obrigatório: sim

SNMPTrapperFile

Um arquivo temporário usado para passar dados do daemon de interceptação SNMP para o proxy.
Deve ser o mesmo que em zabbix_trap_receiver.pl ou no arquivo de configuração SNMPTT.

Padrão: `/tmp/zabbix_traps.tmp`

SocketDir

O diretório para armazenar os soquetes IPC usados pelos serviços internos do Zabbix.

Padrão: `/tmp`

SourceIP

O endereço IP de origem para:

- conexões de saída para o Zabbix server
- conexões sem agent (VMware, SSH, JMX, SNMP, Telnet e verificações simples)
- conexões do HTTP agent
- solicitações HTTP JavaScript de item de script
- solicitações HTTP JavaScript de pré-processamento
- conexões com o Vault

SSHKeyLocation

O local das chaves públicas e privadas para verificações e ações de SSH.

SSLCertLocation

O local dos arquivos de certificado do cliente SSL para autenticação do cliente.
Esse parâmetro é usado somente no monitoramento da Web.

SSLKeyLocation

O local dos arquivos de chave privada SSL para autenticação do cliente.
Esse parâmetro é usado somente no monitoramento da Web.

SSLCALocation

O local dos arquivos da autoridade certificadora (CA) para verificação do certificado SSL do servidor.
Observe que o valor deste parâmetro será definido como a opção CURLOPT_CAPATH do libcurl. Para versões do libcurl anteriores à 7.42.0, isso só tem efeito se o libcurl foi compilado para usar o OpenSSL. Para mais informações, consulte a página da web do cURL.
Este parâmetro é usado no monitoramento web e na autenticação SMTP.

StartAgentPollers

O número de instâncias pré-forked de agent pollers do Zabbix. Veja MaxConcurrentChecksPerPoller.

Padrão: 1
 Intervalo: 0-1000

StartBrowserPollers

O número de instâncias pré-forked de browser item pollers.

Padrão: 1
 Intervalo: 0-1000

StartDBSyncers

O número de instâncias pré-forked de history syncers.
Nota: Tenha cuidado ao alterar este valor, aumentá-lo pode causar mais problemas do que benefícios.

Padrão: 4
 Intervalo: 1-100

StartDiscoverers

O número de instâncias pré-forked de trabalhadores de descoberta.

Padrão: 5
 Intervalo: 0-1000

StartHTTPAgentPollers

O número de instâncias pré-forked de HTTP agent pollers. Veja MaxConcurrentChecksPerPoller.

Padrão: 1
 Intervalo: 0-1000

StartHTTPPollers

O número de instâncias pré-forked de HTTP pollers.

Padrão: 1
 Intervalo: 0-1000

StartIPMIPollers

O número de instâncias pré-forked de IPMI pollers.

Padrão: 0
 Intervalo: 0-1000

StartJavaPollers

O número de instâncias pré-forked de Java pollers.

Padrão: 0
 Intervalo: 0-1000

StartODBCPollers

O número de instâncias pré-forked de ODBC pollers.

Padrão: 1
 Intervalo: 0-1000

StartPingers

O número de instâncias pré-forked de ICMP pingers.

Padrão: 1
 Intervalo: 0-1000

StartPollersUnreachable

O número de instâncias pré-forked de pollers para hosts inacessíveis (incluindo IPMI e Java). Pelo menos um poller para hosts inacessíveis deve estar em execução se pollers regulares, IPMI ou Java forem iniciados.

Padrão: 1
 Intervalo: 0-1000

StartPollers

O número de instâncias pré-forked de pollers.

Padrão: 5
 Intervalo: 0-1000

StartPreprocessors

O número de instâncias pré-iniciadas de threads de worker de pré-processamento deve ser definido para não menos do que a contagem de núcleos de CPU disponíveis. Mais workers devem ser definidos se o pré-processamento não for limitado pela CPU e tiver muitas solicitações de rede.

Padrão: 16 (antes do Zabbix 7.0.6: 3)
 Intervalo: 1-1000

StartSNMPPollers

O número de instâncias pré-forked de SNMP pollers. Veja MaxConcurrentChecksPerPoller.

Padrão: 1
 Intervalo: 0-1000

StartSNMPTrapper

Se definido como 1, um processo SNMP trapper será iniciado.

Padrão: 0
 Intervalo: 0-1

StartTrappers

O número de instâncias pré-forked de trappers.
Trappers aceitam conexões recebidas do Zabbix sender e agents ativos.

Padrão: 5
 Intervalo: 0-1000

StartVMwareCollectors

O número de instâncias de coletor VMware pré-forked.

Padrão: 0
 Intervalo: 0-250

StatsAllowedIP

Uma lista de endereços IP separados por vírgula, opcionalmente na notação CIDR, ou nomes DNS de instâncias externas do Zabbix. A solicitação de estatísticas será aceita apenas dos endereços listados aqui. Se este parâmetro não for definido, nenhuma solicitação de estatísticas será aceita.
Se o suporte a IPv6 estiver habilitado, então '127.0.0.1', '::127.0.0.1', '::ffff:127.0.0.1' são tratados igualmente e '::/0' permitirá qualquer endereço IPv4 ou IPv6. '0.0.0.0/0' pode ser usado para permitir qualquer endereço IPv4.

Exemplo:

```
StatsAllowedIP=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com
```

Timeout

Especifica quanto tempo esperar (em segundos) para estabelecer conexão e trocar dados com o Zabbix server, agent, serviço web e verificações SNMP legadas (número OID único ou string).

Este parâmetro define a duração para várias operações de comunicação:

- Execução de comando remoto no Zabbix agent
- Execução de comando SSH / Telnet
- Reagendamento de items quando a interface IPMI se torna indisponível
- Envio de resposta ao Zabbix server ao falhar na troca de dados devido a problemas de direitos ou criptografia
- Prazo para sockets IPC assíncronos e opções de controle de tempo de execução
- Solicitações DNS do poller assíncrono
- Resposta para heartbeat de verificação ativa
- Recuperação de dados (valores) do Zabbix agent de agents ativos
- Recuperação de dados do Zabbix sender
- Envio da lista de verificações ativas para o Zabbix agent

Este timeout **não** será usado para aquelas verificações que possuem configurações de timeout flexível definidas no frontend (em nível global, proxy ou por item). Por exemplo, items SNMP `walk[OID]` e `get[OID]` usam o timeout configurado no frontend; verificações SNMP legadas ainda usam o valor de timeout do server.

Padrão: 3
 Intervalo: 1-30

TLSAccept

Quais conexões de entrada aceitar do Zabbix server. Usado para proxy passivo, ignorado em proxy ativo. Vários valores podem ser especificados, separados por vírgula:
unencrypted - aceitar conexões sem criptografia (padrão)
psk - aceitar conexões com TLS e uma chave pré-compartilhada (PSK)
cert - aceitar conexões com TLS e um certificado

Obrigatório: sim para proxy passivo, se parâmetros de certificado TLS ou PSK forem definidos (mesmo para conexão *unencrypted*); caso contrário, não

TLSCAFile

O nome completo do caminho de um arquivo que contém os certificados da(s) CA(s) de nível superior para verificação de certificados de pares, usados para comunicações criptografadas entre os componentes do Zabbix.

TLSCertFile

O nome completo do caminho de um arquivo que contém o certificado de proxy ou a cadeia de certificados, usado para comunicações criptografadas entre os componentes do Zabbix.

TLSCipherAll

A string de prioridade do GnuTLS ou a string de cifra do OpenSSL (TLS 1.2). Substitui os critérios padrão de seleção de ciphersuite para criptografia baseada em certificado e PSK.

Exemplo:

```
TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSCipherAll13

A string de cifra para OpenSSL 1.1.1 ou mais recente em TLS 1.3. Substitui os critérios padrão de seleção de ciphersuite para criptografia baseada em certificado e PSK.

Exemplo para GnuTLS:

```
NONE:+VERS-TLS1.2:+ECDHE-RSA:+RSA:+ECDHE-PSK:+PSK:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL
```

Exemplo para OpenSSL:

```
EECDH+aRSA+AES128:RSA+aRSA+AES128:kECDHEPSK+AES128:kPSK+AES128
```

TLSCipherCert

A string de prioridade do GnuTLS ou a string de cifra do OpenSSL (TLS 1.2). Substitui os critérios padrão de seleção de ciphersuite para criptografia baseada em certificado.

Exemplo para GnuTLS:

```
NONE:+VERS-TLS1.2:+ECDHE-RSA:+RSA:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL:+COMP-NULL:+SIG
```

Exemplo para OpenSSL:

```
EECDH+aRSA+AES128:RSA+aRSA+AES128
```

TLSCipherCert13

A string de cifra para OpenSSL 1.1.1 ou mais recente em TLS 1.3. Substitui os critérios padrão de seleção de ciphersuite para criptografia baseada em certificado.

TLSCipherPSK

A string de prioridade do GnuTLS ou a string de cifra do OpenSSL (TLS 1.2). Substitui os critérios padrão de seleção de ciphersuite para criptografia baseada em PSK.

Exemplo para GnuTLS:

```
NONE:+VERS-TLS1.2:+ECDHE-PSK:+PSK:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL:+COMP-NULL:+SIG
```

Exemplo para OpenSSL:

```
kECDHEPSK+AES128:kPSK+AES128
```

TLSCipherPSK13

A string de cifra para OpenSSL 1.1.1 ou mais recente no TLS 1.3. Substitui os critérios padrão de seleção de ciphersuite para criptografia baseada em PSK.

Exemplo:

```
TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSConnect

Como o proxy deve se conectar ao Zabbix server. Usado para proxy ativo, ignorado em proxy passivo. Apenas um valor pode ser especificado:
unencrypted - conectar sem criptografia (padrão)
psk - conectar usando TLS e uma chave pré-compartilhada (PSK)
cert - conectar usando TLS e um certificado

Obrigatório: sim para proxy ativo, se os parâmetros de certificado TLS ou PSK estiverem definidos (mesmo para conexão *unencrypted*); caso contrário, não

TLSCRLFile

O caminho completo de um arquivo contendo certificados revogados. Este parâmetro é usado para comunicações criptografadas entre os componentes do Zabbix.

TLSKeyFile

O nome completo do caminho de um arquivo que contém a chave privada do proxy, usada para comunicações criptografadas entre os componentes do Zabbix.

TLSPSKFile

O nome completo do caminho de um arquivo que contém a chave pré-compartilhada do proxy, usada para comunicações criptografadas com o servidor Zabbix.

TLSPSKIdentity

A string de identidade da chave pré-compartilhada, usada para comunicações criptografadas com o servidor Zabbix.

TLSServerCertIssuer

O emissor do certificado do servidor permitido.

TLSServerCertSubject

O assunto do certificado do servidor permitido.

TmpDir

O diretório temporário.

Padrão: `/tmp`

TrapperTimeout

Especifica o tempo limite em segundos para:
 - recuperação de dados de configuração do Zabbix server;
 - execução de script global ou execução de comando remoto.

Padrão: 300
 Intervalo: 1-300

UnavailableDelay

Com que frequência um host é verificado quanto à disponibilidade durante o período de indisponibilidade, em segundos.

Padrão: 60
 Intervalo: 1-3600

UnreachableDelay

Com que frequência um host é verificado quanto à disponibilidade durante o período de inacessibilidade em segundos.

Padrão: 15
 Intervalo: 1-3600

UnreachablePeriod

Após quantos segundos de inacessibilidade tratar um host como indisponível.

Padrão: 45
 Intervalo: 1-3600

Usuário

Reduz os privilégios de um usuário específico existente no sistema.
Só terá efeito se for executado como ”root” e se AllowRoot estiver desativado.

Padrão: `zabbix`

Vault

O provedor do cofre:
HashiCorp - HashiCorp KV Secrets Engine versão 2
CyberArk - CyberArk Central Credential Provider
Deve corresponder ao provedor do cofre definido no frontend.

Padrão: `HashiCorp`

VaultDBPath

Caminho ou consulta do Vault, dependendo do Vault, de onde as credenciais para o banco de dados serão recuperadas pelas chaves.

As chaves usadas para HashiCorp são ’password’ e ’username’.

Exemplo de caminho com VaultPrefix=/v1/secret/data/zabbix/:

```
database
```

Exemplo de caminho sem VaultPrefix:

```
secret/zabbix/database
```

As chaves usadas para CyberArk são ’Content’ e ’UserName’.

Exemplo:

```
AppID=zabbix_server&Query=Safe=passwordSafe;Object=zabbix_proxy_database
```

Esta opção só pode ser usada se DBUser e DBPassword não estiverem especificados.

VaultPrefix

Um prefixo personalizado para o caminho ou consulta do Vault, dependendo do Vault. Os padrões mais adequados serão usados se não especificado.
Observe que 'data' é automaticamente anexado após o ponto de montagem para HashiCorp se VaultPrefix não for especificado.

Exemplo de prefixo para Hashicorp:

```
v1/secret/data/zabbix/
```

Exemplo de prefixo para Cyberark:

```
/AIMWebService/api/Accounts?
```

VaultTLSCertFile

O nome do arquivo de certificado SSL usado para autenticação do cliente. O arquivo de certificado deve estar no formato PEM1.
Se o arquivo de certificado também contiver a chave privada, deixe o campo do arquivo de chave SSL vazio.
O diretório que contém este arquivo é especificado pelo parâmetro de configuração SSLCertLocation.
Esta opção pode ser omitida, mas é recomendada para o cofre CyberArkCCP.

VaultTLSKeyFile

O nome do arquivo de chave privada SSL usado para autenticação do cliente. O arquivo de chave privada deve estar no formato PEM1.
O diretório que contém este arquivo é especificado pelo parâmetro de configuração SSLKeyLocation.
Esta opção pode ser omitida, mas é recomendada para o cofre CyberArkCCP.

VaultToken

O token de autenticação do cofre da HashiCorp que deveria ter sido gerado exclusivamente para o proxy do Zabbix com permissão somente leitura para o caminho especificado no parâmetro de configuração opcional VaultDBPath.
É um erro se o VaultToken e a variável de ambiente VAULT_TOKEN forem definidos ao mesmo tempo.

Obrigatório: Sim, se o Vault estiver definido como *HashiCorp*; caso contrário, não

VaultURL

A URL HTTP[S] do servidor Vault. O diretório de certificados CA do sistema será usado se SSLCALocation não for especificado.

Padrão: `https://127.0.0.1:8200`

VMwareCacheSize

O tamanho da memória compartilhada para armazenar dados do VMware.
Uma verificação interna do VMware zabbix[vmware,buffer,...] pode ser usada para monitorar o uso do cache do VMware (consulte Internal checks).
Observe que a memória compartilhada não é alocada se não houver instâncias do coletor do VMware configuradas para iniciar.

Padrão: 8M
 Intervalo: 256K-2G

VMwareFrequency

O atraso, em segundos, entre a coleta de dados de um único serviço da VMware.
Esse atraso deve ser definido como o menor intervalo de atualização de qualquer item de monitoramento da VMware.

Padrão: 60
 Intervalo: 10-86400

VMwarePerfFrequency

O atraso em segundos entre a recuperação das estatísticas do contador de desempenho de um único serviço VMware.
Esse atraso deve ser definido para o menor intervalo de atualização de qualquer item de monitoramento VMware que use contadores de desempenho VMware.

Padrão: 60
 Intervalo: 10-86400

VMwareTimeout

O número máximo de segundos que um coletor vmware aguardará por uma resposta do serviço VMware (hipervisor vCenter ou ESX).

Padrão: 10
 Intervalo: 1-300

WebDriverURL

URL HTTP[S] da interface WebDriver.

Exemplo (usado com o Selenium WebDriver standalone server):

```
WebDriverURL=http://localhost:4444
```

Notas de rodapé

[1] DBSocket e DBPort são mutuamente exclusivos na configuração do proxy. Especifique apenas um ou deixe ambos indefinidos.

3 Agente Zabbix (UNIX)

Visão geral

Os parâmetros suportados pelo arquivo de configuração do agent Zabbix (zabbix_agentd.conf) estão listados nesta seção.

Os parâmetros são listados sem informações adicionais. Clique no parâmetro para ver os detalhes completos.

Parâmetro	Descrição
Alias	Define um alias para uma chave de item.
AllowKey	Permite a execução das chaves de item que correspondam a um padrão.
AllowRoot	Permite que o agent seja executado como 'root'.
BufferSend	Não manter dados no buffer por mais de N segundos.
BufferSize	O número máximo de valores no buffer de memória.
DebugLevel	O nível de debug.
DenyKey	Nega a execução das chaves de item que correspondam a um padrão.
EnableRemoteCommands	Se comandos remotos do Zabbix server são permitidos.
HeartbeatFrequency	A frequência das mensagens de heartbeat em segundos.
HostInterface	Um parâmetro opcional que define a interface do host.
HostInterfaceItem	Um parâmetro opcional que define um item usado para obter a interface do host.
HostMetadata	Um parâmetro opcional que define os metadados do host.
HostMetadataItem	Um parâmetro opcional que define um item do agent Zabbix usado para obter os metadados do host.
Hostname	Um parâmetro opcional que define o nome do host.
HostnameItem	Um parâmetro opcional que define um item do agent Zabbix usado para obter o nome do host.
Include	Você pode incluir arquivos individuais ou todos os arquivos em um diretório no arquivo de configuração.
ListenBacklog	O número máximo de conexões pendentes na fila TCP.
ListenIP	Uma lista de endereços IP separados por vírgula nos quais o agent deve escutar.
ListenPort	O agent irá escutar neste porto por conexões do server.
LoadModule	O módulo a ser carregado na inicialização do agent.
LoadModulePath	O caminho completo para a localização dos módulos do agent.
LogFile	O nome do arquivo de log.
LogFileSize	O tamanho máximo do arquivo de log.
LogRemoteCommands	Habilita o registro de comandos shell executados como avisos.
LogType	O tipo de saída do log.
MaxLinesPerSecond	O número máximo de novas linhas que o agent enviará por segundo para o Zabbix server ou proxy ao processar verificações ativas 'log' e 'logrt'.
PidFile	O nome do arquivo PID.
RefreshActiveChecks	Com que frequência a lista de verificações ativas é atualizada.
Server	Uma lista de endereços IP separados por vírgula, opcionalmente na notação CIDR, ou nomes DNS dos Zabbix servers e Zabbix proxies.
ServerActive	O endereço do Zabbix server/proxy ou configuração de cluster para obter verificações ativas.
SourceIP	O endereço IP de origem.
StartAgents	O número de instâncias pré-criadas do zabbix_agentd que processam verificações passivas.
Timeout	Especifica quanto tempo esperar (em segundos) para estabelecer conexão e trocar dados com o Zabbix proxy ou server.
TLSAccept	Quais conexões de entrada aceitar.
TLSCAFile	O caminho completo de um arquivo contendo os certificados da(s) CA(s) de nível superior para verificação do certificado do peer, usado para comunicações criptografadas entre componentes do Zabbix.
TLSCertFile	O caminho completo de um arquivo contendo o certificado do agent ou cadeia de certificados, usado para comunicações criptografadas entre componentes do Zabbix.
TLSCipherAll	A string de prioridade do GnuTLS ou a string de cifra do OpenSSL (TLS 1.2). Substitui os critérios padrão de seleção de ciphersuite para criptografia baseada em certificado e PSK.
TLSCipherAll13	A string de cifra para OpenSSL 1.1.1 ou mais recente em TLS 1.3. Substitui os critérios padrão de seleção de ciphersuite para criptografia baseada em certificado e PSK.

Parâmetro	Descrição
TLSCipherCert	A string de prioridade do GnuTLS ou a string de cifra do OpenSSL (TLS 1.2). Substitui os critérios padrão de seleção de ciphersuite para criptografia baseada em certificado.
TLSCipherCert13	A string de cifra para OpenSSL 1.1.1 ou mais recente em TLS 1.3. Substitui os critérios padrão de seleção de ciphersuite para criptografia baseada em certificado.
TLSCipherPSK	A string de prioridade do GnuTLS ou a string de cifra do OpenSSL (TLS 1.2). Substitui os critérios padrão de seleção de ciphersuite para criptografia baseada em PSK.
TLSCipherPSK13	A string de cifra para OpenSSL 1.1.1 ou mais recente em TLS 1.3. Substitui os critérios padrão de seleção de ciphersuite para criptografia baseada em PSK.
TLSConnect	Como o agent deve se conectar ao Zabbix server ou proxy.
TLSCRLFile	O caminho completo de um arquivo contendo certificados revogados. Este parâmetro é usado para comunicações criptografadas entre componentes do Zabbix.
TLSKeyFile	O caminho completo de um arquivo contendo a chave privada do agent, usado para comunicações criptografadas entre componentes do Zabbix.
TLSPSKFile	O caminho completo de um arquivo contendo a chave pré-compartilhada do agent, usado para comunicações criptografadas com o Zabbix server.
TLSPSKIdentity	A string de identidade da chave pré-compartilhada, usada para comunicações criptografadas com o Zabbix server.
TLSServerCertIssuer	O emissor do certificado do server (proxy) permitido.
TLSServerCertSubject	O assunto do certificado do server (proxy) permitido.
UnsafeUserParameters	Permite que todos os caracteres sejam passados como argumentos para parâmetros definidos pelo usuário.
User	Reduz privilégios para um usuário específico e existente no sistema.
UserParameter	Um parâmetro definido pelo usuário para monitorar.
UserParameterDir	O caminho de pesquisa padrão para comandos UserParameter.

Todos os parâmetros são não obrigatórios, a menos que explicitamente declarado que o parâmetro é obrigatório.

Observe que:

- Os valores padrão refletem os padrões do daemon, não os valores nos arquivos de configuração fornecidos;
- O Zabbix suporta arquivos de configuração apenas em codificação UTF-8 sem BOM;
- Comentários iniciados com "#" são suportados apenas no início da linha.

Detalhes do parâmetro

Alias

Define um alias para uma chave de item. Pode ser usado para substituir uma chave de item longa e complexa por uma mais curta e simples.
 Vários parâmetros *Alias* podem estar presentes. Não são permitidos vários parâmetros com a mesma chave *Alias*.
 Chaves *Alias* diferentes podem fazer referência à mesma chave de item.
 Aliases podem ser usados em *HostMetadataItem*, mas não no parâmetro *HostnameItem*.

Exemplo 1: Recuperando o ID do usuário 'zabbix'.

```
Alias=zabbix.userid:vfs.file.regexp[/etc/passwd,"^zabbix:.:([0-9]+)",,,,\1]
```

Agora a chave abreviada **zabbix.userid** pode ser usada para recuperar dados.

Exemplo 2: Obtendo a utilização da CPU com parâmetros padrão e personalizados.

```
Alias=cpu.util:system.cpu.util
Alias=cpu.util[*]:system.cpu.util[*]
```

Isso permite usar a chave **cpu.util** para obter a porcentagem de utilização da CPU com parâmetros padrão, bem como usar **cpu.util[all, idle, avg15]** para obter dados específicos sobre a utilização da CPU.

Exemplo 3: Executando várias regras de descoberta de baixo nível processando os mesmos itens de descoberta.

```
Alias=vfs.fs.discovery[*]:vfs.fs.discovery
```

Agora é possível configurar várias regras de descoberta usando **vfs.fs.discovery** com parâmetros diferentes para cada regra, por exemplo, **vfs.fs.discovery[foo]**, **vfs.fs.discovery[bar]**, etc.

AllowKey

Permite a execução das chaves de item que correspondam a um padrão. O padrão de chave é uma expressão curinga que suporta o caractere "*" para corresponder a qualquer número de quaisquer caracteres.
Várias regras de correspondência de chaves

podem ser definidas em combinação com DenyKey. Os parâmetros são processados um a um de acordo com a ordem de aparição. Veja também: Restringindo verificações do agent.

AllowRoot

Permite que o agent seja executado como 'root'. Se desabilitado e o agent for iniciado por 'root', o agent tentará alternar para o usuário 'zabbix'. Não tem efeito se iniciado por um usuário comum.

Padrão: 0
 Valores: 0 - não permitir; 1 - permitir

BufferSend

Não mantenha dados com mais de N segundos no buffer.

Padrão: 5
 Intervalo: 1-3600

BufferSize

O número máximo de valores no buffer de memória. O agent enviará todos os dados coletados para o Zabbix server ou proxy se o buffer estiver cheio.

Padrão: 100
 Intervalo: 2-65535

DebugLevel

Especifique o nível de depuração:
0 - informações básicas sobre a inicialização e a interrupção dos processos do Zabbix
1 - informações críticas;
2 - informações sobre erros;
3 - avisos;
4 - para depuração (produz muitas informações);
5 - depuração estendida (produz ainda mais informações).

Padrão: 3
 Intervalo: 0-5

DenyKey

Nega a execução das keys de item que correspondam a um padrão. O padrão da key é uma expressão curinga que suporta o caractere "*" para corresponder a qualquer número de quaisquer caracteres.
Várias regras de correspondência de key podem ser definidas em combinação com AllowKey. Os parâmetros são processados um a um de acordo com sua ordem de aparição. Veja também: Restringindo verificações do agent.

EnableRemoteCommands

Se comandos remotos do Zabbix server são permitidos. Este parâmetro está **obsoleto**, use AllowKey=system.run[*] ou DenyKey=system.run[*] em vez disso.
É um alias interno para os parâmetros AllowKey/DenyKey dependendo do valor:
0 - DenyKey=system.run[*]
1 - AllowKey=system.run[*]

Padrão: 0
 Valores: 0 - não permitir, 1 - permitir

HeartbeatFrequency

A frequência das mensagens de heartbeat em segundos. Usado para monitorar a disponibilidade de verificações ativas.
0 - mensagens de heartbeat desabilitadas.

Padrão: 60
 Intervalo: 0-3600

HostInterface

Um parâmetro opcional que define a interface do host (endereço IP ou nome DNS) usada durante o processo de autoregistro do host. Esse valor será usado para preencher a interface no host recém-criado e permite configurar explicitamente o endereço IP ou DNS. Para mais detalhes, consulte Usando DNS como interface padrão.

Se não for definido, o valor será adquirido do HostInterfaceItem.

O agent emitirá um erro e não iniciará se o valor ultrapassar o limite de 255 caracteres.

Intervalo: 0-255 caracteres

HostInterfaceItem

Um parâmetro opcional que define um item usado para determinar a interface do host (endereço IP ou nome DNS) durante o processo de autoregistro do host. Esse valor é usado apenas se HostInterface não estiver definido. Para mais detalhes, consulte Usando DNS como interface padrão.

Durante uma solicitação de autoregistro, o agent registrará uma mensagem de aviso se o valor retornado pelo item especificado exceder o limite de 255 caracteres.

O item system.run[] é suportado independentemente das configurações AllowKey/DenyKey.

HostMetadata

Um parâmetro opcional que define os metadados usados para identificar ou distinguir o host durante o processo de autoregistro do host (agent ativo). HostMetadata permite distinguir entre hosts além do nome do host.

Se não for definido, o valor será adquirido de HostMetadataItem.

O agent emitirá um erro e não iniciará se o valor especificado exceder o limite de 2034 bytes ou for uma string não UTF-8. Quando um parâmetro espera um endereço IP ou nome DNS, valores que são UTF-8 válidos, mas não são IPs ou nomes DNS válidos, também serão rejeitados e relatados como inválidos.

Intervalo: 0-2034 bytes

HostMetadataItem

Um parâmetro opcional que define um item do agent Zabbix usado para obter metadados do host. Esta opção só é usada quando HostMetadata não está definido.

O valor de HostMetadataItem é obtido em cada tentativa de autoregistro e é usado apenas no processo de autoregistro do host (agent ativo). HostMetadataItem permite distinguir entre hosts além do nome do host.

Parâmetros de usuário e aliases são suportados. O item system.run[] é suportado independentemente das configurações AllowKey/DenyKey.

Durante uma solicitação de autoregistro, o agent registrará uma mensagem de aviso se o valor retornado pelo item especificado exceder o limite de 65535 pontos de código UTF-8. O valor retornado pelo item deve ser uma string UTF-8, caso contrário, será ignorado. Quando um parâmetro espera um endereço IP ou nome DNS, valores que são UTF-8 válidos, mas não são IPs ou nomes DNS válidos, também serão rejeitados e relatados como inválidos.

Hostname

Uma lista de nomes de host exclusivos, sensíveis a maiúsculas e minúsculas, separados por vírgulas. Necessário para verificações ativas e deve corresponder aos nomes de host configurados no server. O valor é obtido de HostnameItem se não estiver definido.
Caracteres permitidos: alfanuméricos, '.', ' ', '_' e '-'. Comprimento máximo: 128 caracteres por nome de host, 2048 caracteres para toda a linha.

Padrão: Definido por HostnameItem

HostnameItem

Um parâmetro opcional que define um item do agent Zabbix usado para obter o nome do host. Esta opção só é usada quando Hostname não está definido. Parâmetros de usuário ou aliases não são suportados, mas o item system.run[] é suportado independentemente dos valores AllowKey/DenyKey.

Padrão: `system.hostname`

Include

Você pode incluir arquivos individuais ou todos os arquivos em um diretório no arquivo de configuração. Para incluir apenas arquivos relevantes no diretório especificado, o caractere curinga asterisco é suportado para correspondência de padrões.
Veja notas especiais sobre limitações.

Exemplo:

```
Include=/absolute/path/to/config/files/*.conf
```

ListenBacklog

O número máximo de conexões pendentes na fila TCP.
O valor padrão é uma constante codificada, que depende do sistema.
O valor máximo suportado depende do sistema; valores muito altos podem ser silenciosamente truncados para o "máximo especificado pela implementação".

Padrão: `SOMAXCONN`
 Intervalo: 0 - INT_MAX

ListenIP

Uma lista de endereços IP delimitados por vírgulas nos quais o agente deve escutar.

Padrão: `0.0.0.0`

ListenPort

O agente escutará nessa porta as conexões do servidor.

Padrão: 10050
 Intervalo: 1024-32767

LoadModule

O módulo a ser carregado na inicialização do agent. Os módulos são usados para estender a funcionalidade do agent. O módulo deve estar localizado no diretório especificado por LoadModulePath ou o caminho deve preceder o nome do módulo. Se o caminho anterior for absoluto (começar com '/') então LoadModulePath será ignorado.
Formatos:
LoadModule=<module.so>
LoadModule
permitido incluir vários parâmetros LoadModule.

LoadModulePath

O caminho completo para o local dos módulos do agent. O padrão depende das opções de compilação.

LogFile

O nome do arquivo de registro.

Obrigatório: Sim, se LogType estiver definido como *file*; caso contrário, não

LogFileSize

O tamanho máximo de um arquivo de log em MB.
0 - desativa a rotação automática de logs.
Nota: Se o limite de tamanho do arquivo de log for atingido e a rotação do arquivo falhar, por qualquer motivo, o arquivo de log existente será truncado e iniciado novamente.

Padrão: 1
 Intervalo: 0-1024

LogRemoteCommands

Habilita o registro dos comandos shell executados como avisos. Os comandos serão registrados apenas se executados remotamente. As entradas de log não serão criadas se system.run[] for iniciado localmente pelos parâmetros HostMetadataItem, HostInterfaceItem ou HostnameItem.

Padrão: 0
 Valores: 0 - desabilitado, 1 - habilitado

LogType

O tipo da saída de registro:
file - grava o registro no arquivo especificado pelo parâmetro LogFile;
system - grava o registro no syslog;
console - grava o registro na saída padrão.

Padrão: `file`

MaxLinesPerSecond

O número máximo de novas linhas que o agent enviará por segundo para o Zabbix server ou proxy ao processar verificações ativas 'log' e 'logrt'. O valor fornecido será substituído pelo parâmetro 'maxlines', fornecido na chave do item 'log' ou 'logrt'.
Nota: O Zabbix processará 10 vezes mais novas linhas do que definido em *MaxLinesPerSecond* para buscar a string necessária nos itens de log.

Padrão: 20
 Intervalo: 1-1000

PidFile

O nome do arquivo PID.

Padrão: `/tmp/zabbix_agentd.pid`

RefreshActiveChecks

Com que frequência a lista de verificações ativas é atualizada, em segundos. Observe que, após falhar ao atualizar as verificações ativas, a próxima atualização será tentada em 60 segundos.

Padrão: 5
 Intervalo: 1-86400

Server

Uma lista de endereços IP separados por vírgula, opcionalmente na notação CIDR, ou nomes DNS de servidores Zabbix e proxies Zabbix. Conexões de entrada serão aceitas apenas dos hosts listados aqui. Se o suporte a IPv6 estiver habilitado, então '127.0.0.1', '::127.0.0.1', '::ffff:127.0.0.1' são tratados igualmente e '::/0' permitirá qualquer endereço IPv4 ou IPv6. '0.0.0.0/0' pode ser usado para permitir qualquer endereço IPv4. Observe que "endereços IPv6 compatíveis com IPv4" (prefixo 0000::/96) são suportados, mas obsoletos pelo RFC4291. Espaços são permitidos.

Exemplo:

```
Server=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com
```

Obrigatório: sim, se StartAgents não estiver explicitamente definido como 0

ServerActive

O endereço do Zabbix server/proxy ou configuração de cluster para obter verificações ativas. O endereço do server/proxy é um endereço IP ou nome DNS e porta opcional separada por dois pontos.
A configuração do cluster é um ou mais endereços de

membros do grupo de server ou proxy separados por ponto e vírgula. Vários Zabbix servers/clusters e Zabbix proxies podem ser especificados, separados por vírgula. A menos que esteja usando grupos de proxy, não deve ser especificado mais de um Zabbix proxy de cada Zabbix server/cluster. Se o Zabbix proxy for especificado, então o Zabbix server/cluster para esse proxy não deve ser especificado.
Vários endereços separados por vírgula podem ser fornecidos para usar vários Zabbix servers independentes em paralelo. Espaços são permitidos.
Se a porta não for especificada, a porta padrão será usada.
Endereços IPv6 devem ser colocados entre colchetes se a porta para esse host for especificada. Se a porta não for especificada, os colchetes para endereços IPv6 são opcionais.
Se este parâmetro não for especificado, as verificações ativas serão desativadas.

Exemplo para Zabbix proxy:

```
ServerActive=127.0.0.1:10051
```

Exemplo para grupo de Zabbix proxy:

```
ServerActive=proxy1.example.com;proxy2.example.com;proxy3.example.com;proxy4.example.com;proxy5.example.co
```

Exemplo para múltiplos servers:

```
ServerActive=127.0.0.1:20051,zabbix.domain,[::1]:30051,::1,[12fc::1]
```

Exemplo para alta disponibilidade:

```
ServerActive=zabbix.cluster.node1;zabbix.cluster.node2:20051;zabbix.cluster.node3
```

Exemplo para alta disponibilidade com dois clusters e um server:

```
ServerActive=zabbix.cluster.node1;zabbix.cluster.node2:20051,zabbix.cluster2.node1;zabbix.cluster2.node2,z
```

SourceIP

O endereço IP de origem para:

- conexões de saída para o Zabbix server ou Zabbix proxy;
- fazer conexões ao executar alguns items (web.page.get, net.tcp.port, etc.).

StartAgents

O número de instâncias pré-forked do zabbix_agentd que processam verificações passivas. Se definido como 0, as verificações passivas são desativadas e o agent não irá escutar em nenhuma porta TCP.

Padrão: 10
 Intervalo: 0-100

Timeout

Especifica por quanto tempo aguardar (em segundos) para estabelecer conexão e trocar dados com o proxy ou server Zabbix.

Este parâmetro define a duração de várias operações de comunicação, incluindo:

- aguardar uma resposta do server Zabbix;
- enviar solicitações ao server Zabbix, incluindo solicitações de configuração de item e dados de item em verificações ativas;
- recuperar dados de log através do logfile;
- enviar mensagens de heartbeat;
- duração máxima para verificações `vfs.*`;
- sendo usado pelos módulos do agent Zabbix;
- servindo como fallback em cenários onde um server ou proxy anterior à versão 7.0 envia verificações sem timeouts.

Este timeout **não** será usado para aquelas verificações de agent que possuem configurações de timeout configuráveis no frontend (em nível global, de proxy ou por item).

Padrão: 3
 Intervalo: 1-30

TLSAccept

Quais conexões de entrada aceitar. Usado para verificações passivas. Vários valores podem ser especificados, separados por vírgula:
unencrypted - aceitar conexões sem criptografia (padrão)
psk - aceitar conexões com TLS e uma chave pré-compartilhada (PSK)
cert - aceitar conexões com TLS e um certificado

Obrigatório: sim, se os parâmetros de certificado TLS ou PSK forem definidos (mesmo para conexão *unencrypted*); caso contrário, não

TLSCAFile

O nome completo do caminho do arquivo que contém os certificados da(s) CA(s) de nível superior para verificação de certificados de pares, usados para comunicações criptografadas entre os componentes do Zabbix.

TLSCertFile

O nome completo do caminho do arquivo que contém o certificado do agente ou a cadeia de certificados, usado para comunicações criptografadas com os componentes do Zabbix.

TLSCipherAll

A string de prioridade do GnuTLS ou a string de cifra do OpenSSL (TLS 1.2). Substitui os critérios padrão de seleção de ciphersuite para criptografia baseada em certificado e PSK.

Exemplo:

```
TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSCipherAll13

A string de cifra para OpenSSL 1.1.1 ou mais recente no TLS 1.3. Substitua os critérios de seleção de ciphersuite padrão para criptografia baseada em certificado e PSK.

Exemplo para GnuTLS:

```
NONE:+VERS-TLS1.2:+ECDHE-RSA:+RSA:+ECDHE-PSK:+PSK:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL
```

Exemplo para OpenSSL:

```
EECDH+aRSA+AES128:RSA+aRSA+AES128:kECDHEPSK+AES128:kPSK+AES128
```

TLSCipherCert

A string de prioridade do GnuTLS ou a string de cifra do OpenSSL (TLS 1.2). Substitui os critérios padrão de seleção de ciphersuite para criptografia baseada em certificado.

Exemplo para GnuTLS:

```
NONE:+VERS-TLS1.2:+ECDHE-RSA:+RSA:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL:+COMP-NULL:+SIG
```

Exemplo para OpenSSL:

```
EECDH+aRSA+AES128:RSA+aRSA+AES128
```

TLSCipherCert13

A string de cifra para OpenSSL 1.1.1 ou mais recente no TLS 1.3. Substitui os critérios padrão de seleção de ciphersuite para criptografia baseada em certificado.

TLSCipherPSK

A string de prioridade do GnuTLS ou a string de cifra do OpenSSL (TLS 1.2). Substitui os critérios padrão de seleção de ciphersuite para criptografia baseada em PSK.

Exemplo para GnuTLS:

```
NONE:+VERS-TLS1.2:+ECDHE-PSK:+PSK:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL:+COMP-NULL:+SIG
```

Exemplo para OpenSSL:

```
kECDHEPSK+AES128:kPSK+AES128
```

TLSCipherPSK13

A string de cifra para OpenSSL 1.1.1 ou mais recente no TLS 1.3. Substitui os critérios padrão de seleção de ciphersuite para criptografia baseada em PSK.

Exemplo:

```
TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSConnect

Como o agent deve se conectar ao Zabbix server ou proxy. Usado para verificações ativas. Apenas um valor pode ser especificado:
unencrypted - conectar sem criptografia (padrão)
psk - conectar usando TLS e uma chave pré-compartilhada (PSK)
cert - conectar usando TLS e um certificado

Obrigatório: sim, se os parâmetros de certificado TLS ou PSK forem definidos (mesmo para conexão *unencrypted*); caso contrário, não

TLSCRLFile

O caminho completo do arquivo contendo certificados revogados. Este parâmetro é usado para comunicações criptografadas entre componentes do Zabbix.

TLSKeyFile

O nome completo do caminho do arquivo que contém a chave privada do agente, usada para comunicações criptografadas entre os componentes do Zabbix.

TLSPSKFile

O nome completo do caminho do arquivo que contém a chave pré-compartilhada do agente, usada para comunicações criptografadas com o servidor Zabbix.

TLSPSKIdentity

A string de identidade da chave pré-compartilhada, usada para comunicações criptografadas com o servidor Zabbix.

TLSServerCertIssuer

O emissor do certificado do servidor (proxy) permitido.

TLSServerCertSubject

O assunto do certificado do servidor (proxy) permitido.

UnsafeUserParameters

Permitir que todos os caracteres sejam passados como argumentos para parâmetros definidos pelo usuário. Os seguintes caracteres não são permitidos: \ ' " ` * ? [ ] { } ~ $! & ; ( ) < > | # @
Além disso, caracteres de nova linha não são permitidos.

Padrão: 0
 Valores: 0 - não permitir, 1 - permitir

Usuário

Reduz os privilégios de um usuário específico existente no sistema.
Só terá efeito se for executado como "root" e se AllowRoot estiver desativado.

Padrão: `zabbix`

UserParameter

Um parâmetro definido pelo usuário para monitoramento. Pode haver vários parâmetros definidos pelo usuário.
Formato: UserParameter=<key>,<comando shell>
Observe que o comando shell não deve retornar uma string vazia ou apenas EOL. Comandos shell podem ter caminhos relativos, se o parâmetro UserParameterDir for especificado.

Exemplo:

```
UserParameter=system.test,who|wc -l
UserParameter=check_cpu,./custom_script.sh
```

UserParameterDir

O caminho de pesquisa padrão para comandos UserParameter. Se usado, o agent mudará seu diretório de trabalho para o especificado aqui antes de executar um comando. Assim, os comandos UserParameter podem ter um prefixo relativo `./` em vez de um caminho completo.
Apenas uma entrada é permitida.

Exemplo:

```
UserParameterDir=/opt/myscripts
```

Veja também

1. Diferenças na configuração do agent Zabbix para checagens ativas e passivas a partir da versão 2.0.0

4 Agente Zabbix 2 (UNIX)

Visão geral

O Zabbix agent 2 é uma nova geração do Zabbix agent e pode ser usado no lugar do Zabbix agent.

Os parâmetros suportados pelo arquivo de configuração do Zabbix agent 2 (zabbix_agent2.conf) estão listados nesta seção.

Os parâmetros são listados sem informações adicionais. Clique no parâmetro para ver os detalhes completos.

Parâmetro	Descrição
Alias	Define um alias para uma chave de item.
AllowKey	Permite a execução das chaves de item que correspondem a um padrão.
BufferSend	Não manter dados no buffer por mais de N segundos.
BufferSize	O número máximo de valores no buffer de memória.
ControlSocket	O socket de controle, usado para enviar comandos em tempo de execução com a opção '-R'.

Parâmetro	Descrição
DebugLevel	O nível de debug.
DenyKey	Nega a execução das chaves de item que correspondem a um padrão.
EnablePersistentBuffer	Habilita o uso de armazenamento persistente local para itens ativos.
ForceActiveChecksOnStart	Executa verificações ativas imediatamente após a reinicialização para a primeira configuração recebida.
HeartbeatFrequency	A frequência das mensagens de heartbeat em segundos.
HostInterface	Um parâmetro opcional que define a interface do host.
HostInterfaceItem	Um parâmetro opcional que define um item usado para obter a interface do host.
HostMetadata	Um parâmetro opcional que define os metadados do host.
HostMetadataItem	Um parâmetro opcional que define um item do Zabbix agent usado para obter os metadados do host.
Hostname	Um parâmetro opcional que define o nome do host.
HostnameItem	Um parâmetro opcional que define um item do Zabbix agent usado para obter o nome do host.
Include	Você pode incluir arquivos individuais ou todos os arquivos em um diretório no arquivo de configuração.
ListenIP	Uma lista de endereços IP separados por vírgula nos quais o agent deve escutar.
ListenPort	O agent irá escutar neste porto por conexões do server.
LogFile	O nome do arquivo de log.
LogFileSize	O tamanho máximo do arquivo de log.
LogType	O tipo de saída do log.
PersistentBufferFile	O arquivo onde o Zabbix agent 2 deve manter o banco de dados SQLite.
PersistentBufferPeriod	O período de tempo pelo qual os dados devem ser armazenados quando não houver conexão com o server ou proxy.
PidFile	O nome do arquivo PID.
Plugins.<PluginName>.System.Capacity	O limite de verificações por plugin que podem ser executadas ao mesmo tempo.
Plugins.Log.MaxLinesPerSecond	O número máximo de novas linhas que o agent enviará por segundo para o Zabbix server ou proxy ao processar verificações ativas 'log' e 'logrt'.
Plugins.SystemRun.LogRemoteCommands	Habilita o registro dos comandos shell executados como avisos.
PluginSocket	O caminho para o socket UNIX para comunicações de plugins carregáveis.
PluginTimeout	O tempo limite para conexões com plugins carregáveis, em segundos.
RefreshActiveChecks	Com que frequência a lista de verificações ativas é atualizada.
Server	Uma lista de endereços IP separados por vírgula, opcionalmente em notação CIDR, ou nomes DNS dos Zabbix servers e Zabbix proxies.
ServerActive	O endereço do Zabbix server/proxy ou configuração de cluster para obter verificações ativas.
SourceIP	O endereço IP de origem.
StatusPort	Se definido, o agent irá escutar neste porto por solicitações de status HTTP (http://localhost:<port>/status).
Timeout	Especifica quanto tempo esperar (em segundos) para estabelecer conexão e trocar dados com o Zabbix proxy ou server.
TLSAccept	Quais conexões de entrada aceitar.
TLSCAFile	O caminho completo de um arquivo contendo os certificados da(s) CA(s) de nível superior para verificação do certificado do par, usado para comunicações criptografadas entre componentes do Zabbix.
TLSCertFile	O caminho completo de um arquivo contendo o certificado do agent ou cadeia de certificados, usado para comunicações criptografadas entre componentes do Zabbix.
TLSCipherAll	A string de cifra do OpenSSL (TLS 1.2). Substitui os critérios padrão de seleção de ciphersuite para criptografia baseada em certificado e PSK.
TLSCipherAll13	A string de cifra do OpenSSL (TLS 1.3) para OpenSSL 1.1.1 ou mais recente. Substitui os critérios padrão de seleção de ciphersuite para criptografia baseada em certificado e PSK.
TLSCipherCert	A string de cifra do OpenSSL (TLS 1.2). Substitui os critérios padrão de seleção de ciphersuite para criptografia baseada em certificado.
TLSCipherCert13	A string de cifra do OpenSSL (TLS 1.3) para OpenSSL 1.1.1 ou mais recente. Substitui os critérios padrão de seleção de ciphersuite para criptografia baseada em certificado.
TLSCipherPSK	A string de cifra do OpenSSL (TLS 1.2). Substitui os critérios padrão de seleção de ciphersuite para criptografia baseada em PSK.
TLSCipherPSK13	A string de cifra do OpenSSL (TLS 1.3) para OpenSSL 1.1.1 ou mais recente. Substitui os critérios padrão de seleção de ciphersuite para criptografia baseada em PSK.
TLSConnect	Como o agent deve se conectar ao Zabbix server ou proxy.
TLSCRLFile	O caminho completo de um arquivo contendo certificados revogados. Este parâmetro é usado para comunicações criptografadas entre componentes do Zabbix.
TLSKeyFile	O caminho completo de um arquivo contendo a chave privada do agent, usado para comunicações criptografadas entre componentes do Zabbix.

Parâmetro	Descrição
TLSPSKFile	O caminho completo de um arquivo contendo a chave pré-compartilhada do agent, usado para comunicações criptografadas com o Zabbix server.
TLSPSKIdentity	A string de identidade da chave pré-compartilhada, usada para comunicações criptografadas com o Zabbix server.
TLSServerCertIssuer	O emissor do certificado do server (proxy) permitido.
TLSServerCertSubject	O assunto do certificado do server (proxy) permitido.
UnsafeUserParameters	Permite que todos os caracteres sejam passados como argumentos para parâmetros definidos pelo usuário.
UserParameter	Um parâmetro definido pelo usuário para monitorar.
UserParameterDir	O caminho de pesquisa padrão para comandos UserParameter.

Todos os parâmetros são não obrigatórios, a menos que explicitamente declarado que o parâmetro é obrigatório.

Observe que:

- Os valores padrão refletem os padrões do processo, não os valores nos arquivos de configuração fornecidos.
- O Zabbix suporta arquivos de configuração apenas na codificação UTF-8 sem BOM.
- Comentários iniciados com "#" são suportados apenas no início da linha.

Detalhes do parâmetro

Alias

Define um alias para uma chave de item. Pode ser usado para substituir uma chave de item longa e complexa por uma mais curta e simples.
 Vários parâmetros *Alias* podem estar presentes. Não são permitidos vários parâmetros com a mesma chave *Alias*.
 Chaves *Alias* diferentes podem referenciar a mesma chave de item.
 Aliases podem ser usados em *HostMetadataItem*, mas não no parâmetro *HostnameItem*.

Exemplo 1: Recuperando o ID do usuário 'zabbix'.

```
Alias=zabbix.userid:vfs.file.regexp[/etc/passwd,"^zabbix:.:([0-9]+)",,,,\1]
```

Agora a chave abreviada **zabbix.userid** pode ser usada para recuperar dados.

Exemplo 2: Obtendo a utilização da CPU com parâmetros padrão e personalizados.

```
Alias=cpu.util:system.cpu.util
Alias=cpu.util[*]:system.cpu.util[*]
```

Isso permite usar a chave **cpu.util** para obter a porcentagem de utilização da CPU com parâmetros padrão, bem como usar **cpu.util[all, idle, avg15]** para obter dados específicos sobre a utilização da CPU.

Exemplo 3: Executando várias regras de descoberta de baixo nível processando os mesmos itens de descoberta.

```
Alias=vfs.fs.discovery[*]:vfs.fs.discovery
```

Agora é possível configurar várias regras de descoberta usando **vfs.fs.discovery** com parâmetros diferentes para cada regra, por exemplo, **vfs.fs.discovery[foo]**, **vfs.fs.discovery[bar]**, etc.

AllowKey

Permite a execução das chaves de item que correspondam a um padrão. O padrão de chave é uma expressão curinga que suporta o caractere "*" para corresponder a qualquer número de quaisquer caracteres.
Várias regras de correspondência de chaves podem ser definidas em combinação com DenyKey. Os parâmetros são processados um a um de acordo com a ordem de sua aparição. Veja também: Restringindo verificações do agent.

BufferSend

O intervalo de tempo em segundos que determina com que frequência os valores são enviados do buffer para o Zabbix server. Observe que, se o buffer estiver cheio, os dados serão enviados mais cedo.

Padrão: 5
 Intervalo: 1-3600

BufferSize

O número máximo de valores no buffer de memória. O agent enviará todos os dados coletados para o Zabbix server ou proxy se o buffer estiver cheio. Este parâmetro só deve ser usado se o buffer persistente estiver desabilitado (*EnablePersistentBuffer=0*).

Padrão: 1000
 Intervalo: 2-65535

ControlSocket

O soquete de controle, usado para enviar comandos de tempo de execução com a opção '-R'.

Padrão: `/tmp/agent.sock`

DebugLevel

Especifique o nível de depuração:
0 - informações básicas sobre a inicialização e a interrupção dos processos do Zabbix
1 - informações críticas;
2 - informações sobre erros;
3 - avisos;
4 - para depuração (produz muitas informações);
5 - depuração estendida (produz ainda mais informações).

Padrão: 3
 Intervalo: 0-5

DenyKey

Nega a execução das keys de item que correspondam a um padrão. O padrão da key é uma expressão curinga que suporta o caractere "*" para corresponder a qualquer número de caracteres.
Várias regras de correspondência de key podem ser definidas em combinação com AllowKey. Os parâmetros são processados um por um de acordo com sua ordem de aparição. Veja também: Restringindo verificações do agent.

EnablePersistentBuffer

Habilita o uso de armazenamento persistente local para items ativos. Se o armazenamento persistente estiver desabilitado, o buffer de memória será usado.

Padrão: 0
 Valores: 0 - desabilitado, 1 - habilitado

ForceActiveChecksOnStart

Executa verificações ativas imediatamente após a reinicialização para a primeira configuração recebida. Também disponível como um parâmetro de configuração por plugin, por exemplo: `Plugins.Uptime.System.ForceActiveChecksOnStart=1`

Padrão: 0
 Valores: 0 - desabilitado, 1 - habilitado

HeartbeatFrequency

A frequência das mensagens de heartbeat em segundos. Usado para monitorar a disponibilidade de verificações ativas.
0 - mensagens de heartbeat desabilitadas.

Padrão: 60
 Intervalo: 0-3600

HostInterface

Um parâmetro opcional que define a interface do host (endereço IP ou nome DNS) usada durante o processo de autoregistro do host. Esse valor será usado para preencher a interface no host recém-criado e permite configurar explicitamente o endereço IP ou DNS. Para mais detalhes, consulte Usando DNS como interface padrão.

Se não for definido, o valor será obtido de HostInterfaceItem.

O agent emitirá um erro e não iniciará se o valor ultrapassar o limite de 255 caracteres.

Intervalo: 0-255 caracteres

HostInterfaceItem

Um parâmetro opcional que define um item usado para determinar a interface do host (endereço IP ou nome DNS) durante o processo de autoregistro do host. Este valor é usado apenas se HostInterface não estiver definido. Para mais detalhes, consulte Usando DNS como interface padrão.

Durante uma solicitação de autoregistro, o agent registrará uma mensagem de aviso se o valor retornado pelo item especificado ultrapassar o limite de 255 caracteres.

O item system.run[] é suportado independentemente das configurações AllowKey/DenyKey.

HostMetadata

Um parâmetro opcional que define os metadados usados para identificar ou distinguir o host durante o processo de autoregistro do host. HostMetadata permite distinguir entre hosts além do nome do host.

Se não for definido, o valor será adquirido de HostMetadataItem.

O agent emitirá um erro e não iniciará se o valor especificado exceder o limite de 2034 bytes ou for uma string não UTF-8. Quando um parâmetro espera um endereço IP ou nome DNS, valores que são UTF-8 válidos, mas não são IPs ou nomes DNS válidos, também serão rejeitados e relatados como inválidos.

Intervalo: 0-2034 bytes

HostMetadataItem

Um parâmetro opcional que define um item do agent Zabbix usado para obter os metadados do host. Esta opção só é usada quando HostMetadata não está definido.

O valor de HostMetadataItem é obtido em cada tentativa de autoregistro e é usado apenas no processo de autoregistro do host. HostMetadataItem permite distinguir entre hosts além do nome do host.

Parâmetros de usuário e aliases são suportados. O item system.run[] é suportado independentemente das configurações AllowKey/DenyKey.

Durante uma solicitação de autoregistro, o agent registrará uma mensagem de aviso se o valor retornado pelo item especificado exceder o limite de 65535 pontos de código UTF-8. O valor retornado pelo item deve ser uma string UTF-8, caso contrário, será ignorado. Quando um parâmetro espera um endereço IP ou nome DNS, valores que são UTF-8 válidos, mas não são IPs ou nomes DNS válidos, também serão rejeitados e relatados como inválidos.

Hostname

Uma lista de nomes de host exclusivos, sensíveis a maiúsculas e minúsculas, separados por vírgulas. Necessário para verificações ativas e deve corresponder aos nomes de host configurados no server. O valor é obtido de HostnameItem se não estiver definido.
Caracteres permitidos: alfanuméricos, '.', ' ', '_' e '-'. Comprimento máximo: 128 caracteres por nome de host, 2048 caracteres para toda a linha.

Padrão: Definido por HostnameItem

HostnameItem

Um parâmetro opcional que define um item usado para obter o nome do host. Esta opção só é usada quando Hostname não está definido. Parâmetros de usuário ou aliases não são suportados, mas o item system.run[] é suportado independentemente dos valores AllowKey/DenyKey.

Padrão: `system.hostname`

Include

Você pode incluir arquivos individuais ou todos os arquivos em um diretório no arquivo de configuração. Durante a instalação, o Zabbix criará o diretório de inclusão em /usr/local/etc, a menos que seja modificado durante a compilação. O caminho pode ser relativo à localização do arquivo *zabbix_agent2.conf*.
Para incluir apenas arquivos relevantes no diretório especificado, o caractere curinga asterisco é suportado para correspondência de padrões.
Veja notas especiais sobre limitações.

Exemplo:

```
Include=/absolute/path/to/config/files/*.conf
```

ListenIP

Uma lista de endereços IP separados por vírgula nos quais o agent deve escutar. O primeiro endereço IP é enviado ao Zabbix server, se estiver conectando a ele, para recuperar a lista de verificações ativas.

Padrão: `0.0.0.0`

ListenPort

O agente escutará nessa porta as conexões do servidor.

Padrão: 10050
 Intervalo: 1024-32767

LogFile

O nome do arquivo de registro.

Padrão: `/tmp/zabbix_agent2.log`
 Obrigatório: Sim, se LogType estiver definido como *file*; caso contrário, não

LogFileSize

O tamanho máximo de um arquivo de log em MB.
0 - desativa a rotação automática de logs.
Nota: Se o limite de tamanho do arquivo de log for atingido e a rotação do arquivo falhar, por qualquer motivo, o arquivo de log existente será truncado e iniciado novamente.

Padrão: 1
 Intervalo: 0-1024

LogType

O tipo de saída de registro:
file - grava o registro no arquivo especificado pelo parâmetro LogFile;
system - grava o registro no syslog;
console - grava o registro na saída padrão

Padrão: `file`

PersistentBufferFile

O arquivo onde o agent Zabbix 2 deve manter o banco de dados SQLite. Deve ser um nome de arquivo completo. Este parâmetro só é usado se o buffer persistente estiver habilitado (*EnablePersistentBuffer=1*).

PersistentBufferPeriod

O período de tempo durante o qual os dados devem ser armazenados quando não houver conexão com o server ou proxy. Os dados mais antigos serão perdidos. Os dados de log serão preservados. Este parâmetro só é usado se o buffer persistente estiver ativado (*EnablePersistentBuffer=1*).

Padrão: 1h
 Intervalo: 1m-365d

PidFile

O nome do arquivo PID.

Padrão: `/tmp/zabbix_agent2.pid`

Plugins.<PluginName>.System.Capacity

O limite de verificações por plugin <PluginName> que podem ser executadas ao mesmo tempo.

Padrão: 1000 Intervalo: 1-1000

Plugins.Log.MaxLinesPerSecond

O número máximo de novas linhas que o agent enviará por segundo para o Zabbix server ou proxy ao processar as verificações ativas 'log' e 'logrt'. O valor fornecido será substituído pelo parâmetro 'maxlines', fornecido na chave do item 'log' e 'logrt'.
Nota: O Zabbix processará 10 vezes mais novas linhas do que o definido em *MaxLinesPerSecond* para buscar a string necessária nos itens de log.

Padrão: 20
 Intervalo: 1-1000

Plugins.SystemRun.LogRemoteCommands

Habilita o registro dos comandos shell executados como avisos. Os comandos serão registrados apenas se executados remotamente. As entradas de log não serão criadas se system.run[] for executado localmente pelos parâmetros HostMetadataItem, HostInterfaceItem ou HostnameItem.

Padrão: 0
 Valores: 0 - desabilitado, 1 - habilitado

PluginSocket

O caminho para o soquete UNIX para comunicações de plug-in carregáveis.

Padrão: `/tmp/agent.plugin.sock`

PluginTimeout

O tempo limite para conexões com plug-ins carregáveis, em segundos.

Padrão: `Timeout`
 Intervalo: 1-30

RefreshActiveChecks

Com que frequência a lista de verificações ativas é atualizada, em segundos. Observe que, após falhar ao atualizar as verificações ativas, a próxima atualização será tentada em 60 segundos.

Padrão: 5
 Intervalo: 1-86400

Server

Uma lista de endereços IP separados por vírgula, opcionalmente na notação CIDR, ou nomes DNS de servidores Zabbix ou proxies Zabbix. Conexões de entrada serão aceitas apenas dos hosts listados aqui. Se o suporte a IPv6 estiver habilitado, então '127.0.0.1', '::127.0.0.1', '::ffff:127.0.0.1' são tratados igualmente e '::/0' permitirá qualquer endereço IPv4 ou IPv6. '0.0.0.0/0' pode ser usado para permitir qualquer endereço IPv4. Espaços são permitidos. Se este parâmetro não for especificado, desabilita as verificações passivas e o agent não irá escutar em nenhuma porta TCP.

Exemplo:

```
Server=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com
```

ServerActive

O endereço do Zabbix server/proxy ou a configuração do cluster para obter verificações ativas. O endereço do server/proxy é um endereço IP ou nome DNS e uma porta opcional separada por dois pontos.
A configuração do cluster é um ou mais endereços de membros do grupo de server ou proxy separados por ponto e vírgula. Vários Zabbix servers/clusters e Zabbix proxies podem ser especificados, separados por vírgula. A menos que esteja usando grupos de proxy, não deve ser especificado mais de um Zabbix proxy de cada Zabbix server/cluster. Se um Zabbix proxy for especificado, o Zabbix server/cluster para esse proxy não deve ser

especificado.
Vários endereços separados por vírgula podem ser fornecidos para usar vários Zabbix servers independentes em paralelo. Espaços são permitidos.
Se a porta não for especificada, a porta padrão será usada.
Endereços IPv6 devem ser colocados entre colchetes se a porta para esse host for especificada. Se a porta não for especificada, os colchetes para endereços IPv6 são opcionais.
Se este parâmetro não for especificado, as verificações ativas serão desativadas.

Exemplo para Zabbix proxy:

```
ServerActive=127.0.0.1:10051
```

Exemplo para grupo de Zabbix proxy:

```
ServerActive=proxy1.example.com;proxy2.example.com;proxy3.example.com;proxy4.example.com;proxy5.example.co
```

Exemplo para múltiplos servers:

```
ServerActive=127.0.0.1:20051,zabbix.domain,[::1]:30051,::1,[12fc::1]
```

Exemplo para alta disponibilidade:

```
ServerActive=zabbix.cluster.node1;zabbix.cluster.node2:20051;zabbix.cluster.node3
```

Exemplo para alta disponibilidade com dois clusters e um server:

```
ServerActive=zabbix.cluster.node1;zabbix.cluster.node2:20051,zabbix.cluster2.node1;zabbix.cluster2.node2,z
```

SourceIP

O endereço IP de origem para:

- conexões de saída para o Zabbix server ou Zabbix proxy.
- fazer conexões ao executar alguns items (web.page.get, net.tcp.port, etc.).

StatusPort

Se definido, o agente escutará nessa porta as solicitações de status HTTP (http://localhost:<port>/status).

Intervalo: 1024-32767

Timeout

Especifica quanto tempo esperar (em segundos) para estabelecer conexão e trocar dados com o Zabbix proxy ou server.

Este parâmetro define a duração de várias operações de comunicação, incluindo:

- aguardar uma resposta do Zabbix server;
- enviar solicitações ao Zabbix server, incluindo solicitações de configuração de item e dados de item em verificações ativas;
- recuperar dados de log através do logfile;
- enviar mensagens de heartbeat;
- duração máxima para verificações `vfs.*`;
- servir como fallback em cenários onde um server ou proxy anterior à versão 7.0 envia verificações sem timeouts.

Este timeout **não** será usado para aquelas verificações de agent que possuem configurações de timeout configuráveis no frontend (em nível global, proxy ou por item).

Padrão: 3
 Intervalo: 1-30

TLSAccept

As conexões de entrada a serem aceitas. Usado para verificações passivas. Vários valores podem ser especificados, separados por vírgula:
unencrypted - aceita conexões sem criptografia (padrão)
psk - aceita conexões com TLS e uma chave pré-compartilhada (PSK)
cert - aceita conexões com TLS e um certificado

Obrigatório: sim, se os parâmetros de certificado TLS ou PSK forem definidos (mesmo para conexão *unencrypted*); caso contrário, não

TLSCAFile

O nome completo do caminho do arquivo que contém os certificados da(s) CA(s) de nível superior para verificação de certificados de pares, usados para comunicações criptografadas entre os componentes do Zabbix.

TLSCertFile

O nome completo do caminho do arquivo que contém o certificado do agente ou a cadeia de certificados, usado para comunicações criptografadas com os componentes do Zabbix.

TLSCipherAll

A string de cifra do OpenSSL (TLS 1.2). Substitui os critérios padrão de seleção de ciphersuite para criptografia baseada em certificado e PSK.

Exemplo:

```
EECDH+aRSA+AES128:RSA+aRSA+AES128:kECDHEPSK+AES128:kPSK+AES128
```

Este parâmetro é suportado a partir do Zabbix 7.0.4.

TLSCipherAll13

A string de cifra do OpenSSL (TLS 1.3) para OpenSSL 1.1.1 ou mais recente. Substitui os critérios padrão de seleção de ciphersuite para criptografia baseada em certificado e PSK.

Exemplo:

```
TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

Este parâmetro é suportado desde o Zabbix 7.0.4.

TLSCipherCert

A string de cifra do OpenSSL (TLS 1.2). Substitui os critérios padrão de seleção de ciphersuite para criptografia baseada em certificado.

Exemplo:

```
EECDH+aRSA+AES128:RSA+aRSA+AES128
```

Observe que este parâmetro não pode ser usado junto com `TLSAccept=cert,psk`; para conexão por certificado (`TLSConnect=cert`) use TLSCipherAll.

Este parâmetro é suportado a partir do Zabbix 7.0.4.

TLSCipherCert13

A string de cifra do OpenSSL (TLS 1.3) para OpenSSL 1.1.1 ou mais recente. Substitui os critérios padrão de seleção de ciphersuite para criptografia baseada em certificado.

Observe que este parâmetro não pode ser usado junto com `TLSAccept=cert,psk`; para conexão por certificado (`TLSConnect=cert`) use TLSCipherAll13 em vez disso.

Este parâmetro é suportado desde o Zabbix 7.0.4.

TLSCipherPSK

A string de cifra do OpenSSL (TLS 1.2). Substitui os critérios de seleção do conjunto de cifras padrão para criptografia baseada em PSK.

Exemplo:

```
kECDHEPSK+AES128:kPSK+AES128
```

Este parâmetro é suportado a partir do Zabbix 7.0.4.

TLSCipherPSK13

A string de cifra do OpenSSL (TLS 1.3) para OpenSSL 1.1.1 ou mais recente. Substitui os critérios padrão de seleção de ciphersuite para criptografia baseada em PSK.

Exemplo:

```
TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

Este parâmetro é suportado desde o Zabbix 7.0.4.

TLSConnect

Como o agent deve se conectar ao Zabbix server ou proxy. Usado para verificações ativas. Apenas um valor pode ser especificado:
unencrypted - conectar sem criptografia (padrão)
psk - conectar usando TLS e uma chave pré-compartilhada (PSK)
cert - conectar usando TLS e um certificado

Obrigatório: sim, se os parâmetros de certificado TLS ou PSK forem definidos (mesmo para conexão *unencrypted*); caso contrário, não

TLSCRLFile

O caminho completo do arquivo contendo certificados revogados. Este parâmetro é usado para comunicações criptografadas entre os componentes do Zabbix.

TLSKeyFile

O nome completo do caminho do arquivo que contém a chave privada do agente, usada para comunicações criptografadas entre os componentes do Zabbix.

TLSPSKFile

O nome completo do caminho do arquivo que contém a chave pré-compartilhada do agente, usada para comunicações criptografadas com o servidor Zabbix.

TLSPSKIdentity

A string de identidade da chave pré-compartilhada, usada para comunicações criptografadas com o servidor Zabbix.

TLSServerCertIssuer

O emissor do certificado do servidor (proxy) permitido.

TLSServerCertSubject

O assunto do certificado do servidor (proxy) permitido.

UnsafeUserParameters

Permitir que todos os caracteres sejam passados como argumentos para parâmetros definidos pelo usuário. Os seguintes caracteres não são permitidos: \ ' " ` * ? [ ] { } ~ $! & ; ( ) < > | # @
Além disso, caracteres de nova linha não são permitidos.

Padrão: 0
 Valores: 0 - não permitir, 1 - permitir

UserParameter

Um parâmetro definido pelo usuário para monitoramento. Pode haver vários parâmetros definidos pelo usuário.
Formato: UserParameter=<key>,<comando shell>
Observe que o comando shell não deve retornar uma string vazia ou apenas EOL. Comandos shell podem ter caminhos relativos, se o parâmetro UserParameterDir for especificado.

Exemplo:

```
UserParameter=system.test,who|wc -l
UserParameter=check_cpu,./custom_script.sh
```

UserParameterDir

O caminho de pesquisa padrão para comandos UserParameter. Se usado, o agent mudará seu diretório de trabalho para o especificado aqui antes de executar um comando. Assim, os comandos UserParameter podem ter um prefixo relativo `./` em vez de um caminho completo.
Apenas uma entrada é permitida.

Exemplo:

```
UserParameterDir=/opt/myscripts
```

5 Zabbix agent (Windows)

Visão geral

Os parâmetros suportados pelo arquivo de configuração do agent Zabbix para Windows (zabbix_agentd.conf) estão listados nesta seção.

Os parâmetros são listados sem informações adicionais. Clique no parâmetro para ver os detalhes completos.

Parâmetro	Descrição
Alias	Define um alias para uma chave de item.
AllowKey	Permite a execução das chaves de item que correspondem a um padrão.
BufferSend	Não manter dados no buffer por mais de N segundos.
BufferSize	O número máximo de valores no buffer de memória.
DebugLevel	O nível de debug.
DenyKey	Nega a execução das chaves de item que correspondem a um padrão.
EnableRemoteCommands	Se comandos remotos do Zabbix server são permitidos.
HeartbeatFrequency	A frequência das mensagens de heartbeat em segundos.
HostInterface	Um parâmetro opcional que define a interface do host.
HostInterfaceItem	Um parâmetro opcional que define um item usado para obter a interface do host.
HostMetadata	Um parâmetro opcional que define os metadados do host.

Parâmetro	Descrição
HostMetadataItem	Um parâmetro opcional que define um item do agent Zabbix usado para obter os metadados do host.
Hostname	Um parâmetro opcional que define o nome do host.
HostnameItem	Um parâmetro opcional que define um item do agent Zabbix usado para obter o nome do host.
Include	Você pode incluir arquivos individuais ou todos os arquivos em um diretório no arquivo de configuração.
ListenBacklog	O número máximo de conexões pendentes na fila TCP.
ListenIP	Uma lista de endereços IP separados por vírgula nos quais o agent deve escutar.
ListenPort	O agent irá escutar nesta porta por conexões do server.
LogFile	O nome do arquivo de log.
LogFileSize	O tamanho máximo do arquivo de log.
LogRemoteCommands	Habilita o registro de comandos shell executados como avisos.
LogType	O tipo de saída do log.
MaxLinesPerSecond	O número máximo de novas linhas que o agent enviará por segundo para o Zabbix server ou proxy ao processar verificações ativas 'log' e 'logrt'.
PerfCounter	Define um novo parâmetro <parameter_name> que é o valor médio para o contador de desempenho do sistema <perf_counter_path> para o período de tempo especificado <period> (em segundos).
PerfCounterEn	Define um novo parâmetro <parameter_name> que é o valor médio para o contador de desempenho do sistema <perf_counter_path> para o período de tempo especificado <period> (em segundos). Em comparação com PerfCounter, os caminhos do perfcounter devem estar em inglês.
RefreshActiveChecks	Com que frequência a lista de verificações ativas é atualizada.
Server	Uma lista de endereços IP separados por vírgula, opcionalmente na notação CIDR, ou nomes DNS dos Zabbix servers e Zabbix proxies.
ServerActive	O endereço do Zabbix server/proxy ou configuração de cluster para obter verificações ativas.
SourceIP	O endereço IP de origem.
StartAgents	O número de instâncias pré-forked do zabbix_agentd que processam verificações passivas.
Timeout	Especifica quanto tempo esperar (em segundos) para estabelecer conexão e trocar dados com o Zabbix proxy ou server.
TLSAccept	Quais conexões de entrada aceitar.
TLSCAFile	O caminho completo de um arquivo contendo os certificados da(s) CA(s) de nível superior para verificação do certificado do par, usado para comunicações criptografadas entre componentes do Zabbix.
TLSCertFile	O caminho completo de um arquivo contendo o certificado do agent ou cadeia de certificados, usado para comunicações criptografadas entre componentes do Zabbix.
TLSConnect	Como o agent deve se conectar ao Zabbix server ou proxy.
TLSCRLFile	O caminho completo de um arquivo contendo certificados revogados. Este parâmetro é usado para comunicações criptografadas entre componentes do Zabbix.
TLSKeyFile	O caminho completo de um arquivo contendo a chave privada do agent, usado para comunicações criptografadas entre componentes do Zabbix.
TLSPSKFile	O caminho completo de um arquivo contendo a chave pré-compartilhada do agent, usado para comunicações criptografadas com o Zabbix server.
TLSPSKIdentity	A string de identidade da chave pré-compartilhada, usada para comunicações criptografadas com o Zabbix server.
TLSServerCertIssuer	O emissor do certificado do server (proxy) permitido.
TLSServerCertSubject	O assunto do certificado do server (proxy) permitido.
UnsafeUserParameters	Permite que todos os caracteres sejam passados como argumentos para parâmetros definidos pelo usuário.
UserParameter	Um parâmetro definido pelo usuário para monitorar.
UserParameterDir	O caminho de pesquisa padrão para comandos UserParameter.

Todos os parâmetros são não obrigatórios, a menos que seja explicitamente declarado que o parâmetro é obrigatório.

Observe que:

- Os valores padrão refletem os padrões do daemon, não os valores nos arquivos de configuração fornecidos;
- O Zabbix suporta arquivos de configuração apenas na codificação UTF-8 sem BOM;
- Comentários iniciados com "#" são suportados apenas no início da linha.

Detalhes do parâmetro

Alias

Define um alias para uma chave de item. Pode ser usado para substituir uma chave de item longa e complexa por uma mais curta e simples.
 Vários parâmetros *Alias* podem estar presentes. Não são permitidos vários parâmetros com a mesma chave *Alias*.
 Chaves *Alias* diferentes podem referenciar a mesma chave de item.
 Aliases podem ser usados em *HostMetadataItem*, mas não nos parâmetros *HostnameItem* ou *PerfCounter*.

Exemplo 1: Recuperando o uso do arquivo de paginação em porcentagem do servidor.

```
Alias=pg_usage:perf_counter[\Paging File(_Total)\% Usage]
```

Agora a chave abreviada **pg_usage** pode ser usada para recuperar dados.

Exemplo 2: Obtendo a carga da CPU com parâmetros padrão e personalizados.

```
Alias=cpu.load:system.cpu.load
Alias=cpu.load[*]:system.cpu.load[*]
```

Isso permite usar a chave **cpu.load** para obter a carga da CPU com parâmetros padrão, bem como usar **cpu.load[percpu,avg15]** para obter dados específicos sobre a carga da CPU.

Exemplo 3: Executando várias regras de descoberta de baixo nível processando os mesmos itens de descoberta.

```
Alias=vfs.fs.discovery[*]:vfs.fs.discovery
```

Agora é possível configurar várias regras de descoberta usando **vfs.fs.discovery** com parâmetros diferentes para cada regra, por exemplo, **vfs.fs.discovery[foo]**, **vfs.fs.discovery[bar]**, etc.

AllowKey

Permite a execução das chaves de item que correspondam a um padrão. O padrão de chave é uma expressão curinga que suporta o caractere ”*” para corresponder a qualquer número de quaisquer caracteres.
Várias regras de correspondência de chaves podem ser definidas em combinação com DenyKey. Os parâmetros são processados um a um de acordo com sua ordem de aparição. Veja também: Restringindo verificações do agent.

BufferSend

Não mantenha dados no buffer por mais de N segundos.

Padrão: 5
 Intervalo: 1-3600

BufferSize

O número máximo de valores no buffer de memória. O agent enviará todos os dados coletados para o Zabbix server ou proxy se o buffer estiver cheio.

Padrão: 100
 Intervalo: 2-65535

DebugLevel

Especifica o nível de depuração:
0 - informações básicas sobre o início e a parada dos processos do Zabbix
1 - informações críticas;
2 - informações de erro;
3 - avisos;
4 - para depuração (gera muitas informações);
5 - depuração estendida (gera ainda mais informações).

Padrão: 3
 Intervalo: 0-5

DenyKey

Nega a execução das keys de item que correspondam a um padrão. O padrão de key é uma expressão curinga que suporta o caractere ”*” para corresponder a qualquer número de quaisquer caracteres.
Várias regras de correspondência de key podem ser definidas em combinação com AllowKey. Os parâmetros são processados um por um de acordo com sua ordem de aparição. Veja também: Restringindo verificações do agent.

EnableRemoteCommands

Se comandos remotos do Zabbix server são permitidos. Este parâmetro está **obsoleto**, use AllowKey=system.run[*] ou DenyKey=system.run[*] em vez disso.
É um alias interno para os parâmetros AllowKey/DenyKey dependendo do valor:
0 - DenyKey=system.run[*]
1 - AllowKey=system.run[*]

Padrão: 0
 Valores: 0 - não permitir, 1 - permitir

HeartbeatFrequency

A frequência das mensagens de heartbeat em segundos. Usado para monitorar a disponibilidade de verificações ativas.
0 - mensagens de heartbeat desabilitadas.

Padrão: 60
 Intervalo: 0-3600

HostInterface

Um parâmetro opcional que define a interface do host (endereço IP ou nome DNS) usada durante o processo de autoregistro do host. Esse valor será usado para preencher a interface no host recém-criado e permite configurar explicitamente o endereço IP ou DNS. Para mais detalhes, consulte Usando DNS como interface padrão.

Se não for definido, o valor será adquirido de HostInterfaceItem.

O agent emitirá um erro e não iniciará se o valor ultrapassar o limite de 255 caracteres.

Intervalo: 0-255 caracteres

HostInterfaceItem

Um parâmetro opcional que define um item usado para determinar a interface do host (endereço IP ou nome DNS) durante o processo de autoregistro do host. Esse valor é usado apenas se HostInterface não estiver definido. Para mais detalhes, consulte Usando DNS como interface padrão.

Durante uma solicitação de autoregistro, o agent registrará uma mensagem de aviso se o valor retornado pelo item especificado exceder o limite de 255 caracteres.

O item system.run[] é suportado independentemente das configurações AllowKey/DenyKey.

HostMetadata

Um parâmetro opcional que define os metadados usados para identificar ou distinguir o host durante o processo de autoregistro do host (agent ativo). HostMetadata permite distinguir entre hosts além do nome do host.

Se não for definido, o valor será adquirido de HostMetadataItem.

O agent emitirá um erro e não iniciará se o valor especificado ultrapassar o limite de 2034 bytes ou for uma string que não seja UTF-8. Quando um parâmetro espera um endereço IP ou nome DNS, valores que são UTF-8 válidos, mas não são IPs ou nomes DNS válidos, também serão rejeitados e relatados como inválidos.

Metadados de várias linhas não são suportados — a saída será truncada na primeira quebra de linha.

Intervalo: 0-2034 bytes

HostMetadataItem

Um parâmetro opcional que define um item do agent Zabbix usado para obter os metadados do host. Esta opção só é usada quando HostMetadata não está definido.

O valor de HostMetadataItem é obtido em cada tentativa de autoregistro e é usado apenas no processo de autoregistro do host (agent ativo). HostMetadataItem permite distinguir entre hosts além do nome do host.

Parâmetros de usuário, contadores de desempenho e aliases são suportados. O item system.run[] é suportado independentemente das configurações AllowKey/DenyKey.

Durante uma solicitação de autoregistro, o agent registrará uma mensagem de aviso se o valor retornado pelo item especificado exceder o limite de 65535 pontos de código UTF-8. O valor retornado pelo item deve ser uma string UTF-8, caso contrário, será ignorado. Quando um parâmetro espera um endereço IP ou nome DNS, valores que são UTF-8 válidos, mas não são IPs ou nomes DNS válidos, também serão rejeitados e relatados como inválidos.

Metadados de múltiplas linhas não são suportados — a saída será truncada na primeira quebra de linha.

Hostname

Uma lista de nomes de host exclusivos, sensíveis a maiúsculas e minúsculas, separados por vírgulas. Necessário para verificações ativas e deve corresponder aos nomes de host configurados no server. O valor é obtido de HostnameItem se não for definido.
Caracteres permitidos: alfanuméricos, '.', ' ', '_' e '-'. Comprimento máximo: 128 caracteres por nome de host, 2048 caracteres para toda a linha.

Padrão: Definido por HostnameItem

HostnameItem

Um parâmetro opcional que define um item do agent Zabbix usado para obter o nome do host. Esta opção só é usada quando Hostname não está definido. Parâmetros de usuário, contadores de desempenho ou aliases não são suportados, mas o item system.run[] é suportado independentemente dos valores AllowKey/DenyKey.
Veja também uma descrição mais detalhada.

Padrão: `system.hostname`

Include

Você pode incluir arquivos individuais ou todos os arquivos em um diretório no arquivo de configuração (localizado em `C:\Program Files\Zabbix Agent` por padrão se o agent Zabbix for instalado usando os pacotes do instalador MSI do Windows; localizado na pasta especificada durante a instalação se o agent Zabbix for instalado como um arquivo zip). Todos os arquivos incluídos devem

ter sintaxe correta, caso contrário o agent não iniciará.
Para incluir apenas arquivos relevantes no diretório especificado, o caractere curinga asterisco é suportado para correspondência de padrões.
Veja notas especiais sobre limitações.

Exemplo:

```
Include=C:\Program Files\Zabbix Agent\zabbix_agentd.d\*.conf
```

ListenBacklog

O número máximo de conexões pendentes na fila TCP.
O valor padrão é uma constante codificada, que depende do sistema.
O valor máximo suportado depende do sistema, valores muito altos podem ser truncados silenciosamente para o 'máximo especificado pela implementação'.

Padrão: SOMAXCONN
 Intervalo: 0 - INT_MAX

ListenIP

Uma lista de endereços IP separados por vírgula nos quais o agent deve escutar.

Padrão: 0.0.0.0

ListenPort

O agent irá escutar nesta porta por conexões do server.

Padrão: 10050
 Intervalo: 1024-32767

LogFile

O nome do arquivo de log do agent.

Padrão: c:\zabbix_agentd.log
 Obrigatório: Sim, se LogType estiver definido como *file*; caso contrário, não

LogFileSize

O tamanho máximo de um arquivo de log em MB.
0 - desabilita a rotação automática de logs.
Nota: Se o limite de tamanho do arquivo de log for atingido e a rotação do arquivo falhar, por qualquer motivo, o arquivo de log existente será truncado e iniciado novamente.

Padrão: 1
 Intervalo: 0-1024

LogRemoteCommands

Habilita o registro dos comandos shell executados como avisos. Os comandos serão registrados apenas se executados remotamente. As entradas de log não serão criadas se system.run[] for iniciado localmente pelos parâmetros HostMetadataItem, HostInterfaceItem ou HostnameItem.

Padrão: 0
 Valores: 0 - desabilitado, 1 - habilitado

LogType

O tipo de saída do log:
file - grava o log no arquivo especificado pelo parâmetro LogFile;
system - grava o log no Windows Event Log;
console - grava o log na saída padrão.

Padrão: file

MaxLinesPerSecond

O número máximo de novas linhas que o agent enviará por segundo para o Zabbix server ou proxy ao processar verificações ativas 'log', 'logrt' e 'eventlog'. O valor fornecido será substituído pelo parâmetro 'maxlines', fornecido na chave do item 'log', 'logrt' ou 'eventlog'.
Nota: O Zabbix processará 10 vezes mais novas linhas do que o definido em *MaxLinesPerSecond* para buscar a string necessária em itens de log.

Padrão: 20
 Intervalo: 1-1000

PerfCounter

Define um novo parâmetro <parameter_name> que é o valor médio para o contador de desempenho do sistema <perf_counter_path> para o período de tempo especificado <period> (em segundos).
Sintaxe: <parameter_name>,"<perf_counter_path>",<period>

Por exemplo, se você deseja receber o número médio de interrupções do processador por segundo no último minuto, pode definir um novo parâmetro "interrupts" da seguinte forma:

```
PerfCounter = interrupts,"\Processor(0)\Interrupts/sec",60
```

Observe as aspas duplas ao redor do caminho do contador de desempenho. O nome do parâmetro (interrupts) deve ser usado como a chave do item ao criar um item. As amostras para calcular o valor médio serão coletadas a cada segundo.
Você pode executar "typeperf -qx" para obter a lista de todos os contadores de desempenho disponíveis no Windows.

PerfCounterEn

Define um novo parâmetro <parameter_name> que é o valor médio para o contador de desempenho do sistema <perf_counter_path> para o período de tempo especificado <period> (em segundos). Comparado ao PerfCounter, os caminhos do perfcounter devem estar em inglês. Suportado apenas no **Windows Server 2008/Vista** e posterior.
Sintaxe: <parameter_name>,"<perf_counter_path>",<pe

Por exemplo, se você deseja receber o número médio de interrupções do processador por segundo no último minuto, pode definir um novo parâmetro "interrupts" da seguinte forma:

```
PerfCounterEn = interrupts,"\Processor(0)\Interrupts/sec",60
```

Observe as aspas duplas ao redor do caminho do contador de desempenho. O nome do parâmetro (interrupts) deve ser usado como a chave do item ao criar um item. As amostras para calcular o valor médio serão coletadas a cada segundo.
Você pode encontrar a lista de strings em inglês visualizando a seguinte chave do registro: `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Perflib\009`.

RefreshActiveChecks

Com que frequência a lista de verificações ativas é atualizada, em segundos. Observe que, após falhar ao atualizar as verificações ativas, a próxima atualização será tentada em 60 segundos.

Padrão: 5
 Intervalo: 1-86400

Server

Uma lista de endereços IP separados por vírgula, opcionalmente na notação CIDR, ou nomes DNS de servidores Zabbix ou proxies Zabbix. As conexões recebidas serão aceitas apenas dos hosts listados aqui. Se o suporte a IPv6 estiver habilitado, então '127.0.0.1', '::127.0.0.1', '::ffff:127.0.0.1' são tratados igualmente e '::/0' permitirá qualquer endereço IPv4 ou IPv6. '0.0.0.0/0' pode ser usado para permitir qualquer endereço IPv4. Observe que "endereços IPv6 compatíveis com IPv4" (prefixo 0000::/96) são suportados, mas obsoletos pelo RFC4291. Espaços são permitidos.

Exemplo:

```
Server=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com
```

Obrigatório: sim, se StartAgents não estiver explicitamente definido como 0

ServerActive

O endereço do Zabbix server/proxy ou configuração de cluster para obter verificações ativas. O endereço do server/proxy é um endereço IP ou nome DNS e porta opcional separada por dois pontos.
A configuração do cluster é um ou mais endereços de membros do grupo de server ou proxy separados por ponto e vírgula. Vários Zabbix servers/clusters e Zabbix proxies podem ser especificados, separados por vírgula. A menos que esteja usando grupos de proxy, não deve ser especificado mais de um Zabbix proxy de cada Zabbix server/cluster. Se o Zabbix proxy for especificado, então o Zabbix server/cluster para esse proxy não deve ser especificado.
Vários endereços separados por vírgula podem ser fornecidos para usar vários Zabbix servers independentes em paralelo. Espaços são permitidos.
Se a porta não for especificada, a porta padrão será usada.
Endereços IPv6 devem ser colocados entre colchetes se a porta para esse host for especificada. Se a porta não for especificada, os colchetes para endereços IPv6 são opcionais.
Se este parâmetro não for especificado, as verificações ativas serão desabilitadas.

Exemplo para Zabbix proxy:

```
ServerActive=127.0.0.1:10051
```

Exemplo para grupo de Zabbix proxy:

```
ServerActive=proxy1.example.com;proxy2.example.com;proxy3.example.com;proxy4.example.com;proxy5.example.co
```

Exemplo para múltiplos servers:

```
ServerActive=127.0.0.1:20051,zabbix.domain,[::1]:30051,::1,[12fc::1]
```

Exemplo para alta disponibilidade:

```
ServerActive=zabbix.cluster.node1;zabbix.cluster.node2:20051;zabbix.cluster.node3
```

Exemplo para alta disponibilidade com dois clusters e um server:

```
ServerActive=zabbix.cluster.node1;zabbix.cluster.node2:20051,zabbix.cluster2.node1;zabbix.cluster2.node2,z
```

Intervalo: (*)

SourceIP

O endereço IP de origem para:

- conexões de saída para o Zabbix server ou Zabbix proxy.

- fazer conexões ao executar alguns items (web.page.get, net.tcp.port, etc.).

StartAgents

O número de instâncias pré-forked do zabbix_agentd que processam verificações passivas. Se definido como 0, as verificações passivas são desativadas e o agent não irá escutar em nenhuma porta TCP.

Padrão: 10
 Intervalo: 0-100 (*)

Timeout

Especifica quanto tempo esperar (em segundos) para estabelecer conexão e trocar dados com o Zabbix proxy ou server.

Este parâmetro define a duração de várias operações de comunicação, incluindo:

- aguardar uma resposta do Zabbix server;
- enviar solicitações ao Zabbix server, incluindo solicitações de configuração de item e dados de item em verificações ativas;
- recuperar dados de log por meio do monitoramento de arquivos de log ou do log de eventos do Windows;
- enviar mensagens de heartbeat;
- duração máxima para verificações `vfs.*`;
- usado pelos módulos do Zabbix agent;
- servindo como fallback em cenários onde um server ou proxy anterior à versão 7.0 envia verificações sem timeouts.

Este timeout **não** será usado para aquelas verificações de agent que possuem configurações de timeout configuráveis no frontend (em nível global, proxy ou por item).

Padrão: 3
 Intervalo: 1-30

TLSAccept

As conexões de entrada a serem aceitas. Usado para verificações passivas. Vários valores podem ser especificados, separados por vírgula:
unencrypted - aceita conexões sem criptografia (padrão)
psk - aceita conexões com TLS e uma chave pré-compartilhada (PSK)
cert - aceita conexões com TLS e um certificado

Obrigatório: sim, se os parâmetros de certificado TLS ou PSK forem definidos (mesmo para conexão *unencrypted*); caso contrário, não

TLSCAFile

O caminho completo do arquivo contendo o(s) certificado(s) da(s) CA(s) de nível superior para verificação do certificado do par, usado para comunicações criptografadas entre os componentes do Zabbix.

TLSCertFile

O caminho completo do arquivo que contém o certificado do agent ou cadeia de certificados, usado para comunicações criptografadas com os componentes do Zabbix.

TLSConnect

Como o agent deve se conectar ao Zabbix server ou proxy. Usado para verificações ativas. Apenas um valor pode ser especificado:
unencrypted - conectar sem criptografia (padrão)
psk - conectar usando TLS e uma chave pré-compartilhada (PSK)
cert - conectar usando TLS e um certificado

Obrigatório: sim, se os parâmetros de certificado TLS ou PSK forem definidos (mesmo para conexão *unencrypted*); caso contrário, não

TLSCRLFile

O caminho completo do arquivo contendo certificados revogados. Este parâmetro é usado para comunicações criptografadas entre os componentes do Zabbix.

TLSKeyFile

O caminho completo do arquivo contendo a chave privada do agent, usada para comunicações criptografadas entre os componentes do Zabbix.

TLSPSKFile

O caminho completo do arquivo contendo a chave pré-compartilhada do agent, usada para comunicações criptografadas com o Zabbix server.

TLSPSKIdentity

A string de identidade da chave pré-compartilhada, usada para comunicações criptografadas com o Zabbix server.

TLSServerCertIssuer

O emissor do certificado do servidor (proxy) permitido.

TLSServerCertSubject

O assunto do certificado do server (proxy) permitido.

UnsafeUserParameters

Permitir que todos os caracteres sejam passados como argumentos para parâmetros definidos pelo usuário. Os seguintes caracteres não são permitidos: \ ' " ` * ? [ ] { } ~ $! & ; ( ) < > | # @
Além disso, caracteres de nova linha não são permitidos.

Padrão: 0
 Valores: 0 - não permitir, 1 - permitir

UserParameter

Um parâmetro definido pelo usuário para monitoramento. Podem haver vários parâmetros definidos pelo usuário.
Formato: UserParameter=<chave>,<comando shell>
Observe que o comando shell não deve retornar uma string vazia ou apenas EOL. Comandos shell podem ter caminhos relativos, se o parâmetro UserParameterDir for especificado.

Exemplo:

```
UserParameter=system.test,dir /b | find /c /v ""
UserParameter=check_cpu,.\custom\_script.bat
```

UserParameterDir

O caminho de pesquisa padrão para comandos UserParameter. Se usado, o agent mudará seu diretório de trabalho para o especificado aqui antes de executar um comando. Assim, os comandos UserParameter podem ter um prefixo relativo `./` em vez de um caminho completo. Apenas uma entrada é permitida.

Exemplo:

```
UserParameterDir=C:\opt\myscripts
```

Note:
(*) O número de servidores ativos listados em ServerActive mais o número de instâncias pré-forked para verificações passivas especificadas em StartAgents deve ser menor que 64.

6 Agente Zabbix 2 (Windows)

Visão geral

O Zabbix agent 2 é uma nova geração do Zabbix agent e pode ser usado no lugar do Zabbix agent.

Os parâmetros suportados pelo arquivo de configuração do Zabbix agent 2 para Windows (zabbix_agent2.conf) estão listados nesta seção.

Os parâmetros são listados sem informações adicionais. Clique no parâmetro para ver os detalhes completos.

Parâmetro	Descrição
Alias	Define um alias para uma chave de item.
AllowKey	Permite a execução das chaves de item que correspondem a um padrão.
BufferSend	Não manter dados por mais de N segundos no buffer.
BufferSize	O número máximo de valores no buffer de memória.
ControlSocket	O socket de controle, usado para enviar comandos em tempo de execução com a opção '-R'.
DebugLevel	O nível de debug.
DenyKey	Nega a execução das chaves de item que correspondem a um padrão.
EnablePersistentBuffer	Habilita o uso de armazenamento persistente local para itens ativos.
ForceActiveChecksOnStart	Executa checagens ativas imediatamente após o reinício para a primeira configuração recebida.
HeartbeatFrequency	A frequência das mensagens de heartbeat em segundos.
HostInterface	Um parâmetro opcional que define a interface do host.
HostInterfaceItem	Um parâmetro opcional que define um item usado para obter a interface do host.
HostMetadata	Um parâmetro opcional que define os metadados do host.
HostMetadataItem	Um parâmetro opcional que define um item do Zabbix agent usado para obter os metadados do host.
Hostname	Um parâmetro opcional que define o nome do host.
HostnameItem	Um parâmetro opcional que define um item do Zabbix agent usado para obter o nome do host.
Include	Você pode incluir arquivos individuais ou todos os arquivos em um diretório no arquivo de configuração.

Parâmetro	Descrição
ListenIP	Uma lista de endereços IP separados por vírgula nos quais o agent deve escutar.
ListenPort	O agent irá escutar nesta porta por conexões do server.
LogFile	O nome do arquivo de log.
LogFileSize	O tamanho máximo do arquivo de log.
LogType	O tipo de saída do log.
PersistentBufferFile	O arquivo onde o Zabbix agent 2 deve manter o banco de dados SQLite.
PersistentBufferPeriod	O período de tempo pelo qual os dados devem ser armazenados quando não houver conexão com o server ou proxy.
Plugins.<PluginName>.SystemCapacity	O limite de checagens por plugin que podem ser executadas ao mesmo tempo.
Plugins.Log.MaxLinesPerSecond	O número máximo de novas linhas que o agent enviará por segundo para o Zabbix server ou proxy ao processar checagens ativas 'log' e 'logrt'.
Plugins.SystemRun.LogRemoteCommands	Habilita o registro dos comandos shell executados como avisos.
PluginSocket	O caminho para o socket UNIX para comunicações de plugins carregáveis.
PluginTimeout	O tempo limite para conexões com plugins carregáveis, em segundos.
PerfCounter	Define um novo parâmetro <parameter_name> que é o valor médio para o contador de desempenho do sistema <perf_counter_path> para o período de tempo especificado <period> (em segundos).
PerfCounterEn	Define um novo parâmetro <parameter_name> que é o valor médio para o contador de desempenho do sistema <perf_counter_path> para o período de tempo especificado <period> (em segundos). Em comparação com PerfCounter, os caminhos perfcounter devem estar em inglês.
RefreshActiveChecks	Com que frequência a lista de checagens ativas é atualizada.
Server	Uma lista de endereços IP separados por vírgula, opcionalmente em notação CIDR, ou nomes DNS dos Zabbix servers e Zabbix proxies.
ServerActive	O endereço do Zabbix server/proxy ou configuração de cluster para obter checagens ativas.
SourceIP	O endereço IP de origem.
StatusPort	Se definido, o agent irá escutar nesta porta por requisições de status HTTP (http://localhost:<port>/status).
Timeout	Especifica quanto tempo esperar (em segundos) para estabelecer conexão e trocar dados com o Zabbix proxy ou server.
TLSAccept	Quais conexões de entrada aceitar.
TLSCAFile	O caminho completo de um arquivo contendo os certificados CA(s) de nível superior para verificação de certificado de peer, usado para comunicações criptografadas entre componentes do Zabbix.
TLSCertFile	O caminho completo de um arquivo contendo o certificado do agent ou cadeia de certificados, usado para comunicações criptografadas entre componentes do Zabbix.
TLSCipherAll13	A string de cifra do OpenSSL (TLS 1.3). Substitui os critérios padrão de seleção de ciphersuite para criptografia baseada em certificado e PSK.
TLSCipherCert13	A string de cifra do OpenSSL (TLS 1.3). Substitui os critérios padrão de seleção de ciphersuite para criptografia baseada em certificado.
TLSCipherPSK13	A string de cifra do OpenSSL (TLS 1.3). Substitui os critérios padrão de seleção de ciphersuite para criptografia baseada em PSK.
TLSConnect	Como o agent deve se conectar ao Zabbix server ou proxy.
TLSCRLFile	O caminho completo de um arquivo contendo certificados revogados. Este parâmetro é usado para comunicações criptografadas entre componentes do Zabbix.
TLSKeyFile	O caminho completo de um arquivo contendo a chave privada do agent, usado para comunicações criptografadas entre componentes do Zabbix.
TLSPSKFile	O caminho completo de um arquivo contendo a chave pré-compartilhada do agent, usado para comunicações criptografadas com o Zabbix server.
TLSPSKIdentity	A string de identidade da chave pré-compartilhada, usada para comunicações criptografadas com o Zabbix server.
TLSServerCertIssuer	O emissor do certificado do server (proxy) permitido.
TLSServerCertSubject	O assunto do certificado do server (proxy) permitido.
UnsafeUserParameters	Permite que todos os caracteres sejam passados como argumentos para parâmetros definidos pelo usuário.
UserParameter	Um parâmetro definido pelo usuário para monitorar.
UserParameterDir	O caminho de busca padrão para comandos UserParameter.

Todos os parâmetros são não obrigatórios, a menos que explicitamente declarado que o parâmetro é obrigatório.

Observe que:

- Os valores padrão refletem os padrões do processo, não os valores nos arquivos de configuração fornecidos;
- O Zabbix suporta arquivos de configuração apenas na codificação UTF-8 sem BOM;
- Comentários iniciados com "#" são suportados apenas no início da linha.

Detalhes do parâmetro

Alias

Define um alias para uma chave de item. Pode ser usado para substituir uma chave de item longa e complexa por uma mais curta e simples.
 Vários parâmetros *Alias* podem estar presentes. Não são permitidos vários parâmetros com a mesma chave *Alias*.
 Chaves *Alias* diferentes podem referenciar a mesma chave de item.
 Aliases podem ser usados em *HostMetadataItem*, mas não no parâmetro *HostnameItem*.

Exemplo 1: Recuperando o uso do arquivo de paginação em porcentagem do server.

```
Alias=pg_usage:perf_counter[\Paging File(_Total)\% Usage]
```

Agora a chave abreviada **pg_usage** pode ser usada para recuperar dados.

Exemplo 2: Obtendo a carga da CPU com parâmetros padrão e personalizados.

```
Alias=cpu.load:system.cpu.load
Alias=cpu.load[*]:system.cpu.load[*]
```

Isso permite usar a chave **cpu.load** para obter a carga da CPU com parâmetros padrão, bem como usar **cpu.load[percpu,avg15]** para obter dados específicos sobre a carga da CPU.

Exemplo 3: Executando várias regras de descoberta de baixo nível processando os mesmos itens de descoberta.

```
Alias=vfs.fs.discovery[*]:vfs.fs.discovery
```

Agora é possível configurar várias regras de descoberta usando **vfs.fs.discovery** com parâmetros diferentes para cada regra, por exemplo, **vfs.fs.discovery[foo]**, **vfs.fs.discovery[bar]**, etc.

AllowKey

Permite a execução de chaves de item que correspondam a um padrão. O padrão da chave é uma expressão curinga que suporta o caractere "*" para corresponder a qualquer número de quaisquer caracteres.
Várias regras de correspondência de chaves podem ser definidas em combinação com DenyKey. Os parâmetros são processados um a um de acordo com a ordem de aparição. Veja também: Restringindo verificações do agent.

BufferSend

O intervalo de tempo em segundos que determina a frequência com que os valores são enviados do buffer para o servidor Zabbix.
Observe que, se o buffer estiver cheio, os dados serão enviados mais cedo.

Padrão: 5
 Intervalo: 1-3600

BufferSize

O número máximo de valores no buffer de memória. O agent enviará todos os dados coletados para o Zabbix server ou proxy se o buffer estiver cheio.
Este parâmetro só deve ser usado se o buffer persistente estiver desabilitado (*EnablePersistentBuffer=0*).

Padrão: 1000
 Intervalo: 2-65535

ControlSocket

O soquete de controle, usado para enviar comandos de tempo de execução com a opção '-R'.

Padrão: `\\.\pipe\agent.sock`

DebugLevel

Especifique o nível de depuração:
0 - informações básicas sobre a inicialização e a interrupção dos processos do Zabbix
1 - informações críticas;
2 - informações sobre erros;
3 - avisos;
4 - para depuração (produz muitas informações);
5 - depuração estendida (produz ainda mais informações).

Padrão: 3
 Intervalo: 0-5

DenyKey

Nega a execução das chaves de item que correspondam a um padrão. O padrão da chave é uma expressão curinga que suporta o caractere "*" para corresponder a qualquer número de quaisquer caracteres.
Várias regras de correspondência de chaves podem ser definidas em combinação com AllowKey. Os parâmetros são processados um a um de acordo com a ordem de aparição. Veja também: Restringindo verificações do agent.

EnablePersistentBuffer

Habilita o uso de armazenamento persistente local para items ativos. Se o armazenamento persistente estiver desabilitado, o buffer de memória será usado.

Padrão: 0
 Valores: 0 - desabilitado, 1 - habilitado

ForceActiveChecksOnStart

Executa verificações ativas imediatamente após a reinicialização para a primeira configuração recebida. Também disponível como um parâmetro de configuração por plugin, por exemplo: `Plugins.Uptime.System.ForceActiveChecksOnStart=1`

Padrão: 0
 Valores: 0 - desabilitado, 1 - habilitado

HeartbeatFrequency

A frequência das mensagens de heartbeat em segundos. Usado para monitorar a disponibilidade de verificações ativas.
0 - mensagens de heartbeat desabilitadas.

Padrão: 60
 Intervalo: 0-3600

HostInterface

Um parâmetro opcional que define a interface do host (endereço IP ou nome DNS) usada durante o processo de autoregistro do host. Esse valor será usado para preencher a interface no host recém-criado e permite configurar explicitamente o endereço IP ou DNS. Para mais detalhes, consulte Usando DNS como interface padrão.

Se não for definido, o valor será obtido de HostInterfaceItem.

O agent emitirá um erro e não iniciará se o valor ultrapassar o limite de 255 caracteres.

Intervalo: 0-255 caracteres

HostInterfaceItem

Um parâmetro opcional que define um item usado para determinar a interface do host (endereço IP ou nome DNS) durante o processo de autoregistro do host. Este valor é usado apenas se HostInterface não estiver definido. Para mais detalhes, consulte Usando DNS como interface padrão.

Durante uma solicitação de autoregistro, o agent registrará uma mensagem de aviso se o valor retornado pelo item especificado exceder o limite de 255 caracteres.

O item system.run[] é suportado independentemente das configurações AllowKey/DenyKey.

HostMetadata

Um parâmetro opcional que define os metadados usados para identificar ou distinguir o host durante o processo de autoregistro do host (agent ativo). HostMetadata permite distinguir entre hosts além do nome do host.

Se não for definido, o valor será adquirido de HostMetadataItem.

O agent emitirá um erro e não iniciará se o valor especificado ultrapassar o limite de 2034 bytes ou for uma string não UTF-8. Quando um parâmetro espera um endereço IP ou nome DNS, valores que são UTF-8 válidos, mas não são IPs ou nomes DNS válidos, também serão rejeitados e relatados como inválidos.

Metadados de várias linhas não são suportados — a saída será truncada na primeira quebra de linha.

Intervalo: 0-2034 bytes

HostMetadataItem

Um parâmetro opcional que define um item do agent Zabbix usado para obter os metadados do host. Esta opção só é usada quando HostMetadata não está definido.

O valor de HostMetadataItem é obtido em cada tentativa de autoregistro e é usado apenas no processo de autoregistro do host (agent ativo). HostMetadataItem permite distinguir entre hosts além do nome do host.

Parâmetros de usuário e aliases são suportados. O item system.run[] é suportado independentemente das configurações AllowKey/DenyKey.

Durante uma solicitação de autoregistro, o agent registrará uma mensagem de aviso se o valor retornado pelo item especificado ultrapassar o limite de 65535 pontos de código UTF-8. O valor retornado pelo item deve ser uma string UTF-8, caso contrário, será ignorado. Quando um parâmetro espera um endereço IP ou nome DNS, valores que são UTF-8 válidos, mas não são IPs ou nomes DNS válidos, também serão rejeitados e relatados como inválidos.

Metadados de múltiplas linhas não são suportados — a saída será truncada na primeira quebra de linha.

Hostname

Uma lista de nomes de host exclusivos, sensíveis a maiúsculas e minúsculas, separados por vírgulas. Necessário para verificações ativas e deve corresponder aos nomes de host configurados no server. O valor é obtido de HostnameItem se não estiver definido.
Caracteres permitidos: alfanuméricos, '.', ' ', '_' e '-'. Comprimento máximo: 128 caracteres por nome de host, 2048 caracteres para toda a linha.

Padrão: Definido por HostnameItem

HostnameItem

Um parâmetro opcional que define um item usado para obter o nome do host. Esta opção só é usada quando Hostname não está definido. Parâmetros de usuário ou aliases não são suportados, mas o item system.run[] é suportado independentemente dos valores AllowKey/DenyKey.

Padrão: `system.hostname`

Include

Você pode incluir arquivos individuais ou todos os arquivos em um diretório no arquivo de configuração (localizado em `C:\Program Files\Zabbix Agent 2` por padrão se o agent do Zabbix for instalado usando os pacotes do instalador MSI do Windows; localizado na pasta especificada durante a instalação se o agent do Zabbix for instalado como um arquivo zip). Todos os arquivos incluídos devem ter sintaxe correta, caso contrário o agent não iniciará. O caminho pode ser relativo à localização do arquivo *zabbix_agent2.conf* (por exemplo, `Include=.\zabbix_agent2.d\plugins.d\*.conf`).
Para incluir apenas arquivos relevantes no diretório especificado, o caractere curinga asterisco é suportado para correspondência de padrões.
Veja notas especiais sobre limitações.

Exemplo:

```
Include=C:\Program Files\Zabbix Agent2\zabbix_agent2.d\*.conf
```

ListenIP

Uma lista de endereços IP separados por vírgula nos quais o agent deve escutar. O primeiro endereço IP é enviado ao Zabbix server, se conectado a ele, para recuperar a lista de verificações ativas.

Padrão: `0.0.0.0`

ListenPort

O agente escutará nessa porta as conexões do servidor.

Padrão: 10050
 Intervalo: 1024-32767

LogFile

O nome do arquivo de log do agent.

Padrão: `c:\zabbix_agent2.log`
 Obrigatório: Sim, se LogType estiver definido como *file*; caso contrário, não

LogFileSize

O tamanho máximo de um arquivo de log em MB.
0 - desativa a rotação automática de logs.
Nota: Se o limite de tamanho do arquivo de log for atingido e a rotação do arquivo falhar, por qualquer motivo, o arquivo de log existente será truncado e iniciado novamente.

Padrão: 1
 Intervalo: 0-1024

LogType

O tipo de saída de registro:
file - grava o registro no arquivo especificado pelo parâmetro LogFile;
console - grava o registro na saída padrão.

Padrão: `file`

PersistentBufferFile

O arquivo onde o agent Zabbix 2 deve manter o banco de dados SQLite. Deve ser um nome de arquivo completo. Este parâmetro só é usado se o buffer persistente estiver habilitado (*EnablePersistentBuffer=1*).

PersistentBufferPeriod

O período de tempo durante o qual os dados devem ser armazenados quando não houver conexão com o server ou proxy. Os dados mais antigos serão perdidos. Os dados de log serão preservados. Este parâmetro só é usado se o buffer persistente estiver ativado (*EnablePersistentBuffer=1*).

Padrão: `1h`
 Intervalo: 1m-365d

Plugins.<PluginName>.System.Capacity

O limite de verificações por plugin <PluginName> que podem ser executadas ao mesmo tempo.

Padrão: 1000 Intervalo: 1-1000

Plugins.Log.MaxLinesPerSecond

O número máximo de novas linhas que o agent enviará por segundo para o Zabbix server ou proxy ao processar verificações ativas 'log', 'logrt' e 'eventlog'. O valor fornecido será substituído pelo parâmetro 'maxlines', fornecido na chave do item 'log', 'logrt' ou 'eventlog'.
Nota: O Zabbix processará 10 vezes mais novas linhas do que o definido em *MaxLinesPerSecond* para buscar a string necessária em itens de log.

Padrão: 20
 Intervalo: 1-1000

Plugins.SystemRun.LogRemoteCommands

Habilita o registro dos comandos shell executados como avisos. Os comandos serão registrados apenas se executados remotamente. Entradas de log não serão criadas se system.run[] for executado localmente pelos parâmetros HostMetadataItem, HostInterfaceItem ou HostnameItem.

Padrão: 0
 Valores: 0 - desabilitado, 1 - habilitado

PluginSocket

O caminho para o soquete UNIX para comunicações de plug-in carregáveis.

Padrão: `\\.\pipe\agent.plugin.sock`

PluginTimeout

O tempo limite para conexões com plug-ins carregáveis, em segundos.

Padrão: `Timeout`
 Intervalo: 1-30

PerfCounter

Define um novo parâmetro <parameter_name> que é o valor médio para o contador de desempenho do sistema <perf_counter_path> para o período de tempo especificado <period> (em segundos).
Sintaxe: <parameter_name>,"<perf_counter_path>",<period>

Por exemplo, se você deseja receber o número médio de interrupções do processador por segundo no último minuto, pode definir um novo parâmetro "interrupts" da seguinte forma:

```
PerfCounter = interrupts,"\Processor(0)\Interrupts/sec",60
```

Observe as aspas duplas ao redor do caminho do contador de desempenho. O nome do parâmetro (interrupts) deve ser usado como a chave do item ao criar um item. As amostras para calcular o valor médio serão coletadas a cada segundo.
Você pode executar "typeperf -qx" para obter a lista de todos os contadores de desempenho disponíveis no Windows.

PerfCounterEn

Define um novo parâmetro <parameter_name> que é o valor médio para o contador de desempenho do sistema <perf_counter_path> para o período de tempo especificado <period> (em segundos). Comparado ao PerfCounter, os caminhos do perfcounter devem estar em inglês. Suportado apenas no **Windows Server 2008/Vista** e posteriores.
Sintaxe: <parameter_name>,"<perf_counter_path>",<

Por exemplo, se você deseja receber o número médio de interrupções do processador por segundo no último minuto, pode definir um novo parâmetro "interrupts" da seguinte forma:

```
PerfCounterEn = interrupts,"\Processor(0)\Interrupts/sec",60
```

Observe as aspas duplas ao redor do caminho do contador de desempenho. O nome do parâmetro (interrupts) deve ser usado como a chave do item ao criar um item. As amostras para calcular o valor médio serão coletadas a cada segundo.
Você pode encontrar a lista de strings em inglês visualizando a seguinte chave de registro: `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Perflib\009`.

RefreshActiveChecks

Com que frequência a lista de verificações ativas é atualizada, em segundos. Observe que, após falhar ao atualizar as verificações ativas, a próxima atualização será tentada em 60 segundos.

Padrão: 5
 Intervalo: 1-86400

Server

Uma lista de endereços IP separados por vírgula, opcionalmente na notação CIDR, ou nomes DNS de servidores Zabbix ou proxies Zabbix. Conexões recebidas serão aceitas apenas dos hosts listados aqui. Se o suporte a IPv6 estiver habilitado, então '127.0.0.1', '::127.0.0.1', '::ffff:127.0.0.1' são tratados igualmente e '::/0' permitirá qualquer endereço IPv4 ou IPv6. '0.0.0.0/0' pode ser usado

para permitir qualquer endereço IPv4. Espaços são permitidos. Se este parâmetro não for especificado, desabilita as verificações passivas e o agent não irá escutar em nenhuma porta TCP.

Exemplo:

```
Server=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com
```

ServerActive

O endereço do Zabbix server/proxy ou a configuração do cluster para obter verificações ativas. O endereço do server/proxy é um endereço IP ou nome DNS e uma porta opcional separada por dois pontos.
A configuração do cluster é um ou mais endereços de membros do grupo de server ou proxy separados por ponto e vírgula. Vários Zabbix servers/clusters e Zabbix proxies podem ser especificados, separados por vírgula. A menos que esteja usando grupos de proxy, não deve ser especificado mais de um Zabbix proxy de cada Zabbix server/cluster. Se um Zabbix proxy for especificado, o Zabbix server/cluster para esse proxy não deve ser especificado.
Vários endereços separados por vírgula podem ser fornecidos para usar vários Zabbix servers independentes em paralelo. Espaços são permitidos.
Se a porta não for especificada, a porta padrão será usada.
Endereços IPv6 devem ser colocados entre colchetes se a porta para esse host for especificada. Se a porta não for especificada, os colchetes para endereços IPv6 são opcionais.
Se este parâmetro não for especificado, as verificações ativas serão desativadas.

Exemplo para Zabbix proxy:

```
ServerActive=127.0.0.1:10051
```

Exemplo para grupo de Zabbix proxy:

```
ServerActive=proxy1.example.com;proxy2.example.com;proxy3.example.com;proxy4.example.com;proxy5.example.co
```

Exemplo para múltiplos servers:

```
ServerActive=127.0.0.1:20051,zabbix.domain,\[::1\]:30051,::1,\[12fc::1\]
```

Exemplo para alta disponibilidade:

```
ServerActive=zabbix.cluster.node1;zabbix.cluster.node2:20051;zabbix.cluster.node3
```

Exemplo para alta disponibilidade com dois clusters e um server:

```
ServerActive=zabbix.cluster.node1;zabbix.cluster.node2:20051,zabbix.cluster2.node1;zabbix.cluster2.node2,z
```

SourceIP

O endereço IP de origem para:

- conexões de saída para o Zabbix server ou Zabbix proxy;
- fazer conexões ao executar alguns items (web.page.get, net.tcp.port, etc.).

StatusPort

Se definido, o agente escutará nessa porta as solicitações de status HTTP (http://localhost:<port>/status).

Intervalo: 1024-32767

Timeout

Especifica por quanto tempo esperar (em segundos) para estabelecer conexão e trocar dados com o Zabbix proxy ou server.

Este parâmetro define a duração de várias operações de comunicação, incluindo:

- aguardar uma resposta do Zabbix server;
- enviar solicitações ao Zabbix server, incluindo solicitações de configuração de item e dados de item em verificações ativas;
- recuperar dados de log por meio do monitoramento de arquivos de log ou do log de eventos do Windows;
- enviar mensagens de heartbeat;
- duração máxima para verificações `vfs.*`;
- servir como fallback em cenários onde um server ou proxy anterior à versão 7.0 envia verificações sem timeouts.

Este timeout **não** será usado para aquelas verificações de agent que possuem configurações de timeout configuráveis no frontend (em nível global, proxy ou por item).

Padrão: 3
 Intervalo: 1-30

TLSAccept

As conexões de entrada a serem aceitas. Usado para verificações passivas. Vários valores podem ser especificados, separados por vírgula:
unencrypted - aceitar conexões sem criptografia (padrão)
psk - aceitar conexões com TLS e uma chave pré-compartilhada (PSK)
cert - aceitar conexões com TLS e um certificado

Obrigatório: sim, se os parâmetros de certificado TLS ou PSK forem definidos (mesmo para conexão *unencrypted*); caso contrário, não

TLSCAFile

O nome completo do caminho do arquivo que contém os certificados da(s) CA(s) de nível superior para verificação de certificados de pares, usados para comunicações criptografadas entre os componentes do Zabbix.

TLSCertFile

O nome completo do caminho do arquivo que contém o certificado do agente ou a cadeia de certificados, usado para comunicações criptografadas com os componentes do Zabbix.

TLSCipherAll13

A string de cifra do OpenSSL (TLS 1.3). Substitui os critérios padrão de seleção de ciphersuite para criptografia baseada em certificado e PSK.

Exemplo:

```
TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

Este parâmetro é suportado desde o Zabbix 7.0.4.

TLSCipherCert13

A string de cifra do OpenSSL (TLS 1.3). Substitui os critérios padrão de seleção de ciphersuite para criptografia baseada em certificado.

Observe que este parâmetro não pode ser usado junto com `TLSAccept=cert,psk`; para conexão por certificado (`TLSConnect=cert`) use TLSCipherAll13 em vez disso.

Este parâmetro é suportado a partir do Zabbix 7.0.4.

TLSCipherPSK13

A string de cifra do OpenSSL (TLS 1.3). Substitui os critérios padrão de seleção de ciphersuite para criptografia baseada em PSK.

Exemplo:

```
TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

Este parâmetro é suportado a partir do Zabbix 7.0.4.

TLSConnect

Como o agent deve se conectar ao Zabbix server ou proxy. Usado para verificações ativas. Apenas um valor pode ser especificado:
unencrypted - conectar sem criptografia (padrão)
psk - conectar usando TLS e uma chave pré-compartilhada (PSK)
cert - conectar usando TLS e um certificado

Obrigatório: sim, se os parâmetros de certificado TLS ou PSK forem definidos (mesmo para conexão *unencrypted*); caso contrário, não

TLSCRLFile

O caminho completo do arquivo contendo certificados revogados. Este parâmetro é usado para comunicações criptografadas entre os componentes do Zabbix.

TLSKeyFile

O nome completo do caminho do arquivo que contém a chave privada do agente, usada para comunicações criptografadas entre os componentes do Zabbix.

TLSPSKFile

O nome completo do caminho do arquivo que contém a chave pré-compartilhada do agente, usada para comunicações criptografadas com o servidor Zabbix.

TLSPSKIdentity

A string de identidade da chave pré-compartilhada, usada para comunicações criptografadas com o servidor Zabbix.

TLSServerCertIssuer

O emissor do certificado do servidor (proxy) permitido.

TLSServerCertSubject

O assunto do certificado do servidor (proxy) permitido.

UnsafeUserParameters

Permitir que todos os caracteres sejam passados como argumentos para parâmetros definidos pelo usuário. Os seguintes caracteres não são permitidos: \ ' " ` * ? [ ] { } ~ $! & ; ( ) < > | # @
Além disso, caracteres de nova linha não são permitidos.

Padrão: 0
 Valores: 0 - não permitir, 1 - permitir

UserParameter

Um parâmetro definido pelo usuário para monitoramento. Pode haver vários parâmetros definidos pelo usuário.
Formato: UserParameter=<key>,<comando shell>
Observe que o comando shell não deve retornar uma string vazia ou apenas EOL. Os comandos shell podem ter caminhos relativos, se o parâmetro UserParameterDir for especificado.

Exemplo:

```
UserParameter=system.test,dir /b | find /c /v ""
UserParameter=check_cpu,.\custom\_script.bat
```

UserParameterDir

O caminho de pesquisa padrão para comandos UserParameter. Se usado, o agent mudará seu diretório de trabalho para o especificado aqui antes de executar um comando. Assim, os comandos UserParameter podem ter um prefixo relativo `.\` em vez de um caminho completo.
Apenas uma entrada é permitida.

Exemplo:

```
UserParameterDir=C:\opt\myscripts
```

7 Plugins do Zabbix Agent 2

Visão geral

Esta seção contém descrições de parâmetros de arquivo de configuração para plugins do Zabbix Agent 2. Por favor utilize a barra lateral para acessar as informações sobre um plugin específico.

1 Plugin Ceph

Visão geral

Esta seção lista os parâmetros suportados no arquivo de configuração do plugin Ceph do Zabbix agent 2 (ceph.conf). Note que:

- Os valores padrão refletem os padrões do processo, não os valores nos arquivos de configuração transportados;
- O Zabbix suporta apenas arquivos de configuração codificados em UTF-8 sem BOM;
- Comentários começando com "#" são suportados apenas no início da linha.

Parâmetros

Parâmetro	Obrigatório	Intervalo	Padrão	Descrição
Plugins.Ceph.Default.ApiKey	não			Chave de API padrão para conexão com o Ceph; usada se nenhum valor for especificado em uma chave de item ou sessão nomeada.
Plugins.Ceph.Default.User	não			Nome de usuário padrão para conexão com o Ceph; usado se nenhum valor for especificado em uma chave de item ou sessão nomeada.
Plugins.Ceph.Default.Uri	não		https://localhost:8003	URI padrão para conexão com o Ceph; usada se nenhum valor for especificado em uma chave de item ou sessão nomeada. Não deve incluir credenciais embutidas (serão ignoradas). Deve corresponder ao formato de URI. Apenas o esquema `https` é suportado; um esquema pode ser omitido. Uma porta pode ser omitida (padrão=8003). Exemplos: `https://127.0.0.1:8003` `localhost`

Parâmetro	Obrigatório	Intervalo	Padrão	Descrição
Plugins.Ceph.InsecureSkipVerify	não	false / true	false	Determina se um cliente http deve verificar a cadeia de certificados do servidor e o nome do host. Se *true*, o TLS aceita qualquer certificado apresentado pelo servidor e qualquer nome de host nesse certificado. Neste modo, o TLS é suscetível a ataques man-in-the-middle (deve ser usado apenas para testes).
Plugins.Ceph.KeepAlive	não	60-900	300	O tempo máximo de espera (em segundos) antes que conexões de plugin não utilizadas sejam fechadas.
Plugins.Ceph.Sessions.<SessionName>.ApiKey	não			Chave de API da sessão nomeada. **<SessionName>** - define o nome de uma sessão para uso em chaves de item.
Plugins.Ceph.Sessions.<SessionName>.User	não			Nome de usuário da sessão nomeada. **<SessionName>** - define o nome de uma sessão para uso em chaves de item.
Plugins.Ceph.Sessions.<SessionName>.Uri	não			String de conexão de uma sessão nomeada. **<SessionName>** - define o nome de uma sessão para uso em chaves de item. Não deve incluir credenciais embutidas (serão ignoradas). Deve corresponder ao formato de URI. Apenas o esquema `https` é suportado; um esquema pode ser omitido. Uma porta pode ser omitida (padrão=8003). Exemplos: `https://127.0.0.1:8003` `localhost`
Plugins.Ceph.Timeout	não	1-30	timeout global	Timeout de execução da requisição (a duração, em segundos, para aguardar a conclusão de uma requisição antes de encerrá-la).

Veja também:

- Descrição dos parâmetros gerais de configuração do Zabbix agent 2: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Instruções para configurar plugins

2 Plugin Docker

Visão geral

Esta seção lista os parâmetros suportados no arquivo de configuração do Docker para o Zabbix Agent 2 (docker.conf). Note que:

- Os valores padrão refletem os padrões do processo, não os valores nos arquivos de configuração transportados;
- O Zabbix suporta apenas arquivos de configuração codificados em UTF-8 sem BOM;
- Comentários começando com "#" são suportados apenas no início da linha.

Parâmetros

Parâmetro	Obrigatório	Intervalo	Padrão	Descrição
Plugins.Docker.Endpoint	não		unix:///var/run/docker.sock	Localização do unix-socket do daemon Docker. Deve conter um esquema (apenas `unix://` é suportado).
Plugins.Docker.Timeout	não	1-30	timeout global	Timeout de execução da requisição (a duração, em segundos, para aguardar a conclusão de uma requisição antes de encerrá-la).

Veja também:

- Descrição dos parâmetros gerais de configuração do Zabbix agent 2: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Instruções para configurar plugins

3 Plugin Ember+

Visão geral

Esta seção lista os parâmetros suportados no arquivo de configuração do plugin Ember+ Zabbix agent 2 (ember.conf).

O plugin Ember+ é um plugin carregável e está disponível e totalmente descrito no repositório do plugin Ember+.

Este plugin atualmente está disponível apenas para ser compilado a partir do código-fonte (tanto para Unix quanto para Windows).

Observe que:

- Os valores padrão refletem os padrões do processo, não os valores nos arquivos de configuração fornecidos.
- O Zabbix suporta arquivos de configuração apenas na codificação UTF-8 sem BOM.
- Comentários iniciados com "#" são suportados apenas no início da linha.

Opções

Parâmetro	Descrição
-V --version	Imprime a versão do plugin e informações de licença.
-h --help	Imprime informações de ajuda (abreviação).

Parâmetros

Parâmetro	Obrigatório	Intervalo	Padrão	Descrição
Plugins.EmberPlus.Default.Uri	não		tcp://localhost:9998	URI padrão para conectar. O único esquema suportado é `tcp://`. Um esquema pode ser omitido. Credenciais embutidas serão ignoradas.
Plugins.EmberPlus.KeepAlive	não	60-900	300	O tempo máximo de espera (em segundos) antes que conexões de plugin não utilizadas sejam fechadas.
Plugins.EmberPlus.Sessions.<SessionName>.Uri	não		tcp://localhost:9998	URI para conectar, para a sessão nomeada. O único esquema suportado é `tcp://`. Um esquema pode ser omitido. Credenciais embutidas serão ignoradas. **<SessionName>** - define o nome de uma sessão para uso nas chaves de item.
Plugins.EmberPlus.System.Path	não			Caminho para o executável do plugin Ember+. Exemplo de uso: `Plugins.EmberPlus.System.Path=/usr/sbin/zabbix-agent2-plu`
Plugins.EmberPlus.Timeout	não	1-30	timeout global	A duração, em segundos, para aguardar a resposta de um servidor ao conectar pela primeira vez e em operações subsequentes na sessão.

Veja também:

- Descrição dos parâmetros gerais de configuração do Zabbix agent 2: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Instruções para configurar plugins

4 Plugin Memcached

Overview

Esta seção lista os parâmetros suportados no arquivo de configuração do Memcached para o Zabbix Agent 2 (memcached.conf).
Note que:

- Os valores padrão refletem os padrões do processo, não os valores nos arquivos de configuração transportados;
- O Zabbix suporta apenas arquivos de configuração codificados em UTF-8 sem BOM;
- Comentários começando com "#" são suportados apenas no início da linha.

Parâmetros

Parâmetro	Obrigatório	Intervalo	Padrão	Descrição
Plugins.Memcached.Default.Password	não			Senha padrão para conectar ao Memcached; usada se nenhum valor for especificado na chave do item ou na sessão nomeada.

Parâmetro	Obrigatório	Intervalo	Padrão	Descrição

Plugins.Memcached.Default.Uri	não		tcp://localhost:11211	URI padrão para conectar ao Memcached; usada se nenhum valor for especificado na chave do item ou na sessão nomeada.
				
				Não deve incluir credenciais embutidas (serão ignoradas).
				Deve corresponder ao formato de URI.
				Esquemas suportados: `tcp`, `unix`; um esquema pode ser omitido.
				Uma porta pode ser omitida (padrão=11211).
				Exemplos: `tcp://localhost:11211`
				`localhost`
				`unix:/var/run/memcached.sock`

Plugins.Memcached.Default.User	não			Nome de usuário padrão para conectar ao Memcached; usado se nenhum valor for especificado na chave do item ou na sessão nomeada.
Plugins.Memcached.KeepAlive	não	60-900	300	Tempo máximo de espera (em segundos) antes que conexões de plugin não utilizadas sejam fechadas.
Plugins.Memcached.Sessions.<SessionName>.Password	não			Senha da sessão nomeada. **<SessionName>** - defina o nome da sessão para usar nas chaves dos itens.

Plugins.Memcached.Sessions.<SessionName>.Uri	não			String de conexão da sessão nomeada.
				<SessionName> - defina o nome da sessão para usar nas chaves dos itens.
				
				Não deve incluir credenciais embutidas (serão ignoradas).
				Deve corresponder ao formato de URI.
				Esquemas suportados: `tcp`, `unix`; um esquema pode ser omitido.
				Uma porta pode ser omitida (padrão=11211).
				Exemplos: `tcp://localhost:11211`
				`localhost`
				`unix:/var/run/memcached.sock`

Plugins.Memcached.Sessions.<SessionName>.User	não			Nome de usuário da sessão nomeada. **<SessionName>** - defina o nome da sessão para usar nas chaves dos itens.
Plugins.Memcached.Timeout	não	1-30	timeout global	Tempo limite de execução da requisição (a duração, em segundos, para aguardar a conclusão de uma requisição antes de encerrá-la).

Veja também:

- Descrição dos parâmetros gerais de configuração do Zabbix agent 2: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Instruções para configurar plugins

5 Plugin Modbus

Visão geral

Esta seção lista os parâmetros suportados no arquivo de configuração do Modbus para o Zabbix Agent 2 (modbus.conf). Note que:

- Os valores padrão refletem os padrões do processo, não os valores nos arquivos de configuração transportados;
- O Zabbix suporta apenas arquivos de configuração codificados em UTF-8 sem BOM;
- Comentários começando com "#" são suportados apenas no início da linha.

Parâmetros

Parâmetro	Obrigatório	Intervalo	Padrão	Descrição
Plugins.Modbus.Sessions.<SessionName>.Endpoint	Não			Endpoint é uma string de conexão composta por um esquema de protocolo, um endereço de host e uma porta ou nome de porta serial e atributos. **<SessionName>** - define o nome de uma sessão para uso nas chaves de item.
Plugins.Modbus.Sessions.<SessionName>.SlaveID	Não			Slave ID de uma sessão nomeada. **<SessionName>** - define o nome de uma sessão para uso nas chaves de item. Exemplo: `Plugins.Modbus.Sessions.MB1.SlaveID=20` *Nota* que este parâmetro de sessão nomeada é verificado apenas se o valor fornecido no parâmetro slave ID da chave de item estiver vazio.
Plugins.Modbus.Sessions.<SessionName>.Timeout	Não			Timeout de uma sessão nomeada em segundos. **<SessionName>** - define o nome de uma sessão para uso nas chaves de item. Exemplo: `Plugins.Modbus.Sessions.MB1.Timeout=2`

Se você precisar definir o tempo limite de execução da requisição (quanto tempo esperar para que uma requisição seja concluída antes de encerrá-la), use o formulário de configuração do item.

Veja também:

- Descrição dos parâmetros gerais de configuração do Zabbix agent 2: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Instruções para configurar plugins

6 Plugin do MongoDB

Visão geral

Esta seção lista os parâmetros suportados no arquivo de configuração do plugin MongoDB Zabbix agent 2 (mongo.conf).

O plugin MongoDB é um plugin carregável e está disponível e totalmente descrito no repositório do plugin MongoDB.

Observe que:

- Os valores padrão refletem os padrões do processo, não os valores nos arquivos de configuração fornecidos.
- O Zabbix suporta arquivos de configuração apenas na codificação UTF-8 sem BOM.
- Comentários iniciados com ”#” são suportados apenas no início da linha.

Options

Parameter	Description
-V --version	Print the plugin version and license information.
-h --help	Print help information (shorthand).

Parâmetros

Parâmetro	Obrigatório	Intervalo	Padrão	Descrição
Plugins.MongoDB.Default.Password	Não			Senha padrão para conexão com o MongoDB; usada se nenhum valor for especificado na chave do item ou na sessão nomeada.

Parâmetro	Obrigatório	Intervalo	Padrão	Descrição
Plugins.MongoDB.Default.Uri	Não			URI padrão para conexão com o MongoDB; usada se nenhum valor for especificado na chave do item ou na sessão nomeada. Não deve incluir credenciais embutidas (serão ignoradas). Deve corresponder ao formato URI. Apenas o esquema `tcp` é suportado; o esquema pode ser omitido. Uma porta pode ser omitida (padrão=27017). Exemplos: `tcp://127.0.0.1:27017`, `tcp:localhost`, `localhost`
Plugins.MongoDB.Default.User	Não			Nome de usuário padrão para conexão com o MongoDB; usado se nenhum valor for especificado na chave do item ou na sessão nomeada.
Plugins.MongoDB.KeepAlive	Não	60-900	300	Tempo máximo de espera (em segundos) antes que conexões de plugin não utilizadas sejam fechadas.
Plugins.MongoDB.Sessions.<SessionName>.Password	Não			Senha da sessão nomeada. **<SessionName>** - defina o nome da sessão para uso nas chaves dos itens.
Plugins.MongoDB.Sessions.<SessionName>.TLSCAFile	Sim, se Plugins.MongoDB.Sessions.<SessionName>.TLSConnect estiver definido como um dos valores: verify_ca, verify_full)			Caminho completo para um arquivo contendo os certificados da(s) CA(s) de nível superior para verificação do certificado do peer, usado para comunicações criptografadas entre o Zabbix agent 2 e os bancos de dados monitorados. **<SessionName>** - defina o nome da sessão para uso nas chaves dos itens.
Plugins.MongoDB.Sessions.<SessionName>.TLSCertFile	Sim, se Plugins.MongoDB.Sessions.<SessionName>.TLSKeyFile for especificado			Caminho completo para um arquivo contendo o certificado do agent ou cadeia de certificados, usado para comunicações criptografadas entre o Zabbix agent 2 e os bancos de dados monitorados. **<SessionName>** - defina o nome da sessão para uso nas chaves dos itens.
Plugins.MongoDB.Sessions.<SessionName>.TLSConnect	Não			Tipo de criptografia para comunicações entre o Zabbix agent 2 e os bancos de dados monitorados. **<SessionName>** - defina o nome da sessão para uso nas chaves dos itens. Valores suportados: *required* - requer conexão TLS; *verify_ca* - verifica certificados; *verify_full* - verifica certificados e endereço IP. Suportado desde a versão 1.2.1 do plugin
Plugins.MongoDB.Sessions.<SessionName>.TLSKeyFile	Sim, se Plugins.MongoDB.Sessions.<SessionName>.TLSCertFile for especificado			Caminho completo para um arquivo contendo a chave privada do banco de dados usada para comunicações criptografadas entre o Zabbix agent 2 e os bancos de dados monitorados. **<SessionName>** - defina o nome da sessão para uso nas chaves dos itens.
Plugins.MongoDB.Sessions.<SessionName>.Uri	Não			String de conexão de uma sessão nomeada. **<SessionName>** - defina o nome da sessão para uso nas chaves dos itens. Não deve incluir credenciais embutidas (serão ignoradas). Deve corresponder ao formato URI. Apenas o esquema `tcp` é suportado; o esquema pode ser omitido. Uma porta pode ser omitida (padrão=27017). Exemplos: `tcp://127.0.0.1:27017`, `tcp:localhost`, `localhost`

Parâmetro	Obrigatório	Intervalo	Padrão	Descrição
Plugins.MongoDB.Sessions.<SessionName>.User	Não			Nome de usuário da sessão nomeada. **<SessionName>** - defina o nome da sessão para uso nas chaves dos itens.
Plugins.MongoDB.System.Path	Não			Caminho para o executável do plugin.
Plugins.MongoDB.Timeout	Não	1-30	timeout global	Timeout de execução da requisição (a duração, em segundos, para aguardar a conclusão de uma requisição antes de encerrá-la).

Veja também:

- Descrição dos parâmetros gerais de configuração do Zabbix agent 2: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Instruções para configurar plugins

7 Plugin MQTT

Visão geral

Esta seção lista os parâmetros suportados no arquivo de configuração do MQTT para o Zabbix Agent 2 (mqtt.conf). Note que:

- Os valores padrão refletem os padrões do processo, não os valores nos arquivos de configuração transportados;
- O Zabbix suporta apenas arquivos de configuração codificados em UTF-8 sem BOM;
- Comentários começando com "#" são suportados apenas no início da linha.

Parâmetros

Parâmetro	Obrigatório	Intervalo	Padrão	Descrição
Plugins.MQTT.Default.Password	Não			Senha padrão para conexão com o MQTT; usada se nenhum valor for especificado na chave do item ou na sessão nomeada.
Plugins.MQTT.Default.TLSCAFile	Não			Caminho completo para um arquivo contendo os certificados da(s) CA(s) de nível superior para verificação do certificado do par para comunicações criptografadas entre o Zabbix agent 2 e o broker MQTT; usado se nenhum valor for especificado em uma sessão nomeada.
Plugins.MQTT.Default.TLSCertFile	Não			Caminho completo para um arquivo contendo o certificado do agent ou cadeia de certificados para comunicações criptografadas entre o Zabbix agent 2 e o broker MQTT; usado se nenhum valor for especificado em uma sessão nomeada.
Plugins.MQTT.Default.TLSKeyFile	Não			Caminho completo para um arquivo contendo a chave privada MQTT para comunicações criptografadas entre o Zabbix agent 2 e o broker MQTT; usado se nenhum valor for especificado em uma sessão nomeada.

Plugins.MQTT.Default.Topic	Não			Tópico padrão para assinatura MQTT; usado se nenhum valor for especificado na chave do item ou na sessão nomeada.
				
				O tópico pode conter curingas ("+","#")
				Exemplos: `path/to/file`
				`path/to/#`
				`path/+/topic`

Plugins.MQTT.Default.Url	Não		tcp://localhost:1883	String de conexão padrão do broker MQTT; usada se nenhum valor for especificado na chave do item ou na sessão nomeada.
				
				Não deve incluir parâmetros de consulta.
				Deve corresponder ao formato da URL.
				Esquemas suportados: `tcp` (padrão), `ws`, `tls`; um esquema pode ser omitido.
				Uma porta pode ser omitida (padrão=1883).
				Exemplos: `tcp://host:1883`
				`localhost`
				`ws://host:8080`

Parâmetro	Obrigatório	Intervalo	Padrão	Descrição
Plugins.MQTT.Default.User	não			Nome de usuário padrão para conexão com o MQTT; usado se nenhum valor for especificado na chave do item ou na sessão nomeada.
Plugins.MQTT.Sessions.<SessionName>.Password	não			Senha da sessão nomeada. **<SessionName>** - defina o nome de uma sessão para usar nas chaves de item.
Plugins.MQTT.Sessions.<SessionName>.TLSCAFile	não			Caminho completo para um arquivo contendo os certificados da(s) CA(s) de nível superior para verificação do certificado do par, usado para comunicações criptografadas entre o Zabbix agent 2 e o broker MQTT. **<SessionName>** - defina o nome de uma sessão para usar nas chaves de item.
Plugins.MQTT.Sessions.<SessionName>.TLSCertFile	não			Caminho completo para um arquivo contendo o certificado do agent ou cadeia de certificados, usado para comunicações criptografadas entre o Zabbix agent 2 e o broker MQTT. **<SessionName>** - defina o nome de uma sessão para usar nas chaves de item.
Plugins.MQTT.Sessions.<SessionName>.TLSKeyFile	não			Caminho completo para um arquivo contendo a chave privada MQTT usada para comunicações criptografadas entre o Zabbix agent 2 e o broker MQTT. **<SessionName>** - defina o nome de uma sessão para usar nas chaves de item.

Plugins.MQTT.Sessions.<SessionName>.Topic	não			Tópico da sessão nomeada para assinatura MQTT.
				<SessionName> - defina o nome de uma sessão para usar nas chaves de item.
				
				O tópico pode conter curingas (”+”,”#”)
				Exemplos: `path/to/file`
				`path/to/#`
				`path/+/topic`

Plugins.MQTT.Sessions.<SessionName>.Url	não			String de conexão de uma sessão nomeada.
				<SessionName> - defina o nome de uma sessão para usar nas chaves de item.
				
				Não deve incluir parâmetros de consulta.
				Deve corresponder ao formato da URL.
				Esquemas suportados: `tcp` (padrão), `ws`, `tls`; um esquema pode ser omitido.
				Uma porta pode ser omitida (padrão=1883).
				Exemplos: `tcp://host:1883`
				`localhost`
				`ws://host:8080`

Plugins.MQTT.Sessions.<SessionName>.User	não			Nome de usuário da sessão nomeada. **<SessionName>** - defina o nome de uma sessão para usar nas chaves de item.

Se você precisar definir o tempo limite de execução da solicitação (quanto tempo esperar para que uma solicitação seja concluída antes de encerrá-la), use o formulário de configuração do item.

Veja também:

- Descrição dos parâmetros gerais de configuração do Zabbix agent 2: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Instruções para configurar plugins

8 Plugin MSSQL

Visão geral

Esta seção lista os parâmetros suportados no arquivo de configuração do plugin MSSQL Zabbix agent 2 (mssql.conf).

O plugin MSSQL é um plugin carregável e está disponível e totalmente descrito no repositório do plugin MSSQL.

Observe que:

- Os valores padrão refletem os padrões do processo, não os valores nos arquivos de configuração fornecidos.
- O Zabbix suporta arquivos de configuração apenas na codificação UTF-8 sem BOM.
- Comentários iniciados com "#" são suportados apenas no início da linha.

Opções

Parâmetro	Descrição
-V --version	Imprime a versão do plugin e informações de licença.
-h --help	Imprime informações de ajuda (forma abreviada).

Parâmetros

Parâmetro	Obrigatório	Intervalo	Padrão	Descrição
Plugins.MSSQL.CustomQueriesDir	Não		vazio	Especifica o caminho do arquivo para um diretório contendo arquivos .sql definidos pelo usuário com consultas personalizadas que o plugin pode executar. O plugin carrega todos os arquivos .sql disponíveis no diretório configurado na inicialização. Isso significa que quaisquer alterações nos arquivos de consulta personalizada não serão refletidas até que o plugin seja reiniciado. O plugin é iniciado e parado junto com o Zabbix agent 2.
Plugins.MSSQL.Default.CACertPath	Não			O caminho padrão do arquivo para o certificado da chave pública da autoridade certificadora (CA) que emitiu o certificado do servidor MSSQL. O certificado deve estar no formato PEM.
Plugins.MSSQL.Default.Database	Não			O nome padrão do banco de dados para conectar.
Plugins.MSSQL.Default.Encrypt	Não			Especifica o tipo de criptografia de conexão padrão. Os valores possíveis são: *true* - o envio de dados entre o plugin e o servidor é criptografado; *false* - o envio de dados entre o plugin e o servidor não é criptografado além do pacote de login; *strict* - o envio de dados entre o plugin e o servidor é criptografado E2E usando TDS8; *disable* - o envio de dados entre o plugin e o servidor não é criptografado.
Plugins.MSSQL.Default.HostNameInCertificate	Não			O nome comum (CN) do certificado do servidor MSSQL por padrão.
Plugins.MSSQL.Default.Password	Não			A senha a ser enviada para um servidor MSSQL protegido por padrão.
Plugins.MSSQL.Default.TLSMinVersion	Não			A versão mínima do TLS a ser usada por padrão. Os valores possíveis são: `1.0`, `1.1`, `1.2`, `1.3`.
Plugins.MSSQL.Default.TrustServerCertificate	Não			Se o plugin deve confiar no certificado do servidor sem validá-lo por padrão. Valores possíveis: `true`, `false`.
Plugins.MSSQL.Default.Uri	Não		sqlserver://localhost:1433	O URI padrão para conectar. O único esquema suportado é `sqlserver://`. Um esquema pode ser omitido. Credenciais embutidas serão ignoradas.
Plugins.MSSQL.Default.User	Não			O nome de usuário padrão a ser enviado para um servidor MSSQL protegido.
Plugins.MSSQL.KeepAlive	Não	60-900	300	O tempo máximo de espera (em segundos) antes que conexões de plugin não utilizadas sejam fechadas.
Plugins.MSSQL.Sessions.<SessionName>.CACertPath	Não			O caminho do arquivo para o certificado da chave pública da autoridade certificadora (CA) que emitiu o certificado do servidor MSSQL para a sessão nomeada. O certificado deve estar no formato PEM. **<SessionName>** - define o nome de uma sessão para uso nas chaves de item.
Plugins.MSSQL.Sessions.<SessionName>.Database	Não			O nome do banco de dados para conectar para a sessão nomeada. **<SessionName>** - define o nome de uma sessão para uso nas chaves de item.

Parâmetro	Obrigatório	Intervalo	Padrão	Descrição
Plugins.MSSQL.Sessions.<SessionName>.Encrypt	Não			Especifica o tipo de criptografia de conexão para a sessão nomeada. Os valores possíveis são: *true* - o envio de dados entre o plugin e o servidor é criptografado; *false* - o envio de dados entre o plugin e o servidor não é criptografado além do pacote de login; *strict* - o envio de dados entre o plugin e o servidor é criptografado E2E usando TDS8; *disable* - o envio de dados entre o plugin e o servidor não é criptografado. **<SessionName>** - define o nome de uma sessão para uso nas chaves de item.
Plugins.MSSQL.Sessions.<SessionName>.HostNameInCertificate	Não			O nome comum (CN) do certificado do servidor MSSQL para a sessão nomeada. **<SessionName>** - define o nome de uma sessão para uso nas chaves de item.
Plugins.MSSQL.Sessions.<SessionName>.Password	Não			A senha a ser enviada para um servidor MSSQL protegido para a sessão nomeada. **<SessionName>** - define o nome de uma sessão para uso nas chaves de item.
Plugins.MSSQL.Sessions.<SessionName>.TLSMinVersion	Não			A versão mínima do TLS a ser usada para a sessão nomeada. Os valores possíveis são: `1.0`, `1.1`, `1.2`, `1.3`. **<SessionName>** - define o nome de uma sessão para uso nas chaves de item.
Plugins.MSSQL.Sessions.<SessionName>.TrustServerCertificate	Não			Se o plugin deve confiar no certificado do servidor sem validá-lo para a sessão nomeada. Valores possíveis: `true`, `false`. **<SessionName>** - define o nome de uma sessão para uso nas chaves de item.
Plugins.MSSQL.Sessions.<SessionName>.Uri	Não		sqlserver://localhost:1433	O URI para a conectar, para a sessão nomeada. O único esquema suportado é `sqlserver://`. Um esquema pode ser omitido. Credenciais embutidas serão ignoradas. **<SessionName>** - define o nome de uma sessão para uso nas chaves de item.
Plugins.MSSQL.Sessions.<SessionName>.User	Não			O nome de usuário a ser enviado para um servidor MSSQL protegido para a sessão nomeada. **<SessionName>** - define o nome de uma sessão para uso nas chaves de item.
Plugins.MSSQL.System.Path	Sim			Caminho para o executável do plugin MSSQL. Configuração global para o plugin MSSQL. Aplicado a todas as conexões. Exemplo de uso: `Plugins.MSSQL.System.Path=/usr/sbin/zabbix-agent2-plugin/`
Plugins.MSSQL.Timeout	Não	1-30	timeout global	A duração, em segundos, para aguardar uma resposta do servidor ao conectar pela primeira vez e em operações subsequentes na sessão.

Veja também:

- Descrição dos parâmetros gerais de configuração do Zabbix agent 2: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Instruções para configurar plugins

9 Plugin MySQL

Visão geral

Esta seção lista os parâmetros suportados no arquivo de configuração do MySQL para o Zabbix Agent 2 (mysql.conf). Note que:

- Os valores padrão refletem os padrões do processo, não os valores nos arquivos de configuração transportados;
- O Zabbix suporta apenas arquivos de configuração codificados em UTF-8 sem BOM;
- Comentários começando com "#" são suportados apenas no início da linha.

Parâmetros

Parâmetro	Obrigatório	Intervalo	Padrão	Descrição
Plugins.Mysql.CallTimeout	não	1-30	timeout global	Tempo máximo em segundos para aguardar a conclusão de uma solicitação.
Plugins.Mysql.CustomQueriesPath	não		vazio	Caminho completo para o diretório usado para armazenar consultas personalizadas.
Plugins.Mysql.Default.Password	não			Senha padrão para conectar ao MySQL; usada se nenhum valor for especificado na chave do item ou na sessão nomeada.
Plugins.Mysql.Default.TLSCAFile	não (sim, se Plugins.Mysql.Default.TLSConnect estiver definido como *verify_ca* ou *verify_full*)			Caminho completo para um arquivo contendo os certificados da(s) CA(s) de nível superior para verificação do certificado do par para comunicações criptografadas entre o Zabbix agent 2 e os bancos de dados monitorados; usado se nenhum valor for especificado em uma sessão nomeada.
Plugins.Mysql.Default.TLSCertFile	não (sim, se Plugins.Mysql.Default.TLSConnect estiver definido como *verify_ca* ou *verify_full*)			Caminho completo para um arquivo contendo o certificado do agent ou cadeia de certificados para comunicações criptografadas entre o Zabbix agent 2 e os bancos de dados monitorados; usado se nenhum valor for especificado em uma sessão nomeada.
Plugins.Mysql.Default.TLSConnect	não			Tipo de criptografia para comunicações entre o Zabbix agent 2 e os bancos de dados monitorados; usado se nenhum valor for especificado em uma sessão nomeada. Valores suportados: *required* - requer conexão TLS; *verify_ca* - verifica certificados; *verify_full* - verifica certificados e endereço IP.
Plugins.Mysql.Default.TLSKeyFile	não (sim, se Plugins.Mysql.Default.TLSConnect estiver definido como *verify_ca* ou *verify_full*)			Caminho completo para um arquivo contendo a chave privada do banco de dados para comunicações criptografadas entre o Zabbix agent 2 e os bancos de dados monitorados; usado se nenhum valor for especificado em uma sessão nomeada.

Plugins.Mysql.Default.Uri	não		tcp://localhost:3306	URI padrão para conectar ao MySQL; usado se nenhum valor for especificado na chave do item ou na sessão nomeada.
				
				Não deve incluir credenciais incorporadas (serão ignoradas).
				Deve corresponder ao formato URI.
				Esquemas suportados: `tcp`, `unix`; um esquema pode ser omitido.
				Uma porta pode ser omitida (padrão=3306).
				Exemplos: `tcp://localhost:3306`
				`localhost`
				`unix:/var/run/mysql.sock`

Plugins.Mysql.Default.User	não			Nome de usuário padrão para conectar ao MySQL; usado se nenhum valor for especificado na chave do item ou na sessão nomeada.
Plugins.Mysql.KeepAlive	não	60-900	300	Tempo máximo de espera (em segundos) antes que conexões de plugin não utilizadas sejam fechadas.
Plugins.Mysql.Sessions.<SessionName>.Password	não			Senha da sessão nomeada. **<SessionName>** - define o nome de uma sessão para uso nas chaves de item.

Parâmetro	Obrigatório	Intervalo	Padrão	Descrição
Plugins.Mysql.Sessions.<SessionName>.TLSCAFile	não (sim, se Plugins.Mysql.Sessions.<SessionName>.TLSConnect estiver definido como *verify_ca* ou *verify_full*)			Caminho completo para um arquivo contendo os certificados da(s) CA(s) de nível superior para verificação do certificado do par, usado para comunicações criptografadas entre o Zabbix agent 2 e os bancos de dados monitorados. **<SessionName>** - define o nome de uma sessão para uso nas chaves de item.
Plugins.Mysql.Sessions.<SessionName>.TLSCertFile	não (sim, se Plugins.Mysql.Sessions.<SessionName>.TLSKeyFile for especificado)			Caminho completo para um arquivo contendo o certificado do agent ou cadeia de certificados, usado para comunicações criptografadas entre o Zabbix agent 2 e os bancos de dados monitorados. **<SessionName>** - define o nome de uma sessão para uso nas chaves de item.
Plugins.Mysql.Sessions.<SessionName>.TLSConnect	não			Tipo de criptografia para comunicações entre o Zabbix agent 2 e os bancos de dados monitorados. **<SessionName>** - define o nome de uma sessão para uso nas chaves de item. Valores suportados: *required* - requer conexão TLS; *verify_ca* - verifica certificados; *verify_full* - verifica certificados e endereço IP.
Plugins.Mysql.Sessions.<SessionName>.TLSKeyFile	sim, se Plugins.Mysql.Sessions.<SessionName>.TLSCertFile for especificado			Caminho completo para um arquivo contendo a chave privada do banco de dados usada para comunicações criptografadas entre o Zabbix agent 2 e os bancos de dados monitorados. **<SessionName>** - define o nome de uma sessão para uso nas chaves de item.

Plugins.Mysql.Sessions.<SessionName>.Uri	não			String de conexão de uma sessão nomeada.
				<SessionName> - define o nome de uma sessão para uso nas chaves de item.
				
				Não deve incluir credenciais incorporadas (serão ignoradas).
				Deve corresponder ao formato URI.
				Esquemas suportados: `tcp`, `unix`; um esquema pode ser omitido.
				Uma porta pode ser omitida (padrão=3306).
				Exemplos: `tcp://localhost:3306`
				`localhost`
				`unix:/var/run/mysql.sock`

Plugins.Mysql.Sessions.<SessionName>.User	não			Nome de usuário da sessão nomeada. **<SessionName>** - define o nome de uma sessão para uso nas chaves de item.
Plugins.Mysql.Timeout	não	1-30	timeout global	Tempo máximo em segundos para aguardar o estabelecimento de uma conexão.

Veja também:

- Descrição dos parâmetros gerais de configuração do Zabbix agent 2: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Instruções para configurar plugins

10 Plugin Oracle

Visão geral

Esta seção lista os parâmetros suportados no arquivo de configuração do plugin Oracle Zabbix agent 2 (oracle.conf).

Observe que:

- Os valores padrão refletem os padrões do processo, não os valores nos arquivos de configuração fornecidos;
- O Zabbix suporta arquivos de configuração apenas na codificação UTF-8 sem BOM;
- Comentários iniciados com ”#” são suportados apenas no início da linha.

Parâmetros

Parâmetro	Obrigatório	Intervalo	Padrão	Descrição
Plugins.Oracle.CallTimeout	não	1-30	timeout global	O tempo máximo de espera em segundos para que uma solicitação seja concluída.
Plugins.Oracle.ConnectTimeout	não	1-30	timeout global	O tempo máximo de espera em segundos para que uma conexão seja estabelecida.
Plugins.Oracle.CustomQueriesPath	não			Caminho completo de um diretório contendo arquivos .sql com consultas personalizadas. Desabilitado por padrão. Exemplo: `/etc/zabbix/oracle/sql`
Plugins.Oracle.Default.Password	não			Senha padrão para conexão com o Oracle; usada se nenhum valor for especificado em uma chave de item ou sessão nomeada.
Plugins.Oracle.Default.Service	não			Nome do serviço padrão para conexão com o Oracle (SID não é suportado); usado se nenhum valor for especificado em uma chave de item ou sessão nomeada.
Plugins.Oracle.Default.Uri	não		tcp://localhost:1521	URI padrão para conexão com o Oracle; usada se nenhum valor for especificado em uma chave de item ou sessão nomeada. Não deve incluir credenciais embutidas (serão ignoradas). Deve corresponder ao formato de URI. Apenas o esquema `tcp` é suportado; um esquema pode ser omitido. Uma porta pode ser omitida (padrão=1521). Desde o **Zabbix 7.0.19** também é possível especificar a chave TNS ou o valor TNS como a string de conexão. O valor TNS deve ser composto sem espaços em branco. Exemplos: `tcp://127.0.0.1:1521` `localhost` `zbx_tns_example` (chave TNS) `(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=localhost)(POR` (valor TNS)
Plugins.Oracle.Default.User	não			Nome de usuário padrão para conexão com o Oracle; usado se nenhum valor for especificado em uma chave de item ou sessão nomeada.
Plugins.Oracle.KeepAlive	não	60-900	300	O tempo máximo de espera (em segundos) antes que conexões de plugin não utilizadas sejam fechadas.
Plugins.Oracle.ResolveTNS	não	true / false	false	A opção especifica como interpretar a string de conexão (ConnString) na chave de métrica 'commonParams' para conectar ao servidor Oracle. Se estiver definida como false, o esquema e a porta padrão serão adicionados à ConnString (se ausentes). Se a opção estiver definida como true, o esquema e a porta padrão serão omitidos (a menos que explicitamente especificados na ConnString), e a ConnString será passada para o cliente Oracle como está. Se o cliente Oracle encontrar a ConnString no arquivo tnsnames.ora, a descrição da conexão encontrada será usada para conectar ao servidor Oracle. Este parâmetro é suportado desde o Zabbix 7.0.19.
Plugins.Oracle.Sessions.<SessionName>.Password	não			Senha da sessão nomeada. **<SessionName>** - define o nome de uma sessão para uso em chaves de item.
Plugins.Oracle.Sessions.<SessionName>.Service	não			Nome do serviço da sessão nomeada a ser usado para conexão (SID não é suportado). **<SessionName>** - define o nome de uma sessão para uso em chaves de item.

Parâmetro	Obrigatório	Intervalo	Padrão	Descrição

Plugins.Oracle.Sessions.<SessionName>.Uri				String de conexão da sessão nomeada para o Oracle.
				<SessionName> - define o nome de uma sessão para uso em chaves de item.
				
				Não deve incluir credenciais embutidas (serão ignoradas).
				Deve corresponder ao formato de URI.
				Apenas o esquema `tcp` é suportado; um esquema pode ser omitido.
				Uma porta pode ser omitida (padrão=1521).
				Desde o **Zabbix 7.0.19** também é possível especificar a chave TNS ou o valor TNS como a string de conexão. O valor TNS deve ser composto sem espaços em branco.
				Exemplos: `tcp://127.0.0.1:1521`
				`localhost`
				`zbx_tns_example` (chave TNS)
				`(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=localhost)(POR`
				(valor TNS)

Plugins.Oracle.Sessions.<SessionName>.User				Nome de usuário da sessão nomeada. **<SessionName>** - define o nome de uma sessão para uso em chaves de item.

Veja também:

- Descrição dos parâmetros gerais de configuração do Zabbix agent 2: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Instruções para configurar plugins

11 Plugin do PostgreSQL

Visão geral

Esta seção lista os parâmetros suportados no arquivo de configuração do plugin PostgreSQL Zabbix agent 2 (postgresql.conf).

O plugin PostgreSQL é um plugin carregável e está disponível e totalmente descrito no repositório do plugin PostgreSQL.

Observe que:

- Os valores padrão refletem os padrões do processo, não os valores nos arquivos de configuração fornecidos.
- O Zabbix suporta arquivos de configuração apenas em codificação UTF-8 sem BOM.
- Comentários iniciados com "#" são suportados apenas no início da linha.

Opções

Parâmetro	Descrição
-V --version	Imprime a versão do plugin e informações de licença.
-h --help	Imprime informações de ajuda (forma abreviada).

Parâmetros

Parâmetro	Obrigatório	Intervalo	Padrão	Descrição
Plugins.PostgreSQL.Default.CacheMode			prepare	Modo de cache para a conexão PostgreSQL. Valores suportados: *prepare* (padrão) - criará instruções preparadas no servidor PostgreSQL; *describe* - usará a instrução preparada anônima para descrever uma instrução sem criar uma instrução no servidor. Observe que "describe" é útil principalmente quando o ambiente não permite instruções preparadas, como ao executar um pool de conexões como o PgBouncer.
Plugins.PostgreSQL.CallTimeout		1-30	global timeout	Tempo máximo de espera (em segundos) para que uma solicitação seja concluída.

Parâmetro	Obrigatório	Intervalo	Padrão	Descrição
Plugins.PostgreSQL.CustomQueriesPath	Não		disabled	Caminho completo do diretório contendo arquivos *.sql* com consultas personalizadas.
Plugins.PostgreSQL.Default.Database	Não			Banco de dados padrão para conectar ao PostgreSQL; usado se nenhum valor for especificado em uma chave de item ou sessão nomeada.
Plugins.PostgreSQL.Default.Password	Não			Senha padrão para conectar ao PostgreSQL; usada se nenhum valor for especificado em uma chave de item ou sessão nomeada.
Plugins.PostgreSQL.Default.TLSCAFile	Não (sim, se Plugins.PostgreSQL.Default.TLSConnect estiver definido como *verify_ca* ou *verify_full*)			Caminho completo de um arquivo contendo o(s) certificado(s) da CA de nível superior para verificação do certificado do par para comunicações criptografadas entre o Zabbix agent 2 e os bancos de dados monitorados; usado se nenhum valor for especificado em uma sessão nomeada.
Plugins.PostgreSQL.Default.TLSCertFile	Não (sim, se Plugins.PostgreSQL.Default.TLSConnect estiver definido como *verify_ca* ou *verify_full*)			Caminho completo de um arquivo contendo o certificado PostgreSQL ou cadeia de certificados para comunicações criptografadas entre o Zabbix agent 2 e os bancos de dados monitorados; usado se nenhum valor for especificado em uma sessão nomeada.
Plugins.PostgreSQL.Default.TLSConnect	Não			Tipo de criptografia para comunicações entre o Zabbix agent 2 e os bancos de dados monitorados; usado se nenhum valor for especificado em uma sessão nomeada. Valores suportados: *required* - conectar usando TLS como modo de transporte sem verificações de identidade; *verify_ca* - conectar usando TLS e verificar o certificado; *verify_full* - conectar usando TLS, verificar o certificado e verificar se a identidade do banco de dados (CN) especificada por DBHost corresponde ao seu certificado. Tipo de criptografia indefinido significa conexão não criptografada.
Plugins.PostgreSQL.Default.TLSKeyFile	Não (sim, se Plugins.PostgreSQL.Default.TLSConnect estiver definido como *verify_ca* ou *verify_full*)			Caminho completo de um arquivo contendo a chave privada do PostgreSQL para comunicações criptografadas entre o Zabbix agent 2 e os bancos de dados monitorados; usado se nenhum valor for especificado em uma sessão nomeada.

Plugins.PostgreSQL.Default.Uri	Não			URI padrão para conectar ao PostgreSQL; usado se nenhum valor for especificado em uma chave de item ou sessão nomeada.
				
				Não deve incluir credenciais incorporadas (elas serão ignoradas).
				Deve corresponder ao formato URI.
				Esquemas suportados: `tcp`, `unix`.
				Exemplos: `tcp://127.0.0.1:5432`
				`tcp://localhost`
				`unix:/var/run/postgresql/.s.PGSQL.5432`

Plugins.PostgreSQL.Default.User	Não			Nome de usuário padrão para conectar ao PostgreSQL; usado se nenhum valor for especificado em uma chave de item ou sessão nomeada.
Plugins.PostgreSQL.KeepAlive	Não	60-900	300	Tempo máximo de espera (em segundos) antes que conexões de plugin não utilizadas sejam fechadas.

Parâmetro	Obrigatório	Intervalo	Padrão	Descrição
Plugins.PostgreSQL.Sessions.<SessionName>.CachePrepareMode	Não		prepare	Modo de cache para a conexão PostgreSQL. **<SessionName>** - define o nome de uma sessão para uso em chaves de item. Valores suportados: *prepare* (padrão) - criará instruções preparadas no servidor PostgreSQL; *describe* - usará a instrução preparada anônima para descrever uma instrução sem criar uma instrução no servidor. Observe que ”describe” é útil principalmente quando o ambiente não permite instruções preparadas, como ao executar um pool de conexões como o PgBouncer.
Plugins.PostgreSQL.Sessions.<SessionName>.Database	Não			Banco de dados para conexão de sessão. **<SessionName>** - define o nome de uma sessão para uso em chaves de item.
Plugins.PostgreSQL.Sessions.<SessionName>.Password	Não	Deve corresponder ao formato de senha.		Senha para conexão de sessão. **<SessionName>** - define o nome de uma sessão para uso em chaves de item.
Plugins.PostgreSQL.Sessions.<SessionName>.TLSCAFile	Não (sim, se Plugins.PostgreSQL.Sessions.<SessionName>.TLSConnect estiver definido como *verify_ca* ou *verify_full*)			Caminho completo de um arquivo contendo o(s) certificado(s) da CA de nível superior para verificação do certificado do par. **<SessionName>** - define o nome de uma sessão para uso em chaves de item.
Plugins.PostgreSQL.Sessions.<SessionName>.TLSCertFile	Não, se Plugins.PostgreSQL.Sessions.<SessionName>.TLSKeyFile for especificado			Caminho completo de um arquivo contendo o certificado PostgreSQL ou cadeia de certificados. **<SessionName>** - define o nome de uma sessão para uso em chaves de item.
Plugins.PostgreSQL.Sessions.<SessionName>.TLSConnect	Não			Tipo de criptografia para conexão PostgreSQL. **<SessionName>** - define o nome de uma sessão para uso em chaves de item. Valores suportados: *required* - conectar usando TLS como modo de transporte sem verificações de identidade; *verify_ca* - conectar usando TLS e verificar o certificado; *verify_full* - conectar usando TLS, verificar o certificado e verificar se a identidade do banco de dados (CN) especificada por DBHost corresponde ao seu certificado. Tipo de criptografia indefinido significa conexão não criptografada.
Plugins.PostgreSQL.Sessions.<SessionName>.TLSKeyFile	Não, se Plugins.PostgreSQL.Sessions.<SessionName>.TLSCertFile for especificado			Caminho completo de um arquivo contendo a chave privada do PostgreSQL. **<SessionName>** - define o nome de uma sessão para uso em chaves de item.

Plugins.PostgreSQL.Sessions.<SessionName>.Uri	Não			String de conexão de uma sessão nomeada.
				<SessionName> - define o nome de uma sessão para uso em chaves de item.
				
				Não deve incluir credenciais incorporadas (elas serão ignoradas).
				Deve corresponder ao formato URI.
				Esquemas suportados: `tcp`, `unix`.
				Exemplos: `tcp://127.0.0.1:5432`
				`tcp://localhost`
				`unix:/var/run/postgresql/.s.PGSQL.5432`

Plugins.PostgreSQL.Sessions.<SessionName>.User	Não			Nome de usuário da sessão nomeada. **<SessionName>** - define o nome de uma sessão para uso em chaves de item.

Parâmetro	Obrigatório	Intervalo	Padrão	Descrição
Plugins.PostgreSQL.System.Path	Não			Caminho para o executável do plugin externo.
Plugins.PostgreSQL.Timeout	Não	1-30	global timeout	Tempo limite de execução da solicitação (a duração, em segundos, para aguardar a conclusão de uma solicitação antes de encerrá-la).

Veja também:

- Descrição dos parâmetros gerais de configuração do Zabbix agent 2: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Instruções para configurar plugins

12 Plugin Redis

Visão geral

Esta seção lista os parâmetros suportados no arquivo de configuração do plugin Redis Zabbix agent 2 (redis.conf).

Observe que:

- Os valores padrão refletem os padrões do processo, não os valores nos arquivos de configuração fornecidos.
- O Zabbix suporta arquivos de configuração apenas na codificação UTF-8 sem BOM.
- Comentários iniciados com "#" são suportados apenas no início da linha.

Parâmetros

Parâmetro	Obrigatório	Intervalo	Padrão	Descrição
Plugins.Redis.Default.Password	Não			Senha padrão para conexão com o Redis; usada se nenhum valor for especificado na chave do item ou na sessão nomeada.
Plugins.Redis.Default.TLSConnect	Não			Requisito de verificação TLS do cliente (Zabbix agent 2). Valores suportados: `required` - criptografado, não verificado (apenas para testes) `verify_ca` - criptografado, certificado do servidor `verify_full` - criptografado, certificado do servidor e nome do servidor verificado com SAN Um valor vazio ou não definido é tratado como sem tls.
Plugins.Redis.Default.TLSCAFile	Não			Caminho completo padrão de um arquivo contendo os certificados da(s) CA(s) de nível superior.
Plugins.Redis.Default.TLSCertFile	Não			Caminho completo padrão de um arquivo contendo o certificado do agent.
Plugins.Redis.Default.TLSKeyFile	Não			Caminho completo padrão de um arquivo contendo a chave privada do agent.

Plugins.Redis.Default.Uri	Não		tcp://localhost:6379	URI padrão para conexão com o Redis; usada se nenhum valor for especificado na chave do item ou na sessão nomeada.
				
				Não deve incluir credenciais incorporadas (serão ignoradas).
				Deve corresponder ao formato URI.
				Esquemas suportados: `tcp`, `unix`; um esquema pode ser omitido.
				Uma porta pode ser omitida (padrão=6379).
				Exemplos: `tcp://localhost:6379`
				`localhost`
				`unix:/var/run/redis.sock`

Plugins.Redis.Default.User	Não		default	Usuário padrão para enviar ao servidor Redis protegido; usado se nenhum valor for especificado na chave do item ou na sessão nomeada.
Plugins.Redis.KeepAlive	Não	60-900	300	Tempo máximo de espera (em segundos) antes que conexões de plugin não utilizadas sejam fechadas.
Plugins.Redis.Sessions.<SessionName>.Password	Não			Senha da sessão nomeada. **<SessionName>** - defina o nome da sessão a ser usado nas chaves dos itens.

Parâmetro	Obrigatório	Intervalo	Padrão	Descrição
Plugins.Redis.Sessions.<SessionName>.TLSConnect	não			Requisito de verificação TLS do cliente (Zabbix agent 2). **<SessionName>** - defina o nome da sessão a ser usado nas chaves dos itens. Valores suportados: `required` - criptografado, não verificado (apenas para testes) `verify_ca` - criptografado, certificado do servidor `verify_full` - criptografado, certificado do servidor e nome do servidor verificado com SAN Um valor vazio ou não definido é tratado como sem tls.
Plugins.Redis.Sessions.<SessionName>.TLSCAFile	não (sim, se TLSConnect for `verify_ca` ou `verify_full`)			Caminho completo de um arquivo contendo os certificados da(s) CA(s) de nível superior para verificação do certificado do servidor Redis. **<SessionName>** - defina o nome da sessão a ser usado nas chaves dos itens.
Plugins.Redis.Sessions.<SessionName>.TLSCertFile	não			Caminho completo de um arquivo contendo o certificado do agent para autenticação do cliente. **<SessionName>** - defina o nome da sessão a ser usado nas chaves dos itens.
Plugins.Redis.Sessions.<SessionName>.TLSKeyFile	não			Caminho completo de um arquivo contendo a chave privada do agent para autenticação do cliente. **<SessionName>** - defina o nome da sessão a ser usado nas chaves dos itens.

Plugins.Redis.Sessions.<SessionName>.Uri	não		localhost:6379	String de conexão de uma sessão nomeada.
				<SessionName> - defina o nome da sessão a ser usado nas chaves dos itens.
				
				Não deve incluir credenciais incorporadas (serão ignoradas).
				Deve corresponder ao formato URI.
				Esquemas suportados: `tcp`, `unix`; um esquema pode ser omitido.
				Uma porta pode ser omitida (padrão=6379).
				Exemplos: `tcp://localhost:6379`
				`localhost`
				`unix:/var/run/redis.sock`

Plugins.Redis.Sessions.<SessionName>.User	não		default	Usuário para enviar ao servidor Redis protegido. **<SessionName>** - defina o nome da sessão a ser usado nas chaves dos itens. Suportado desde a versão 7.0.10.
Plugins.Redis.Timeout	não	1-30	timeout global	Tempo limite de execução da solicitação (a duração, em segundos, para aguardar a conclusão de uma solicitação antes de encerrá-la).

Veja também:

- Descrição dos parâmetros gerais de configuração do Zabbix agent 2: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Instruções para configurar plugins

13 Plugin SMART

Visão geral

Esta seção lista os parâmetros suportados no arquivo de configuração do plugin SMART do Zabbix agent 2 (smart.conf).

Observe que:

- Os valores padrão refletem os padrões do processo, não os valores nos arquivos de configuração fornecidos;
- O caminho para o executável `smartctl` deve ser definido adicionando seu diretório à variável de ambiente `PATH` do sistema ou configurando `Plugins.Smart.Path`; isso se aplica tanto ao Linux quanto ao Windows;

- O Zabbix suporta arquivos de configuração apenas na codificação UTF-8 sem BOM;
- Comentários iniciados com "#" são suportados apenas no início da linha.

Parâmetros

Parâmetro	Obrigatório	Intervalo	Padrão	Descrição
Plugins.Smart.Path	não		smartctl	Caminho para o executável smartctl.
Plugins.Smart.Timeout	não	1-30	timeout global	Timeout de execução da requisição (a duração, em segundos, para aguardar a conclusão de uma requisição antes de encerrá-la).

Veja também:

- Descrição dos parâmetros gerais de configuração do Zabbix agent 2: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Instruções para configurar plugins

8 Zabbix Java gateway

Se você usar os scripts `startup.sh` e `shutdown.sh` para iniciar o Zabbix Java gateway, então você pode especificar os parâmetros de configuração necessários no arquivo `settings.sh`. Os scripts de inicialização e desligamento utilizam o arquivo de configurações e convertem as variáveis de shell (listadas na primeira coluna) em propriedades Java (listadas na segunda coluna).

Se você iniciar o Zabbix Java gateway manualmente executando `java` diretamente, então você deve especificar as propriedades Java correspondentes na linha de comando.

Variável	Propriedade	Obrigatório	Intervalo	Padrão	Descrição
LISTEN_IP	zabbix.listenIP	não		0.0.0.0	Endereço IP para escutar.
LISTEN_PORT	zabbix.listenPort	não	1024-32767	10052	Porta para escutar.
PID_FILE	zabbix.pidFile	não		/tmp/zabbix_java.pid	Nome do arquivo PID. Se omitido, o Zabbix Java Gateway é iniciado como uma aplicação de console.
PROPERTIES_FILE	zabbix.propertiesFile	não			Nome do arquivo de propriedades. Pode ser usado para definir propriedades adicionais usando um formato de chave-valor, de forma que não fiquem visíveis na linha de comando ou para sobrescrever as existentes. Por exemplo: "javax.net.ssl.trustStorePassword=<s
START_POLLERS	zabbix.startPollers	não	1-1000	5	Número de threads de trabalho a serem iniciadas.
TIMEOUT	zabbix.timeout	não	1-30	3	Tempo de espera para operações de rede (em segundos).

Warning:
A porta 10052 não é registrada na IANA.

9 Serviço web do Zabbix

Visão geral

O serviço web do Zabbix é um processo utilizado para comunicação com serviços web externos.

Os parâmetros suportados pelo arquivo de configuração do serviço web do Zabbix (zabbix_web_service.conf) estão listados nesta seção.

Os parâmetros são listados sem informações adicionais. Clique no parâmetro para ver os detalhes completos.

Parâmetro	Descrição
AllowedIP	Uma lista de endereços IP separados por vírgula, opcionalmente na notação CIDR, ou nomes DNS dos servidores Zabbix e proxies Zabbix.
DebugLevel	O nível de debug.
IgnoreURLCertErrors	Especifica o tratamento de erros de validação de certificado TLS ao acessar a URL do frontend.
Include	Você pode incluir arquivos individuais ou todos os arquivos de um diretório no arquivo de configuração.
ListenPort	O serviço irá escutar neste porto por conexões do server.
LogFile	O nome do arquivo de log.
LogFileSize	O tamanho máximo do arquivo de log.
LogType	O tipo de saída do log.
Timeout	O tempo máximo (em segundos) gasto para formatar o relatório em PDF de um dashboard.
TLSAccept	Quais conexões de entrada aceitar.
TLSCAFile	O caminho completo de um arquivo contendo o(s) certificado(s) da(s) CA(s) de nível superior para verificação do certificado do peer, usado para comunicações criptografadas entre componentes do Zabbix.
TLSCertFile	O caminho completo de um arquivo contendo o certificado do serviço ou cadeia de certificados, usado para comunicações criptografadas entre componentes do Zabbix.
TLSKeyFile	O caminho completo de um arquivo contendo a chave privada do serviço, usado para comunicações criptografadas entre componentes do Zabbix.

Todos os parâmetros são não obrigatórios, a menos que explicitamente declarado que o parâmetro é obrigatório.

Observe que:

- Os valores padrão refletem os padrões do processo, não os valores nos arquivos de configuração fornecidos;
- O Zabbix suporta arquivos de configuração apenas na codificação UTF-8 sem BOM;
- Comentários iniciados com ”#” são suportados apenas no início da linha.

Detalhes do parâmetro

AllowedIP

Uma lista de endereços IP separados por vírgula, opcionalmente na notação CIDR, ou nomes DNS de servidores Zabbix e proxies Zabbix. As conexões recebidas serão aceitas apenas dos hosts listados aqui.
Se o suporte a IPv6 estiver habilitado, então `127.0.0.1`, `::127.0.0.1`, `::ffff:127.0.0.1` são tratados igualmente e `::/0` permitirá qualquer endereço IPv4 ou IPv6. `0.0.0.0/0` pode ser usado para permitir qualquer endereço IPv4.

Exemplo:

```
127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com
```

Obrigatório: sim

DebugLevel

Especifique o nível de depuração:
0 - informações básicas sobre o início e a parada dos processos do Zabbix
1 - informações críticas;
2 - informações de erro;
3 - avisos;
4 - para depuração (gera muitas informações);
5 - depuração estendida (gera ainda mais informações).

Padrão: 3
 Intervalo: 0-5

IgnoreURLCertErrors

Especifica o tratamento de erros de validação de certificado TLS ao acessar a URL do frontend:
0 - não ignorar erros de certificado;
1 - ignorar erros de certificado.

Padrão: 0
 Intervalo: 0-1

Include

Você pode incluir arquivos individuais ou todos os arquivos em um diretório no arquivo de configuração. Durante a instalação, o Zabbix criará o diretório de inclusão em /usr/local/etc, a menos que seja modificado durante a compilação. O caminho pode ser

relativo à localização do arquivo *zabbix_web_service.conf*.
Para incluir apenas arquivos relevantes no diretório especificado, o caractere curinga asterisco é suportado para correspondência de padrões.
Veja notas especiais sobre limitações.

Exemplo:

```
Include=/absolute/path/to/config/files/*.conf
```

ListenPort

O serviço irá escutar neste porto por conexões do server.

Padrão: 10053
 Intervalo: 1024-32767

LogFile

O nome do arquivo de log.

Exemplo:

`/tmp/zabbix_web_service.log`

Obrigatório: Sim, se LogType estiver definido como *file*; caso contrário, não

LogFileSize

O tamanho máximo de um arquivo de log em MB.
0 - desabilita a rotação automática de log.
Nota: Se o limite de tamanho do arquivo de log for atingido e a rotação do arquivo falhar, por qualquer motivo, o arquivo de log existente será truncado e iniciado novamente.

Padrão: 1
 Intervalo: 0-1024

LogType

O tipo de saída do log:
file - grava o log no arquivo especificado pelo parâmetro LogFile;
system - grava o log no syslog;
console - grava o log na saída padrão.

Padrão: `file`

Timeout

O tempo máximo (em segundos) gasto na formatação do relatório em PDF de um dashboard.

Padrão: 10
 Intervalo: 1-30

TLSAccept

Quais conexões de entrada aceitar:
unencrypted - aceitar conexões sem criptografia (padrão)
cert - aceitar conexões com TLS e um certificado

Padrão: `unencrypted`

TLSCAFile

O caminho completo do arquivo contendo os certificados da(s) CA(s) de nível superior para verificação do certificado do par, usado para comunicações criptografadas entre os componentes do Zabbix.

TLSCertFile

O caminho completo do arquivo que contém o certificado do serviço ou cadeia de certificados, usado para comunicações criptografadas com os componentes do Zabbix.

TLSKeyFile

O caminho completo do arquivo que contém a chave privada do serviço, usada para comunicações criptografadas entre os componentes do Zabbix.

10 Inclusão

Visão geral

Arquivos ou diretórios adicionais podem ser incluídos na configuração do server, proxy, agent e web service usando o parâmetro `Include`.

Notas sobre inclusão

Se o parâmetro `Include` for usado para incluir um arquivo, o arquivo deve ser legível.

Se o parâmetro `Include` for usado para incluir um diretório:

- Todos os arquivos no diretório devem ser legíveis.
- Não se deve assumir nenhuma ordem específica de inclusão (por exemplo, os arquivos não são incluídos em ordem alfabética). Portanto, não defina um parâmetro em vários arquivos ”Include” (por exemplo, para substituir uma configuração geral por uma específica).
- Todos os arquivos no diretório são incluídos na configuração.
- Cuidado com cópias de backup de arquivos criadas automaticamente por alguns editores de texto. Por exemplo, se ao editar o arquivo ”include/my_specific.conf” for criada uma cópia de backup ”include/my_specific_conf.BAK”, ambos os arquivos serão incluídos. Mova o arquivo ”include/my_specific.conf.BAK” para fora do diretório ”Include”. No Linux, o conteúdo do diretório ”Include” pode ser verificado com o comando ”ls -al” para arquivos desnecessários.

Se o parâmetro `Include` for usado para incluir arquivos usando um padrão:

- Todos os arquivos que correspondam ao padrão devem ser legíveis.
- Não se deve assumir nenhuma ordem específica de inclusão (por exemplo, os arquivos não são incluídos em ordem alfabética). Portanto, não defina um parâmetro em vários arquivos ”Include” (por exemplo, para substituir uma configuração geral por uma específica).

3 Protocolos

Use a barra lateral para acessar o conteúdo desta seção.

1 Protocolo de troca de dados servidor-proxy

Visão geral

Servidor - a troca de dados do proxy é baseada no formato JSON.

As mensagens de solicitação e resposta devem começar com cabeçalho e dados comprimento.

Proxy passivo

Solicitação de configuração

O server enviará primeiro uma solicitação vazia de `proxy config`. Esta solicitação é enviada a cada `ProxyConfigFrequency` (segundos do parâmetro de configuração do server).

O proxy responde com a versão atual do proxy, token de sessão e revisão da configuração. O server responde com os dados de configuração que precisam ser atualizados.

nome	tipo de valor	descrição
server→proxy:		
request	*string*	’proxy config’
proxy→server:		
version	*string*	Versão do proxy (<major>.<minor>.<build>).
session	*string*	Token de sessão de configuração do proxy.
config_revision	*number*	Revisão da configuração do proxy.
server→proxy:		
full_sync	*number*	1 - se todos os dados de configuração forem enviados; ausente - caso contrário (opcional).
data	*array*	Objeto de dados da tabela. Ausente se a configuração não foi alterada (opcional).
<table>	*object*	Um ou mais objetos com dados da tabela <table> (opcional, dependendo das alterações).
fields	*array*	Array de nomes de campos.
-	*string*	Nome do campo.
data	*array*	Array de linhas.
-	*array*	Array de colunas.
-	*string*,*number*	Valor da coluna com tipo dependendo do tipo da coluna no esquema do banco de dados.
macro.secrets	*object*	Informações de macro secreta, ausente se não houver alterações em macros de cofre (opcional).

nome	tipo de valor	descrição
config_revision	*number*	Revisão do cache de configuração - enviada com os dados de configuração (opcional).
del_hostids	*array*	Array de hostids removidos (opcional).
-	*number*	Identificador do host.
del_macro_hostids	*array*	Array de hostids com todas as macros removidas (opcional).
-	*number*	Identificador do host.
proxy→server:		
response	*string*	Informação de sucesso da solicitação ('success' ou 'failed').
version	*string*	Versão do proxy (<major>.<minor>.<build>).

Exemplo:

server→proxy:

```
{
  "request":"proxy config"
}
```

proxy→server:

```
{
  "version": "7.0.0",
  "session": "0033124949800811e5686dbfd9bcea98",
  "config_revision": 0
}
```

server→proxy:

```
{
"full_sync": 1,
"data": {
"hosts": {
"fields": ["hostid", "host", "status", "ipmi_authtype", "ipmi_privilege", "ipmi_username", "ipmi_password"
"data": [
[10084, "Zabbix server", 0, -1, 2, "", "", "Zabbix server", 1, 1, "", "", "", ""]
]
},
"interface": {
"fields": ["interfaceid", "hostid", "main", "type", "useip", "ip", "dns", "port", "available"],
"data": [
[1, 10084, 1, 1, 1, "127.0.0.1", "", "10053", 1]
]
},
"interface_snmp": {
"fields": ["interfaceid", "version", "bulk", "community", "securityname", "securitylevel", "authpassphrase
"data": []
},
"host_inventory": {
"fields": ["hostid", "type", "type_full", "name", "alias", "os", "os_full", "os_short", "serialno_a", "ser
"data": [
[10084, "", "", "",
```

```
"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]
]
},
"config": {
"fields": ["configid", "snmptrap_logging", "hk_history_global", "hk_history", "autoreg_tls_accept"],
"data": [
[1, 1, 0, "90d", 1]
]
},
"httptest": {
"fields": ["httptestid", "name", "delay", "agent", "authentication", "http_user", "http_password", "hostid
"data": []
},
"httptestitem": {
"fields": ["httptestitemid", "httptestid", "itemid", "type"],
"data": []
},
"httptest_field": {
"fields": ["httptest_fieldid", "httptestid", "type", "name", "value"],
"data": []
},
"httpstep": {
"fields": ["httpstepid", "httptestid", "name", "no", "url", "timeout", "posts", "required", "status_codes"
"data": []
},
"httpstepitem": {
"fields": ["httpstepitemid", "httpstepid", "itemid", "type"],
"data": []
},
"httpstep_field": {
"fields": ["httpstep_fieldid", "httpstepid", "type", "name", "value"],
"data": []
},
"config_autoreg_tls": {
"fields": ["autoreg_tlsid", "tls_psk_identity", "tls_psk"],
"data": [
[1, "", ""]
]
}
},
"macro.secrets": {
"AppID=zabbix_server&Query=Safe=passwordSafe;Object=zabbix": {
"Content": "738"
}
},
"config_revision": 2
}
```


proxy→server:


```
{
  "response": "success",
  "version": "7.0.0"
}
```


Solicitação de dados

A solicitação `proxy data` é usada para obter a disponibilidade da interface do host, dados históricos, de descoberta e de autoregistro do proxy. Esta solicitação é enviada a cada `ProxyDataFrequency` (parâmetro de configuração do server) segundos.

nome	tipo de valor	descrição
server→proxy:		
request	*string*	'proxy data'

nome	tipo de valor	descrição
proxy→server:		
session	*string*	Token de sessão de dados.
interface availability	*array*	*(opcional)* Array de objetos de dados de disponibilidade de interface.
interfaceid	*number*	Identificador da interface.
available	*number*	Disponibilidade da interface: **0**, *INTERFACE_AVAILABLE_UNKNOWN* - desconhecida **1**, *INTERFACE_AVAILABLE_TRUE* - disponível **2**, *INTERFACE_AVAILABLE_FALSE* - indisponível
error	*string*	Mensagem de erro da interface ou string vazia.
history data	*array*	*(opcional)* Array de objetos de dados históricos.
itemid	*number*	Identificador do item.
clock	*number*	Timestamp do valor do item (segundos).
ns	*number*	Timestamp do valor do item (nanossegundos).
value	*string*	*(opcional)* Valor do item.
id	*number*	Identificador do valor (contador ascendente, único dentro de uma sessão de dados).
timestamp	*number*	*(opcional)* Timestamp de items do tipo log.
source	*string*	*(opcional)* Valor da fonte do item eventlog.
severity	*number*	*(opcional)* Valor de severidade do item eventlog.
eventid	*number*	*(opcional)* Valor eventid do item eventlog.
state	*string*	*(opcional)* Estado do item: **0**, *ITEM_STATE_NORMAL* **1**, *ITEM_STATE_NOTSUPPORTED*
lastlogsize	*number*	*(opcional)* Último tamanho do log de items do tipo log.
mtime	*number*	*(opcional)* Tempo de modificação de items do tipo log.
discovery data	*array*	*(opcional)* Array de objetos de dados de descoberta.
clock	*number*	Timestamp dos dados de descoberta.
druleid	*number*	Identificador da regra de descoberta.
dcheckid	*number*	Identificador da verificação de descoberta ou nulo para dados da regra de descoberta.
type	*number*	Tipo de verificação de descoberta: **-1** dados da regra de descoberta **0**, *SVC_SSH* - verificação de serviço SSH **1**, *SVC_LDAP* - verificação de serviço LDAP **2**, *SVC_SMTP* - verificação de serviço SMTP **3**, *SVC_FTP* - verificação de serviço FTP **4**, *SVC_HTTP* - verificação de serviço HTTP **5**, *SVC_POP* - verificação de serviço POP **6**, *SVC_NNTP* - verificação de serviço NNTP **7**, *SVC_IMAP* - verificação de serviço IMAP **8**, *SVC_TCP* - verificação de disponibilidade de porta TCP **9**, *SVC_AGENT* - Zabbix agent **10**, *SVC_SNMPv1* - SNMPv1 agent **11**, *SVC_SNMPv2* - SNMPv2 agent **12**, *SVC_ICMPPING* - ICMP ping **13**, *SVC_SNMPv3* - SNMPv3 agent **14**, *SVC_HTTPS* - verificação de serviço HTTPS **15**, *SVC_TELNET* - verificação de disponibilidade Telnet
ip	*string*	Endereço IP do host.
dns	*string*	Nome DNS do host.
port	*number*	*(opcional)* Número da porta do serviço.
key_	*string*	*(opcional)* Chave do item para verificação de descoberta do tipo **9** *SVC_AGENT*
value	*string*	*(opcional)* Valor recebido do serviço, pode estar vazio para a maioria dos serviços.

nome	tipo de valor	descrição
status	*number*	*(opcional)* Status do serviço: **0**, *DOBJECT_STATUS_UP* - Serviço UP **1**, *DOBJECT_STATUS_DOWN* - Serviço DOWN
auto reg- is- tra- tion	*array*	*(opcional)* Array de objetos de dados de autoregistro.
clock	*number*	Timestamp dos dados de autoregistro.
host	*string*	Nome do host.
ip	*string*	*(opcional)* Endereço IP do host.
dns	*string*	*(opcional)* Nome DNS resolvido a partir do endereço IP.
port	*string*	*(opcional)* Porta do host.
host_metadata	*string*	*(opcional)* Metadados do host enviados pelo agent (baseado no parâmetro de configuração do agent HostMetadata ou HostMetadataItem).
tasks	*array*	*(opcional)* Array de tarefas.
type	*number*	Tipo de tarefa: **0**, *ZBX_TM_TASK_PROCESS_REMOTE_COMMAND_RESULT* - resultado do comando remoto
status	*number*	Status da execução do comando remoto: **0**, *ZBX_TM_REMOTE_COMMAND_COMPLETED* - comando remoto concluído com sucesso **1**, *ZBX_TM_REMOTE_COMMAND_FAILED* - comando remoto falhou
error	*string*	*(opcional)* Mensagem de erro.
parent_taskid	*number*	ID da tarefa pai.
more	*number*	*(opcional)* 1 - há mais dados históricos para enviar.
clock	*number*	*(opcional)* Timestamp da transferência de dados (segundos).
ns	*number*	*(opcional)* Timestamp da transferência de dados (nanossegundos).
version	*string*	Versão do proxy (<major>.<minor>.<build>).
server→proxy:		
response	*string*	Informação de sucesso da solicitação (’success’ ou ’failed’).
tasks	*array*	*(opcional)* Array de tarefas.
type	*number*	Tipo de tarefa: **1**, *ZBX_TM_TASK_PROCESS_REMOTE_COMMAND* - comando remoto
clock	*number*	Hora de criação da tarefa.
ttl	*number*	Tempo em segundos após o qual a tarefa expira.
commandtype	*number*	Tipo de comando remoto: **0**, *ZBX_SCRIPT_TYPE_CUSTOM_SCRIPT* - usar script customizado **1**, *ZBX_SCRIPT_TYPE_IPMI* - usar IPMI **2**, *ZBX_SCRIPT_TYPE_SSH* - usar SSH **3**, *ZBX_SCRIPT_TYPE_TELNET* - usar Telnet **4**, *ZBX_SCRIPT_TYPE_GLOBAL_SCRIPT* - usar script global (atualmente funcionalmente equivalente ao script customizado)
command	*string*	Comando remoto a ser executado.
execute_on	*number*	Destino de execução para scripts customizados: **0**, *ZBX_SCRIPT_EXECUTE_ON_AGENT* - executar script no agent **1**, *ZBX_SCRIPT_EXECUTE_ON_SERVER* - executar script no server **2**, *ZBX_SCRIPT_EXECUTE_ON_PROXY* - executar script no proxy
port	*number*	*(opcional)* Porta para comandos Telnet e SSH.
authtype	*number*	*(opcional)* Tipo de autenticação para comandos SSH.
username	*string*	*(opcional)* Nome de usuário para comandos Telnet e SSH.
password	*string*	*(opcional)* Senha para comandos Telnet e SSH.
publickey	*string*	*(opcional)* Chave pública para comandos SSH.
privatekey	*string*	*(opcional)* Chave privada para comandos SSH.
parent_taskid	*number*	ID da tarefa pai.

nome	tipo de valor	descrição
hostid	*number*	ID do host de destino.

Exemplo:

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":"7.0.0"
}
```

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
        }
    ]
}
```

Solicitação de tarefas

A troca de `proxy tasks` lida com a entrega e confirmação de tarefas de comando remoto entre o server e um proxy passivo. Quando o server consulta o proxy para obter os resultados das tarefas, ele envia uma solicitação vazia de `proxy tasks` no intervalo definido por `ProxyDataFrequency`. O proxy responde com sua versão atual e quaisquer resultados de tarefas pendentes (incluindo carimbos de data/hora). O server então confirma o recebimento retornando uma `response` e pode incluir novas tarefas para o proxy executar.

nome	tipo de valor	descrição
server→proxy:		
request	*string*	'proxy tasks'
proxy→server:		
version	*string*	Versão do proxy (<major>.<minor>.<build>).
clock	*number*	*(opcional)* Carimbo de data/hora da transferência de dados (segundos).
ns	*number*	*(opcional)* Carimbo de data/hora da transferência de dados (nanossegundos).
server→proxy:		
response	*string*	Informação de sucesso da solicitação ('success' ou 'failed').

Exemplo:

server→proxy:

```
{
  "request":"proxy tasks"
}
```

proxy→server:

```
{
  "version":"7.0.0",
  "clock":1721059872,
  "ns":848141522
}
```

server→proxy:

```
{
  "response":"success"
}
```

Proxy ativo

Solicitação de configuração

A solicitação `proxy config` é enviada pelo proxy ativo para obter os dados de configuração do proxy. Esta solicitação é enviada a cada `ProxyConfigFrequency` (parâmetro de configuração do proxy) segundos.

nome	tipo de valor	descrição
proxy→server:		
request	*string*	'proxy config'
host	*string *	Nome do proxy.
version	*string*	Versão do proxy (<major>.<minor>.<build>).
session	*string*	Token de sessão de configuração do proxy.
config_revision	*number*	Revisão da configuração do proxy.
hostmap_revision	*number*	Revisão atual do mapeamento host-para-proxy.
server→proxy:		
full_sync	*number*	1 - se os dados completos de configuração forem enviados, ausente caso contrário (opcional).
data	*array*	Objeto de dados da tabela. Ausente se a configuração não foi alterada (opcional).
<table>	*object*	Um ou mais objetos com dados da <table> (opcional, dependendo das alterações).
fields	*array*	Array de nomes de campos.
-	*string*	Nome do campo.
data	*array*	Array de linhas.
-	*array*	Array de colunas.
-	*string,number*	Valor da coluna com tipo dependendo do tipo da coluna no esquema do banco de dados.
macro.secrets	*object*	Informação de macro secreta, ausente se não houver alterações em macros de cofre (opcional).
proxy_group	*string*	O nome do grupo de proxy ao qual o proxy pertence.
config_revision	*number*	Revisão do cache de configuração - enviada com os dados de configuração (opcional).
del_hostids	*array*	Array de hostids removidos (opcional).
-	*number*	Identificador do host.
del_macro_hostids	*array*	Array de hostids com todas as macros removidas (opcional).
-	*number*	Identificador do host.

Exemplo:

proxy→server:

```
{
  "request": "proxy config",
  "host": "Zabbix proxy",
  "version":"7.0.0",
  "session": "fd59a09ff4e9d1fb447de1f04599bcf6",
  "config_revision": 0
}
```

server→proxy:

```
{
"full_sync": 1,
"data": {
"hosts": {
"fields": ["hostid", "host", "status", "ipmi_authtype", "ipmi_privilege", "ipmi_username", "ipmi_password"
"data": [
[10084, "Zabbix server", 0, -1, 2, "", "", "Zabbix server", 1, 1, "", "", "", ""]
]
},
"interface": {
"fields": ["interfaceid", "hostid", "main", "type", "useip", "ip", "dns", "port", "available"],
"data": [
[1, 10084, 1, 1, 1, "127.0.0.1", "", "10053", 1]
]
},
"interface_snmp": {
```

```
"fields": ["interfaceid", "version", "bulk", "community", "securityname", "securitylevel", "authpassphrase
"data": []
},
"host_inventory": {
"fields": ["hostid", "type", "type_full", "name", "alias", "os", "os_full", "os_short", "serialno_a", "ser
"data": [
[10084, "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "5
]
},
"items": {
"fields": ["itemid", "type", "snmp_oid", "hostid", "key_", "delay", "history", "status", "value_type", "tr
"data": [
[44161, 7, "", 10084, "agent.hostmetadata", "10s", "90d", 0, 1, "", "", "", "", 0, "", "", "", "", 0, null
[44162, 0, "", 10084, "agent.ping", "10s", "90d", 0, 3, "", "", "", "", 0, "", "", "", "", 0, 1, 0, "", nu
]
},
"item_rtdata": {
"fields": ["itemid", "lastlogsize", "mtime"],
"data": [
[44161, 0, 0],
[44162, 0, 0]
]
},
"item_preproc": {
"fields": ["item_preprocid", "itemid", "step", "type", "params", "error_handler", "error_handler_params"],
"data": []
},
"item_parameter": {
"fields": ["item_parameterid", "itemid", "name", "value"],
"data": []
},
"globalmacro": {
"fields": ["globalmacroid", "macro", "value", "type"],
"data": [
[2, "{$SNMP_COMMUNITY}", "public", 0]
]
},
"hosts_templates": {
"fields": ["hosttemplateid", "hostid", "templateid", "link_type"],
"data": []
},
"hostmacro": {
"fields": ["hostmacroid", "hostid", "macro", "value", "type", "automatic"],
"data": [
[5676, 10084, "{$M}", "AppID=zabbix_server&Query=Safe=passwordSafe;Object=zabbix:Content", 2, 0]
]
},
"drules": {
"fields": ["druleid", "name", "iprange", "delay"],
"data": [
[2, "Local network", "127.0.0.1", "10s"]
]
},
"dchecks": {
"fields": ["dcheckid", "druleid", "type", "key_", "snmp_community", "ports", "snmpv3_securityname", "snmpv
"data": [
[2, 2, 9, "system.uname", "", "10052", "", 0, "", "", 0, 0, 0, "", 1, 0]
]
},
"regexps": {
"fields": ["regexpid", "name"],
"data": [
```

```
[1, "File systems for discovery"],
[2, "Network interfaces for discovery"],
[3, "Storage devices for SNMP discovery"],
[4, "Windows service names for discovery"],
[5, "Windows service startup states for discovery"]
]
},
"expressions": {
"fields": ["expressionid", "regexpid", "expression", "expression_type", "exp_delimiter", "case_sensitive"]
"data": [
[1, 1, "^(btrfs|ext2|ext3|ext4|reiser|xfs|ffs|ufs|jfs|jfs2|vxfs|hfs|apfs|refs|ntfs|fat32|zfs)$", 3, ",", 0
[3, 3, "^(Physical memory|Virtual memory|Memory buffers|Cached memory|Swap space)$", 4, ",", 1],
[5, 4, "^(MMCSS|gupdate|SysmonLog|clr_optimization_v2.0.50727_32|clr_optimization_v4.0.30319_32)$", 4, ","
[6, 5, "^(automatic|automatic delayed)$", 3, ",", 1],
[7, 2, "^Software Loopback Interface", 4, ",", 1],
[8, 2, "^(In)?[Ll]oop[Bb]ack[0-9._]*$", 4, ",", 1],
[9, 2, "^NULL[0-9.]*$", 4, ",", 1],
[10, 2, "^[Ll]o[0-9.]*$", 4, ",", 1],
[11, 2, "^[Ss]ystem$", 4, ",", 1],
[12, 2, "^Nu[0-9.]*$", 4, ",", 1]
]
},
"config": {
"fields": ["configid", "snmptrap_logging", "hk_history_global", "hk_history", "autoreg_tls_accept"],
"data": [
[1, 1, 0, "90d", 1]
]
},
"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
}
```

Solicitação de dados

A solicitação `proxy data` é enviada pelo proxy para fornecer disponibilidade de interface de host, histórico, descoberta e dados de autorregistro. Esta solicitação é enviada a cada `DataSenderFrequency` (parâmetro de configuração do proxy) segundos. Observe que o proxy ativo ainda irá consultar o Zabbix server a cada segundo para tarefas de comando remoto (com uma solicitação `proxy data` vazia).

nome	tipo de valor	descrição
proxy→server:		
request	*string*	'proxy data'
host	*string*	Nome do proxy.
session	*string*	Token de sessão de dados.
interface availability	*array*	*(opcional)* Array de objetos de dados de disponibilidade de interface.
interfaceid	*number*	Identificador da interface.
available	*number*	Disponibilidade da interface: **0**, *INTERFACE_AVAILABLE_UNKNOWN* - desconhecida **1**, *INTERFACE_AVAILABLE_TRUE* - disponível **2**, *INTERFACE_AVAILABLE_FALSE* - indisponível
error	*string*	Mensagem de erro da interface ou string vazia.
history data	*array*	*(opcional)* Array de objetos de dados históricos.
itemid	*number*	Identificador do item.
clock	*number*	Timestamp do valor do item (segundos).
ns	*number*	Timestamp do valor do item (nanossegundos).
value	*string*	*(opcional)* Valor do item.
id	*number*	Identificador do valor (contador ascendente, único dentro de uma sessão de dados).
timestamp	*number*	*(opcional)* Timestamp de itens do tipo log.
source	*string*	*(opcional)* Valor da fonte do item eventlog.
severity	*number*	*(opcional)* Valor de severidade do item eventlog.
eventid	*number*	*(opcional)* Valor eventid do item eventlog.
state	*string*	*(opcional)* Estado do item: **0**, *ITEM_STATE_NORMAL* **1**, *ITEM_STATE_NOTSUPPORTED*
lastlogsize	*number*	*(opcional)* Último tamanho do log de itens do tipo log.
mtime	*number*	*(opcional)* Tempo de modificação de itens do tipo log.
discovery data	*array*	*(opcional)* Array de objetos de dados de descoberta.
clock	*number*	Timestamp dos dados de descoberta.
druleid	*number*	Identificador da regra de descoberta.
dcheckid	*number*	Identificador da verificação de descoberta ou nulo para dados da regra de descoberta.

nome	tipo de valor	descrição
type	*number*	Tipo de verificação de descoberta: **-1** dados da regra de descoberta **0**, *SVC_SSH* - verificação de serviço SSH **1**, *SVC_LDAP* - verificação de serviço LDAP **2**, *SVC_SMTP* - verificação de serviço SMTP **3**, *SVC_FTP* - verificação de serviço FTP **4**, *SVC_HTTP* - verificação de serviço HTTP **5**, *SVC_POP* - verificação de serviço POP **6**, *SVC_NNTP* - verificação de serviço NNTP **7**, *SVC_IMAP* - verificação de serviço IMAP **8**, *SVC_TCP* - verificação de disponibilidade de porta TCP **9**, *SVC_AGENT* - Zabbix agent **10**, *SVC_SNMPv1* - SNMPv1 agent **11**, *SVC_SNMPv2* - SNMPv2 agent **12**, *SVC_ICMPPING* - ICMP ping **13**, *SVC_SNMPv3* - SNMPv3 agent **14**, *SVC_HTTPS* - verificação de serviço HTTPS **15**, *SVC_TELNET* - verificação de disponibilidade Telnet
ip	*string*	Endereço IP do host.
dns	*string*	Nome DNS do host.
port	*number*	*(opcional)* Número da porta do serviço.
key_	*string*	*(opcional)* Chave do item para verificação de descoberta do tipo **9** *SVC_AGENT*
value	*string*	*(opcional)* Valor recebido do serviço, pode estar vazio para a maioria dos serviços.
status	*number*	*(opcional)* Status do serviço: **0**, *DOBJECT_STATUS_UP* - Serviço ATIVO **1**, *DOBJECT_STATUS_DOWN* - Serviço INATIVO
autoregistration	*array*	*(opcional)* Array de objetos de dados de autorregistro.
clock	*number*	Timestamp dos dados de autorregistro.
host	*string*	Nome do host.
ip	*string*	*(opcional)* Endereço IP do host.
dns	*string*	*(opcional)* Nome DNS resolvido a partir do endereço IP.
port	*string*	*(opcional)* Porta do host.
host_metadata	*string*	*(opcional)* Metadados do host enviados pelo agent (com base no parâmetro de configuração do agent HostMetadata ou HostMetadataItem).
tasks	*array*	*(opcional)* Array de tarefas.
type	*number*	Tipo de tarefa: **0**, *ZBX_TM_TASK_PROCESS_REMOTE_COMMAND_RESULT* - resultado do comando remoto
status	*number*	Status da execução do comando remoto: **0**, *ZBX_TM_REMOTE_COMMAND_COMPLETED* - comando remoto concluído com sucesso **1**, *ZBX_TM_REMOTE_COMMAND_FAILED* - comando remoto falhou
error	*string*	*(opcional)* Mensagem de erro.
parent_taskid	*number*	ID da tarefa pai.
more	*number*	*(opcional)* 1 - há mais dados históricos para enviar
clock	*number*	*(opcional)* Timestamp da transferência de dados (segundos).
ns	*number*	*(opcional)* Timestamp da transferência de dados (nanossegundos).
version	*string*	Versão do proxy (<major>.<minor>.<build>).
server→proxy:		
response	*string*	Informação de sucesso da solicitação (’success’ ou ’failed’).

nome	tipo de valor	descrição
upload	*string*	Controle de upload para dados históricos (histórico, autorregistro, disponibilidade de host, descoberta de rede). Valores possíveis: **enabled** - operação normal **disabled** - o server não está aceitando dados (possivelmente devido ao limite interno do cache)
tasks	*array*	*(opcional)* Array de tarefas.
type	*number*	Tipo de tarefa: **1**, *ZBX_TM_TASK_PROCESS_REMOTE_COMMAND* - comando remoto
clock	*number*	Hora de criação da tarefa.
ttl	*number*	Tempo em segundos após o qual a tarefa expira.
commandtype	*number*	Tipo de comando remoto: **0**, *ZBX_SCRIPT_TYPE_CUSTOM_SCRIPT* - usar script personalizado **1**, *ZBX_SCRIPT_TYPE_IPMI* - usar IPMI **2**, *ZBX_SCRIPT_TYPE_SSH* - usar SSH **3**, *ZBX_SCRIPT_TYPE_TELNET* - usar Telnet **4**, *ZBX_SCRIPT_TYPE_GLOBAL_SCRIPT* - usar script global (atualmente funcionalmente equivalente ao script personalizado)
command	*string*	Comando remoto a ser executado.
execute_on	*number*	Alvo de execução para scripts personalizados: **0**, *ZBX_SCRIPT_EXECUTE_ON_AGENT* - executar script no agent **1**, *ZBX_SCRIPT_EXECUTE_ON_SERVER* - executar script no server **2**, *ZBX_SCRIPT_EXECUTE_ON_PROXY* - executar script no proxy
port	*number*	*(opcional)* Porta para comandos Telnet e SSH.
authtype	*number*	*(opcional)* Tipo de autenticação para comandos SSH.
username	*string*	*(opcional)* Nome de usuário para comandos Telnet e SSH.
password	*string*	*(opcional)* Senha para comandos Telnet e SSH.
publickey	*string*	*(opcional)* Chave pública para comandos SSH.
privatekey	*string*	*(opcional)* Chave privada para comandos SSH.
parent_taskid	*number*	ID da tarefa pai.
hostid	*number*	ID do host de destino.

Exemplo:

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": "7.0.0",
"clock": 1665730643,
"ns": 65389964
}
```


server→proxy:


```
{
"upload": "enabled",
"response": "success",
"tasks": [{
"type": 2,
"clock": 1665730986,
"ttl": 600,
"commandtype": 0,
"command": "ping -c 3 127.0.0.1; case $? in [01]) true;; *) false;; esac",
"execute_on": 2,
"port": 0,
"authtype": 0,
"username": "",
"password": "",
"publickey": "",
"privatekey": "",
"alertid": 0,
"parent_taskid": 4,
"hostid": 10084
```

```
}]
}
```

2 Protocolo do agent/agent2 do Zabbix

Consulte a página Verificações passivas e ativas do agent para obter mais informações sobre os protocolos do agent do Zabbix e do agent 2 do Zabbix.

4 Zabbix agent 2 plugin protocol

Zabbix agent 2 protocol is based on code, size and data model.

Código

Tipo	Tamanho	Comentários
Byte	4	Tipo de payload, atualmente apenas JSON é suportado.

Tamanho

Tipo	Tamanho	Comentários
Byte	4	Tamanho da carga útil atual em bytes.

Dados do payload

Tipo	Tamanho	Comentários
Byte	Definido pelo campo *Size*	Dados formatados em JSON.

Definição de dados do payload

Dados comuns

Esses parâmetros estão presentes em todas as solicitações/respostas:

Nome	Tipo	Comentários
id	uint32	Para solicitações - o identificador incremental usado para vincular solicitações com respostas. Único dentro de uma direção de solicitação (ou seja, do agent para o plugin ou do plugin para o agent). Para respostas - ID da solicitação correspondente.
type	uint32	O tipo de solicitação.

Solicitação de log

Uma solicitação enviada por um plugin para gravar uma mensagem de log no arquivo de log do agent.

direção	plugin → agent
resposta	não

Parâmetros específicos para solicitações de log:

Nome	Tipo	Comentários
severity	uint32	A gravidade da mensagem (nível de log).
message	string	A mensagem a ser registrada.

Exemplo:

```
{"id":0,"type":1,"severity":3,"message":"message"}
```

Solicitação de registro

Uma solicitação enviada pelo agent durante a fase de inicialização do agent para obter as métricas fornecidas para registrar um plugin.

direção	agent → plugin
resposta	sim

Parâmetros específicos para solicitações de registro:

Nome	Tipo	Comentários
version	string	A versão do protocolo <major>.<minor>

Exemplo:

```
{"id":1,"type":2,"version":"1.0"}
```

Resposta de registro

Resposta do plugin à solicitação de registro.

direction	plugin → agent
response	n/a

Parâmetros específicos para respostas de registro:

Nome	Tipo	Comentários
name	string	O nome do plugin.
metrics	array de strings (opcional)	As métricas com descrições conforme usadas no plugin. Retorna RegisterMetrics(). Ausente se um erro for retornado.
interfaces	uint32 (opcional)	A máscara de bits das interfaces suportadas pelo plugin. Ausente se um erro for retornado.
error	string (opcional)	Uma mensagem de erro retornada se um plugin não puder ser iniciado. Ausente se as métricas forem retornadas.

Exemplos:

```
{"id":2,"type":3,"metrics":["external.test", "External exporter Test."], "interfaces": 4}
```

ou

```
{"id":2,"type":3,"error":"mensagem de erro"}
```

Solicitação de início

Uma solicitação para executar a função Start da interface Runner.

direction	agent → plugin
response	não

A solicitação não possui parâmetros específicos, apenas contém os parâmetros de dados comuns.

Exemplo:

```
{"id":3,"type":4}
```

Encerrar solicitação

Uma solicitação enviada pelo agent para desligar um plugin.

direction	agent → plugin
response	não

A solicitação não possui parâmetros específicos, ela contém apenas os parâmetros de dados comuns.

Exemplo:

```
{"id":3,"type":5}
```

Solicitação de exportação

Uma solicitação para executar a função Export da interface Exporter.

direction	agent → plugin
response	não

Parâmetros específicos para solicitações de exportação:

Nome	Tipo	Comentários
key	string	A chave do plugin.
parameters	array de strings (opcional)	Os parâmetros para a função Export.

Exemplo:

```
{"id":4,"type":6,"key":"test.key","parameters":["foo","bar"]}
```

Resposta de exportação

Resposta da função Export da interface Exporter.

direction	plugin → agent
response	n/a

Parâmetros específicos para respostas de exportação:

Nome	Tipo	Comentários
value	string (opcional)	Valor de resposta da função Export. Ausente, se um erro for retornado.
error	string (opcional)	Mensagem de erro se a função Export não foi executada com sucesso. Ausente, se o valor for retornado.

Exemplos:

```
{"id":5,"type":7,"value":"response"}
```

ou

```
{"id":5,"type":7,"error":"mensagem de erro"}
```

Configurar solicitação

Uma solicitação para executar a função *Configure* da interface *Configurator*.

direction	agent → plugin
response	n/a

Parâmetros específicos para solicitações *Configure*:

Nome	Tipo	Comentários
global_options	JSON object	Objeto JSON contendo opções globais de configuração do agent.
private_options	JSON object (opcional)	Objeto JSON contendo opções privadas de configuração do plugin, se fornecido.

Exemplo:

```
{"id":6,"type":8,"global_options":{...},"private_options":{...}}
```

Validar solicitação

Uma solicitação para executar a função *Validate* da interface *Configurator*.

direção	agent → plugin
resposta	sim

Parâmetros específicos para solicitações *Validate*:

Nome	Tipo	Comentários
private_options	JSON object (opcional)	Objeto JSON contendo opções de configuração privadas do plugin, se fornecido.

Exemplo:

```
{"id":7,"type":9,"private_options":{...}}
```

Validar resposta

Resposta da função *Validate* da interface *Configurator*.

direction	plugin → agent
response	n/a

Parâmetros específicos para respostas *Validate*:

Nome	Tipo	Comentários
error	string (opcional)	Uma mensagem de erro retornada se a função Validate não for executada com sucesso. Ausente se executada com sucesso.

Exemplo:

```
{"id":8,"type":10}
```

ou

```
{"id":8,"type":10,"error":"mensagem de erro"}
```

5 Protocolo do Zabbix sender

Visão geral

O Zabbix server e o Zabbix proxy usam um protocolo de comunicação baseado em JSON para receber dados do Zabbix sender. Os dados podem ser recebidos com a ajuda de um item trapper ou de um item HTTP agent com trapping habilitado.

As mensagens de solicitação e resposta devem começar com header e comprimento dos dados.

Requisição do Zabbix sender

```
{
    "request": "sender data",
    "data": [
        {
            "host": "<hostname>",
            "key": "trap",
            "value": "test value"
        }
    ]
}
```

Resposta do Zabbix server

```
{
    "response": "success",
    "info": "processed: 1; failed: 0; total: 1; seconds spent: 0.060753"
}
```

Requisição do Zabbix sender com timestamp

Alternativamente, o Zabbix sender pode enviar uma requisição com timestamp e nanossegundos.

```
{
    "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
}
```

Resposta do Zabbix server

```
{
    "response": "success",
    "info": "processed: 2; failed: 0; total: 2; seconds spent: 0.060904"
}
```

6 Cabeçalho

Visão geral

O cabeçalho está presente em todas as mensagens de solicitação e resposta entre os componentes do Zabbix. Ele é necessário para determinar o comprimento da mensagem, se ela está compactada ou não, se é um pacote grande ou não.

O protocolo de comunicação do Zabbix possui um limite de tamanho de pacote de 1GB por conexão. O limite de 1GB é aplicado tanto ao comprimento dos dados do pacote recebido quanto ao comprimento dos dados descompactados.

Ao enviar a configuração para o proxy do Zabbix, o limite de tamanho do pacote é aumentado para 4GB para permitir a sincronização de grandes configurações. Quando o comprimento dos dados antes da compactação excede 4GB, o server do Zabbix automaticamente começa a usar o formato de pacote grande (flag 0x04), o que aumenta o limite de tamanho do pacote para 16GB.

Observe que, embora um formato de pacote grande possa ser usado para enviar quaisquer dados, atualmente apenas o sincronizador de configuração do proxy do Zabbix pode lidar com pacotes maiores que 1GB.

Estrutura

O cabeçalho consiste em quatro campos. Todos os números no cabeçalho são formatados como little-endian.

Campo	Tamanho	Tamanho (pacote grande)	Descrição
<PROTOCOL>	4	4	"ZBXD" ou 5A 42 58 44
<FLAGS>	1	1	Flags do protocolo: 0x01 - protocolo de comunicação Zabbix 0x02 - compressão 0x04 - pacote grande
<DATALEN>	4	8	Tamanho dos dados.
<RESERVED>	4	8	Quando a compressão é usada (flag 0x02) - o tamanho dos dados descompactados Quando a compressão não é usada - 00 00 00 00

Exemplos

Aqui estão alguns trechos de código mostrando como adicionar o cabeçalho do protocolo Zabbix aos dados que você deseja enviar para obter o pacote que deve ser enviado ao Zabbix para que seja interpretado corretamente. Esses trechos de código assumem que os dados não são maiores que 1GB, portanto, o formato de pacote grande não é usado.

Python

```
packet = b"ZBXD\1" + struct.pack("<II", len(data), 0) + data
```

ou

```
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;
```

ou

```
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;
```

ou

```
function zbx_create_header($plain_data_size, $compressed_data_size = null)
{
    $protocol = "ZBXD";
    $flags = 0x01;
    if (is_null($compressed_data_size))
    {
        $datalen = $plain_data_size;
        $reserved = 0;
    }
    else
    {
        $flags |= 0x02;
        $datalen = $compressed_data_size;
        $reserved = $plain_data_size;
    }
    return $protocol . chr($flags) . pack("VV", $datalen, $reserved);
}

$packet = zbx_create_header(strlen($data)) . $data;
```

Bash

```
datalen=$(printf "%08x" ${#data})
datalen="\\x${datalen:6:2}\\x${datalen:4:2}\\x${datalen:2:2}\\x${datalen:0:2}"
printf "ZBXD\1${datalen}\0\0\0\0%s" "$data"
```

7 Protocolo de exportação JSON delimitado por nova linha

Esta seção apresenta detalhes do protocolo de exportação no formato JSON delimitado por nova linha, usado em:

- exportação de dados para arquivos
- streaming para sistemas externos

Podem ser exportados:

- eventos de trigger
- valores de item
- tendências (apenas exportação para arquivos)

Todos os arquivos têm a extensão .ndjson. Cada linha do arquivo de exportação é um objeto JSON.

Eventos de trigger

As seguintes informações são exportadas para um evento de problema:

Campo		Tipo	Descrição
clock		number	Número de segundos desde a Época até o momento em que o problema foi detectado (parte inteira).
ns		number	Número de nanossegundos a serem adicionados ao `clock` para obter o tempo preciso de detecção do problema.
value		number	1 (sempre).
eventid		number	ID do evento de problema.
name		string	Nome do evento de problema.
severity		number	Severidade do evento de problema (0 - Não classificado, 1 - Informação, 2 - Aviso, 3 - Médio, 4 - Alto, 5 - Desastre).
hosts		array	Lista de hosts envolvidos na expressão do trigger; deve haver pelo menos um elemento no array.
-		object	
	host	string	Nome do host.
	name	string	Nome visível do host.
groups		array	Lista de grupos de hosts de todos os hosts envolvidos na expressão do trigger; deve haver pelo menos um elemento no array.
-		string	Nome do grupo de hosts.
tags		array	Lista de tags de problema (pode estar vazia).
-		object	
	tag	string	Nome da tag.
	value	string	Valor da tag (pode estar vazio).

As seguintes informações são exportadas para um evento de recuperação:

Campo	Tipo	Descrição
clock	number	Número de segundos desde a Época até o momento em que o problema foi resolvido (parte inteira).
ns	number	Número de nanossegundos a serem adicionados ao `clock` para obter o tempo preciso de resolução do problema.
value	number	0 (sempre).
eventid	number	ID do evento de recuperação.
p_eventid	number	ID do evento de problema.

Exemplos

Problema:

```
{"clock":1519304285,"ns":123456789,"value":1,"name":"O agente Zabbix está inacessível no Host B ou os pesq
```

Recuperação:

```
{"clock":1519304345,"ns":987654321,"value":0,"eventid":43,"p_eventid":42}
```

Problema (geração de eventos de problemas múltiplos):

```
{"clock":1519304286,"ns":123456789,"value":1,"eventid":43,"name":"O agente Zabbix está inacessível no Host
```

```
{"clock":1519304286,"ns":123456789,"value":1,"eventid":43,"name":"O agente Zabbix está inacessível no Host
```

Recuperação:

```
{"clock":1519304346,"ns":987654321,"value":0,"eventid":44,"p_eventid":43}
```

```
{"clock":1519304346,"ns":987654321,"value":0,"eventid":44,"p_eventid":42}
```

Valores de item

As seguintes informações são exportadas para um valor de item coletado:

Campo		Tipo	Descrição
host		objeto	Nome do host do item.
	host	string	Nome do host.

Campo	Tipo	Descrição
name	string	Nome visível do host.
groups	array	Lista de grupos de hosts do host do item; deve haver pelo menos um elemento no array.
-	string	Nome do grupo de hosts.
item_tags	array	Lista de tags do item (pode estar vazia).
-	objeto	
tag	string	Nome da tag.
value	string	Valor da tag (pode estar vazio).
itemid	número	ID do item.
name	string	Nome visível do item.
clock	número	Número de segundos desde a Época até o momento em que o valor foi coletado (parte inteira).
ns	número	Número de nanossegundos a serem adicionados ao `clock` para obter o tempo preciso de coleta do valor.
timestamp (*Log* apenas)	número	0 se não disponível.
source (*Log* apenas)	string	String vazia se não disponível.
severity (*Log* apenas)	número	0 se não disponível.
eventid (*Log* apenas)	número	0 se não disponível.
value	número (para itens numéricos) ou string (para itens de texto)	Valor do item coletado.
type	número	Tipo de valor coletado: 0 - numérico float, 1 - caractere, 2 - log, 3 - numérico sem sinal, 4 - texto, 5 - binário

Exemplos

Valor numérico (sem sinal):


```
{"host":{"host":"Host B","name":"Host B visible"},"groups":["Group X","Group Y","Group Z"],"item_tags":[{"
```


Valor numérico (float):


```
{"host":{"host":"Host B","name":"Host B visible"},"groups":["Group X","Group Y","Group Z"],"item_tags":[{"
```


Valor de caractere, texto:


```
{"host":{"host":"Host B","name":"Host B visible"},"groups":["Group X","Group Y","Group Z"],"item_tags":[{"
```


Valor de log:


```
{"host":{"host":"Host A","name":"Host A visible"},"groups":["Group X","Group Y","Group Z"],"item_tags":[{"
```


Tendências

As seguintes informações são exportadas para um valor de tendência calculado:

Campo	Tipo	Descrição
host	objeto	Nome do host do item.
host	string	Nome do host.
name	string	Nome visível do host.

Campo	Tipo	Descrição
groups	array	Lista de grupos de hosts do host do item; deve haver pelo menos um elemento no array.
-	string	Nome do grupo de hosts.
item_tags	array	Lista de tags do item (pode estar vazia).
-	objeto	
tag	string	Nome da tag.
value	string	Valor da tag (pode estar vazio).
itemid	number	ID do item.
name	string	Nome visível do item.
clock	number	Número de segundos desde a Época até o momento em que o valor foi coletado (parte inteira).
count	number	Número de valores coletados para uma determinada hora.
min	number	Valor mínimo do item para uma determinada hora.
avg	number	Valor médio do item para uma determinada hora.
max	number	Valor máximo do item para uma determinada hora.
type	number	Tipo de valor: 0 - numérico float, 3 - numérico sem sinal

Exemplos

Valor numérico (sem sinal):


```
{"host":{"host":"Host B","name":"Host B visible"},"groups":["Group X","Group Y","Group Z"],"item_tags":[{"
```


Valor numérico (float):


```
{"host":{"host":"Host B","name":"Host B visible"},"groups":["Group X","Group Y","Group Z"],"item_tags":[{"
```


4 Items

Use a barra lateral para acessar o conteúdo desta seção.

1 parâmetros vm.memory.size

Visão geral

Esta seção fornece alguns detalhes dos parâmetros para o item agent vm.memory.size[<mode>].

Parâmetros

Os seguintes parâmetros estão disponíveis para este item:

- **active** - memória atualmente em uso ou usada recentemente, portanto está na RAM
- **anon** - memória não associada a um arquivo (não pode ser relida a partir dele)
- **available** - memória disponível, calculada de forma diferente dependendo da plataforma (veja a tabela abaixo)
- **buffers** - cache para coisas como metadados do sistema de arquivos
- **cached** - cache para várias coisas
- **exec** - código executável, normalmente de um arquivo (programa)
- **file** - cache para conteúdos de arquivos acessados recentemente
- **free** - memória que está prontamente disponível para qualquer entidade que solicitar memória
- **inactive** - memória marcada como não utilizada
- **pavailable** - memória 'disponível' como porcentagem do 'total' (calculada como `available/total*100`)
- **pinned** - igual a 'wired'
- **pused** - memória 'usada' como porcentagem do 'total' (calculada como `used/total*100`)
- **shared** - memória que pode ser acessada simultaneamente por vários processos
- **slab** - quantidade total de memória usada pelo kernel para armazenar estruturas de dados para seu próprio uso
- **total** - memória física total disponível
- **used** - memória usada, calculada de forma diferente dependendo da plataforma (veja a tabela abaixo)
- **wired** - memória marcada para sempre permanecer na RAM. Nunca é movida para o disco.

Warning:
Alguns desses parâmetros são específicos de plataforma e podem não estar disponíveis na sua plataforma. Veja itens do agent Zabbix para detalhes.

Cálculo específico de plataforma de **available** e **used**:

Plataforma	**"available"**	**"used"**
AIX	free + cached	memória real em uso
FreeBSD	inactive + cached + free	active + wired + cached
HP UX	free	total - free
Linux<3.14	free + buffers + cached	total - free
Linux 3.14+ (também retroportado para 3.10 no RHEL 7)	/proc/meminfo, veja "MemAvailable" na documentação do kernel Linux para detalhes. Observe que free + buffers + cached não é mais igual a 'available' devido ao fato de nem todo o cache de páginas poder ser liberado e ao uso do low watermark no cálculo.	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:
A soma de *vm.memory.size[used]* e *vm.memory.size[available]* não é necessariamente igual ao total. Por exemplo, no FreeBSD:
 * Memórias ativas, inativas, wired e cached são consideradas usadas, porque armazenam algumas informações úteis.
 * Ao mesmo tempo, memórias inativas, cached e livres são consideradas disponíveis, porque esses tipos de memórias podem ser dadas instantaneamente a processos que solicitam mais memória.
Portanto, a memória inativa é usada e disponível simultaneamente. Por causa disso, o item *vm.memory.size[used]* é projetado apenas para fins informativos, enquanto *vm.memory.size[available]* é projetado para ser usado em triggers.

Veja também

1. Detalhes adicionais sobre cálculo de memória em diferentes SO

2 Passive and active agent checks

Visão geral

Esta seção fornece detalhes sobre as verificações passivas e ativas realizadas pelo agent Zabbix e pelo agent Zabbix 2.

O Zabbix usa um protocolo de comunicação baseado em JSON para se comunicar com os agents.

Os protocolos do agent Zabbix e do agent Zabbix 2 foram unificados desde o Zabbix 7.0. A diferença entre as solicitações/respostas do agent Zabbix e do agent Zabbix 2 é expressa pelo valor da tag "variant".

Checagens passivas

Uma checagem passiva é uma simples solicitação de dados. O server ou proxy do Zabbix solicita alguns dados (por exemplo, carga da CPU) e o agent do Zabbix envia o resultado de volta ao server.

As checagens passivas são executadas de forma assíncrona - não é necessário receber a resposta de uma solicitação antes que outras checagens sejam iniciadas. A resolução de DNS também é assíncrona.

O agent poller tentará se conectar a todos os endereços retornados pela consulta DNS. Isso garante que, se um endereço IP estiver inacessível, o poller tentará o próximo endereço disponível, aumentando a probabilidade de uma conexão bem-sucedida. Essa melhoria se aplica tanto ao server quanto ao proxy do Zabbix.

A concorrência máxima de checagens assíncronas é 1000 (definida por MaxConcurrentChecksPerPoller).

O número de agent pollers assíncronos é definido pelo parâmetro StartAgentPollers.

Solicitação do server

Para definição do cabeçalho e comprimento dos dados, consulte detalhes do protocolo.

```
{
  "request": "passive checks",
  "data": [
    {
      "key": "agent.version",
      "timeout": 3
    }
  ]
}
```

Campo		Tipo	Obrigatório	Valor
request		*string*	sim	"passive checks"
data		*array de objeto*	sim	Item de checagem passiva.
	key	*string*	sim	Chave do item com macros expandidas.
	timeout	*number*	sim	Tempo limite de comunicação.

Resposta do agent

```
{
  "version": "7.0.0",
  "variant": 2,
  "data": [
    {
      "value": "7.0.0"
    }
  ]
}
```

Campo		Tipo	Obrigatório	Valor
version		*string*	sim	O número da versão do agent.
variant		*number*	sim	A variante do agent (*1* - Zabbix agent, *2* - Zabbix agent 2).
data		*array de objeto*	sim	Contém o resultado da checagem.
	value	*string*	não	O valor do item se a checagem foi bem-sucedida.
	error	*string*	não	A mensagem de erro se a checagem não foi bem-sucedida.

Por exemplo, para itens suportados:

1. O server abre uma conexão TCP
2. O server envia **<HEADER><DATALEN>{"request":"passive checks","data":[{"key":"agent.ping","timeout":3}]}**
3. O agent lê a solicitação e responde com **<HEADER><DATALEN>{"version":"7.0.0","variant":2,"data":[{"value":1}]}**
4. O server processa os dados para obter o valor, '1' neste caso
5. A conexão TCP é fechada

Para itens não suportados:

1. O server abre uma conexão TCP
2. O server envia **<HEADER><DATALEN>{"request":"passive checks","data":[{"key":"vfs.fs.size[/nono]","timeout":3}]}**
3. O agent lê a solicitação e responde com **<HEADER><DATALEN>{"version":"7.0.0","variant":2,"data":[{"error":"Unsupport item key."}]}**
4. O server processa os dados, altera o estado do item para não suportado com a mensagem de erro especificada
5. A conexão TCP é fechada

Failover para o protocolo antigo

Para garantir que o Zabbix server ou proxy possa trabalhar com agents de versões anteriores à 7.0, que utilizam o protocolo em texto simples, foi implementado um failover para o protocolo antigo.

As verificações passivas são realizadas usando o protocolo JSON (7.0 e posteriores) após a reinicialização ou quando a configuração da interface é alterada. Se nenhuma resposta JSON válida for recebida (o agent enviou "ZBX_NOTSUPPORTED"), o Zabbix irá armazenar em cache a interface como protocolo antigo e **tentar novamente** a verificação enviando apenas a chave do item.

Observe que a cada hora o Zabbix server/proxy tentará novamente trabalhar com o novo protocolo em todas as interfaces, retornando ao protocolo antigo se necessário.

Checagens ativas

As checagens ativas requerem um processamento mais complexo. O agent deve primeiro recuperar do server/proxy uma lista de items e/ou comandos remotos para processamento independente.

Os servers/proxies dos quais obter as checagens ativas são listados no parâmetro 'ServerActive' do arquivo de configuração do agent. A frequência de solicitação dessas checagens é definida pelo parâmetro 'RefreshActiveChecks' no mesmo arquivo de configuração. No entanto, se a atualização das checagens ativas falhar, ela será tentada novamente após 60 segundos fixos.

Note:
Desde o Zabbix 6.4, o agent (em modo ativo) não recebe mais do server/proxy uma cópia completa da configuração a cada dois minutos (padrão). Em vez disso, para diminuir o tráfego de rede e o uso de recursos, uma sincronização incremental da configuração é realizada a cada 5 segundos (padrão), na qual o server/proxy fornece uma cópia completa da configuração **apenas** se o agent ainda não a recebeu, ou se algo mudou na configuração do host, macros globais ou expressões regulares globais.

O agent então periodicamente envia os novos valores para o(s) server(s). Se o agent recebeu algum comando remoto para executar, o resultado da execução também será enviado. Observe que a execução de comandos remotos em um agent ativo é suportada desde o Zabbix agent 7.0.

Note:
Se um agent estiver atrás do firewall, você pode considerar usar apenas verificações ativas porque, nesse caso, não seria necessário modificar o firewall para permitir conexões de entrada iniciais.

Obtendo a lista de items

Requisição do agent

A requisição de verificações ativas é usada para obter as verificações ativas a serem processadas pelo agent. Esta requisição é enviada pelo agent ao iniciar e, em seguida, em intervalos definidos por RefreshActiveChecks.

```
{
  "request": "active checks",
  "host": "Zabbix server",
  "host_metadata": "mysql,nginx",
  "interface": "zabbix.server.lan",
  "ip": "159.168.1.1",
  "port": 12050,
  "version": "7.0.0",
  "variant": 2,
  "config_revision": 1,
  "session": "e3dcbd9ace2c9694e1d7bbd030eeef6e"
}
```

Campo	Tipo	Obrigatório	Valor
request	*string*	sim	`active checks`
host	*string*	sim	Nome do host.
host_metadata	*string*	não	O parâmetro de configuração HostMetadata ou o valor da métrica HostMetadataItem.
interface	*string*	não	O parâmetro de configuração HostInterface ou o valor da métrica HostInterfaceItem.
ip	*string*	não	O primeiro IP do parâmetro de configuração ListenIP, se definido.
port	*number*	não	O valor do parâmetro de configuração ListenPort, se definido e não for a porta padrão de escuta do agent.
version	*string*	sim	O número da versão do agent.
variant	*number*	sim	A variante do agent (*1* - Zabbix agent, *2* - Zabbix agent 2).
config_revision	*number*	não	Identificador de configuração para sincronização incremental de configuração.
session	*string*	não	Identificador de sessão para sincronização incremental de configuração.

Resposta do server

A resposta de verificações ativas é enviada pelo server de volta ao agent após processar a requisição de verificações ativas.

```
{
  "response": "success",
  "config_revision": 2,
  "data": [
    {
      "key": "system.uptime",
      "itemid": 1234,
      "delay": "10s",
      "lastlogsize": 0,
      "mtime": 0
    },
    {
      "key": "agent.version",
      "itemid": 5678,
      "delay": "10m",
      "lastlogsize": 0,
      "mtime": 0,
      "timeout": "30s"
    }
  ],
  "commands": [
    {
      "command": "df -h --output=source,size / | awk 'NR>1 {print $2}'",
      "id": 1324,
      "wait": 1
    }
  ]
}
```


Campo	Tipo	Obrigatório	Valor
response	*string*	sim	`success` | `failed`
info	*string*	não	Informação de erro em caso de falha.
data	*array of objects*	não	Items de verificação ativa. Omitido se a configuração do host não for alterada.
key	*string*	não	Chave do item com macros expandidas.
itemid	*number*	não	Identificador do item.
delay	*string*	não	Intervalo de atualização do item. Intervalos flexíveis/agendados são suportados tanto pelo Zabbix agent quanto pelo Zabbix agent 2 desde o Zabbix 7.0.
lastlogsize	*number*	não	lastlogsize do item.
mtime	*number*	não	mtime do item.
timeout	*string*	não	timeout do item.
refresh_unsupported	*number*	não	Intervalo de atualização de item não suportado.
regexp	*array of objects*	não	Expressões regulares globais.
name	*string*	não	Nome da expressão regular global.
expression	*string*	não	Expressão regular global.
expression_type	*number*	não	Tipo da expressão regular global.
exp_delimiter	*string*	não	Delimitador da expressão regular global.
case_sensitive	*number*	não	Configuração de sensibilidade a maiúsculas/minúsculas da expressão regular global.
commands	*array of objects*	não	Comandos remotos a serem executados. Incluído se a execução de comando remoto foi acionada por uma operação de ação ou execução manual de script. Observe que a execução de comando remoto em um agent ativo é suportada desde o Zabbix agent 7.0. Agents ativos mais antigos ignorarão quaisquer comandos remotos incluídos na resposta do server de verificações ativas.
command	*string*	não	Comando remoto.
id	*number*	não	Identificador do comando remoto.

Campo	Tipo	Obrigatório	Valor
wait	*number*	não	Modo de execução do comando remoto (”0” (nowait) para comandos de operações de ação; ”1” (wait) para comandos de execução manual de script).
config_revision	*number*	não	Identificador de configuração para sincronização incremental de configuração. Omitido se a configuração do host não for alterada. Incrementado se a configuração do host for alterada.

O server deve responder com sucesso.

Por exemplo:

1. O agent abre uma conexão TCP
2. O agent solicita a lista de verificações
3. O server responde com uma lista de items e comandos remotos a serem executados
4. O agent analisa a resposta
5. A conexão TCP é fechada
6. O agent inicia a coleta periódica de dados e executa comandos remotos (suportado desde o Zabbix agent 7.0)

Attention:
Observe que dados de configuração (sensíveis) podem ficar disponíveis para partes que tenham acesso à porta trapper do Zabbix server ao usar uma verificação ativa. Isso é possível porque qualquer pessoa pode fingir ser um agent ativo e solicitar dados de configuração de item; a autenticação não ocorre a menos que você use as opções de criptografia.

Enviando dados coletados

Agent envia

A solicitação de dados do agent contém os valores dos items coletados e os valores dos comandos remotos executados (se houver).

```
{
  "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": "7.0.0",
  "variant": 2
}
```

Campo	Tipo	Obrigatório	Valor
request	*string*	sim	`agent data`

Campo	Tipo	Obrigatório	Valor
data	*array de objetos*	sim	Valores dos items.
id	*number*	sim	O identificador do valor (contador incremental usado para verificar valores duplicados no caso de problemas de rede).
itemid	*number*	sim	O identificador do item.
value	*string*	não	O valor do item.
lastlogsize	*number*	não	O lastlogsize do item.
mtime	*number*	não	O mtime do item.
state	*number*	não	O state do item.
source	*string*	não	O source do log de eventos do valor.
eventid	*number*	não	O eventid do log de eventos do valor.
severity	*number*	não	O severity do log de eventos do valor.
timestamp	*number*	não	O timestamp do log de eventos do valor.
clock	*number*	sim	O timestamp do valor (segundos desde Epoch).
ns	*number*	sim	O timestamp do valor em nanossegundos.
commands	*array de objetos*	não	Resultado da execução de comandos remotos. Observe que a execução de comandos remotos em um agent ativo é suportada desde o Zabbix agent 7.0. Agents ativos mais antigos ignorarão quaisquer comandos remotos incluídos na resposta do servidor de verificações ativas.
id	*number*	não	Identificador do comando remoto.
value	*string*	não	Resultado da execução do comando remoto se a execução foi bem-sucedida.
error	*string*	não	Mensagem de erro da execução do comando remoto se a execução falhou.
session	*string*	sim	Identificador de sessão exclusivo gerado cada vez que o agent é iniciado.
host	*string*	sim	Nome do host.
version	*string*	sim	O número da versão do agent.
variant	*number*	sim	A variante do agent (*1* - Zabbix agent, *2* - Zabbix agent 2).

Um ID virtual é atribuído a cada valor. O ID do valor é um contador ascendente simples, exclusivo dentro de uma sessão de dados (identificada pelo token de sessão). Esse ID é usado para descartar valores duplicados que podem ser enviados em ambientes com conectividade ruim.

Resposta do servidor

A resposta de dados do agent é enviada pelo servidor de volta ao agent após o processamento da solicitação de dados do agent.

```
{
  "response": "success",
  "info": "processed: 2; failed: 0; total: 2; seconds spent: 0.003534"
}
```

Campo	Tipo	Obrigatório	Valor
response	*string*	sim	`success` | `failed`
info	*string*	sim	Resultados do processamento dos items.

Attention:
Se o envio de alguns valores falhar no servidor (por exemplo, porque o host ou o item foi desabilitado ou excluído), o agent não tentará reenviar esses valores.

Por exemplo:

1. O agent abre uma conexão TCP
2. O agent envia uma lista de valores
3. O servidor processa os dados e envia o status de volta
4. A conexão TCP é fechada

Attention:
A mensagem de erro será limitada a 2048 símbolos no lado do servidor.

Mensagem de heartbeat

Agent envia

A mensagem de heartbeat é enviada por um agent ativo para o Zabbix server/proxy a cada HeartbeatFrequency segundos (configurado no arquivo de configuração do Zabbix agent/ agent 2).

Ela é usada para monitorar a disponibilidade de checagens ativas.

```
{
  "request": "active check heartbeat",
  "host": "Zabbix server",
  "heartbeat_freq": 60,
  "version": "7.0.0",
  "variant": 2
}
```

Campo	Tipo	Obrigatório	Valor
request	*string*	sim	`active check heartbeat`
host	*string*	sim	O nome do host.
heartbeat_freq	*number*	sim	A frequência de heartbeat do agent (parâmetro de configuração HeartbeatFrequency).
version	*string*	sim	O número da versão do agent.
variant	*number*	sim	A variante do agent (*1* - Zabbix agent, *2* - Zabbix agent 2).

Resposta de redirecionamento

Quando um host foi reatribuído, o server pode instruir o agent a redirecionar seu heartbeat (e as checagens ativas subsequentes) para outro proxy ou instância de server.

```
  {
    "response": "failed",
    "redirect": {
      "revision": 2,
      "address": "192.0.2.0:10055"
    }
  }
```

Campo		Tipo	Obrigatório	Valor
response		*string*	sim	`success` | `failed`
redirect		*object*	sim	Instruções de redirecionamento.
	revision	*number*	sim	Identificador da revisão da configuração.
	address	*string*	sim	Endereço do server/proxy de destino.

Protocolo XML antigo

Note:
O Zabbix irá tratar até 16 MB de dados XML codificados em Base64, mas um único valor decodificado não poderá ser superior a 64 KB ou será truncado para 64 KB durante a decodificação.

Veja também

1. Mais detalhes sobre o protocolo do Zabbix Agent (inglês)

3 Nível mínimo de permissão para itens do agent do Windows

Visão geral

Ao monitorar sistemas usando um agente, uma boa prática é obter métricas do host no qual o agente está instalado. Para usar o princípio do privilégio mínimo, é necessário determinar quais métricas são obtidos do agente.

A tabela neste documento permite que você selecione os direitos mínimos para garantia de operação correta do agente Zabbix.

Se um usuário diferente for selecionado para o agente trabalhar, em vez de 'LocalSystem', então para a operação do agente como um serviço do Windows, o novo usuário deve ter os direitos "Logon como um serviço" de "Local Política → Cessão de Direitos

do Usuário" e o direito de criar, escrever e excluir o arquivo de log do agente Zabbix. Um usuário do Active Directory deve ser adicionado ao Grupo *Usuários do Monitor de Desempenho*.

Note:
Ao trabalhar com os direitos de um agente com base na grupo "mínimo tecnicamente aceitável", provisão prévia de direitos de objetos para monitoramento é necessário.

Itens comuns do agent suportados no Windows

Chave do item	Grupo de usuários	
	Recomendado	Mínimo tecnicamente aceitável (funcionalidade limitada)
agent.hostname	Convidados	Convidados
agent.ping	Convidados	Convidados
agent.variant	Convidados	Convidados
agent.version	Convidados	Convidados
log	Administradores	Convidados
log.count	Administradores	Convidados
logrt	Administradores	Convidados
logrt.count	Administradores	Convidados
net.dns	Convidados	Convidados
net.dns.perf	Convidados	Convidados
net.dns.record	Convidados	Convidados
net.if.discovery	Convidados	Convidados
net.if.in	Convidados	Convidados
net.if.out	Convidados	Convidados
net.if.total	Convidados	Convidados
net.tcp.listen	Convidados	Convidados
net.tcp.port	Convidados	Convidados
net.tcp.service	Convidados	Convidados
net.tcp.service.perf	Convidados	Convidados
net.udp.service	Convidados	Convidados
net.udp.service.perf	Convidados	Convidados
proc.num	Administradores	Convidados
system.cpu.discovery	Usuários do Monitor de Desempenho	Usuários do Monitor de Desempenho
system.cpu.load	Usuários do Monitor de Desempenho	Usuários do Monitor de Desempenho
system.cpu.num	Convidados	Convidados

Chave do item	Grupo de usuários	
system.cpu.util	Usuários do Monitor de Desempenho	Usuários do Monitor de Desempenho
system.hostname	Convidados	Convidados
system.localtime	Convidados	Convidados
system.run	Administradores	Convidados
system.sw.arch	Convidados	Convidados
system.swap.size	Convidados	Convidados
system.uname	Convidados	Convidados
system.uptime	Usuários do Monitor de Desempenho	Usuários do Monitor de Desempenho
vfs.dir.count	Administradores	Convidados
vfs.dir.get	Administradores	Convidados
vfs.dir.size	Administradores	Convidados
vfs.file.cksum	Administradores	Convidados
vfs.file.contents	Administradores	Convidados
vfs.file.exists	Administradores	Convidados
vfs.file.md5sum	Administradores	Convidados
vfs.file.regexp	Administradores	Convidados
vfs.file.regmatch	Administradores	Convidados
vfs.file.size	Administradores	Convidados
vfs.file.time	Administradores	Convidados
vfs.fs.discovery	Administradores	Convidados
vfs.fs.get	Administradores	Convidados
vfs.fs.size	Administradores	Convidados
vm.memory.size	Convidados	Convidados
web.page.get	Convidados	Convidados
web.page.perf	Convidados	Convidados
web.page.regexp	Convidados	Convidados
zabbix.stats	Convidados	Convidados

Teclas de itens específicas do Windows

Chave do item	Grupo de usuários	
	Recomendado	Mínimo tecnicamente aceitável (funcionalidade limitada)
eventlog	Leitores de log de eventos	Convidados
net.if.list	Convidados	Convidados

Chave do item	Grupo de usuários	
perf_counter	Usuários do Monitor de Desempenho	Usuários do Monitor de Desempenho
proc_info	Administradores	Convidados
service.discovery	Convidados	Convidados
service.info	Convidados	Convidados
serviços	Hóspedes	Hóspedes
wmi.get	Administradores	Convidados
vm.vmemory.size	Convidados	Convidados

4 Codificação dos valores retornados

O servidor Zabbix espera que todo valor de texto retornado esteja na codificação UTF8. Isso se aplica a qualquer tipo de verificação: agent Zabbix, SSH, Telnet, etc.

Diferentes sistemas/dispositivos monitorados e verificações podem retornar caracteres não-ASCII no valor. Para esses casos, quase todas as possíveis chaves **zabbix** contêm um parâmetro adicional de chave de item - **<encoding>**. Este parâmetro de chave é opcional, mas deve ser especificado se o valor retornado não estiver na codificação UTF8 e contiver caracteres não-ASCII. Caso contrário, o resultado pode ser inesperado e imprevisível.

A seguir, uma descrição do comportamento com diferentes backends de banco de dados nesses casos.

MySQL

Se um valor contiver um caractere não ASCII em uma codificação diferente de UTF8 - este caractere e o seguinte serão descartados quando o banco de dados armazenar este valor. Nenhuma mensagem de aviso será gravada no *zabbix_server.log*.
 Relevante para pelo menos a versão 5.1.61 do MySQL.

PostgreSQL

Se um valor contiver um caractere não ASCII em uma codificação diferente de UTF8 - isso resultará em uma consulta SQL com falha (PGRES_FATAL_ERROR:ERROR invalid byte sequence for encoding) e os dados não serão armazenados. Uma mensagem de aviso apropriada será gravada no *zabbix_server.log*.
 Relevante para pelo menos a versão 9.1.3 do PostgreSQL.

5 Suporte a arquivos grandes

O suporte a arquivos grandes, frequentemente abreviado como LFS, é o termo aplicado à capacidade de trabalhar com arquivos maiores que 2 GB em sistemas operacionais de 32 bits. O suporte a arquivos grandes afeta pelo menos o monitoramento de arquivos de log e todos os itens vfs.file.*. O suporte a arquivos grandes depende das capacidades do sistema no momento da compilação do Zabbix, mas é completamente desabilitado em um Solaris de 32 bits devido à sua incompatibilidade com procfs e swapctl.

6 Sensor

Visão geral

Cada chip de sensor recebe seu próprio diretório na árvore sysfs `/sys/devices`. Para encontrar todos os chips de sensores, é mais fácil seguir os links simbólicos de dispositivos a partir de `/sys/class/hwmon/hwmon*`, onde `*` é um número real (0,1,2,...).

As leituras dos sensores estão localizadas no diretório `/sys/class/hwmon/hwmon*/` para dispositivos virtuais, ou no diretório `/sys/class/hwmon/hwmon*/device` para dispositivos não virtuais. Um arquivo chamado `name`, localizado dentro dos diretórios `hwmon*` ou `hwmon*/device`, contém o nome do chip, que corresponde ao nome do driver do kernel usado pelo chip do sensor.

Há apenas um valor de leitura de sensor por arquivo. O esquema comum para nomear os arquivos que contêm leituras de sensores dentro de qualquer um dos diretórios mencionados acima é: `<type><number>_<item>`, onde

- **type** - para chips de sensores é ”in” (voltagem), ”temp” (temperatura), ”fan” (ventilador), etc.
- **item** - ”input” (valor medido), ”max” (limite superior), ”min” (limite inferior), etc.

- **number** - sempre usado para elementos que podem estar presentes mais de uma vez (geralmente começa em 1, exceto para voltagens que começam em 0). Se os arquivos não se referem a um elemento específico, eles têm um nome simples sem número.

As informações sobre sensores disponíveis no host podem ser adquiridas usando as ferramentas **sensors-detect** e **sensors** (pacote lm-sensors). **Sensors-detect** ajuda a determinar quais módulos são necessários para os sensores disponíveis. Quando os módulos são carregados, o programa **sensors** pode ser usado para mostrar as leituras de todos os chips de sensores. A rotulagem das leituras dos sensores, usada por este programa, pode ser diferente do esquema de nomenclatura comum (`<type><number>_<item>`):

- se houver um arquivo chamado `<type><number>_label`, então o rótulo dentro deste arquivo será usado em vez do nome `<type><number><item>`;
- se não houver arquivo `<type><number>_label`, então o programa procura dentro do `/etc/sensors.conf` (pode ser também `/etc/sensors3.conf`, ou diferente) pela substituição do nome.

Essa rotulagem permite ao usuário determinar que tipo de hardware está sendo usado. Se não houver arquivo `<type><number>_label` nem rótulo dentro do arquivo de configuração, o tipo de hardware pode ser determinado pelo atributo name (`hwmon*/device/name`). Os nomes reais dos sensores, que o `zabbix_agent` aceita, podem ser obtidos executando o programa **sensors** com o parâmetro `-u` (`sensors -u`).

No programa **sensors**, os sensores disponíveis são separados pelo tipo de barramento (adaptador ISA, adaptador PCI, adaptador SPI, dispositivo virtual, interface ACPI, adaptador HID).

Obtendo identificadores de sensores

Os rótulos dos sensores, conforme impressos pelo comando `sensors`, nem sempre podem ser usados diretamente porque a rotulagem varia entre os fornecedores de chips. Por exemplo, a saída do `sensors` pode conter as seguintes linhas:

```
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)
```

Desses, apenas um nome de exibição corresponde diretamente a um identificador interno:

```
zabbix_get -s 127.0.0.1 -k sensor[lm85-i2c-0-2e,in0]
2.240000
```

Tentar usar outros rótulos de exibição (como `Vcore` ou `+12V`) não funcionará:

```
zabbix_get -s 127.0.0.1 -k sensor[lm85-i2c-0-2e,Vcore]
ZBX_NOTSUPPORTED
```

Para encontrar o identificador interno do sensor que o Zabbix aceita, execute `sensors -u`. Na saída, o seguinte pode ser observado:

```
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
...
```

Assim, o rótulo `Vcore` corresponde ao identificador `in1` e `+12V` corresponde a `in4`. De acordo com a especificação do kernel hwmon sysfs, essas são tensões nos pinos do chip e, de modo geral, podem precisar de escala.

```
zabbix_get -s 127.0.0.1 -k sensor[lm85-i2c-0-2e,in1]
1.301000
```

Não apenas leituras de tensão (`in`), mas também de corrente (`curr`), temperatura (`temp`) e velocidade do ventilador (`fan`) podem ser recuperadas pelo Zabbix.

Compatibilidade

Distribuições legadas

As leituras dos sensores são obtidas do diretório `/proc/sys/dev/sensors`:

- **device** - nome do dispositivo (se <mode> for usado, é uma expressão regular)
- **sensor** - nome do sensor (se <mode> for usado, é uma expressão regular)
- **mode** - valores possíveis: `avg`, `max`, `min` (se este parâmetro for omitido, device e sensor são tratados literalmente)

Exemplo de chave: `sensor[w83781d-i2c-0-2d,temp1]`.

Distribuições atuais

As leituras dos sensores são obtidas do diretório `/sys/class/hwmon`:

- **device** - nome do dispositivo (não é uma expressão regular). O nome do dispositivo pode ser o nome real do dispositivo (por exemplo, `0000:00:18.3`) ou o nome adquirido usando o programa sensors (por exemplo, `k8temp-pci-00c3`). Cabe ao usuário escolher qual nome usar.
- **sensor** - nome do sensor (não é uma expressão regular).
- **mode** - valores possíveis: `avg`, `max`, `min` (se este parâmetro for omitido, o dispositivo e o sensor são tratados literalmente).

Exemplos de chaves:

`sensor[k8temp-pci-00c3,temp,max]` ou `sensor[0000:00:18.3,temp1]`.

`sensor[smsc47b397-isa-0880,in,avg]` ou `sensor[smsc47b397.2176,in1]`.

7 Notas sobre o parâmetro memtype em itens proc.mem

Visão geral

O parâmetro **memtype** é suportado no Linux, AIX, FreeBSD, e Solaris.

Os três tipos mais comuns do 'memtype' são suportados em todas as plataformas: `pmem`, `rss` e `vsize`. Adicionalmente, tipos específicos de cada plataforma serão nelas suportados.

AIX

Veja os valores suportados para o parâmetro 'memtype' no AIX na tabela.

Valor suportado	Descrição	Fonte na estrutura procentry64	Tenta ser compatível com
vsize [1]	Tamanho da memória virtual	pi_size	
pmem	Porcentagem de memória real	pi_prm	ps -o pmem
rss	Tamanho do conjunto residente	pi_trss + pi_drss	ps -o rssize
size	Tamanho do processo (código + dados)	pi_dvm	Coluna SIZE do "ps gvw"
dsize	Tamanho dos dados	pi_dsize	
tsize	Tamanho do texto (código)	pi_tsize	Coluna TSIZ do "ps gvw"
sdsize	Tamanho dos dados da biblioteca compartilhada	pi_sdsize	
drss	Tamanho do conjunto residente de dados	pi_drss	
trss	Tamanho do conjunto residente de texto	pi_trss	

Notas para AIX:

1. Ao escolher parâmetros para a chave do item proc.mem[] no AIX, tente especificar critérios de seleção de processo mais restritos. Caso contrário, há o risco de obter processos indesejados contados no resultado de proc.mem[].

Exemplo:

```
$ zabbix_agentd -t proc.mem[,,,NonExistingProcess,rss]
proc.mem[,,,NonExistingProcess,rss]           [u|2879488]
```

Este exemplo mostra como especificar apenas o parâmetro da linha de comando (expressão regular para corresponder) resulta na auto-contabilização do agent do Zabbix - provavelmente não é o que você deseja.

2. Não use "ps -ef" para navegar pelos processos - ele mostra apenas processos não-kernel. Use "ps -Af" para ver todos os processos que serão vistos pelo agent Zabbix.

3. Vamos passar pelo exemplo do 'topasrec' para mostrar como o agent Zabbix proc.mem[] seleciona processos.

```
$ 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[] tem os argumentos:

```
proc.mem[<name>,<user>,<mode>,<cmdline>,<memtype>]
```

O primeiro critério é um nome de processo (argumento <name>). No nosso exemplo, o agent do Zabbix verá como 'topasrec'. Para corresponder, você precisa especificar 'topasrec' ou deixar em branco. O segundo critério é um nome de usuário (argumento <user>). Para corresponder, você precisa especificar 'root' ou deixar em branco. O terceiro critério usado na seleção do processo é um argumento <cmdline>. O agent do Zabbix verá seu valor como '/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'. Para corresponder, você precisa especificar uma expressão regular que corresponda a essa string ou deixar em branco.

Os argumentos <mode> e <memtype> são aplicados após o uso dos três critérios mencionados acima.

FreeBSD

Veja os valores suportados para o parâmetro 'memtype' no FreeBSD na tabela.

Valor suportado	Descrição	Fonte na estrutura kinfo_proc	Tenta ser compatível com
vsize	Tamanho da memória virtual	kp_eproc.e_vm.vm_map_size ou ki_size	ps -o vsz
pmem	Porcentagem de memória real	calculado a partir de rss	ps -o pmem
rss	Tamanho do conjunto residente	kp_eproc.e_vm.vm_rssize ou ki_rssize	ps -o rss
size [1]	Tamanho do processo (código + dados + pilha)	tsize + dsize + ssize	
tsize	Tamanho do texto (código)	kp_eproc.e_vm.vm_tsize ou ki_tsize	ps -o tsiz
dsize	Tamanho dos dados	kp_eproc.e_vm.vm_dsize ou ki_dsize	ps -o dsiz
ssize	Tamanho da pilha	kp_eproc.e_vm.vm_ssize ou ki_ssize	ps -o ssiz

Linux

Veja os valores suportados para o parâmetro 'memtype' no Linux na tabela.

Valor suportado	Descrição	Fonte no arquivo /proc/<pid>/status
vsize [1]	Tamanho da memória virtual	VmSize
pmem	Porcentagem de memória real	(VmRSS/total_memory) * 100
rss	Tamanho do conjunto residente	VmRSS
data	Tamanho do segmento de dados	VmData
exe	Tamanho do segmento de código	VmExe
hwm	Pico do conjunto residente	VmHWM
lck	Tamanho da memória travada	VmLck
lib	Tamanho das bibliotecas compartilhadas	VmLib
peak	Pico de memória virtual	VmPeak
pin	Tamanho das páginas fixadas	VmPin
pte	Tamanho das entradas da tabela de páginas	VmPTE
size	Tamanho do código do processo + dados + segmentos de pilha	VmExe + VmData + VmStk
stk	Tamanho do segmento de pilha	VmStk
swap	Tamanho do espaço de swap utilizado	VmSwap

Notas para Linux:

1. Nem todos os valores de 'memtype' são suportados por kernels Linux mais antigos. Por exemplo, kernels Linux 2.4 não suportam os valores `hwm`, `pin`, `peak`, `pte` e `swap`.

2. Observamos que o auto-monitoramento do processo de verificação ativa do agent Zabbix com `proc.mem[...,...,...,...,data]` mostra um valor que é 4 kB maior do que o relatado pela linha `VmData` no arquivo /proc/<pid>/status do agent. No momento da auto-medição, o segmento de dados do agent aumenta em 4 kB e depois retorna ao tamanho anterior.

Solaris

Veja os valores suportados para o parâmetro 'memtype' no Solaris na tabela.

Valor suportado	Descrição	Fonte na estrutura psinfo	Tenta ser compatível com
vsize [1]	Tamanho da imagem do processo	pr_size	ps -o vsz
pmem	Porcentagem de memória real	pr_pctmem	ps -o pmem
rss	Tamanho do conjunto residente Pode ser subestimado - veja a descrição de rss em "man ps".	pr_rssize	ps -o rss

Notas de rodapé

[1] Valor padrão.

8 Notas sobre a seleção de processos em itens proc.mem e proc.num

Processos que modificam sua linha de comando

Alguns programas usam a modificação de sua linha de comando como um método para exibir sua atividade atual. Um usuário pode ver a atividade executando os comandos `ps` e `top`. Exemplos de tais programas incluem *PostgreSQL*, *Sendmail*, *Zabbix*.

Vamos ver um exemplo no Linux. Vamos supor que queremos monitorar o número de processos do agent Zabbix.

O comando `ps` mostra os processos de interesse como

```
$ 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]
...
```

Selecionar processos por nome e usuário faz o trabalho:

```
$ zabbix_get -s localhost -k 'proc.num[zabbix_agentd,zabbix]'
6
```

Agora vamos renomear o executável `zabbix_agentd` para `zabbix_agentd_30` e reiniciá-lo.

O `ps` agora mostra

```
$ ps -fu zabbix
UID        PID  PPID  C STIME TTY          TIME CMD
...
zabbix    6715     1  0 12:53 ?        00:00:00 sbin/zabbix_agentd_30 -c /home/zabbix/zabbix_agentd.conf
zabbix    6716  6715  0 12:53 ?        00:00:00 sbin/zabbix_agentd_30: collector [idle 1 sec]
zabbix    6717  6715  0 12:53 ?        00:00:00 sbin/zabbix_agentd_30: listener #1 [waiting for connection
zabbix    6718  6715  0 12:53 ?        00:00:00 sbin/zabbix_agentd_30: listener #2 [waiting for connection
zabbix    6719  6715  0 12:53 ?        00:00:00 sbin/zabbix_agentd_30: listener #3 [waiting for connection
zabbix    6720  6715  0 12:53 ?        00:00:00 sbin/zabbix_agentd_30: active checks #1 [idle 1 sec]
...
```

Agora, selecionar processos por nome e usuário produz um resultado incorreto:

```
$ zabbix_get -s localhost -k 'proc.num[zabbix_agentd_30,zabbix]'
1
```

Por que uma simples renomeação do executável para um nome mais longo levou a um resultado tão diferente?

O agent Zabbix começa verificando o nome do processo. O arquivo /proc/<pid>/status é aberto e a linha Name é verificada. No nosso caso, as linhas Name são:

```
$ 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
```

O nome do processo no arquivo status é truncado para 15 caracteres.

Um resultado semelhante pode ser visto com o comando ps:

```
$ ps -u zabbix
  PID TTY          TIME CMD
...
 6715 ?        00:00:00 zabbix_agentd_3
 6716 ?        00:00:01 zabbix_agentd_3
 6717 ?        00:00:00 zabbix_agentd_3
 6718 ?        00:00:00 zabbix_agentd_3
 6719 ?        00:00:00 zabbix_agentd_3
 6720 ?        00:00:00 zabbix_agentd_3
 ...
```

Obviamente, isso não é igual ao valor do parâmetro name do nosso proc.num[], zabbix_agentd_30. Não tendo conseguido corresponder o nome do processo do arquivo status, o agent Zabbix recorre ao arquivo /proc/<pid>/cmdline.

Como o agent vê o arquivo "cmdline" pode ser ilustrado executando um comando

```
$ for i in 6715 6716 6717 6718 6719 6720; do cat /proc/$i/cmdline | awk '{gsub(/\x0/,"<NUL>"); print};'; d
sbin/zabbix_agentd_30<NUL>-c<NUL>/home/zabbix/zabbix_agentd.conf<NUL>
sbin/zabbix_agentd_30: collector [idle 1 sec]<NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><
sbin/zabbix_agentd_30: listener #1 [waiting for connection]<NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><N
sbin/zabbix_agentd_30: listener #2 [waiting for connection]<NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><N
sbin/zabbix_agentd_30: listener #3 [waiting for connection]<NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><N
sbin/zabbix_agentd_30: active checks #1 [idle 1 sec]<NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL
```

Os arquivos /proc/<pid>/cmdline no nosso caso contêm bytes nulos invisíveis e não imprimíveis, usados para terminar strings na linguagem *C*. Os bytes nulos são mostrados como "<NUL>" neste exemplo.

O agent Zabbix verifica o "cmdline" para o processo principal e pega um zabbix_agentd_30, que corresponde ao valor do nosso parâmetro name zabbix_agentd_30. Portanto, o processo principal é contado pelo item proc.num[zabbix_agentd_30,zabbix].

Ao verificar o próximo processo, o agent pega zabbix_agentd_30: collector [idle 1 sec] do arquivo cmdline e ele não corresponde ao nosso parâmetro name zabbix_agentd_30. Portanto, apenas o processo principal que não modifica sua linha de comando é contado. Outros processos do agent modificam sua linha de comando e são ignorados.

Este exemplo mostra que o parâmetro name não pode ser usado em proc.mem[] e proc.num[] para selecionar processos neste caso.

> **Note:**
> Para o item proc.get[], quando o agent Zabbix verifica o "cmdline" para o nome do processo, ele usará apenas parte do nome começando a partir da última barra e até o primeiro espaço ou sinal de dois pontos. O nome do processo recebido do arquivo cmdline só será usado se seu início corresponder completamente ao nome do processo encurtado no arquivo status. O algoritmo é o mesmo tanto para o nome do processo no filtro quanto na saída JSON.

Usar o parâmetro cmdline com uma expressão regular adequada produz um resultado correto:

```
$ zabbix_get -s localhost -k 'proc.num[,zabbix,,zabbix_agentd_30[ :]]'
6
```

Tenha cuidado ao usar os itens proc.get[], proc.mem[] e proc.num[] para monitorar programas que modificam suas linhas de comando.

Antes de colocar os parâmetros name e cmdline nos itens proc.get[], proc.mem[] e proc.num[], você pode querer testar os parâmetros usando o item proc.num[] e o comando ps.

Threads do kernel Linux

Threads não podem ser selecionadas com o parâmetro cmdline em itens proc.get[], proc.mem[] e proc.num[]

Vamos tomar como exemplo uma das threads do kernel:

```
$ ps -ef| grep kthreadd
root         2     0  0 09:33 ?        00:00:00 [kthreadd]
```

Ela pode ser selecionada com o parâmetro name do processo:

```
$ zabbix_get -s localhost -k 'proc.num[kthreadd,root]'
1
```

Mas a seleção pelo parâmetro cmdline do processo não funciona:

```
$ zabbix_get -s localhost -k 'proc.num[,root,,kthreadd]'
0
```

A razão é que o agent do Zabbix pega a expressão regular especificada no parâmetro cmdline e a aplica ao conteúdo do processo /proc/<pid>/cmdline. Para threads do kernel, seus arquivos /proc/<pid>/cmdline estão vazios. Portanto, o parâmetro cmdline nunca corresponde.

Contagem de threads em itens proc.mem[] e proc.num[]

threads do kernel Linux são contados pelo item proc.num[], mas não relatam memória no item proc.mem[]. Por exemplo:

```
$ ps -ef | grep kthreadd
raiz 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: Não é possível obter a quantidade de memória "VmSize".
```

Mas o que acontece se houver um processo de usuário com o mesmo nome de um thread do núcleo? Então poderia ficar assim:

```
$ ps -ef | grep kthreadd
raiz 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[] contou tanto a thread do kernel quanto o processo do usuário. proc.mem[] reporta a memória apenas para o processo do usuário e conta o memória de thread do kernel como se fosse 0. Isso é diferente do caso acima quando ZBX_NOTSUPPORTED foi relatado.

Tenha cuidado ao usar os itens proc.mem[] e proc.num[] se o programa name acontece para corresponder a um dos threads.

Antes de colocar parâmetros nos itens proc.mem[] e proc.num[], você pode querer testar os parâmetros usando o item proc.num[] e ps comando.

9 Detalhes de implementação das verificações net.tcp.service e net.udp.service

A implementação das verificações net.tcp.service e net.udp.service é detalhada nesta página para vários serviços especificados no parâmetro service.

Parâmetros do item net.tcp.service

ftp

Cria uma conexão TCP e espera que os primeiros 4 caracteres da resposta sejam ”220 ”, então envia ”QUIT\r\n”. A porta padrão 21 é usada se não for especificada.

http

Cria uma conexão TCP sem esperar ou enviar nada. A porta padrão 80 é usada se não for especificada.

https

Usa (e só funciona com) libcurl, não verifica a autenticidade do certificado, não verifica o nome do host no certificado SSL, apenas busca o cabeçalho da resposta (requisição HEAD). A porta padrão 443 é usada se não for especificada.

imap

Cria uma conexão TCP e espera que os primeiros 4 caracteres da resposta sejam ”* OK”, então envia ”a1 LOGOUT\r\n”. A porta padrão 143 é usada se não for especificada.

ldap

Abre uma conexão com um servidor LDAP e executa uma operação de busca LDAP com o filtro definido como (objectClass=*). Espera a recuperação bem-sucedida do primeiro atributo da primeira entrada. A porta padrão 389 é usada se não for especificada.

nntp

Cria uma conexão TCP e espera que os primeiros 3 caracteres da resposta sejam ”200” ou ”201”, então envia ”QUIT\r\n”. A porta padrão 119 é usada se não for especificada.

pop

Cria uma conexão TCP e espera que os primeiros 3 caracteres da resposta sejam ”+OK”, então envia ”QUIT\r\n”. A porta padrão 110 é usada se não for especificada.

smtp

Cria uma conexão TCP e espera que os primeiros 3 caracteres da resposta sejam ”220”, seguidos por um espaço, o final da linha ou um traço. As linhas contendo um traço pertencem a uma resposta multilinha e a resposta será relida até que uma linha sem o traço seja recebida. Em seguida, envia ”QUIT\r\n”. A porta padrão 25 é usada se não for especificada.

ssh

Cria uma conexão TCP. Se a conexão foi estabelecida, ambos os lados trocam uma string de identificação (SSH-major.minor-XXXX), onde major e minor são as versões do protocolo e XXXX é uma string. O Zabbix verifica se a string correspondente à especificação é encontrada e então envia de volta a string ”SSH-major.minor-zabbix_agent\r\n” ou ”0\n” em caso de incompatibilidade. A porta padrão 22 é usada se não for especificada.

tcp

Cria uma conexão TCP sem esperar ou enviar nada. Diferente das outras verificações, exige que o parâmetro da porta seja especificado.

telnet

Cria uma conexão TCP e espera um prompt de login (’:’ no final). A porta padrão 23 é usada se não for especificada.

Item net.udp.service parameters

ntp

Envia um pacote SNTP sobre UDP e valida a resposta conforme RFC 4330, seção 5. O padrão é a porta 123 .

10 parâmetros proc.get

Visão geral

O item **proc.get**[<name>,<user>,<cmdline>,<mode>] é suportado em Linux, Windows, FreeBSD, OpenBSD e NetBSD.

A lista de parâmetros de processo retornados pelo item varia dependendo do sistema operacional e do valor do argumento ’mode’.

Linux

Os seguintes parâmetros de processo são retornados no Linux para cada modo:

mode=process	mode=thread	mode=summary
pid: PID	pid: PID	name: nome do processo
ppid: PID pai	ppid: PID pai	processes: número de processos
name: nome do processo	name: nome do processo	vsize: tamanho da memória virtual
cmdline: comando com argumentos	user: usuário (real) sob o qual o processo é executado	pmem: porcentagem de memória real
user: usuário (real) sob o qual o processo é executado	group: grupo (real) sob o qual o processo é executado	rss: tamanho do conjunto residente

mode=process	mode=thread	mode=summary
group: grupo (real) sob o qual o processo é executado	uid: ID do usuário	data: tamanho do segmento de dados
uid: ID do usuário	gid: ID do grupo sob o qual o processo é executado	exe: tamanho do segmento de código
gid: ID do grupo sob o qual o processo é executado	tid: ID da thread	lib: tamanho das bibliotecas compartilhadas
vsize: tamanho da memória virtual	tname: nome da thread	lck: tamanho da memória travada
pmem: porcentagem de memória real	cputime_user: tempo total de CPU gasto no modo usuário (valor expresso em clock ticks — divida por `getconf CLK_TCK` ou `sysconf(_SC_CLK_TCK)` para obter segundos)	pin: tamanho das páginas fixadas
rss: tamanho do conjunto residente	cputime_system: tempo total de CPU gasto no modo sistema (valor expresso em clock ticks — divida por `getconf CLK_TCK` ou `sysconf(_SC_CLK_TCK)` para obter segundos)	pte: tamanho das entradas da tabela de páginas
data: tamanho do segmento de dados	state: estado da thread	size: tamanho do código do processo + dados + segmentos de pilha
exe: tamanho do segmento de código	ctx_switches: número de trocas de contexto	stk: tamanho do segmento de pilha
hwm: pico do conjunto residente	page_faults: número de falhas de página	swap: tamanho do espaço de swap usado
lck: tamanho da memória travada		cputime_user: tempo total de CPU gasto no modo usuário (valor expresso em clock ticks — divida por `getconf CLK_TCK` ou `sysconf(_SC_CLK_TCK)` para obter segundos)
lib: tamanho das bibliotecas compartilhadas		cputime_system: tempo total de CPU gasto no modo sistema (valor expresso em clock ticks — divida por `getconf CLK_TCK` ou `sysconf(_SC_CLK_TCK)` para obter segundos)
peak: pico da memória virtual		ctx_switches: número de trocas de contexto
pin: tamanho das páginas fixadas		threads: número de threads
pte: tamanho das entradas da tabela de páginas		page_faults: número de falhas de página
size: tamanho do código do processo + dados + segmentos de pilha		pss: memória proporcional do conjunto residente
stk: tamanho do segmento de pilha		
swap: tamanho do espaço de swap usado		
cputime_user: tempo total de CPU gasto no modo usuário (valor expresso em clock ticks — divida por `getconf CLK_TCK` ou `sysconf(_SC_CLK_TCK)` para obter segundos)		
cputime_system: tempo total de CPU gasto no modo sistema (valor expresso em clock ticks — divida por `getconf CLK_TCK` ou `sysconf(_SC_CLK_TCK)` para obter segundos)		
state: estado do processo (recuperado transparentemente do procfs, forma longa)		

mode=process	mode=thread	mode=summary
ctx_switches: número de trocas de contexto threads: número de threads page_faults: número de falhas de página pss: memória proporcional do conjunto residente		

Sistemas operacionais baseados em BSD

Os seguintes parâmetros de processo são retornados no FreeBSD, OpenBSD e NetBSD para cada modo:

mode=process	mode=thread	mode=summary
pid: PID	pid: PID	name: nome do processo
ppid: PID do processo pai	ppid: PID do processo pai	processes: número de processos
jid: ID da jail (apenas FreeBSD)	jid: ID da jail (apenas FreeBSD)	vsize: tamanho da memória virtual
jname: nome da jail (apenas FreeBSD)	jname: nome da jail (apenas FreeBSD)	pmem: porcentagem de memória real (apenas FreeBSD)
name: nome do processo	name: nome do processo	rss: tamanho do conjunto residente
cmdline: comando com argumentos	user: usuário (real) sob o qual o processo é executado	size: tamanho do processo (código + dados + pilha)
user: usuário (real) sob o qual o processo é executado	group: grupo (real) sob o qual o processo é executado	tsize: tamanho do texto (código)
group: grupo (real) sob o qual o processo é executado	uid: ID do usuário	dsize: tamanho dos dados
uid: ID do usuário	gid: ID do grupo sob o qual o processo é executado	ssize: tamanho da pilha
gid: ID do grupo sob o qual o processo é executado	tid: ID da thread	cputime_user: tempo total de CPU gasto no modo usuário (relatado em segundos)
vsize: tamanho da memória virtual	tname: nome da thread	cputime_system: tempo total de CPU gasto no modo sistema (relatado em segundos)
pmem: porcentagem de memória real (apenas FreeBSD)	cputime_user: tempo total de CPU gasto no modo usuário (relatado em segundos)	ctx_switches: número de trocas de contexto
rss: tamanho do conjunto residente	cputime_system: tempo total de CPU gasto no modo sistema (relatado em segundos)	threads: número de threads (não suportado para NetBSD)
size: tamanho do processo (código + dados + pilha)	state: estado da thread	stk: tamanho do segmento de pilha
tsize: tamanho do texto (código)	ctx_switches: número de trocas de contexto	page_faults: número de falhas de página
dsize: tamanho dos dados	io_read_op: número de vezes que o sistema teve que realizar entrada	fds: número de descritores de arquivos (apenas OpenBSD)
ssize: tamanho da pilha	io_write_op: número de vezes que o sistema teve que realizar saída	swap: tamanho do espaço de swap usado
cputime_user: tempo total de CPU gasto no modo usuário (relatado em segundos)		io_read_op: número de vezes que o sistema teve que realizar entrada
cputime_system: tempo total de CPU gasto no modo sistema (relatado em segundos)		io_write_op: número de vezes que o sistema teve que realizar saída
state: estado do processo (espera por disco/executando/dormindo/parada por rastreamento/zumbi/outro)		
ctx_switches: número de trocas de contexto		
threads: número de threads (não suportado para NetBSD)		

mode=process	mode=thread	mode=summary
page_faults: número de falhas de página fds: número de descritores de arquivos (apenas OpenBSD) swap: tamanho do espaço de swap usado io_read_op: número de vezes que o sistema teve que realizar entrada io_write_op: número de vezes que o sistema teve que realizar saída		

Windows

Os seguintes parâmetros de processo são retornados no Windows para cada modo:

mode=process	mode=thread	mode=summary
pid: PID ppid: PID pai name: nome do processo user: usuário sob o qual o processo é executado sid: SID do usuário vmsize: tamanho da memória virtual wkset: tamanho do conjunto de trabalho do processo cputime_user: tempo total de CPU gasto no modo usuário (relatado em milissegundos) cputime_system: tempo total de CPU gasto no modo sistema (relatado em milissegundos) threads: número de threads page_faults: número de falhas de página handles: número de handles io_read_b: bytes de IO lidos io_write_b: bytes de IO gravados io_read_op: operações de leitura de IO io_write_op: operações de gravação de IO io_other_b: bytes de IO transferidos, exceto operações de leitura e gravação io_other_op: operações de IO, exceto operações de leitura e gravação	pid: PID ppid: PID pai name: nome do processo user: usuário sob o qual o processo é executado sid: SID do usuário tid: ID do thread	name: nome do processo processes: número de processos vmsize: tamanho da memória virtual wkset: tamanho do conjunto de trabalho do processo cputime_user: tempo total de CPU gasto no modo usuário (relatado em milissegundos) cputime_system: tempo total de CPU gasto no modo sistema (relatado em milissegundos) threads: número de threads page_faults: número de falhas de página handles: número de handles io_read_b: bytes de IO lidos io_write_b: bytes de IO gravados io_read_op: operações de leitura de IO io_write_op: operações de gravação de IO io_other_b: bytes de IO transferidos, exceto operações de leitura e gravação io_other_op: operações de IO, exceto operações de leitura e gravação

11 Configurações de interface de host inacessível/indisponível

Visão geral

Diversos parâmetros de configuração definem como o Zabbix Server irá agir quando uma verificação de agente (Zabbix, SNMP, IPMI, JMX) falhar e um host passar a se tornar inacessível.

Interface inacessível

Uma interface de host é tratada como inacessível após uma verificação com falha (erro de rede, timeout) pelos agentes Zabbix, SNMP, IPMI ou JMX. Desde o Zabbix 6.2.0, as verificações ativas do agent Zabbix também afetam a disponibilidade da interface. Quando as verificações ativas se tornam indisponíveis, elas contribuem para o status geral de disponibilidade da interface do agent.

A partir do momento em que uma interface se torna inacessível, o parâmetro **UnreachableDelay** define com que frequência ela será rechecada usando um dos items (incluindo regras LLD). Essas rechecagens são realizadas por pollers de inacessibilidade (ou pollers IPMI para verificações IPMI). Por padrão, o intervalo entre verificações consecutivas de acessibilidade é de 15 segundos.

Attention:
As verificações realizadas por pollers assíncronos não são movidas para pollers de inacessibilidade.

No log do server Zabbix, a inacessibilidade é indicada por mensagens como:

```
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
```

As mensagens de log especificam exatamente o item que falhou e seu tipo (agent Zabbix).

Note:
O parâmetro *Timeout* também afetará o quão cedo uma interface será rechecada durante a inacessibilidade. Se o Timeout estiver definido para 20 segundos e o UnreachableDelay para 30 segundos, a próxima verificação será em 50 segundos após a primeira tentativa.

O parâmetro **UnreachablePeriod** define a duração total do período de inacessibilidade. Por padrão, o UnreachablePeriod é de 45 segundos. Esse valor deve ser várias vezes maior que o UnreachableDelay para garantir que uma interface seja rechecada várias vezes antes de ser marcada como indisponível.

Um item interno, zabbix[host,active_agent,available], permite monitorar a disponibilidade de verificações ativas em cenários de inacessibilidade.

Alternando a interface de volta para disponível

Quando o período de inacessibilidade termina, a interface é sondada novamente, diminuindo a prioridade para o item que tornou a interface inacessível Estado. Se a interface inacessível reaparecer, o monitoramento retorna ao normal automaticamente:

```
retomando as verificações do agente Zabbix no host "Novo host": conexão restaurada
```

Note:
Uma vez que a interface se torna disponível, o host não pesquisa todos seus itens imediatamente por dois motivos:
- Pode sobrecarregar o host.
- O tempo de restauração da interface nem sempre corresponde ao item planejado horário da agenda de votação.

Assim, após a disponibilização da interface, os itens não são pesquisados imediatamente, mas eles estão sendo remarcados para a próxima votação volta.

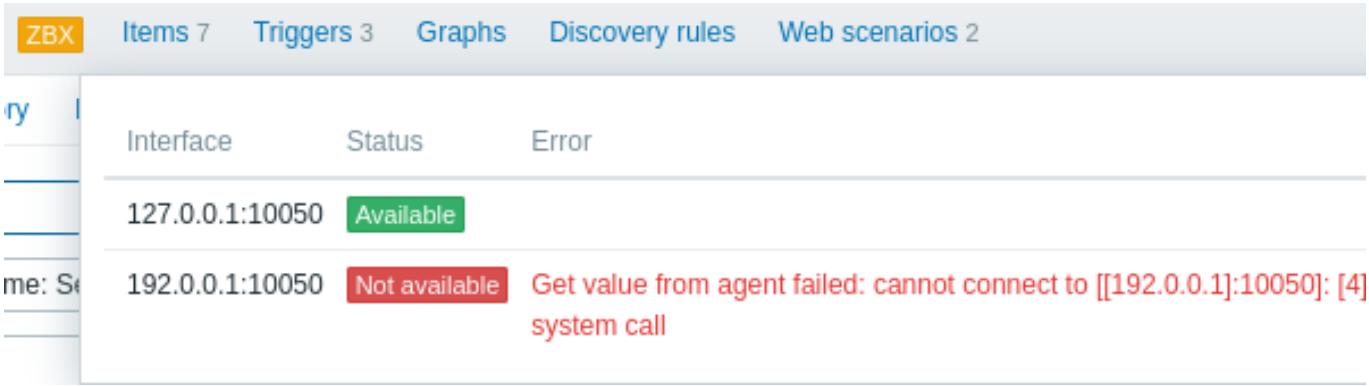
Interface indisponível

Após o término do UnreachablePeriod e a interface não ter reaparecido, a interface é tratada como indisponível.

No log do server isso é indicado por mensagens como estas:

```
temporarily disabling Zabbix agent checks on host "New host": interface unavailable
```

e no frontend o ícone de disponibilidade do host vai de verde/cinza para amarelo/vermelho (os detalhes da interface indisponível podem ser vistos na caixa de dica que é exibida quando o mouse é posicionado sobre o ícone de disponibilidade do host):



O parâmetro **UnavailableDelay** define com que frequência uma interface é verificada durante a indisponibilidade da interface.

Por padrão, é de 60 segundos (então, neste caso, ”temporarily disabling”, da mensagem de log acima, significará desabilitar as verificações por um minuto).

Quando a conexão com a interface é restaurada, o monitoramento retorna ao normal automaticamente também:

```
enabling Zabbix agent checks on host "New host": interface became available
```

12 Monitoramento remoto das estatísticas do Zabbix

Visão geral

É possível tornar algumas métricas internas do Zabbix server e proxy acessíveis remotamente por outra instância do Zabbix ou por uma ferramenta de terceiros. Isso pode ser útil para que equipes de suporte/provedores de serviço possam monitorar remotamente os servidores/proxies Zabbix de seus clientes ou, em organizações onde o Zabbix não é a principal ferramenta de monitoramento, para que as métricas internas do Zabbix possam ser monitoradas por um sistema de terceiros em uma configuração de monitoramento centralizado.

As estatísticas internas do Zabbix são expostas para um conjunto configurável de endereços listados no novo parâmetro ’StatsAllowedIP’ do server/proxy. As solicitações serão aceitas apenas desses endereços.

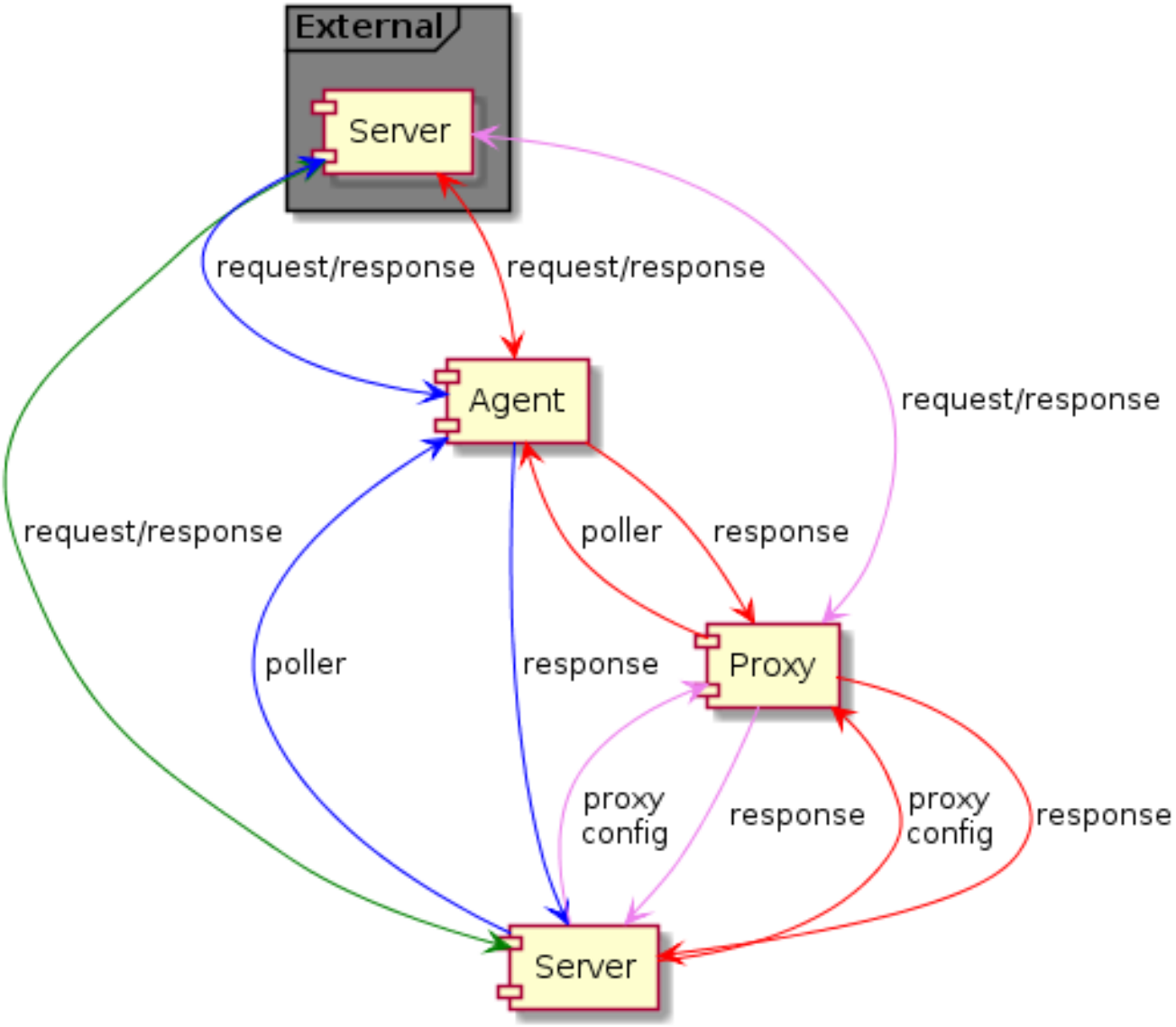
Itens

Para configurar a consulta de estatísticas internas em outra instância do Zabbix, você pode usar dois itens:

- `zabbix[stats,<ip>,<port>]` item interno - para controle remoto direto consultas do servidor/proxy Zabbix. <ip> e <port> são usados para identificar a instância de destino.
- `zabbix.stats[<ip>,<port>]` item de agente - para remoto baseado em agente consultas do servidor/proxy Zabbix. <ip> e <port> são usados para identificar a instância de destino.

Veja também: Itens internos, itens do agente Zabbix

O diagrama a seguir ilustra o uso de qualquer item dependendo o contexto.



- - Servidor → instância externa do Zabbix (`zabbix[stats,<ip>,<port>]`)
- - Servidor → proxy → instância externa do Zabbix (`zabbix[stats,<ip>,<port>]`)
- - Servidor → agente → instância externa do Zabbix (`zabbix.stats[<ip>,<port>]`)
- - Servidor → proxy → agente → instância externa do Zabbix (`zabbix.stats[<ip>,<port>]`)

Para certificar-se de que a instância de destino permite consultá-la pelo instância, liste o endereço da instância externa no Parâmetro 'StatsAllowedIP' na instância de destino.

Métricas expostas

Os itens de estatísticas coletam as estatísticas em lote e retornam um JSON, que é a base para que itens dependentes obtenham seus dados. As seguintes métricas internas são retornadas por qualquer um dos dois itens:

- `zabbix[boottime]`
- `zabbix[hosts]`
- `zabbix[items]`
- `zabbix[items_unsupported]`
- `zabbix[preprocessing]` (apenas no server)
- `zabbix[preprocessing_queue]` (apenas no server)
- `zabbix[process,<type>,<mode>,<state>]` (apenas estatísticas baseadas em tipo de processo)
- `zabbix[rcache,<cache>,<mode>]`
- `zabbix[requiredperformance]`
- `zabbix[triggers]` (apenas no server)
- `zabbix[uptime]`
- `zabbix[vcache,buffer,<mode>]` (apenas no server)

- zabbix[vcache,cache,<parameter>]
- zabbix[version]
- zabbix[vmware,buffer,<mode>]
- zabbix[wcache,<cache>,<mode>] (tipo de cache 'trends' apenas no server)

Templates

Templates estão disponíveis para o monitoramento remoto de métricas internas do Zabbix server ou proxy a partir de uma instância externa:

- Saúde remota do Zabbix server
- Saúde remota do Zabbix proxy

Observe que, para usar um template para monitoramento remoto de várias instâncias externas, é necessário um host separado para cada monitoramento de instância externa.

Processo trapper

O recebimento de solicitações de métricas internas de uma instância Zabbix externa é tratado pelo processo trapper, que valida a solicitação, coleta as métricas, cria o buffer de dados JSON e envia o JSON preparado de volta, por exemplo, do server:

```
{
  "response": "success",
  "data": {
    "boottime": N,
    "uptime": N,
    "hosts": N,
    "items": N,
    "items_unsupported": N,
    "preprocessing": {
        "queued": {
          "count": N,
          "size": N
        },
        "direct": {
          "count": N,
          "size": N
        },
        "queue": N
    },
    "preprocessing_queue": N,
    "process": {
      "alert manager": {
        "busy": {
          "avg": N,
          "max": N,
          "min": N
        },
        "idle": {
          "avg": N,
          "max": N,
          "min": N
        },
        "count": N
      },
...
    },
    "queue": N,
    "rcache": {
      "total": N,
      "free": N,
      "pfree": N,
      "used": N,
      "pused": N
    },
    "requiredperformance": N,
```

```
    "triggers": N,
    "uptime": N,
    "vcache": {
      "buffer": {
        "total": N,
        "free": N,
        "pfree": N,
        "used": N,
        "pused": N
      },
      "cache": {
        "requests": N,
        "hits": N,
        "misses": N,
        "mode": N
      }
    },
    "vmware": {
      "total": N,
      "free": N,
      "pfree": N,
      "used": N,
      "pused": N
    },
    "version": "N",
    "wcache": {
      "values": {
        "all": N,
        "float": N,
        "uint": N,
        "str": N,
        "log": N,
        "text": N,
        "not supported": N
      },
      "history": {
        "pfree": N,
        "free": N,
        "total": N,
        "used": N,
        "pused": N
      },
      "index": {
        "pfree": N,
        "free": N,
        "total": N,
        "used": N,
        "pused": N
      },
      "trend": {
        "pfree": N,
        "free": N,
        "total": N,
        "used": N,
        "pused": N
      }
    }
  }
}
```

Itens da fila interna

Há também outros dois itens que permitem especificamente a consulta remota estatísticas da fila interna em outra instância do

Zabbix:

- `zabbix[stats,<ip>,<port>,queue,<from>,<to>]` item interno - para consultas de filas internas diretas para servidor/proxy Zabbix remoto
- `zabbix.stats[<ip>,<port>,queue,<from>,<to>]` item do agente - para consultas de filas internas baseadas em agente para servidor/proxy Zabbix remoto

Veja também: Itens internos, itens do agente Zabbix

13 Configurando o Kerberos com o Zabbix

Visão geral

A autenticação Kerberos pode ser usada no monitoramento da web e em items HTTP no Zabbix.

Esta página descreve um exemplo de configuração do Kerberos para o servidor Zabbix realizar o monitoramento da web de `www.example.com` com um principal Kerberos para o processo Zabbix no Debian/Ubuntu.

Configuração

1. Instale o KDC e as utilitários do cliente:

```
sudo apt update
sudo apt install krb5-kdc krb5-admin-server krb5-user
```

Durante a configuração do pacote, responda às perguntas, por exemplo:

```
Default Kerberos version 5 realm: EXAMPLE.COM
Kerberos servers for your realm: localhost (ou seu FQDN)
Administrative server for your Kerberos realm: localhost (ou seu FQDN)
```

2. Mapeie um hostname amigável (opcional, para testes locais).

Edite o /etc/hosts e adicione uma entrada para seu DC e servidor web se você não tiver DNS:

```
sudo vi /etc/hosts
```

Exemplo de linha que você pode adicionar:

```
192.168.1.100  dc01.example.com dc01
```

3. Configure o cliente Kerberos e o realm do KDC:

```
sudo vi /etc/krb5.conf
```

Exemplo de configurações:

```
[libdefaults]
    default_realm = EXAMPLE.COM
    dns_lookup_realm = false
    dns_lookup_kdc = false
    rdns = false
    ticket_lifetime = 24h
    renew_lifetime = 7d
    forwardable = true

[realms]
    EXAMPLE.COM = {
        kdc = dc01.example.com
        admin_server = dc01.example.com
    }

[domain_realm]
    .example.com = EXAMPLE.COM
    example.com = EXAMPLE.COM
```

Se você planeja usar `.localdomain` ou outros nomes não públicos, adicione mapeamentos explícitos de domínio→realm para que o mapeamento de hostname→realm funcione. Inconsistências aqui causam erros `Server not found in Kerberos database`.

4. Inicialize o banco de dados Kerberos (uma vez, no host do KDC). Defina uma senha mestre segura quando solicitado:

```
sudo krb5_newrealm
```

5. Crie o principal `HTTP/host.fqdn@REALM` usando exatamente o hostname que os clientes usarão; prefira letras minúsculas (por exemplo, `HTTP/web.example.com@EXAMPLE.COM`). Uma diferença de maiúsculas/minúsculas ou nome causa `Server not found in Kerberos database`.

```
sudo kadmin.local
```

Dentro do `kadmin.local`:

```
addprinc kerb-admin@EXAMPLE.COM     # principal administrativo
addprinc -randkey HTTP/dc01.example.com@EXAMPLE.COM
ktadd -k /etc/apache2/http.keytab HTTP/dc01.example.com@EXAMPLE.COM
quit
```

Mova o keytab para o host web (ou mantenha local se for a mesma máquina) e defina permissões utilizáveis pelo Apache:

```
chown www-data:www-data /etc/apache2/http.keytab
chmod 600 /etc/apache2/http.keytab
#### verificar
sudo -u www-data -k /etc/apache2/http.keytab
```

6. Instale e habilite o módulo Apache GSSAPI:

```
sudo apt install libapache2-mod-auth-gssapi
sudo a2enmod auth_gssapi
sudo a2enmod headers
sudo systemctl restart apache2
```

> **Attention:**
> Nem todas as versões do mod_auth_gssapi suportam todas as diretivas `Gssapi*`. Se o Apache falhar com `Invalid command 'GssapiCredStore'`, remova a diretiva não suportada ou atualize o módulo.

7. Configure um VirtualHost (ajuste o `DocumentRoot` / caminho para sua interface Zabbix):

```
sudo vi /etc/apache2/sites-available/zabbix.conf
```

Dentro do `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>
```

Reinicie o Apache:

```
sudo systemctl restart apache2
```

8. Habilite/inicie os serviços do KDC e verifique as portas de escuta (host do KDC):

```
sudo systemctl enable --now krb5-kdc krb5-admin-server
ss -tnlp | grep :80    # ou: sudo netstat -tnlp | grep :80
```

9. Obtenha um TGT para teste (execute como o usuário que usará o ticket).

Espere ver `krbtgt/EXAMPLE.COM@EXAMPLE.COM` na lista de tickets. Execute `kinit` como o mesmo usuário do SO que precisa do ticket (por exemplo, `zabbix` para verificações web ou `www-data`/Apache para testes interativos de SSO no navegador). Tickets

emitidos para um usuário diferente do SO não serão visíveis a menos que KRB5CCNAME e permissões sejam ajustados.

```
kinit kerb-admin@EXAMPLE.COM
klist
```

10. Teste a troca SPNEGO com curl (de um cliente com um TGT válido). Um 200 OK (ou redirecionamento para o app) indica que o SPNEGO foi bem-sucedido:

```
curl -v --negotiate -u : http://dc01.example.com/
```

11. Opcionalmente, se a interface Zabbix deve aceitar logins autenticados por HTTP, habilite a autenticação HTTP no frontend do Zabbix (ui/conf/zabbix.conf.php):

```
$ALLOW_HTTP_AUTH = true;
```

Na interface web, vá para *Usuários* > *Autenticação* e vá para a guia *Configurações HTTP*. Marque a caixa *Habilitar autenticação HTTP* e clique em *Ok* no pop-up. Selecione "Formulário de login HTTP" no menu suspenso *Formulário de login padrão*. Decida se *Login com diferenciação de maiúsculas/minúsculas* se encaixa na política do seu diretório. Clique no botão *Atualizar* para finalizar.

12. Configuração do navegador (Firefox é usado como exemplo): defina `network.negotiate-auth.trusted-uris` para o(s) host(s) que executam o Negotiate (`dc01.example.com`) para que o navegador envie tokens Kerberos automaticamente.

Dentro do `about:config`:

```
network.negotiate-auth.trusted-uris = dc01.example.com
```

Agora, ao visitar `http://dc01.example.com`, você deve entrar diretamente no Zabbix sem o formulário.

13. Mantenha as chaves/tickets atualizados. O tempo de vida padrão do ticket Kerberos é de aproximadamente 10 horas. Adicione um cron/timer systemd para evitar expirações:

```
####para o serviço web
kinit -kt /etc/apache2/http.keytab HTTP/dc01.example.com@EXAMPLE.COM
####para o usuário de monitoramento
kinit -kt /var/lib/zabbix/kerb.keytab kerb-admin@EXAMPLE.COM
```

14. Verificações de manutenção:

- `klist -k /etc/apache2/http.keytab` — verifique se o principal do serviço está presente no keytab.
- `sudo tail -f /var/log/apache2/error.log` — observe erros do GSSAPI (`gss_acquire_cred[_from]() failed to get server creds` significa keytab/permissões ou principal ausente).
- `curl --negotiate` retornando 401/403 geralmente significa principal errado, nenhum ticket, incompatibilidade do cabeçalho do host ou problema de permissão do sistema de arquivos; verifique os logs e os mapeamentos de domínio em `/etc/krb5.conf`.

Notas de segurança e permissões de arquivos

Os arquivos keytab devem ser legíveis apenas pela conta que precisa deles. Permissões de exemplo: 0400 de propriedade de `zabbix:zabbix` para um keytab de usuário zabbix, ou 0440 e `root:www-data` para um keytab do Apache.

Evite armazenar senhas em texto simples de longa duração no host. Use keytabs ou principais de máquina associados ao domínio sempre que possível.

Ao executar testes ou scripts que definem KRB5CCNAME ou copiam keytabs, verifique novamente a propriedade e as permissões após a operação — um servidor web rejeitando credenciais geralmente é um problema de permissão de arquivo.

14 parâmetros modbus.get

Visão geral

A tabela abaixo apresenta detalhes dos parâmetros do item `modbus.get`.

Parâmetros

Parâmetro	Descrição	Padrão	Exemplo
endpoint	Protocolo e endereço do endpoint, definido como `protocol://connection_string` Valores possíveis para o protocolo: *rtu*, *ascii* (apenas Agent 2), *tcp* Formato da string de conexão: com *tcp* - `address:port` com linha serial: *rtu*, *ascii* - `port_name:speed:params` onde 'speed' - 1200, 9600 etc 'params' - bits de dados (5,6,7 ou 8), paridade (n,e ou o para nenhuma/par/e/ou ímpar), bits de parada (1 ou 2)	protocol: nenhum protocolo *rtu/ascii*: port_name: nenhum speed: 115200 params: 8n1 protocolo *tcp*: address: nenhum 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	Endereço Modbus do dispositivo para o qual se destina (1 a 247), veja MODBUS Messaging Implementation Guide (página 23) dispositivo tcp (não GW) irá ignorar o campo	serial: 1 tcp: 255 (0xFF)	2
function	Vazio ou valor de uma função suportada: 1 - Read Coil, 2 - Read Discrete Input, 3 - Read Holding Registers, 4 - Read Input Registers	vazio	3
address	Endereço do primeiro registrador, bobina ou entrada. Se 'function' estiver vazio, então 'address' deve estar no intervalo para: Coil - 00001 - 09999 Discrete input - 10001 - 19999 Input register - 30001 - 39999 Holding register - 40001 - 49999 Se 'function' não estiver vazio, o campo 'address' será de 0 até 65535 e usado sem modificação (PDU)	função vazia: 00001 função não vazia: 0	9999
count	Quantidade sequencial de 'type' que será lida do dispositivo, onde: para Coil ou Discrete input o 'type' = 1 bit para outros casos: (count*sizeof(type))/2 = quantidade real de registradores para leitura Se 'offset' não for 0, o valor será adicionado à 'quantidade real' O intervalo aceitável para 'quantidade real' é 1:65535	1	2
type	Tipo de dado: para Read Coil e Read Discrete Input - *bit* para Read Holding Registers e Read Input Registers: *int8* - 8bit *uint8* - 8bit (sem sinal) *int16* - 16bit *uint16* - 16bit (sem sinal) *int32* - 32bit *uint32* - 32bit (sem sinal) *float* - 32bit *uint64* - 64bit (sem sinal) *double* - 64bit	bit uint16	uint64

Parâmetro	Descrição	Padrão	Exemplo
endianness	Tipo de endianness: *be* - Big Endian *le* - Little Endian *mbe* - Mid-Big Endian *mle* - Mid-Little Endian Limitações: para 1 bit - be para 8 bits - be,le para 16 bits - be,le	be	le
offset	Número de registradores, a partir de 'address', cujo resultado será descartado. O tamanho de cada registrador é 16bit (necessário para suportar equipamentos que não suportam acesso de leitura aleatória).	0	4

15 Criando nomes personalizados de contadores de desempenho para VMware

Visão geral

O caminho do contador de desempenho do VMware tem o formato `group/counter[rollup]`, onde:

- `group` - o grupo do contador de desempenho, por exemplo *cpu*
- `counter` - o nome do contador de desempenho, por exemplo *usagemhz*
- `rollup` - o tipo de rollup do contador de desempenho, por exemplo *average*

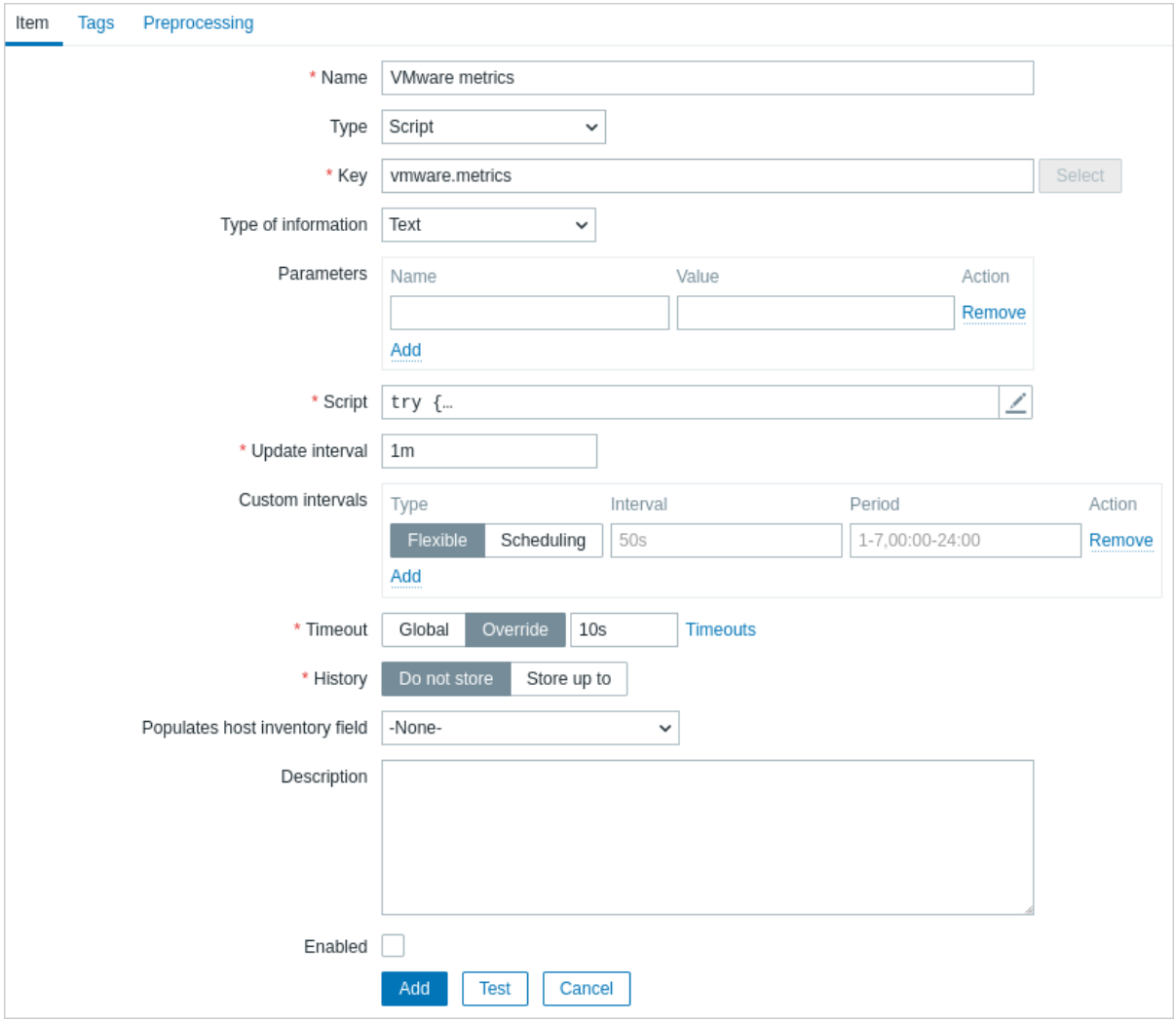
Portanto, o exemplo acima resultaria no seguinte caminho de contador: `cpu/usagemhz[average]`

As descrições dos grupos de contadores de desempenho, nomes de contadores e tipos de rollup podem ser encontradas na documentação da VMware.

É possível obter nomes internos e criar nomes personalizados de contadores de desempenho usando um item de script no Zabbix.

Configuração

1. Crie um item Script desabilitado no host principal do VMware (onde o item **eventlog[]** está presente) com os seguintes parâmetros:



- *Nome*: VMware metrics
- *Tipo*: Script
- *Chave*: vmware.metrics
- *Tipo de informação*: Texto
- *Script*: copie e cole o script fornecido abaixo
- *Timeout*: 10
- *Histórico*: Não armazenar
- *Habilitado*: desmarcado

Script

```
try {
    Zabbix.log(4, 'vmware metrics script');

    var result, resp,
    req = new HttpRequest();
    req.addHeader('Content-Type: application/xml');
    req.addHeader('SOAPAction: "urn:vim25/6.0"');

    login = '<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:urn="urn:vi
    <soapenv:Header/>\
    <soapenv:Body>\
        <urn:Login>\
            <urn:_this type="SessionManager">SessionManager</urn:_this>\
            <urn:userName>{$VMWARE.USERNAME}</urn:userName>\
            <urn:password>{$VMWARE.PASSWORD}</urn:password>\
        </urn:Login>\
    </soapenv:Body>\
</soapenv:Envelope>'
    resp = req.post("{$VMWARE.URL}", login);
    if (req.getStatus() != 200) {
```

```
        throw 'Response code: '+req.getStatus();
    }

    query = '<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:urn="urn:vi
<soapenv:Header/>\
    <soapenv:Body>\
        <urn:RetrieveProperties>\
            <urn:_this type="PropertyCollector">propertyCollector</urn:_this>\
            <urn:specSet>\
                <urn:propSet>\
                   <urn:type>PerformanceManager</urn:type>\
                   <urn:pathSet>perfCounter</urn:pathSet>\
                </urn:propSet>\
                <urn:objectSet>\
                   <urn:obj type="PerformanceManager">PerfMgr</urn:obj>\
                </urn:objectSet>\
            </urn:specSet>\
        </urn:RetrieveProperties>\
    </soapenv:Body>\
</soapenv:Envelope>'
    resp = req.post("{$VMWARE.URL}", query);
    if (req.getStatus() != 200) {
        throw 'Response code: '+req.getStatus();
    }
    Zabbix.log(4, 'vmware metrics=' + resp);
    result = resp;

    logout = '<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:urn="urn:v
    <soapenv:Header/>\
    <soapenv:Body>\
        <urn:Logout>\
            <urn:_this type="SessionManager">SessionManager</urn:_this>\
        </urn:Logout>\
    </soapenv:Body>\
</soapenv:Envelope>'

    resp = req.post("{$VMWARE.URL}",logout);
    if (req.getStatus() != 200) {
        throw 'Response code: '+req.getStatus();
    }

} catch (error) {
    Zabbix.log(4, 'vmware call failed : '+error);
    result = {};
}

return result;
```

Depois que o item for configurado, pressione o botão *Testar* e, em seguida, pressione *Obter valor*.

Copie o XML recebido para qualquer formatador de XML e encontre a métrica desejada.

Um exemplo de XML para uma métrica:

```
<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>
```

Use XPath para extrair o caminho do contador do XML recebido. Para o exemplo acima, o XPath será:

campo	xPath	valor
group	//groupInfo[../key=6]/key	cpu
counter	//nameInfo[../key=6]/key	usagemhz
rollup	//rollupType[../key=6]	average

O caminho do contador de desempenho resultante, neste caso, é: `cpu/usagemhz[average]`

16 Valores de retorno para system.sw.packages.get

Visão geral

Esta seção fornece detalhes sobre o valor de retorno para o item do agent Zabbix `system.sw.packages.get`.

Detalhes

A saída deste item é um array de objetos, cada um contendo as seguintes chaves:

- **name** - nome do pacote
- **manager** - gerenciador de pacotes que reportou este dado (`rpm`, `dpkg`, `pacman`, `pkgtool` ou `portage`)
- **version** - versão do pacote
- **size** - tamanho descompactado do pacote em bytes (se não disponível, definido como 0)
- **arch** - arquitetura do pacote
- **buildtime** - um objeto com 2 entradas:
  - **timestamp** - timestamp UNIX de quando o pacote foi construído (se não disponível, definido como 0)
  - **value** - data e hora legíveis de quando o pacote foi construído (se não disponível, definido como string vazia)
- **installtime** - um objeto com 2 entradas:
  - **timestamp** - timestamp UNIX de quando o pacote foi instalado (se não disponível, definido como 0)
  - **value** - data e hora legíveis de quando o pacote foi instalado (se não disponível, definido como string vazia)

Exemplo:

```
[
    {
        "name": "util-linux-core",
        "manager": "rpm",
        "version": "2.37.4-3.el9",
        "size": 1296335,
        "arch": "x86_64",
        "buildtime": {
            "timestamp" : 1653552239,
            "value" : "Sep 20 01:39:40 2021 UTC"
```

```
        },
        "installtime": {
            "timestamp" : 1660780885,
            "value" : "Aug 18 00:01:25 2022 UTC"
        }
    },
    {
        "name": "xfonts-base",
        "manager": "dpkg",
        "version": "1:1.0.5",
        "size": 7337984,
        "arch": "all",
        "buildtime": {
            "timestamp": 0,
            "value": ""
        },
        "installtime": {
            "timestamp": 0,
            "value": ""
        }
    }
]
```

17 Valores de retorno para net.dns.get

Visão geral

Esta seção fornece detalhes sobre o valor de retorno para o item `net.dns.get` do Zabbix agent 2.

Detalhes

A saída deste item é um objeto contendo informações do registro DNS com base nos parâmetros fornecidos na chave do item.

Por exemplo, o item `net.dns.get[,example.com]` pode retornar o seguinte JSON de uma consulta recusada:

```
{
    "flags": [
        "RA"
    ],
    "query_time": "0.019030",
    "question_section": [
        {
            "qclass": "IN",
            "qname": "example.com.",
            "qtype": "SOA"
        }
    ],
    "response_code": "REFUSED",
    "zbx_error_code": 0
}
```

Ao especificar o endereço IP do servidor DNS, o item `net.dns.get[192.0.2.0,example.com]` pode retornar o seguinte JSON:

```
{
    "answer_section": [
        {
            "class": "IN",
            "name": "example.com.",
            "rdata": {
                "expire": 1209600,
                "mbox": "noc.dns.example.org.",
                "minttl": 3600,
                "ns": "ns.example.org.",
                "refresh": 7200,
                "retry": 3600,
```

```
                "serial": 2022091378
            },
            "rdlength": 44,
            "ttl": 1205,
            "type": "SOA"
        }
    ],
    "flags": [
        "RA"
    ],
    "query_time": "0.029556",
    "question_section": [
        {
            "qclass": "IN",
            "qname": "example.com.",
            "qtype": "SOA"
        }
    ],
    "response_code": "NOERROR",
    "zbx_error_code": 0
}
```


Se houver um problema de conexão, o item `net.dns.get[192.0.2.0,example.com]` pode retornar um erro:


```
{
    "zbx_error_code": -1,
    "zbx_error_msg": "Communication error: read udp 192.0.2.0:12345->192.0.2.0:53: i/o timeout"
}
```


Os seguintes tipos de códigos de erro são possíveis:

Cenário	”zbx_error_code”	”zbx_error_msg”
Sem erros e a resposta DNS foi recebida e analisada.	0	
DNS está fora do ar.	-1	”Communication error”
Ocorre erro durante a análise do JSON	-2	”Received unexpected response”

Com parâmetros adicionais, o item `net.dns.get[192.0.2.0,example.com,ANY,5,5,tcp,"cdflag,rdflag,dnssec,nsid,edns0,a` pode retornar o seguinte JSON:


```
{
    "additional_section": [
        {
            "extended_rcode": 32768,
            "name": ".",
            "rdata": {
                "options": [
                {
                    "code": 0,
                    "nsid": "67 70 64 6e 73 2d 6c 70 70"
                }
                ]
        },
            "rdlength": 13,
            "type": "OPT",
            "udp_payload": 512
        }
    ],
    "answer_section": [
        {
            "class": "IN",
            "name": "example.com.",
            "rdata": {
                "a": "192.0.2.0"
```

```
            },
            "rdlength": 4,
            "ttl": 19308,
            "type": "A"
        },
        {
            "class": "IN",
            "name": "example.com.",
            "rdata": {
                "algorithm": 13,
                "expiration": 1704715951,
                "inception": 1702910624,
                "key_tag": 21021,
                "labels": 2,
                "orig_ttl": 86400,
                "signature": "HVBOBcJJQy0S08J3f8kviPj8UkEUj7wmyiMyQqPSWgQIY9SCEJ5plq6KuxJmtAek1txZWXDo+6tp
                "signer_name": "example.com.",
                "type_covered": "A"
            },
            "rdlength": 95,
            "ttl": 19308,
            "type": "RRSIG"
        }
    ],
    "flags": [
        "RD",
        "RA",
        "AD",
        "CD"
    ],
    "query_time": "0.058221",
    "question_section": [
        {
            "qclass": "IN",
            "qname": "example.com.",
            "qtype": "ANY"
        }
    ],
    "response_code": "NOERROR",
    "zbx_error_code": 0
}
```


Veja também

Para mais informações sobre registros DNS, consulte:

- Domain Names - Implementation and Specification
- Domain Name System (DNS) Parameters

18 Notas sobre os itens system.cpu.util no Windows

O item system.cpu.util fornece a porcentagem de utilização da CPU.

Quando um processo coletor é iniciado no agent Zabbix para Windows, um buffer para N items para N processadores lógicos (threads) é inicializado. Os valores são periodicamente atualizados no buffer usando contadores de desempenho do Windows. Esses valores são reportados quando `system.cpu.util[n]` é solicitado, onde 0 <= n < N.

Mais um item no buffer é usado para `system.cpu.util[total]`.

O Zabbix seleciona automaticamente diferentes contadores de desempenho para suportar tanto sistemas NUMA quanto versões antigas do Windows sem suporte a grupos de processadores.

Se a contagem de processadores lógicos (threads) for menor ou igual a 64 e o número de grupos de processadores for igual a 1, então:

```
N é a contagem de processadores lógicos (threads).

\Processor(_Total)\% Processor Time
\Processor(0)\% Processor Time
\Processor(1)\% Processor Time
\Processor(2)\% Processor Time
...
\Processor(N-1)\% Processor Time
```

Caso contrário:

```
G é a contagem de grupos de processadores e N é a contagem de processadores lógicos (threads) no grupo.

\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
```

5 Funções suportadas

Clique no respectivo grupo de funções para ver mais detalhes.

Grupo de funções	Funções
Funções agregadas	avg, bucket_percentile, count, histogram_quantile, item_count, kurtosis, mad, max, min, skewness, stddevpop, stddevsamp, sum, sumofsquares, varpop, varsamp
Funções foreach	avg_foreach, bucket_rate_foreach, count_foreach, exists_foreach, last_foreach, max_foreach, min_foreach, sum_foreach
Funções bitwise	bitand, bitlshift, bitnot, bitor, bitrshift, bitxor
Funções de data e hora	date, dayofmonth, dayofweek, now, time
Funções de histórico	change, changecount, count, countunique, find, first, fuzzytime, last, logeventid, logseverity, logsource, monodec, monoinc, nodata, percentile, rate
Funções de tendência	baselinedev, baselinewma, trendavg, trendcount, trendmax, trendmin, trendstl, trendsum
Funções matemáticas	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
Funções de operador	between, in

Grupo de funções	Funções
Funções de pre-visão	forecast, timeleft
Funções de string	ascii, bitlength, bytelength, char, concat, insert, jsonpath, left, length, ltrim, mid, repeat, replace, right, rtrim, trim, xmlxpath

Exceto quando indicado de outra forma, essas funções são suportadas em:

- Expressões de trigger
- Items calculados
- Macros de expressão

As funções foreach são suportadas apenas para cálculos agregados.

1 Funções agregadas

Exceto quando indicado de outra forma, todas as funções listadas aqui são suportadas em:

- Expressões de trigger
- Itens calculados
- Macros de expressão

As funções agregadas podem trabalhar com:

- histórico de items, por exemplo, `min(/host/key,1h)`
- funções foreach como o único parâmetro, por exemplo, `min(last_foreach(/*/key))` (apenas em itens calculados; não pode ser usado em triggers)

As funções são listadas sem informações adicionais. Clique na função para ver os detalhes completos.

Função	Descrição
avg	O valor médio de um item dentro do período de avaliação definido.
bucket_percentile	Calcula o percentil a partir dos buckets de um histograma.
count	A contagem de valores em um array retornado por uma função foreach.
histogram_quantile	Calcula o φ-quantil a partir dos buckets de um histograma.
item_count	A contagem de itens existentes na configuração que correspondem aos critérios de filtro.
kurtosis	A "caudacidade" da distribuição de probabilidade nos valores coletados dentro do período de avaliação definido.
mad	O desvio absoluto da mediana nos valores coletados dentro do período de avaliação definido.
max	O maior valor de um item dentro do período de avaliação definido.
min	O menor valor de um item dentro do período de avaliação definido.
skewness	A assimetria da distribuição de probabilidade nos valores coletados dentro do período de avaliação definido.
stddevpop	O desvio padrão da população nos valores coletados dentro do período de avaliação definido.
stddevsamp	O desvio padrão da amostra nos valores coletados dentro do período de avaliação definido.
sum	A soma dos valores coletados dentro do período de avaliação definido.
sumofsquares	A soma dos quadrados nos valores coletados dentro do período de avaliação definido.
varpop	A variância da população dos valores coletados dentro do período de avaliação definido.
varsamp	A variância da amostra dos valores coletados dentro do período de avaliação definido.

Parâmetros comuns

- `/host/key` é um parâmetro obrigatório comum para as funções que referenciam o histórico do item do host
- `(sec|#num)<:deslocamento de tempo>` é um parâmetro comum para as funções que referenciam o histórico do item do host, onde:
  - **sec** - período de avaliação máximo em segundos (sufixos de tempo podem ser usados), ou
  - **#num** - intervalo de avaliação máximo nos últimos valores coletados (se precedido por um sinal de hash)

- **deslocamento de tempo** (opcional) permite mover o ponto de avaliação para trás no tempo. Veja mais detalhes sobre como especificar o deslocamento de tempo.

Detalhes da função

Algumas observações gerais sobre os parâmetros da função:

- Os parâmetros da função são separados por vírgula
- Parâmetros opcionais da função (ou partes do parâmetro) são indicados por < >
- Parâmetros específicos da função são descritos com cada função
- Os parâmetros `/host/key` e `(sec|#num)<:time shift>` nunca devem ser colocados entre aspas

avg(/host/key,(sec|#num)<:time shift>)

O valor médio de um item dentro do período de avaliação definido.
 Tipos de valores suportados: *Float*, *Integer*.
 Funções foreach suportadas: *avg_foreach*, *count_foreach*, *exists_foreach*, *last_foreach*, *max_foreach*, *min_foreach*, *sum_foreach*.

Parâmetros: veja parâmetros comuns.

O deslocamento de tempo é útil quando há necessidade de comparar o valor médio atual com o valor médio de algum tempo atrás.

Exemplos:

```
avg(/host/key,1h) #o valor médio da última hora até agora
avg(/host/key,1h:now-1d) #o valor médio de uma hora de 25 horas atrás até 24 horas atrás a partir de agora
avg(/host/key,#5) #o valor médio dos cinco valores mais recentes
avg(/host/key,#5:now-1d) #o valor médio dos cinco valores mais recentes, excluindo os valores recebidos na
```

bucket_percentile(filtro de item,período de tempo,porcentagem)

Calcula o percentil a partir dos buckets de um histograma.

Parâmetros:

- **filtro de item** - veja filtro de item;
- **período de tempo** - veja período de tempo;
- **porcentagem** - porcentagem (0-100).

Comentários:

- Suportado apenas em items calculados;
- Esta função é um alias para `histogram_quantile(porcentagem/100, bucket_rate_foreach(filtro de item, período de tempo, 1))`.

count(func_foreach(item filter,<período de tempo>),<operador>,<padrão>)

A contagem de valores em um array retornado por uma função foreach.
 Funções foreach suportadas: *avg_foreach*, *count_foreach*, *exists_foreach*, *last_foreach*, *max_foreach*, *min_foreach*, *sum_foreach*.

Parâmetros:

- **func_foreach** - função foreach para a qual o número de valores retornados deve ser contado. Veja funções foreach para detalhes. Observe que count_foreach e bucket_rate_foreach suportam parâmetros adicionais.
- **item filter** - veja filtro de item;
- **período de tempo** - veja período de tempo;
- **operador** (deve estar entre aspas duplas). `operadores` suportados:
eq - igual
ne - diferente
gt - maior
ge - maior ou igual
lt - menor
le - menor ou igual
like - corresponde se contiver o padrão (case-sensitive)
bitand - AND bit a bit
regexp - correspondência case-sensitive da expressão regular fornecida em `padrão`
iregexp - correspondência case-insensitive da expressão regular fornecida em `padrão`
- **padrão** - o padrão necessário (argumentos de string devem estar entre aspas duplas); suportado se *operador* for especificado no terceiro parâmetro.

Comentários:

- Usar **count()** com uma função foreach relacionada ao histórico (max_foreach, avg_foreach, etc.) pode ter implicações de desempenho, enquanto usar **exists_foreach()**, que funciona apenas com dados de configuração, não terá esse efeito.
- Os parâmetros opcionais *operador* ou *padrão* não podem ser deixados vazios após uma vírgula, apenas totalmente omitidos.
- Com *bitand* como terceiro parâmetro, o quarto parâmetro `padrão` pode ser especificado como dois números, separados por '/': **numero_para_comparar/mascara**. count() calcula o "AND bit a bit" do valor e da *máscara* e compara o resultado com *numero_para_comparar*. Se o resultado do "AND bit a bit" for igual a *numero_para_comparar*, o valor é contado.
Se *numero_para_comparar* e *máscara* forem iguais, apenas a *máscara* precisa ser especificada (sem '/').

- Com *regexp* ou *iregexp* como terceiro parâmetro, o quarto parâmetro padrão pode ser uma expressão regular comum ou global (começando com '@'). No caso de expressões regulares globais, a sensibilidade a maiúsculas e minúsculas é herdada das configurações da expressão regular global. Para fins de correspondência regexp, valores float sempre serão representados com 4 dígitos decimais após o '.'. Observe também que, para números grandes, a diferença na representação decimal (armazenada no banco de dados) e binária (usada pelo Zabbix server) pode afetar o 4º dígito decimal.

Exemplos:

```
count(max_foreach(/*/net.if.in[*],1h)) #o número de items net.if.in que receberam dados na última hora até
count(last_foreach(/*/vfs.fs.size[*,pused]),"gt",95) #o número de sistemas de arquivos com mais de 95% do
```

histogram_quantile(quantile,bucket1,value1,bucket2,value2,...)

Calcula o φ-quantil a partir dos buckets de um histograma.
 Função foreach suportada: *bucket_rate_foreach*.

Parâmetros:

- **quantile** - 0 ≤ φ ≤ 1;
- **bucketN, valueN** - pares de parâmetros inseridos manualmente (>=2) ou a resposta de bucket_rate_foreach.

Comentários:

- Suportado apenas em items calculados;
- Corresponde funcionalmente ao 'histogram_quantile' do PromQL;
- Retorna -1 se os valores do último bucket 'Infinity' ("+*inf*") forem iguais a 0.

Exemplos:

```
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)

A contagem de items existentes na configuração que correspondem aos critérios do filtro.
 Tipo de valor suportado: *Integer*.

Parâmetro:

- **item filter** - critérios para seleção de item, permite referência por grupo de hosts, host, chave do item e tags. Caracteres curinga são suportados. Veja item filter para mais detalhes.

Comentários:

- Suportado apenas em items calculados;
- Funciona como um alias para a função *count(exists_foreach(item_filter))*.

Exemplos:

```
item_count(/*/agent.ping?[group="Host group 1"]) #o número de hosts com o item *agent.ping* no "Host group
```

kurtosis(/host/key,(sec|#num)<:time shift>)

A "cauda" da distribuição de probabilidade nos valores coletados dentro do período de avaliação definido. Veja também: Kurtosis.
 Tipos de valores suportados: *Float*, *Integer*.
 Função foreach suportada: *last_foreach*.

Parâmetros: veja parâmetros comuns.

Exemplo:

```
kurtosis(/host/key,1h) #kurtosis para a última hora até agora
```

mad(/host/key,(sec|#num)<:time shift>)

O desvio absoluto da mediana nos valores coletados dentro do período de avaliação definido. Veja também: Desvio absoluto da mediana.
 Tipos de valores suportados: *Float*, *Integer*.
 Função foreach suportada: *last_foreach*.

Parâmetros: veja parâmetros comuns.

Exemplo:

```
mad(/host/key,1h) #desvio absoluto da mediana para a última hora até agora
```

max(/host/key,(sec|#num)<:time shift>)

O maior valor de um item dentro do período de avaliação definido.
 Tipos de valores suportados: *Float*, *Integer*.
 Funções foreach suportadas: *avg_foreach*, *count_foreach*, *exists_foreach*, *last_foreach*, *max_foreach*, *min_foreach*, *sum_foreach*.

Parâmetros: veja parâmetros comuns.

Exemplo:

```
max(/host/key,1h) - min(/host/key,1h) #calcula a diferença entre os valores máximo e mínimo dentro da últi
```

min(/host/key,(sec|#num)<:time shift>)

O menor valor de um item dentro do período de avaliação definido.
 Tipos de valores suportados: *Float*, *Integer*.
 Funções foreach suportadas: *avg_foreach*, *count_foreach*, *exists_foreach*, *last_foreach*, *max_foreach*, *min_foreach*, *sum_foreach*.

Parâmetros: veja parâmetros comuns.

Exemplo:

```
max(/host/key,1h) - min(/host/key,1h) #calcula a diferença entre os valores máximo e mínimo dentro da últi
```

skewness(/host/key,(sec|#num)<:time shift>)

A assimetria da distribuição de probabilidade nos valores coletados dentro do período de avaliação definido. Veja também: Assimetria.
 Tipos de valores suportados: *Float*, *Integer*.
 Função foreach suportada: *last_foreach*.

Parâmetros: veja parâmetros comuns.

Exemplo:

```
skewness(/host/key,1h) #a assimetria para a última hora até agora
```

stddevpop(/host/key,(sec|#num)<:time shift>)

O desvio padrão da população nos valores coletados dentro do período de avaliação definido. Veja também: Desvio padrão.
 Tipos de valores suportados: *Float*, *Integer*.
 Função foreach suportada: *last_foreach*.

Parâmetros: veja parâmetros comuns.

Exemplo:

```
stddevpop(/host/key,1h) #o desvio padrão da população para a última hora até agora
```

stddevsamp(/host/key,(sec|#num)<:time shift>)

O desvio padrão amostral nos valores coletados dentro do período de avaliação definido. Veja também: Desvio padrão.
 Tipos de valores suportados: *Float*, *Integer*.
 Função foreach suportada: *last_foreach*.

Parâmetros: veja parâmetros comuns.

Pelo menos dois valores de dados são necessários para que esta função funcione.

Exemplo:

```
stddevsamp(/host/key,1h) #o desvio padrão amostral para a última hora até agora
```

sum(/host/key,(sec|#num)<:time shift>)

A soma dos valores coletados dentro do período de avaliação definido.
 Tipos de valores suportados: *Float*, *Integer*.
 Funções foreach suportadas: *avg_foreach*, *count_foreach*, *exists_foreach*, *last_foreach*, *max_foreach*, *min_foreach*, *sum_foreach*.

Parâmetros: veja parâmetros comuns.

Exemplo:

```
sum(/host/key,1h) #a soma dos valores da última hora até agora
```

sumofsquares(/host/key,(sec|#num)<:time shift>)

A soma dos quadrados dos valores coletados dentro do período de avaliação definido.
 Tipos de valores suportados: *Float*, *Integer*.
 Função foreach suportada: *last_foreach*.

Parâmetros: veja parâmetros comuns.

Exemplo:

```
sumofsquares(/host/key,1h) #a soma dos quadrados da última hora até agora
```

varpop(/host/key,(sec|#num)<:time shift>)

A variância populacional dos valores coletados dentro do período de avaliação definido. Veja também: Variance.
 Tipos de valores suportados: *Float*, *Integer*.
 Função foreach suportada: *last_foreach*.

Parâmetros: veja parâmetros comuns.

Exemplo:

```
varpop(/host/key,1h) #a variância populacional para a última hora até agora
```

varsamp(/host/key,(sec|#num)<:time shift>)

A variância amostral dos valores coletados dentro do período de avaliação definido. Veja também: Variance.
 Tipos de valores suportados: *Float*, *Integer*.
 Função foreach suportada: *last_foreach*.

Parâmetros: veja parâmetros comuns.

Pelo menos dois valores de dados são necessários para que esta função funcione.

Exemplo:

```
varsamp(/host/key,1h) #a variância amostral para a última hora até agora
```

Veja todas as funções suportadas.

1 Funções Foreach

Visão geral

As funções foreach são usadas em cálculos agregados para retornar um valor agregado para cada item que é selecionado pelo **filtro de item** utilizado. Um array de valores é retornado.

> **Attention:**
> As funções foreach são suportadas apenas em Itens calculados como parte de fórmulas de cálculo agregado. Elas não podem ser invocadas em chaves de item simples ou expressões de trigger fora desse contexto.

Por exemplo, a função *avg_foreach* retornará um array de valores, onde cada valor é o valor médio do histórico do item selecionado, durante o intervalo de tempo especificado.

O filtro de item faz parte da sintaxe usada pelas funções foreach. O uso de curingas é suportado no filtro de item, assim os itens necessários podem ser selecionados de forma bastante flexível.

Funções suportadas

Função	Descrição
avg_foreach	Retorna o valor médio para cada item.
bucket_rate_foreach	Retorna pares (limite superior do bucket, valor da taxa) adequados para uso na função histogram_quantile(), onde "limite superior do bucket" é o valor do parâmetro da chave do item definido pelo <número do parâmetro> parâmetro.
count_foreach	Retorna o número de valores para cada item.
exists_foreach	Retorna '1' para cada item habilitado.
last_foreach	Retorna o último valor para cada item.
max_foreach	Retorna o valor máximo para cada item.
min_foreach	Retorna o valor mínimo para cada item.
sum_foreach	Retorna a soma dos valores para cada item.

Sintaxe da função

As funções foreach suportam dois parâmetros comuns: `filtro de item` (veja detalhes abaixo) e `período de tempo`:

```
foreach_function(filtro de item,período de tempo)
```

Por exemplo:

```
avg_foreach(/*/mysql.qps?[group="MySQL Servers"],5m)
```

retornará a média de cinco minutos de cada item 'mysql.qps' no grupo de servidores MySQL.

Observe que algumas funções suportam parâmetros adicionais.

Sintaxe do filtro de item

O filtro de item:

```
/host/key[parameters]?[conditions]
```

consiste em quatro partes, onde:

- host - nome do host
- key - chave do item (sem parâmetros)
- parameters - parâmetros da chave do item

- conditions - condições baseadas em grupo de host e/ou tag de item (como expressão)

Espaços são permitidos apenas dentro da expressão de condições.

Uso de curinga

- O curinga pode ser usado para substituir o nome do host, a chave do item ou um parâmetro individual da chave do item.
- Ou o host ou a chave do item devem ser especificados sem curinga. Portanto, `/host/*` e `/*/key` são filtros válidos, mas `/*/*` é inválido.
- O curinga não pode ser usado para *parte* do nome do host, chave do item ou parâmetro da chave do item.
- O curinga não corresponde a mais de um parâmetro da chave do item. Portanto, um curinga deve ser especificado para cada parâmetro separadamente (ou seja, `key[abc,*,*]`).

Expressão de condições

A expressão de condições suporta:

- operandos:
  - group - grupo de host
  - tag - tag de item
  - "<text>" - constante de string, com o caractere de escape \ para escapar " e \
- operadores de comparação de string sensíveis a maiúsculas e minúsculas: =, <>
- operadores lógicos: `and`, `or`, `not`
- agrupamento com parênteses: ( )

A citação de constantes de string é obrigatória. Apenas a comparação de string completa e sensível a maiúsculas e minúsculas é suportada.

Warning:
Ao especificar tags no filtro (ou seja, `tag="tagname:value"`), o dois-pontos ":" é usado como delimitador. Tudo após ele é considerado o valor da tag. Portanto, atualmente não é suportado especificar um nome de tag contendo ":".

Exemplos

Um filtro complexo pode ser usado, referenciando a chave do item, grupo de host e tags, conforme ilustrado pelos exemplos:

Exemplo de sintaxe	Descrição
`/host/key[abc,*]`	Corresponde a itens semelhantes neste host.
`/*/key`	Corresponde ao mesmo item de qualquer host.
`/*/key?[group="ABC" and tag="tagname:value"]`	Corresponde ao mesmo item de qualquer host do grupo ABC que tenha tags 'tagname:value'.
`/*/key[a,*,c]?[(group="ABC" and tag="Tag1") or (group="DEF" and (tag="Tag2" or tag="Tag3:value"))]`	Corresponde a itens semelhantes de qualquer host do grupo ABC ou DEF com as respectivas tags.

Todos os itens referenciados devem existir e coletar dados. Apenas itens habilitados em hosts habilitados são incluídos nos cálculos. Itens em estado não suportado não são incluídos.

Attention:
Se a chave do item de um item referenciado for alterada, o filtro deverá ser atualizado manualmente.

Especificar um grupo de host pai inclui o grupo pai e todos os grupos de host aninhados com seus itens.

Período de tempo

O **segundo** parâmetro permite especificar o período de tempo para agregação. O período de tempo só pode ser expresso como tempo, a quantidade de valores (prefixada com **#**) não é suportada.

Unidades de tempo suportadas podem ser usadas neste parâmetro para conveniência, por exemplo, '5m' (cinco minutos) ao invés de '300s' (300 segundos) ou '1d' (um dia) ao invés de '86400' (86400 segundos).

Para a função *last_foreach*, o período de tempo é um parâmetro opcional (suportado desde o Zabbix 7.0), que pode ser omitido:

```
last_foreach(/*/key?[group="host group"])
```

O período de tempo não é suportado com a função *exists_foreach*.

Parâmetros adicionais

bucket_rate_foreach

Um terceiro parâmetro opcional é suportado pela função *bucket_rate_foreach*:

`bucket_rate_foreach(item filter,time period,<parameter number>)`

onde <parameter number> é a posição do valor ”bucket” na chave do item. Por exemplo, se o valor ”bucket” em myItem[aaa,0.2] for ’0.2’, então sua posição é 2.

O valor padrão de <parameter number> é ’1’.

count_foreach

Terceiro e quarto parâmetros opcionais são suportados pela função *count_foreach*:

`count_foreach(item filter,time period,<operator>,<pattern>)`

Onde:

- **operator** é o operador condicional para valores de item (deve estar entre aspas duplas). `operators` suportados:
eq - igual
ne - diferente
gt - maior
ge - maior ou igual
lt - menor
le - menor ou igual
like - corresponde se contiver o padrão (case-sensitive)
bitand - AND bit a bit
regexp - correspondência case-sensitive da expressão regular fornecida em `pattern`
iregexp - correspondência case-insensitive da expressão regular fornecida em `pattern`
- **pattern** é o padrão necessário (argumentos de string devem estar entre aspas duplas); suportado se *operator* for especificado no terceiro parâmetro.

Comentários:

- Os parâmetros opcionais *operator* ou *pattern* não podem ser deixados vazios após uma vírgula, apenas totalmente omitidos.
- Com *bitand* como terceiro parâmetro, o quarto parâmetro `pattern` pode ser especificado como dois números, separados por ’/’: **number_to_compare_with/mask**. count_foreach() calcula o ”AND bit a bit” do valor e da *mask* e compara o resultado com *number_to_compare_with*. Se o resultado do ”AND bit a bit” for igual a *number_to_compare_with*, o valor é contado.
Se *number_to_compare_with* e *mask* forem iguais, apenas a *mask* precisa ser especificada (sem ’/’).
- Com *regexp* ou *iregexp* como terceiro parâmetro, o quarto parâmetro `pattern` pode ser uma expressão regular comum ou global (começando com ’@’). No caso de expressões regulares globais, a sensibilidade a maiúsculas e minúsculas é herdada das configurações da expressão regular global. Para fins de correspondência regexp, valores float sempre serão representados com 4 dígitos decimais após o ’.’. Observe também que, para números grandes, a diferença na representação decimal (armazenada no banco de dados) e binária (usada pelo Zabbix server) pode afetar o 4º dígito decimal.

Veja cálculos agregados para mais detalhes e exemplos sobre o uso de funções foreach.

Comportamento dependendo da disponibilidade

A tabela a seguir ilustra como cada função se comporta em casos de disponibilidade limitada de host/item e dados de histórico.

Função	Host desabilitado	Host indisponível com dados	Host indisponível sem dados	Item desabilitado	Item não suportado	Erro de recuperação de dados (SQL)
avg_foreach	ignorar	retornar média	ignorar	ignorar	ignorar	ignorar
bucket_rate_foreach	ignorar	retornar bucket rate	ignorar	ignorar	ignorar	ignorar
count_foreach	ignorar	retornar contagem	0	ignorar	ignorar	ignorar
exists_foreach	ignorar	1	1	ignorar	1	n/a
last_foreach	ignorar	retornar último	ignorar	ignorar	ignorar	ignorar
max_foreach	ignorar	retornar máximo	ignorar	ignorar	ignorar	ignorar
min_foreach	ignorar	retornar mínimo	ignorar	ignorar	ignorar	ignorar
sum_foreach	ignorar	retornar soma	ignorar	ignorar	ignorar	ignorar

Se o item for *ignorado*, nada é adicionado à agregação.

2 Funções Bitwise

Todas as funções listadas aqui são suportadas em:

- Expressões de trigger

- Itens calculados
- Macros de expressão

As funções são listadas sem informações adicionais. Clique na função para ver os detalhes completos.

Função	Descrição
bitand	O valor do ”E bit a bit” de um valor de item e máscara.
bitlshift	O deslocamento bit a bit para a esquerda de um valor de item.
bitnot	O valor do ”NÃO bit a bit” de um valor de item.
bitor	O valor do ”OU bit a bit” de um valor de item e máscara.
bitrshift	O deslocamento bit a bit para a direita de um valor de item.
bitxor	O valor do ”OU exclusivo bit a bit” de um valor de item e máscara.

Detalhes da função

Algumas observações gerais sobre os parâmetros da função:

- Os parâmetros da função são separados por vírgula
- Expressões são aceitas como parâmetros
- Parâmetros opcionais da função (ou partes do parâmetro) são indicados por < >

bitand(value,mask)

O valor de ”E bit a bit” de um valor de item e máscara.
 Tipos de valores suportados: *Integer*.

Parâmetros:

- **value** - o valor a ser verificado;
- **mask** (obrigatório) - um inteiro sem sinal de 64 bits (0 - 18446744073709551615).

Embora a comparação seja feita de forma bit a bit, todos os valores devem ser fornecidos e retornados em decimal. Por exemplo, verificar o 3º bit é feito comparando com 4, não 100.

Exemplos:

```
bitand(last(/host/key),12)=8 or bitand(last(/host/key),12)=4 #3º ou 4º bit definido, mas não ambos ao mesm
bitand(last(/host/key),20)=16 #3º bit não definido e 5º bit definido
```

bitlshift(value,bits to shift)

O deslocamento de bits para a esquerda de um valor de item.
 Tipos de valores suportados: *Integer*.

Parâmetros:

- **value** - o valor a ser verificado;
- **bits to shift** (obrigatório) - o número de bits a serem deslocados.

Embora a comparação seja feita de forma bit a bit, todos os valores devem ser fornecidos e retornados em decimal. Por exemplo, a verificação do 3º bit é feita comparando com 4, não com 100.

bitnot(value)

O valor de ”bitwise NOT” de um valor de item.
 Tipos de valores suportados: *Integer*.

Parâmetro:

- **value** - o valor a ser verificado.

Embora a comparação seja feita de forma bit a bit, todos os valores devem ser fornecidos e retornados em decimal. Por exemplo, a verificação do 3º bit é feita comparando com 4, não com 100.

bitor(value,mask)

O valor de ”OU bit a bit” de um valor de item e máscara.
 Tipos de valores suportados: *Integer*.

Parâmetros:

- **value** - o valor a ser verificado;
- **mask** (obrigatório) - um inteiro sem sinal de 64 bits (0 - 18446744073709551615).

Embora a comparação seja feita de forma bit a bit, todos os valores devem ser fornecidos e retornados em decimal. Por exemplo, a verificação do 3º bit é feita comparando com 4, não com 100.

bitrshift(value,bits to shift)

O deslocamento de bits para a direita de um valor de item.
 Tipos de valores suportados: *Integer*.

Parâmetros:

- **value** - o valor a ser verificado;
- **bits to shift** (obrigatório) - o número de bits a serem deslocados.

Embora a comparação seja feita de forma bit a bit, todos os valores devem ser fornecidos e retornados em decimal. Por exemplo, a verificação do 3º bit é feita comparando com 4, não com 100.

bitxor(value,mask)

O valor de "OU exclusivo bit a bit" de um valor de item e máscara.
 Tipos de valores suportados: *Integer*.

Parâmetros:

- **value** - o valor a ser verificado;
- **mask** (obrigatório) - um inteiro sem sinal de 64 bits (0 - 18446744073709551615).

Embora a comparação seja feita de forma bit a bit, todos os valores devem ser fornecidos e retornados em decimal. Por exemplo, a verificação do 3º bit é feita comparando com 4, não com 100.

Veja todas as funções suportadas.

3 Funções de data e hora

Todas as funções listadas aqui são suportadas em:

- Expressões de trigger
- Itens calculados
- Macros de expressão

> **Attention:**
> As funções de data e hora não podem ser usadas sozinhas na expressão; pelo menos uma função de outro grupo, referenciando o item do host, deve ser incluída na expressão (exceto a função nodata()). Para informações detalhadas sobre como as funções de data e hora funcionam dentro das expressões, consulte Tempo de cálculo.

As funções são listadas sem informações adicionais. Clique na função para ver os detalhes completos.

Função	Descrição
date	A data atual no formato AAAAMMDD.
dayofmonth	O dia do mês no intervalo de 1 a 31.
dayofweek	O dia da semana no intervalo de 1 a 7.
now	O número de segundos desde a Época (00:00:00 UTC, 1 de janeiro de 1970).
time	A hora atual no formato HHMMSS.

Detalhes da função

date

A data atual no formato YYYYMMDD.

Exemplo:

```
date()<20220101
```

dayofmonth

O dia do mês no intervalo de 1 a 31.

Exemplo:

```
dayofmonth()=1
```

dayofweek

O dia da semana no intervalo de 1 a 7 (Seg - 1, Dom - 7).

Exemplo (apenas dias da semana):

```
dayofweek()<6
```

Exemplo (apenas fim de semana):

`dayofweek()>5`

now

O número de segundos desde a Época (00:00:00 UTC, 1 de janeiro de 1970).

Exemplo:

`now()<1640998800`

time

A hora atual no formato HHMMSS.

Exemplo (apenas noturno, 00:00-06:00):

`time()<060000`

Veja todas as funções suportadas.

4 Funções de histórico

Todas as funções listadas aqui são suportadas em:

- Expressões de trigger
- Itens calculados
- Macros de expressão

As funções são listadas sem informações adicionais. Clique na função para ver os detalhes completos.

Função	Descrição
change	A quantidade de diferença entre o valor anterior e o mais recente.
changecount	O número de mudanças entre valores adjacentes dentro do período de avaliação definido.
count	O número de valores dentro do período de avaliação definido.
countunique	O número de valores únicos dentro do período de avaliação definido.
find	Encontra uma correspondência de valor dentro do período de avaliação definido.
first	O primeiro (mais antigo) valor dentro do período de avaliação definido.
fuzzytime	Verifica quanto o tempo do agent passivo difere do tempo do Zabbix server/proxy.
last	O valor mais recente.
logeventid	Verifica se o ID do evento da última entrada de log corresponde a uma expressão regular.
logseverity	A severidade do log da última entrada de log.
logsource	Verifica se a origem do log da última entrada de log corresponde a uma expressão regular.
monodec	Verifica se houve uma diminuição monótona nos valores.
monoinc	Verifica se houve um aumento monótono nos valores.
nodata	Verifica se não há dados recebidos.
percentile	O percentil P de um período, onde P (percentual) é especificado pelo terceiro parâmetro.
rate	A taxa média por segundo do aumento em um contador monótono dentro do período de tempo definido.

Parâmetros comuns

- `/host/key` é um parâmetro obrigatório comum para as funções que referenciam o histórico do item do host
- `(sec|#num)<:deslocamento de tempo>` é um parâmetro comum para as funções que referenciam o histórico do item do host, onde:
  - **sec** - período de avaliação máximo em segundos (sufixos de tempo podem ser usados), ou
  - **#num** - intervalo de avaliação máximo nos últimos valores coletados (se precedido por um símbolo de hash)
  - **deslocamento de tempo** (opcional) permite mover o ponto de avaliação para trás no tempo. Veja mais detalhes sobre como especificar o deslocamento de tempo.

Detalhes da função

Algumas observações gerais sobre os parâmetros da função:

- Os parâmetros da função são separados por uma vírgula
- Parâmetros opcionais da função (ou partes do parâmetro) são indicados por < >
- Parâmetros específicos da função são descritos com cada função
- Os parâmetros `/host/key` e `(sec|#num)<:deslocamento de tempo>` nunca devem ser colocados entre aspas

change(/host/key)

A quantidade de diferença entre o valor anterior e o mais recente.
 Tipos de valores suportados: *Float*, *Integer*, *String*, *Text*, *Log*.
 Para strings retorna: 0 - valores são iguais; 1 - valores diferem.

Parâmetros: veja parâmetros comuns.

Comentários:

- A diferença numérica será calculada, conforme visto nestes valores de exemplo recebidos ('valor anterior' e 'valor mais recente' = diferença):
'1' e '5' = +4
'3' e '1' = −2
'0' e '-2.5' = −2.5.
- Veja também: abs para comparação.

Exemplos:

```
change(/host/system.uptime)<0 #a alteração no tempo de atividade do sistema foi negativa desde o último va
change(/host/system.cpu.load[all,avg1])>2 #a carga da CPU (por um minuto) aumentou em mais de 2 desde o úl
change(/host/vfs.fs.size[/,free])<-1G #o espaço livre em disco caiu mais de 1 GB entre as verificações
```

changecount(/host/key,(sec|#num)<:time shift>,<mode>)

O número de alterações entre valores adjacentes dentro do período de avaliação definido.
 Tipos de valor suportados: *Float*, *Integer*, *String*, *Text*, *Log*.

Parâmetros:

- Veja parâmetros comuns
- **mode** (deve ser entre aspas duplas) - valores possíveis: *all* - conta todas as alterações (padrão); *dec* - conta diminuições; *inc* - conta aumentos

Para tipos de valor não numéricos, o parâmetro *mode* é ignorado.

Exemplos:

```
changecount(/host/icmpping,10m)>5 #o status do ping mudou mais de 5 vezes em 10 minutos
changecount(/host/vfs.file.contents["/sys/class/net/eth0/operstate"],1h)>5 #o estado operacional da eth0 m
changecount(/host/proc.num[httpd],15m)>10 #o número de processos httpd mudou mais de 10 vezes em 15 minuto
changecount(/host/key,#10,"inc") #o número de aumentos de valor (em relação ao valor adjacente) entre os ú
changecount(/host/key,24h,"dec") #o número de diminuições de valor (em relação ao valor adjacente) nas últ
```

count(/host/key,(sec|#num)<:time shift>,<operator>,<pattern>)

O número de valores dentro do período de avaliação definido.
 Tipos de valores suportados: *Float*, *Integer*, *String*, *Text*, *Log*.

Parâmetros:

- Veja parâmetros comuns
- **operator** (deve ser colocado entre aspas duplas) `operators` suportados:
eq - igual (padrão para integer, float)
ne - diferente
gt - maior
ge - maior ou igual
lt - menor
le - menor ou igual
like (padrão para string, text, log) - corresponde se contiver o padrão (case-sensitive)
bitand - AND bit a bit
regexp - correspondência case-sensitive da expressão regular fornecida em `pattern`
iregexp - correspondência case-insensitive da expressão regular fornecida em `pattern`
- **pattern** - o padrão necessário (argumentos de string devem ser colocados entre aspas duplas)

Comentários:

- Itens do tipo float correspondem com a precisão de 2.22e-16.
- *like* não é suportado como operador para valores inteiros.
- *like* e *bitand* não são suportados como operadores para valores float.
- Para valores string, text e log, apenas os operadores *eq*, *ne*, *like*, *regexp* e *iregexp* são suportados.
- Com *bitand* como operador, o quarto parâmetro `pattern` pode ser especificado como dois números, separados por '/': **number_to_compare_with/mask**. count() calcula o "AND bit a bit" do valor e da *mask* e compara o resultado com *number_to_compare_with*. Se o resultado do "AND bit a bit" for igual a *number_to_compare_with*, o valor é contado.
Se *number_to_compare_with* e *mask* forem iguais, apenas a *mask* precisa ser especificada (sem '/').
- Com *regexp* ou *iregexp* como operador, o quarto parâmetro `pattern` pode ser uma expressão regular comum ou global (começando com '@'). No caso de expressões regulares globais, a sensibilidade a maiúsculas e minúsculas é herdada das configurações da expressão regular global. Para fins de correspondência regexp, valores float sempre serão representados com 4 dígitos decimais após o '.'. Observe também que, para números grandes, a diferença na representação decimal (armazenada no banco de dados) e binária (usada pelo Zabbix server) pode afetar o 4º dígito decimal.

Exemplos:

```
count(/host/icmpping,30m,,"0")>5 #ping falhou mais de 5 vezes em 30 minutos
count(/host/key,10m,"like","error") #o número de valores nos últimos 10 minutos até agora que contêm 'erro
count(/host/key,10m,,12) #o número de valores nos últimos 10 minutos até agora que são iguais a '12'
count(/host/key,10m,"gt",12) #o número de valores nos últimos 10 minutos até agora que são maiores que '12
count(/host/key,#10,"gt",12) #o número de valores entre os últimos 10 valores até agora que são maiores qu
count(/host/key,10m:now-1d,"gt",12) #o número de valores entre 24 horas e 10 minutos e 24 horas atrás a pa
count(/host/key,10m,"bitand","6/7") #o número de valores nos últimos 10 minutos até agora que têm '110' (e
count(/host/key,10m:now-1d) #o número de valores entre 24 horas e 10 minutos e 24 horas atrás a partir de
```

countunique(/host/key,(sec|#num)<:time shift>,<operator>,<pattern>)

O número de valores únicos dentro do período de avaliação definido.
 Tipos de valores suportados: *Float*, *Integer*, *String*, *Text*, *Log*.

Parâmetros:

- Veja parâmetros comuns
- **operator** (deve ser colocado entre aspas duplas). `operators` suportados:
eq - igual (padrão para integer, float)
ne - diferente
gt - maior
ge - maior ou igual
lt - menor
le - menor ou igual
like (padrão para string, text, log) - corresponde se contiver o padrão (case-sensitive)
bitand - AND bit a bit
regexp - correspondência case-sensitive da expressão regular fornecida em `pattern`
iregexp - correspondência case-insensitive da expressão regular fornecida em `pattern`
- **pattern** - o padrão necessário (argumentos de string devem ser colocados entre aspas duplas)

Comentários:

- Itens do tipo float correspondem com a precisão de 2.22e-16.
- *like* não é suportado como operador para valores integer.
- *like* e *bitand* não são suportados como operadores para valores float.
- Para valores string, text e log, apenas os operadores *eq*, *ne*, *like*, *regexp* e *iregexp* são suportados.
- Com *bitand* como operador, o quarto parâmetro `pattern` pode ser especificado como dois números, separados por '/': **number_to_compare_with/mask**. countunique() calcula o "AND bit a bit" do valor e da *mask* e compara o resultado com *number_to_compare_with*. Se o resultado do "AND bit a bit" for igual a *number_to_compare_with*, o valor é contado.
Se *number_to_compare_with* e *mask* forem iguais, apenas a *mask* precisa ser especificada (sem '/').
- Com *regexp* ou *iregexp* como operador, o quarto parâmetro `pattern` pode ser uma expressão regular comum ou global (começando com '@'). No caso de expressões regulares globais, a sensibilidade a maiúsculas e minúsculas é herdada das configurações da expressão regular global. Para fins de correspondência regexp, valores float sempre serão representados com 4 dígitos decimais após o '.'. Observe também que, para números grandes, a diferença na representação decimal (armazenada no banco de dados) e binária (usada pelo Zabbix server) pode afetar o 4º dígito decimal.

Exemplos:

```
countunique(/host/key,10m) #o número de valores únicos nos últimos 10 minutos até agora
countunique(/host/key,10m,"like","error") #o número de valores únicos nos últimos 10 minutos até agora que
countunique(/host/key,10m,,12) #o número de valores únicos nos últimos 10 minutos até agora que são iguais
countunique(/host/key,10m,"gt",12) #o número de valores únicos nos últimos 10 minutos até agora que são ma
countunique(/host/key,#10,"gt",12) #o número de valores únicos nos últimos 10 valores até agora que são ma
countunique(/host/key,10m:now-1d,"gt",12) #o número de valores únicos entre 24 horas e 10 minutos e 24 hor
countunique(/host/key,10m,"bitand","6/7") #o número de valores únicos nos últimos 10 minutos até agora que
countunique(/host/key,10m:now-1d) #o número de valores únicos entre 24 horas e 10 minutos e 24 horas atrás
```

find(/host/key,(sec|#num)<:time shift>,<operator>,<pattern>)

Encontra uma correspondência de valor dentro do período de avaliação definido.
 Tipos de valor suportados: *Float*, *Integer*, *String*, *Text*, *Log*.
 Retorna: 1 - encontrado; 0 - caso contrário.

Parâmetros:

- Veja parâmetros comuns
- **sec** ou **#num** (opcional) - por padrão, usa o valor mais recente se não especificado
- **operator** (deve ser entre aspas duplas) `operators` suportados:
eq - igual (padrão para integer, float)
ne - diferente
gt - maior
ge - maior ou igual
lt - menor
le - menor ou igual
like (padrão para string, text, log) - corresponde se contiver a string fornecida em `pattern` (case-sensitive)
bitand - AND bit a bit
regexp - correspondência case-sensitive da expressão regular fornecida em `pattern`
iregexp - correspondência case-insensitive da expressão regular fornecida em `pattern`
- **pattern** - o padrão necessário (argumentos de string devem estar entre aspas duplas); expressão regular Perl Compatible Regular Expression (PCRE) se `operator` for *regexp*, *iregexp*

Comentários:

- Se mais de um valor for processado, '1' será retornado se houver pelo menos um valor correspondente.
- *like* não é suportado como operador para valores inteiros.
- *like* e *bitand* não são suportados como operadores para valores float.
- Para valores string, text e log, apenas os operadores *eq*, *ne*, *like*, *regexp* e *iregexp* são suportados.
- Com *regexp* ou *iregexp* como operador, o quarto parâmetro `pattern` pode ser uma expressão regular comum ou global (começando com '@'). No caso de expressões regulares globais, a sensibilidade a maiúsculas e minúsculas é herdada das configurações da expressão regular global.

Exemplos:

```
find(/host/key,10m,"like","error") #encontra um valor que contenha 'error' nos últimos 10 minutos até agor
find(/host/agent.version,,"like","beta")=1 #O agent Zabbix está em versão beta, deve ser atualizado
find(/host/log[/var/log/nginx/access.log],,"regexp"," 500 ")=1 #foi encontrado erro interno do servidor we
```

first(/host/key,sec<:time shift>)

O primeiro (mais antigo) valor dentro do período de avaliação definido.
 Tipos de valores suportados: *Float*, *Integer*, *String*, *Text*, *Log*.

Parâmetros:

- Veja parâmetros comuns

Veja também last().

Exemplo:

```
first(/host/key,1h) #recupera o valor mais antigo dentro da última hora até agora
```

fuzzytime(/host/key,sec)

Verifica quanto o tempo do agent passivo difere do tempo do Zabbix server/proxy.
 Tipos de valores suportados: *Float*, *Integer*.
 Retorna: 1 - a diferença entre o valor do item passivo (como timestamp) e o timestamp do Zabbix server/proxy (o relógio da coleta do valor) é menor ou igual a *sec* segundos; 0 - caso contrário.

Parâmetros:

- Veja parâmetros comuns.

Comentários:

- Normalmente usado com o item 'system.localtime' para verificar se o horário local está sincronizado com o horário local do Zabbix server. *Nota* que 'system.localtime' deve ser configurado como um cheque passivo para o Zabbix agent; no Zabbix agent 2 pode ser configurado como um cheque ativo.
- Também pode ser usado com a chave `vfs.file.time[/path/file,modify]` para verificar se o arquivo não recebeu atualizações por muito tempo.
- Esta função não é recomendada para uso em expressões de trigger complexas (com múltiplos items envolvidos), pois pode causar resultados inesperados (a diferença de tempo será medida com a métrica mais recente), por exemplo, em `fuzzytime(/Host/system.localtime,60s)=0 or last(/Host/trap)<>0`.

Exemplos:

```
fuzzytime(/host/system.localtime,5m)=0 #o horário local do cliente difere do horário do Zabbix server/prox
fuzzytime(/host/system.localtime,5m)=0 and nodata(/host/system.localtime,10m)=0 #o horário local do client
```

last(/host/key,<#num<:time shift»)

O valor mais recente.
 Tipos de valores suportados: *Float*, *Integer*, *String*, *Text*, *Log*.

Parâmetros:

- Veja parâmetros comuns
- **#num** (opcional) - o N-ésimo valor mais recente

Comentários:

- Observe que um período de tempo com hashtag (#N) funciona de forma diferente aqui do que com muitas outras funções. Por exemplo: `last(/host/key)` é sempre igual a `last(/host/key,#1)`; `last(/host/key,#3)` - o terceiro valor mais recente (*não* os três valores mais recentes).
- O Zabbix não garante a ordem exata dos valores se mais de dois valores existirem dentro de um segundo no histórico.
- Veja também first().

Exemplos:

```
last(/host/key) #recupera o último valor
last(/host/key,#2) #recupera o valor anterior
last(/host/key,#1) <> last(/host/key,#2) #os valores atual e anterior são diferentes
```

logeventid(/host/key,<#num<:time shift»,<pattern>)

Verifica se o ID do evento da última entrada de log corresponde a uma expressão regular.
 Tipos de valor suportados: *Log*.
 Retorna: 0 - não corresponde; 1 - corresponde.

Parâmetros:

- Veja parâmetros comuns
- **#num** (opcional) - o N-ésimo valor mais recente
- **pattern** (opcional) - a expressão regular que descreve o padrão necessário, no estilo Perl Compatible Regular Expression (PCRE) (argumentos de string devem estar entre aspas duplas)

Exemplos:

```
logeventid(/host/eventlog[Security],,"4625")=1 #uma entrada de log com ID correspondente a "4625" (autenti
logeventid(/host/eventlog[System],,"6008|41")=1 #uma entrada de log com ID correspondente a "6008" ou "41"
```

logseverity(/host/key,<#num<:time shift»)

Gravidade do log da última entrada de log.
 Tipos de valor suportados: *Log*.
 Retorna: 0 - gravidade padrão; N - gravidade (inteiro, útil para logs de eventos do Windows: 1 - Informação, 2 - Aviso, 4 - Erro, 7 - Falha de Auditoria, 8 - Sucesso de Auditoria, 9 - Crítico, 10 - Detalhado).

Parâmetros:

- Veja parâmetros comuns
- **#num** (opcional) - o N-ésimo valor mais recente

O Zabbix obtém a gravidade do log a partir do campo **Information** do log de eventos do Windows.

Exemplos:

```
logseverity(/host/log[/var/log/syslog],10m)>3 #uma entrada de log com gravidade acima de "3" encontrada
logseverity(/host/eventlog[System],10m)=4 #uma entrada de log com gravidade igual a "Erro" encontrada
```

logsource(/host/key,<#num<:time shift»,<pattern>)

Verifica se a fonte do último registro de log corresponde a uma expressão regular.
 Tipos de valor suportados: *Log*.
 Retorna: 0 - não corresponde; 1 - corresponde.

Parâmetros:

- Veja parâmetros comuns
- **#num** (opcional) - o N-ésimo valor mais recente
- **pattern** (opcional) - a expressão regular que descreve o padrão necessário, no estilo Perl Compatible Regular Expression (PCRE) (argumentos de string devem estar entre aspas duplas)

Normalmente usado para logs de eventos do Windows.

Exemplos:

```
logsource(/host/eventlog[Application],,"MSSQLSERVER")=1 #um registro de log com fonte correspondente a "MS
logsource(/host/eventlog[System],,"Service Control Manager")=1 #um registro de log com fonte correspondent
logsource(/host/eventlog[System],,"Service Control Manager")=1 and logeventid(/host/eventlog[System],,"703
```

monodec(/host/key,(sec|#num)<:time shift>,<mode>)

Verifica se houve uma diminuição monótona nos valores.
 Tipos de valores suportados: *Integer*.
 Retorna: 1 - se todos os elementos no período de tempo diminuírem continuamente; 0 - caso contrário.

Parâmetros:

- Veja parâmetros comuns
- **mode** (deve estar entre aspas duplas) - *weak* (cada valor é menor ou igual ao anterior; padrão) ou *strict* (cada valor diminuiu)

Exemplos:

```
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 #o número de processos nginx diminuiu monotonamente em 10 mi
```

monoinc(/host/key,(sec|#num)<:time shift>,<mode>)

Verifica se houve um aumento monótono nos valores.
 Tipos de valores suportados: *Integer*.
 Retorna: 1 - se todos os elementos no período de tempo aumentarem continuamente; 0 - caso contrário.

Parâmetros:

- Veja parâmetros comuns
- **mode** (deve ser colocado entre aspas duplas) - *weak* (todo valor é maior ou igual ao anterior; padrão) ou *strict* (todo valor aumentou)

Exemplos:

```
monoinc(/host/system.localtime,#3,"strict")=0 #o horário local do sistema não aumentou consistentemente
monoinc(/host/vfs.dir.count[/mnt/data/logs],24h,"weak")=0 #dispara se a contagem de arquivos estagnar por
```

nodata(/host/key,sec,<mode>)

Verifica se não há dados recebidos.
 Tipos de valores suportados: *Integer*, *Float*, *Character*, *Text*, *Log*.
 Retorna: 1 - se nenhum dado for recebido durante o período de tempo definido; 0 - caso contrário.

Parâmetros:

- Veja parâmetros comuns
- **sec** - o período não deve ser inferior a 30 segundos porque o processo history syncer calcula esta função apenas a cada 30 segundos; `nodata(/host/key,0)` não é permitido
- **mode** - se definido como *strict* (entre aspas duplas), esta função será insensível à disponibilidade do proxy (veja os comentários para detalhes)

Comentários:

- os triggers 'nodata' monitorados por proxy são, por padrão, sensíveis à disponibilidade do proxy - se o proxy ficar indisponível, os triggers 'nodata' não serão disparados imediatamente após a restauração da conexão, mas pularão os dados pelo período de atraso. Observe que para proxies passivos a supressão é ativada se a conexão for restaurada mais de 15 segundos e não menos de 2 segundos depois. Para proxies ativos, a supressão é ativada se a conexão for restaurada mais de 15 segundos depois. Para desativar a sensibilidade à disponibilidade do proxy, use o terceiro parâmetro, por exemplo: `nodata(/host/key,5m,"strict")`; neste caso, a função será disparada assim que o período de avaliação (cinco minutos) sem dados tiver passado.
- Esta função exibirá um erro se, dentro do período do 1º parâmetro:
- não houver dados e o Zabbix server for reiniciado
- não houver dados e a manutenção for concluída
- não houver dados e o item for adicionado ou reativado.
- Os erros são exibidos na coluna *Info* na configuração do trigger.
- Esta função pode não funcionar corretamente se houver diferenças de horário entre o Zabbix server, proxy e agent. Veja também: Requisito de sincronização de horário.

Exemplo:

```
nodata(/host/agent.ping,5m)=1 #trigger se nenhum dado for recebido do Zabbix agent por 5 minutos
```

percentile(/host/key,(sec|#num)<:time shift>,percentage)

O percentil P de um período, onde P (percentual) é especificado pelo terceiro parâmetro.
 Tipos de valor suportados: *Float*, *Integer*.

Parâmetros:

- Veja parâmetros comuns
- **percentage** - um número de ponto flutuante entre 0 e 100 (inclusive) com até 4 dígitos após o ponto decimal

Exemplos:

```
percentile(/host/net.if.in[eth0,bytes],1h,95)>1000000 #o percentil 95 da entrada de rede (bytes/seg) em 1
percentile(/host/system.cpu.util,5m,95)>80 #o percentil 95 da porcentagem de tempo de usuário da utilizaçã
percentile(/host/icmppingsec[192.168.0.2],15m,95)>0.15 #a maioria das medições de latência está abaixo de
percentile(/host/net.if.in[eth0,bytes],1h,50) #calcula o percentil 50 (o valor mediano) do tráfego de rede
(percentile(/host/net.if.in[eth0,bytes],1h,50)+percentile(/host/net.if.in[eth0,bytes],1h,51))/2 #calcula o
```

rate(/host/key,sec<:time shift>)

A taxa média por segundo do aumento em um contador monotonicamente crescente dentro do período de tempo definido.
 Tipos de valor suportados: *Float*, *Integer*.

Parâmetros:

- Veja parâmetros comuns

Corresponde funcionalmente ao 'rate' do PromQL.

Exemplos:

```
rate(/host/key,30s) #se o aumento monotônico em 30 segundos for 20, esta função retorna 0,67.
rate(/host/net.if.in[eth0,bytes],5m)>500000 #a taxa de tráfego de entrada na interface eth0 excedeu 500 KB
rate(/host/app.requests.count,1m)>100 #o contador de solicitações aumentou em >100 solicitações por segund
```

Veja todas as funções suportadas.

5 Funções de tendência

As funções de tendência, ao contrário das funções de histórico, usam dados de tendência para cálculos.

Tendências armazenam valores agregados por hora As funções de tendência usam essas médias horárias e, portanto, são úteis para análise de longo prazo.

Os resultados das funções de tendência são armazenados em cache, portanto, várias chamadas para a mesma função com os mesmos parâmetros buscam informações no banco de dados apenas uma vez. O cache da função de tendência é controlado pelo parâmetro TrendFunctionCacheSize do server.

Triggers que referenciam **apenas** funções de tendência são avaliados uma vez por menor período de tempo na expressão. Por exemplo, um trigger como

```
trendavg(/host/key,1d:now/d) > 1 or trendavg(/host/key2,1w:now/w) > 2
```

será avaliado uma vez por dia. Se o trigger contiver funções de tendência e histórico (ou data e hora e/ou nodata()), ele será calculado de acordo com os princípios usuais.

Todas as funções listadas aqui são suportadas em:

- Expressões de trigger
- Itens calculados
- Macros de expressão

As funções são listadas sem informações adicionais. Clique na função para ver os detalhes completos.

Função	Descrição
baselinedev	Retorna o número de desvios (pelo algoritmo stddevpop) entre o último período de dados e os mesmos períodos de dados em temporadas anteriores.
baselinewma	Calcula a linha de base pela média dos dados do mesmo intervalo de tempo em vários períodos de tempo iguais ('temporadas') usando o algoritmo de média móvel ponderada.
trendavg	A média dos valores de tendência dentro do período de tempo definido.
trendcount	O número de valores de histórico recuperados com sucesso usados para calcular o valor de tendência dentro do período de tempo definido.
trendmax	O máximo nos valores de tendência dentro do período de tempo definido.
trendmin	O mínimo nos valores de tendência dentro do período de tempo definido.
trendstl	Retorna a taxa de anomalias durante o período de detecção - um valor decimal entre 0 e 1 que é `((o número de valores de anomalia)/(número total de valores))`.
trendsum	A soma dos valores de tendência dentro do período de tempo definido.

Parâmetros comuns

- `/host/key` é um parâmetro obrigatório comum
- `período de tempo:deslocamento de tempo` é um segundo parâmetro comum, onde:
  - **período de tempo** - o período de tempo (mínimo '1h'), definido como <N><unidade de tempo> onde `N` - o número de unidades de tempo, `unidade de tempo` - h (hora), d (dia), w (semana), M (mês) ou y (ano).
  - **deslocamento de tempo** - o deslocamento do período de tempo (veja exemplos de funções)

Detalhes da função

Algumas observações gerais sobre os parâmetros da função:

- Os parâmetros da função são separados por vírgula
- Parâmetros opcionais da função (ou partes do parâmetro) são indicados por < >
- Parâmetros específicos da função são descritos com cada função

- Os parâmetros `/host/key` e `período de tempo:deslocamento de tempo` nunca devem ser colocados entre aspas

baselinedev(/host/key,data period:time shift,season unit,num seasons)

Retorna o número de desvios (pelo algoritmo stddevpop) entre o último período de dados e os mesmos períodos de dados nas temporadas anteriores.

Parâmetros:

- Veja parâmetros comuns
- **data period** - o período de coleta de dados dentro de uma temporada, definido como <N><unidade de tempo> onde:
`N` - o número de unidades de tempo
`unidade de tempo` - h (hora), d (dia), w (semana), M (mês) ou y (ano), deve ser igual ou menor que a temporada
- **season unit** - a unidade de calendário que define uma temporada (h, d, w, M, y); não pode ser menor que o período de dados
- **num seasons** - o número de temporadas a serem avaliadas

Exemplos:

```
baselinedev(/host/key,1d:now/d,"M",6) #calcula o número de desvios padrão (população) entre o dia anterior
baselinedev(/host/key,1h:now/h,"d",10)  #calcula o número de desvios padrão da população entre a hora ante
```

baselinewma(/host/key,data period:time shift,season unit,num seasons)

Calcula a linha de base pela média dos dados do mesmo período de tempo em vários períodos de tempo iguais (’seasons’) usando o algoritmo de média móvel ponderada.

Parâmetros:

- Veja parâmetros comuns
- **data period** - o período de coleta de dados dentro de uma season, definido como <N><time unit> onde:
`N` - o número de unidades de tempo
`time unit` - h (hora), d (dia), w (semana), M (mês) ou y (ano), deve ser igual ou menor que a season
Time shift - o deslocamento do período de tempo, define o final do período de coleta de dados em seasons (veja exemplos)
- **season unit** - a unidade de calendário que define uma season (h, d, w, M, y); não pode ser menor que o data period
- **num seasons** - o número de seasons a serem avaliadas

Exemplos:

```
baselinewma(/host/key,1h:now/h,"d",3) #calculando a linha de base da mesma hora do dia nos últimos 3 dias
baselinewma(/host/key,2h:now/h,"d",3) #calculando a linha de base do mesmo período de duas horas nos últim
baselinewma(/host/key,1d:now/d,"M",4) #calculando a linha de base do mesmo dia do mês que 'ontem' nos 4 me
```

trendavg(/host/key,time period:time shift)

A média dos valores de tendência dentro do período de tempo definido.

Parâmetros:

- Veja parâmetros comuns

Exemplos:

```
trendavg(/host/key,1h:now/h) #a média da hora anterior (por exemplo, 12:00-13:00)
trendavg(/host/key,1h:now/h-1h) #a média de duas horas atrás (11:00-12:00)
trendavg(/host/key,1h:now/h-2h) #a média de três horas atrás (10:00-11:00)
trendavg(/host/key,1M:now/M-1y) #a média do mês anterior há um ano
```

trendcount(/host/key,time period:time shift)

O número de valores de histórico recuperados com sucesso usados para calcular o valor de tendência dentro do período de tempo definido.

Parâmetros:

- Veja parâmetros comuns

Exemplos:

```
trendcount(/host/key,1h:now/h) #a contagem de valores para a hora anterior (por exemplo, 12:00-13:00)
trendcount(/host/key,1h:now/h-1h) #a contagem de valores para duas horas atrás (11:00-12:00)
trendcount(/host/key,1h:now/h-2h) #a contagem de valores para três horas atrás (10:00-11:00)
trendcount(/host/key,1M:now/M-1y) #a contagem de valores para o mês anterior há um ano
```

trendmax(/host/key,time period:time shift)

O máximo nos valores de tendência dentro do período de tempo definido.

Parâmetros:

- Veja parâmetros comuns

Exemplos:

```
trendmax(/host/key,1h:now/h) #o máximo para a hora anterior (por exemplo, 12:00-13:00)
trendmax(/host/key,1h:now/h) - trendmin(/host/key,1h:now/h) → calcula a diferença entre os valores máximo
trendmax(/host/key,1h:now/h-1h) #o máximo para duas horas atrás (11:00-12:00)
trendmax(/host/key,1h:now/h-2h) #o máximo para três horas atrás (10:00-11:00)
trendmax(/host/key,1M:now/M-1y) #o máximo para o mês anterior há um ano
```

trendmin(/host/key,time period:time shift)

O mínimo nos valores de tendência dentro do período de tempo definido.

Parâmetros:

- Veja parâmetros comuns

Exemplos:

```
trendmin(/host/key,1h:now/h) #o mínimo para a hora anterior (por exemplo, 12:00-13:00)
trendmax(/host/key,1h:now/h) - trendmin(/host/key,1h:now/h) → calcula a diferença entre os valores máximo
trendmin(/host/key,1h:now/h-1h) #o mínimo para duas horas atrás (11:00-12:00)
trendmin(/host/key,1h:now/h-2h) #o mínimo para três horas atrás (10:00-11:00)
trendmin(/host/key,1M:now/M-1y) #o mínimo para o mês anterior há um ano
```

trendstl(/host/key,eval period:time shift,detection period,season,<deviations>,<devalg>,<s window>)

Retorna a taxa de anomalias durante o período de detecção - um valor decimal entre 0 e 1 que é `((o número de valores anômalos)/(número total de valores))`.

Parâmetros:

- Veja parâmetros comuns
- **eval period** - o período de tempo que deve ser decomposto (mínimo '1h'), definido como <N><unidade de tempo> onde
`N` - o número de unidades de tempo
`unidade de tempo` - h (hora), d (dia), w (semana), M (mês) ou y (ano)
- **detection period** - o período de tempo antes do final do eval period para o qual as anomalias são calculadas (mínimo '1h', não pode ser maior que eval period), definido como <N><unidade de tempo> onde
`N` - o número de unidades de tempo
`unidade de tempo` - h (hora), d (dia), w (semana)
- **season** - o menor período de tempo onde um padrão repetitivo ("sazonalidade") é esperado (mínimo '2h', não pode ser maior que eval period, o número de entradas no eval period deve ser maior que duas vezes a frequência resultante (season/h)), definido como <N><unidade de tempo> onde
`N` - o número de unidades de tempo
`unidade de tempo` - h (hora), d (dia), w (semana)
- **deviations** - o número de desvios (calculado por devalg) para contar como anomalia (pode ser decimal), (deve ser maior ou igual a 1, padrão é 3)
- **devalg** (deve ser entre aspas duplas) - o algoritmo de desvio, pode ser *stddevpop*, *stddevsamp* ou *mad* (padrão)
- **s window** - o intervalo (em lags) da janela loess para extração sazonal (padrão é 10 * número de entradas no eval period + 1)

Exemplos:

```
trendstl(/host/key,100h:now/h,10h,2h) #analisa as últimas 100 horas de dados de tendência, encontra a taxa
trendstl(/host/key,100h:now/h-10h,100h,2h,2.1,"mad") #analisa o período de 100 horas de dados de tendência
trendstl(/host/key,100d:now/d-1d,10d,1d,4,,10) #analisa 100 dias de dados de tendência até um dia atrás, e
trendstl(/host/key,1M:now/M-1y,1d,2h,,"stddevsamp") #analisa o mês anterior há um ano, encontra a taxa de
```

trendsum(/host/key,time period:time shift)

A soma dos valores de tendência dentro do período de tempo definido.

Parâmetros:

- Veja parâmetros comuns

Exemplos:

```
trendsum(/host/key,1h:now/h) #a soma para a hora anterior (por exemplo, 12:00-13:00)
trendsum(/host/key,1h:now/h-1h) #a soma para duas horas atrás (11:00-12:00)
trendsum(/host/key,1h:now/h-2h) #a soma para três horas atrás (10:00-11:00)
trendsum(/host/key,1M:now/M-1y) #a soma para o mês anterior há um ano
```

Veja todas as funções suportadas.

6 Funções matemáticas

Todas as funções listadas aqui são suportadas em:

- Expressões de trigger
- Itens calculados
- Macros de expressão

Funções matemáticas são suportadas com tipos de valores float e integer, a menos que indicado de outra forma.

As funções são listadas sem informações adicionais. Clique na função para ver os detalhes completos.

Função	Descrição
abs	O valor absoluto de um valor.
acos	O arco cosseno de um valor como um ângulo, expresso em radianos.
asin	O arco seno de um valor como um ângulo, expresso em radianos.
atan	O arco tangente de um valor como um ângulo, expresso em radianos.
atan2	O arco tangente das coordenadas ordenada (valor) e abscissa especificadas como um ângulo, expresso em radianos.
avg	O valor médio dos valores de item referenciados.
cbrt	A raiz cúbica de um valor.
ceil	Arredonda o valor para cima para o inteiro mais próximo maior ou igual.
cos	O cosseno de um valor, onde o valor é um ângulo expresso em radianos.
cosh	O cosseno hiperbólico de um valor.
cot	A cotangente de um valor, onde o valor é um ângulo expresso em radianos.
degrees	Converte um valor de radianos para graus.
e	O número de Euler (2.718281828459045).
exp	O número de Euler elevado à potência de um valor.
expm1	O número de Euler elevado à potência de um valor menos 1.
floor	Arredonda o valor para baixo para o inteiro mais próximo menor ou igual.
log	O logaritmo natural.
log10	O logaritmo decimal.
max	O maior valor dos valores de item referenciados.
min	O menor valor dos valores de item referenciados.
mod	O resto da divisão.
pi	A constante Pi (3.14159265358979).
power	A potência de um valor.
radians	Converte um valor de graus para radianos.
rand	Retorna um valor inteiro aleatório.
round	Arredonda o valor para casas decimais.
signum	Retorna '-1' se um valor for negativo, '0' se um valor for zero, '1' se um valor for positivo.
sin	O seno de um valor, onde o valor é um ângulo expresso em radianos.
sinh	O seno hiperbólico de um valor, onde o valor é um ângulo expresso em radianos.
sqrt	A raiz quadrada de um valor.
sum	A soma dos valores de item referenciados.
tan	A tangente de um valor.
truncate	Trunca o valor para casas decimais.

Detalhes da função

Algumas observações gerais sobre os parâmetros da função:

- Os parâmetros da função são separados por vírgula
- Expressões são aceitas como parâmetros
- Parâmetros opcionais da função (ou partes do parâmetro) são indicados por < >

abs(value)

O valor absoluto (a partir de 0) de um valor.

Parâmetro:

- **value** - o valor a ser verificado

Por exemplo, o valor absoluto de '3' ou '-3' será '3'.

Exemplo:

```
abs(last(/host/key))>10
```

acos(value)

O arco-cosseno de um valor como um ângulo, expresso em radianos.

Parâmetro:

- **value** - o valor a ser verificado

O valor deve estar entre -1 e 1. Por exemplo, o arco-cosseno de um valor '0.5' será '2.0943951'.

Exemplo:

```
acos(last(/host/key))
```

asin(value)

O arco seno de um valor como um ângulo, expresso em radianos.

Parâmetro:

- **value** - o valor a ser verificado

O valor deve estar entre -1 e 1. Por exemplo, o arco seno de um valor '0.5' será '-0.523598776'.

Exemplo:

```
asin(last(/host/key))
```

atan(value)

O arco-tangente de um valor como um ângulo, expresso em radianos.

Parâmetro:

- **value** - o valor a ser verificado

Por exemplo, o arco-tangente de um valor '1' será '0.785398163'.

Exemplo:

```
atan(last(/host/key))
```

atan2(value,abscissa)

O arco-tangente das coordenadas de ordenada (valor) e abscissa especificadas como um ângulo, expresso em radianos.

Parâmetros:

- **value** - o valor a ser verificado;
- **abscissa** - o valor da abscissa.

Por exemplo, o arco-tangente das coordenadas de ordenada e abscissa de um valor '1' será '2.21429744'.

Exemplo:

```
atan2(last(/host/key),2)
```

avg(<value1>,<value2>,...)

O valor médio dos valores dos items referenciados.

Parâmetro:

- **valueX** - o valor retornado por outra função que está trabalhando com o histórico do item.

Exemplo:

```
avg(avg(/host/key,1h),avg(/host2/key2,1h))
```

cbrt(value)

A raiz cúbica de um valor.

Parâmetro:

- **value** - o valor a ser verificado

Por exemplo, a raiz cúbica de ’64’ será ’4’, de ’63’ será ’3.97905721’.

Exemplo:

```
cbrt(last(/host/key))
```

ceil(value)

Arredonda o valor para o inteiro mais próximo, maior ou igual.

Parâmetro:

- **value** - o valor a ser verificado

Por exemplo, ’2.4’ será arredondado para ’3’. Veja também floor().

Exemplo:

```
ceil(last(/host/key))
```

cos(value)

O cosseno de um valor, onde o valor é um ângulo expresso em radianos.

Parâmetro:

- **value** - o valor a ser verificado

Por exemplo, o cosseno de um valor ’1’ será ’0.54030230586’.

Exemplo:

```
cos(last(/host/key))
```

cosh(value)

O cosseno hiperbólico de um valor. Retorna o valor como um número real, não como notação científica.

Parâmetro:

- **value** - o valor a ser verificado

Por exemplo, o cosseno hiperbólico de um valor ’1’ será ’1.54308063482’.

Exemplo:

```
cosh(last(/host/key))
```

cot(value)

O cotangente de um valor, onde o valor é um ângulo expresso em radianos.

Parâmetro:

- **value** - o valor a ser verificado

Por exemplo, o cotangente de um valor ’1’ será ’0.54030230586’.

Exemplo:

```
cot(last(/host/key))
```

degrees(value)

Converte um valor de radianos para graus.

Parâmetro:

- **value** - o valor a ser verificado

Por exemplo, um valor ’1’ convertido para graus será ’57.2957795’.

Exemplo:

```
degrees(last(/host/key))
```

e

O número de Euler (2.718281828459045).

Exemplo:

```
e()
```

exp(value)

O número de Euler elevado a uma potência de um valor.

Parâmetro:

- **value** - o valor a ser verificado

Por exemplo, o número de Euler elevado à potência de um valor '2' será '7.38905609893065'.

Exemplo:

```
exp(last(/host/key))
```

expm1(value)

O número de Euler elevado a um valor menos 1.

Parâmetro:

- **value** - o valor a ser verificado

Por exemplo, o número de Euler elevado ao valor '2' menos 1 será '6.38905609893065'.

Exemplo:

```
expm1(last(/host/key))
```

floor(value)

Arredonda o valor para baixo para o inteiro menor ou igual mais próximo.

Parâmetro:

- **value** - o valor a ser verificado

Por exemplo, '2.6' será arredondado para baixo para '2'. Veja também ceil().

Exemplo:

```
floor(last(/host/key))
```

log(value)

O logaritmo natural.

Parâmetro:

- **value** - o valor a ser verificado

Por exemplo, o logaritmo natural de um valor '2' será '0.69314718055994529'.

Exemplo:

```
log(last(/host/key))
```

log10(value)

O logaritmo decimal.

Parâmetro:

- **value** - o valor a ser verificado

Por exemplo, o logaritmo decimal de um valor '5' será '0.69897000433'.

Exemplo:

```
log10(last(/host/key))
```

max(<value1>,<value2>,...)

O maior valor dos valores de item referenciados.

Parâmetro:

- **valueX** - o valor retornado por outra função que está trabalhando com o histórico do item.

Exemplo:

```
max(avg(/host/key,1h),avg(/host2/key2,1h))
```

min(<valor1>,<valor2>,...)

O menor valor dos valores de item referenciados.

Parâmetro:

- **valorX** - o valor retornado por outra função que está trabalhando com o histórico do item.

Exemplo:

```
min(avg(/host/key,1h),avg(/host2/key2,1h))
```

mod(value,denominator)

O resto da divisão.

Parâmetros:

- **value** - o valor a ser verificado;
- **denominator** - o denominador da divisão.

Por exemplo, o resto da divisão de um valor '5' com o denominador da divisão '2' será '1'.

Exemplo:

```
mod(last(/host/key),2)
```

pi

A constante Pi (3.14159265358979).

Exemplo:

```
pi()
```

power(value,power value)

A potência de um valor.

Parâmetros:

- **value** - o valor a ser verificado;
- **power value** - a potência N a ser usada.

Por exemplo, a 3ª potência de um valor '2' será '8'.

Exemplo:

```
power(last(/host/key),3)
```

radians(value)

Converte um valor de graus para radianos.

Parâmetro:

- **value** - o valor a ser verificado

Por exemplo, um valor '1' convertido para radianos será '0.0174532925'.

Exemplo:

```
radians(last(/host/key))
```

rand

Retorna um valor inteiro aleatório. Um número pseudo-aleatório gerado usando o tempo como semente (suficiente para fins matemáticos, mas não para criptografia).

Exemplo:

```
rand()
```

round(value,decimal places)

Arredonda o valor para casas decimais.

Parâmetros:

- **value** - o valor a ser verificado;
- **decimal places** - especifica as casas decimais para arredondamento (0 também é possível).

Por exemplo, um valor '2.5482' arredondado para 2 casas decimais será '2.55'.

Exemplo:

```
round(last(/host/key),2)
```

signum(value)

Retorna '-1' se o valor for negativo, '0' se o valor for zero, '1' se o valor for positivo.

Parâmetro:

- **value** - o valor a ser verificado.

Exemplo:

```
signum(last(/host/key))
```

sin(value)

O seno de um valor, onde o valor é um ângulo expresso em radianos.

Parâmetro:

- **value** - o valor a ser verificado

Por exemplo, o seno de um valor '1' será '0.8414709848'.

Exemplo:

```
sin(last(/host/key))
```

sinh(value)

O seno hiperbólico de um valor.

Parâmetro:

- **value** - o valor a ser verificado

Por exemplo, o seno hiperbólico de um valor '1' será '1.17520119364'.

Exemplo:

```
sinh(last(/host/key))
```

sqrt(value)

A raiz quadrada de um valor.
 Esta função irá falhar com um valor negativo.

Parâmetro:

- **value** - o valor a ser verificado

Por exemplo, a raiz quadrada de um valor '3.5' será '1.87082869339'.

Exemplo:

```
sqrt(last(/host/key))
```

sum(<value1>,<value2>,...)

A soma dos valores dos items referenciados.

Parâmetro:

- **valueX** - o valor retornado por outra função que está trabalhando com o histórico do item.

Exemplo:

```
sum(avg(/host/key,1h),avg(/host2/key2,1h))
```

tan(value)

A tangente de um valor.

Parâmetro:

- **value** - o valor a ser verificado

Por exemplo, a tangente de um valor '1' será '1.55740772465'.

Exemplo:

```
tan(last(/host/key))
```

truncate(value,decimal places)

Trunca o valor para casas decimais.

Parâmetros:

- **value** - o valor a ser verificado;
- **decimal places** - especifica as casas decimais para truncar (0 também é possível).

Por exemplo, um valor '2.5482' truncado para 2 casas decimais será '2.54'.

Exemplo:

```
truncate(last(/host/key),2)
```

Veja todas as funções suportadas.

7 Funções de operador

Todas as funções listadas aqui são suportadas em:

- Expressões de trigger
- Items calculados
- Macros de expressão

As funções são listadas sem informações adicionais. Clique na função para ver os detalhes completos.

Função	Descrição
between	Verifica se o valor pertence ao intervalo fornecido.
in	Verifica se o valor é igual a pelo menos um dos valores listados.

Detalhes da função

Algumas observações gerais sobre os parâmetros da função:

- Os parâmetros da função são separados por vírgula
- Expressões são aceitas como parâmetros

between(value,min,max)

Verifica se o valor pertence ao intervalo fornecido.
 Tipos de valores suportados: *Integer*, *Float*.
 Retorna: 1 - dentro do intervalo; 0 - caso contrário.

Parâmetros:

- **value** - o valor a ser verificado;
- **min** - o valor mínimo;
- **max** - o valor máximo.

Exemplo:

```
between(last(/host/key),1,10)=1 #trigger se o valor estiver entre 1 e 10
```

in(value,value1,value2,...valueN)

Verifica se o valor é igual a pelo menos um dos valores listados.
 Tipos de valores suportados: *Integer*, *Float*, *Character*, *Text*, *Log*.
 Retorna: 1 - se igual; 0 - caso contrário.

Parâmetros:

- **value** - o valor a ser verificado;

- **valueX** - valores listados (valores de string devem estar entre aspas duplas).

O valor é comparado aos valores listados como números, se todos esses valores puderem ser convertidos para numéricos; caso contrário, são comparados como strings.

Exemplo:

```
in(last(/host/key),5,10)=1 #trigger se o último valor for igual a 5 ou 10
in("text",last(/host/key),last(/host/key,#2))=1 #trigger se "text" for igual a qualquer um dos 2 últimos v
```

Veja todas as funções suportadas.

8 Funções de previsão

Todas as funções listadas aqui são suportadas em:

- Expressões de trigger
- Itens calculados
- Macros de expressão

As funções são listadas sem informações adicionais. Clique na função para ver os detalhes completos.

Função	Descrição
forecast	O valor futuro, max, min, delta ou avg do item.
timeleft	O tempo em segundos necessário para que um item atinja o limite especificado.

Parâmetros comuns

- `/host/key` é um parâmetro obrigatório comum para as funções que fazem referência ao histórico do item do host
- `(sec|#num)<:time shift>` é um parâmetro comum para as funções que fazem referência ao histórico do item do host, onde:
  - **sec** - período de avaliação máximo em segundos (sufixos de tempo podem ser usados), ou
  - **#num** - intervalo de avaliação máximo nos últimos valores coletados (se precedido por um sinal de hash)
  - **time shift** (opcional) permite mover o ponto de avaliação para trás no tempo. Veja mais detalhes sobre como especificar o deslocamento de tempo.

Detalhes da função

Algumas observações gerais sobre os parâmetros da função:

- Os parâmetros da função são separados por vírgula
- Parâmetros opcionais da função (ou partes do parâmetro) são indicados por < >
- Parâmetros específicos da função são descritos com cada função
- Os parâmetros `/host/key` e `(sec|#num)<:deslocamento de tempo>` nunca devem ser colocados entre aspas

forecast(/host/key,(sec|#num)<:time shift>,time,<fit>,<mode>)

O valor futuro, máximo, mínimo, delta ou média do item.
 Tipos de valores suportados: *Float*, *Integer*.

Parâmetros:

- Veja parâmetros comuns;
- **time** - o horizonte de previsão em segundos (sufixos de tempo podem ser usados); valores negativos são suportados;
- **fit** (opcional; deve ser entre aspas duplas) - a função usada para ajustar os dados históricos. Ajustes suportados:
linear - função linear (padrão)
polynomialN - polinomial de grau N (1 <= N <= 6)
exponential - função exponencial
logarithmic - função logarítmica
power - função de potência
Observe que *polynomial1* é equivalente a *linear*;
- **mode** (opcional; deve ser entre aspas duplas) - a saída desejada. Modos suportados:
value - valor (padrão)
max - máximo
min - mínimo
delta - *max-min*
avg - média
Observe que *value* estima o valor do item no momento `now + time`; *max*, *min*, *delta* e *avg* investigam a estimativa do valor do item no intervalo entre `now` e `now + time`.

Comentários:

- Se o valor a ser retornado for maior que 1.7976931348623158E+308 ou menor que -1.7976931348623158E+308, o valor de retorno é limitado a 1.7976931348623158E+308 ou -1.7976931348623158E+308, respectivamente;
- Torna-se não suportado apenas se for usado incorretamente na expressão (tipo de item errado, parâmetros inválidos), caso contrário retorna -1 em caso de erros;
- Veja também informações adicionais sobre funções de trigger preditivas.

Exemplos:

```
forecast(/host/key,#10,1h) #prevê o valor do item em uma hora com base nos últimos 10 valores
forecast(/host/key,1h,30m) #prevê o valor do item em 30 minutos com base nos dados da última hora
forecast(/host/key,1h:now-1d,12h) #prevê o valor do item em 12 horas com base em uma hora de um dia atrás
forecast(/host/key,1h,10m,"exponential") #prevê o valor do item em 10 minutos com base nos dados da última
forecast(/host/key,1h,2h,"polynomial3","max") #prevê o valor máximo que o item pode atingir nas próximas d
forecast(/host/key,#2,-20m) #estima o valor do item 20 minutos atrás com base nos dois últimos valores (is
```

timeleft(/host/key,(sec|#num)<:time shift>,threshold,<fit>)

O tempo em segundos necessário para que um item atinja o limite especificado.
 Tipos de valor suportados: *Float*, *Integer*.

Parâmetros:

- Veja parâmetros comuns;
- **threshold** - o valor a ser atingido (sufixos de unidade podem ser usados);
- **fit** (opcional; deve ser colocado entre aspas duplas) - veja forecast().

Comentários:

- Se o valor a ser retornado for maior que 1.7976931348623158E+308, o valor de retorno é limitado a 1.7976931348623158E+308;
- Retorna 1.7976931348623158E+308 se o limite não puder ser atingido;
- Torna-se não suportado apenas se for usado incorretamente na expressão (tipo de item errado, parâmetros inválidos), caso contrário retorna -1 em caso de erros;
- Veja também informações adicionais sobre funções de trigger preditivas.

Exemplos:

```
timeleft(/host/key,#10,0) #o tempo até que o valor do item atinja zero com base nos últimos 10 valores
timeleft(/host/key,1h,100) #o tempo até que o valor do item atinja 100 com base nos dados da última hora
timeleft(/host/key,1h:now-1d,100) #o tempo até que o valor do item atinja 100 com base em uma hora de um d
timeleft(/host/key,1h,200,"polynomial2") #o tempo até que o valor do item atinja 200 com base nos dados da
```

Veja todas as funções suportadas.

9 Funções de string

Todas as funções listadas aqui são suportadas em:

- Expressões de trigger
- Itens calculados
- Macros de expressão

As funções são listadas sem informações adicionais. Clique na função para ver os detalhes completos.

Função	Descrição
ascii	O código ASCII do caractere mais à esquerda do valor.
bitlength	O comprimento do valor em bits.
bytelength	O comprimento do valor em bytes.
char	Retorna o caractere interpretando o valor como código ASCII.
concat	A string resultante da concatenação dos valores dos itens referenciados ou valores constantes.
insert	Insere caracteres ou espaços especificados na string de caracteres começando na posição especificada na string.
jsonpath	Retorna o resultado do JSONPath.
left	Retorna os caracteres mais à esquerda do valor.
length	O comprimento do valor em caracteres.
ltrim	Remove caracteres especificados do início da string.
mid	Retorna uma substring de N caracteres começando na posição de caractere especificada por 'start'.
repeat	Repete uma string.
replace	Encontra o padrão no valor e substitui pela substituição.
right	Retorna os caracteres mais à direita do valor.
rtrim	Remove caracteres especificados do final da string.
trim	Remove caracteres especificados do início e do final da string.
xmlxpath	Retorna o resultado do XML XPath.

Detalhes da função

Algumas observações gerais sobre os parâmetros da função:

- Os parâmetros da função são separados por vírgula
- Expressões são aceitas como parâmetros
- Parâmetros de string devem estar entre aspas duplas; caso contrário, podem ser interpretados incorretamente
- Parâmetros de função opcionais (ou partes de parâmetros) são indicados por < >

ascii(value)

O código ASCII do caractere mais à esquerda do valor.
 Tipos de valor suportados: *String*, *Text*, *Log*.

Parâmetro:

- **value** - o valor a ser verificado

Por exemplo, um valor como 'Abc' retornará '65' (código ASCII para 'A').

Exemplo:

```
ascii(last(/host/key))
```

bitlength(value)

O comprimento do valor em bits.
 Tipos de valores suportados: *String*, *Text*, *Log*, *Integer*.

Parâmetro:

- **value** - o valor a ser verificado

Exemplo:

```
bitlength(last(/host/key))
```

bytelength(value)

O comprimento do valor em bytes.
 Tipos de valor suportados: *String*, *Text*, *Log*, *Integer*.

Parâmetro:

- **value** - o valor a ser verificado

Exemplo:

```
bytelength(last(/host/key))
```

char(value)

Retorna o caractere interpretando o valor como código ASCII.
 Tipos de valor suportados: *Integer*.

Parâmetro:

- **value** - o valor a ser verificado

O valor deve estar no intervalo de 0-255. Por exemplo, um valor como '65' (interpretado como código ASCII) retornará 'A'.

Exemplo:

```
char(last(/host/key))
```

concat(<valor1>,<valor2>,...)

A string resultante da concatenação dos valores dos items referenciados ou valores constantes.
 Tipos de valores suportados: *String*, *Text*, *Log*, *Float*, *Integer*.

Parâmetro:

- **valorX** - o valor retornado por uma das funções de histórico ou um valor constante (string, inteiro ou número de ponto flutuante). Deve conter pelo menos dois parâmetros.

Por exemplo, um valor como 'Zab' concatenado com 'bix' (a string constante) retornará 'Zabbix'.

Exemplos:

```
concat(last(/host/key),"bix")
concat("1 min: ",last(/host/system.cpu.load[all,avg1]),", 15 min: ",last(/host/system.cpu.load[all,avg15]))
```

insert(value,start,length,replacement)

Insere caracteres ou espaços especificados na string de caracteres começando na posição especificada na string.
 Tipos de valor suportados: *String*, *Text*, *Log*.

Parâmetros:

- **value** - o valor a ser verificado;
- **start** - posição inicial;
- **length** - posições a serem substituídas;
- **replacement** - string de substituição.

Por exemplo, um valor como 'Zabbbix' será substituído por 'Zabbix' se 'bb' (posição inicial 3, posições a serem substituídas 2) for substituído por 'b'.

Exemplo:

```
insert(last(/host/key),3,2,"b")
```

jsonpath(value,path,<default>)

Retorna o resultado do JSONPath.
 Tipos de valor suportados: *String*, *Text*, *Log*.

Parâmetros:

- **value** - o valor a ser verificado;
- **path** - o caminho (deve estar entre aspas);
- **default** - o valor opcional de fallback se a consulta JSONPath não retornar dados. Note que em outros erros será retornada falha (por exemplo, "unsupported construct").

Exemplo:

```
jsonpath(last(/host/proc.get[zabbix_agentd,,,summary]),"$..size")
```

left(value,count)

Retorna os caracteres mais à esquerda do valor.
 Tipos de valores suportados: *String*, *Text*, *Log*.

Parâmetros:

- **value** - o valor a ser verificado;
- **count** - o número de caracteres a serem retornados.

Por exemplo, você pode retornar 'Zab' de 'Zabbix' especificando 3 caracteres mais à esquerda para retornar. Veja também right().

Exemplo:

```
left(last(/host/key),3) #retorna três caracteres mais à esquerda
```

length(value)

O comprimento do valor em caracteres.
 Tipos de valor suportados: *String*, *Text*, *Log*.

Parâmetro:

- **value** - o valor a ser verificado.

Exemplos:

```
length(last(/host/key)) #o comprimento do valor mais recente
length(last(/host/key,#3)) #o comprimento do terceiro valor mais recente
length(last(/host/key,#1:now-1d)) #o comprimento do valor mais recente há um dia atrás
```

ltrim(value,<chars>)

Remove os caracteres especificados do início da string.
 Tipos de valor suportados: *String*, *Text*, *Log*.

Parâmetros:

- **value** - o valor a ser verificado;
- **chars** (opcional) - especifica os caracteres a serem removidos.

Espaços em branco são removidos do início por padrão (se nenhum caractere opcional for especificado). Veja também: rtrim(), trim().

Exemplos:

```
ltrim(last(/host/key)) #remove espaços em branco do início da string
ltrim(last(/host/key),"Z") #remove qualquer 'Z' do início da string
ltrim(last(/host/key)," Z") #remove qualquer espaço e 'Z' do início da string
```

mid(value,start,length)

Retorna uma substring de N caracteres começando na posição de caractere especificada por 'start'.
 Tipos de valor suportados: *String*, *Text*, *Log*.

Parâmetros:

- **value** - o valor a ser verificado;
- **start** - posição inicial da substring;
- **length** - posições a serem retornadas na substring.

Por exemplo, é possível retornar 'abbi' de um valor como 'Zabbix' se a posição inicial for 2 e as posições a serem retornadas forem 4.

Exemplo:

```
mid(last(/host/key),2,4)="abbi"
```

repeat(value,count)

Repete uma string.
 Tipos de valor suportados: *String*, *Text*, *Log*.

Parâmetros:

- **value** - o valor a ser verificado;
- **count** - o número de vezes a repetir.

Exemplo:

```
repeat(last(/host/key),2) #repete o valor duas vezes
```

replace(value,pattern,replacement)

Procura o padrão no valor e substitui pelo valor de substituição. Todas as ocorrências do padrão serão substituídas.
 Tipos de valores suportados: *String*, *Text*, *Log*.

Parâmetros:

- **value** - o valor a ser verificado;
- **pattern** - o padrão a ser encontrado;
- **replacement** - a string para substituir o padrão.

Exemplo:

```
replace(last(/host/key),"ibb","abb") #substitui todos os 'ibb' por 'abb'
```

right(value,count)

Retorna os caracteres mais à direita do valor.
 Tipos de valor suportados: *String*, *Text*, *Log*.

Parâmetros:

- **value** - o valor a ser verificado;
- **count** - o número de caracteres a serem retornados.

Por exemplo, você pode retornar 'bix' de 'Zabbix' especificando 3 caracteres mais à direita para retornar. Veja também left().

Exemplo:

```
right(last(/host/key),3) #retorna três caracteres mais à direita
```

rtrim(value,<chars>)

Remove os caracteres especificados do final da string.
 Tipos de valor suportados: *String*, *Text*, *Log*.

Parâmetros:

- **value** - o valor a ser verificado;
- **chars** (opcional) - especifica os caracteres a serem removidos.

Espaços em branco são removidos à direita por padrão (se nenhum caractere opcional for especificado). Veja também: ltrim(), trim().

Exemplos:

```
rtrim(last(/host/key)) #remove espaços em branco do final da string
rtrim(last(/host/key),"x") #remove qualquer 'x' do final da string
rtrim(last(/host/key),"x ") #remove qualquer 'x' e espaço do final da string
```

trim(value,<chars>)

Remove os caracteres especificados do início e do final da string.
 Tipos de valor suportados: *String*, *Text*, *Log*.

Parâmetros:

- **value** - o valor a ser verificado;
- **chars** (opcional) - especifica os caracteres a serem removidos.

Espaços em branco são removidos de ambos os lados por padrão (se nenhum caractere opcional for especificado). Veja também: ltrim(), rtrim().

Exemplos:

```
trim(last(/host/key)) #remove espaços em branco do início e do final da string
trim(last(/host/key),"_") #remove '_' do início e do final da string
```

xmlxpath(value,path,<default>)

Retorna o resultado do XML XPath.
 Tipos de valores suportados: *String*, *Text*, *Log*.

Parâmetros:

- **value** - o valor a ser verificado;
- **path** - o caminho (deve estar entre aspas);
- **default** - o valor opcional de fallback se a consulta XML XPath retornar um conjunto de nós vazio. Não será retornado se o resultado vazio não for um conjunto de nós (ou seja, string vazia). Em outros erros, será retornada uma falha (por exemplo, ”invalid expression”).

Exemplo:

```
xmlxpath(last(/host/xml_result),"/response/error/status")
```

Veja todas as funções suportadas.

6 Macros

É possível usar Macros suportadas prontas para uso e Macros de usuário suportadas por local.

1 Macros suportadas

Visão geral

Esta página contém uma lista completa de macros internas suportadas pelo Zabbix, agrupadas por área de aplicação.

> **Note:**
> Para visualizar todas as macros suportadas em um local específico, cole o nome do local (por exemplo, ”map URL”) na caixa de pesquisa do seu navegador (acessível pressionando CTRL+F) e pesquise por *próximo*.

> **Note:**
> Para personalizar os valores das macros (por exemplo, encurtar ou extrair substrings específicas), você pode usar funções de macro.

Ações

Macro	Suportado em	Descrição
{ACTION.ID}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Notificações e comandos baseados em serviço → Notificações e comandos de atualização de serviço → Notificações e comandos de descoberta → Notificações e comandos de autorregistro → Notificações internas	*ID numérico da ação disparada.*

Macro	Suportado em	Descrição
{ACTION.NAME}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Notificações e comandos baseados em serviço → Notificações e comandos de atualização de serviço → Notificações e comandos de descoberta → Notificações e comandos de autorregistro → Notificações internas	*Nome da ação disparada.*
{ALERT.MESSAGE}	→ Parâmetros de script de alerta → Parâmetros de webhook	*Valor de 'Mensagem padrão' da configuração da ação.*
{ALERT.SENDTO}	→ Parâmetros de script de alerta → Parâmetros de webhook	*Valor de 'Enviar para' da configuração de mídia do usuário.*
{ALERT.SUBJECT}	→ Parâmetros de script de alerta → Parâmetros de webhook	*Valor de 'Assunto padrão' da configuração da ação.*
{ESC.HISTORY}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Notificações e comandos baseados em serviço → Notificações e comandos de atualização de serviço → Notificações internas	*Histórico de escalonamento. Log de mensagens enviadas anteriormente.* Mostra as notificações enviadas anteriormente, em qual etapa de escalonamento foram enviadas e seu status (*enviado*, *em andamento* ou *falhou*).

Data e hora

Macro	Suportado em	Descrição
{DATE}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Notificações e comandos baseados em serviço → Notificações e comandos de atualização de serviço → Notificações e comandos de descoberta → Notificações e comandos de autorregistro → Notificações internas → Comandos de script de ação de evento manual	*Data atual no formato aaaa.mm.dd.*
{TIME}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Notificações e comandos baseados em serviço → Notificações e comandos de atualização de serviço → Notificações e comandos de descoberta → Notificações e comandos de autorregistro → Notificações internas → Nomes de eventos de trigger → Comandos de script de ação de evento manual → Campos de assunto/mensagem de relatórios agendados	*Hora atual no formato hh:mm:ss.*

Descoberta

Macro	Suportado em	Descrição
{DISCOVERY.DEVICE.IPADDRESS}	→ Notificações e comandos de descoberta	*Endereço IP do dispositivo descoberto.* Disponível sempre, não depende do host ser adicionado.
{DISCOVERY.DEVICE.DNS}	→ Notificações e comandos de descoberta	*Nome DNS do dispositivo descoberto.* Disponível sempre, não depende do host ser adicionado.
{DISCOVERY.DEVICE.STATUS}	→ Notificações e comandos de descoberta	*Status do dispositivo descoberto*: pode ser UP ou DOWN.

Macro	Suportado em	Descrição
{DISCOVERY.DEVICE.UPTIME}	→ Notificações e comandos de descoberta	*Tempo desde a última alteração do status de descoberta para um determinado dispositivo*, com precisão de até um segundo. Por exemplo: 1h 29m 01s. Para dispositivos com status DOWN, este é o período de inatividade.
{DISCOVERY.RULE.NAME}	→ Notificações e comandos de descoberta	*Nome da regra de descoberta que detectou a presença ou ausência do dispositivo ou serviço.*
{DISCOVERY.SERVICE.NAME}	→ Notificações e comandos de descoberta	*Nome do serviço que foi descoberto.* Por exemplo: HTTP.
{DISCOVERY.SERVICE.PORT}	→ Notificações e comandos de descoberta	*Porta do serviço que foi descoberto.* Por exemplo: 80.
{DISCOVERY.SERVICE.STATUS}	→ Notificações e comandos de descoberta	*Status do serviço descoberto*: pode ser UP ou DOWN.
{DISCOVERY.SERVICE.UPTIME}	→ Notificações e comandos de descoberta	*Tempo desde a última alteração do status de descoberta para um determinado serviço*, com precisão de até um segundo. Por exemplo: 1h 29m 01s. Para serviços com status DOWN, este é o período de inatividade.

Eventos

Macro	Suportado em	Descrição
{EVENT.ACK.STATUS}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Comandos de script de ação de evento manual	*Status de reconhecimento do evento (Sim/Não)*.
{EVENT.AGE}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Notificações e comandos baseados em serviço → Notificações e comandos de atualização de serviço → Notificações e comandos de recuperação de serviço → Notificações e comandos de descoberta → Notificações e comandos de autorregistro → Notificações internas → Comandos de script de ação de evento manual	*Idade do evento que disparou uma ação*, com precisão de até um segundo. Útil em mensagens escalonadas. Por exemplo: 9m 13s.
{EVENT.DATE}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Notificações e comandos baseados em serviço → Notificações e comandos de atualização de serviço → Notificações e comandos de recuperação de serviço → Notificações e comandos de descoberta → Notificações e comandos de autorregistro → Notificações internas → Comandos de script de ação de evento manual	*Data do evento que disparou uma ação no formato aaaa.mm.dd.* Por exemplo: 2025.04.14.
{EVENT.DURATION}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Notificações e comandos baseados em serviço → Notificações e comandos de atualização de serviço → Notificações e comandos de recuperação de serviço → Notificações internas → Comandos de script de ação de evento manual	*Duração do evento (diferença de tempo entre eventos de problema e recuperação)*, com precisão de até um segundo. Útil em mensagens de recuperação de problema.

Macro	Suportado em	Descrição
{EVENT.ID}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Notificações e comandos baseados em serviço → Notificações e comandos de atualização de serviço → Notificações e comandos de recuperação de serviço → Notificações e comandos de descoberta → Notificações e comandos de autorregistro → Notificações internas → URLs de trigger → Comandos de script de ação de evento manual, texto de confirmação → URLs de ação de evento/host manual, texto de confirmação	*ID numérico do evento que disparou uma ação.*
{EVENT.NAME}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Notificações e comandos baseados em serviço → Notificações e comandos de atualização de serviço → Notificações e comandos de recuperação de serviço → Notificações internas → Comandos de script de ação de evento manual, texto de confirmação → URLs de ação de evento/host manual, texto de confirmação	*Nome do evento de problema que disparou uma ação.*
{EVENT.NSEVERITY}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Notificações e comandos baseados em serviço → Notificações e comandos de atualização de serviço → Notificações e comandos de recuperação de serviço → Comandos de script de ação de evento manual, texto de confirmação → URLs de ação de evento/host manual, texto de confirmação	*Valor numérico da severidade do evento.* Valores possíveis: 0 - Não classificado, 1 - Informação, 2 - Aviso, 3 - Médio, 4 - Alto, 5 - Desastre.
{EVENT.OBJECT}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Notificações e comandos baseados em serviço → Notificações e comandos de atualização de serviço → Notificações e comandos de recuperação de serviço → Notificações e comandos de descoberta → Notificações e comandos de autorregistro → Notificações internas → Comandos de script de ação de evento manual	*Valor numérico do objeto do evento.* Valores possíveis: 0 - Trigger, 1 - Host descoberto, 2 - Serviço descoberto, 3 - Autorregistro, 4 - Item, 5 - Regra de descoberta de baixo nível.
{EVENT.OPDATA}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Comandos de script de ação de evento manual	*Dados operacionais do trigger subjacente de um problema.*
{EVENT.RECOVERY.DATE}	→ Notificações de recuperação de problema e comandos → Notificações e comandos de atualização de problema (se a recuperação ocorreu) → Notificações e comandos de recuperação de serviço → Comandos de script de ação de evento manual (se a recuperação ocorreu)	*Data do evento de recuperação no formato aaaa.mm.dd.*

Macro	Suportado em	Descrição
{EVENT.RECOVERY.ID}	→ Notificações de recuperação de problema e comandos → Notificações e comandos de atualização de problema (se a recuperação ocorreu) → Notificações e comandos de recuperação de serviço → Comandos de script de ação de evento manual (se a recuperação ocorreu)	*ID numérico do evento de recuperação.*
{EVENT.RECOVERY.NAME}	→ Notificações de recuperação de problema e comandos → Notificações e comandos de atualização de problema (se a recuperação ocorreu) → Notificações e comandos de recuperação de serviço → Comandos de script de ação de evento manual (se a recuperação ocorreu)	*Nome do evento de recuperação.*
{EVENT.RECOVERY.STATUS}	→ Notificações de recuperação de problema e comandos → Notificações e comandos de atualização de problema (se a recuperação ocorreu) → Notificações e comandos de recuperação de serviço → Comandos de script de ação de evento manual (se a recuperação ocorreu)	*Valor verbal do evento de recuperação.*
{EVENT.RECOVERY.TAGS}	→ Notificações de recuperação de problema e comandos → Notificações e comandos de atualização de problema (se a recuperação ocorreu) → Notificações e comandos de recuperação de serviço → Notificações internas → Comandos de script de ação de evento manual (se a recuperação ocorreu)	*Uma lista separada por vírgulas das tags do evento de recuperação.* Expandido para uma string vazia se não existirem tags.
{EVENT.RECOVERY.TAGSJSON}	→ Notificações de recuperação de problema e comandos → Notificações e comandos de atualização de problema (se a recuperação ocorreu) → Notificações e comandos de recuperação de serviço → Notificações internas → Comandos de script de ação de evento manual (se a recuperação ocorreu)	*Um array JSON contendo objetos de tag de evento.* Expandido para um array vazio se não existirem tags.
{EVENT.RECOVERY.TIME}	→ Notificações de recuperação de problema e comandos → Notificações e comandos de atualização de problema (se a recuperação ocorreu) → Notificações e comandos de recuperação de serviço → Comandos de script de ação de evento manual (se a recuperação ocorreu)	*Hora do evento de recuperação no formato hh:mm:ss.*
{EVENT.RECOVERY.VALUE}	→ Notificações de recuperação de problema e comandos → Notificações e comandos de atualização de problema (se a recuperação ocorreu) → Notificações e comandos de recuperação de serviço → Comandos de script de ação de evento manual (se a recuperação ocorreu)	*Valor numérico do evento de recuperação.*

Macro	Suportado em	Descrição
{EVENT.SEVERITY}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Notificações e comandos baseados em serviço → Notificações e comandos de atualização de serviço → Notificações e comandos de recuperação de serviço → Comandos de script de ação de evento manual, texto de confirmação → URLs de ação de evento/host manual, texto de confirmação	*Nome da severidade do evento.*
{EVENT.SOURCE}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Notificações e comandos baseados em serviço → Notificações e comandos de atualização de serviço → Notificações e comandos de recuperação de serviço → Notificações e comandos de descoberta → Notificações e comandos de autorregistro → Notificações internas → Comandos de script de ação de evento manual	*Valor numérico da fonte do evento.* Valores possíveis: 0 - Trigger, 1 - Descoberta, 2 - Autorregistro, 3 - Interno, 4 - Serviço.
{EVENT.STATUS}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Notificações e comandos baseados em serviço → Notificações e comandos de atualização de serviço → Notificações e comandos de recuperação de serviço → Notificações internas → Comandos de script de ação de evento manual, texto de confirmação → URLs de ação de evento/host manual, texto de confirmação	*Valor verbal do evento que disparou uma ação.*
{EVENT.TAGS}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Notificações e comandos baseados em serviço → Notificações e comandos de atualização de serviço → Notificações e comandos de recuperação de serviço → Notificações internas → Comandos de script de ação de evento manual	*Uma lista separada por vírgulas das tags do evento.* Expandido para uma string vazia se não existirem tags.
{EVENT.TAGSJSON}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Notificações e comandos baseados em serviço → Notificações e comandos de atualização de serviço → Notificações e comandos de recuperação de serviço → Notificações internas → Comandos de script de ação de evento manual	*Um array JSON contendo objetos de tag de evento.* Expandido para um array vazio se não existirem tags.
{EVENT.TAGS.<tag name>}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Notificações e comandos baseados em serviço → Notificações e comandos de atualização de serviço → Notificações e comandos de recuperação de serviço → Notificações internas → Nomes e URLs de tipo de mídia webhook → Comandos de script de ação de evento manual	*Valor da tag do evento referenciado pelo nome da tag.* Um nome de tag contendo caracteres não alfanuméricos (incluindo caracteres multibyte-UTF não ingleses) deve ser colocado entre aspas duplas. Aspas e barras invertidas dentro de um nome de tag entre aspas devem ser escapadas com uma barra invertida.

Macro	Suportado em	Descrição
{EVENT.TIME}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Notificações e comandos baseados em serviço → Notificações e comandos de atualização de serviço → Notificações e comandos de recuperação de serviço → Notificações e comandos de descoberta → Notificações e comandos de autorregistro → Notificações internas → Comandos de script de ação de evento manual	*Hora do evento que disparou uma ação no formato hh:mm:ss.* Por exemplo: 12:57:53.
{EVENT.UPDATE.ACTION}	→ Notificações e comandos de atualização de problema	*Nome legível da(s) ação(ões)* realizada(s) durante a atualização do problema. Resolve para os seguintes valores: *reconhecido*, *não reconhecido*, *comentado*, *alterada severidade de (severidade original) para (severidade atualizada)* e *fechado* (dependendo de quantas ações são realizadas em uma atualização).
{EVENT.UPDATE.DATE}	→ Notificações e comandos de atualização de problema → Notificações e comandos de atualização de serviço	*Data da atualização do evento (reconhecimento, etc) no formato aaaa.mm.dd.* `{ACK.DATE}` está obsoleto e pode não ser resolvido nos locais suportados.
{EVENT.UPDATE.HISTORY}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Comandos de script de ação de evento manual	*Log de atualizações de problema (reconhecimentos, etc).* `{EVENT.ACK.HISTORY}` está obsoleto e pode não ser resolvido nos locais suportados.
{EVENT.UPDATE.MESSAGE}	→ Notificações e comandos de atualização de problema	*Mensagem de atualização do problema.* `{ACK.MESSAGE}` está obsoleto e pode não ser resolvido nos locais suportados.
{EVENT.UPDATE.NSEVERITY}	→ Notificações e comandos de atualização de serviço	*Valor numérico da nova severidade do evento definida durante a operação de atualização do problema.*
{EVENT.UPDATE.SEVERITY}	→ Notificações e comandos de atualização de serviço	*Nome da nova severidade do evento definida durante a operação de atualização do problema.*
{EVENT.UPDATE.STATUS}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Notificações e comandos de atualização de serviço (desde o Zabbix 7.0.2) → Comandos de script de ação de evento manual	*Valor numérico do status de atualização do problema.* Valores possíveis: 0 - Webhook foi chamado devido a evento de problema/recuperação, 1 - Operação de atualização.
{EVENT.UPDATE.TIME}	→ Notificações e comandos de atualização de problema → Notificações e comandos de atualização de serviço	*Hora da atualização do evento (reconhecimento, etc) no formato hh:mm:ss.* `{ACK.TIME}` está obsoleto e pode não ser resolvido nos locais suportados.
{EVENT.VALUE}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Notificações e comandos baseados em serviço → Notificações e comandos de atualização de serviço → Notificações e comandos de recuperação de serviço → Notificações internas → Comandos de script de ação de evento manual, texto de confirmação → URLs de ação de evento/host manual, texto de confirmação	*Valor numérico do evento que disparou uma ação (1 para problema, 0 para recuperação).*

Eventos de causa e sintoma

As macros {EVENT.CAUSE.*} são usadas no contexto de um evento de sintoma, por exemplo, em notificações; elas retornam informações sobre o evento de causa.

A macro {EVENT.SYMPTOMS} é usada no contexto do evento de causa e retorna informações sobre eventos de sintoma.

Macro	Suportado em	Descrição
{EVENT.CAUSE.ACK.STATUS}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Comandos de ação de evento manual script	*Status de reconhecimento do evento de causa (Sim/Não)*.
{EVENT.CAUSE.AGE}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Comandos de ação de evento manual script	*Idade do evento de causa*, com precisão de até um segundo. Útil em mensagens escalonadas.
{EVENT.CAUSE.DATE}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Comandos de ação de evento manual script	*Data do evento de causa no formato aaaa.mm.dd.*
{EVENT.CAUSE.DURATION}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Comandos de ação de evento manual script	*Duração do evento de causa (diferença de tempo entre os eventos de problema e de recuperação)*, com precisão de até um segundo. Útil em mensagens de recuperação de problema.
{EVENT.CAUSE.ID}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Comandos de ação de evento manual script	*ID numérico do evento de causa.*
{EVENT.CAUSE.NAME}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Comandos de ação de evento manual script	*Nome do evento de problema de causa.*
{EVENT.CAUSE.NSEVERITY}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Comandos de ação de evento manual script	*Valor numérico da severidade do evento de causa.* Valores possíveis: 0 - Não classificado, 1 - Informação, 2 - Aviso, 3 - Médio, 4 - Alto, 5 - Desastre.
{EVENT.CAUSE.OBJECT}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Comandos de ação de evento manual script	*Valor numérico do objeto do evento de causa.* Valores possíveis: 0 - Trigger, 1 - Host descoberto, 2 - Serviço descoberto, 3 - Autoregistro, 4 - Item, 5 - Regra de descoberta de baixo nível.
{EVENT.CAUSE.OPDATA}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Comandos de ação de evento manual script	*Dados operacionais do trigger subjacente do problema de causa.*
{EVENT.CAUSE.SEVERITY}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Comandos de ação de evento manual script	*Nome da severidade do evento de causa.*
{EVENT.CAUSE.SOURCE}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Comandos de ação de evento manual script	*Valor numérico da fonte do evento de causa.* Valores possíveis: 0 - Trigger, 1 - Descoberta, 2 - Autoregistro, 3 - Interno.
{EVENT.CAUSE.STATUS}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Comandos de ação de evento manual script	*Valor verbal do evento de causa.*
{EVENT.CAUSE.TAGS}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Comandos de ação de evento manual script	*Uma lista separada por vírgulas das tags do evento de causa.* Expandido para uma string vazia se não existirem tags.
{EVENT.CAUSE.TAGSJSON}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Comandos de ação de evento manual script	*Um array JSON contendo as tags do evento de causa objetos.* Expandido para um array vazio se não existirem tags.

Macro	Suportado em	Descrição
{EVENT.CAUSE.TAG<1-9>.name}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Comandos de ação de evento manual script	*Valor da tag do evento de causa referenciado pelo nome da tag.* Um nome de tag contendo caracteres não alfanuméricos (incluindo caracteres multibyte-UTF não ingleses) deve ser colocado entre aspas duplas. Aspas e barras invertidas dentro de um nome de tag entre aspas devem ser escapadas com uma barra invertida.
{EVENT.CAUSE.TIME}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Comandos de ação de evento manual script	*Hora do evento de causa no formato hh:mm:ss.*
{EVENT.CAUSE.UPDATE.HISTORY}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Comandos de ação de evento manual script	*Log de atualizações do problema de causa (reconhecimentos, etc).*
{EVENT.CAUSE.VALUE}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Comandos de ação de evento manual script	*Valor numérico do evento de causa (1 para problema, 0 para recuperação).*
{EVENT.SYMPTOMS}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Comandos de ação de evento manual script	*A lista de eventos de sintoma.* Inclui os seguintes detalhes: nome do host, nome do evento, severidade, idade, tags de serviço e valores.

Funções

Macro	Suportado em	Descrição
{FUNCTION.VALUE}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Comandos de script de ação de evento manual → Nomes de eventos	*Resultados da N-ésima função baseada em item na expressão do trigger no momento do evento.* Somente funções com */host/key* como o primeiro parâmetro são contadas. Esta macro pode ser usada com um índice numérico, por exemplo, {FUNCTION.VALUE<1-9>} para apontar para a primeira, segunda, terceira, etc. função em uma expressão de trigger. Veja macros indexadas. **Nota** que antes do Zabbix 7.0.5, {FUNCTION.VALUE} não é suportada; use {FUNCTION.VALUE1} em vez disso.
{FUNCTION.RECOVERY.VALUE}	→ Notificações de recuperação de problema e comandos → Notificações e comandos de atualização de problema → Comandos de script de ação de evento manual	*Resultados da N-ésima função baseada em item na expressão de recuperação no momento do evento.* Somente funções com */host/key* como o primeiro parâmetro são contadas. Esta macro pode ser usada com um índice numérico, por exemplo, {FUNCTION.RECOVERY.VALUE<1-9>} para apontar para a primeira, segunda, terceira, etc. função em uma expressão de trigger. Veja macros indexadas. **Nota** que antes do Zabbix 7.0.5, {FUNCTION.RECOVERY.VALUE} não é suportada; use {FUNCTION.RECOVERY.VALUE1} em vez disso.

Hosts

Macro	Suportado em	Descrição
{HOST.CONN}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Notificações internas → Rótulos de elementos de mapa, nomes e valores de URL de mapa → Parâmetros de chave de item[1] → IP/DNS da interface do host → Campo ”Hosts permitidos” do item trapper → Parâmetros adicionais de monitoramento de banco de dados → Scripts SSH e Telnet → Campo endpoint do item JMX → Monitoramento web[4] → Expressões regulares de filtro de regra de descoberta de baixo nível → Campo URL do widget de dashboard de URL dinâmico → Nomes de trigger, nomes de evento, dados operacionais e descrições → URLs de trigger → Nomes e valores de tags → Nomes e valores de parâmetros de item do tipo Script e item do tipo Navegador, protótipo de item e regra de descoberta → Campos de item do tipo HTTP agent, protótipo de item e regra de descoberta: URL, campos de Query, corpo da requisição, cabeçalhos, arquivo de certificado SSL, arquivo de chave SSL, hosts permitidos. → Comandos de script de ação manual de host, texto de confirmação → Comandos de script de ação manual de evento, texto de confirmação → URLs de ação manual de evento/host, texto de confirmação → Parâmetro *Descrição* no widget *Valor do item* e *Gauge* → Parâmetro *Texto* do rótulo primário/secundário no widget *Honeycomb* → Coluna do tipo de dado *Texto* no widget *Top hosts*	*Endereço IP ou nome DNS do host, dependendo das configurações do host*[2]. Pode ser usado com um índice numérico como {HOST.CONN<1-9>} para apontar para o primeiro, segundo, terceiro, etc. host em uma expressão de trigger. Veja macros indexadas.
{HOST.DESCRIPTION}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Notificações internas → Rótulos de elementos de mapa → Comandos de script de ação manual de evento → Parâmetro *Descrição* no widget *Valor do item* e *Gauge* → Parâmetro *Texto* do rótulo primário/secundário no widget *Honeycomb* → Coluna do tipo de dado *Texto* no widget *Top hosts*	*Descrição do host.* Esta macro pode ser usada com um índice numérico, por exemplo, {HOST.DESCRIPTION<1-9>} para apontar para o primeiro, segundo, terceiro, etc. host em uma expressão de trigger. Veja macros indexadas.

Macro	Suportado em	Descrição
{HOST.DNS}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Notificações internas → Rótulos de elementos de mapa, nomes e valores de URL de mapa → Parâmetros de chave de item[1] → IP/DNS da interface do host → Campo ”Hosts permitidos” do item trapper → Parâmetros adicionais de monitoramento de banco de dados → Scripts SSH e Telnet → Campo endpoint do item JMX → Monitoramento web[4] → Expressões regulares de filtro de regra de descoberta de baixo nível → Campo URL do widget de dashboard de URL dinâmico → Nomes de trigger, nomes de evento, dados operacionais e descrições → URLs de trigger → Nomes e valores de tags → Nomes e valores de parâmetros de item do tipo Script e item do tipo Navegador, protótipo de item e regra de descoberta → Campos de item do tipo HTTP agent, protótipo de item e regra de descoberta: URL, campos de Query, corpo da requisição, cabeçalhos, arquivo de certificado SSL, arquivo de chave SSL, hosts permitidos. → Comandos de script de ação manual de host, texto de confirmação → Comandos de script de ação manual de evento, texto de confirmação → URLs de ação manual de evento/host, texto de confirmação → Parâmetro *Descrição* no widget *Valor do item* e *Gauge* → Parâmetro *Texto* do rótulo primário/secundário no widget *Honeycomb* → Coluna do tipo de dado *Texto* no widget *Top hosts*	*Nome DNS do host*[2]. Esta macro pode ser usada com um índice numérico, por exemplo, {HOST.DNS<1-9>} para apontar para o primeiro, segundo, terceiro, etc. host em uma expressão de trigger. Veja macros indexadas.

Macro	Suportado em	Descrição
{HOST.HOST}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Notificações e comandos de autorregistro → Notificações internas → Parâmetros de chave de item → Rótulos de elementos de mapa, nomes e valores de URL de mapa → IP/DNS da interface do host → Campo ”Hosts permitidos” do item trapper → Parâmetros adicionais de monitoramento de banco de dados → Scripts SSH e Telnet → Campo endpoint do item JMX → Monitoramento web[4] → Expressões regulares de filtro de regra de descoberta de baixo nível → Campo URL do widget de dashboard de URL dinâmico → Nomes de trigger, nomes de evento, dados operacionais e descrições → URLs de trigger → Nomes e valores de tags → Nomes e valores de parâmetros de item do tipo Script e item do tipo Navegador, protótipo de item e regra de descoberta → Campos de item do tipo HTTP agent, protótipo de item e regra de descoberta: URL, campos de Query, corpo da requisição, cabeçalhos, arquivo de certificado SSL, arquivo de chave SSL, hosts permitidos. → Comandos de script de ação manual de host, texto de confirmação → Comandos de script de ação manual de evento, texto de confirmação → URLs de ação manual de evento/host, texto de confirmação → Parâmetro *Descrição* no widget *Valor do item* e *Gauge* → Parâmetro *Texto* do rótulo primário/secundário no widget *Honeycomb* → Coluna do tipo de dado *Texto* no widget *Top hosts*	*Nome do host.* Esta macro pode ser usada com um índice numérico, por exemplo, {HOST.HOST<1-9>} para apontar para o primeiro, segundo, terceiro, etc. host em uma expressão de trigger. Veja macros indexadas. {HOSTNAME<1-9>} está obsoleta e pode não ser resolvida nos locais suportados. Funções de macro não são suportadas para esta macro se ela for usada como um placeholder no primeiro parâmetro de uma função de histórico, por exemplo, `last(/{HOST.HOST}/{ITEM.KEY})`.

Macro	Suportado em	Descrição
{HOST.ID}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Notificações internas → Rótulos de elementos de mapa, nomes e valores de URL de mapa → Campo URL do widget de dashboard de URL dinâmico → URLs de trigger → Nomes e valores de tags → Texto de confirmação de script de ação manual de host → Comandos de script de ação manual de evento, texto de confirmação → URLs de ação manual de evento/host, texto de confirmação → Parâmetro *Descrição* no widget *Valor do item* e *Gauge* → Parâmetro *Texto* do rótulo primário/secundário no widget *Honeycomb* → Coluna do tipo de dado *Texto* no widget *Top hosts*	*ID do host.* Pode ser usado com um índice numérico como {HOST.ID<1-9>} para apontar para o primeiro, segundo, terceiro, etc. host em uma expressão de trigger. Veja macros indexadas.

Macro	Suportado em	Descrição
{HOST.IP}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Notificações e comandos de autorregistro → Notificações internas → Rótulos de elementos de mapa, nomes e valores de URL de mapa → Parâmetros de chave de item[1] → IP/DNS da interface do host → Campo "Hosts permitidos" do item trapper → Parâmetros adicionais de monitoramento de banco de dados → Scripts SSH e Telnet → Campo endpoint do item JMX → Monitoramento web[4] → Expressões regulares de filtro de regra de descoberta de baixo nível → Campo URL do widget de dashboard de URL dinâmico → Nomes de trigger, nomes de evento, dados operacionais e descrições → URLs de trigger → Nomes e valores de tags → Nomes e valores de parâmetros de item do tipo Script e item do tipo Navegador, protótipo de item e regra de descoberta → Campos de item do tipo HTTP agent, protótipo de item e regra de descoberta: URL, campos de Query, corpo da requisição, cabeçalhos, arquivo de certificado SSL, arquivo de chave SSL, hosts permitidos. → Comandos de script de ação manual de host, texto de confirmação → Comandos de script de ação manual de evento, texto de confirmação → URLs de ação manual de evento/host, texto de confirmação → Parâmetro *Descrição* no widget *Valor do item* e *Gauge* → Parâmetro *Texto* do rótulo primário/secundário no widget *Honeycomb* → Coluna do tipo de dado *Texto* no widget *Top hosts*	*Endereço IP do host*[2]. Esta macro pode ser usada com um índice numérico, por exemplo, {HOST.IP<1-9>} para apontar para o primeiro, segundo, terceiro, etc. host em uma expressão de trigger. Veja macros indexadas. {IPADDRESS<1-9>} está obsoleta e pode não ser resolvida nos locais suportados.
{HOST.METADATA}	→ Notificações e comandos de autorregistro	*Metadados do host.* Usado apenas para autorregistro de agent ativo.

Macro	Suportado em	Descrição
{HOST.NAME}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Notificações internas → Rótulos de elementos de mapa, nomes e valores de URL de mapa → Parâmetros de chave de item → IP/DNS da interface do host → Campo ”Hosts permitidos” do item trapper → Parâmetros adicionais de monitoramento de banco de dados → Scripts SSH e Telnet → Monitoramento web[4] → Expressões regulares de filtro de regra de descoberta de baixo nível → Campo URL do widget de dashboard de URL dinâmico → Nomes de trigger, nomes de evento, dados operacionais e descrições → URLs de trigger → Nomes e valores de tags → Nomes e valores de parâmetros de item do tipo Script e item do tipo Navegador, protótipo de item e regra de descoberta → Campos de item do tipo HTTP agent, protótipo de item e regra de descoberta: URL, campos de Query, corpo da requisição, cabeçalhos, arquivo de certificado SSL, arquivo de chave SSL, hosts permitidos. → Comandos de script de ação manual de host, texto de confirmação → Comandos de script de ação manual de evento, texto de confirmação → URLs de ação manual de evento/host, texto de confirmação → Parâmetro *Descrição* no widget *Valor do item* e *Gauge* → Parâmetro *Texto* do rótulo primário/secundário no widget *Honeycomb* → Coluna do tipo de dado *Texto* no widget *Top hosts*	*Nome visível do host.* Esta macro pode ser usada com um índice numérico, por exemplo, {HOST.NAME<1-9>} para apontar para o primeiro, segundo, terceiro, etc. host em uma expressão de trigger. Veja macros indexadas.
{HOST.PORT}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Notificações e comandos de autorregistro → Notificações internas → Nomes de trigger, nomes de evento, dados operacionais e descrições → URLs de trigger → Campo endpoint do item JMX → Nomes e valores de tags → Comandos de script de ação manual de host, texto de confirmação → Comandos de script de ação manual de evento, texto de confirmação → URLs de ação manual de evento/host, texto de confirmação → Parâmetro *Descrição* no widget *Valor do item* e *Gauge* → Parâmetro *Texto* do rótulo primário/secundário no widget *Honeycomb* → Coluna do tipo de dado *Texto* no widget *Top hosts*	*Porta do host (agent)*[2]. Esta macro pode ser usada com um índice numérico, por exemplo, {HOST.PORT<1-9>} para apontar para o primeiro, segundo, terceiro, etc. host em uma expressão de trigger. Veja macros indexadas.

Macro	Suportado em	Descrição
{HOST.TARGET.CONN}	→ Comandos baseados em trigger → Comandos de atualização de problema → Comandos de descoberta → Comandos de autorregistro	*Endereço IP ou nome DNS do host de destino*, dependendo das configurações do host.
{HOST.TARGET.DNS}	→ Comandos baseados em trigger → Comandos de atualização de problema → Comandos de descoberta → Comandos de autorregistro	*Nome DNS do host de destino*.
{HOST.TARGET.HOST}	→ Comandos baseados em trigger → Comandos de atualização de problema → Comandos de descoberta → Comandos de autorregistro	*Nome técnico do host de destino*.
{HOST.TARGET.IP}	→ Comandos baseados em trigger → Comandos de atualização de problema → Comandos de descoberta → Comandos de autorregistro	*Endereço IP do host de destino*.
{HOST.TARGET.NAME}	→ Comandos baseados em trigger → Comandos de atualização de problema → Comandos de descoberta → Comandos de autorregistro	*Nome visível do host de destino*.

Veja também: Inventário do host

Grupos de hosts

Macro	Suportado em	Descrição
{HOSTGROUP.ID}	→ Rótulos de elementos de mapa, nomes e valores de URL de mapa	*ID do grupo de hosts.*

Inventário do host

Macro	Suportado em	Descrição
{INVENTORY.ALIAS}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Notificações internas → Nomes e valores de tags → Rótulos de elementos de mapa, nomes e valores de URL de mapa → Items do tipo Script e items do tipo Navegador[6] → Comandos de script de ação manual de host script, texto de confirmação[6] → Comandos de script de ação manual de evento script, texto de confirmação → URLs de ação manual de evento/host, texto de confirmação → Parâmetro *Descrição* no widget *Valor do item* e *Medidor* → Parâmetro *Texto* do rótulo primário/secundário no widget *Honeycomb* → Coluna do tipo de dado *Texto* no widget *Top hosts*	*Campo de alias no inventário do host.* Esta macro pode ser usada com um índice numérico, por exemplo, {INVENTORY.ALIAS<1-9>} para apontar para o primeiro, segundo, terceiro, etc. host em uma expressão de trigger. Veja macros indexadas.

Macro	Suportado em	Descrição
{INVENTORY.ASSET.TAG}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Notificações internas → Nomes e valores de tags → Rótulos de elementos de mapa, nomes e valores de URL de mapa → Items do tipo Script e items do tipo Navegador[6] → Comandos de script de ação manual de host script, texto de confirmação[6] → Comandos de script de ação manual de evento script, texto de confirmação → URLs de ação manual de evento/host, texto de confirmação → Parâmetro *Descrição* no widget *Valor do item* e *Medidor* → Parâmetro *Texto* do rótulo primário/secundário no widget *Honeycomb* → Coluna do tipo de dado *Texto* no widget *Top hosts*	*Campo de tag de ativo no inventário do host.* Esta macro pode ser usada com um índice numérico, por exemplo, {INVENTORY.ASSET.TAG<1-9>} para apontar para o primeiro, segundo, terceiro, etc. host em uma expressão de trigger. Veja macros indexadas.
{INVENTORY.CHASSIS}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Notificações internas → Nomes e valores de tags → Rótulos de elementos de mapa, nomes e valores de URL de mapa → Items do tipo Script e items do tipo Navegador[6] → Comandos de script de ação manual de host script, texto de confirmação[6] → Comandos de script de ação manual de evento script, texto de confirmação → URLs de ação manual de evento/host, texto de confirmação → Parâmetro *Descrição* no widget *Valor do item* e *Medidor* → Parâmetro *Texto* do rótulo primário/secundário no widget *Honeycomb* → Coluna do tipo de dado *Texto* no widget *Top hosts*	*Campo de chassi no inventário do host.* Esta macro pode ser usada com um índice numérico, por exemplo, {INVENTORY.CHASSIS<1-9>} para apontar para o primeiro, segundo, terceiro, etc. host em uma expressão de trigger. Veja macros indexadas.
{INVENTORY.CONTACT}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Notificações internas → Nomes e valores de tags → Rótulos de elementos de mapa, nomes e valores de URL de mapa → Items do tipo Script e items do tipo Navegador[6] → Comandos de script de ação manual de host script, texto de confirmação[6] → Comandos de script de ação manual de evento script, texto de confirmação → URLs de ação manual de evento/host, texto de confirmação → Parâmetro *Descrição* no widget *Valor do item* e *Medidor* → Parâmetro *Texto* do rótulo primário/secundário no widget *Honeycomb* → Coluna do tipo de dado *Texto* no widget *Top hosts*	*Campo de contato no inventário do host.* Esta macro pode ser usada com um índice numérico, por exemplo, {INVENTORY.CONTACT<1-9>} para apontar para o primeiro, segundo, terceiro, etc. host em uma expressão de trigger. Veja macros indexadas. `{PROFILE.CONTACT<1-9>}` está obsoleta e pode não ser resolvida nos locais suportados.

Macro	Suportado em	Descrição
{INVENTORY.CONTRACT.NUMBER}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Notificações internas → Nomes e valores de tags → Rótulos de elementos de mapa, nomes e valores de URL de mapa → Items do tipo Script e items do tipo Navegador[6] → Comandos de script de ação manual de host script, texto de confirmação[6] → Comandos de script de ação manual de evento script, texto de confirmação → URLs de ação manual de evento/host, texto de confirmação → Parâmetro *Descrição* no widget *Valor do item* e *Medidor* → Parâmetro *Texto* do rótulo primário/secundário no widget *Honeycomb* → Coluna do tipo de dado *Texto* no widget *Top hosts*	*Campo de número de contrato no inventário do host.* Esta macro pode ser usada com um índice numérico, por exemplo, {INVENTORY.CONTRACT.NUMBER<1-9>} para apontar para o primeiro, segundo, terceiro, etc. host em uma expressão de trigger. Veja macros indexadas.
{INVENTORY.DEPLOYMENT.STATUS}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Notificações internas → Nomes e valores de tags → Rótulos de elementos de mapa, nomes e valores de URL de mapa → Items do tipo Script e items do tipo Navegador[6] → Comandos de script de ação manual de host script, texto de confirmação[6] → Comandos de script de ação manual de evento script, texto de confirmação → URLs de ação manual de evento/host, texto de confirmação → Parâmetro *Descrição* no widget *Valor do item* e *Medidor* → Parâmetro *Texto* do rótulo primário/secundário no widget *Honeycomb* → Coluna do tipo de dado *Texto* no widget *Top hosts*	*Campo de status de implantação no inventário do host.* Esta macro pode ser usada com um índice numérico, por exemplo, {INVENTORY.DEPLOYMENT.STATUS<1-9>} para apontar para o primeiro, segundo, terceiro, etc. host em uma expressão de trigger. Veja macros indexadas.
{INVENTORY.HARDWARE}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Notificações internas → Nomes e valores de tags → Rótulos de elementos de mapa, nomes e valores de URL de mapa → Items do tipo Script e items do tipo Navegador[6] → Comandos de script de ação manual de host script, texto de confirmação[6] → Comandos de script de ação manual de evento script, texto de confirmação → URLs de ação manual de evento/host, texto de confirmação → Parâmetro *Descrição* no widget *Valor do item* e *Medidor* → Parâmetro *Texto* do rótulo primário/secundário no widget *Honeycomb* → Coluna do tipo de dado *Texto* no widget *Top hosts*	*Campo de hardware no inventário do host.* Esta macro pode ser usada com um índice numérico, por exemplo, {INVENTORY.HARDWARE<1-9>} para apontar para o primeiro, segundo, terceiro, etc. host em uma expressão de trigger. Veja macros indexadas. `{PROFILE.HARDWARE<1-9>}` está obsoleta e pode não ser resolvida nos locais suportados.

Macro	Suportado em	Descrição
{INVENTORY.HARDWARE.FULL}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Notificações internas → Nomes e valores de tags → Rótulos de elementos de mapa, nomes e valores de URL de mapa → Items do tipo Script e items do tipo Navegador[6] → Comandos de script de ação manual de host script, texto de confirmação[6] → Comandos de script de ação manual de evento script, texto de confirmação → URLs de ação manual de evento/host, texto de confirmação → Parâmetro *Descrição* no widget *Valor do item* e *Medidor* → Parâmetro *Texto* do rótulo primário/secundário no widget *Honeycomb* → Coluna do tipo de dado *Texto* no widget *Top hosts*	*Campo de hardware (detalhes completos) no inventário do host.* Esta macro pode ser usada com um índice numérico, por exemplo, {INVENTORY.HARDWARE.FULL<1-9>} para apontar para o primeiro, segundo, terceiro, etc. host em uma expressão de trigger. Veja macros indexadas.
{INVENTORY.HOST.NETMASK}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Notificações internas → Nomes e valores de tags → Rótulos de elementos de mapa, nomes e valores de URL de mapa → Items do tipo Script e items do tipo Navegador[6] → Comandos de script de ação manual de host script, texto de confirmação[6] → Comandos de script de ação manual de evento script, texto de confirmação → URLs de ação manual de evento/host, texto de confirmação → Parâmetro *Descrição* no widget *Valor do item* e *Medidor* → Parâmetro *Texto* do rótulo primário/secundário no widget *Honeycomb* → Coluna do tipo de dado *Texto* no widget *Top hosts*	*Campo de máscara de sub-rede do host no inventário do host.* Esta macro pode ser usada com um índice numérico, por exemplo, {INVENTORY.HOST.NETMASK<1-9>} para apontar para o primeiro, segundo, terceiro, etc. host em uma expressão de trigger. Veja macros indexadas.
{INVENTORY.HOST.NETWORKS}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Notificações internas → Nomes e valores de tags → Rótulos de elementos de mapa, nomes e valores de URL de mapa → Items do tipo Script e items do tipo Navegador[6] → Comandos de script de ação manual de host script, texto de confirmação[6] → Comandos de script de ação manual de evento script, texto de confirmação → URLs de ação manual de evento/host, texto de confirmação → Parâmetro *Descrição* no widget *Valor do item* e *Medidor* → Parâmetro *Texto* do rótulo primário/secundário no widget *Honeycomb* → Coluna do tipo de dado *Texto* no widget *Top hosts*	*Campo de redes do host no inventário do host.* Esta macro pode ser usada com um índice numérico, por exemplo, {INVENTORY.HOST.NETWORKS<1-9>} para apontar para o primeiro, segundo, terceiro, etc. host em uma expressão de trigger. Veja macros indexadas.

Macro	Suportado em	Descrição
{INVENTORY.HOST.ROUTER}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Notificações internas → Nomes e valores de tags → Rótulos de elementos de mapa, nomes e valores de URL de mapa → Items do tipo Script e items do tipo Navegador[6] → Comandos de script de ação manual de host script, texto de confirmação[6] → Comandos de script de ação manual de evento script, texto de confirmação → URLs de ação manual de evento/host, texto de confirmação → Parâmetro *Descrição* no widget *Valor do item* e *Medidor* → Parâmetro *Texto* do rótulo primário/secundário no widget *Honeycomb* → Coluna do tipo de dado *Texto* no widget *Top hosts*	*Campo de roteador do host no inventário do host.* Esta macro pode ser usada com um índice numérico, por exemplo, {INVENTORY.HOST.ROUTER<1-9>} para apontar para o primeiro, segundo, terceiro, etc. host em uma expressão de trigger. Veja macros indexadas.
{INVENTORY.HW.ARCH}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Notificações internas → Nomes e valores de tags → Rótulos de elementos de mapa, nomes e valores de URL de mapa → Items do tipo Script e items do tipo Navegador[6] → Comandos de script de ação manual de host script, texto de confirmação[6] → Comandos de script de ação manual de evento script, texto de confirmação → URLs de ação manual de evento/host, texto de confirmação → Parâmetro *Descrição* no widget *Valor do item* e *Medidor* → Parâmetro *Texto* do rótulo primário/secundário no widget *Honeycomb* → Coluna do tipo de dado *Texto* no widget *Top hosts*	*Campo de arquitetura de hardware no inventário do host.* Esta macro pode ser usada com um índice numérico, por exemplo, {INVENTORY.HW.ARCH<1-9>} para apontar para o primeiro, segundo, terceiro, etc. host em uma expressão de trigger. Veja macros indexadas.
{INVENTORY.HW.DATE.DECOMM}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Notificações internas → Nomes e valores de tags → Rótulos de elementos de mapa, nomes e valores de URL de mapa → Items do tipo Script e items do tipo Navegador[6] → Comandos de script de ação manual de host script, texto de confirmação[6] → Comandos de script de ação manual de evento script, texto de confirmação → URLs de ação manual de evento/host, texto de confirmação → Parâmetro *Descrição* no widget *Valor do item* e *Medidor* → Parâmetro *Texto* do rótulo primário/secundário no widget *Honeycomb* → Coluna do tipo de dado *Texto* no widget *Top hosts*	*Campo de data de descomissionamento do hardware no inventário do host.* Esta macro pode ser usada com um índice numérico, por exemplo, {INVENTORY.HW.DATE.DECOMM<1-9>} para apontar para o primeiro, segundo, terceiro, etc. host em uma expressão de trigger. Veja macros indexadas.

Macro	Suportado em	Descrição
{INVENTORY.HW.DATE.EXPIRY}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Notificações internas → Nomes e valores de tags → Rótulos de elementos de mapa, nomes e valores de URL de mapa → Items do tipo Script e items do tipo Navegador[6] → Comandos de script de ação manual de host script, texto de confirmação[6] → Comandos de script de ação manual de evento script, texto de confirmação → URLs de ação manual de evento/host, texto de confirmação → Parâmetro *Descrição* no widget *Valor do item* e *Medidor* → Parâmetro *Texto* do rótulo primário/secundário no widget *Honeycomb* → Coluna do tipo de dado *Texto* no widget *Top hosts*	*Campo de data de expiração da manutenção do hardware no inventário do host.* Esta macro pode ser usada com um índice numérico, por exemplo, {INVENTORY.HW.DATE.EXPIRY<1-9>} para apontar para o primeiro, segundo, terceiro, etc. host em uma expressão de trigger. Veja macros indexadas.
{INVENTORY.HW.DATE.INSTALL}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Notificações internas → Nomes e valores de tags → Rótulos de elementos de mapa, nomes e valores de URL de mapa → Items do tipo Script e items do tipo Navegador[6] → Comandos de script de ação manual de host script, texto de confirmação[6] → Comandos de script de ação manual de evento script, texto de confirmação → URLs de ação manual de evento/host, texto de confirmação → Parâmetro *Descrição* no widget *Valor do item* e *Medidor* → Parâmetro *Texto* do rótulo primário/secundário no widget *Honeycomb* → Coluna do tipo de dado *Texto* no widget *Top hosts*	*Campo de data de instalação do hardware no inventário do host.* Esta macro pode ser usada com um índice numérico, por exemplo, {INVENTORY.HW.DATE.INSTALL<1-9>} para apontar para o primeiro, segundo, terceiro, etc. host em uma expressão de trigger. Veja macros indexadas.
{INVENTORY.HW.DATE.PURCHASE}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Notificações internas → Nomes e valores de tags → Rótulos de elementos de mapa, nomes e valores de URL de mapa → Items do tipo Script e items do tipo Navegador[6] → Comandos de script de ação manual de host script, texto de confirmação[6] → Comandos de script de ação manual de evento script, texto de confirmação → URLs de ação manual de evento/host, texto de confirmação → Parâmetro *Descrição* no widget *Valor do item* e *Medidor* → Parâmetro *Texto* do rótulo primário/secundário no widget *Honeycomb* → Coluna do tipo de dado *Texto* no widget *Top hosts*	*Campo de data de compra do hardware no inventário do host.* Esta macro pode ser usada com um índice numérico, por exemplo, {INVENTORY.HW.DATE.PURCHASE<1-9>} para apontar para o primeiro, segundo, terceiro, etc. host em uma expressão de trigger. Veja macros indexadas.

Macro	Suportado em	Descrição
{INVENTORY.INSTALLER.NAME}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Notificações internas → Nomes e valores de tags → Rótulos de elementos de mapa, nomes e valores de URL de mapa → Items do tipo Script e items do tipo Navegador[6] → Comandos de script de ação manual de host script, texto de confirmação[6] → Comandos de script de ação manual de evento script, texto de confirmação → URLs de ação manual de evento/host, texto de confirmação → Parâmetro *Descrição* no widget *Valor do item* e *Medidor* → Parâmetro *Texto* do rótulo primário/secundário no widget *Honeycomb* → Coluna do tipo de dado *Texto* no widget *Top hosts*	*Campo de nome do instalador no inventário do host.* Esta macro pode ser usada com um índice numérico, por exemplo, {INVENTORY.INSTALLER.NAME<1-9>} para apontar para o primeiro, segundo, terceiro, etc. host em uma expressão de trigger. Veja macros indexadas.
{INVENTORY.LOCATION}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Notificações internas → Nomes e valores de tags → Rótulos de elementos de mapa, nomes e valores de URL de mapa → Items do tipo Script e items do tipo Navegador[6] → Comandos de script de ação manual de host script, texto de confirmação[6] → Comandos de script de ação manual de evento script, texto de confirmação → URLs de ação manual de evento/host, texto de confirmação → Parâmetro *Descrição* no widget *Valor do item* e *Medidor* → Parâmetro *Texto* do rótulo primário/secundário no widget *Honeycomb* → Coluna do tipo de dado *Texto* no widget *Top hosts*	*Campo de localização no inventário do host.* Esta macro pode ser usada com um índice numérico, por exemplo, {INVENTORY.LOCATION<1-9>} para apontar para o primeiro, segundo, terceiro, etc. host em uma expressão de trigger. Veja macros indexadas. `{PROFILE.LOCATION<1-9>}` está obsoleta e pode não ser resolvida nos locais suportados.
{INVENTORY.LOCATION.LAT}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Notificações internas → Nomes e valores de tags → Rótulos de elementos de mapa, nomes e valores de URL de mapa → Items do tipo Script e items do tipo Navegador[6] → Comandos de script de ação manual de host script, texto de confirmação[6] → Comandos de script de ação manual de evento script, texto de confirmação → URLs de ação manual de evento/host, texto de confirmação → Parâmetro *Descrição* no widget *Valor do item* e *Medidor* widget → Parâmetro *Texto* do rótulo primário/secundário no widget *Honeycomb* widget → Coluna do tipo de dado *Texto* no widget *Top hosts* widget	*Campo de latitude da localização no inventário do host.* Esta macro pode ser usada com um índice numérico, por exemplo, {INVENTORY.LOCATION.LAT<1-9>} para apontar para o primeiro, segundo, terceiro, etc. host em uma expressão de trigger. Veja macros indexadas.

Macro	Suportado em	Descrição
{INVENTORY.LOCATION.LON}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Notificações internas → Nomes e valores de tags → Rótulos de elementos de mapa, nomes e valores de URL de mapa → Items do tipo Script e items do tipo Navegador[6] → Comandos de script de ação manual de host script, texto de confirmação[6] → Comandos de script de ação manual de evento script, texto de confirmação → URLs de ação manual de evento/host, texto de confirmação → Parâmetro *Descrição* no widget *Valor do item* e *Medidor* widget → Parâmetro *Texto* do rótulo primário/secundário no widget *Honeycomb* widget → Coluna do tipo de dado *Texto* no widget *Top hosts* widget	*Campo de longitude da localização no inventário do host.* Esta macro pode ser usada com um índice numérico, por exemplo, {INVENTORY.LOCATION.LON<1-9>} para apontar para o primeiro, segundo, terceiro, etc. host em uma expressão de trigger. Veja macros indexadas.
{INVENTORY.MACADDRESS.A}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Notificações internas → Nomes e valores de tags → Rótulos de elementos de mapa, nomes e valores de URL de mapa → Items do tipo Script e items do tipo Navegador[6] → Comandos de script de ação manual de host script, texto de confirmação[6] → Comandos de script de ação manual de evento script, texto de confirmação → URLs de ação manual de evento/host, texto de confirmação → Parâmetro *Descrição* no widget *Valor do item* e *Medidor* widget → Parâmetro *Texto* do rótulo primário/secundário no widget *Honeycomb* widget → Coluna do tipo de dado *Texto* no widget *Top hosts* widget	*Campo de endereço MAC A no inventário do host.* Esta macro pode ser usada com um índice numérico, por exemplo, {INVENTORY.MACADDRESS.A<1-9>} para apontar para o primeiro, segundo, terceiro, etc. host em uma expressão de trigger. Veja macros indexadas. `{PROFILE.MACADDRESS<1-9>}` está obsoleta e pode não ser resolvida nos locais suportados.
{INVENTORY.MACADDRESS.B}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Notificações internas → Nomes e valores de tags → Rótulos de elementos de mapa, nomes e valores de URL de mapa → Items do tipo Script e items do tipo Navegador[6] → Comandos de script de ação manual de host script, texto de confirmação[6] → Comandos de script de ação manual de evento script, texto de confirmação → URLs de ação manual de evento/host, texto de confirmação → Parâmetro *Descrição* no widget *Valor do item* e *Medidor* widget → Parâmetro *Texto* do rótulo primário/secundário no widget *Honeycomb* widget → Coluna do tipo de dado *Texto* no widget *Top hosts* widget	*Campo de endereço MAC B no inventário do host.* Esta macro pode ser usada com um índice numérico, por exemplo, {INVENTORY.MACADDRESS.B<1-9>} para apontar para o primeiro, segundo, terceiro, etc. host em uma expressão de trigger. Veja macros indexadas.

Macro	Suportado em	Descrição
{INVENTORY.MODEL}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Notificações internas → Nomes e valores de tags → Rótulos de elementos de mapa, nomes e valores de URL de mapa → Items do tipo Script e items do tipo Navegador[6] → Comandos de script de ação manual de host script, texto de confirmação[6] → Comandos de script de ação manual de evento script, texto de confirmação → URLs de ação manual de evento/host, texto de confirmação → Parâmetro *Descrição* no widget *Valor do item* e *Medidor* widget → Parâmetro *Texto* do rótulo primário/secundário no widget *Honeycomb* widget → Coluna do tipo de dado *Texto* no widget *Top hosts* widget	*Campo de modelo no inventário do host.* Esta macro pode ser usada com um índice numérico, por exemplo, {INVENTORY.MODEL<1-9>} para apontar para o primeiro, segundo, terceiro, etc. host em uma expressão de trigger. Veja macros indexadas.
{INVENTORY.NAME}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Notificações internas → Nomes e valores de tags → Rótulos de elementos de mapa, nomes e valores de URL de mapa → Items do tipo Script e items do tipo Navegador[6] → Comandos de script de ação manual de host script, texto de confirmação[6] → Comandos de script de ação manual de evento script, texto de confirmação → URLs de ação manual de evento/host, texto de confirmação → Parâmetro *Descrição* no widget *Valor do item* e *Medidor* widget → Parâmetro *Texto* do rótulo primário/secundário no widget *Honeycomb* widget → Coluna do tipo de dado *Texto* no widget *Top hosts* widget	*Campo de nome no inventário do host.* Esta macro pode ser usada com um índice numérico, por exemplo, {INVENTORY.NAME<1-9>} para apontar para o primeiro, segundo, terceiro, etc. host em uma expressão de trigger. Veja macros indexadas. {PROFILE.NAME<1-9>} está obsoleta e pode não ser resolvida nos locais suportados.
{INVENTORY.NOTES}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Notificações internas → Nomes e valores de tags → Rótulos de elementos de mapa, nomes e valores de URL de mapa → Items do tipo Script e items do tipo Navegador[6] → Comandos de script de ação manual de host script, texto de confirmação[6] → Comandos de script de ação manual de evento script, texto de confirmação → URLs de ação manual de evento/host, texto de confirmação → Parâmetro *Descrição* no widget *Valor do item* e *Medidor* widget → Parâmetro *Texto* do rótulo primário/secundário no widget *Honeycomb* widget → Coluna do tipo de dado *Texto* no widget *Top hosts* widget	*Campo de observações no inventário do host.* Esta macro pode ser usada com um índice numérico, por exemplo, {INVENTORY.NOTES<1-9>} para apontar para o primeiro, segundo, terceiro, etc. host em uma expressão de trigger. Veja macros indexadas. {PROFILE.NOTES<1-9>} está obsoleta e pode não ser resolvida nos locais suportados.

Macro	Suportado em	Descrição
{INVENTORY.OOB.IP}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Notificações internas → Nomes e valores de tags → Rótulos de elementos de mapa, nomes e valores de URL de mapa → Items do tipo Script e items do tipo Navegador[6] → Comandos de script de ação manual de host script, texto de confirmação[6] → Comandos de script de ação manual de evento script, texto de confirmação → URLs de ação manual de evento/host, texto de confirmação → Parâmetro *Descrição* no widget *Valor do item* e *Medidor* widget → Parâmetro *Texto* do rótulo primário/secundário no widget *Honeycomb* widget → Coluna do tipo de dado *Texto* no widget *Top hosts* widget	*Campo de endereço IP OOB no inventário do host.* Esta macro pode ser usada com um índice numérico, por exemplo, {INVENTORY.OOB.IP<1-9>} para apontar para o primeiro, segundo, terceiro, etc. host em uma expressão de trigger. Veja macros indexadas.
{INVENTORY.OOB.NETMASK}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Notificações internas → Nomes e valores de tags → Rótulos de elementos de mapa, nomes e valores de URL de mapa → Items do tipo Script e items do tipo Navegador[6] → Comandos de script de ação manual de host script, texto de confirmação[6] → Comandos de script de ação manual de evento script, texto de confirmação → URLs de ação manual de evento/host, texto de confirmação → Parâmetro *Descrição* no widget *Valor do item* e *Medidor* widget → Parâmetro *Texto* do rótulo primário/secundário no widget *Honeycomb* widget → Coluna do tipo de dado *Texto* no widget *Top hosts* widget	*Campo de máscara de sub-rede OOB no inventário do host.* Esta macro pode ser usada com um índice numérico, por exemplo, {INVENTORY.OOB.NETMASK<1-9>} para apontar para o primeiro, segundo, terceiro, etc. host em uma expressão de trigger. Veja macros indexadas.
{INVENTORY.OOB.ROUTER}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Notificações internas → Nomes e valores de tags → Rótulos de elementos de mapa, nomes e valores de URL de mapa → Items do tipo Script e items do tipo Navegador[6] → Comandos de script de ação manual de host script, texto de confirmação[6] → Comandos de script de ação manual de evento script, texto de confirmação → URLs de ação manual de evento/host, texto de confirmação → Parâmetro *Descrição* no widget *Valor do item* e *Medidor* widget → Parâmetro *Texto* do rótulo primário/secundário no widget *Honeycomb* widget → Coluna do tipo de dado *Texto* no widget *Top hosts* widget	*Campo de roteador OOB no inventário do host.* Esta macro pode ser usada com um índice numérico, por exemplo, {INVENTORY.OOB.ROUTER<1-9>} para apontar para o primeiro, segundo, terceiro, etc. host em uma expressão de trigger. Veja macros indexadas.

Macro	Suportado em	Descrição
{INVENTORY.OS}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Notificações internas → Nomes e valores de tags → Rótulos de elementos de mapa, nomes e valores de URL de mapa → Items do tipo Script e items do tipo Navegador[6] → Comandos de script de ação manual de host script, texto de confirmação[6] → Comandos de script de ação manual de evento script, texto de confirmação → URLs de ação manual de evento/host, texto de confirmação → Parâmetro *Descrição* no widget *Valor do item* e *Medidor* widget → Parâmetro *Texto* do rótulo primário/secundário no widget *Honeycomb* widget → Coluna do tipo de dado *Texto* no widget *Top hosts* widget	*Campo de SO no inventário do host.* Esta macro pode ser usada com um índice numérico, por exemplo, {INVENTORY.OS<1-9>} para apontar para o primeiro, segundo, terceiro, etc. host em uma expressão de trigger. Veja macros indexadas. {PROFILE.OS<1-9>} está obsoleta e pode não ser resolvida nos locais suportados.
{INVENTORY.OS.FULL}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Notificações internas → Nomes e valores de tags → Rótulos de elementos de mapa, nomes e valores de URL de mapa → Items do tipo Script e items do tipo Navegador[6] → Comandos de script de ação manual de host script, texto de confirmação[6] → Comandos de script de ação manual de evento script, texto de confirmação → URLs de ação manual de evento/host, texto de confirmação → Parâmetro *Descrição* no widget *Valor do item* e *Medidor* widget → Parâmetro *Texto* do rótulo primário/secundário no widget *Honeycomb* widget → Coluna do tipo de dado *Texto* no widget *Top hosts* widget	*Campo de SO (detalhes completos) no inventário do host.* Esta macro pode ser usada com um índice numérico, por exemplo, {INVENTORY.OS.FULL<1-9>} para apontar para o primeiro, segundo, terceiro, etc. host em uma expressão de trigger. Veja macros indexadas.
{INVENTORY.OS.SHORT}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Notificações internas → Nomes e valores de tags → Rótulos de elementos de mapa, nomes e valores de URL de mapa → Items do tipo Script e items do tipo Navegador[6] → Comandos de script de ação manual de host script, texto de confirmação[6] → Comandos de script de ação manual de evento script, texto de confirmação → URLs de ação manual de evento/host, texto de confirmação → Parâmetro *Descrição* no widget *Valor do item* e *Medidor* widget → Parâmetro *Texto* do rótulo primário/secundário no widget *Honeycomb* widget → Coluna do tipo de dado *Texto* no widget *Top hosts* widget	*Campo de SO (curto) no inventário do host.* Esta macro pode ser usada com um índice numérico, por exemplo, {INVENTORY.OS.SHORT<1-9>} para apontar para o primeiro, segundo, terceiro, etc. host em uma expressão de trigger. Veja macros indexadas.

Macro	Suportado em	Descrição
{INVENTORY.POC.PRIMARY.CELL}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Notificações internas → Nomes e valores de tags → Rótulos de elementos de mapa, nomes e valores de URL de mapa → Items do tipo Script e items do tipo Navegador[6] → Comandos de script de ação manual de host script, texto de confirmação[6] → Comandos de script de ação manual de evento script, texto de confirmação → URLs de ação manual de evento/host, texto de confirmação → Parâmetro *Descrição* no widget *Valor do item* e *Medidor* widget → Parâmetro *Texto* do rótulo primário/secundário no widget *Honeycomb* widget → Coluna do tipo de dado *Texto* no widget *Top hosts* widget	*Campo de celular do POC primário no inventário do host.* Esta macro pode ser usada com um índice numérico, por exemplo, {INVENTORY.POC.PRIMARY.CELL<1-9>} para apontar para o primeiro, segundo, terceiro, etc. host em uma expressão de trigger. Veja macros indexadas.
{INVENTORY.POC.PRIMARY.EMAIL}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Notificações internas → Nomes e valores de tags → Rótulos de elementos de mapa, nomes e valores de URL de mapa → Items do tipo Script e items do tipo Navegador[6] → Comandos de script de ação manual de host script, texto de confirmação[6] → Comandos de script de ação manual de evento script, texto de confirmação → URLs de ação manual de evento/host, texto de confirmação → Parâmetro *Descrição* no widget *Valor do item* e *Medidor* widget → Parâmetro *Texto* do rótulo primário/secundário no widget *Honeycomb* widget → Coluna do tipo de dado *Texto* no widget *Top hosts* widget	*Campo de e-mail do POC primário no inventário do host.* Esta macro pode ser usada com um índice numérico, por exemplo, {INVENTORY.POC.PRIMARY.EMAIL<1-9>} para apontar para o primeiro, segundo, terceiro, etc. host em uma expressão de trigger. Veja macros indexadas.
{INVENTORY.POC.PRIMARY.NAME}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Notificações internas → Nomes e valores de tags → Rótulos de elementos de mapa, nomes e valores de URL de mapa → Items do tipo Script e items do tipo Navegador[6] → Comandos de script de ação manual de host script, texto de confirmação[6] → Comandos de script de ação manual de evento script, texto de confirmação → URLs de ação manual de evento/host, texto de confirmação → Parâmetro *Descrição* no widget *Valor do item* e *Medidor* widget → Parâmetro *Texto* do rótulo primário/secundário no widget *Honeycomb* widget → Coluna do tipo de dado *Texto* no widget *Top hosts* widget	*Campo de nome do POC primário no inventário do host.* Esta macro pode ser usada com um índice numérico, por exemplo, {INVENTORY.POC.PRIMARY.NAME<1-9>} para apontar para o primeiro, segundo, terceiro, etc. host em uma expressão de trigger. Veja macros indexadas.

Macro	Suportado em	Descrição
{INVENTORY.POC.PRIMARY.NOTES}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Notificações internas → Nomes e valores de tags → Rótulos de elementos de mapa, nomes e valores de URL de mapa → Items do tipo Script e items do tipo Navegador[6] → Comandos de script de ação manual de host script, texto de confirmação[6] → Comandos de script de ação manual de evento script, texto de confirmação → URLs de ação manual de evento/host, texto de confirmação → Parâmetro *Descrição* no widget *Valor do item* e *Medidor* widget → Parâmetro *Texto* do rótulo primário/secundário no widget *Honeycomb* widget → Coluna do tipo de dado *Texto* no widget *Top hosts* widget	*Campo de observações do POC primário no inventário do host.* Esta macro pode ser usada com um índice numérico, por exemplo, {INVENTORY.POC.PRIMARY.NOTES<1-9>} para apontar para o primeiro, segundo, terceiro, etc. host em uma expressão de trigger. Veja macros indexadas.
{INVENTORY.POC.PRIMARY.PHONE.A}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Notificações internas → Nomes e valores de tags → Rótulos de elementos de mapa, nomes e valores de URL de mapa → Items do tipo Script e items do tipo Navegador[6] → Comandos de script de ação manual de host script, texto de confirmação[6] → Comandos de script de ação manual de evento script, texto de confirmação → URLs de ação manual de evento/host, texto de confirmação → Parâmetro *Descrição* no widget *Valor do item* e *Medidor* widget → Parâmetro *Texto* do rótulo primário/secundário no widget *Honeycomb* widget → Coluna do tipo de dado *Texto* no widget *Top hosts* widget	*Campo de telefone A do POC primário no inventário do host.* Esta macro pode ser usada com um índice numérico, por exemplo, {INVENTORY.POC.PRIMARY.PHONE.A<1-9>} para apontar para o primeiro, segundo, terceiro, etc. host em uma expressão de trigger. Veja macros indexadas.
{INVENTORY.POC.PRIMARY.PHONE.B}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Notificações internas → Nomes e valores de tags → Rótulos de elementos de mapa, nomes e valores de URL de mapa → Items do tipo Script e items do tipo Navegador[6] → Comandos de script de ação manual de host script, texto de confirmação[6] → Comandos de script de ação manual de evento script, texto de confirmação → URLs de ação manual de evento/host, texto de confirmação → Parâmetro *Descrição* no widget *Valor do item* e *Medidor* widget → Parâmetro *Texto* do rótulo primário/secundário no widget *Honeycomb* widget → Coluna do tipo de dado *Texto* no widget *Top hosts* widget	*Campo de telefone B do POC primário no inventário do host.* Esta macro pode ser usada com um índice numérico, por exemplo, {INVENTORY.POC.PRIMARY.PHONE.B<1-9>} para apontar para o primeiro, segundo, terceiro, etc. host em uma expressão de trigger. Veja macros indexadas.

Macro	Suportado em	Descrição
{INVENTORY.POC.PRIMARY.SCREEN}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Notificações internas → Nomes e valores de tags → Rótulos de elementos de mapa, nomes e valores de URL de mapa → Items do tipo Script e items do tipo Navegador[6] → Comandos de script de ação manual de host script, texto de confirmação[6] → Comandos de script de ação manual de evento script, texto de confirmação → URLs de ação manual de evento/host, texto de confirmação → Parâmetro *Descrição* no widget *Valor do item* e *Medidor* widget → Parâmetro *Texto* do rótulo primário/secundário no widget *Honeycomb* widget → Coluna do tipo de dado *Texto* no widget *Top hosts* widget	*Campo de nome de tela do POC primário no inventário do host.* Esta macro pode ser usada com um índice numérico, por exemplo, {INVENTORY.POC.PRIMARY.SCREEN<1-9>} para apontar para o primeiro, segundo, terceiro, etc. host em uma expressão de trigger. Veja macros indexadas.
{INVENTORY.POC.SECONDARY.SCREEN}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Notificações internas → Nomes e valores de tags → Rótulos de elementos de mapa, nomes e valores de URL de mapa → Items do tipo Script e items do tipo Navegador[6] → Comandos de script de ação manual de host script, texto de confirmação[6] → Comandos de script de ação manual de evento script, texto de confirmação → URLs de ação manual de evento/host, texto de confirmação → Parâmetro *Descrição* no widget [*	

Items

Macro	Suportado em	Descrição
{ITEM.DESCRIPTION}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Notificações internas → Comandos de script de ação de evento manual → Parâmetro *Descrição* no widget *Valor do item* e *Gauge* → Parâmetro *Texto* do rótulo primário/secundário no widget *Honeycomb*	*Descrição do N-ésimo item na expressão do trigger que causou uma notificação.* Esta macro pode ser usada com um índice numérico, por exemplo, {ITEM.DESCRIPTION<1-9>} para apontar para o primeiro, segundo, terceiro, etc. item em uma expressão de trigger. Veja macros indexadas.
{ITEM.DESCRIPTION.ORIG}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Notificações internas → Comandos de script de ação de evento manual → Parâmetro *Descrição* no widget *Valor do item* e *Gauge* → Parâmetro *Texto* do rótulo primário/secundário no widget *Honeycomb*	*Descrição (com macros não resolvidas) do N-ésimo item na expressão do trigger que causou uma notificação.* Esta macro pode ser usada com um índice numérico, por exemplo, {ITEM.DESCRIPTION.ORIG<1-9>} para apontar para o primeiro, segundo, terceiro, etc. item em uma expressão de trigger. Veja macros indexadas.

Macro	Suportado em	Descrição
{ITEM.ID}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Notificações internas → Nomes e valores de parâmetros de item do tipo Script e item do tipo Navegador, protótipo de item e regra de descoberta → Campos de item do tipo agente HTTP, protótipo de item e regra de descoberta: URL, campos de consulta, corpo da requisição, cabeçalhos, proxy, arquivo de certificado SSL, arquivo de chave SSL → Comandos de script de ação de evento manual → Parâmetro *Descrição* no widget *Valor do item* e *Gauge* → Parâmetro *Texto* do rótulo primário/secundário no widget *Honeycomb*	*ID numérico do N-ésimo item na expressão do trigger que causou uma notificação.* Esta macro pode ser usada com um índice numérico, por exemplo, {ITEM.ID<1-9>} para apontar para o primeiro, segundo, terceiro, etc. item em uma expressão de trigger. Veja macros indexadas.
{ITEM.KEY}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Notificações internas → Nomes e valores de parâmetros de item do tipo Script e item do tipo Navegador, protótipo de item e regra de descoberta → Campos de item do tipo agente HTTP, protótipo de item e regra de descoberta: URL, campos de consulta, corpo da requisição, cabeçalhos, proxy, arquivo de certificado SSL, arquivo de chave SSL → Comandos de script de ação de evento manual → Parâmetro *Descrição* no widget *Valor do item* e *Gauge* → Parâmetro *Texto* do rótulo primário/secundário no widget *Honeycomb*	*Chave do N-ésimo item na expressão do trigger que causou uma notificação.* Esta macro pode ser usada com um índice numérico, por exemplo, {ITEM.KEY<1-9>} para apontar para o primeiro, segundo, terceiro, etc. item em uma expressão de trigger. Veja macros indexadas. `{TRIGGER.KEY}` está obsoleta e pode não ser resolvida nos locais suportados. Funções de macro não são suportadas para esta macro se ela for usada como um placeholder no primeiro parâmetro de uma função de histórico, por exemplo, `last(/{HOST.HOST}/{ITEM.KEY})`.
{ITEM.KEY.ORIG}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Notificações internas → Nomes e valores de parâmetros de item do tipo Script e item do tipo Navegador, protótipo de item e regra de descoberta → Campos de item do tipo agente HTTP, protótipo de item e regra de descoberta: URL, campos de consulta, corpo da requisição, cabeçalhos, proxy, arquivo de certificado SSL, arquivo de chave SSL → Comandos de script de ação de evento manual → Parâmetro *Descrição* no widget *Valor do item* e *Gauge* → Parâmetro *Texto* do rótulo primário/secundário no widget *Honeycomb*	*Chave original (com macros não expandidas) do N-ésimo item na expressão do trigger que causou uma notificação* [4]*.* Esta macro pode ser usada com um índice numérico, por exemplo, {ITEM.KEY.ORIG<1-9>} para apontar para o primeiro, segundo, terceiro, etc. item em uma expressão de trigger. Veja macros indexadas.

Macro	Suportado em	Descrição
{ITEM.LASTVALUE}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Nomes de triggers, nomes de eventos, dados operacionais e descrições → Nomes e valores de tags → URLs de triggers → Comandos de script de ação de evento manual → Parâmetro *Descrição* no widget *Valor do item* e *Gauge* → Parâmetro *Texto* do rótulo primário/secundário no widget *Honeycomb*	*O valor mais recente do N-ésimo item na expressão do trigger que causou uma notificação.* Será resolvido como *UNKNOWN* no frontend se o valor mais recente do histórico tiver sido coletado há mais tempo do que o período definido em *Período máximo de exibição do histórico* (definido na seção de menu Administração→Geral). Quando usado no nome do problema, a macro não será resolvida para o valor mais recente do item ao visualizar eventos de problema; em vez disso, manterá o valor do item do momento em que o problema ocorreu. Quando usado em notificações, em alguns casos a macro pode não ser resolvida para o valor mais recente do item no momento em que o trigger disparou. Por exemplo, se um item receber rapidamente dois valores, ”A” e ”B”, e o trigger disparar para ”A”, as notificações podem mostrar ”B” como o valor mais recente devido a um pequeno atraso de processamento - o valor mais recente do item mudou entre o momento em que o trigger disparou e quando a notificação foi criada. Para evitar isso, você pode usar a macro {ITEM.VALUE}, que é resolvida para o valor no momento em que o trigger dispara, garantindo que o valor correto seja usado na notificação. É um alias para `last(/{HOST.HOST}/{ITEM.KEY})`. O valor resolvido para itens de texto/log é truncado para 20 caracteres pelo frontend nos seguintes locais: - Dados operacionais; - Descrição do trigger; - URLs do trigger; - Rótulos de URL do trigger; - Descrição do widget de valor do item. Para resolver para um valor completo, você pode usar funções de macro, pois nenhum valor é truncado pelo server. Por exemplo: `{{ITEM.LASTVALUE}.regsub("(.*)", \1)}` Esta macro pode ser usada com um índice numérico, por exemplo, {ITEM.LASTVALUE<1-9>} para apontar para o primeiro, segundo, terceiro, etc. item em uma expressão de trigger. Veja macros indexadas.
{ITEM.LOG.AGE}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Nomes de triggers, dados operacionais e descrições → URLs de triggers → Tags de eventos e valores → Comandos de script de ação de evento manual → Parâmetro *Descrição* no widget *Valor do item* → Parâmetro *Texto* do rótulo primário/secundário no widget *Honeycomb*	*Idade do evento de log*, com precisão de até um segundo. Esta macro pode ser usada com um índice numérico, por exemplo, {ITEM.LOG.AGE<1-9>} para apontar para o primeiro, segundo, terceiro, etc. item em uma expressão de trigger. Veja macros indexadas.

Macro	Suportado em	Descrição
{ITEM.LOG.DATE}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Nomes de triggers, dados operacionais e descrições → URLs de triggers → Tags de eventos e valores → Comandos de script de ação de evento manual → Parâmetro *Descrição* no widget *Valor do item* → Parâmetro *Texto* do rótulo primário/secundário no widget *Honeycomb*	*Data em que a entrada de log foi gravada no log.* Esta macro pode ser usada com um índice numérico, por exemplo, {ITEM.LOG.DATE<1-9>} para apontar para o primeiro, segundo, terceiro, etc. item em uma expressão de trigger. Veja macros indexadas.
{ITEM.LOG.EVENTID}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Nomes de triggers, dados operacionais e descrições → URLs de triggers → Tags de eventos e valores → Comandos de script de ação de evento manual → Parâmetro *Descrição* no widget *Valor do item* → Parâmetro *Texto* do rótulo primário/secundário no widget *Honeycomb*	*ID do evento no log de eventos.* Somente para monitoramento de log de eventos do Windows. Esta macro pode ser usada com um índice numérico, por exemplo, {ITEM.LOG.EVENTID<1-9>} para apontar para o primeiro, segundo, terceiro, etc. item em uma expressão de trigger. Veja macros indexadas.
{ITEM.LOG.NSEVERITY}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Nomes de triggers, dados operacionais e descrições → URLs de triggers → Tags de eventos e valores → Comandos de script de ação de evento manual → Parâmetro *Descrição* no widget *Valor do item* → Parâmetro *Texto* do rótulo primário/secundário no widget *Honeycomb*	*Gravidade numérica do evento no log de eventos.* Somente para monitoramento de log de eventos do Windows. Esta macro pode ser usada com um índice numérico, por exemplo, {ITEM.LOG.NSEVERITY<1-9>} para apontar para o primeiro, segundo, terceiro, etc. item em uma expressão de trigger. Veja macros indexadas.
{ITEM.LOG.SEVERITY}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Nomes de triggers, dados operacionais e descrições → URLs de triggers → Tags de eventos e valores → Comandos de script de ação de evento manual → Parâmetro *Descrição* no widget *Valor do item* → Parâmetro *Texto* do rótulo primário/secundário no widget *Honeycomb*	*Gravidade verbal do evento no log de eventos.* Somente para monitoramento de log de eventos do Windows. Esta macro pode ser usada com um índice numérico, por exemplo, {ITEM.LOG.SEVERITY<1-9>} para apontar para o primeiro, segundo, terceiro, etc. item em uma expressão de trigger. Veja macros indexadas.
{ITEM.LOG.SOURCE}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Nomes de triggers, dados operacionais e descrições → URLs de triggers → Tags de eventos e valores → Comandos de script de ação de evento manual → Parâmetro *Descrição* no widget *Valor do item* → Parâmetro *Texto* do rótulo primário/secundário no widget *Honeycomb*	*Fonte do evento no log de eventos.* Somente para monitoramento de log de eventos do Windows. Esta macro pode ser usada com um índice numérico, por exemplo, {ITEM.LOG.SOURCE<1-9>} para apontar para o primeiro, segundo, terceiro, etc. item em uma expressão de trigger. Veja macros indexadas.

Macro	Suportado em	Descrição
{ITEM.LOG.TIME}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Nomes de triggers, dados operacionais e descrições → URLs de triggers → Tags de eventos e valores → Comandos de script de ação de evento manual → Parâmetro *Descrição* no widget *Valor do item* → Parâmetro *Texto* do rótulo primário/secundário no widget *Honeycomb*	*Hora em que a entrada de log foi gravada no log.* Esta macro pode ser usada com um índice numérico, por exemplo, {ITEM.LOG.TIME<1-9>} para apontar para o primeiro, segundo, terceiro, etc. item em uma expressão de trigger. Veja macros indexadas.
{ITEM.NAME}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Notificações internas → Comandos de script de ação de evento manual → Parâmetro *Descrição* no widget *Valor do item* e *Gauge* → Parâmetro *Texto* do rótulo primário/secundário no widget *Honeycomb*	*Nome do item* com todas as macros resolvidas. Esta macro pode ser usada com um índice numérico, por exemplo, {ITEM.NAME<1-9>} para apontar para o primeiro, segundo, terceiro, etc. item em uma expressão de trigger. Veja macros indexadas.
{ITEM.NAME.ORIG}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Notificações internas → Comandos de script de ação de evento manual → Parâmetro *Descrição* no widget *Valor do item* e *Gauge* → Parâmetro *Texto* do rótulo primário/secundário no widget *Honeycomb*	Esta macro é usada para resolver para o *nome original (ou seja, sem macros resolvidas) do item*. Esta macro pode ser usada com um índice numérico, por exemplo, {ITEM.NAME.ORIG<1-9>} para apontar para o primeiro, segundo, terceiro, etc. item em uma expressão de trigger. Veja macros indexadas.
{ITEM.STATE}	→ Notificações internas baseadas em item → Parâmetro *Descrição* no widget *Valor do item* e *Gauge* → Parâmetro *Texto* do rótulo primário/secundário no widget *Honeycomb*	*O estado mais recente do N-ésimo item na expressão do trigger que causou uma notificação.* Valores possíveis: **Não suportado** e **Normal**. Esta macro pode ser usada com um índice numérico, por exemplo, {ITEM.STATE<1-9>} para apontar para o primeiro, segundo, terceiro, etc. item em uma expressão de trigger. Veja macros indexadas.
{ITEM.STATE.ERROR}	→ Notificações internas baseadas em item	*Mensagem de erro com detalhes do motivo pelo qual um item se tornou não suportado.* Se um item entrar no estado não suportado e, em seguida, imediatamente se tornar suportado novamente, o campo de erro pode ficar vazio.

Macro	Suportado em	Descrição
{ITEM.VALUE}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Nomes de triggers, nomes de eventos, dados operacionais e descrições → Nomes e valores de tags → URLs de triggers → Comandos de script de ação de evento manual → Parâmetro *Descrição* no widget *Valor do item* e *Gauge* → Parâmetro *Texto* do rótulo primário/secundário no widget *Honeycomb*	Resolvido para: - o valor histórico (no momento do evento) do N-ésimo item na expressão do trigger, se usado no contexto de alteração de status do trigger, por exemplo, ao exibir eventos ou enviar notificações; - o valor mais recente do N-ésimo item na expressão do trigger, se usado sem o contexto de alteração de status do trigger, por exemplo, ao exibir uma lista de triggers em uma janela pop-up de seleção. Neste caso, funciona da mesma forma que {ITEM.LASTVALUE}. No primeiro caso, será resolvido como *UNKNOWN* se o valor do histórico já tiver sido excluído ou nunca tiver sido armazenado. No segundo caso, e somente no frontend, será resolvido como *UNKNOWN* se o valor mais recente do histórico tiver sido coletado há mais tempo do que o período definido em *Período máximo de exibição do histórico* (definido na seção de menu Administração→Geral). O valor resolvido para itens de texto/log é truncado para 20 caracteres pelo frontend nos seguintes locais: - Dados operacionais; - Descrição do trigger; - URLs do trigger; - Rótulos de URL do trigger; - Descrição do widget de valor do item. Para resolver para um valor completo, você pode usar funções de macro, pois nenhum valor é truncado pelo server. Por exemplo: `{{ITEM.VALUE}.regsub("(.*)", \1)}` Esta macro pode ser usada com um índice numérico, por exemplo, {ITEM.VALUE<1-9>} para apontar para o primeiro, segundo, terceiro, etc. item em uma expressão de trigger. Veja macros indexadas.
{ITEM.VALUETYPE}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Notificações internas → Comandos de script de ação de evento manual → Parâmetro *Descrição* no widget *Valor do item* e *Gauge* → Parâmetro *Texto* do rótulo primário/secundário no widget *Honeycomb*	*Tipo de valor do N-ésimo item na expressão do trigger que causou uma notificação.* Valores possíveis: 0 - numérico float, 1 - caractere, 2 - log, 3 - numérico sem sinal, 4 - texto. Esta macro pode ser usada com um índice numérico, por exemplo, {ITEM.VALUETYPE<1-9>} para apontar para o primeiro, segundo, terceiro, etc. item em uma expressão de trigger. Veja macros indexadas.

Regras de descoberta de baixo nível

Macro	Suportado em	Descrição
{LLDRULE.DESCRIPTION}	→ Notificações internas baseadas em regras de LLD	*Descrição da regra de descoberta de baixo nível que causou uma notificação.*
{LLDRULE.DESCRIPTION.ORIG}	→ Notificações internas baseadas em regras de LLD	*Descrição (com macros não resolvidas) da regra de descoberta de baixo nível que causou uma notificação.*
{LLDRULE.ID}	→ Notificações internas baseadas em regras de LLD	*ID numérico da regra de descoberta de baixo nível que causou uma notificação.*
{LLDRULE.KEY}	→ Notificações internas baseadas em regras de LLD	*Chave da regra de descoberta de baixo nível que causou uma notificação.*

Macro	Suportado em	Descrição
{LLDRULE.KEY.ORIG}	→ Notificações internas baseadas em regras de LLD	*Chave original (com macros não expandidas) da regra de descoberta de baixo nível que causou uma notificação.*
{LLDRULE.NAME}	→ Notificações internas baseadas em regras de LLD	*Nome da regra de descoberta de baixo nível (com macros resolvidas) que causou uma notificação.*
{LLDRULE.NAME.ORIG}	→ Notificações internas baseadas em regras de LLD	*Nome original (ou seja, sem macros resolvidas) da regra de descoberta de baixo nível que causou uma notificação.*
{LLDRULE.STATE}	→ Notificações internas baseadas em regras de LLD	*O último estado da regra de descoberta de baixo nível.* Valores possíveis: **Não suportado** e **Normal**.
{LLDRULE.STATE.ERROR}	→ Notificações internas baseadas em regras de LLD	*Mensagem de erro com detalhes do motivo pelo qual uma regra de LLD se tornou não suportada.* Se uma regra de LLD entrar no estado não suportado e, em seguida, imediatamente voltar a ser suportada, o campo de erro pode estar vazio.

Mapas

Macro	Suportado em	Descrição
{MAP.ID}	→ Rótulos de elementos do mapa, nomes e valores de URLs do mapa	*ID do mapa de rede.*
{MAP.NAME}	→ Rótulos de elementos do mapa, nomes e valores de URLs do mapa → Campo de texto em formas do mapa	*Nome do mapa de rede.*

Proxies

Macro	Suportado em	Descrição
{PROXY.DESCRIPTION}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Notificações e comandos de descoberta → Notificações e comandos de autorregistro → Notificações internas → Comandos de script de ação de evento manual	*Descrição do proxy*. Resolve para: - proxy do N-ésimo item na expressão do trigger (em notificações baseadas em trigger). Você pode usar macros indexadas aqui. - proxy que executou a descoberta (em notificações de descoberta). Use {PROXY.DESCRIPTION} aqui, sem indexação. - proxy ao qual um agent ativo se registrou (em notificações de autorregistro). Use {PROXY.DESCRIPTION} aqui, sem indexação. Esta macro pode ser usada com um índice numérico, por exemplo, {PROXY.DESCRIPTION<1-9>} para apontar para o primeiro, segundo, terceiro, etc. item em uma expressão de trigger. Veja macros indexadas.

Macro	Suportado em	Descrição
{PROXY.NAME}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Notificações e comandos de descoberta → Notificações e comandos de autorregistro → Notificações internas → Comandos de script de ação de evento manual	*Nome do proxy*. Resolve para: - proxy do N-ésimo item na expressão do trigger (em notificações baseadas em trigger). Você pode usar macros indexadas aqui. - proxy que executou a descoberta (em notificações de descoberta). Use {PROXY.NAME} aqui, sem indexação. - proxy ao qual um agent ativo se registrou (em notificações de autorregistro). Use {PROXY.NAME} aqui, sem indexação. Esta macro pode ser usada com um índice numérico, por exemplo, {PROXY.NAME<1-9>} para apontar para o primeiro, segundo, terceiro, etc. item em uma expressão de trigger. Veja macros indexadas.

Scripts

Macro	Suportado em	Descrição
{MANUALINPUT}	→ Comandos de script de ação manual de host, texto de confirmação → Comandos de script de ação manual de evento, texto de confirmação → URLs de ação manual de evento/host, texto de confirmação	*Valor de entrada manual* especificado pelo usuário no momento da execução do script.

Serviços

Macro	Suportado em	Descrição
{SERVICE.DESCRIPTION}	→ Notificações e comandos baseados em serviço → Notificações e comandos de atualização de serviço	*Descrição do serviço* (com macros resolvidas).
{SERVICE.NAME}	→ Notificações e comandos baseados em serviço → Notificações e comandos de atualização de serviço	*Nome do serviço* (com macros resolvidas).
{SERVICE.ROOTCAUSE}	→ Notificações e comandos baseados em serviço → Notificações e comandos de atualização de serviço	*Lista de eventos de problema de trigger que causaram a falha de um serviço*, ordenados por severidade e nome do host. Inclui os seguintes detalhes: nome do host, nome do evento, severidade, idade, tags do serviço e valores.
{SERVICE.TAGS}	→ Notificações e comandos baseados em serviço → Notificações e comandos de atualização de serviço	*Uma lista separada por vírgulas de tags de eventos de serviço.* As tags de eventos de serviço podem ser definidas na seção de configuração de serviço Tags. Expandido para uma string vazia se não existirem tags.
{SERVICE.TAGSJSON}	→ Notificações e comandos baseados em serviço → Notificações e comandos de atualização de serviço	*Um array JSON contendo objetos de tag de evento de serviço.* As tags de eventos de serviço podem ser definidas na seção de configuração de serviço Tags. Expandido para um array vazio se não existirem tags.
{SERVICE.TAGS.<tag name>}	→ Notificações e comandos baseados em serviço → Notificações e comandos de atualização de serviço	*Valor da tag de evento de serviço referenciado pelo nome da tag.* As tags de eventos de serviço podem ser definidas na seção de configuração de serviço Tags. Um nome de tag contendo caracteres não alfanuméricos (incluindo caracteres multibyte-UTF não ingleses) deve ser colocado entre aspas duplas. Aspas e barras invertidas dentro de um nome de tag entre aspas devem ser escapadas com uma barra invertida.

Macro	Suportado em	Descrição
{TRIGGER.DESCRIPTION}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Notificações internas baseadas em trigger → Comandos de ação de evento manual script	*Descrição do trigger.* Todas as macros suportadas em uma descrição de trigger serão expandidas se {TRIGGER.DESCRIPTION} for usada no texto da notificação. {TRIGGER.COMMENT} está obsoleta e pode não ser resolvida nos locais suportados.
{TRIGGER.EXPRESSION.EXPLAIN}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Comandos de ação de evento manual script → Nomes de eventos	*Expressão de trigger parcialmente avaliada.* Funções baseadas em item são avaliadas e substituídas pelos resultados no momento da geração do evento, enquanto todas as outras funções são exibidas conforme escritas na expressão. Pode ser usada para depuração de expressões de trigger.
{TRIGGER.EXPRESSION.RECOVERY.EXPLAIN}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema e comandos → Notificações e comandos de atualização de problema → Comandos de ação de evento manual script	*Expressão de recuperação de trigger parcialmente avaliada.* Funções baseadas em item são avaliadas e substituídas pelos resultados no momento da geração do evento, enquanto todas as outras funções são exibidas conforme escritas na expressão. Pode ser usada para depuração de expressões de recuperação de trigger.
{TRIGGER.EVENTS.ACK}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Rótulos de elementos de mapa → Comandos de ação de evento manual script	*Número de eventos reconhecidos para um elemento de mapa em mapas, ou para o trigger que gerou o evento atual em notificações.*
{TRIGGER.EVENTS.PROBLEM.ACK}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Rótulos de elementos de mapa → Comandos de ação de evento manual script	*Número de eventos PROBLEM reconhecidos para todos os triggers, desconsiderando seu estado.*
{TRIGGER.EVENTS.PROBLEM.UNACK}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Rótulos de elementos de mapa → Comandos de ação de evento manual script	*Número de eventos PROBLEM não reconhecidos para todos os triggers, desconsiderando seu estado.*
{TRIGGER.EVENTS.UNACK}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Rótulos de elementos de mapa → Comandos de ação de evento manual script	*Número de eventos não reconhecidos para um elemento de mapa em mapas, ou para o trigger que gerou o evento atual em notificações.*
{TRIGGER.HOSTGROUP.NAME}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Notificações internas baseadas em trigger → Comandos de ação de evento manual script	*Uma lista ordenada (por consulta SQL), separada por vírgula e espaço, de grupos de hosts nos quais o trigger está definido.*
{TRIGGER.PROBLEM.EVENTS.ACK}	→ Rótulos de elementos de mapa	*Número de eventos PROBLEM reconhecidos para triggers em estado PROBLEM.*
{TRIGGER.PROBLEM.EVENTS.UNACK}	→ Rótulos de elementos de mapa	*Número de eventos PROBLEM não reconhecidos para triggers em estado PROBLEM.*
{TRIGGER.EXPRESSION}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Notificações internas baseadas em trigger → Comandos de ação de evento manual script	*Expressão do trigger.*

Macro	Suportado em	Descrição
{TRIGGER.EXPRESSION.RECOVERY}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Notificações internas baseadas em trigger → Comandos de ação de evento manual script	*Expressão de recuperação do trigger* se *Geração de evento OK* na configuração do trigger estiver definida como 'Expressão de recuperação'; caso contrário, uma string vazia é retornada.
{TRIGGER.ID}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Notificações internas baseadas em trigger → Rótulos de elementos de mapa, nomes e valores de URL de mapa → URLs de trigger → Valor de tag de trigger → Comandos de ação de evento manual script	*ID numérico do trigger que acionou esta ação.* Suportado em valores de tag de trigger.
{TRIGGER.NAME}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Notificações internas baseadas em trigger → Comandos de ação de evento manual script	*Nome do trigger* (com macros resolvidas). Observe que desde a versão 4.0.0 {EVENT.NAME} pode ser usada em ações para exibir o nome do evento/problema acionado com macros resolvidas.
{TRIGGER.NAME.ORIG}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Notificações internas baseadas em trigger → Comandos de ação de evento manual script	*Nome original do trigger* (ou seja, sem macros resolvidas).
{TRIGGER.NSEVERITY}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Notificações internas baseadas em trigger → Comandos de ação de evento manual script	*Gravidade numérica do trigger.* Valores possíveis: 0 - Não classificado, 1 - Informação, 2 - Aviso, 3 - Médio, 4 - Alto, 5 - Desastre.
{TRIGGER.SEVERITY}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Notificações internas baseadas em trigger → Comandos de ação de evento manual script	*Nome da gravidade do trigger.* Pode ser definido em *Administração → Geral → Opções de exibição de trigger*.
{TRIGGER.STATE}	→ Notificações internas baseadas em trigger	*O último estado do trigger.* Valores possíveis: **Desconhecido** e **Normal**.
{TRIGGER.STATE.ERROR}	→ Notificações internas baseadas em trigger	*Mensagem de erro com detalhes do motivo pelo qual um trigger se tornou não suportado.* Se um trigger entrar no estado não suportado e, em seguida, imediatamente se tornar suportado novamente, o campo de erro pode estar vazio.
{TRIGGER.STATUS}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Comandos de ação de evento manual script	*Valor do trigger no momento da execução da etapa da operação.* Pode ser PROBLEM ou OK. `{STATUS}` está obsoleta e pode não ser resolvida nos locais suportados.
{TRIGGER.TEMPLATE.NAME}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Notificações internas baseadas em trigger → Comandos de ação de evento manual scripts	*Uma lista ordenada (por consulta SQL), separada por vírgula e espaço, de templates nos quais o trigger está definido, ou *UNKNOWN* se o trigger estiver definido em um host.*
{TRIGGER.URL}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Notificações internas baseadas em trigger → Comandos de ação de evento manual script	*URL do trigger.*
{TRIGGER.URL.NAME}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Notificações internas baseadas em trigger → Comandos de ação de evento manual script	*O rótulo para a URL do trigger.*

Macro	Suportado em	Descrição
{TRIGGER.VALUE}	→ Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Expressões de trigger → Comandos de ação de evento manual script	*Valor numérico atual do trigger*: 0 - trigger está em estado OK, 1 - trigger está em estado PROBLEM.
{TRIGGERS.UNACK}	→ Rótulos de elementos de mapa	*Número de triggers não reconhecidos para um elemento de mapa, desconsiderando o estado do trigger.* Um trigger é considerado não reconhecido se pelo menos um de seus eventos PROBLEM não for reconhecido.
{TRIGGERS.PROBLEM.UNACK}	→ Rótulos de elementos de mapa	*Número de triggers PROBLEM não reconhecidos para um elemento de mapa.* Um trigger é considerado não reconhecido se pelo menos um de seus eventos PROBLEM não for reconhecido.
{TRIGGERS.ACK}	→ Rótulos de elementos de mapa	*Número de triggers reconhecidos para um elemento de mapa, desconsiderando o estado do trigger.* Um trigger é considerado reconhecido se todos os seus eventos PROBLEM forem reconhecidos.
{TRIGGERS.PROBLEM.ACK}	→ Rótulos de elementos de mapa	*Número de triggers PROBLEM reconhecidos para um elemento de mapa.* Um trigger é considerado reconhecido se todos os seus eventos PROBLEM forem reconhecidos.

Usuários

Macro	Suportado em	Descrição
{USER.FULLNAME}	→ Notificações e comandos de atualização de problema → Comandos de script de ação manual de host, texto de confirmação → Comandos de script de ação manual de evento, texto de confirmação → URLs de ação manual de evento/host, texto de confirmação	*Nome, sobrenome e nome de usuário do usuário* que adicionou o reconhecimento do evento ou iniciou o script.
{USER.NAME}	→ Comandos de script de ação manual de host, texto de confirmação → Comandos de script de ação manual de evento, texto de confirmação → URLs de ação manual de evento/host, texto de confirmação	*Nome do usuário* que iniciou o script.
{USER.SURNAME}	→ Comandos de script de ação manual de host, texto de confirmação → Comandos de script de ação manual de evento, texto de confirmação → URLs de ação manual de evento/host, texto de confirmação	*Sobrenome do usuário* que iniciou o script.
{USER.USERNAME}	→ Comandos de script de ação manual de host, texto de confirmação → Comandos de script de ação manual de evento, texto de confirmação → URLs de ação manual de evento/host, texto de confirmação	*Nome de usuário do usuário* que iniciou o script. {USER.ALIAS}, suportado antes do Zabbix 5.4.0, agora está obsoleto e pode não ser resolvido nos locais suportados.

Outros tipos de macros

Macro	Suportado em	Descrição
{$MACRO}	→ Veja: Macros de usuário suportadas por local	*Macros definíveis pelo usuário.*
{#MACRO}	→ Veja: Macros de descoberta de baixo nível	*Macros de descoberta de baixo nível.*
{?EXPRESSION}	→ Nomes de eventos de trigger → Notificações e comandos baseados em trigger → Notificações e comandos de atualização de problema → Comandos de script e seus parâmetros de webhook → Rótulos de elementos de mapa[3] → Rótulos de formas em mapas[3] → Rótulos de links em mapas[3] → Nomes de gráficos[5]	Veja macros de expressão.
$1...$9	→ Nomes de trigger → Comandos de parâmetro de usuário	*Macros/referências posicionais.*

Notas de rodapé

[1] As macros {HOST.*} suportadas nos parâmetros de chave de item serão resolvidas para a interface que está selecionada para o item. Quando usadas em itens sem interfaces, elas serão resolvidas para a interface do agent Zabbix, SNMP, JMX ou IPMI do host nesta ordem de prioridade ou para 'UNKNOWN' se o host não tiver nenhuma interface.

[2] Em scripts globais, campos de IP/DNS de interface e cenários web, a macro será resolvida para a interface principal do agent. Se uma interface de agent não for definida, a interface principal SNMP será usada. Se uma interface SNMP também não for definida, a interface principal JMX será usada. Se uma interface JMX também não for definida, a interface principal IPMI será usada. Se o host não tiver nenhuma interface, a macro será resolvida para 'UNKNOWN'.

[3] Apenas uma única função **avg**, **last**, **max**, **min** com segundos como parâmetro (sufixos de tempo suffixes podem ser usados) é suportada nesta macro em rótulos de mapas.

```
#### Exemplo de uso:
{?avg(/{HOST.HOST}/item.key,1h)}

#### Uso não suportado:
{?last(/host/item1)/last(/host/item2)}
{?last(/host/item1)*10}
{?count(/host/item1,5m)}
```

[4] As macros {HOST.*} são suportadas nos campos *Variáveis*, *Cabeçalhos*, *Arquivo de certificado SSL* e *Arquivo de chave SSL* do cenário web e nos campos *URL*, *Post*, *Cabeçalhos* e *String obrigatória* da etapa do cenário. Desde o Zabbix 5.2.2, as macros {HOST.*} não são mais suportadas nos campos *Nome* do cenário web e *Nome* da etapa do cenário web.

[5] Apenas uma única função **avg**, **last**, **max**, **min** com segundos como parâmetro (sufixos de tempo suffixes podem ser usados) é suportada dentro desta macro em nomes de gráficos (veja exemplos na nota de rodapé 3). A macro {HOST.HOST<1-9>} pode ser usada como host dentro da macro (veja também macros indexadas).

[6] Suportado em itens do tipo Script e itens do tipo Browser e scripts de ação manual de host para o Zabbix server e Zabbix proxy.

Macros indexadas

A sintaxe de macro indexada **{MACRO<1-9>}** só pode fazer referência ao N-ésimo item, função ou host no campo *Expressão* de um *trigger*:

- **{HOST.IP1}, {HOST.IP2}, {HOST.IP3}** são resolvidas para os endereços IP do primeiro, segundo e terceiro hosts na expressão do trigger (se presentes).
- **{ITEM.VALUE1}, {ITEM.VALUE2}, {ITEM.VALUE3}** são resolvidas para os valores do primeiro, segundo e terceiro items na expressão do trigger no momento do evento (se presentes).
- **{FUNCTION.VALUE1}, {FUNCTION.VALUE2}, {FUNCTION.VALUE3}** são resolvidas para os valores da primeira, segunda e terceira funções baseadas em item no momento do evento (se presentes).

No contexto de triggers, as macros indexadas sempre se referem ao campo *Expressão* da configuração do trigger, não à *Expressão de recuperação*. Por exemplo, em um evento de recuperação, **{ITEM.VALUE2}** será resolvida para o valor do segundo item da expressão de problema no momento da recuperação.

A macro **{HOST.HOST<1-9>}** também é suportada dentro da macro de expressão {?func(/host/key,param)} em **nomes de gráficos**. Por exemplo, {?func(/{HOST.HOST2}/key,param)} em um nome de gráfico será resolvido para o host do segundo item no gráfico.

Warning:
As macros indexadas não serão resolvidas em nenhum outro contexto além dos casos mencionados aqui. Para outros contextos, use macros sem índice (**{HOST.HOST}, {HOST.IP}**, etc.).

2 Macros de usuário suportadas por local

Visão geral

Esta seção contém uma lista de locais onde macros definíveis pelo usuário são suportadas.

Note:
Apenas macros de usuário em nível global são suportadas para *Ações*, *Descoberta de rede*, *Proxies* e todos os locais listados na seção *Outros locais* desta página. Nos locais mencionados, macros em nível de host e em nível de template não serão resolvidas.

Note:
Para personalizar valores de macros (por exemplo, encurtar ou extrair substrings específicas), você pode usar funções de macro.

Ações

Em ações, macros de usuário podem ser usadas nos seguintes campos:

Localização		Múltiplas macros/misturar com texto[1]
Notificações e comandos baseados em trigger		sim
Notificações internas baseadas em trigger		sim
Notificações de atualização de problema		sim
Notificações e comandos baseados em serviço		sim
Notificações de atualização de serviço		sim
Condição de período de tempo		não
Operações		
	Duração padrão do passo da operação	não
	Duração do passo	não

Hosts/protótipos de host

Na configuração de um host e de um protótipo de host, macros de usuário podem ser usadas nos seguintes campos:

Localização		Múltiplas macros/misturar com texto[1]
IP/DNS da interface		apenas DNS
Porta da interface		não
SNMP v1, v2		
	Comunidade SNMP	sim
SNMP v3		
	Nome do contexto	sim
	Nome de segurança	sim
	Senha de autenticação	sim
	Senha de privacidade	sim
IPMI		
	Nome de usuário	sim
	Senha	sim
Tags[2]		
	Nomes das tags	sim
	Valores das tags	sim

Items / protótipos de item

Em uma configuração de item ou de protótipo de item, macros de usuário podem ser usadas nos seguintes campos:

Localização	Múltiplas macros/misturar com texto[1]
Nome do item	sim
Parâmetros da chave do item	sim
Intervalo de atualização	não
Intervalos personalizados	não
Timeout (disponível para tipos de item suportados)	não
Armazenar até (para histórico e tendências)	não
Descrição	sim
Item calculado/agregado	
Fórmula (constantes de expressão e parâmetros de função; parâmetros da chave do item; (*apenas item agregado*) condições de filtro (nome do grupo de hosts e nome da tag))	sim
Monitor de banco de dados	
Nome de usuário	sim
Senha	sim
Consulta SQL	sim
HTTP agent	
URL[3]	sim
Campos de consulta	sim
Corpo da requisição	sim
Cabeçalhos (nomes e valores)	sim

Localização	Múltiplas macros/misturar com texto[1]
Códigos de status necessários	sim
Proxy HTTP	sim
Nome de usuário de autenticação HTTP	sim
Senha de autenticação HTTP	sim
Arquivo de certificado SSL	sim
Arquivo de chave SSL	sim
Senha da chave SSL	sim
Hosts permitidos	sim
JMX agent	
Endpoint JMX	sim
Item de script	
Nomes e valores dos parâmetros	sim
Item de navegador	
Nomes e valores dos parâmetros	sim
SNMP agent	
OID SNMP	sim
SSH agent	
Nome de usuário	sim
Arquivo de chave pública	sim
Arquivo de chave privada	sim
Senha	sim
Script	sim
TELNET agent	
Nome de usuário	sim
Senha	sim
Script	sim
Zabbix trapper	
Hosts permitidos	sim
Tags[2]	
Nomes das tags	sim
Valores das tags	sim
Etapas de pré-processamento	
Parâmetros (incluindo scripts personalizados)	sim
Parâmetros personalizados de tratamento de erro (*Definir valor para* e campos *Definir erro para*)	sim

Descoberta de baixo nível

Em uma regra de descoberta de baixo nível, macros de usuário podem ser usadas nos seguintes campos:

Localização	Múltiplas macros/misturar com texto[1]
Parâmetros da chave	sim
Intervalo de atualização	não
Intervalo personalizado	não
Timeout (disponível para tipos de item suportados)	não
Excluir recursos perdidos	não
Desabilitar recursos perdidos	não
Descrição	sim
SNMP agent	
SNMP OID	sim
SSH agent	
Nome de usuário	sim
Arquivo de chave pública	sim
Arquivo de chave privada	sim
Senha	sim
Script	sim
TELNET agent	
Nome de usuário	sim
Senha	sim
Script	sim
Zabbix trapper	
Hosts permitidos	sim

Localização	Múltiplas macros/misturar com texto[1]
Database monitor	
Nome de usuário	sim
Senha	sim
Consulta SQL	sim
JMX agent	
JMX endpoint	sim
HTTP agent	
URL[3]	sim
Campos de consulta	sim
Corpo da requisição	sim
Cabeçalhos (nomes e valores)	sim
Códigos de status necessários	sim
Nome de usuário de autenticação HTTP	sim
Senha de autenticação HTTP	sim
Filtros	
Expressão regular	sim
Substituições	
Filtros: expressão regular	sim
Operações: intervalo de atualização (para protótipos de item)	não
Operações: período de armazenamento do histórico (para protótipos de item)	não
Operações: período de armazenamento de tendências (para protótipos de item)	não

Descoberta de rede

Em uma regra de descoberta de rede, macros de usuário podem ser usadas nos seguintes campos:

Localização		Múltiplas macros/misturar com texto[1]
Intervalo de atualização		não
SNMP v1, v2		
	Comunidade SNMP	sim
	OID SNMP	sim
SNMP v3		
	Nome do contexto	sim
	Nome de segurança	sim
	Senha de autenticação	sim
	Senha de privacidade	sim
	OID SNMP	sim

Proxies

Em uma configuração de proxy, macros de usuário podem ser usadas nos seguintes campos:

Localização	Múltiplas macros/misturar com texto[1]
Endereço para agents ativos > *Porta* (se o proxy pertence a um grupo)	não
Endereço e porta da interface (para proxy passivo)	não
Timeouts para tipos de item	não

Grupos de proxy

Em uma configuração de grupo de proxy, macros de usuário podem ser usadas nos seguintes campos:

Localização	Múltiplas macros/misturar com texto[1]
Período de failover	não
Número mínimo de proxies	não

Templates

Em uma configuração de template, macros de usuário podem ser usadas nos seguintes campos:

Localização		Múltiplas macros/misturar com texto[1]
Tags[2]		
	Nomes das tags	sim
	Valores das tags	sim

Triggers

Em uma trigger, macros de usuário podem ser usadas nos seguintes campos:

Localização	Múltiplas macros/mistura com texto[1]
Nome	sim
Dados operacionais	sim
Expressão (apenas em constantes e parâmetros de função; macros secretas não são suportadas)	sim
Tag para correspondência	sim
Nome da entrada do menu	sim

Localização	Múltiplas macros/mistura com texto[1]
URL da entrada do menu[3]	sim
Descrição	sim
Tags[2]	
Nomes das tags	sim
Valores das tags	sim

Cenário web

Em uma configuração de cenário web, macros de usuário podem ser usadas nos seguintes campos:

Localização		Múltiplas macros/misturar com texto[1]
Nome		sim
Intervalo de atualização		não
Agent		sim
Proxy HTTP		sim
Variáveis (apenas valores)		sim
Cabeçalhos (nomes e valores)		sim
Etapas		
	Nome	sim
	URL[3]	sim
	Variáveis (apenas valores)	sim
	Cabeçalhos (nomes e valores)	sim
	Timeout	não
	String obrigatória	sim
	Códigos de status obrigatórios	não
Autenticação		
	Usuário	sim
	Senha	sim
	Certificado SSL	sim
	Arquivo de chave SSL	sim
	Senha da chave SSL	sim
Tags[2]		
	Nomes das tags	sim
	Valores das tags	sim

Outros locais

Além dos locais listados aqui, as macros de usuário podem ser usadas nos seguintes campos:

Localização	Múltiplas macros/mistura com texto[1]
Scripts globais (URL, script, SSH, Telnet, IPMI), incluindo texto de confirmação	sim
Webhooks	
Script JavaScript	não
Nome do parâmetro do script JavaScript	não
Valor do parâmetro do script JavaScript	sim
Dashboards	
Coluna do tipo de dado *Texto* no widget de dashboard *Top hosts*	sim
Parâmetro *Descrição* nos widgets de dashboard *Item value* e *Gauge*	sim
Parâmetro *Texto* do rótulo primário/secundário no widget de dashboard *Honeycomb*	sim
Parâmetro *URL*[3] no widget de dashboard *URL*	sim
Usuários → *Usuários* → *Mídias*	
Quando ativo	não
Administração → *Geral* → *GUI*	
Horário de trabalho	não
Administração → *Geral* → *Timeouts*	
Timeouts para tipos de item	não
Administração → *Geral* → *Conectores*	
URL	sim
Nome de usuário	sim
Senha	sim
Bearer token	sim
Timeout	não
Proxy HTTP	sim
Arquivo de certificado SSL	sim
Arquivo de chave SSL	sim
Senha da chave SSL	sim

Localização	Múltiplas macros/mistura com texto[1]
Alertas → Tipos de mídia → Modelos de mensagem	
Assunto	sim
Mensagem	sim
Alertas → Tipos de mídia → Script	
Parâmetros do script	sim
Alertas → Tipos de mídia → Tipo de mídia	
Campos *Nome de usuário* e *Senha* para o tipo de mídia *E-mail* (quando *Autenticação* está definida como "Nome de usuário e senha"; macros secretas recomendadas)	sim

Para uma lista completa de todas as macros suportadas no Zabbix, consulte macros suportadas.

Notas de rodapé

[1] Se vários macros em um campo ou macros misturados com texto não forem suportados para o local, um único macro deve preencher todo o campo.

[2] Macros usados em nomes e valores de tags são resolvidos apenas durante o processo de geração de eventos.

[3] URLs que contêm um macro secreto não funcionarão, pois o macro nelas será resolvido como "******".

7 Símbolos de unidade

Visão geral

Trabalhar com valores grandes como 86400, 104857600 ou 1000000 pode ser desafiador e levar a erros. Para simplificar a configuração e melhorar a legibilidade, o Zabbix suporta símbolos de unidade (sufixos) que funcionam como multiplicadores de valor.

Expressões de trigger sem sufixos:

```
last(/host/system.uptime)<86400
avg(/host/system.cpu.load,600s)<10
last(/host/vm.memory.size[available])<20971520
```

Expressões de trigger com sufixos:

```
last(/host/system.uptime)<1d
avg(/host/system.cpu.load,10m)<10
last(/host/vm.memory.size[available])<20M
```

Os sufixos também podem simplificar a configuração de outras entidades, como items, widgets, etc., e ajudar a exibir os valores dos items em um formato legível por humanos.

> **Note:**
> Para saber se um campo de configuração de entidade suporta sufixos, sempre consulte a página relevante para a entidade que está sendo configurada.

Sufixos de tempo

O frontend do Zabbix suporta os seguintes sufixos de tempo na configuração de entidades:

- **s** - segundos *(quando usado, funciona da mesma forma que o valor bruto)*
- **m** - minutos
- **h** - horas
- **d** - dias
- **w** - semanas
- **M** - meses *(funções de tendências apenas)*
- **y** - anos *(funções de tendências apenas)*

Os sufixos de tempo são suportados apenas com números inteiros. Por exemplo, 1h é suportado, mas 1,5h ou 1.5h não é suportado; use 90m em vez disso.

Sufixos de tamanho de memória

O Zabbix suporta os seguintes sufixos de tamanho de memória:

- **K** - kilobytes
- **M** - megabytes
- **G** - gigabytes
- **T** - terabytes

Sufixos de valor do item

Sufixos também podem ser usados para exibir valores numéricos de item em um formato legível por humanos.

Para habilitar isso, use um dos seguintes sufixos no campo *Unidades* ao configurar um item:

- **B** - bytes
- **Bps** - bytes por segundo
- **s** - segundos, exibidos usando até três maiores unidades de tempo diferentes de zero
- **uptime** - tempo decorrido em hh:mm:ss ou N dias, hh:mm:ss
- **unixtime** - timestamp Unix, formatado como yyyy.mm.dd hh:mm:ss

As seguintes regras adicionais se aplicam a como esses sufixos são interpretados e exibidos:

- Para B e Bps, o Zabbix usa conversão em base 2 (1K = 1024B), seguindo o padrão JEDEC.
- Para outras unidades (como Hz, W, etc.), o Zabbix usa conversão em base 10 (1K = 1000).
- Para s (segundos):
  - O formato inclui yyy mmm ddd hhh mmm sss ms; apenas até três maiores unidades de tempo diferentes de zero são exibidas (por exemplo, 1M 10d 4h).
  - Se uma unidade for zero e estiver entre duas unidades diferentes de zero, ela é omitida (por exemplo, 10d 56m em vez de 10d 0h 56m).

Quando *Unidades* são usadas, os seguintes sufixos multiplicadores são aplicados automaticamente aos valores dos itens:

- **K**, **M**, **G**, **T** - quilo, mega, giga, tera
- **P**, **E**, **Z**, **Y** - peta, exa, zetta, yotta *(estes são aplicados apenas no frontend)*

Para evitar a conversão de unidades, use o prefixo ! (por exemplo, !B ou !s).

Os exemplos a seguir mostram como os valores dos itens recebidos são convertidos com base nas unidades especificadas:

```
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:
Antes do Zabbix 4.0, havia uma lista de unidades bloqueadas codificada, composta por `ms`, `rpm`, `RPM`, `%`. Esta lista foi descontinuada, portanto, a maneira correta de evitar a conversão dessas unidades é `!ms`, `!rpm`, `!RPM`, `!%`.

8 Sintaxe do período de tempo

Visão geral

Para definir um período de tempo, o seguinte formato deve ser usado:

`d-d,hh:mm-hh:mm`

onde os símbolos significam o seguinte:

Símbolo	Descrição
d	Dia da semana: 1 - Segunda-feira, 2 - Terça-feira, ..., 7 - Domingo
hh	Horas: 00-24
mm	Minutos: 00-59

Você pode especificar mais de um período de tempo usando um ponto e vírgula (;) como separador:

`d-d,hh:mm-hh:mm;d-d,hh:mm-hh:mm...`

Deixar o período de tempo vazio equivale a 1-7,00:00-24:00, que é o valor padrão.

Attention:
O limite superior de um período de tempo não está incluído. Assim, se você especificar 09:00-18:00, o último segundo incluído no período de tempo é 17:59:59.

1 Descrição

Símbolo	Descrição
d	Dia da semana: 1 - Segunda, 2 - Terça ,... , 7 - Domingo
hh	Horas: 00-24
mm	Minutos: 00-59

2 Padrão

Deixar um período de tempo vazio é o mesmo que definir `01-07,00:00-24:00`, que é o valor padrão.

Attention:
O limite de um período de tempo não é incluído. Assim, se você definir `09:00-18:00` o último momento considerado no período de tempo será `17:59:59`. É assim que é tratado desde o Zabbix 1.8.7, e sempre foi tradado desta forma nos horários de trabalho.

3 Exemplos

Horário de trabalho. Segunda - Sexta das 9:00 às 18:00:

`1-5,09:00-18:00`

Horário de trabalho com final de semana. Segunda - Sexta das 9:00 às 18:00 e sábados e domingos das 10:00 às 16:00:

`1-5,09:00-18:00;6-7,10:00-16:00`

9 Execução de comandos

O Zabbix utiliza funcionalidades comuns para verificações externas, parâmetros de usuário, itens system.run, scripts de alerta personalizados, comandos remotos e scripts globais.

Etapas de execução

Note:
Por padrão, todos os scripts no Zabbix são executados usando o shell *sh*, e não é possível modificar o shell padrão. Para utilizar um shell diferente, você pode empregar uma solução alternativa: criar um arquivo de script e invocar esse script durante a execução do comando.

O comando/script é executado de forma semelhante nas plataformas Unix e Windows:

1. O Zabbix (o processo pai) cria um pipe para comunicação
2. O Zabbix define o pipe como a saída para o processo filho a ser criado
3. O Zabbix cria o processo filho (executa o comando/script)
4. Um novo grupo de processos (no Unix) ou um job (no Windows) é criado para o processo filho
5. O Zabbix lê do pipe até que ocorra o timeout ou ninguém esteja escrevendo na outra extremidade (TODOS os handles/descritores de arquivo foram fechados). Observe que o processo filho pode criar mais processos e sair antes que eles saiam ou fechem o handle/descritor de arquivo.
6. Se o timeout não foi atingido, o Zabbix aguarda até que o processo filho inicial termine ou ocorra o timeout
7. Se o processo filho inicial terminou e o timeout não foi atingido, o Zabbix verifica o código de saída do processo filho inicial e o compara com 0 (um valor diferente de zero é considerado como falha na execução, apenas para scripts de alerta personalizados, comandos remotos e scripts de usuário executados no Zabbix server e Zabbix proxy)
8. Neste ponto, assume-se que tudo foi concluído e toda a árvore de processos (ou seja, o grupo de processos ou o job) é encerrada

Attention:
O Zabbix assume que um comando/script concluiu o processamento quando o processo filho inicial terminou E nenhum outro processo ainda mantém o handle/descritor de arquivo de saída aberto. Quando o processamento é concluído, TODOS os processos criados são encerrados.

Todas as aspas duplas e barras invertidas no comando são escapadas com barras invertidas e o comando é colocado entre aspas duplas.

Verificação do código de saída

Os códigos de saída são verificados com as seguintes condições:

- Apenas para scripts de alerta personalizados, comandos remotos e scripts de usuário executados no Zabbix server e Zabbix proxy.
- Qualquer código de saída diferente de 0 é considerado como falha na execução.
- O conteúdo do erro padrão e da saída padrão para execuções com falha é coletado e está disponível no frontend (onde o resultado da execução é exibido).
- Uma entrada de log adicional pode ser criada para comandos remotos executados no Zabbix agent/proxy ativando o parâmetro LogRemoteCommands no agent/proxy.

Possíveis mensagens do frontend e entradas de log para comandos/scripts com falha:

- Conteúdo do erro padrão e da saída padrão para execuções com falha (se houver).
- ”Process exited with code: N.” (para saída vazia e código de saída diferente de 0).
- ”Process killed by signal: N.” (para processo terminado por um sinal, apenas no Linux).
- ”Process terminated unexpectedly.” (para processo terminado por motivos desconhecidos).

Veja também

- Verificações externas
- Parâmetros de usuário
- Itens system.run
- Scripts de alerta personalizados
- Comandos remotos
- Scripts globais

10 Compatibilidade de versões

Agents suportados

Para ser compatível com o Zabbix 7.0, o agent do Zabbix não deve ser mais antigo que a versão 1.4 e não deve ser mais novo que 7.0.

Pode ser necessário revisar a configuração de agents mais antigos, pois alguns parâmetros foram alterados, por exemplo, parâmetros relacionados ao log para versões anteriores à 3.0.

Para aproveitar ao máximo as funcionalidades mais recentes, métricas, desempenho aprimorado e redução do uso de memória, utilize o agent mais recente suportado.

Notas para Windows XP

- No Windows XP 32 bits, não utilize agents do Zabbix mais recentes que 6.0.x;
- No Windows XP/Server 2003, não utilize templates de agent mais recentes que o Zabbix 4.0.x. Os templates mais recentes usam contadores de desempenho em inglês, que só são suportados a partir do Windows Vista/Server 2008.

Agentes 2 suportados

Agentes Zabbix 2 mais antigos da versão 4.4 em diante são compatíveis com o Zabbix 7.0; o agent 2 do Zabbix não deve ser mais recente que a versão 7.0.

Observe que, ao usar as versões 4.4 e 5.0 do agent 2 do Zabbix, o intervalo padrão de 10 minutos é usado para atualizar os items não suportados.

Para aproveitar ao máximo as funcionalidades mais recentes, métricas, desempenho aprimorado e menor uso de memória, use o agent 2 mais recente suportado.

Proxies Zabbix suportados

Para total compatibilidade com o Zabbix 7.0, os proxies devem corresponder à versão principal do server. Apenas proxies Zabbix 7.0.x são totalmente compatíveis com um server Zabbix 7.0.x.

Proxies desatualizados são parcialmente suportados: eles ainda podem coletar dados e executar scripts, mas não podem receber atualizações de configuração, como novos items.

Em relação ao Zabbix server, os proxies podem ser:

- *Atual* (proxy e server possuem a mesma versão principal);
- *Desatualizado* (a versão do proxy é mais antiga que a do server, mas é parcialmente suportada);
- *Não suportado* (a versão do proxy é mais antiga que a versão LTS anterior do server *ou* a versão do proxy é mais recente que a versão principal do server).

Exemplos:

Versão do server	*Versão atual* do proxy	*Versão desatualizada* do proxy	*Versão não suportada* do proxy
6.4	6.4	6.0, 6.2	Anterior a 6.0; posterior a 6.4
7.0	7.0	6.0, 6.2, 6.4	Anterior a 6.0; posterior a 7.0
7.2	7.2	7.0	Anterior a 7.0; posterior a 7.2

Funcionalidade suportada por proxies:

Versão do proxy	Atualização de dados	Atualização de configuração	Tarefas
Atual	Sim	Sim	Sim
Desatualizado	Sim	Não	Comandos remotos (por exemplo, scripts shell); Verificações imediatas de valores de item (ou seja, *Executar agora*); Nota: Testes de pré-processamento com valor real não são suportados.
Não suportado	Não	Não	Não

Avisos sobre o uso de versões incompatíveis dos daemons do Zabbix são registrados.

Arquivos XML suportados

Arquivos XML não mais antigos que a versão 1.8 são suportados para importação no Zabbix 7.0.

Attention:
No formato de exportação XML, as dependências de trigger são armazenadas apenas pelo nome. Se houver vários triggers com o mesmo nome (por exemplo, com diferentes severidades e expressões) que tenham uma dependência definida entre eles, não será possível importá-los. Essas dependências devem ser removidas manualmente do arquivo XML e adicionadas novamente após a importação.

12 Biblioteca de vínculo dinâmico do Zabbix sender para Windows

Visão geral

Em um ambiente Windows, as aplicações podem enviar dados para o Zabbix server/proxy usando a biblioteca de vínculo dinâmico do Zabbix sender (zabbix_sender.dll) em vez de ter que iniciar um processo externo (zabbix_sender.exe).

zabbix_sender.h e zabbix_sender.lib são necessários para compilar aplicações de usuário com zabbix_sender.dll.

Obtendo

Existem duas maneiras de obter o zabbix_sender.dll.

1. Baixe os arquivos zabbix_sender.h, zabbix_sender.lib e zabbix_sender.dll como um arquivo ZIP.

Ao escolher as opções de download, certifique-se de selecionar "No encryption" em *Encryption* e "Archive" em *Packaging*. Em seguida, baixe o Zabbix agent (não o Zabbix agent 2).

Os arquivos zabbix_sender.h, zabbix_sender.lib e zabbix_sender.dll estarão dentro do arquivo ZIP baixado no diretório `bin\dev`. Extraia os arquivos onde precisar.

2. Compile o zabbix_sender.dll a partir do código-fonte (veja as instruções).

A biblioteca de vínculo dinâmico com os arquivos de desenvolvimento estará localizada no diretório `bin\winXX\dev`. Para usá-la, inclua o arquivo de cabeçalho zabbix_sender.h e faça o link com a biblioteca zabbix_sender.lib.

Veja também

- exemplo de uma utilidade simples do Zabbix sender implementada com a biblioteca de vínculo dinâmico do Zabbix sender para ilustrar o uso da biblioteca;
- arquivo zabbix_sender.h para as funções de interface da biblioteca de vínculo dinâmico do Zabbix sender. Este arquivo contém documentação explicando o propósito de cada função de interface, seus argumentos e valor de retorno.

13 Atualização do monitoramento de serviços

Visão geral No Zabbix 6.0, a funcionalidade de monitoramento de serviços foi significativamente reformulada (consulte O que há de novo no Zabbix 6.0.0 para a lista de alterações).

Esta página descreve como os serviços e SLAs, definidos em versões anteriores do Zabbix, são alterados durante uma atualização para o Zabbix 6.0 ou mais recente.

Serviços Em versões anteriores do Zabbix, os serviços tinham dois tipos de dependências: soft e hard. Após uma atualização, todas as dependências se tornarão iguais.

Se um serviço ”Serviço filho” foi previamente vinculado ao ”Serviço pai 1” por meio de dependência hard e adicionalmente ao ”Serviço pai 2” por meio de dependência soft, após uma atualização o ”Serviço filho” terá dois serviços pai: ”Serviço pai 1” e ”Serviço pai 2”.

O mapeamento baseado em trigger entre problemas e serviços foi substituído pelo mapeamento baseado em tags. No Zabbix 6.0 e mais recente, o formulário de configuração de serviço possui um novo parâmetro *Tags de problema*, que permite especificar um ou vários pares de nome e valor de tag para correspondência de problemas. Os triggers que foram vinculados a um serviço receberão uma nova tag `ServiceLink` : `<ID do trigger>:<nome do trigger>` (o valor da tag será truncado para 32 caracteres). Os serviços vinculados receberão a tag de problema `ServiceLink` com o mesmo valor.

Regras de cálculo de status

O ’Algoritmo de cálculo de status’ será atualizado usando as seguintes regras:

- Não calcular → Definir status como OK
- Problema, se pelo menos um filho tiver um problema → Mais crítico dos serviços filhos
- Problema, se todos os filhos tiverem problemas → Mais crítico se todos os filhos tiverem problemas

SLAs Anteriormente, as metas de SLA precisavam ser definidas separadamente para cada serviço. Desde o Zabbix 6.0, o SLA tornou-se uma entidade separada, que contém informações sobre o agendamento do serviço, o objetivo de nível de serviço (SLO) esperado e os períodos de indisponibilidade a serem excluídos do cálculo. Uma vez configurado, um SLA pode ser atribuído a vários serviços por meio de tags de serviço.

Durante uma atualização:

- SLAs idênticos definidos para cada serviço serão agrupados e um SLA para cada grupo será criado.
- Cada serviço afetado receberá uma tag especial `SLA:<ID>` e a mesma tag será especificada no parâmetro *Tags de serviço* do SLA correspondente.
- O tempo de criação do serviço, uma nova métrica nos relatórios de SLA, será definido como 01/01/2000 00:00 para serviços existentes.

14 Outros problemas

Login e systemd

Recomendamos criando um usuário *zabbix* usuário como usuário do sistema, ou seja, sem capacidade de efetuar login. Alguns usuários ignore esta recomendação e use a mesma conta para fazer login (e. g.usando SSH) para hospedar o Zabbix em execução. Isso pode travar o daemon do Zabbix no log externo. Neste caso, você obterá algo como o seguinte no Zabbix registro do servidor:

```
zabbix_server [27730]: [file:'selfmon.c',line:375] lock failed: [22] Invalid argument
zabbix_server [27716]: [file:'dbconfig.c',line:5266] lock failed: [22] Invalid argument
zabbix_server [27706]: [file:'log.c',line:238] lock failed: [22] Invalid argument
```

e no log do agente Zabbix: zabbix_agentd [27796]: [file:’log.c’,line:238] lock failed: [22] Invalid argument

Isso acontece por causa da configuração padrão do systemd `RemoveIPC=yes` configurado em `/etc/systemd/logind.conf`. Ao sair do sistema os semáforos criados pelo Zabbix anteriormente são removidos o que faz com colidam.

Uma citação da documentação do systemd:

`RemoveIPC=`

```
Controls whether System V and POSIX IPC objects belonging to the user shall be removed when the
user fully logs out. Takes a boolean argument. If enabled, the user may not consume IPC resources
```

`after the last of the user's sessions terminated. This covers System V semaphores, shared memory and message queues, as well as POSIX shared memory and message queues. Note that IPC objects of the root user and other system users are excluded from the effect of this setting. Defaults to "yes".`

Existem 2 soluções para este problema: 1. (recommended) Stop using *zabbix* account for anything else than Zabbix processes, create a dedicated account for other things. 2. (não recomendado) definir `RemoveIPC=no` em `/etc/systemd/logind.conf` e reinicialize o sistema. Observe que `RemoveIPC` é um sistema parâmetro, alterá-lo afetará todo o sistema.

Usando o frontend Zabbix por trás do proxy

Se o frontend do Zabbix for executado atrás do servidor proxy, o caminho do cookie no arquivo de configuração do proxy precisa ser reescrito para corresponder ao caminho com proxy reverso. Veja exemplos abaixo. Se o caminho do cookie não for reescrito, os usuários podem ter problemas de autorização ao tentar entre no frontend do Zabbix.

Configuração de exemplo para nginx

```
# ..
local / {
# ..
proxy_cookie_path /zabbix /;
proxy_pass http://192.168.0.94/zabbix/;
# ..
```

Configuração de exemplo para Apache

```
# ..
ProxyPass "/" http://host/zabbix/
ProxyPassReverse "/" http://host/zabbix/
ProxyPassReverseCookiePath /zabbix/
ProxyPassReverseCookieDomain host zabbix.example.com
# ..
```

15 Comparação entre agent e agent 2

Esta seção descreve as diferenças entre o agent Zabbix e o agent 2 Zabbix.

Parâmetro	agent Zabbix	agent 2 Zabbix
Linguagem de programação	C	Go com algumas partes em C
Daemonização	sim	apenas por systemd (sim no Windows)
Extensões suportadas	Módulos carregáveis personalizados em C.	Plugins personalizados em Go.
Requisitos		
Plataformas suportadas	Linux, IBM AIX, FreeBSD, NetBSD, OpenBSD, HP-UX, Mac OS X, Solaris: 9, 10, 11, Windows: todas as versões desktop e server desde XP	Linux, Windows: todas as versões desktop e server, nas quais uma versão Go suportada pode ser instalada.
Bibliotecas de criptografia suportadas	GnuTLS 3.1.18 e mais recentes OpenSSL 1.0.1, 1.0.2, 1.1.0, 1.1.1, 3.0.x LibreSSL - testado com as versões 2.7.4, 2.8.2 (certas limitações se aplicam, veja a página Criptografia para detalhes).	Linux: OpenSSL 1.0.1 e posteriores. MS Windows: OpenSSL 1.1.1 ou posterior. A biblioteca OpenSSL deve ter suporte a PSK habilitado. LibreSSL não é suportado.
Monitoramento de processos		
Processos	Um processo de checagem ativa separado para cada registro de server/proxy.	Processo único com threads criadas automaticamente. O número máximo de threads é determinado pela variável de ambiente GOMAXPROCS.

Parâmetro	agent Zabbix	agent 2 Zabbix
Métricas	**UNIX**: veja uma lista de items suportados. **Windows**: veja uma lista de items adicionais específicos do Windows.	**UNIX**: Todas as métricas suportadas pelo agent Zabbix. Adicionalmente, o agent 2 fornece solução de monitoramento nativa do Zabbix para: Docker, Memcached, MySQL, PostgreSQL, Redis, systemd e outros alvos de monitoramento - veja uma lista completa de items específicos do agent 2. **Windows**: Todas as métricas suportadas pelo agent Zabbix, e também checagens net.tcp.service* de HTTPS, LDAP. Adicionalmente, o agent 2 fornece solução de monitoramento nativa do Zabbix para: PostgreSQL, Redis.
Concorrência	Checagens ativas para um único server são executadas sequencialmente.	Checagens de diferentes plugins ou múltiplas checagens dentro de um plugin podem ser executadas concorrentemente.
Traps de terceiros	não	sim
Recursos adicionais		
Armazenamento persistente	não	sim
Arquivos persistentes para métricas log*[]	sim (apenas em Unix)	não
Upload de dados de log	Pode ser realizado durante a coleta de logs para liberar o buffer.	A coleta de logs é interrompida quando o buffer está cheio, portanto o parâmetro BufferSize deve ser pelo menos MaxLinesPerSecond x 2.
Altera o usuário em tempo de execução	sim (apenas em sistemas tipo Unix)	não (controlado pelo systemd)
Ciphersuites configuráveis pelo usuário	sim	não

Veja também:

- *Descrição dos processos do Zabbix*: agent Zabbix, agent 2 Zabbix
- *Parâmetros de configuração*: agent Zabbix UNIX / Windows, agent 2 Zabbix UNIX / Windows

16 Exemplos de escape

Visão geral

Esta página fornece exemplos de uso de escape correto ao usar expressões regulares em vários contextos.

> **Note:**
> Ao usar o construtor de expressões de trigger, o escape correto em expressões regulares é adicionado automaticamente.

Exemplos

Macro de usuário com contexto

Expressão regular: `\.+\"[a-z]+<br>` Macro de usuário com contexto: `{$MACRO:regex:"\.+\\"[a-z]+"}`

Observação:

- as barras invertidas não são escapadas;
- as aspas são escapadas.

Função de macro dentro do parâmetro da chave do item

Expressão regular: `.+:(\d+)$`
 Chave do item: `net.tcp.service[tcp,,"{{$ENDPOINT}.regsub(\".+:(\d+)$\",\1)}"]`

Observação:

- a expressão regular dentro da função de macro `regsub` está entre aspas duplas (porque contém parêntese de fechamento);
- as aspas ao redor da expressão regular são escapadas (porque todo o terceiro parâmetro do item está entre aspas duplas);
- o terceiro parâmetro da chave do item está entre aspas duplas porque contém uma vírgula.

Função de macro LLD

Expressão regular: `\.+\"([a-z]+)`
 Macro LLD: `{{#MACRO}.iregsub("\.+\\"([a-z]+)", \1)}`

Observação:

- as barras invertidas não são escapadas;
- as aspas são escapadas.

Função de macro LLD dentro do contexto da macro de usuário

Expressão regular: `\.+\"([a-z]+)`
 Macro LLD: `{{#MACRO}.iregsub("\.+\\"([a-z]+)", \1)}`
 Macro de usuário com contexto: `{$MACRO:"{{#MACRO}.iregsub(\"\.+\\\"[a-z]+\", \1)}"}`

Observação:

- o escape de barra invertida para LLD não muda;
- ao inserir a macro LLD no contexto da macro de usuário, precisamos colocá-la em uma string:

1. Aspas são adicionadas ao redor da expressão da macro;
2. As aspas são escapadas; no total, 3 novas barras invertidas são introduzidas.

Parâmetro de string de função (qualquer)

`concat` é usado como exemplo.

Conteúdo da string: `\.+\"[a-z]+`
 Expressão: `concat("abc", "\\.\\\"[a-z]+")`

Observação:

- Parâmetros de string exigem escape tanto para barras invertidas quanto para aspas.

Função de macro LLD dentro do parâmetro de string da função

Expressão regular: `\.+\"([a-z]+)`
 Macro LLD: `{{#MACRO}.iregsub("\.+\\"([a-z]+)", \1)}`
 Expressão: `concat("abc, "{{#MACRO}.iregsub(\"\\.+\\\\\"([a-z]+)\", \\1)}")`

Observação:

- Parâmetros de string exigem escape tanto para barras invertidas quanto para aspas;
- Outra camada de escape é adicionada, porque a macro será resolvida apenas após a string ser desmarcada;

Macro de usuário com contexto dentro do parâmetro de string da função

Expressão regular: `\.+\"[a-z]+`
 Macro de usuário com contexto: `{$MACRO:regex:"\.+\\"[a-z]+"}`
 Expressão: `concat("abc, "{$MACRO:regex:\"\\.+\\\\\"[a-z]+\"}")`

Observação:

- Assim como no exemplo anterior, uma camada adicional de escape é necessária;
- As barras invertidas e as aspas são escapadas apenas para o escape de nível superior (por ser um parâmetro de string).

Função de macro LLD dentro do contexto da macro de usuário dentro da função

Expressão regular: `\.+\"([a-z])+`
 Macro LLD: `{{#MACRO}.iregsub("\.+\\"([a-z]+)", \1)}`
 Macro de usuário com contexto: `{$MACRO:"{{#MACRO}.iregsub(\"\.+\\\"([a-z]+)\", \1)}"}`
 Expressão: `concat("abc, "{$MACRO:\"{{#MACRO}.iregsub(\\\"\.+\\\\\\\"([a-z]+)\\\", \\1)}\"}")`

Observe as três camadas de escape:

1. Para a função de macro LLD, sem escape de barras invertidas;
2. Para a macro de usuário com contexto, sem escape de barras invertidas;
3. Para o parâmetro de string de uma função, com escape de barras invertidas.

Macro de usuário com contexto apenas dentro da string

Expressão regular: `\.+\"[a-z]+`
 Macro de usuário com contexto: `{$MACRO:regex:"\.+\\"[a-z]+"}`
 Dentro da string de alguma expressão, por exemplo: `func(arg1, arg2, arg3)="{$MACRO:regex:\"\\.+\\\\\"[a-z]+\"}"`

Observação:

- Strings também exigem escape de barras invertidas;
- Strings também exigem escape de aspas;
- Novamente um caso com 2 níveis de escape:

1. Escape para contexto de macro de usuário sem escape de barra invertida;
2. Escape por ser uma string com escape de barra invertida.

23 Guias de referência rápida

Visão geral

Esta seção da documentação contém receitas rápidas para configurar o Zabbix para alguns objetivos de monitoramento comumente necessários.

Ela foi projetada para o novo usuário do Zabbix e pode ser usada como um navegador por outras seções da documentação que contêm informações necessárias para resolver a tarefa.

Os seguintes guias de referência rápida estão disponíveis:

- Monitorar Linux com o agent Zabbix
- Monitorar Windows com o agent Zabbix
- Monitorar Apache via HTTP
- Monitorar MySQL com o agent 2 do Zabbix
- Monitorar VMware com o Zabbix
- Monitorar tráfego de rede com o Zabbix
- Monitorar tráfego de rede com o Zabbix usando verificações ativas
- Monitorar sites com itens Browser
- Monitorar certificados de sites com o agent 2 do Zabbix (passivo)
- Monitorar um switch ou roteador de rede com o Zabbix
- Monitorar o log de eventos do Windows usando verificações ativas

1 Monitorar Linux com o agent do Zabbix

Introdução Esta página orienta você nas etapas necessárias para iniciar o monitoramento básico de máquinas Linux com o Zabbix. As etapas descritas neste tutorial podem ser aplicadas a qualquer sistema operacional baseado em Linux.

Para quem é este guia

Este guia foi desenvolvido para novos usuários do Zabbix e contém o conjunto mínimo de etapas necessárias para habilitar o monitoramento básico da sua máquina Linux. Se você procura opções de personalização avançadas ou precisa de uma configuração mais detalhada, consulte a seção Configuração do manual do Zabbix.

Pré-requisitos

Antes de prosseguir com este guia, você precisa baixar e instalar o Zabbix server e o Zabbix frontend de acordo com as instruções para o seu sistema operacional.

Instalar o agent Zabbix O agent Zabbix é o processo responsável pela coleta de dados.

Verifique a versão do seu server Zabbix:

```
zabbix_server -V
```

Instale o agent Zabbix da mesma versão (recomendado) na máquina Linux que você deseja monitorar. Com base em suas necessidades de monitoramento, pode ser a mesma máquina onde o server Zabbix está instalado ou uma máquina completamente diferente.

Escolha o método de instalação mais adequado:

- Execute como um contêiner Docker - veja a lista de imagens disponíveis no repositório Docker do Zabbix.
- Instale a partir dos pacotes do Zabbix (disponível para Alma Linux, CentOS, Debian, Oracle Linux, Raspberry Pi OS, RHEL, Rocky Linux, SUSE Linux Enterprise Server, Ubuntu).

- Compile a partir do código-fonte.

Configure o Zabbix para monitoramento O agent do Zabbix pode coletar métricas nos modos ativo ou passivo (simultaneamente).

> **Note:**
> Uma verificação passiva é uma simples solicitação de dados. O server ou proxy do Zabbix solicita alguns dados (por exemplo, carga da CPU) e o agent do Zabbix envia o resultado de volta ao server. As verificações ativas exigem um processamento mais complexo. O agent deve primeiro recuperar do(s) server(s) uma lista de items para processamento independente e, em seguida, enviar os dados em lote de volta. Veja Verificações ativas e passivas do agent para mais informações.

Os templates de monitoramento fornecidos pelo Zabbix geralmente oferecem duas alternativas - um template para agent do Zabbix e um template para agent do Zabbix (ativo). Com a primeira opção, o agent coletará métricas no modo passivo. Esses templates fornecerão resultados de monitoramento idênticos, mas usando protocolos de comunicação diferentes.

A configuração adicional do Zabbix depende se você seleciona um template para verificações do agent do Zabbix ativas ou passivas.

Checagens passivas Zabbix agent

1. Abra o arquivo de configuração do Agent na máquina onde o Agent está instalado.

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

2. Adicione o endereço IP ou o nome DNS do seu Zabbix server ao parâmetro Server.

```
Server=192.0.2.22
```

3. Reinicie o Zabbix agent.

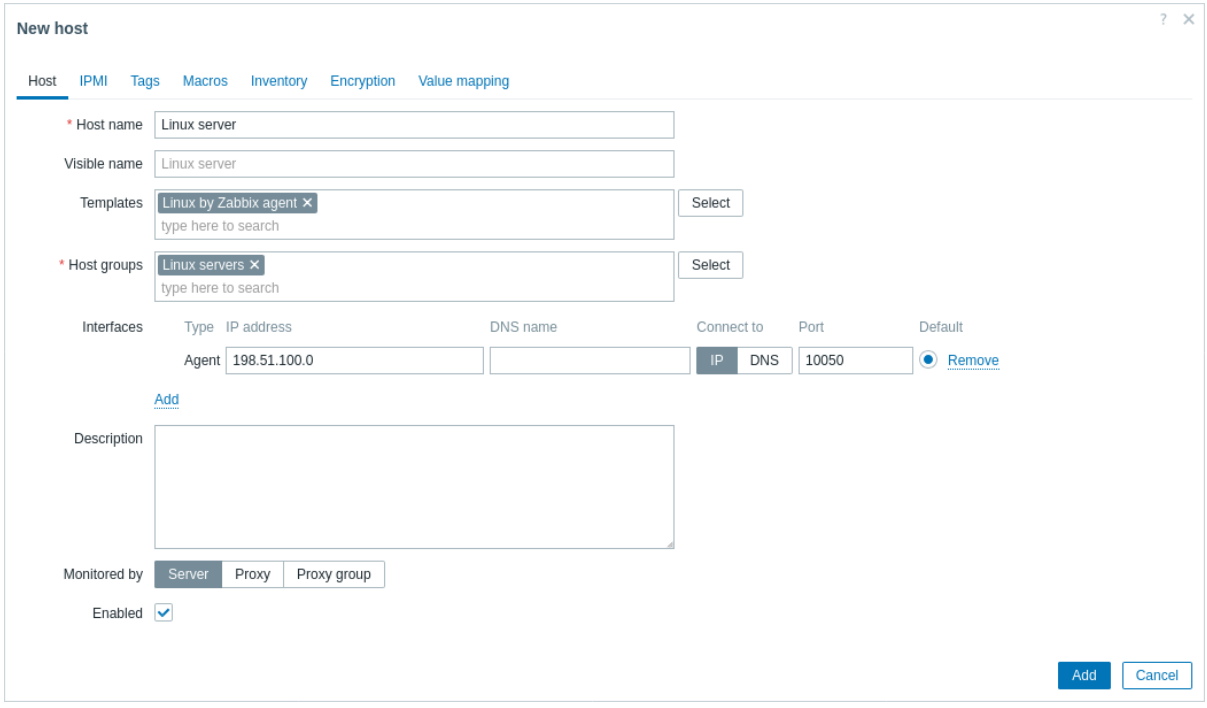
```
systemctl restart zabbix-agent
```

Frontend do Zabbix

1. Faça login no frontend do Zabbix.

2. Crie um host na interface web do Zabbix.

- No campo *Nome do host*, insira um nome de host (por exemplo, ”Servidor Linux”).
- No campo *Templates*, digite ou selecione o template ”Linux by Zabbix agent” que será vinculado ao host.
- No campo *Grupos de hosts*, digite ou selecione um grupo de hosts (por exemplo, ”Servidores Linux”).
- No parâmetro *Interfaces*, adicione a interface *Agent* e especifique o endereço IP ou nome DNS da máquina Linux onde o agent está instalado.



3. Clique em *Adicionar* para adicionar o host. Este host representará a máquina Linux monitorada.

Checagens ativas Zabbix agent

1. Abra o arquivo de configuração do Agent na máquina onde o Agent está instalado.

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

2. Adicione o endereço IP ou nome DNS (e porta opcional) do seu Zabbix server ao parâmetro ServerActive. Por exemplo:

```
ServerActive=192.0.2.0:10051
```

O Zabbix agent usará este endereço para se conectar à porta trapper do Zabbix server (padrão: 10051) e solicitar os dados de configuração para verificações ativas.

3. Defina o parâmetro Hostname, que deve corresponder ao nome do host que será definido no Zabbix frontend. No nosso exemplo, é:

```
Hostname=Linux server
```

O valor de Hostname deve corresponder porque, para verificações ativas, o Zabbix agent o utiliza para recuperar a configuração correta do host do server. Especificamente, o Agent inicia uma conexão com o Server e se identifica usando o valor de Hostname. O Server então fornece a configuração de monitoramento para esse host. Se esses valores forem diferentes, o Agent não receberá a configuração apropriada, resultando em métricas ausentes ou problemas de monitoramento.

4. Reinicie o Zabbix agent.

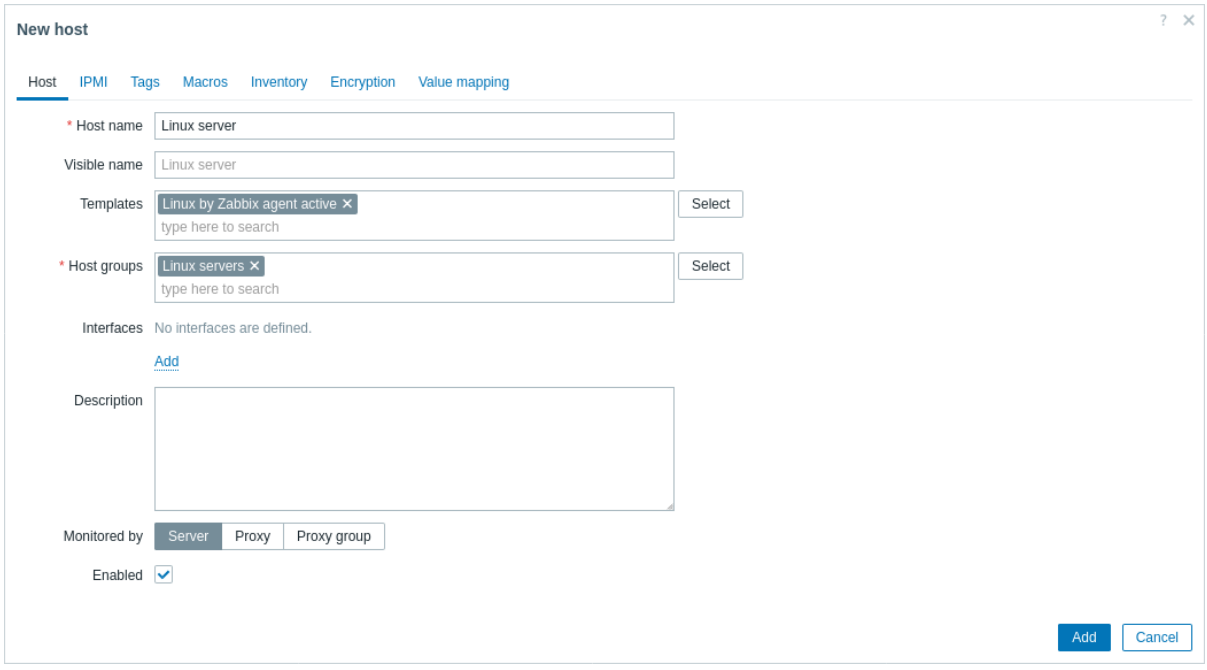
```
systemctl restart zabbix-agent
```

Frontend do Zabbix

1. Faça login no frontend do Zabbix.

2. Crie um host na interface web do Zabbix.

- No campo *Nome do host*, insira um nome de host (por exemplo, "Linux server") que corresponda ao valor do parâmetro *Hostname* definido anteriormente no arquivo de configuração do Agent.
- No campo *Templates*, digite ou selecione o template "Linux by Zabbix agent active" que será vinculado ao host.
- No campo *Grupos de hosts*, digite ou selecione um grupo de hosts (por exemplo, "Linux servers").
- Deixe *Interfaces* indefinido; uma interface não é necessária para checagens ativas porque o Agent inicia a conexão com o server em vez de aguardar uma conexão do server.

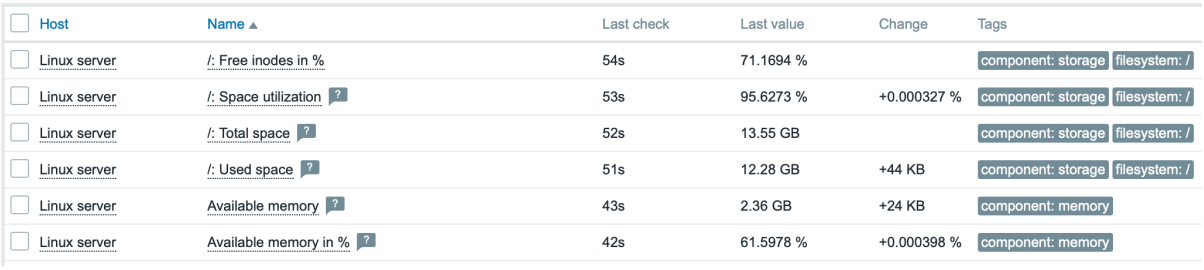


3. Clique em *Adicionar* para adicionar o host. Este host representará a máquina Linux monitorada.

Visualizar métricas coletadas Parabéns! Neste ponto, o Zabbix já está monitorando sua máquina Linux.

Para visualizar as métricas coletadas, abra a seção de menu *Monitoramento->Hosts* menu section e clique em *Últimos dados* ao lado do host.

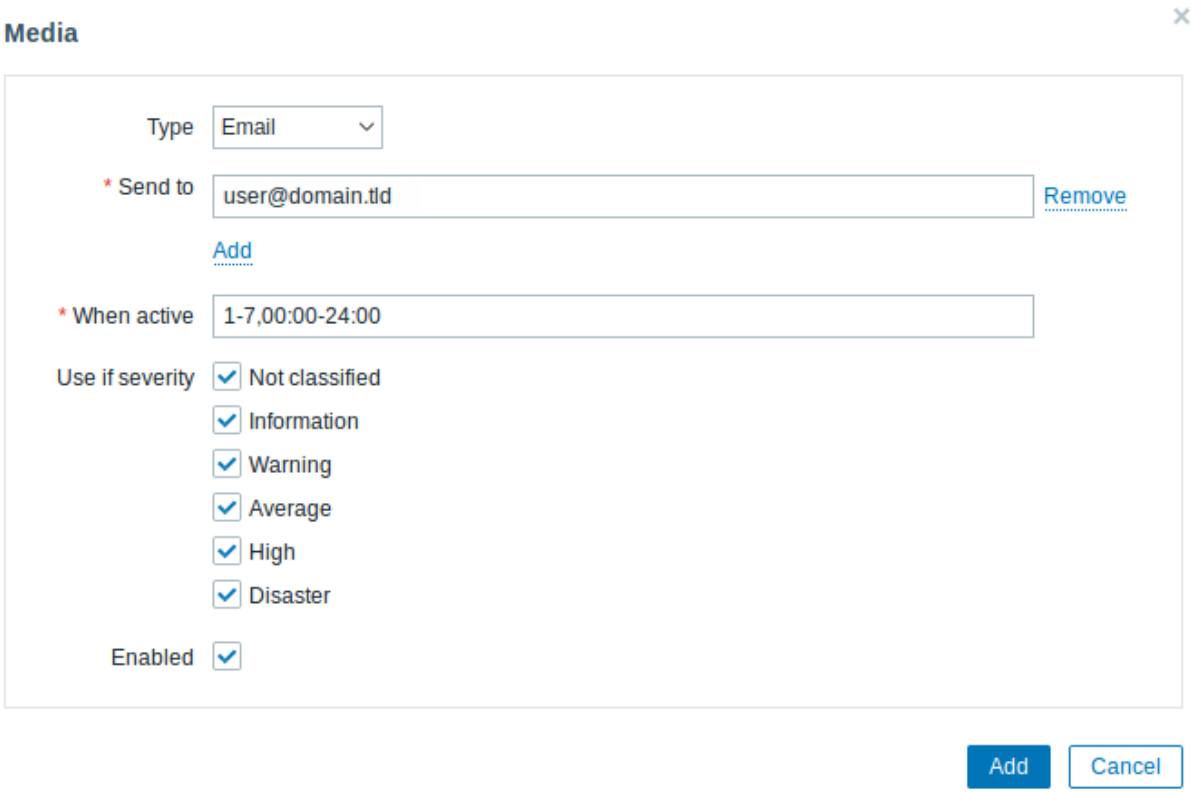
Esta ação abrirá uma lista de todas as métricas mais recentes coletadas do host do servidor Linux.



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

Configurar alertas de problemas O Zabbix pode notificá-lo sobre um problema em sua infraestrutura usando uma variedade de métodos. Este guia fornece etapas de configuração para o envio de alertas por e-mail.

1. Vá para *Configurações do usuário* -> *Perfil*, mude para a aba *Mídia* e adicione seu e-mail.



2. Siga o guia para Receber notificação de problema.

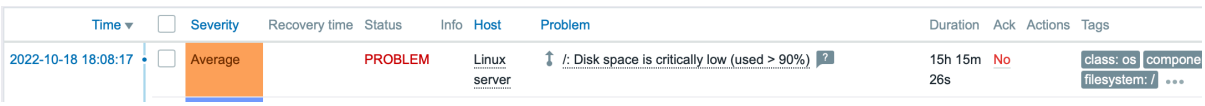
Da próxima vez que o Zabbix detectar um problema, você deverá receber um alerta por e-mail.

Teste sua configuração No Linux, você pode simular uma alta carga de CPU e, como resultado, receber um alerta de problema executando:

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

Pode ser necessário executar vários processos md5sum para que a carga da CPU ultrapasse o limite.

Quando o Zabbix detectar o problema, ele aparecerá na seção Monitoramento->Problemas.



Time ▼	Severity	Recovery time	Status	Info	Host	Problem	Duration	Ack	Actions	Tags
2022-10-18 18:08:17	Average		PROBLEM		Linux server	/: Disk space is critically low (used > 90%)	15h 15m 26s	No		class: os compone filesystem: / ...

Se os alertas estiverem configurados, você também receberá a notificação do problema.

Veja também:

- Criando um item - como começar a monitorar métricas adicionais (monitoramento personalizado sem templates).
- Itens do Zabbix agent, Itens do Zabbix agent para Windows - lista completa de métricas que você pode monitorar usando o Zabbix agent no Windows.

- Escalonamentos de problemas - como criar cenários de alerta em várias etapas (por exemplo, primeiro enviar mensagem para o administrador do sistema, depois, se o problema não for resolvido em 45 minutos, enviar mensagem para o gerente do data center).
- Instalação a partir de pacotes - como instalar os componentes do Zabbix usando pacotes oficiais RPM e DEB para várias distribuições Linux, garantindo acesso aos recursos mais recentes e correções de bugs.

2 Monitor Windows with Zabbix agent

Introdução Esta página orienta você nas etapas necessárias para iniciar o monitoramento básico de máquinas Windows com o Zabbix.

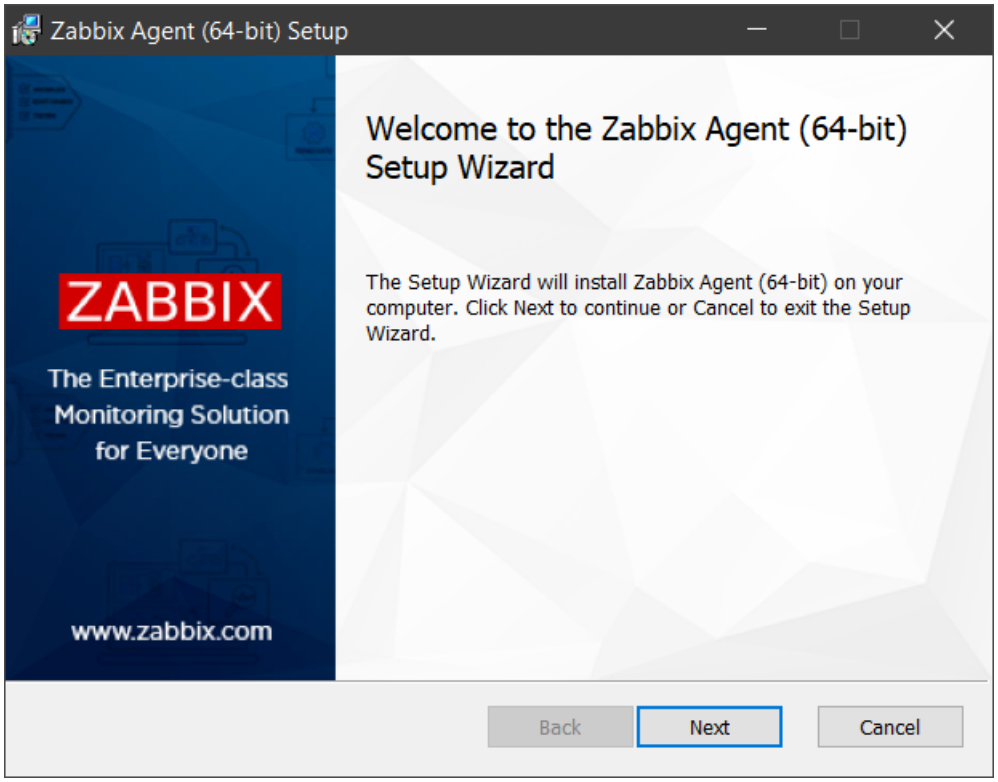
Para quem é este guia

Este guia foi desenvolvido para novos usuários do Zabbix e contém o conjunto mínimo de etapas necessárias para habilitar o monitoramento básico da sua máquina Windows. Se você procura opções de personalização avançadas ou precisa de uma configuração mais detalhada, consulte a seção Configuração do manual do Zabbix.

Pré-requisitos

Antes de prosseguir com este guia de instalação, você deve baixar e instalar o Zabbix server e o Zabbix frontend de acordo com as instruções para o seu sistema operacional.

Instalar o agent do Zabbix O agent do Zabbix é o processo responsável pela coleta de dados. Você precisa instalá-lo na máquina Windows que deseja monitorar. Siga as instruções de instalação do agent do Zabbix para Windows.



Configure o Zabbix para monitoramento O agent do Zabbix pode coletar métricas nos modos ativo ou passivo (simultaneamente).

Note:
Uma verificação passiva é uma simples solicitação de dados. O server ou proxy do Zabbix solicita alguns dados (por exemplo, carga da CPU) e o agent do Zabbix envia o resultado de volta ao server. As verificações ativas exigem um processamento mais complexo. O agent deve primeiro recuperar do(s) server(s) uma lista de items para processamento independente e, em seguida, enviar os dados em lote de volta. Veja Verificações ativas e passivas do agent para mais informações.

Os templates de monitoramento fornecidos pelo Zabbix geralmente oferecem duas alternativas - um template para agent do Zabbix e um template para agent do Zabbix (ativo). Com a primeira opção, o agent coletará métricas no modo passivo. Esses templates fornecerão resultados de monitoramento idênticos, mas usando protocolos de comunicação diferentes.

A configuração adicional do Zabbix depende se você seleciona um template para verificações do agent do Zabbix ativas ou passivas.

Checagens passivas Zabbix frontend

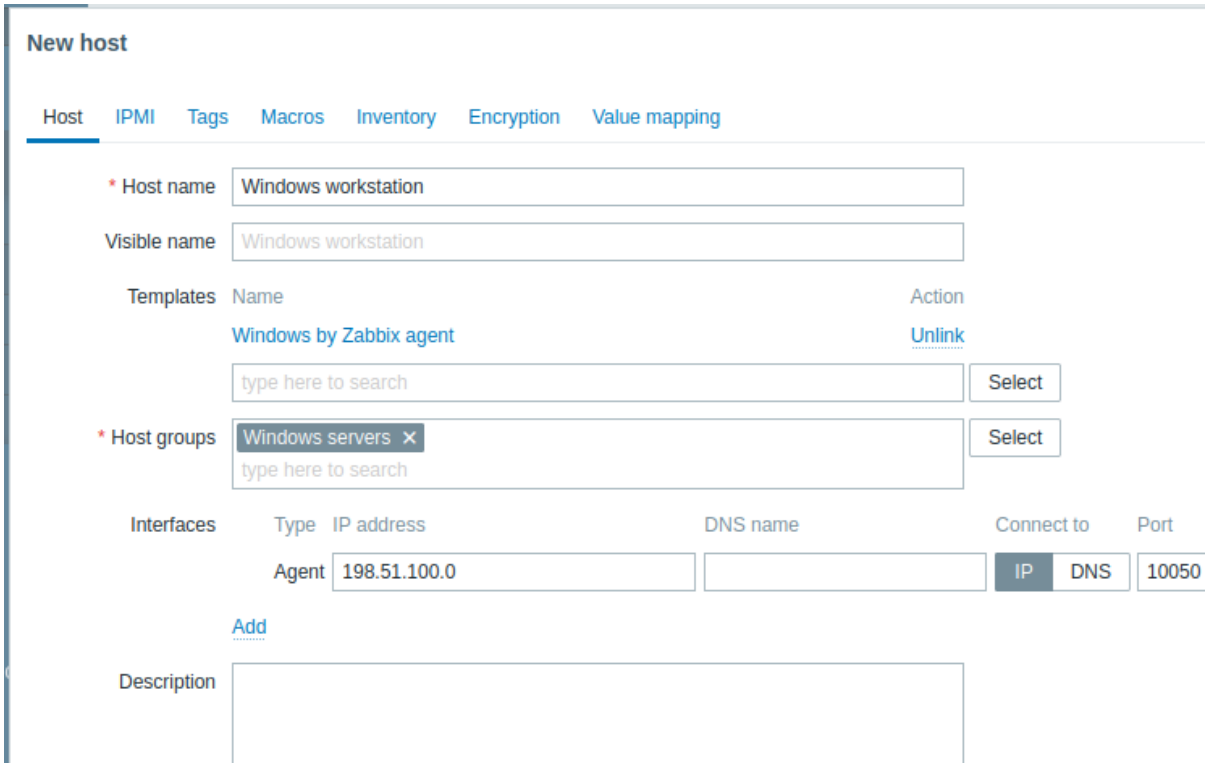
1. Log into Zabbix frontend.

2. Create a host in Zabbix web interface.

This host will represent your Windows machine.

3. In the *Interfaces* parameter, add *Agent* interface and specify the IP address or DNS name of the Windows machine where the agent is installed.

4. In the *Templates* parameter, type or select *Windows by Zabbix agent*.



Zabbix agent

For passive checks Zabbix agent needs to know the IP address or DNS name of Zabbix server. If you have provided correct information during the agent installation, the configuration file is already updated. Otherwise, you need to manually specify it. Go to the `C:\Program files\Zabbix Agent` folder, open the file *zabbix_agentd.conf* and add the IP/DNS of your Zabbix server to the *Server* parameter.

Example:

```
Server=192.0.2.22
```

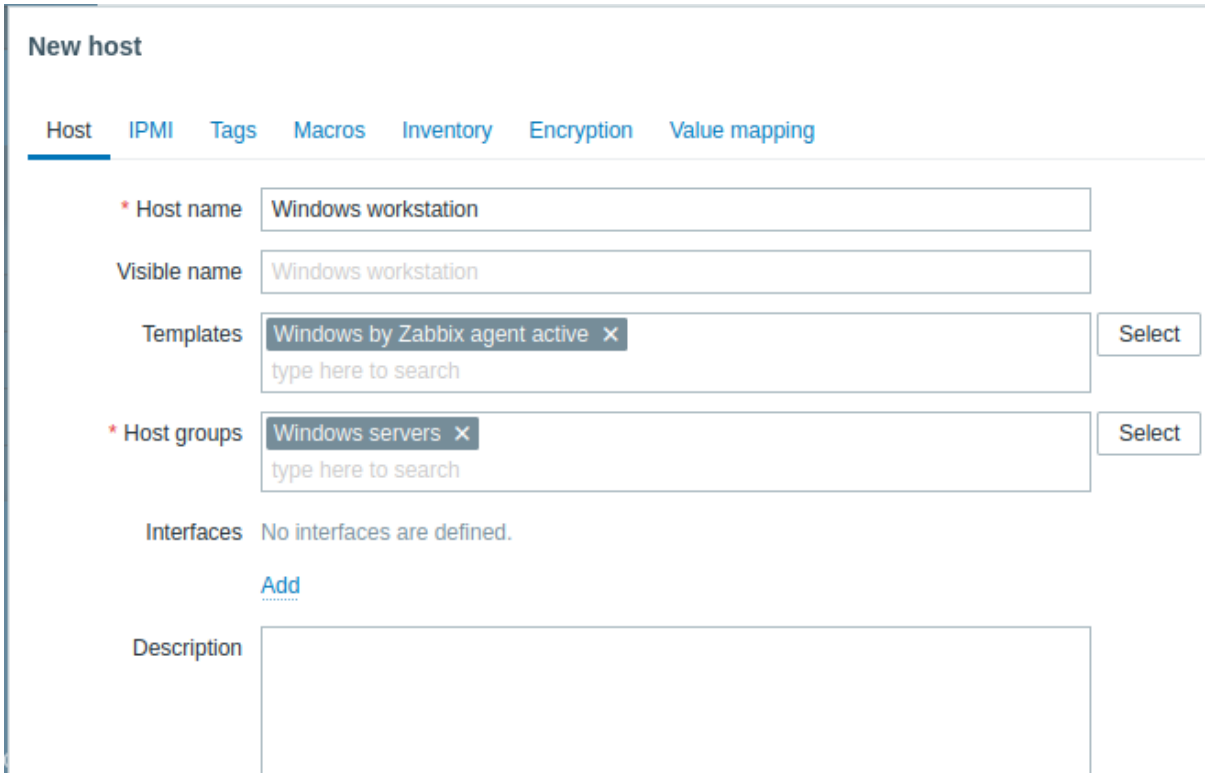
Checagens ativas Frontend do Zabbix

1. Faça login no frontend do Zabbix.

2. Crie um host na interface web do Zabbix.

Este host representará sua máquina Windows.

3. No parâmetro *Templates*, digite ou selecione *Windows by Zabbix agent active*.



Zabbix agent

In the `C:\Program files\Zabbix Agent` folder open the file *zabbix_agentd.conf* and add:

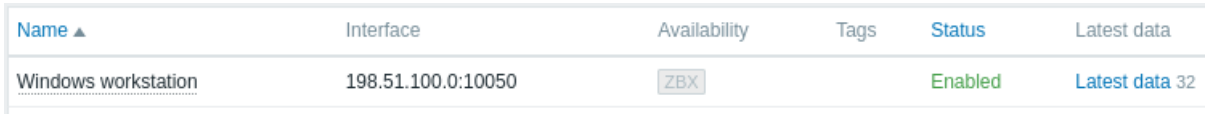
- The name of the host you created in Zabbix web interface to the *Hostname* parameter.
- The IP address or DNS name of your Zabbix server machine to the *ServerActive* parameter (might be prefilled if you have provided it during Zabbix agent setup).

Example:

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

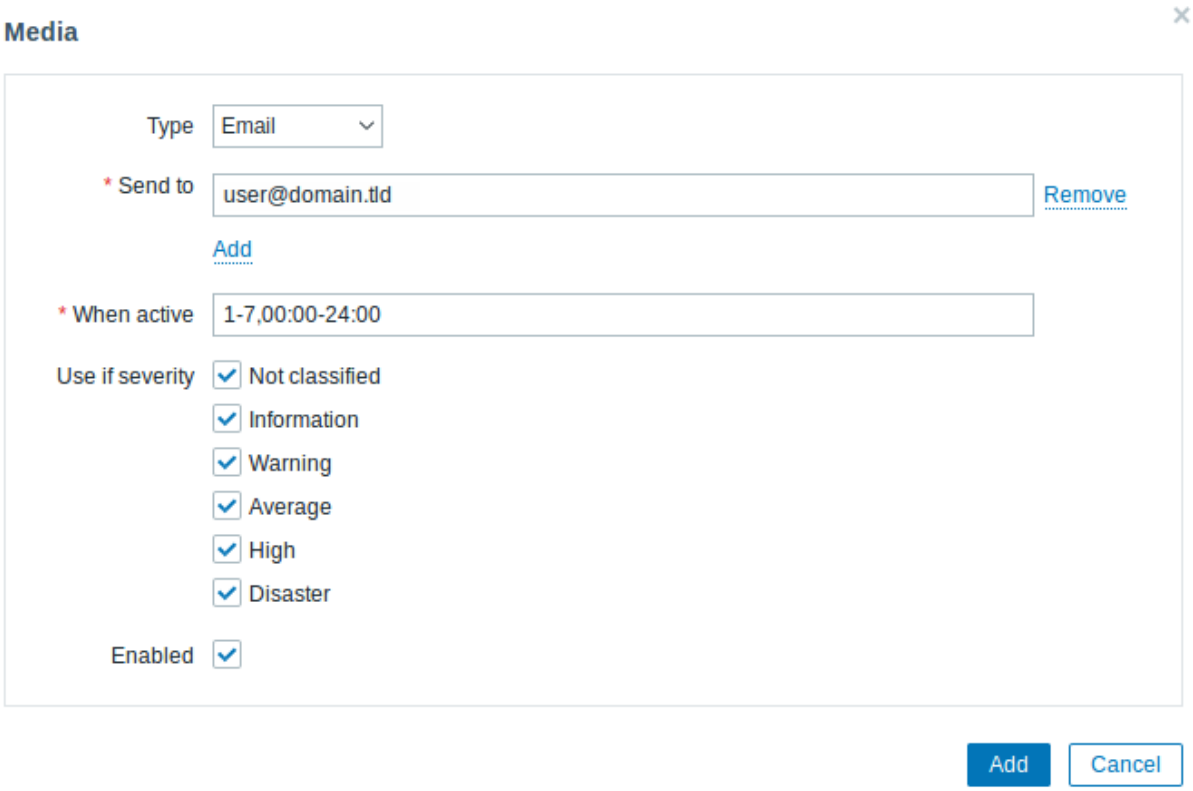
Visualizar métricas coletadas Parabéns! Neste ponto, o Zabbix já está monitorando sua máquina Windows.

Para visualizar as métricas coletadas, abra a seção de menu *Monitoramento->Hosts* menu section e clique em *Últimos dados* ao lado do host.



Configurar alertas de problemas O Zabbix pode notificá-lo sobre um problema em sua infraestrutura usando uma variedade de métodos. Este guia fornece etapas de configuração para o envio de alertas por e-mail.

1. Vá para *Configurações do usuário -> Perfil*, alterne para a guia *Mídia* e adicione seu e-mail.



2. Siga o guia para Receber notificação de problema.

Da próxima vez que o Zabbix detectar um problema, você deverá receber um alerta por e-mail.

Note:
No Windows, você pode usar o utilitário CpuStres para simular alta carga de CPU e, como resultado, receber um alerta de problema.

Veja também:

- Criando um item - como começar a monitorar métricas adicionais (monitoramento personalizado sem templates).
- Itens do Zabbix agent, Itens do Zabbix agent para Windows - lista completa de métricas que você pode monitorar usando o Zabbix agent no Windows.
- Escalonamentos de problemas - como criar cenários de alerta em várias etapas (por exemplo, primeiro enviar mensagem para o administrador do sistema, depois, se um problema não for resolvido em 45 minutos, enviar mensagem para o gerente do data center).

3 Monitorar o Apache via HTTP

Introdução Esta página mostra uma maneira rápida e simples de começar a monitorar um servidor web Apache sem instalar nenhum software adicional.

Para quem é este guia

Este guia foi desenvolvido para novos usuários do Zabbix e contém o conjunto mínimo de etapas necessárias para habilitar o monitoramento básico da sua instalação do Apache. Se você procura opções de personalização avançadas ou precisa de uma configuração mais detalhada, consulte a seção Configuração do manual do Zabbix.

Pré-requisitos

Antes de prosseguir com este guia de instalação, você deve baixar e instalar o Zabbix server e o Zabbix frontend de acordo com as instruções para o seu sistema operacional.

Preparar o Apache 1. Verifique qual versão do Apache você está usando:

Em sistemas baseados em RHEL, execute:

```
httpd -v
```

No Debian/Ubuntu, execute:

```
apache2 -v
```

2. Certifique-se de que o módulo Status está habilitado em sua instância do Apache.

Em sistemas baseados em RHEL, execute:

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

No Debian/Ubuntu, execute:

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

Se você não visualizar o status_module na lista, habilite o módulo executando:

Em sistemas baseados em RHEL, execute:

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

No Debian/Ubuntu, execute:

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

3. Edite o arquivo de configuração do Apache para permitir o acesso aos relatórios de status a partir do IP do Zabbix server.

Em sistemas baseados em RHEL: `/etc/httpd/conf.modules.d/status.conf`:

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

No Debian/Ubuntu: `/etc/apache2/mods-enabled/status.conf`:

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

Adicione as seguintes linhas ao arquivo (**substitua 198.51.100.255** pelo endereço IP do seu Zabbix server):

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

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

4. Reinicie o Apache

Em sistemas baseados em RHEL, execute:

```
sudo systemctl restart httpd
```

No Debian/Ubuntu, execute:

```
sudo systemctl restart apache2
```

5. Para verificar se tudo está configurado corretamente, execute (**substitua 198.51.100.255** pelo endereço IP do seu Zabbix server):

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

A resposta deve conter as estatísticas do servidor web Apache.

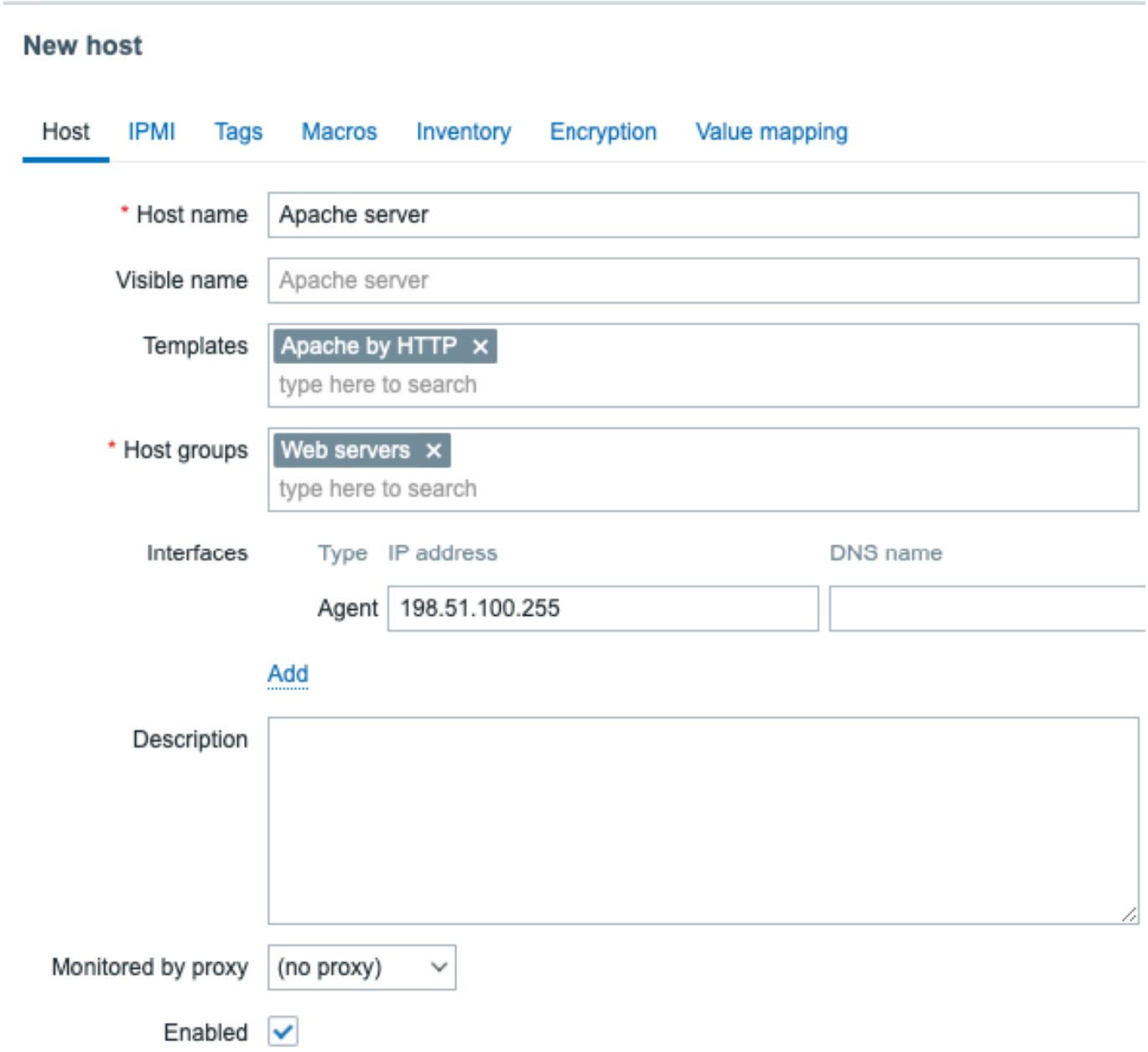
Configure o Zabbix para monitoramento 1. Faça login no frontend do Zabbix.

2. Crie um host na interface web do Zabbix.

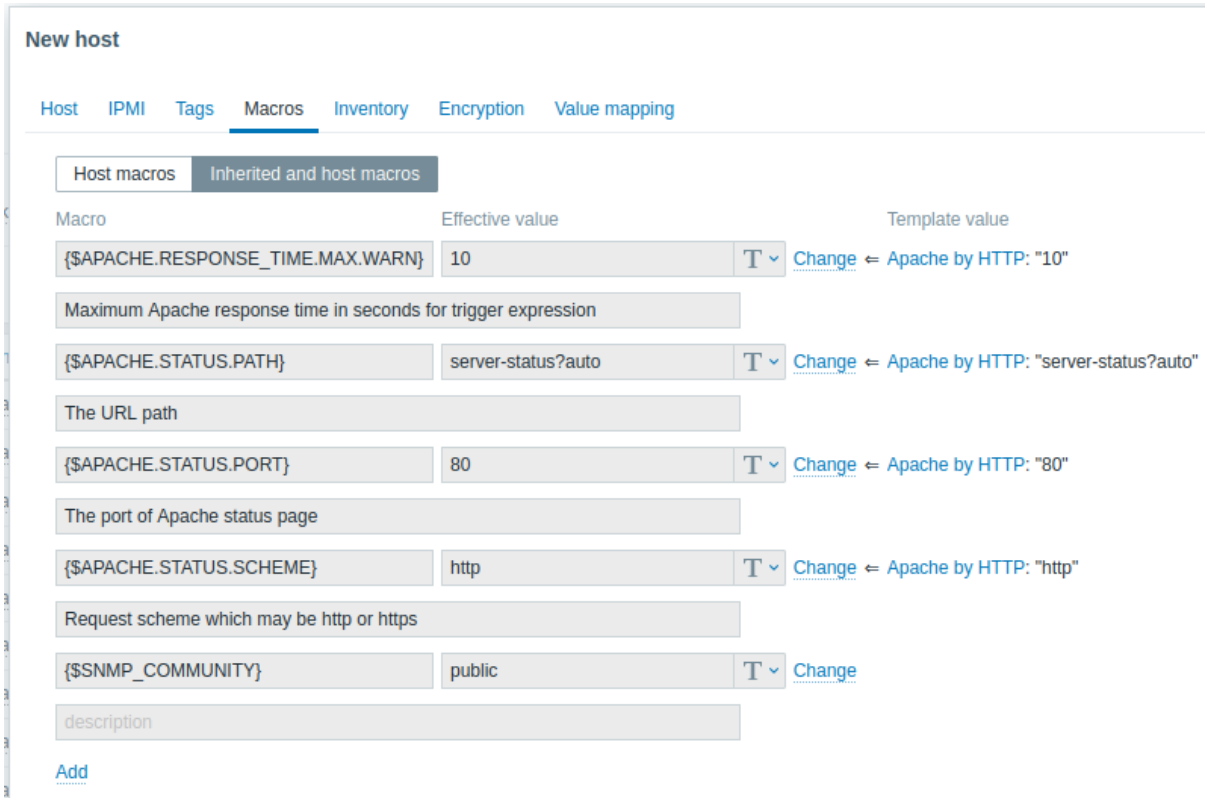
Este host representará seu servidor Apache.

3. No parâmetro *Interfaces*, adicione a interface *Agent* e especifique o endereço IP da instância do Apache. **Você não precisa instalar o agent do Zabbix na máquina**, a interface será usada apenas para resolver a macro {HOST.CONN}. Esta macro é usada nos items do template para localizar a instância do Apache.

4. No parâmetro *Templates*, digite ou selecione *Apache by HTTP*.



5. Mude para a guia **Macros** e selecione o modo *Macros herdadas e do host*. Verifique se os valores das macros {$APACHE.STATUS.PORT} e {$APACHE.STATUS.SCHEME} correspondem às configurações da sua instalação. Por padrão, a porta é 80 e o esquema é http. Altere os valores das macros se você usar uma porta e/ou esquema diferente.

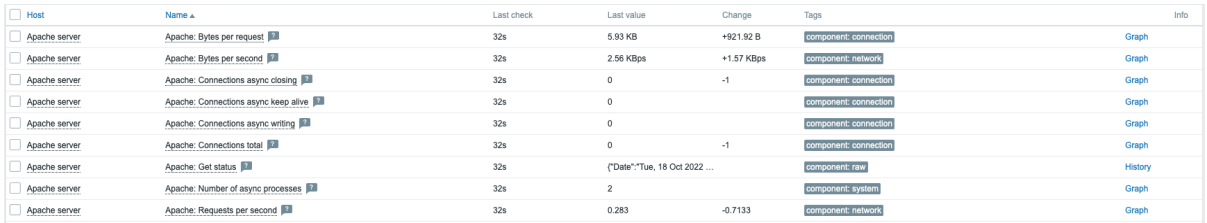


Visualizar métricas coletadas Parabéns! Neste ponto, o Zabbix já está monitorando seu servidor web Apache.

Para visualizar as métricas coletadas, abra a seção de menu *Monitoramento->Hosts* menu section e clique em *Dashboards* ao lado do host.

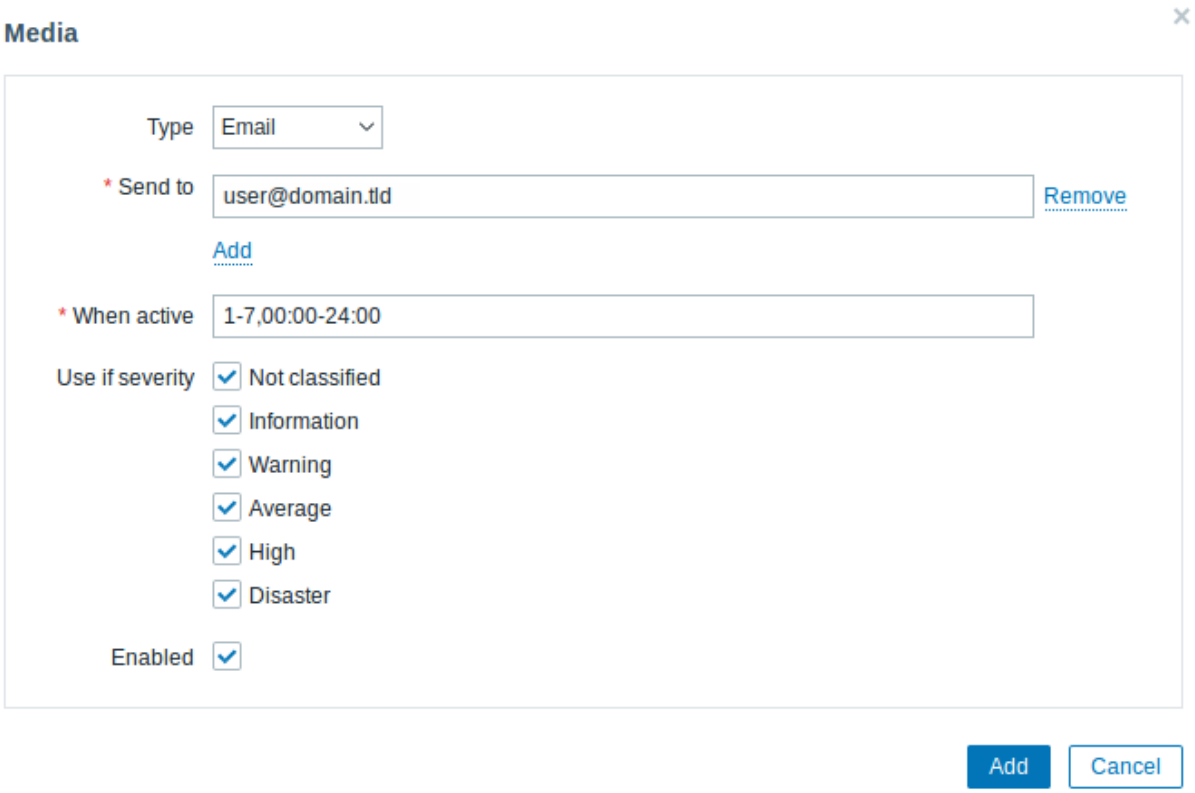
Esta ação o levará ao dashboard do host com as métricas mais importantes coletadas da página Apache /server-status.

Alternativamente, em *Monitoramento->Hosts*, você pode clicar em *Últimos dados* para visualizar todas as métricas coletadas mais recentes em uma lista.



Configurar alertas de problemas O Zabbix pode notificá-lo sobre um problema em sua infraestrutura usando uma variedade de métodos. Este guia fornece etapas de configuração para o envio de alertas por e-mail.

1. Vá para *Configurações do usuário -> Perfil*, alterne para a guia *Mídia* e adicione seu e-mail.



2. Siga o guia para Receber notificação de problema.

Da próxima vez que o Zabbix detectar um problema, você deverá receber um alerta por e-mail.

Teste sua configuração Para simular um problema real e receber um alerta de problema de teste:

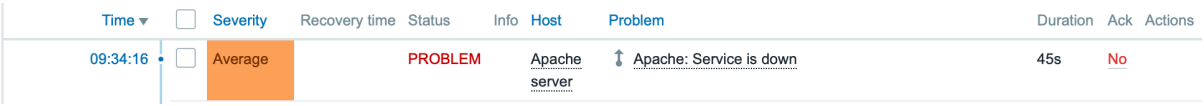
1. Abra a configuração do host *Apache server* no Zabbix.

2. Mude para a aba Macros e selecione *Macros herdadas e do host*.

3. Pressione *Alterar* ao lado da macro {$APACHE.STATUS.PORT} e defina uma porta diferente.

4. Pressione Atualizar para salvar a configuração do host.

5. Em alguns minutos, o Zabbix detectará o problema *Apache service is down*, porque agora não pode se conectar à instância. Ele aparecerá na seção Monitoramento->Problemas.



Se os alertas estiverem configurados, você também receberá a notificação do problema.

6. Altere o valor da macro de volta para resolver o problema e continue monitorando o Apache.

Veja também:

- Fortalecimento do servidor web - configurações recomendadas para maior segurança do servidor web.
- Criando um item - como começar a monitorar métricas adicionais.
- Itens HTTP - como monitorar métricas personalizadas usando o agente HTTP.
- Escalonamentos de problemas - como criar cenários de alerta em várias etapas (por exemplo, primeiro enviar mensagem para o administrador do sistema, depois, se um problema não for resolvido em 45 minutos, enviar mensagem para o gerente do data center).

4 Monitorar MySQL com o Zabbix agent 2

Introdução

Esta página orienta você nas etapas necessárias para iniciar o monitoramento básico de um servidor MySQL.

Para monitorar um servidor MySQL, existem várias abordagens: agent Zabbix, agent Zabbix 2 ou o padrão Open Database Connectivity (ODBC). O foco principal deste guia é o monitoramento de um servidor MySQL com o agent Zabbix 2, que é a abordagem **recomendada** devido à sua configuração simplificada em diferentes ambientes. No entanto, esta página também oferece instruções para as outras abordagens, portanto, fique à vontade para escolher a que melhor se adapta às suas necessidades.

Para quem é este guia

Este guia foi desenvolvido para novos usuários do Zabbix e contém o conjunto mínimo de etapas necessárias para habilitar o monitoramento básico de um servidor MySQL. Se você procura opções de personalização avançadas ou precisa de uma configuração mais avançada, consulte a seção Configuração do manual do Zabbix.

Pré-requisitos

Antes de prosseguir com este guia, você precisa baixar e instalar o Zabbix server, o Zabbix frontend e o Zabbix agent 2 de acordo com as instruções para o seu sistema operacional.

Dependendo do seu ambiente, algumas etapas deste guia podem ser ligeiramente diferentes. Este guia é baseado em um ambiente rodando Ubuntu.

Criar usuário MySQL

Para monitorar um servidor MySQL, o Zabbix requer acesso a ele e seus processos. Sua instalação do MySQL já possui um usuário com o nível de acesso necessário (o usuário "zabbix" que foi criado ao instalar o Zabbix), no entanto, este usuário possui mais privilégios do que o necessário para uma simples monitoração (privilégios para DROP de bancos de dados, DELETE de entradas de tabelas, etc.). Portanto, um usuário MySQL para o propósito de *apenas* monitorar o servidor MySQL precisa ser criado.

1. Conecte-se ao cliente MySQL, crie um usuário "zbx_monitor" (substitua *<password>* para o usuário "zbx_monitor" por uma senha de sua escolha), e GRANT os privilégios necessários ao usuário:

```
mysql -u root -p
# Digite a senha:

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

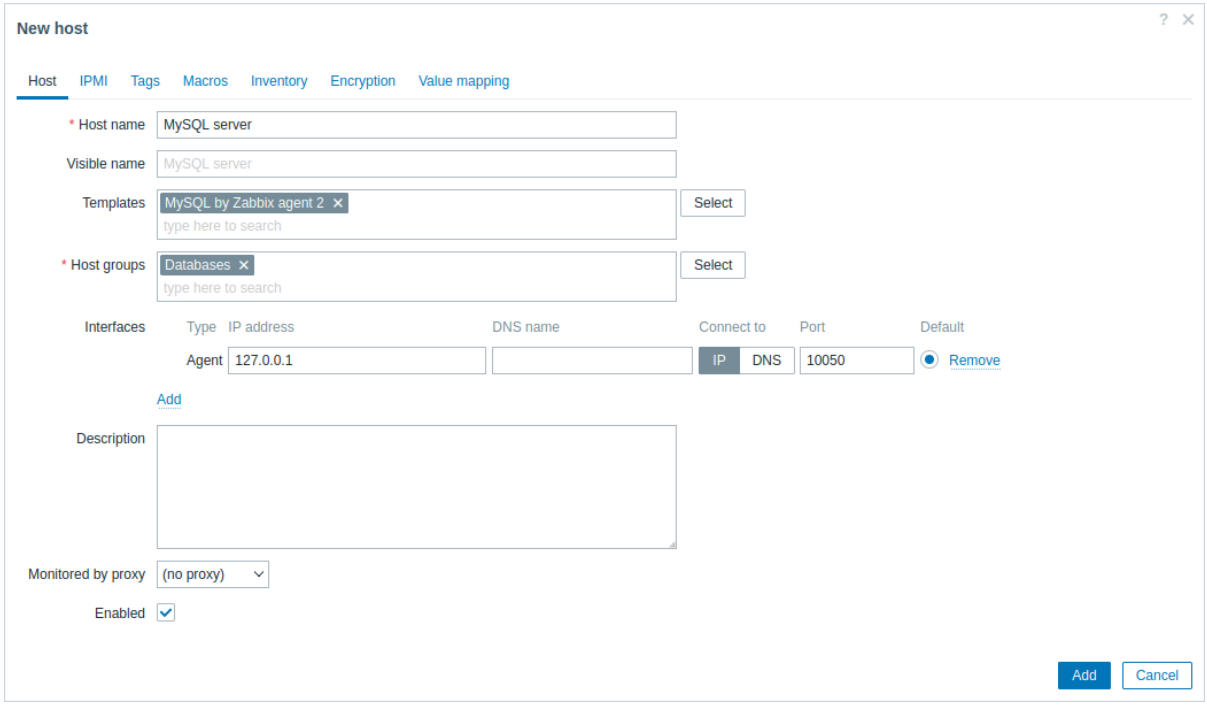
Depois que o usuário for criado, você pode passar para o próximo passo.

Configure o frontend do Zabbix

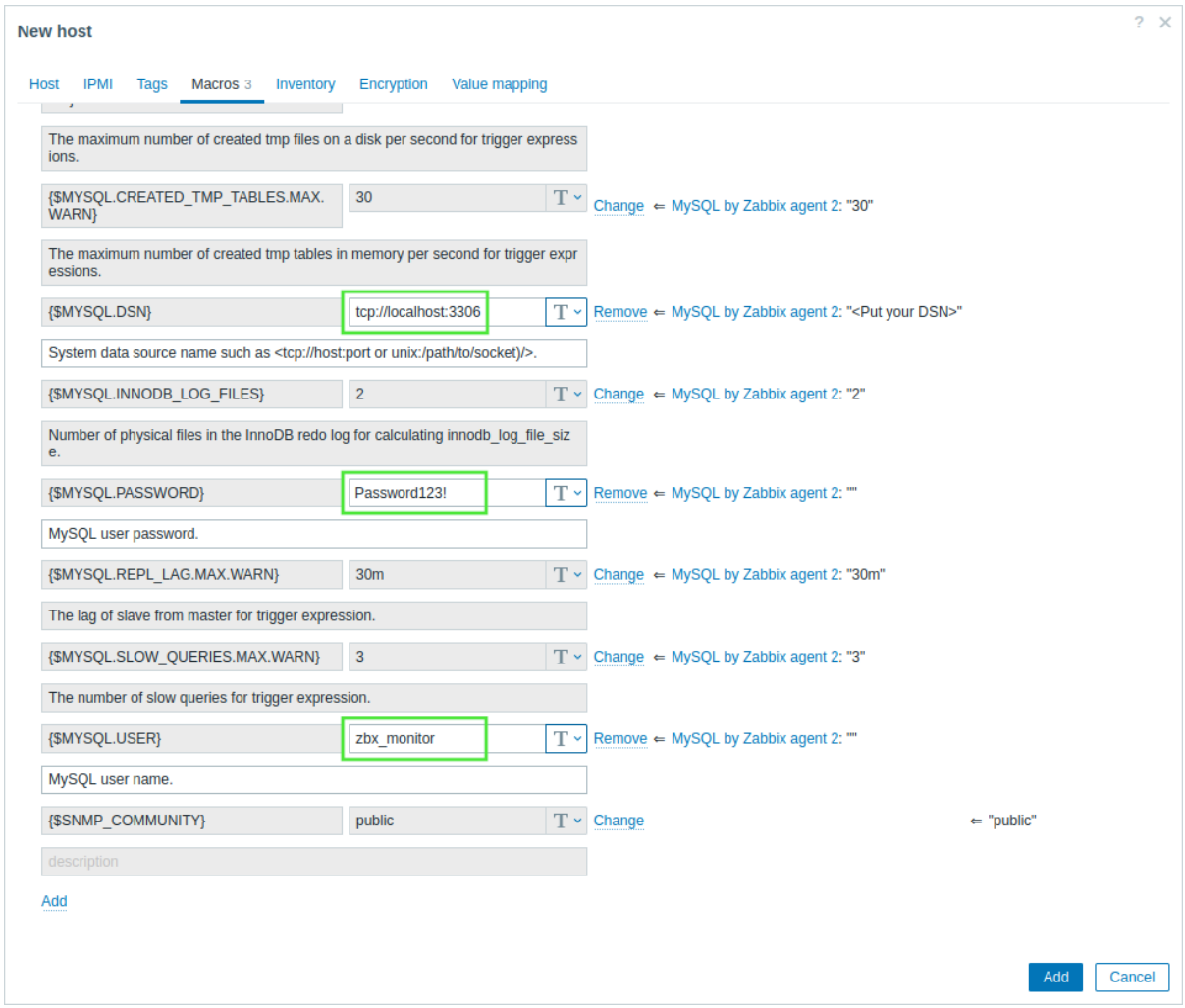
1. Faça login no frontend do Zabbix.

2. Crie um host na interface web do Zabbix:

- No campo *Nome do host*, insira um nome de host (por exemplo, "Servidor MySQL").
- No campo *Templates*, digite ou selecione o template "MySQL by Zabbix agent 2" que será vinculado ao host.
- No campo *Grupos de hosts*, digite ou selecione um grupo de hosts (por exemplo, "Databases").
- No campo *Interfaces*, adicione uma interface do tipo "Agent" e especifique o endereço IP do seu servidor MySQL. Este guia usa "127.0.0.1" (localhost) para monitorar um servidor MySQL que está instalado na mesma máquina que o Zabbix server e o Zabbix agent 2.



- Na guia *Macros*, alterne para *Macros herdadas e do host*, procure as seguintes macros e clique em *Alterar* ao lado do valor da macro para atualizá-lo:
  - {$MYSQL.DSN} - defina a fonte de dados do servidor MySQL (a string de conexão de uma sessão nomeada do arquivo de configuração do plugin MySQL do Zabbix agent 2). Este guia usa a fonte de dados padrão "tcp://localhost:3306" para monitorar um servidor MySQL que está instalado na mesma máquina que o Zabbix server e o Zabbix agent 2.
  - {$MYSQL.PASSWORD} - defina a senha do usuário MySQL criado anteriormente "zbx_monitor".
  - {$MYSQL.USER} - defina o nome do usuário MySQL criado anteriormente "zbx_monitor".

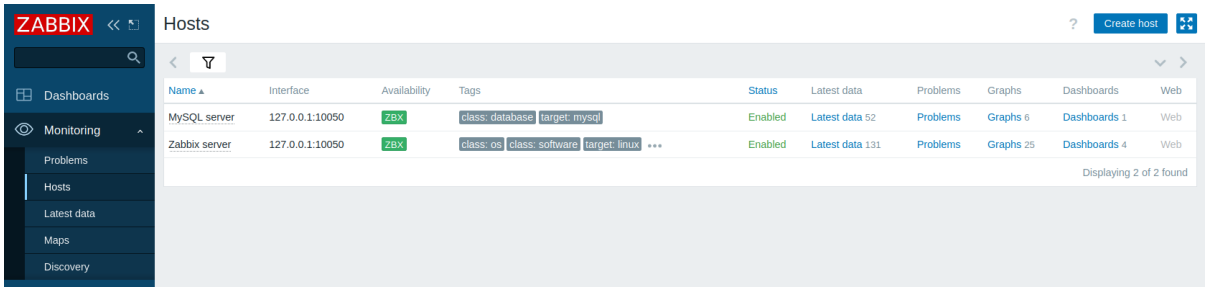


3. Clique em *Adicionar* para adicionar o host. Este host representará seu servidor MySQL.

Visualizar métricas coletadas

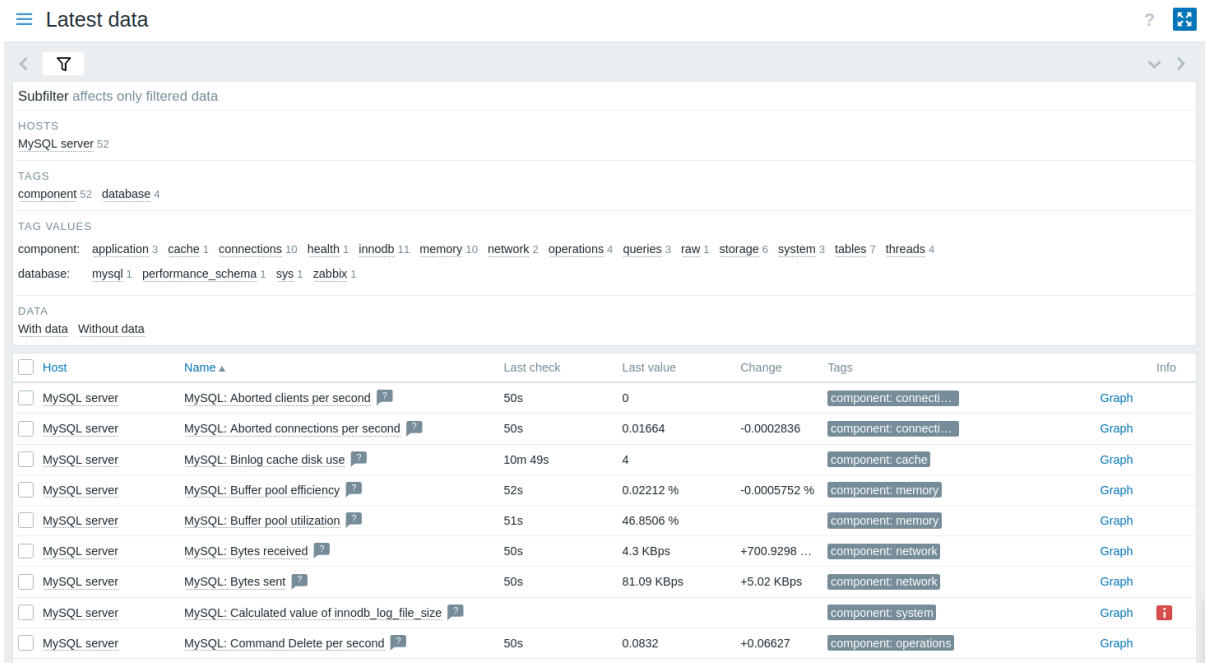
Parabéns! Neste ponto, o Zabbix já está monitorando seu servidor MySQL.

Para visualizar as métricas coletadas, navegue até a seção de menu *Monitoramento → Hosts* e clique em *Dashboards* ao lado do host.



Esta ação o levará ao dashboard do host (configurado no nível do template) com as métricas mais importantes coletadas do servidor MySQL.

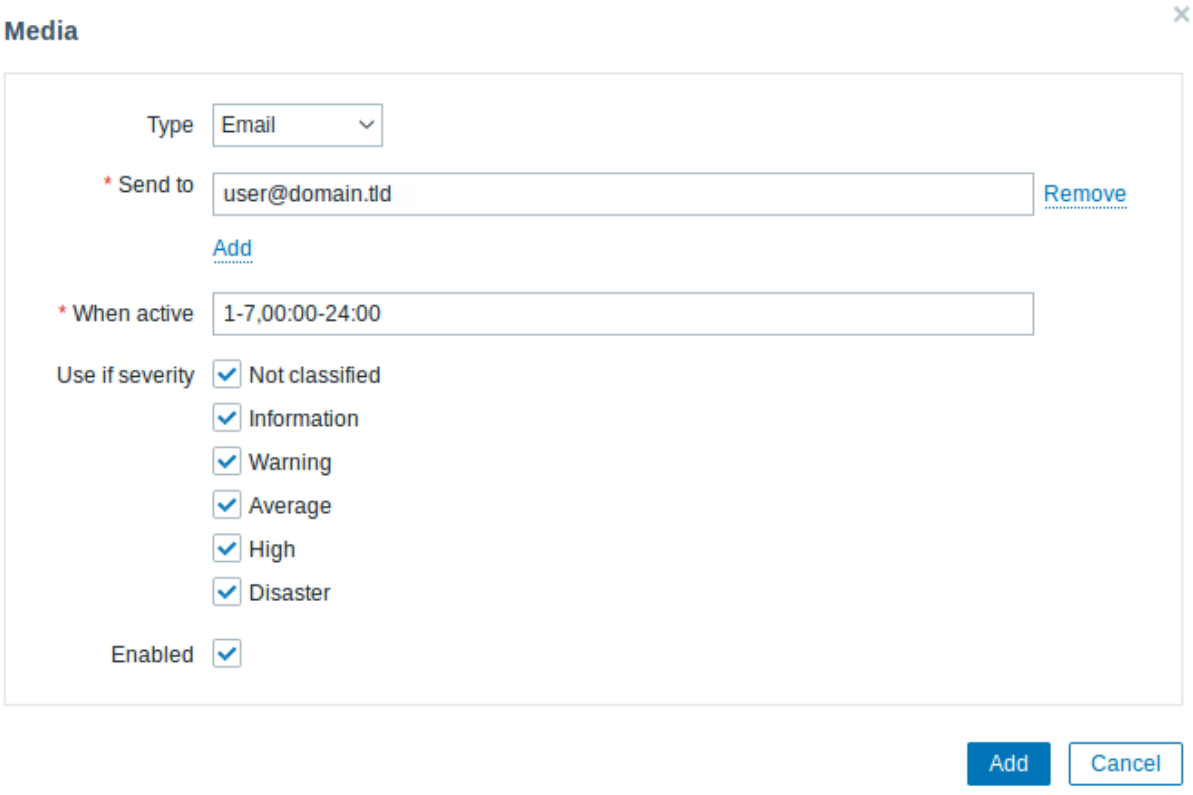
Alternativamente, na seção de menu *Monitoramento → Hosts*, você pode clicar em *Últimos dados* para visualizar todas as métricas coletadas mais recentes em uma lista. Observe que o item *MySQL: Calculated value of innodb_log_file_size* deve estar sem dados, pois o valor será calculado a partir dos dados da última hora.



Configure alertas de problemas

O Zabbix pode notificá-lo sobre um problema em sua infraestrutura usando uma variedade de métodos. Este guia fornece etapas básicas de configuração para o envio de alertas por e-mail.

1. Navegue até *Configurações do usuário → Perfil*, alterne para a guia *Mídia* e adicione seu e-mail.



2. Siga o guia para Receber uma notificação de problema.

Da próxima vez que o Zabbix detectar um problema, você deverá receber um alerta por e-mail.

Teste sua configuração

Para testar sua configuração, podemos simular um problema real atualizando a configuração do host no frontend do Zabbix.

1. Abra a configuração do host do seu servidor MySQL no Zabbix.

2. Mude para a aba *Macros* e selecione *Macros herdadas e do host*.

3. Clique em *Alterar* ao lado, por exemplo, do valor da macro {$MYSQL.USER} configurada anteriormente e defina um nome de usuário MySQL diferente.

4. Clique em *Atualizar* para atualizar a configuração do host.

5. Em alguns instantes, o Zabbix detectará o problema "MySQL: Service is down", pois não conseguirá se conectar ao servidor MySQL. O problema aparecerá em *Monitoramento → Problemas*.

Se os alertas estiverem configurados, você também receberá a notificação do problema.

6. Altere o valor da macro de volta ao valor anterior para resolver o problema e continuar monitorando o servidor MySQL.

Outras abordagens para monitorar o MySQL

Em vez de monitorar um servidor MySQL com o Zabbix agent 2, você também pode usar o Zabbix agent ou o padrão Open Database Connectivity (ODBC). Embora o uso do Zabbix agent 2 seja recomendado, pode haver algumas configurações que não suportam o Zabbix agent 2 ou que exigem uma abordagem personalizada.

A principal diferença entre o Zabbix agent e o ODBC está no método de coleta de dados - o Zabbix agent é instalado diretamente no servidor MySQL e coleta dados usando sua funcionalidade interna, enquanto o ODBC depende de um driver ODBC para estabelecer uma conexão com o servidor MySQL e recuperar dados usando consultas SQL.

Embora muitas das etapas de configuração sejam semelhantes ao monitoramento de um servidor MySQL com o Zabbix agent 2, existem algumas diferenças significativas - você precisa configurar o Zabbix agent ou o ODBC para poder monitorar um servidor MySQL. As instruções a seguir orientam você por essas **diferenças**.

Monitorar o MySQL com o agent Zabbix

Para monitorar um servidor MySQL com o agent Zabbix, você precisa baixar e instalar o server Zabbix, o frontend Zabbix e o agent Zabbix de acordo com as instruções para o seu sistema operacional.

Depois de instalar com sucesso os componentes necessários do Zabbix, você precisa criar um usuário MySQL conforme descrito na seção *Criar usuário MySQL*.

Depois de criar o usuário MySQL, você precisa configurar o agent Zabbix para poder estabelecer uma conexão com o servidor MySQL e monitorá-lo. Isso inclui configurar vários parâmetros de usuário para executar verificações personalizadas do agent, bem como fornecer ao agent Zabbix as credenciais necessárias para conectar-se ao servidor MySQL como o usuário "zbx_monitor" criado anteriormente.

Configure o agent Zabbix

1. Navegue até o diretório de configurações adicionais do agent Zabbix.

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

> **Attention:**
> O diretório de configurações adicionais do agent Zabbix deve estar localizado no mesmo diretório que o arquivo de configuração do agent Zabbix (*zabbix_agentd.conf*). Dependendo do seu sistema operacional e da instalação do Zabbix, este diretório pode ter um local diferente do especificado neste guia. Para locais padrão, verifique o parâmetro `Include` no arquivo de configuração do agent Zabbix.

Em vez de definir todos os parâmetros de usuário necessários para monitorar o servidor MySQL no arquivo de configuração do agent Zabbix, esses parâmetros serão definidos em um arquivo separado no diretório de configurações adicionais.

2. Crie um arquivo *template_db_mysql.conf* no diretório de configurações adicionais do agent Zabbix.

```
vi template_db_mysql.conf
```

3. Copie o conteúdo do arquivo *template_db_mysql.conf* (localizado no repositório do Zabbix) para o arquivo *template_db_mysql.conf* que você criou e salve.

4. Reinicie o agent Zabbix para atualizar sua configuração.

```
systemctl restart zabbix-agent
```

Depois de configurar os parâmetros de usuário do agent Zabbix, você pode prosseguir para configurar as credenciais que permitirão que o agent Zabbix acesse o servidor MySQL.

5. Navegue até o diretório inicial do agent Zabbix (se não existir em seu sistema, você precisa criá-lo; padrão: */var/lib/zabbix*).

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

6. Crie um arquivo *.my.cnf* no diretório inicial do agent Zabbix.

```
vi .my.cnf
```

7. Copie o seguinte conteúdo para o arquivo *.my.cnf* (substitua *<password>* pela senha do usuário "zbx_monitor").

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

Configure o frontend do Zabbix e teste sua configuração

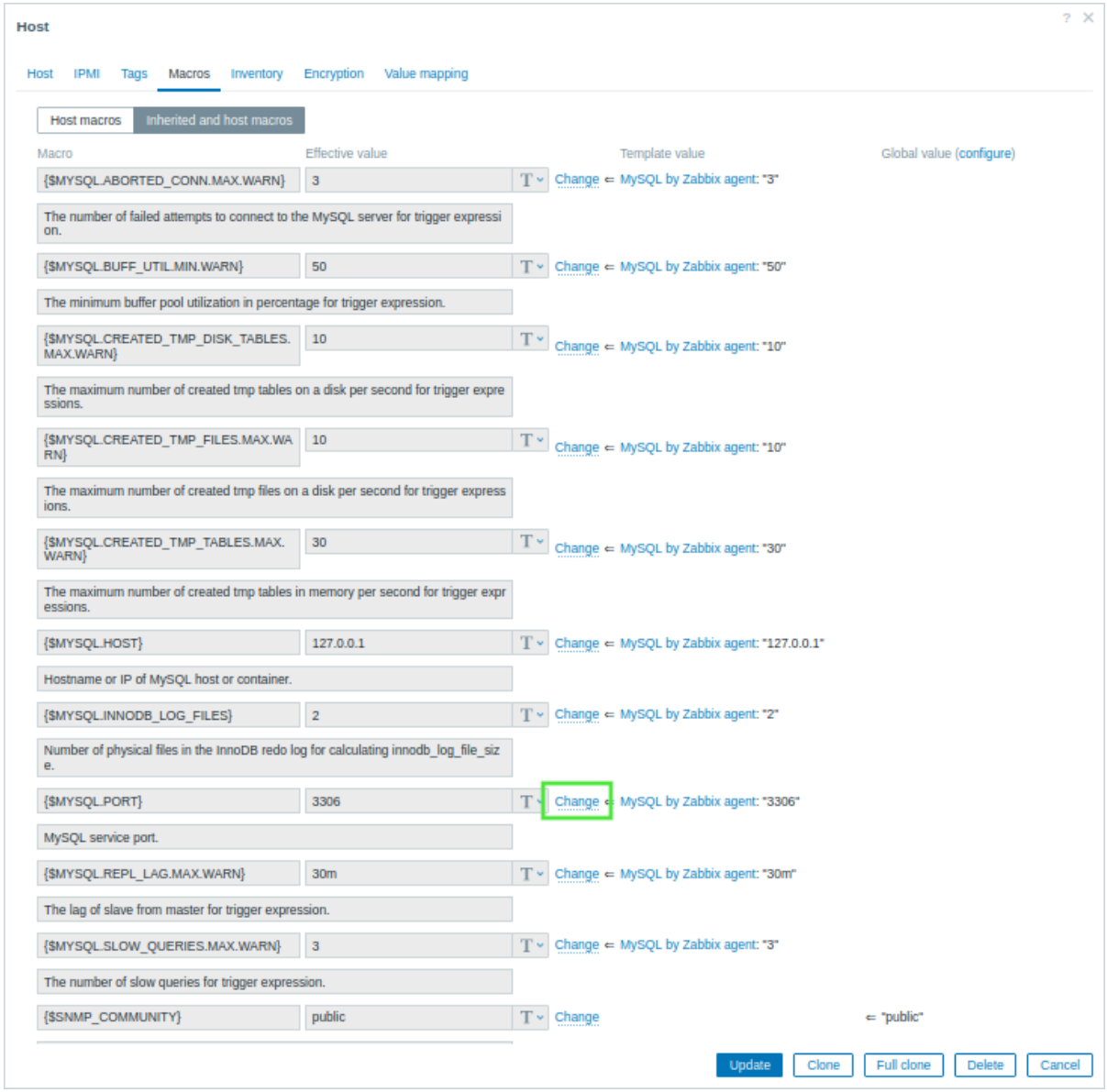
Para configurar o frontend do Zabbix, siga as instruções na seção *Configure o frontend do Zabbix* com os seguintes ajustes:

- No campo *Templates*, digite ou selecione o template "MySQL by Zabbix agent" que será vinculado ao host.
- Não é necessário configurar *Macros*.

Depois de configurar o frontend do Zabbix, você pode visualizar as métricas coletadas e configurar alertas de problemas.

Para testar sua configuração, siga as instruções na seção *Teste sua configuração* com os seguintes ajustes:

- Na seção *Macros herdadas e do host* da configuração do host do servidor MySQL, clique em *Alterar* ao lado do valor da macro {$MYSQL.PORT} e defina uma porta diferente (por exemplo, "6033").



Monitorar MySQL com ODBC

Para monitorar um servidor MySQL com ODBC, você precisa baixar e instalar o Zabbix server e o Zabbix frontend.

Depois de instalar com sucesso os componentes necessários do Zabbix, você precisa criar um usuário MySQL conforme descrito na seção *Criar usuário MySQL*.

Após criar o usuário MySQL, é necessário configurar o ODBC. Isso inclui instalar uma das implementações de API ODBC de código aberto mais utilizadas - unixODBC - e um driver unixODBC, além de editar o arquivo de configuração do driver ODBC.

Configurar ODBC

1. Instale o unixODBC. A maneira sugerida de instalar o unixODBC é usar os repositórios de pacotes padrão do sistema operacional Linux.

```
apt install unixodbc
```

2. Instale o driver de banco de dados unixODBC do MariaDB. Embora você tenha um banco de dados MySQL, o driver unixODBC do MariaDB é usado por questões de compatibilidade.

```
apt install odbc-mariadb
```

3. Verifique a localização dos arquivos de configuração ODBC *odbcinst.ini* e *odbc.ini*.

```
odbcinst -j
```

O resultado da execução deste comando deve ser semelhante ao seguinte.

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

4. Para configurar o driver ODBC para monitorar um banco de dados MySQL, você precisa do nome do driver, que está localizado no arquivo *odbcinst.ini*. No exemplo de arquivo *odbcinst.ini* a seguir, o nome do driver é "MariaDB Unicode".

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

5. Copie o seguinte conteúdo para o arquivo *odbc.ini* (substitua <*password*> pela senha do usuário "zbx_monitor"). Este guia usa "127.0.0.1" (localhost) como o endereço do servidor MySQL para monitorar um servidor MySQL que está instalado na mesma máquina que o driver ODBC. Observe o nome da fonte de dados (DSN) "test", que será necessário ao configurar o frontend do Zabbix.

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

Configure o frontend do Zabbix e teste sua configuração

Para configurar o frontend do Zabbix, siga as instruções da seção *Configure o frontend do Zabbix* com os seguintes ajustes:

- No campo *Templates*, digite ou selecione o template "MySQL by ODBC" que será vinculado ao host.
- Não é necessário configurar *Interfaces*.
- O valor da macro {$MYSQL.DSN} na seção *Macros herdadas e do host* da configuração do host do servidor MySQL deve ser definido para o nome DSN do arquivo *odbc.ini*.

Depois de configurar o frontend do Zabbix, você pode visualizar as métricas coletadas, configurar alertas de problemas e testar sua configuração.

Veja também

- Criando um item - como começar a monitorar métricas adicionais.
- Escalonamento de problemas - como criar cenários de alerta em várias etapas (por exemplo, primeiro enviar mensagem para o administrador do sistema, depois, se o problema não for resolvido em 45 minutos, enviar mensagem para o gerente do data center).
- Monitoramento ODBC - como configurar ODBC em outras distribuições Linux e como começar a monitorar métricas adicionais relacionadas ao banco de dados com ODBC.
- Template *MySQL por Zabbix agent* - informações adicionais sobre o template *MySQL por Zabbix agent*.
- Template *MySQL por Zabbix agent 2* - informações adicionais sobre o template *MySQL por Zabbix agent 2*.
- Template *MySQL por ODBC* - informações adicionais sobre o template *MySQL por ODBC*.

5 Monitorar VMware com o Zabbix

Introdução

Esta página orienta você nas etapas necessárias para iniciar o monitoramento básico do VMware.

Para quem é este guia

Este guia foi desenvolvido para novos usuários do Zabbix e contém o conjunto mínimo de etapas necessárias para habilitar o monitoramento básico do VMware. Se você procura opções de personalização avançadas ou precisa de uma configuração mais avançada, consulte a seção Monitoramento de máquinas virtuais ou a seção Configuração do manual do Zabbix.

Pré-requisitos

Antes de prosseguir com este guia, você precisa baixar e instalar o Zabbix server e o Zabbix frontend de acordo com as instruções para o seu sistema operacional.

Dependendo do seu ambiente, alguns passos deste guia podem ser ligeiramente diferentes. Este guia é baseado em um ambiente rodando Ubuntu.

Pressupõe-se que o VMware já esteja configurado. Este guia não cobre a configuração do VMware.

Configurar o Zabbix server

Para monitorar o VMware, os processos *vmware collector* do Zabbix precisam ser habilitados. Para mais informações sobre como o monitoramento do VMware é realizado, consulte Monitoramento de máquinas virtuais.

1. Abra o arquivo de configuração do Zabbix server.

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

2. Localize e defina o parâmetro `StartVMwareCollectors` no arquivo de configuração do Zabbix server para 2 ou mais (o valor padrão é 0).

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

StartVMwareCollectors=2
```

3. Reinicie o Zabbix server.

```
systemctl restart zabbix-server
```

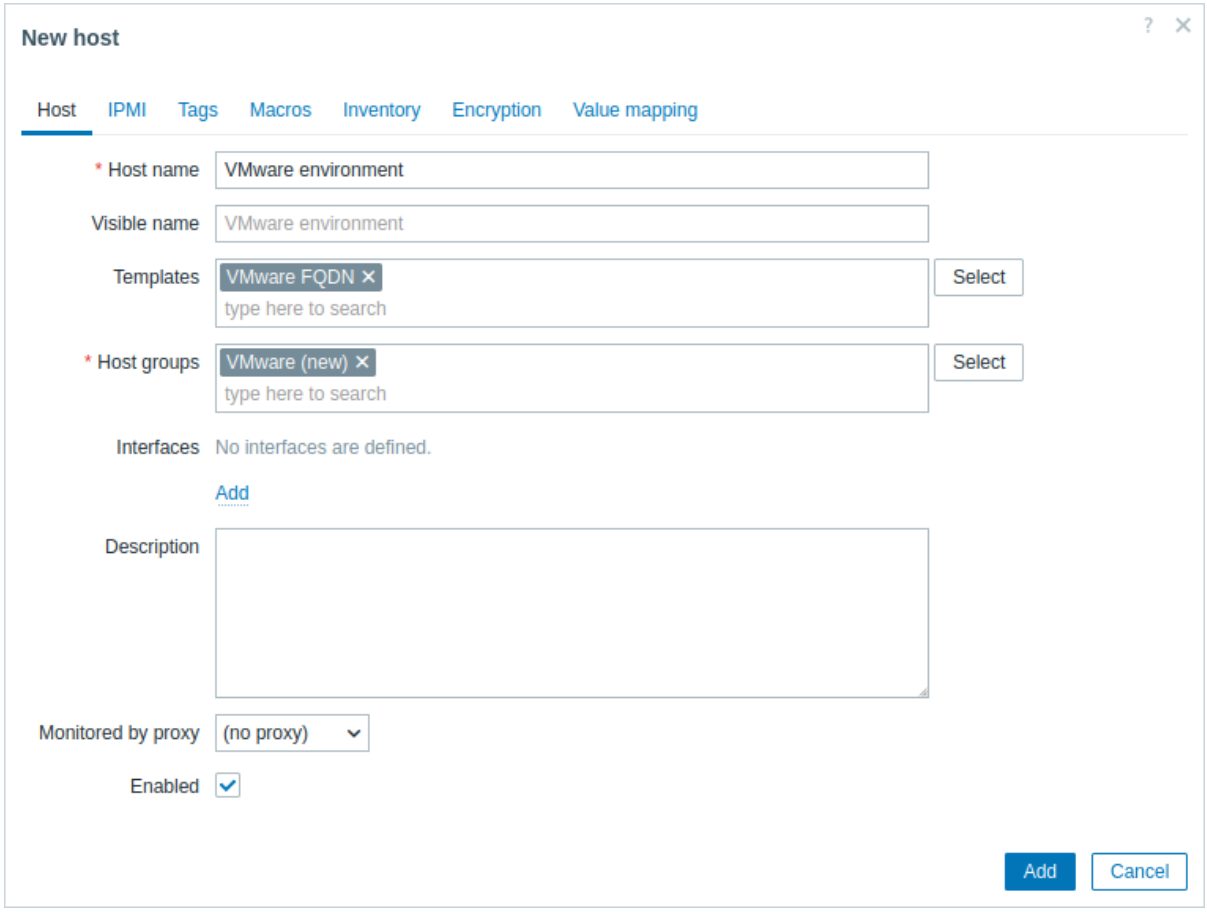
Assim que os processos *vmware collector* forem iniciados, prossiga para a próxima etapa.

Configure o frontend do Zabbix

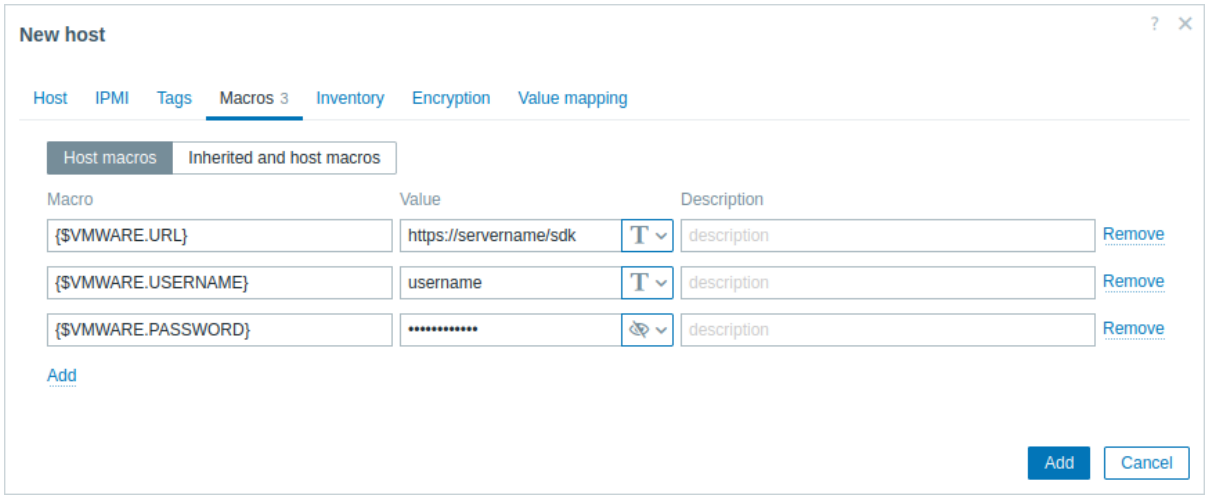
1. Faça login no frontend do Zabbix.

2. Crie um host na interface web do Zabbix:

- No campo *Nome do host*, insira um nome de host (por exemplo, ”Ambiente VMware”).
- No campo *Templates*, digite ou selecione o template ”VMware FQDN” (ou ”VMware”). Para mais informações sobre esses templates, consulte Monitoramento de máquina virtual.
- No campo *Grupos de hosts*, digite ou selecione um grupo de hosts (por exemplo, um novo grupo de hosts ”VMware”).



- Na aba *Macros*, defina as seguintes macros de host:
  - {$VMWARE.URL} - URL do SDK do serviço VMware (vCenter ou hipervisor ESXi) (https://nomedoservidor/sdk)
  - {$VMWARE.USERNAME} - Nome de usuário do serviço VMware
  - {$VMWARE.PASSWORD} - Senha do usuário {$VMWARE.USERNAME} do serviço VMware



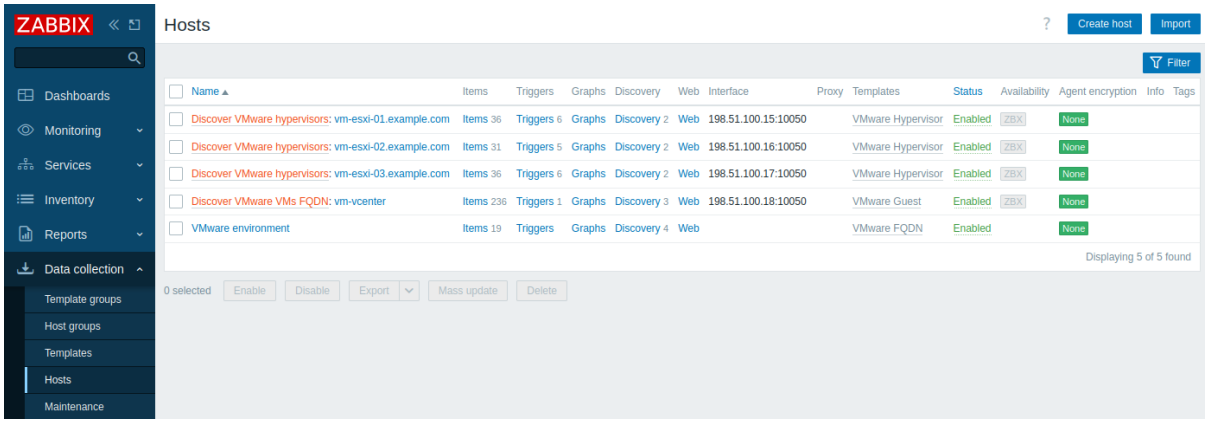
3. Clique no botão *Adicionar* para criar o host. Este host representará seu ambiente VMware.

Visualizar métricas coletadas

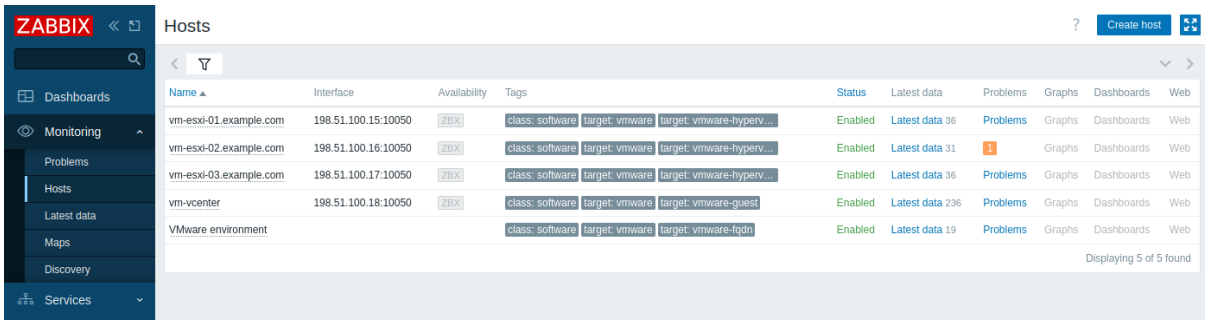
Parabéns! Neste ponto, o Zabbix já está monitorando seu ambiente VMware.

Dependendo da configuração do seu ambiente VMware, o Zabbix pode descobrir e então criar hosts para as entidades descobertas. Observe que a descoberta e a criação de hosts também podem ser executadas manualmente, se necessário.

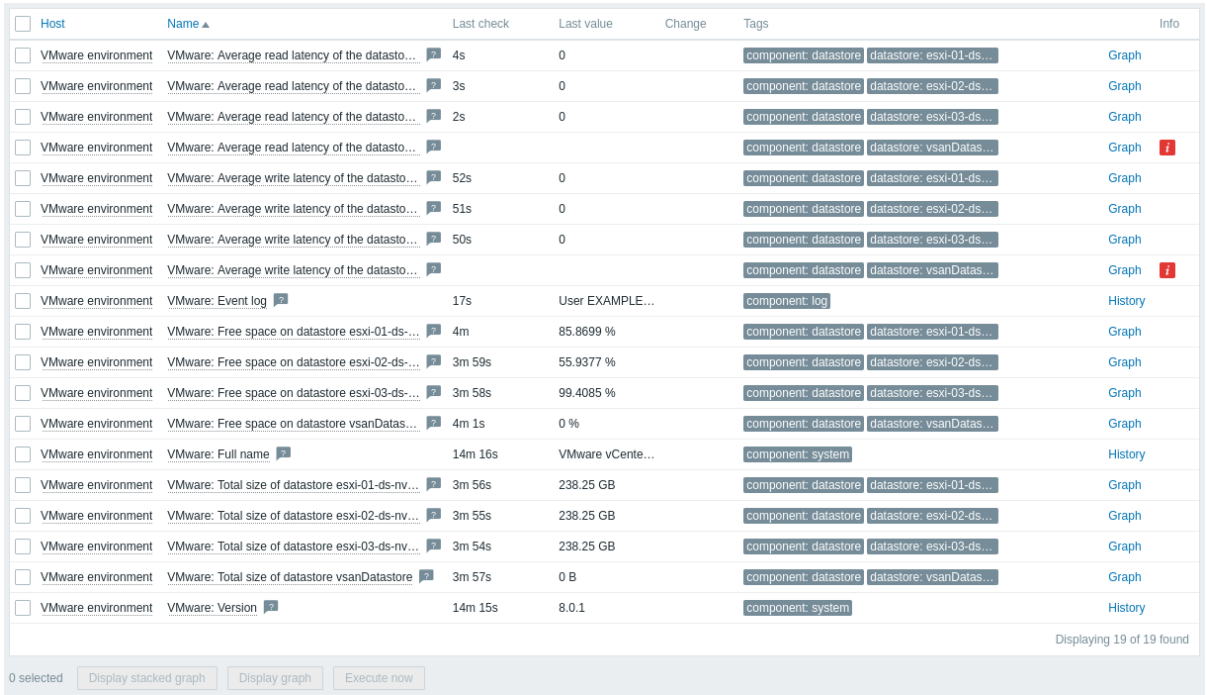
Para visualizar os hosts criados, navegue até a seção de menu *Coleta de dados → Hosts*.



Para visualizar as métricas coletadas, navegue até a seção de menu *Monitoramento → Hosts* e clique em *Últimos dados* ao lado do host "VMware environment" criado ou de um dos hosts que foram criados para as entidades descobertas.



Esta ação abrirá uma lista de todas as métricas mais recentes coletadas do host selecionado.

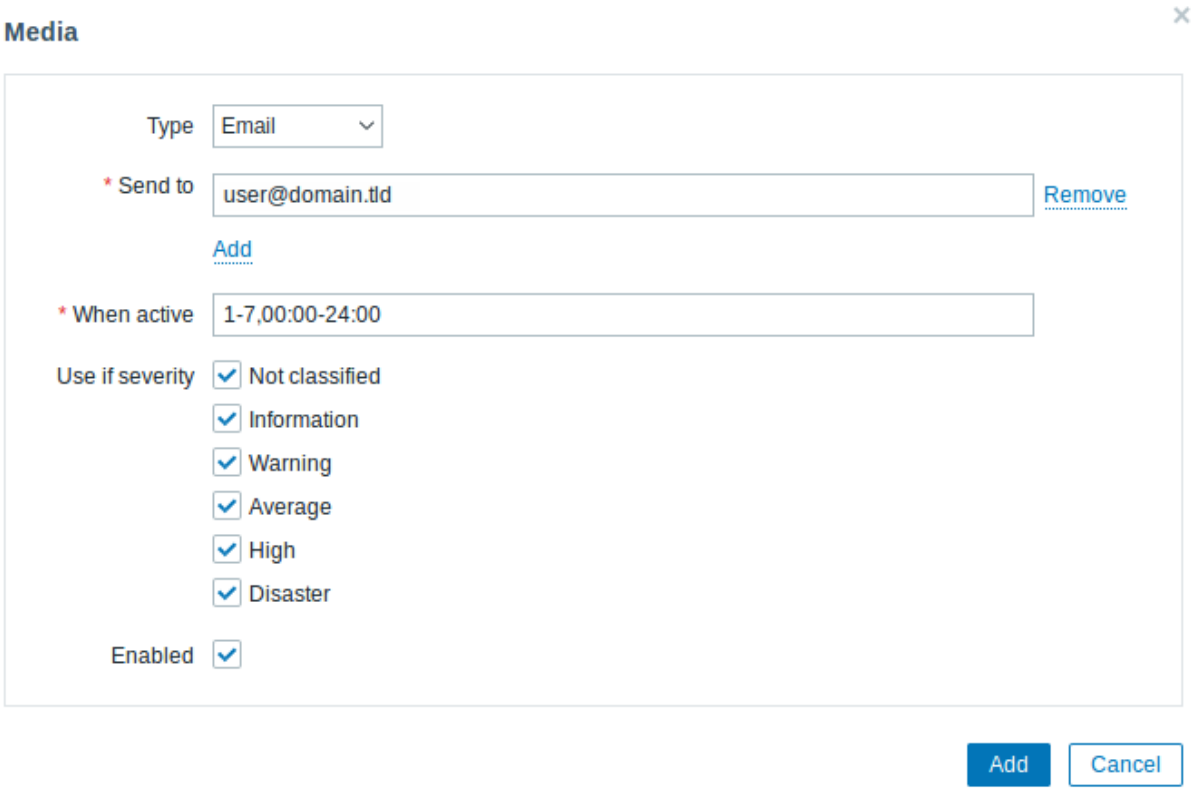


Observe que alguns items não possuem dados e estão no estado *Não suportado*. Isso ocorre porque o Zabbix não consegue encontrar contadores de desempenho válidos no datastore específico, pois ele não está habilitado no ambiente VMware monitorado.

Configure alertas de problemas

O Zabbix pode notificá-lo sobre um problema em sua infraestrutura usando vários métodos. Este guia fornece etapas básicas de configuração para o envio de alertas por e-mail.

1. Navegue até *Configurações do usuário → Perfil*, alterne para a guia *Mídia* e adicione seu e-mail.



2. Siga o guia para Receber uma notificação de problema.

Da próxima vez que o Zabbix detectar um problema, você deverá receber um alerta por e-mail.

Veja também

- Criando um item - como começar a monitorar métricas adicionais.
- Escalonamentos de problemas - como criar cenários de alerta em várias etapas (por exemplo, primeiro enviar mensagem ao administrador do sistema, depois, se o problema não for resolvido em 45 minutos, enviar mensagem ao gerente do data center).
- Monitoramento de máquinas virtuais - informações adicionais sobre o monitoramento do VMware (processo de coleta de dados, opções de configuração do server, orientações para solução de problemas, etc.).
- Chaves de item de monitoramento do VMware - uma lista completa de métricas do VMware que podem ser monitoradas usando o Zabbix.
- Template *VMware* - informações adicionais sobre o template *VMware*.
- Template *VMware FQDN* - informações adicionais sobre o template *VMware FQDN*.

6 Monitorar o tráfego de rede com o Zabbix

Introdução Esta página orienta você nas etapas necessárias para iniciar o monitoramento básico do tráfego de rede com o Zabbix.

Para quem é este guia

Este guia foi desenvolvido para novos usuários do Zabbix e contém o conjunto mínimo de etapas necessárias para ativar o monitoramento básico do tráfego de rede. Se você procura opções de personalização avançadas ou precisa de uma configuração mais detalhada, consulte a seção Configuração do manual do Zabbix.

Pré-requisitos

Antes de prosseguir com este guia, você precisa baixar e instalar o Zabbix server, o Zabbix frontend e o Zabbix agent de acordo com as instruções para o seu sistema operacional. Observe que você deve instalar o Zabbix agent na máquina que requer o monitoramento de tráfego. Esta pode ser a mesma máquina onde o Zabbix server está instalado ou uma máquina diferente.

Este guia fornecerá as instruções para configurar o monitoramento do tráfego de rede da interface *eth0* em uma máquina separada chamada *Remote host*.

Configurar o Zabbix para monitoramento O agent do Zabbix pode coletar métricas nos modos ativo ou passivo (simultaneamente). Veja Verificações ativas e passivas do agent para mais informações. Neste guia, o monitoramento por meio de verificações passivas será descrito.

Configurando o agent Zabbix

1. Abra o arquivo de configuração do agent na máquina onde o agent está instalado (por padrão, o caminho é */usr/local/etc/zabbix_agentd.conf*

```
sudo vi /usr/local/etc/zabbix_agentd.conf
```

2. Adicione o endereço IP ou o nome DNS do seu server Zabbix ao parâmetro *Server*. Por exemplo:

```
Server=192.0.2.22
```

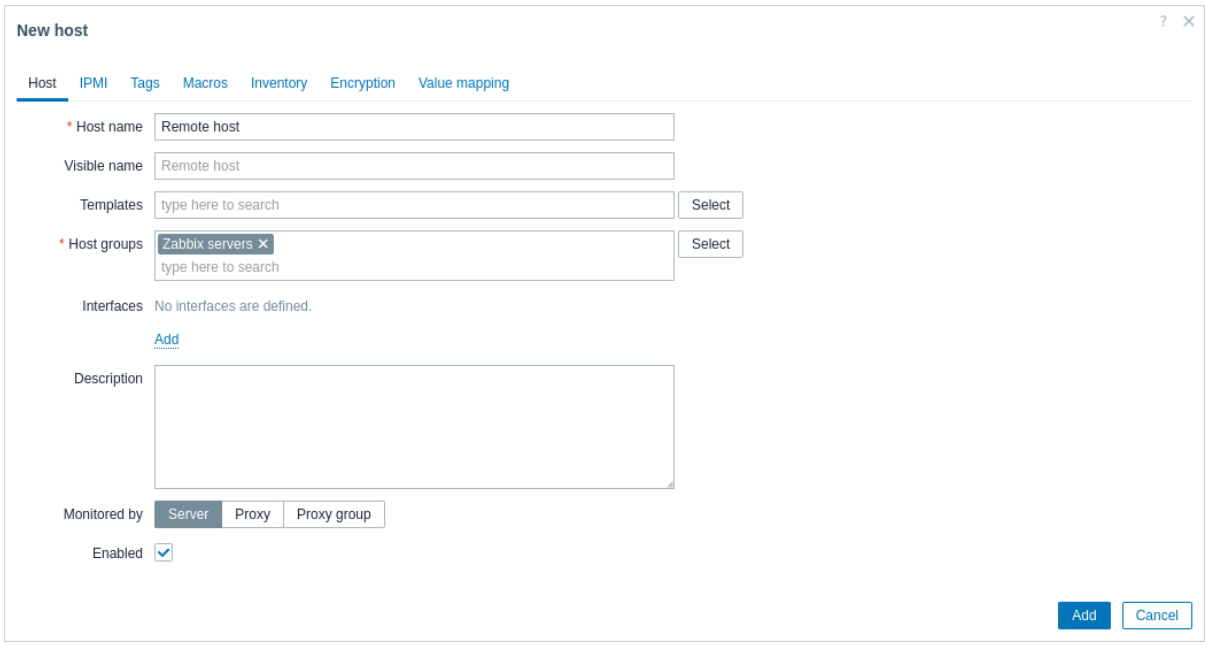
3. Reinicie o agent Zabbix:

```
systemctl restart zabbix-agent
```

Frontend do Zabbix

1. Faça login no frontend do Zabbix.

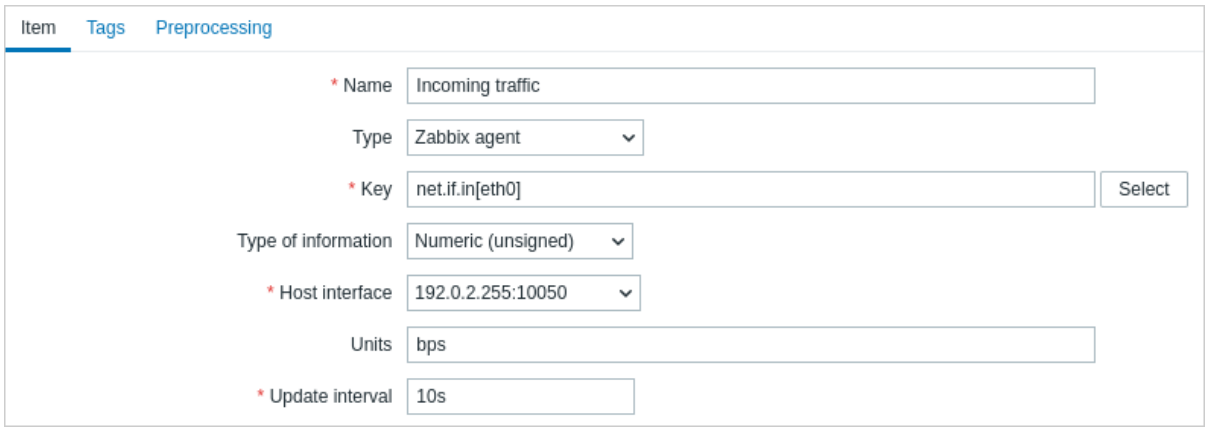
2. Crie um host na interface web do Zabbix, especificando o endereço IP ou o nome DNS da máquina na qual o agent está instalado.



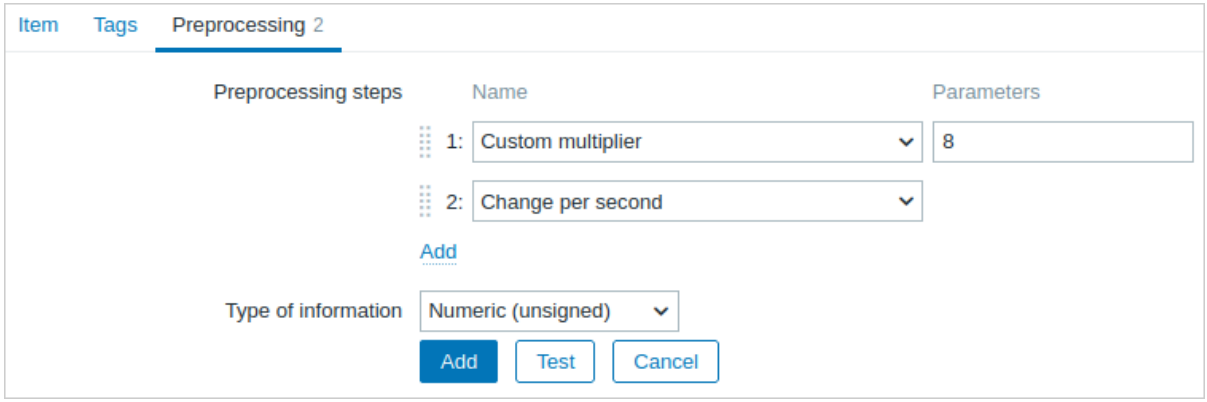
Criar items Siga as instruções em criando um item para adicionar os items para monitoramento de tráfego, a saber:

- Tráfego de entrada
- Tráfego de saída
- Tráfego total

Uma configuração simples para o item de monitoramento de tráfego de entrada seria a seguinte:

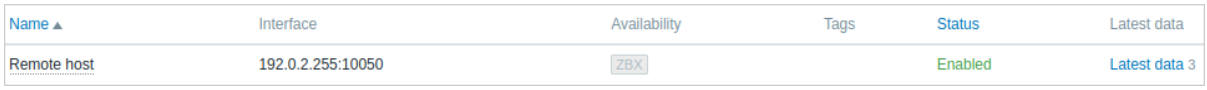


Para tornar os dados coletados adequados para uso prático, você pode definir algumas etapas de pré-processamento ao criar os items. No caso apresentado, pode ser a multiplicação por 8 (para converter bytes em bits) e apresentação como alteração por segundo.

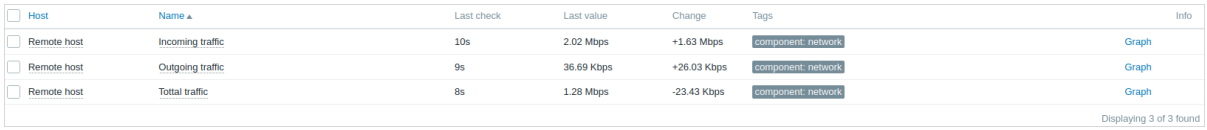


Visualizar dados coletados Parabéns! Neste ponto, o Zabbix já está monitorando o tráfego da sua rede.

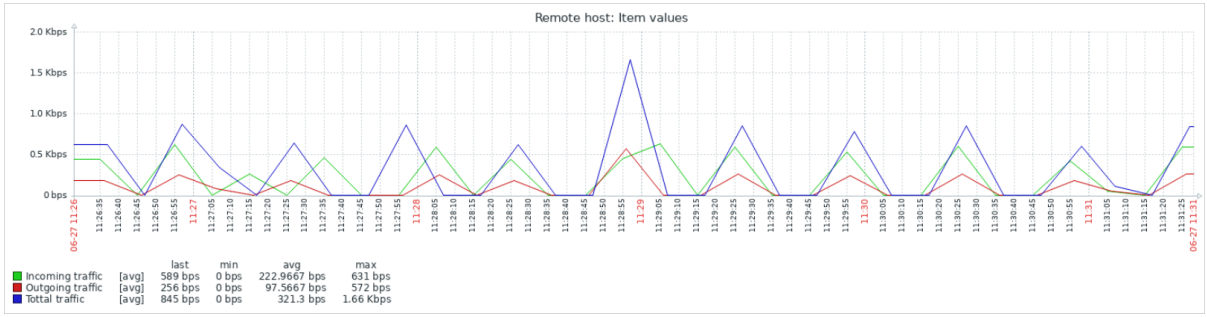
Para visualizar as métricas coletadas, abra a seção de menu *Monitoramento → Hosts* e clique em *Últimos dados* na linha do host.



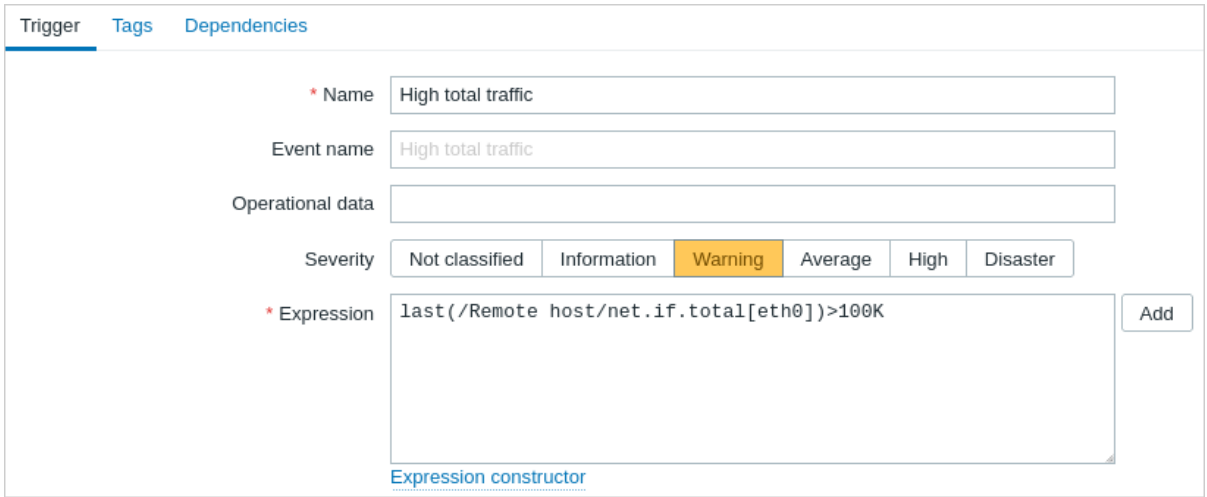
Você verá os dados de tráfego.



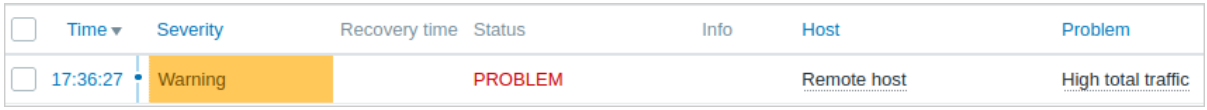
Visualizar gráficos Os dados coletados podem ser exibidos como gráficos. Para visualizá-los, na seção *Últimos dados*, clique em *Gráfico* na linha do item ou selecione os itens necessários e clique em *Exibir gráfico* abaixo.



Configurar triggers Você pode definir triggers para detectar tráfego de rede anormal. Veja as instruções sobre como configurar um trigger e adicione o trigger sinalizando que o tráfego total está muito alto, por exemplo:



Agora faça o tráfego exceder o limite que você definiu na expressão do trigger e navegue até *Monitoramento → Problemas* para verificar se o problema está listado lá.



Configurar alertas de problema Existem várias maneiras de receber notificações sobre o problema. O e-mail é a mais popular, siga as instruções para configurar uma notificação de problema por e-mail. Você também pode escolher outros tipos de mídia para serem usados na entrega da notificação.

Veja também:

- Escalonamentos de problemas - como criar cenários de alerta em várias etapas (por exemplo, primeiro enviar mensagem ao administrador do sistema, depois, se o problema não for resolvido em 45 minutos, enviar mensagem ao gerente do data center).
- Reconhecimento de problemas - como indicar que o problema é conhecido, fazer comentários sobre sua resolução, suprimir ou fechar o problema.
- Monitorar Linux com Zabbix agent - como iniciar o monitoramento básico dos itens mais importantes vinculando um template pré-configurado.

7 Monitorar o tráfego de rede usando verificações ativas

Introdução Esta página orienta você nas etapas necessárias para iniciar o monitoramento básico do tráfego de rede com o Zabbix usando verificações ativas.

Para quem é este guia

Este guia foi desenvolvido para novos usuários do Zabbix e contém o conjunto mínimo de etapas necessárias para habilitar o monitoramento básico do tráfego de rede usando verificações ativas. Se você procura opções de personalização avançadas ou precisa de uma configuração mais avançada, consulte a seção Configuração do manual do Zabbix.

Pré-requisitos

Antes de prosseguir com este guia, você precisa baixar e instalar o Zabbix server, o Zabbix frontend e o Zabbix agent de acordo com as instruções para o seu sistema operacional. Observe que você deve instalar o Zabbix agent na máquina que requer o monitoramento de tráfego. Esta pode ser a mesma máquina onde o Zabbix server está instalado ou uma máquina diferente.

Este guia fornecerá as instruções para configurar o monitoramento do tráfego de rede da interface *eth0* em uma máquina separada chamada *Remote host*.

Configurar o Zabbix para monitoramento O agent do Zabbix pode coletar métricas nos modos ativo ou passivo (simultaneamente). Veja Verificações ativas e passivas do agent para mais informações. Neste guia, o monitoramento por meio de **verificações ativas** será descrito.

Configurando o agent Zabbix

1. Abra o arquivo de configuração do agent na máquina onde o agent está instalado.

Se você estiver usando o agent Zabbix:

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

Se você estiver usando o agent Zabbix 2:

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

2. Adicione o endereço IP ou nome DNS (e porta opcional) do seu server Zabbix ao parâmetro ServerActive. Por exemplo:

```
ServerActive=192.0.2.0:10051
```

O agent Zabbix usará este endereço para se conectar à porta trapper do server Zabbix (padrão: 10051) e solicitar os dados de configuração para verificações ativas.

3. Defina o parâmetro Hostname, que deve corresponder ao nome do host que será definido no frontend Zabbix. No nosso exemplo, é:

```
Hostname=Remote host
```

O valor de Hostname deve corresponder porque, para verificações ativas, o agent Zabbix o utiliza para recuperar a configuração correta do host a partir do server. Especificamente, o Agent inicia uma conexão com o Server e se identifica usando o valor de

Hostname. O Server então fornece a configuração de monitoramento para esse host. Se esses valores forem diferentes, o Agent não receberá a configuração apropriada, resultando em métricas ausentes ou problemas de monitoramento.

4. Reinicie o agent Zabbix.

Se você estiver usando o agent Zabbix:

```
systemctl restart zabbix-agent
```

Se você estiver usando o agent Zabbix 2:

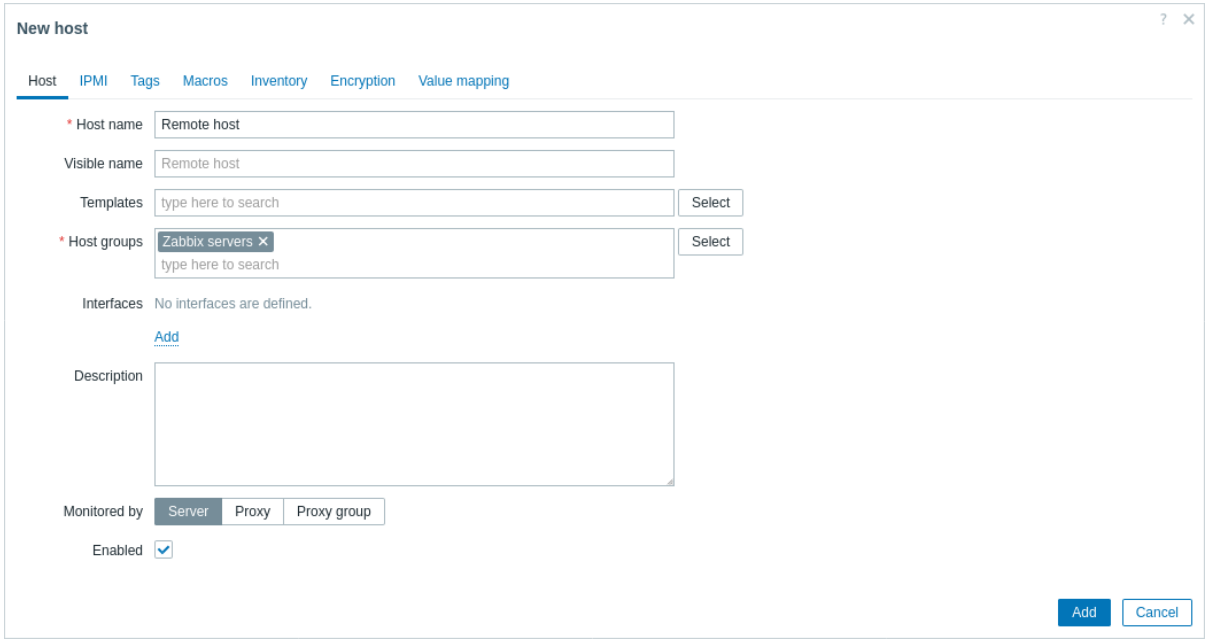
```
systemctl restart zabbix-agent2
```

Frontend do Zabbix

1. Faça login no frontend do Zabbix.

2. Crie um host na interface web do Zabbix.

- No campo *Nome do host*, insira um nome de host (por exemplo, ”Host remoto”) que corresponda ao valor do parâmetro *Hostname* definido anteriormente no arquivo de configuração do agent.
- No campo *Grupos de hosts*, digite ou selecione um grupo de hosts (por exemplo, ”Zabbix servers”).
- Deixe *Interfaces* indefinido; uma interface não é necessária para checagens ativas porque o agent inicia a conexão com o server em vez de aguardar uma conexão do server.

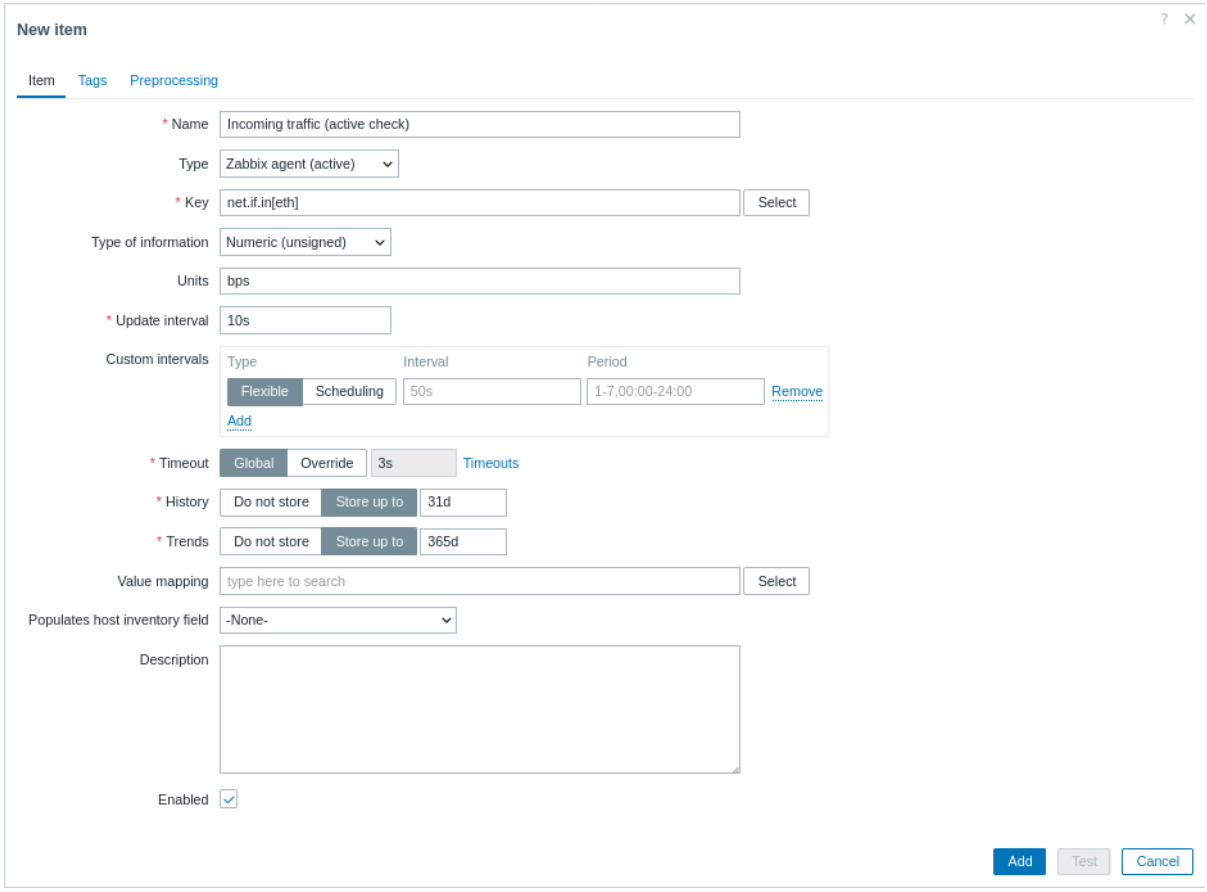


3. Clique em *Adicionar* para adicionar o host. Este host representará a máquina Linux monitorada.

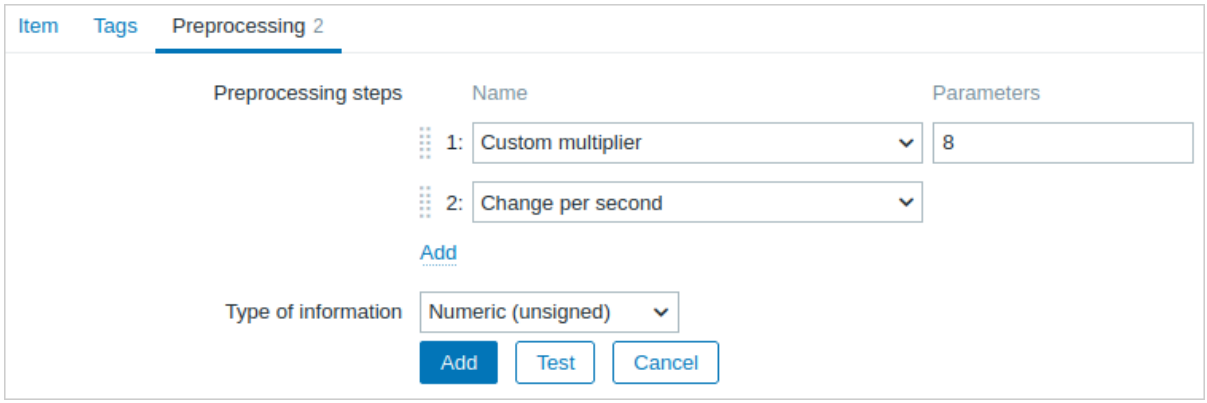
Criar items Siga as instruções em criando um item para adicionar os items para monitoramento de tráfego, a saber:

- Tráfego de entrada
- Tráfego de saída
- Tráfego total

Uma configuração simples para o item de monitoramento de tráfego de entrada com uma checagem ativa ficaria assim:

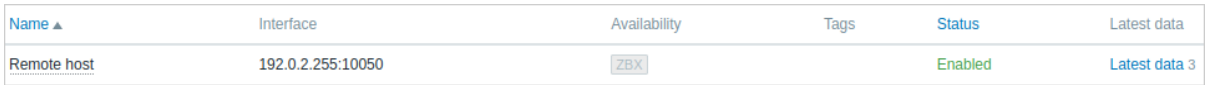


Para tornar os dados coletados adequados para uso prático, você pode definir algumas etapas de pré-processamento ao criar os items. No caso dado, essas etapas podem ser multiplicação por 8 (para converter bytes em bits) e apresentação como alteração por segundo.

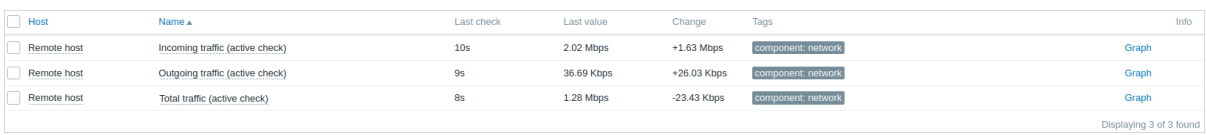


Visualizar dados coletados Parabéns! Neste ponto, o Zabbix já está monitorando o tráfego da sua rede.

Para visualizar as métricas coletadas, abra a seção de menu *Monitoramento → Hosts* e clique em *Últimos dados* na linha do host.



Você verá os dados de tráfego.



Veja também:

- Visualizar gráficos - como exibir os dados coletados em gráficos.
- Configurar triggers - como configurar triggers para detectar tráfego de rede anormal.
- Configurar alertas de problemas - como configurar notificações sobre situações de problema.

- Escalonamento de problemas - como criar cenários de alerta em várias etapas (por exemplo, primeiro enviar mensagem ao administrador do sistema, depois, se o problema não for resolvido em 45 minutos, enviar mensagem ao gerente do data center).
- Reconhecimento de problemas - como indicar que o problema é conhecido, fazer comentários sobre sua resolução, suprimir ou fechar o problema.
- Monitorar Linux com o agent Zabbix - como iniciar o monitoramento básico dos itens mais importantes vinculando um template pré-configurado.
- Instalação a partir de pacotes - como instalar componentes do Zabbix usando pacotes oficiais RPM e DEB para várias distribuições Linux, garantindo acesso aos recursos mais recentes e correções de bugs.

8 Monitorar sites com itens de Navegador

Introdução

Esta página orienta você nas etapas necessárias para iniciar o monitoramento básico de sites com itens do tipo Browser.

Para quem é este guia

Este guia foi desenvolvido para novos usuários do Zabbix e contém o conjunto mínimo de etapas necessárias para habilitar o monitoramento básico de sites com items do tipo Browser. Se você procura opções de personalização avançadas ou precisa de uma configuração mais detalhada, consulte a página Items do tipo Browser ou a seção Configuração do manual do Zabbix.

Pré-requisitos

Antes de prosseguir com este guia, você precisa baixar e instalar o Zabbix server e o Zabbix frontend de acordo com as instruções para o seu sistema operacional.

Dependendo do seu ambiente, alguns passos deste guia podem ser ligeiramente diferentes. Este guia é baseado em um ambiente rodando Ubuntu.

Configurar o WebDriver

Itens de navegador requerem um framework de automação (Selenium Server ou um WebDriver simples, por exemplo, ChromeDriver) como um endpoint de teste web que controla e interage com um navegador, executando comandos de teste como clicar em botões ou inserir texto. Como exemplo, este guia usará o Selenium Server com Chrome em um container Docker.

Pressupõe-se que o Docker já esteja configurado. Este guia não cobre a configuração do Docker. Para instruções de instalação, consulte Instalar Docker Engine no Ubuntu.

1. Inicie o Selenium Server com Chrome em um container Docker com as seguintes opções:

- **docker run --name browser** - executa um novo container Docker chamado "browser";
- **-p 4444:4444** - mapeia a porta 4444 da sua máquina host para a porta 4444 do container (esta é a porta usada pelo Selenium Server para aceitar comandos);
- **-p 7900:7900** - mapeia a porta 7900 da sua máquina host para a porta 7900 do container (esta é a porta usada pelo servidor Virtual Network Computing (VNC), permitindo que você visualize a interface gráfica do navegador remotamente; requer um cliente VNC);
- **--shm-size="2g"** - aloca 2GB de memória compartilhada para o container (isso é importante para o Chrome funcionar corretamente, pois pode exigir uma quantidade significativa de memória compartilhada para evitar falhas);
- **-d** - executa o container em modo destacado, ou seja, ele será executado em segundo plano;
- **selenium/standalone-chrome:latest** - especifica a imagem Docker a ser usada; neste caso, a versão mais recente do Selenium Server com Chrome.

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

2. Certifique-se de que o container Docker `browser` está em execução e acessível.

- Recupere o endereço IP do container (neste exemplo, 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
###    ...
```

- Teste a conexão com o container usando o Ncat:

```
nc -zv 192.0.2.1 4444

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

- Recupere o conteúdo da página web do Selenium Server usando o 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:
Para solução de problemas, consulte a documentação do Docker.

Configurar o servidor Zabbix

Os itens do navegador são executados e processados pelos processos *browser poller* do Zabbix, que precisam ser habilitados ajustando o parâmetro de configuração do servidor StartBrowserPollers. Além disso, o parâmetro WebDriverURL deve especificar o endpoint de teste web configurado anteriormente.

Por padrão, o parâmetro StartBrowserPollers é definido como 1, portanto, você só precisa especificar o endpoint de teste web.

1. Abra o arquivo de configuração do servidor Zabbix.

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

2. Localize e defina o parâmetro WebDriverURL no arquivo de configuração do servidor Zabbix:

```
##### Option: WebDriverURL
###WebDriver interface HTTP[S] URL. For example http://localhost:4444 used with Selenium WebDriver standal
###
### Mandatory: no
### Default:
### WebDriverURL=

WebDriverURL=192.0.2.1:4444
```

3. Reinicie o servidor Zabbix.

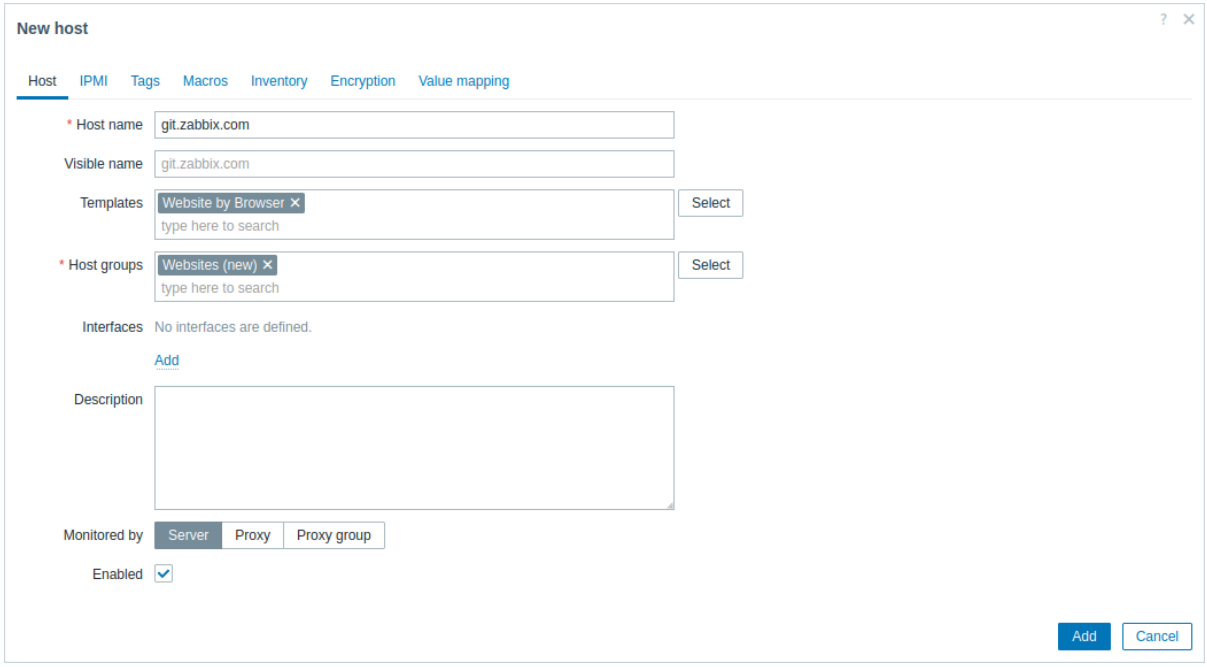
```
systemctl restart zabbix-server
```

Configure o frontend do Zabbix

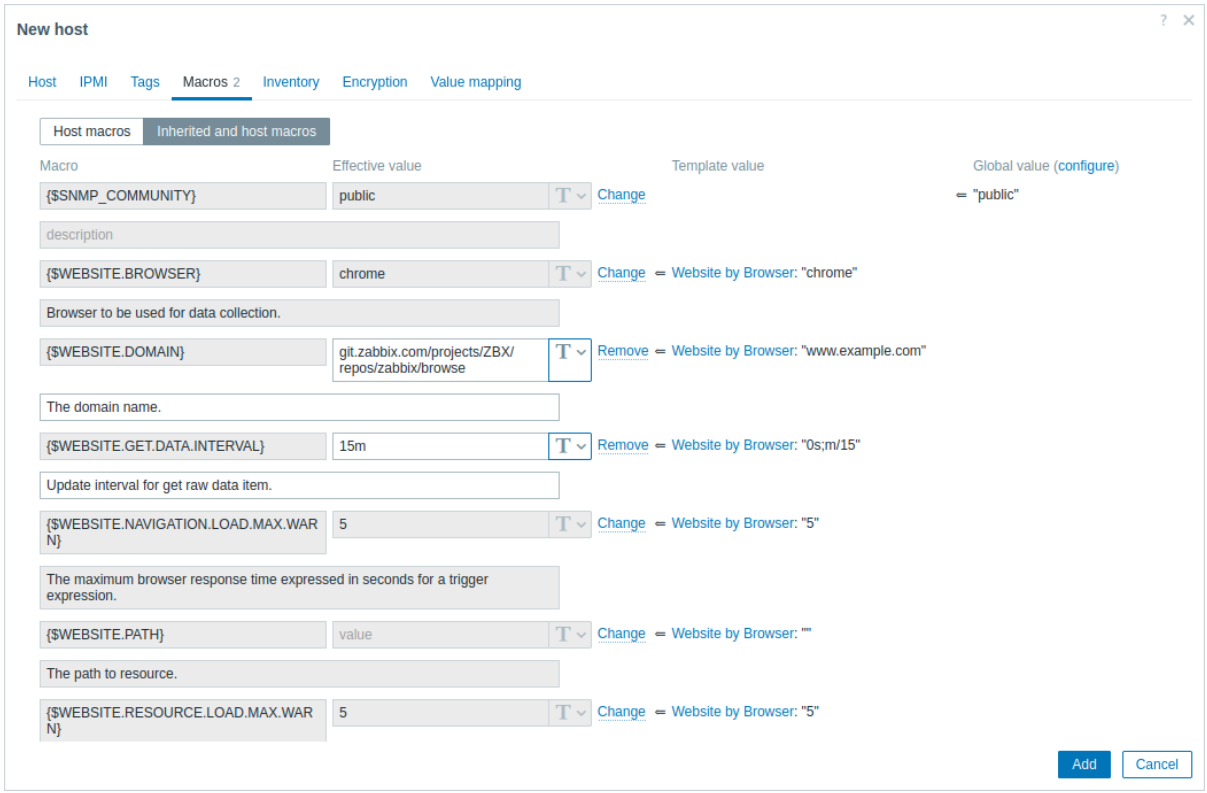
1. Faça login no frontend do Zabbix.

2. Crie um host na interface web do Zabbix:

- No campo *Nome do host*, insira um nome de host (por exemplo, ”git.zabbix.com”).
- No campo *Templates*, digite ou selecione o template ”Website by Browser”. Para mais informações sobre este template, consulte Website by Browser.
- No campo *Grupos de hosts*, digite ou selecione um grupo de hosts (por exemplo, um novo grupo de hosts ”Websites”).



- Na guia *Macros*, alterne para *Macros herdadas e do host*, procure as seguintes macros e clique em *Alterar* ao lado do valor da macro para atualizá-lo:
  - {$WEBSITE.DOMAIN} - Nome do domínio (por exemplo, git.zabbix.com/projects/ZBX/repos/zabbix/browse)
  - {$WEBSITE.GET.DATA.INTERVAL} - Intervalo de atualização dos dados do item (por exemplo, 15m)

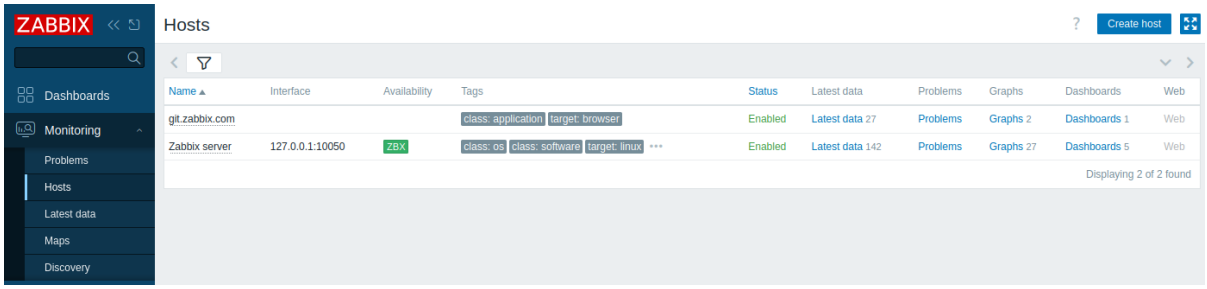


3. Clique no botão *Adicionar* para criar o host. Este host representará o site que você deseja monitorar.

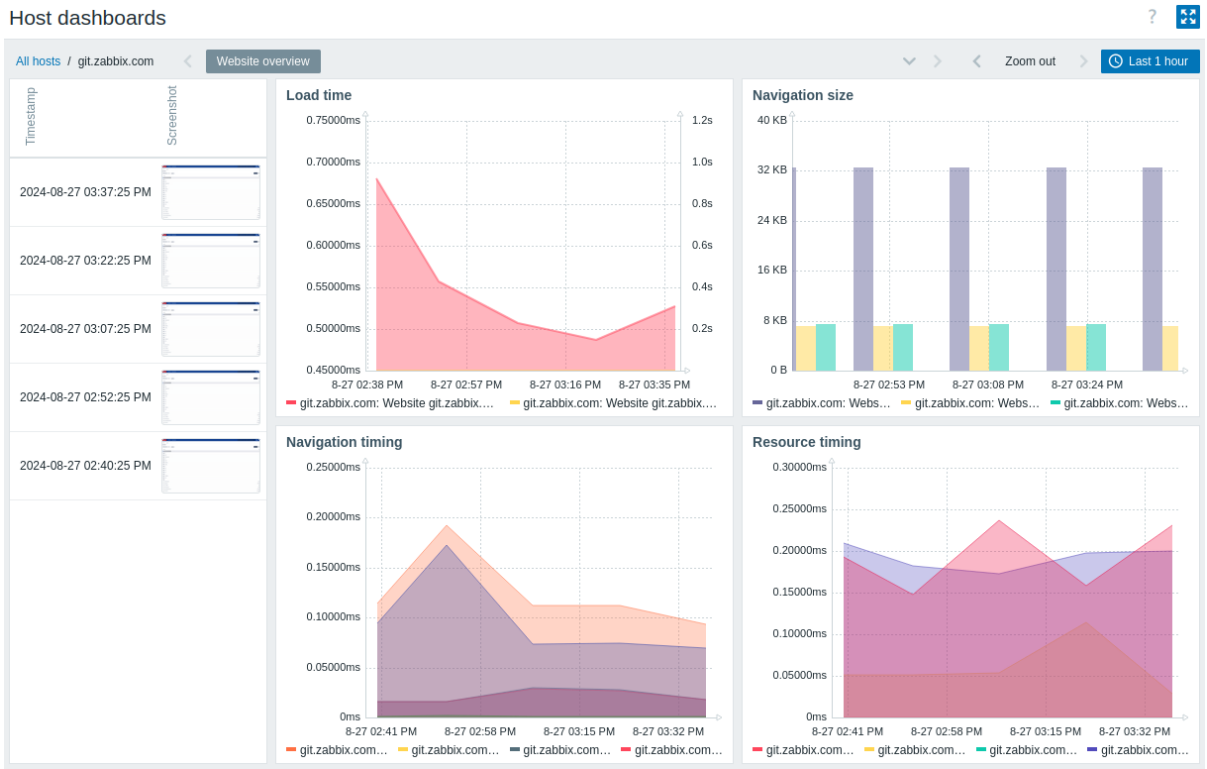
Visualizar métricas coletadas

Parabéns! Neste ponto, o Zabbix já está monitorando o site que você especificou.

Para visualizar as métricas coletadas, navegue até a seção de menu *Monitoramento → Hosts* e clique em *Dashboards* ao lado do host.



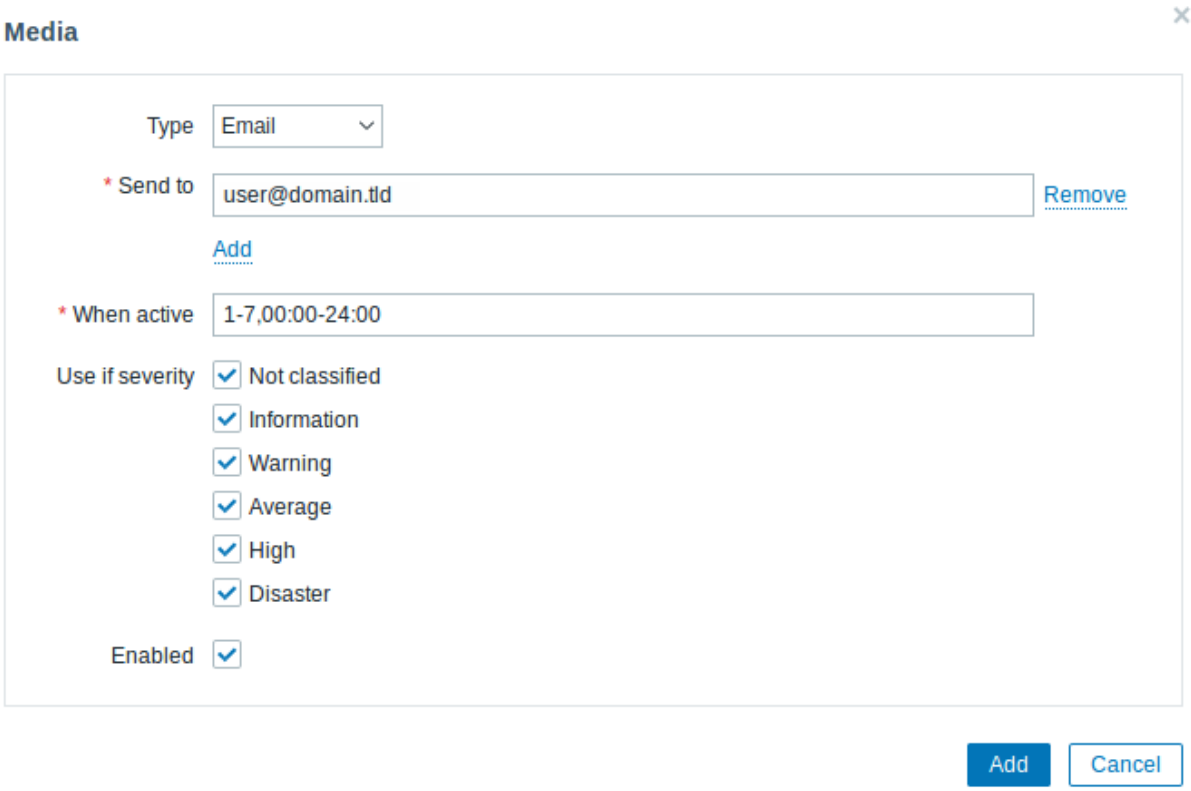
Esta ação o levará ao dashboard do host (configurado no nível do template) com as métricas mais importantes coletadas do site.



Configure alertas de problemas

O Zabbix pode notificá-lo sobre um problema em sua infraestrutura usando vários métodos. Este guia fornece etapas básicas de configuração para o envio de alertas por e-mail.

1. Navegue até *Configurações do usuário → Perfil*, vá para a guia *Mídia* e adicione seu e-mail.



2. Siga o guia para Receber uma notificação de problema.

Da próxima vez que o Zabbix detectar um problema, você deverá receber um alerta por e-mail.

Teste sua configuração

Para testar sua configuração, podemos simular um problema real atualizando a configuração do host no frontend do Zabbix.

1. Abra a configuração do host do seu site no Zabbix.

2. Mude para a guia *Macros* e selecione *Macros herdadas e do host*.

3. Clique em *Alterar* ao lado, por exemplo, do valor da macro {$WEBSITE.DOMAIN} configurada anteriormente e defina um nome de domínio incorreto (por exemplo, /git.zabbix.com/projects/ZBX/repos/zabbix/browse).

4. Clique em *Atualizar* para atualizar a configuração do host.

5. Em alguns instantes, o Zabbix detectará o problema "Falha ao obter o JSON do site solicitado", pois não conseguirá se conectar ao site especificado. O problema aparecerá em *Monitoramento → Problemas*.

Se os alertas estiverem configurados, você também receberá a notificação do problema.

6. Altere o valor da macro de volta para o valor anterior para resolver o problema e continuar monitorando o site.

Veja também

- Criando um item - como começar a monitorar métricas adicionais.
- Escalonamentos de problemas - como criar cenários de alerta em várias etapas (por exemplo, primeiro enviar mensagem para o administrador do sistema, depois, se um problema não for resolvido em 45 minutos, enviar mensagem para o gerente do data center).
- Itens de Navegador - como configurar itens de Navegador.
- Template Website by Browser - informações adicionais sobre o template *Website by Browser*.

9 Monitorar certificados de sites com o Zabbix agent 2 (passivo)

Introdução

Este guia fornece uma visão geral abrangente de como configurar e monitorar certificados SSL/TLS usando a chave `web.certificate.get` no Zabbix agent 2. Ele foi projetado para simplificar o monitoramento de certificados para um ou vários sites, garantindo que os administradores possam identificar rapidamente possíveis problemas, como certificados expirados ou inválidos.

Para quem é este guia

Este guia foi desenvolvido para novos usuários do Zabbix e contém o conjunto mínimo de etapas necessárias para habilitar o monitoramento básico de certificados de sites. Se você procura opções de personalização avançadas ou precisa de uma configuração mais avançada, consulte a seção Configuração do manual do Zabbix.

Pré-requisitos

Antes de prosseguir com este guia, você precisa baixar e instalar o Zabbix server, o Zabbix frontend e o Zabbix agent 2 de acordo com as instruções para o seu sistema operacional. Este tutorial assume que tanto o Zabbix server quanto o agent estão instalados na mesma máquina; portanto, `127.0.0.1` é usado na configuração.

Configure o Zabbix agent 2

1. Abra o arquivo de configuração do Zabbix agent (caminho padrão: `/etc/zabbix/zabbix_agent2.conf`):

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

2. Defina o parâmetro Server como 127.0.0.1, já que o agent e o server estão rodando na mesma máquina:

```
Server=127.0.0.1
```

3. Salve o arquivo e reinicie o serviço do Zabbix agent 2:

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

4. Após instalar e configurar o Zabbix agent 2, teste sua disponibilidade com:

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

O Zabbix agent 2 inclui o plugin WebCertificate por padrão, portanto, nenhuma instalação ou configuração separada é necessária.

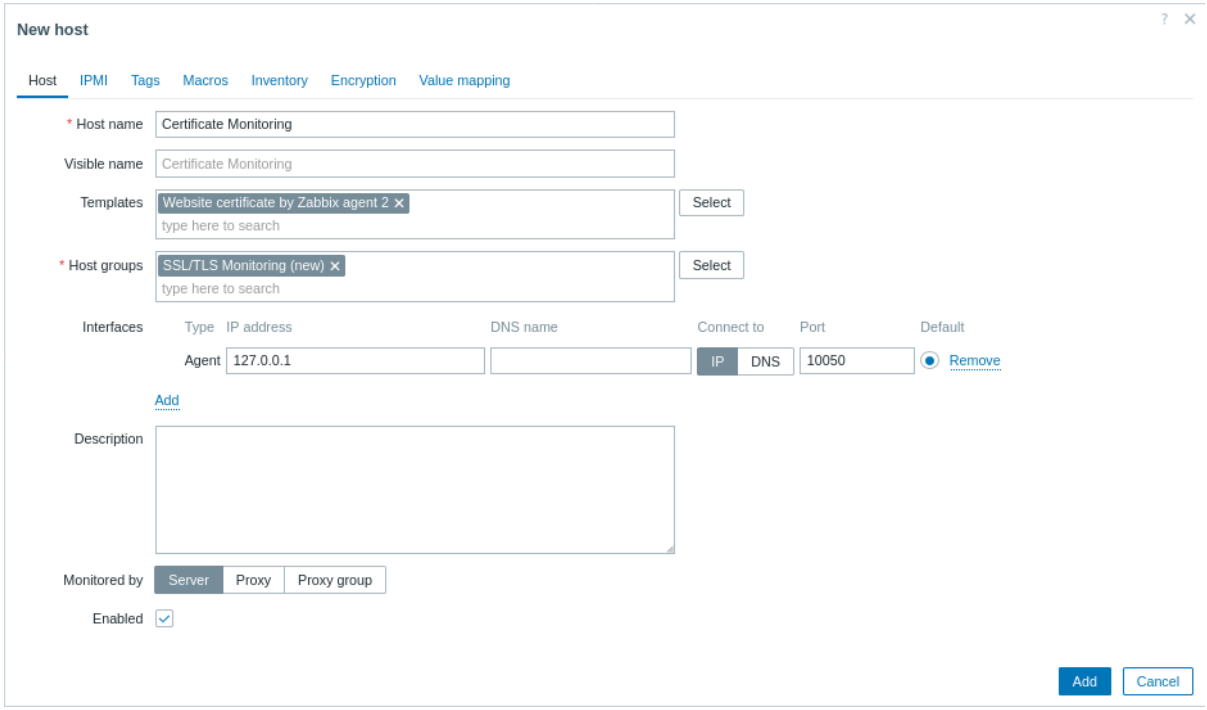
Configure o frontend do Zabbix

1. Faça login no frontend do Zabbix.

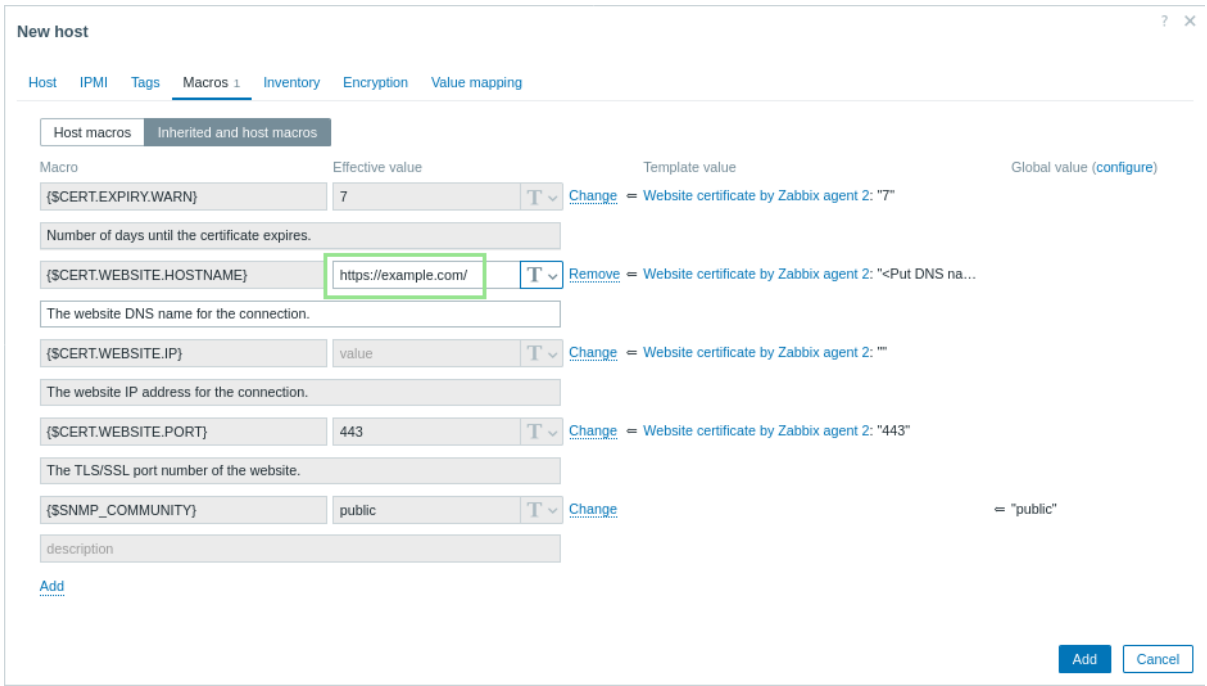
2. Navegue até *Monitoramento* > *Hosts*.

3. Clique em um host existente onde deseja monitorar certificados de sites ou crie um host se necessário:

- No campo *Nome do host*, insira um nome de host (por exemplo, ”Monitoramento de Certificado”).
- No campo *Templates*, digite ou selecione o template ”Website certificate by Zabbix agent 2” que será vinculado ao host.
- No campo *Grupos de hosts*, digite ou selecione um grupo de hosts (por exemplo, ”Monitoramento SSL/TLS”).
- No campo *Interfaces*, adicione uma interface do tipo ”Agent” e especifique um endereço IP. Neste exemplo, ”127.0.0.1” é usado.



- Na guia *Macros*, alterne para *Macros herdadas e do host*, procure as seguintes macros e clique em *Alterar* ao lado do valor da macro para atualizá-lo:
  - `{$CERT.WEBSITE.HOSTNAME}` - insira o nome DNS do site desejado como valor.



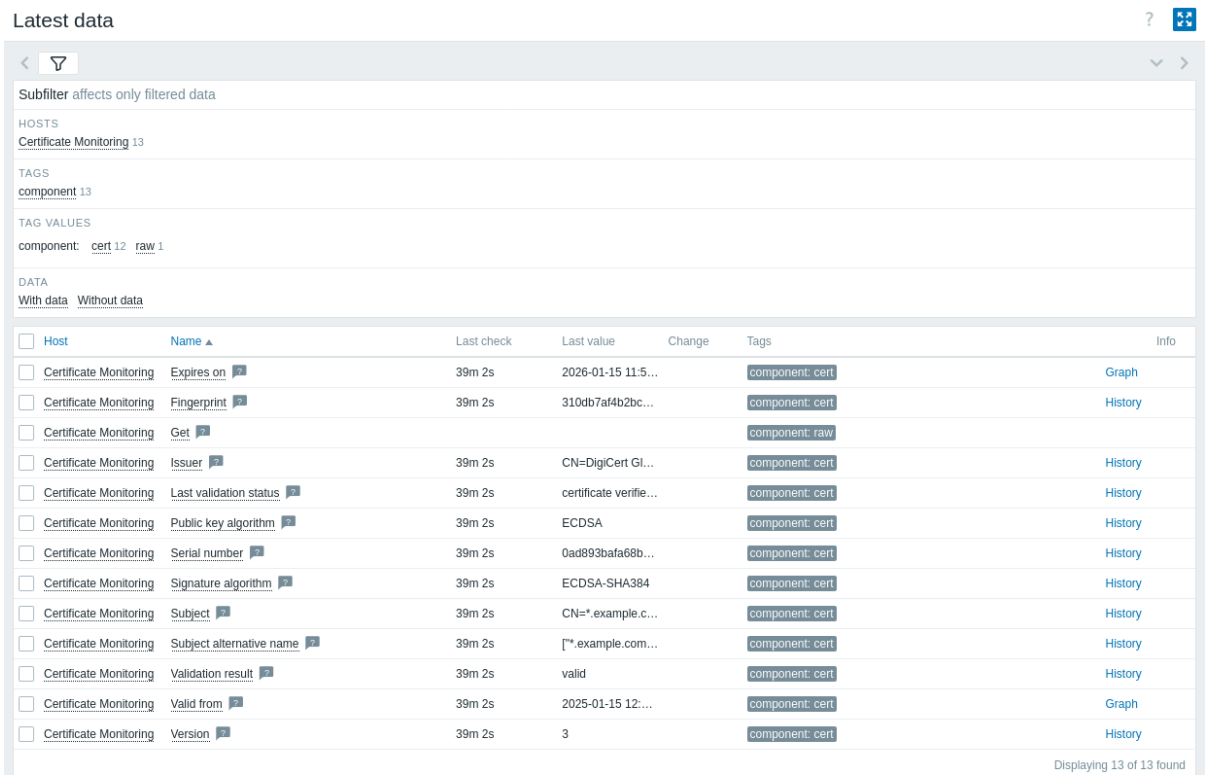
4. Clique em *Adicionar* para adicionar o host.

Para monitorar vários sites, repita as etapas acima para criar hosts separados para cada site. Cada host deve usar o template "Website certificate by Zabbix agent 2" e ser configurado com as macros correspondentes. Para simplificar o processo, considere clonar hosts existentes. Ao clonar, certifique-se de atualizar tanto o *Nome do host* quanto a macro `{$CERT.WEBSITE.HOSTNAME}` com os valores desejados.

Visualizar métricas coletadas

Parabéns! Neste ponto, o Zabbix já está monitorando o certificado web desejado.

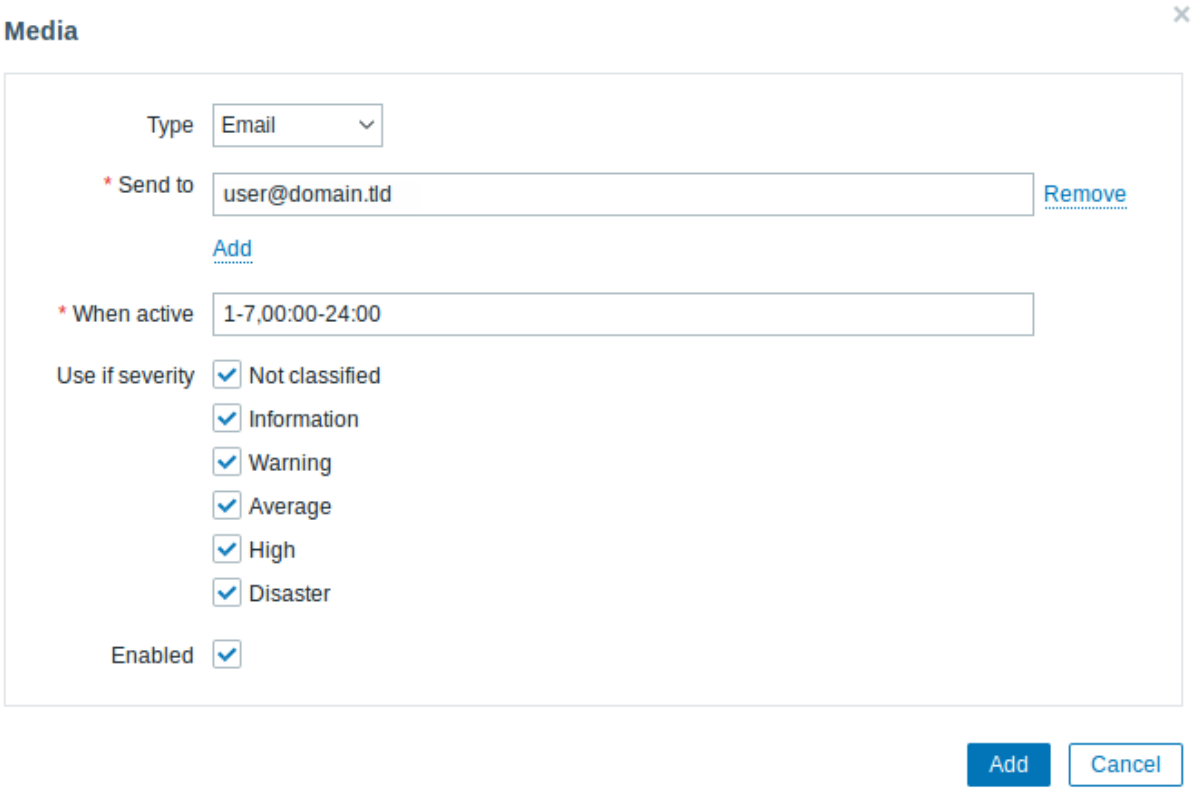
Para visualizar as métricas coletadas, navegue até a seção de menu *Monitoramento > Hosts* e clique em *Últimos dados* ao lado do host para visualizar todas as métricas coletadas mais recentes em uma lista, como data de expiração, emissor e assunto.



Configure alertas de problemas

O Zabbix pode notificá-lo sobre problemas de infraestrutura. Este guia fornece etapas básicas de configuração para o envio de alertas por e-mail.

1. Navegue até *Configurações do usuário > Perfil*, vá para a guia *Mídia* e adicione seu e-mail.



2. Siga o guia para Receber uma notificação de problema.

Da próxima vez que o Zabbix detectar um problema, você deverá receber um alerta por e-mail.

Teste sua configuração

Para testar sua configuração, podemos simular um problema real atualizando a configuração do host no frontend do Zabbix.

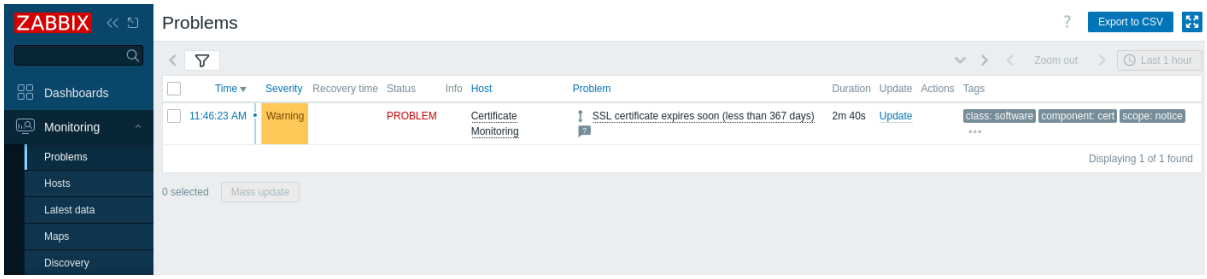
1. Abra a configuração do seu host ”Certificate Monitoring” no Zabbix.

2. Mude para a aba *Macros* e selecione *Macros herdadas e do host*.

3. Clique em *Alterar* ao lado do valor da macro `{$CERT.EXPIRY.WARN}` configurada anteriormente e defina um número de dias muito alto (maior que 365 dias deve ser suficiente) para receber um aviso antes que o certificado expire.

4. Clique em *Atualizar* para atualizar a configuração do host.

5. Em alguns instantes, o Zabbix detectará o problema "SSL certificate expires soon", com o número de dias até a expiração. O problema aparecerá em *Monitoramento > Problemas*.



Se os alertas estiverem configurados, você também receberá a notificação do problema.

6. Altere o valor da macro de volta ao valor anterior para resolver o problema e continuar monitorando os valores do certificado.

Veja também

- Agent Zabbix 2 - lista as chaves de item.
- Template *Website certificate by Zabbix agent 2* - informações adicionais sobre o template *Website certificate by Zabbix agent 2*.
- Template *Website certificate by Zabbix agent 2 active* - informações adicionais sobre o template *Website certificate by Zabbix agent 2 active*.
- Monitorar sites com itens Browser - como iniciar o monitoramento básico de sites com itens Browser.

10 Monitore um switch ou roteador de rede com o Zabbix

Introdução Este guia orienta você nas etapas necessárias para iniciar o monitoramento básico do seu switch ou roteador de rede usando o Zabbix. Um roteador Cisco é usado como exemplo, mas o procedimento se aplica a qualquer dispositivo de rede habilitado para SNMP.

Para quem é este guia

Este guia foi desenvolvido para novos usuários do Zabbix e administradores de rede que desejam ativar rapidamente o monitoramento básico de dispositivos de rede. Se você precisar de personalização avançada ou opções de configuração avançadas, consulte a página do agente SNMP ou a seção Configuração do manual do Zabbix.

Pré-requisitos

Antes de prosseguir com este guia, certifique-se de que você tenha:

- Zabbix server e Zabbix frontend instalados: instale de acordo com as instruções para o seu sistema operacional (consulte Instalação a partir de pacotes e Instalação da interface web).
- Zabbix agent instalado se for monitorar métricas de rede locais.
- Dispositivo habilitado para SNMP: um switch ou roteador de rede (por exemplo, um roteador Cisco) com SNMP ativado.
- Arquivos MIB instalados: a instalação dos arquivos MIB permite que o Zabbix traduza OIDs numéricos em nomes e descrições legíveis. Sem o suporte adequado a MIB, você pode ver apenas valores numéricos, tornando mais difícil configurar items e solucionar problemas.

Para instalar arquivos MIB no Ubuntu:

1. Instale o pacote de download de MIB:

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

Se você precisar adicionar MIBs específicas de fornecedores (por exemplo, da Cisco, Juniper), coloque-as no diretório MIB apropriado:

- Para sistemas baseados em Linux, os locais comuns incluem /usr/share/snmp/mibs/ ou /usr/local/share/snmp/mibs/.
- Para instalações do Zabbix, os arquivos MIB podem ser armazenados em /var/lib/zabbix/mibs/.

Certifique-se de que a variável de ambiente MIBDIRS ou o arquivo snmp.conf inclua o caminho correto.

Para verificar se o seu sistema reconhece as novas MIBs, use:

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

Para instruções detalhadas, consulte a documentação da sua biblioteca SNMP:

- Cisco MIBs
- Juniper MIBs

2. Edite o arquivo `/etc/snmp/snmp.conf` e comente a linha que começa com `mibs :` para permitir que o sistema carregue todas as MIBs disponíveis.

3. Verifique executando um `snmpwalk` (por exemplo, `snmpwalk -v 2c -c <sua_string_de_comunidade> <IP_do_dispositivo>`) e verifique se os OIDs são exibidos com nomes descritivos.

Dependendo do seu ambiente, algumas etapas deste guia podem ser ligeiramente diferentes. Este guia é baseado em um ambiente executando Ubuntu e um dispositivo de rede Cisco Catalyst 3750V2-24FS sendo monitorado.

Pressupõe-se que seu dispositivo de rede já esteja fisicamente instalado e conectado.

Configurar o dispositivo de rede (exemplo de roteador Cisco) Para monitoramento via SNMP, você deve configurar seu dispositivo de rede para permitir consultas SNMP. O exemplo abaixo é para SNMPv2 e não leva em consideração as configurações existentes. Atenção: aplicar esses comandos pode substituir as configurações SNMP atuais.

Para um roteador Cisco, a configuração normalmente envolve as etapas descritas abaixo.

Exemplo SNMPv2 1. Habilite o SNMP e defina a string da comunidade.

Faça login no console do seu roteador Cisco e entre no modo de configuração:

```
configure terminal
```

Em seguida, habilite o SNMP especificando uma string de comunidade somente leitura. Por exemplo:

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

Substitua `<your_community_string>` pela sua string de comunidade segura. Nota: a opção RO (Read-Only) permite que o SNMP recupere dados do dispositivo, mas impede quaisquer alterações de configuração.

Recomenda-se restringir o acesso SNMP apenas aos dispositivos necessários por motivos de segurança. Para obter mais orientações sobre como configurar listas de controle de acesso (ACLs), consulte a documentação oficial da Cisco.

2. Salve a configuração.

Salve suas alterações para garantir que as configurações do SNMP persistam após uma reinicialização:

```
write memory
```

Exemplo SNMPv3 O SNMPv3 oferece segurança aprimorada com autenticação e criptografia. Sua configuração é mais segura do que o SNMPv2 e deve ser verificada de acordo com a documentação específica do seu dispositivo.

1. Crie um grupo SNMP.

Configure um grupo SNMPv3 com privacidade (criptografia) ativada:

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

2. Crie um usuário SNMP.

Adicione um usuário SNMPv3 com autenticação e privacidade. Substitua os espaços reservados pelos valores desejados:

```
snmp-server user <seu_usuario> <seu_grupo> v3 auth md5 <senha_autenticacao> priv aes 128 <senha_privacidad
```

3. Salve a configuração:

```
write memory
```

Para mais detalhes ou instruções específicas para o modelo, consulte os tutoriais de configuração SNMP da Cisco. Este guia, no entanto, fornece as etapas básicas para habilitar o monitoramento SNMP.

Configurar o frontend do Zabbix

Crie um host no frontend do Zabbix 1. Faça login no frontend do Zabbix.

2. Adicione um novo host.

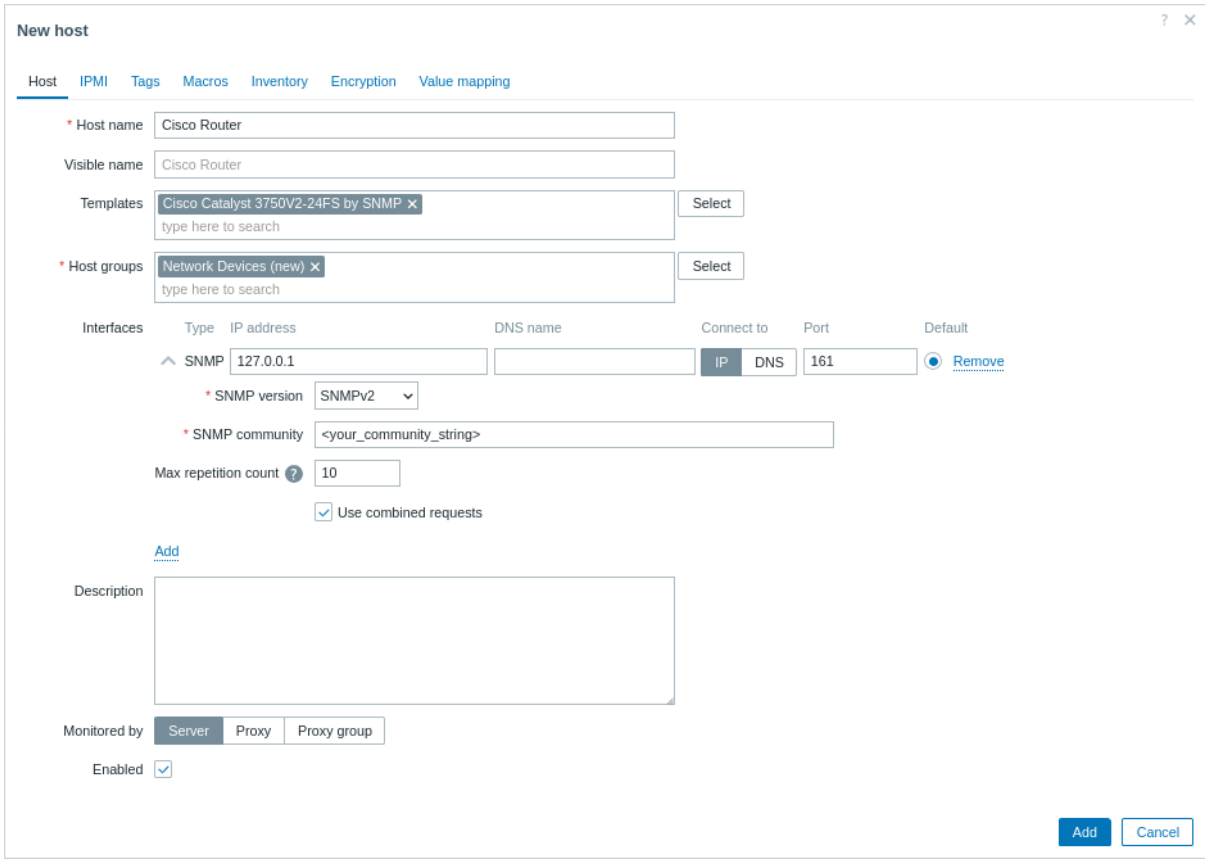
Navegue até *Coleta de dados* > *Hosts* e clique em *Criar host*.

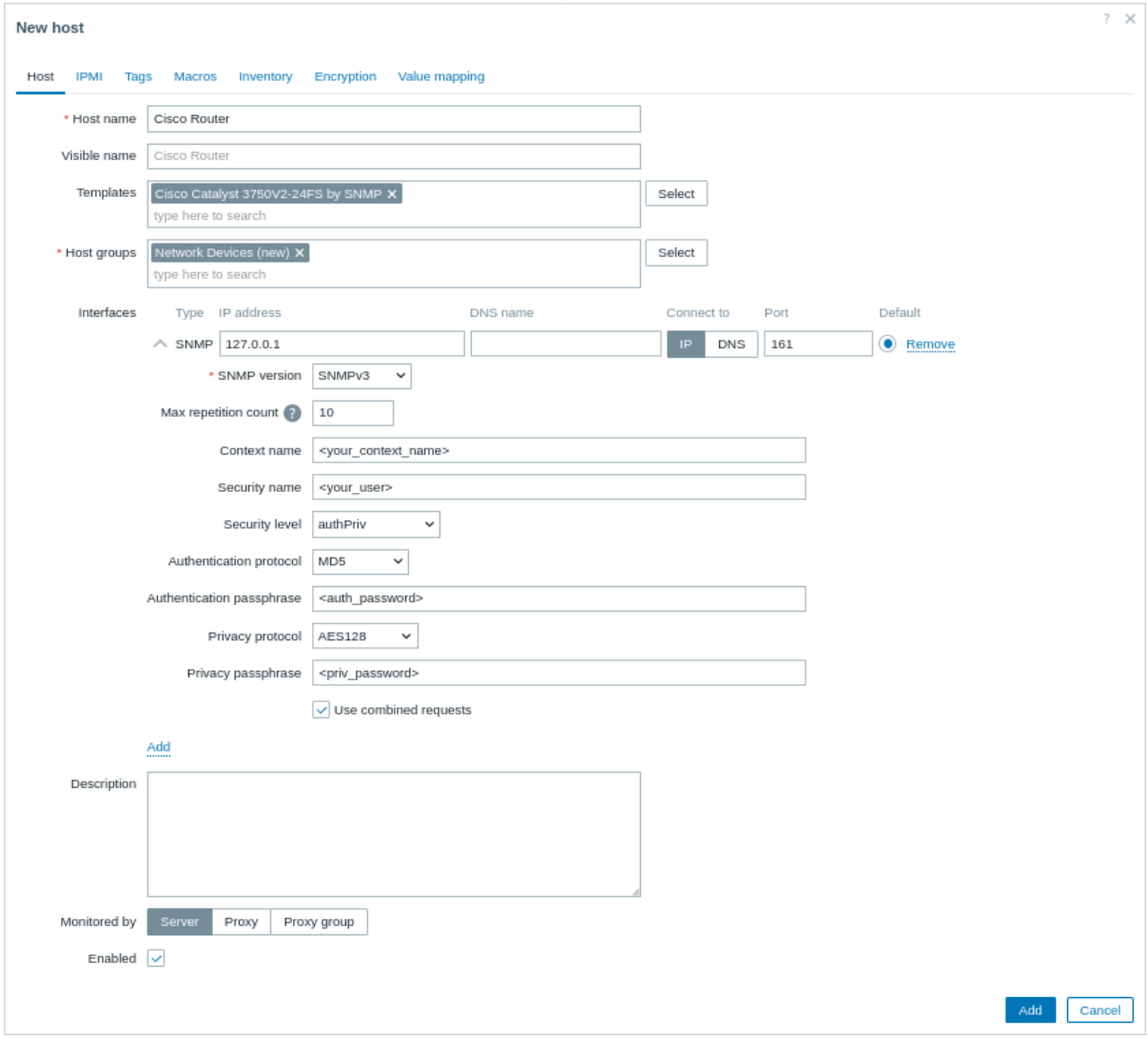
- *Nome do host*: insira um nome para o seu dispositivo (por exemplo, ”Cisco Router”).
- Grupos de hosts: selecione um grupo existente ou crie um novo grupo, como ”Dispositivos de Rede”.
- Interfaces:
  - Clique em *Adicionar* em Interfaces.
  - Escolha *SNMP* como o tipo de interface.
  - Insira o endereço IP ou nome DNS do seu roteador Cisco.
  - Defina a porta SNMP padrão (geralmente 161).
  - Use o menu suspenso para selecionar a versão SNMP apropriada (por exemplo, SNMPv2).
  - Para SNMPv1/v2, insira a string de comunidade no campo *Comunidade SNMP*. Para SNMPv3, credenciais adicionais (*Nome do contexto*, *Nome de segurança* e *Nível de segurança*, etc.) serão solicitadas.

3. Vincule templates

No campo *Templates*, selecione o template SNMP que melhor corresponde ao seu dispositivo. O Zabbix fornece uma variedade de templates SNMP pré-construídos para várias famílias de dispositivos. Por exemplo, se você estiver monitorando um dispositivo Cisco, escolha o template que corresponde ao sistema operacional ou modelo do seu dispositivo (como Cisco IOS SNMP ou Cisco Catalyst 3750<modelo do dispositivo> SNMP).

4. Clique em *Adicionar* para salvar o host.

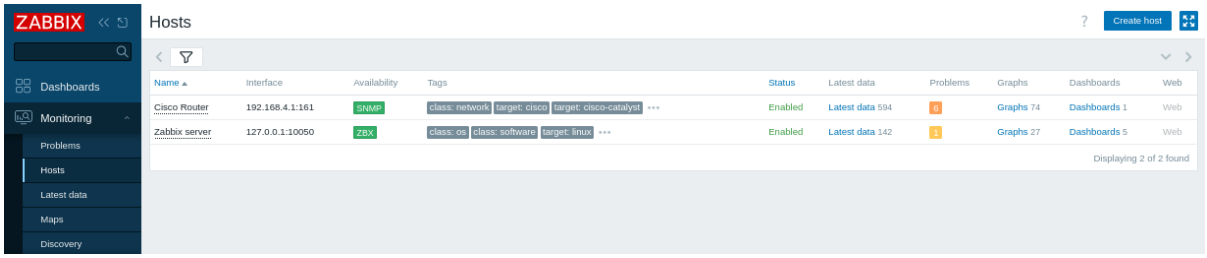




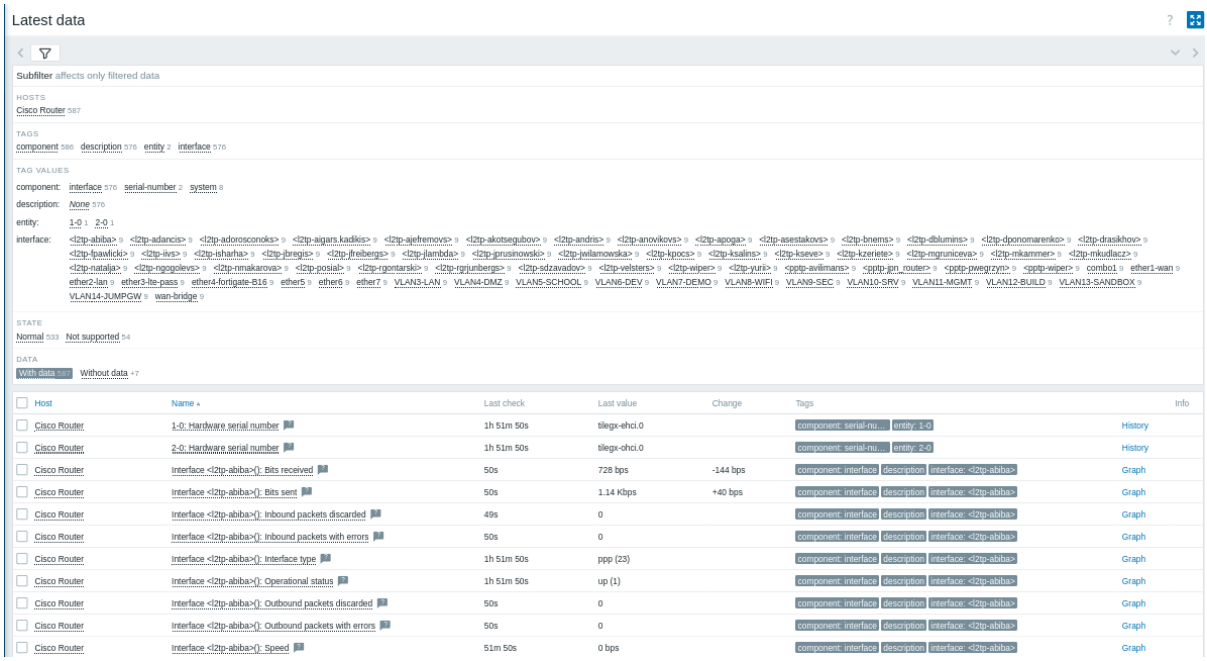
Visualizar métricas coletadas Parabéns! O Zabbix agora está configurado para monitorar seu dispositivo de rede.

Últimos Dados:

- Navegue até Monitoramento > Últimos dados no frontend do Zabbix.



- Selecione seu host “Cisco Router” (ou hosts descobertos) para visualizar métricas como tempo de atividade de hardware e rede, perda de ICMP, ping, tempo de resposta, etc.



- Gráficos e telas:

Para visualizar os dados de desempenho, clique em *Gráficos* ao lado dos itens SNMP para ver métricas detalhadas.

Como próximo passo, você pode:

- Adicionar itens SNMP personalizados para monitorar métricas adicionais.
- Configurar alertas de problemas para receber notificações sobre possíveis problemas.

Criar items SNMP Depois que o host estiver configurado, você pode criar items para monitorar métricas específicas. Nota: esta etapa é opcional se você estiver usando um template, pois os templates já contêm conjuntos padrão de items.

1. Identifique o OID SNMP:

Use o comando `snmpwalk` para listar os OIDs disponíveis em seu dispositivo. Por exemplo:

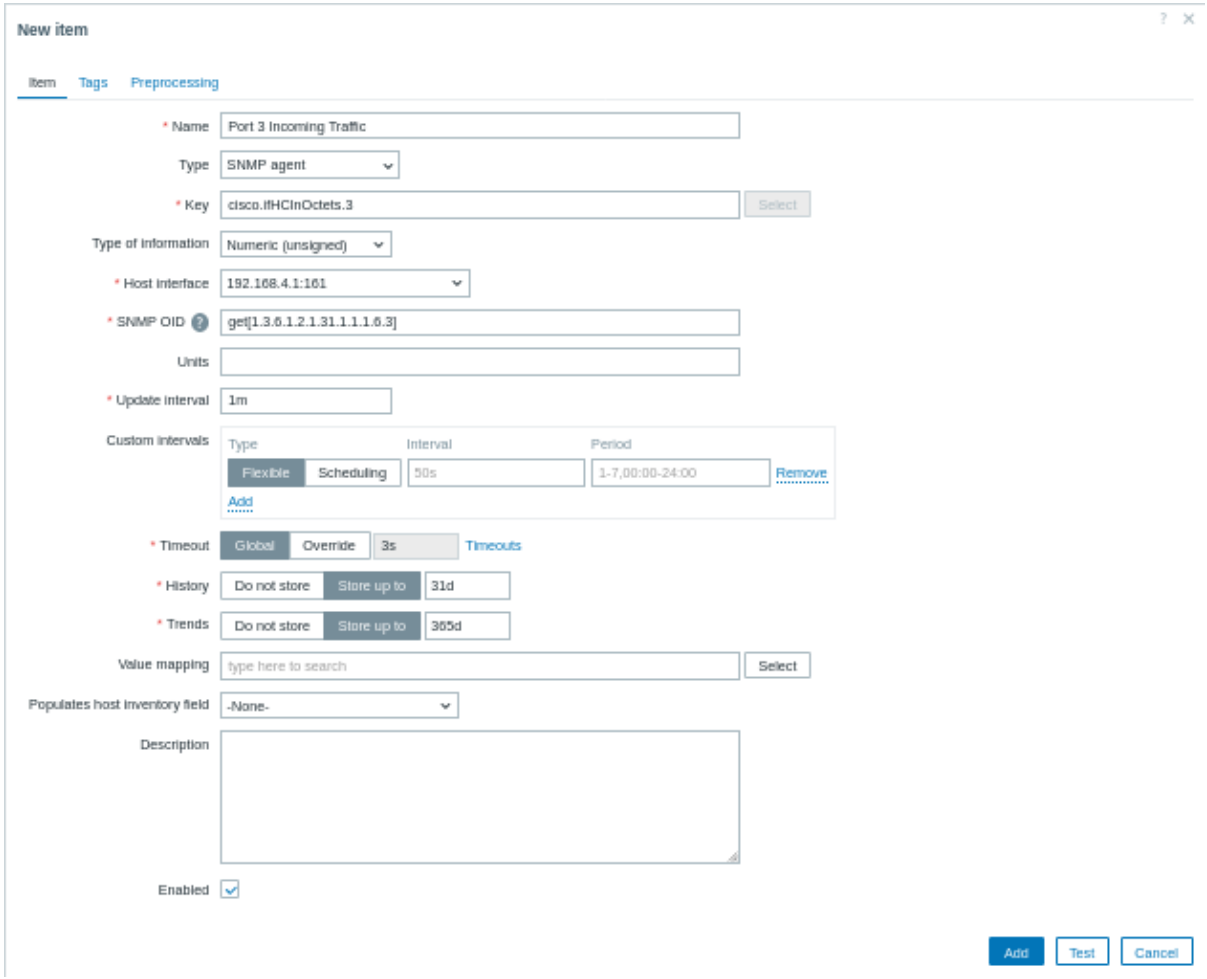
```
snmpwalk -v 2c -c <sua_string_de_comunidade> <IP_do_dispositivo> .
```

Encontre o OID para a métrica que deseja monitorar (por exemplo, IF-MIB::ifHCInOctets.3 para tráfego de entrada na porta 3). Para obter o OID numérico, você pode usar:

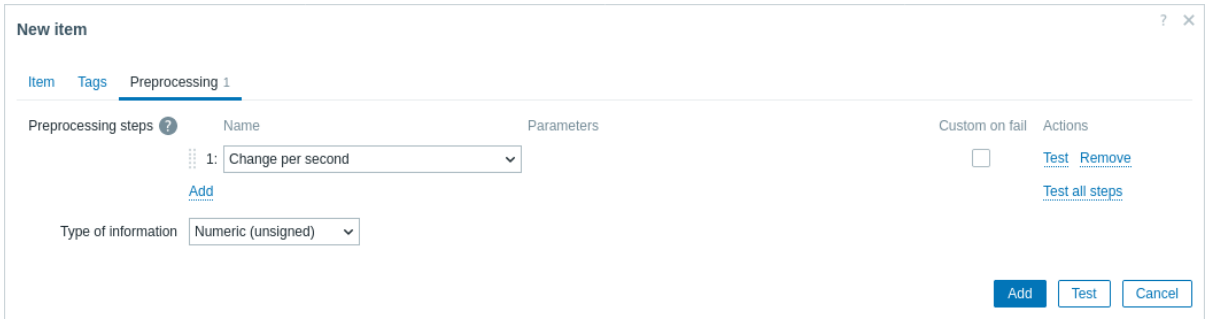
```
snmpget -v 2c -c <sua_string_de_comunidade> -On <IP_do_dispositivo> IF-MIB::ifHCInOctets.3
```

2. Crie um item SNMP:

- Navegue até *Coleta de dados* > *Hosts*, clique na guia *Items* do seu host SNMP e clique em *Criar item*.
- *Nome*: insira um nome descritivo (por exemplo, “Tráfego de entrada da porta 3”).
- *Tipo*: selecione *SNMP agent*.
- *Chave*: forneça uma chave significativa (por exemplo, `cisco.ifHCInOctets.3`).
- *Interface do host*: certifique-se de que a interface SNMP está selecionada.
- *OID SNMP*: insira o OID usando um dos formatos suportados, por exemplo:
  - `get[1.3.6.1.2.1.31.1.1.1.6.3]` para um único valor;
  - `walk[1.3.6.1.2.1.31.1.1.1.6.3]` para recuperar uma subárvore de valores de forma assíncrona.



- *Pré-processamento* (se necessário): se o item retornar um contador cumulativo (como tráfego de interface), navegue até a guia *Pré-processamento* e adicione uma etapa de pré-processamento como "Alteração por segundo" para calcular a taxa.



Para recuperar vários valores em uma única transação SNMP, você pode especificar vários OIDs usando a sintaxe `walk[OID1,OID2,...]`.

Traduzindo OIDs entre Números e Nomes de MIB Ao trabalhar com SNMP, pode ser necessário converter entre OIDs numéricos e seus respectivos nomes de MIB. Essa tradução ajuda a identificar e solucionar problemas de métricas com mais facilidade.

- Traduzindo um nome de MIB para um OID numérico: use o comando `snmptranslate` com a opção `-On`. Por exemplo, para traduzir o nome de MIB `IF-MIB::ifHCInOctets.3` para seu OID numérico, execute:

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

Esse comando pode retornar:

```
.1.3.6.1.2.1.31.1.1.1.6.3
```

- Traduzindo um OID numérico para seu nome de MIB: use o comando `snmptranslate` com a opção `-IR` (ou `-m ALL`) para reverter a tradução. Por exemplo, para traduzir o OID numérico `.1.3.6.1.2.1.31.1.1.1.6.3` de volta para seu nome de MIB, execute:

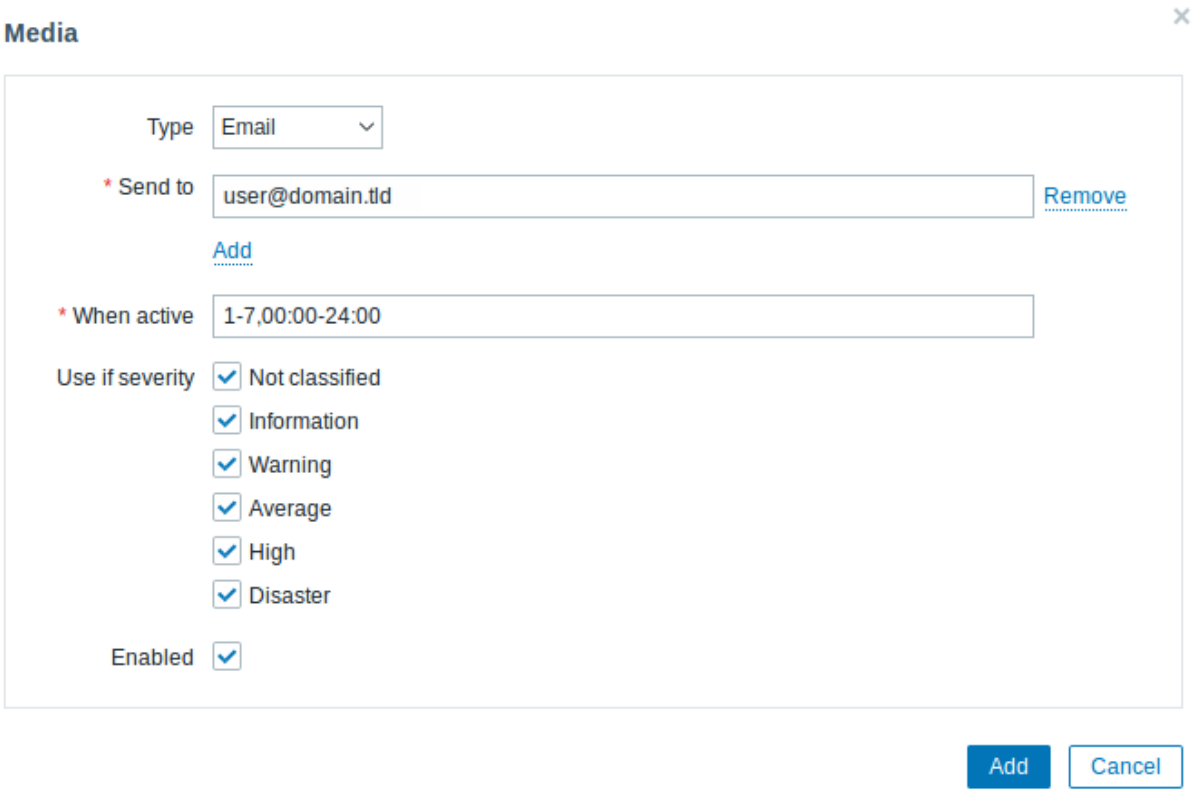
```
snmptranslate -IR -On .1.3.6.1.2.1.31.1.1.1.6.3
```

Esse comando pode retornar:

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

Configurar alertas de problemas Este guia fornece etapas básicas de configuração para o envio de alertas por e-mail.

1. Navegue até *Configurações do usuário > Perfil*, alterne para a guia *Mídia* e adicione seu e-mail.

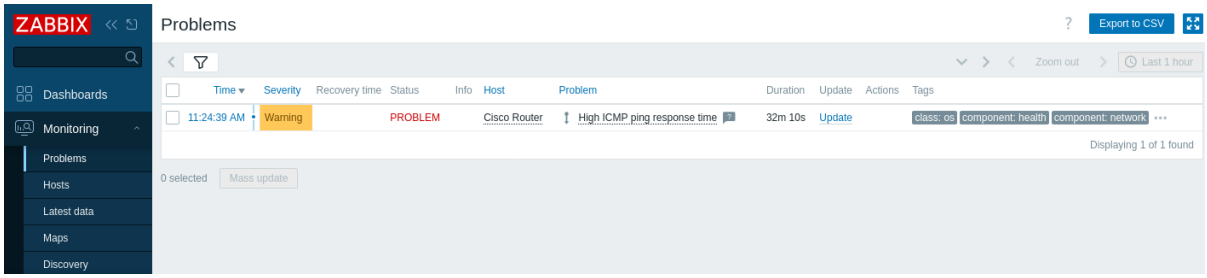


2. Siga o guia para Receber uma notificação de problema.

Da próxima vez que o Zabbix detectar um problema, você deverá receber um alerta por e-mail.

Teste sua configuração Para garantir que o Zabbix detecte corretamente problemas de desempenho de rede, simule um problema real aumentando o limite de tempo de resposta do ping ICMP.

1. Abra a configuração do seu host "Cisco Router" no Zabbix.
2. Navegue até a guia *Macros* e selecione *Macros herdadas e do host*.
3. Localize a macro `{$ICMP_RESPONSE_TIME_WARN}` (ou uma macro semelhante de limite de tempo de resposta).
4. Defina um valor muito baixo (por exemplo, 0.001) para acionar um alerta quando a resposta do ping exceder esse valor.
5. Clique em *Atualizar* para aplicar as alterações.
6. Aguarde alguns instantes para que o Zabbix detecte o problema simulado.
7. Navegue até *Monitoramento > Problemas* para verificar se um alerta aparece (por exemplo, "Tempo de resposta do ping ICMP alto").



Se os alertas estiverem configurados, você também deverá receber um alerta de problema.

8. Retorne o valor da macro para sua configuração original e clique em *Atualizar* para salvar as alterações.
9. Confirme que o problema foi resolvido e desapareceu da seção *Problemas*.

Solução de problemas de monitoramento SNMP Se você notar que o ícone SNMP no frontend do Zabbix aparece VERMELHO ou nenhum dado é coletado, tente as seguintes etapas:

1. Verifique a conectividade SNMP.

Para SNMPv2, execute o seguinte comando a partir do seu servidor Zabbix:

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

Este comando verifica se o dispositivo responde a consultas SNMP.

Para SNMPv3, inclua as credenciais SNMPv3 apropriadas:

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

Isso verifica se as credenciais SNMPv3 estão corretas e se o dispositivo está respondendo de forma segura.

2. Certifique-se de que os arquivos MIB estejam instalados e habilitados conforme descrito nos pré-requisitos. Para garantir isso, o seguinte comando não deve apresentar erro ao consultar um dispositivo de rede:

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

Isso deve retornar OIDs traduzidos sem erros.

3. Confirme se a versão SNMP e as credenciais configuradas no Zabbix correspondem às definidas em seu dispositivo. Por exemplo, revise as configurações SNMP na configuração do host do Zabbix e verifique-as em relação à configuração do seu dispositivo. Em um dispositivo Cisco, você pode verificar as configurações SNMP executando:

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

Isso garante que a string de comunidade (para SNMPv2) ou os detalhes do usuário SNMPv3 estejam corretos.

4. Verifique se o SNMP está corretamente habilitado em seu dispositivo de rede. Em um roteador Cisco, faça login no console e execute:

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

Este comando exibe a configuração SNMP ativa e ajuda a confirmar se o SNMP está configurado corretamente.

5. Certifique-se de que nenhum firewall ou problema de rede esteja bloqueando o tráfego SNMP (normalmente na porta 161) entre o servidor Zabbix e o dispositivo. Você pode testar a conectividade usando:

nc -zv <device_IP> 161

`nc -zv` verifica se a porta 161 está aberta e ouvindo no dispositivo.

Além disso, se você estiver usando UFW no Ubuntu, verifique o status do firewall:

```
sudo ufw status
```

Ou, para iptables:

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

6. Revise os arquivos de log do servidor Zabbix para quaisquer erros relacionados ao SNMP para ajudar a identificar o problema:

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

`tail -f` permite monitorar as atualizações do log em tempo real.

Veja também:

- Criando um item - saiba como adicionar métricas adicionais.
- SNMP agent - informações adicionais sobre o monitoramento SNMP com o Zabbix.
- Templates padronizados para dispositivos de rede - informações sobre os templates SNMP disponíveis.
- Descoberta de OIDs SNMP - informações adicionais sobre a descoberta SNMP em um switch.
- Configurando uma regra de descoberta de rede - informações adicionais sobre como configurar uma regra de descoberta de rede usada pelo Zabbix para descobrir hosts e serviços.

11 Monitorar o log de eventos do Windows usando verificações ativas

Introdução

Este guia explica como monitorar logs de eventos do Windows com o Zabbix usando verificações ativas. Com as chaves de item específicas do Windows do Zabbix, você pode coletar e analisar eventos críticos (como tentativas de login falhas, erros do sistema, etc.) em tempo real.

Para quem é este guia

Este guia é destinado a novos usuários do Zabbix e administradores de rede que desejam monitorar logs de eventos do Windows. Para opções avançadas de configuração, consulte a documentação de chaves de item específicas do Windows.

Pré-requisitos

Antes de prosseguir com este guia, você precisa baixar e instalar o Zabbix server e o Zabbix frontend de acordo com as instruções para o seu sistema operacional. Você também precisa do Zabbix agent baixado e instalado na máquina Windows que deseja monitorar.

Configure o agent Zabbix para monitoramento do log de eventos do Windows

1. Abra o arquivo `zabbix_agentd.conf` (caminho padrão `C:\Program Files\Zabbix Agent\zabbix_agentd.conf`) no seu host Windows e certifique-se de que o parâmetro ServerActive está definido para o endereço IP do seu server Zabbix, e que o parâmetro Hostname corresponde ao nome do host que será definido no frontend Zabbix. Isso permite que o agent solicite verificações ativas para seu host e a partir do server Zabbix especificado. Por exemplo:

```
ServerActive=192.0.2.0
Hostname=MyWindowsHost
```

2. Reinicie o serviço do agent Zabbix para aplicar as alterações:

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

3. Verifique se o host Windows está em execução:

- Certifique-se de que o serviço do agent Zabbix está em execução no host Windows.
- Verifique se o host Windows pode se conectar ao server Zabbix na porta 10051. Para testar a conectividade a partir do host Windows, abra o PowerShell e execute o seguinte comando:

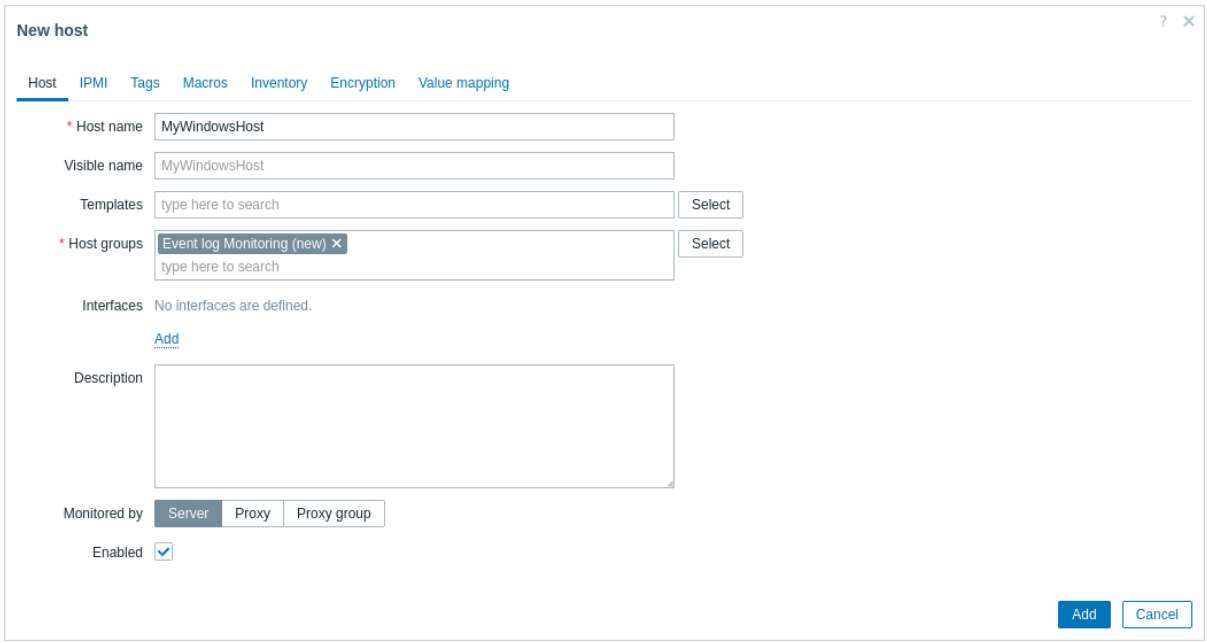
```
Test-NetConnection -ComputerName <Zabbix-server-IP> -Port 10051
```

Configure o frontend do Zabbix

1. Navegue até *Coleta de dados* > *Hosts* e crie um host:

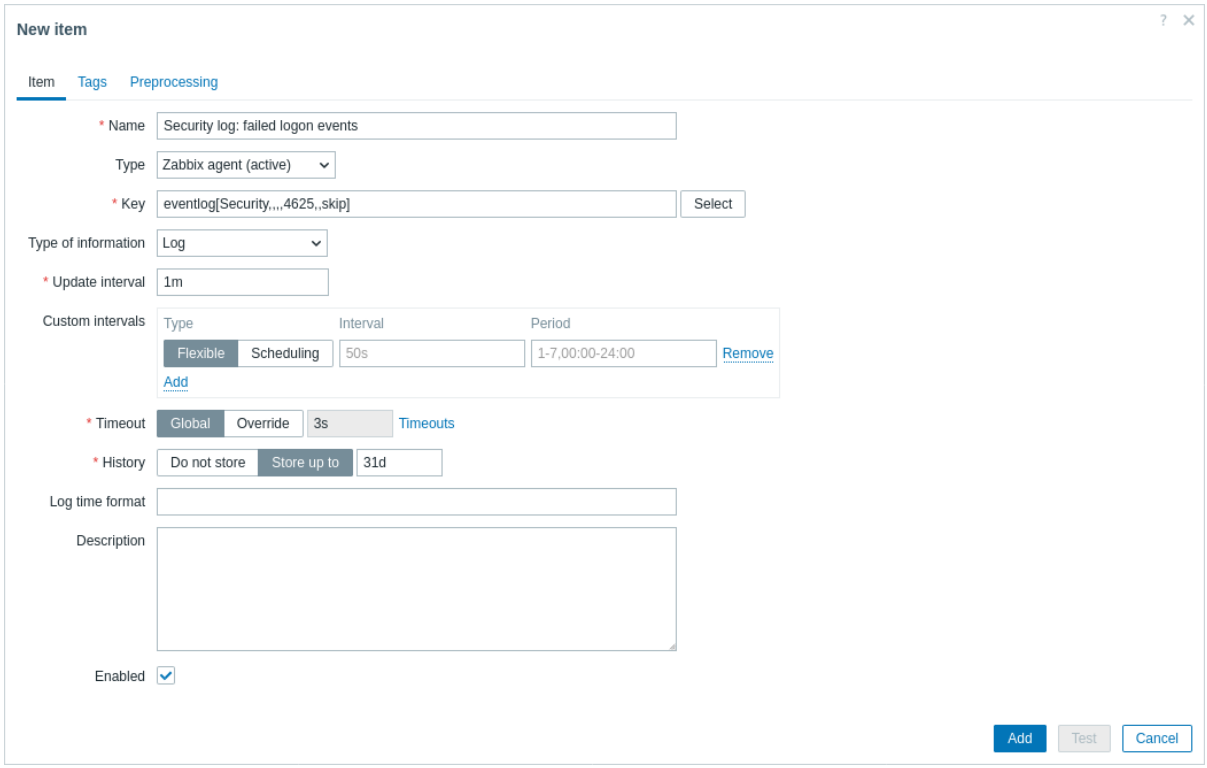
- No campo *Nome do host*, insira um nome de host (por exemplo, "MeuHostWindows").
- No campo *Grupos de hosts*, digite ou selecione um grupo de hosts (por exemplo, "Monitoramento de log de eventos").
- Pressione *Adicionar* para salvar o host configurado.

Note:
No campo *Templates* você pode adicionar o template "Windows by Zabbix agent active" para ajudar na solução de problemas, observando se outros itens ativos no mesmo host estão sendo atualizados.



2. Crie um novo item com os seguintes parâmetros:

- No campo *Nome*, insira um nome descritivo para o item (por exemplo, ”Log de segurança: eventos de falha de logon”).
- No menu suspenso *Tipo*, selecione ”Zabbix agent (active)” (obrigatório para monitoramento de log de eventos).
- No campo *Chave*, utilize a chave de item eventlog. Por exemplo, para monitorar tentativas de logon com falha (ID do evento: 4625) no log de Segurança e ignorar entradas mais antigas que a última verificação do item (usando o parâmetro `skip`), insira a seguinte chave de item: `eventlog[Security,,,,4625,,skip]`
- No menu suspenso *Tipo de informação*, selecione ”Log”.



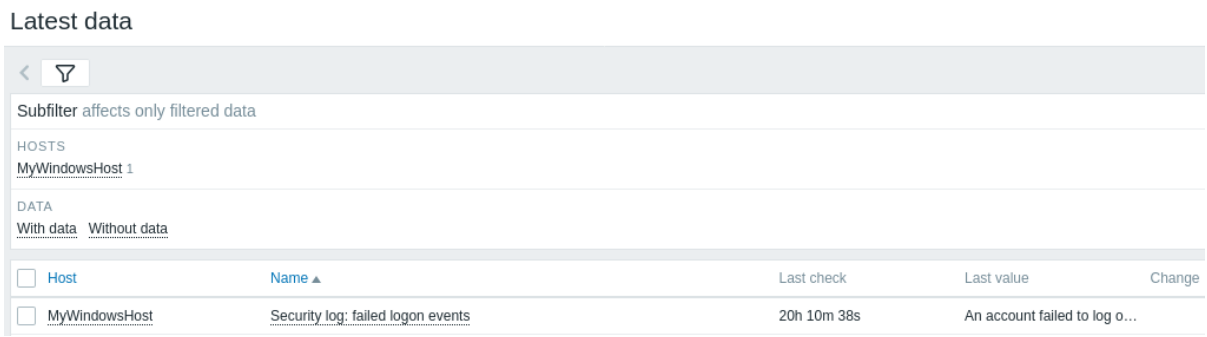
3. Clique em *Adicionar* para salvar o item.

Teste e visualize as métricas coletadas

Parabéns! O Zabbix agora está configurado para coletar os logs de eventos do Windows. Para verificar se os logs de eventos estão sendo coletados, você pode testar o item ”Security log: failed logon events” saindo da sua conta do Windows e tentando fazer login usando credenciais incorretas.

Em seguida, visualize os logs coletados no frontend do Zabbix:

1. Navegue até *Monitoramento* > *Últimos dados* no frontend do Zabbix.



2. Filtre pelo seu host ”MyWindowsHost” no campo *Nome*.

3. Clique em *Histórico* para visualizar os valores de log registrados.

MyWindowsHost: Security log: failed logon events

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

4. Se os valores de log estiverem ausentes, prossiga para a seção Solução de problemas deste guia.

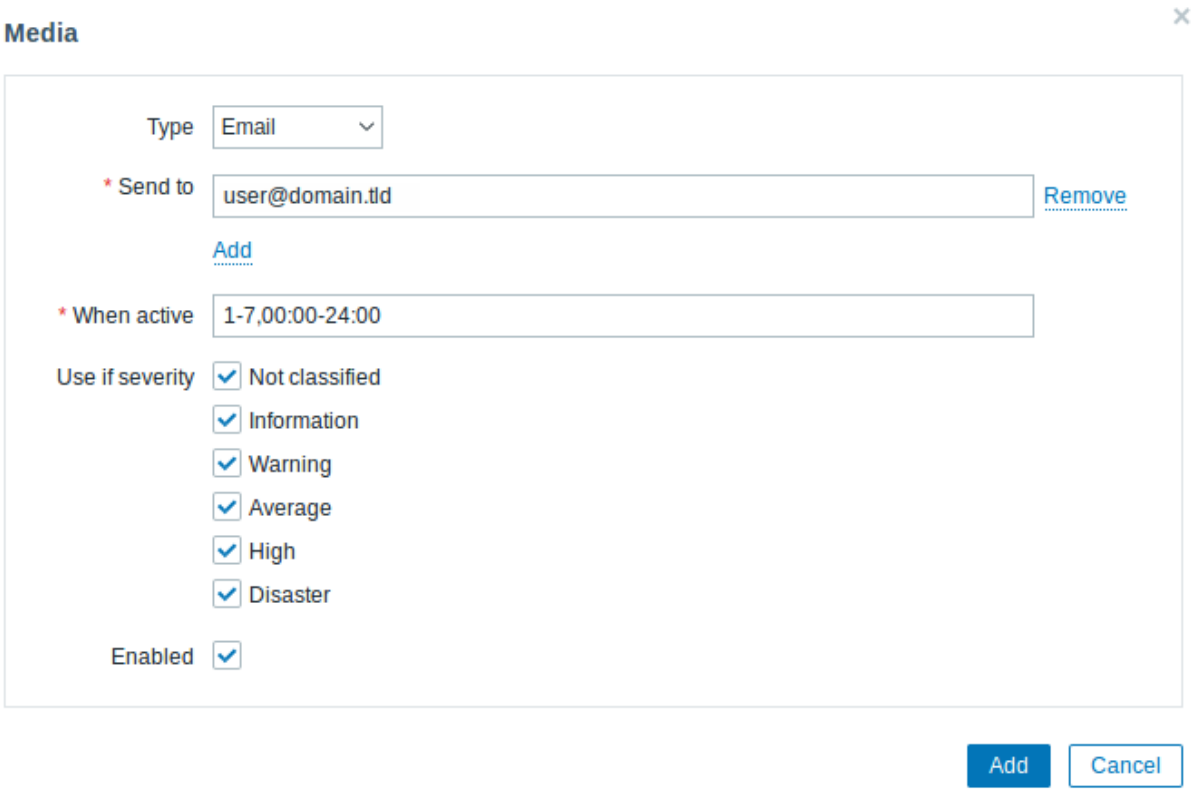
Configurar alertas de problema

Este guia fornece etapas básicas de configuração para o envio de alertas por e-mail.

1. Navegue até Coleta de dados > Hosts para definir um trigger que será acionado quando o item de log de eventos registrar o padrão desejado. Por exemplo, para capturar tentativas de login com falha no log de Segurança, use a função find():

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

2. Navegue até *Configurações do usuário > Perfil*, alterne para a guia *Mídia* e adicione seu e-mail.



3. Siga o guia para Receber uma notificação de problema.

Da próxima vez que o Zabbix detectar um problema, você deverá receber um alerta por e-mail.

Solução de problemas

Se você encontrar problemas ao coletar ou visualizar logs de eventos do Windows, use as dicas abaixo para identificar e resolver problemas comuns:

1. No Zabbix server (Linux), liste suas regras do iptables com o seguinte comando:

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

e verifique se há uma regra ACCEPT para a porta TCP 10051.

2. Certifique-se de que sua chave `eventlog[...]` usa exatamente o nome do log (diferencia maiúsculas de minúsculas), ID do evento, modo (por exemplo, skip) e outros parâmetros exatamente como mostrado nas chaves de item específicas do Windows.

Veja também:

- Criando um item - saiba como adicionar métricas adicionais.
- Agent Zabbix no Microsoft Windows - instruções detalhadas de instalação.
- Monitorando Windows com agent Zabbix - um guia abrangente sobre como configurar o monitoramento básico para máquinas Windows usando o agent Zabbix.
- Chaves de item específicas do Windows - informações detalhadas sobre as chaves de item específicas do Windows suportadas pelos agents Zabbix, incluindo aquelas para monitoramento de logs de eventos.
- Monitoramento de arquivos de log - instruções sobre como configurar o Zabbix para monitoramento centralizado e análise de arquivos de log, aplicável aos logs de eventos do Windows.

Zabbix Cloud

Zabbix Cloud

Descubra as opções de configuração disponíveis e os controles de acesso para organizações e nós no Zabbix Cloud.

Configurar e gerenciar o Zabbix Cloud

Comece a usar o Zabbix Cloud Instruções guiadas para colocar sua instância Zabbix Cloud em funcionamento rapidamente, incluindo a configuração inicial e a conexão dos dispositivos monitorados.

Configuração do nó Aprenda como configurar seu nó Zabbix Cloud, com foco em acesso, manipulação de dados e configurações de recursos.

Adicionando usuários Descubra como adicionar e atribuir funções a usuários dentro de uma organização Zabbix Cloud e seus nós.

Saiba mais sobre o Zabbix Cloud

Zabbix Cloud vs. on-premises Compare as diferenças de gerenciamento, escalabilidade e manutenção entre as duas implantações e descubra qual delas pode suportar seu fluxo de trabalho de forma mais eficiente.

Explore o Zabbix Cloud Obtenha uma visão geral do Zabbix Cloud, incluindo seus principais recursos, vantagens e o que está incluído na assinatura. Para perguntas comuns sobre configuração e uso, consulte o FAQ.

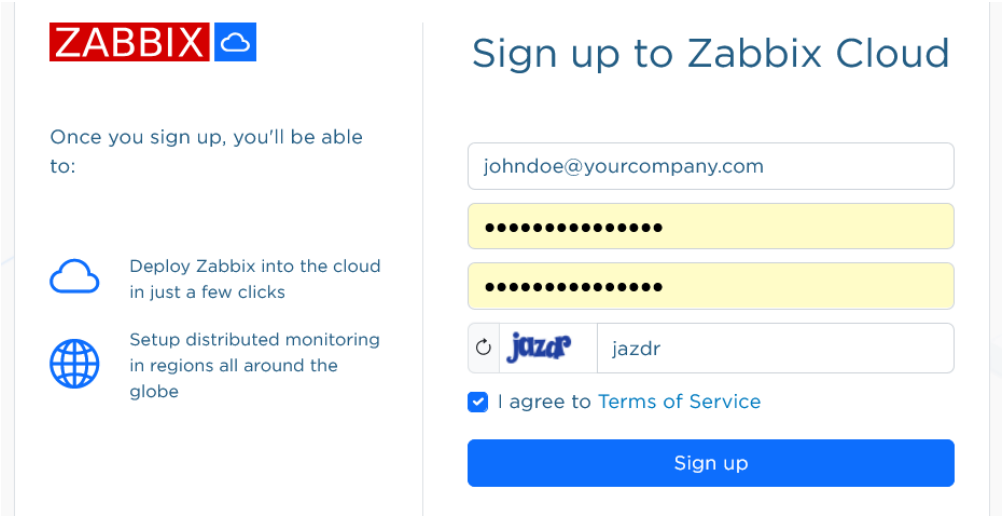
1 Implante o Zabbix na nuvem

Para iniciar o Zabbix na nuvem, siga este processo:

- Faça login ou cadastre-se em cloud.zabbix.com
- Crie um nó na nuvem
- Comece a usar o Zabbix na nuvem

Faça login ou cadastre-se Você pode fazer login em cloud.zabbix.com com uma conta existente do Google, GitHub ou Microsoft.

Como alternativa, acesse a página de cadastro para se registrar.



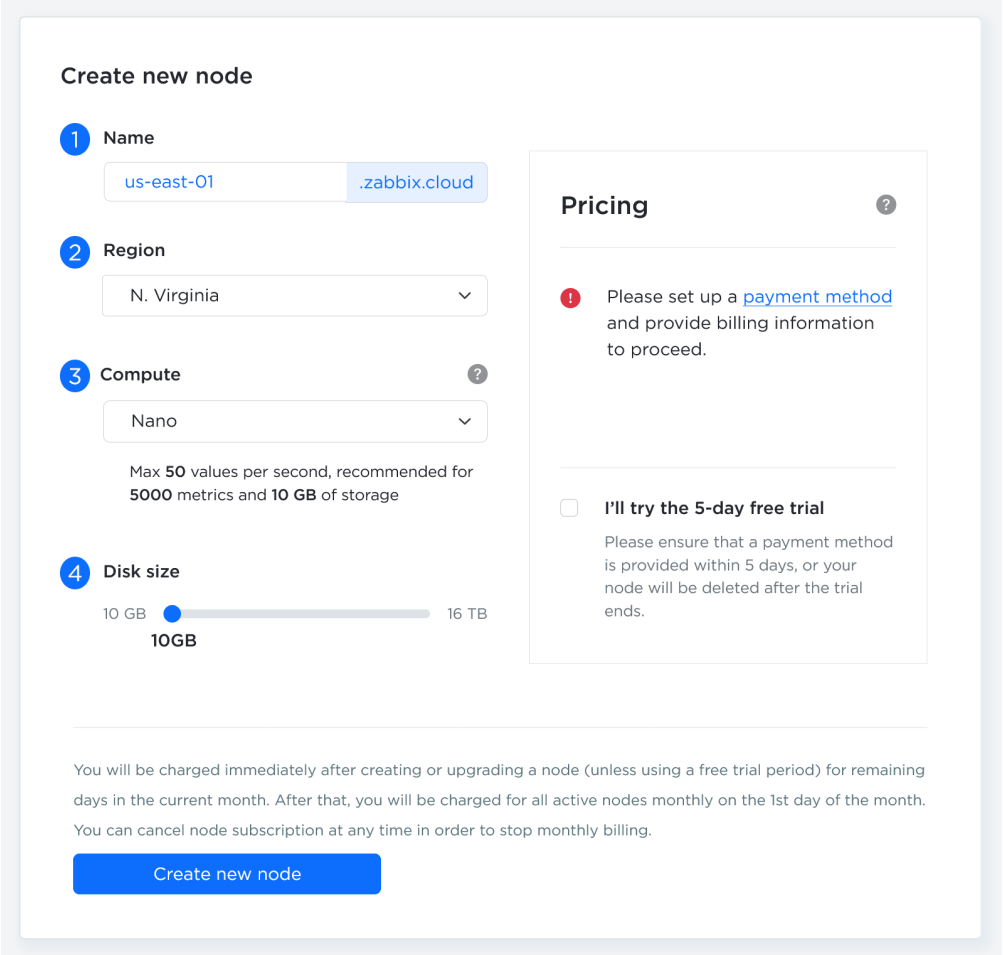
Após o cadastro, você receberá uma confirmação da sua nova conta por e-mail.

Acesse cloud.zabbix.com para fazer login. Você será solicitado a informar uma senha de uso único ao fazer login. Essa senha será enviada para você por e-mail.

Criar nó na nuvem Um nó na nuvem é uma instância do Zabbix, baseada na nuvem. Inclui o Zabbix server, o banco de dados e o frontend web.

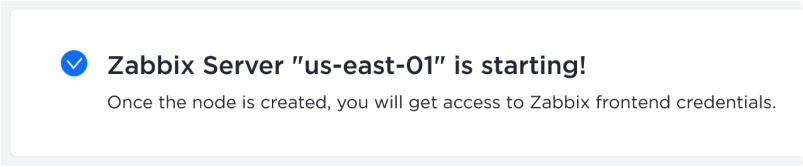
Ao fazer login, você verá que uma organização *Privada* já existe por padrão para sua conta. Você pode renomear a organização padrão posteriormente.

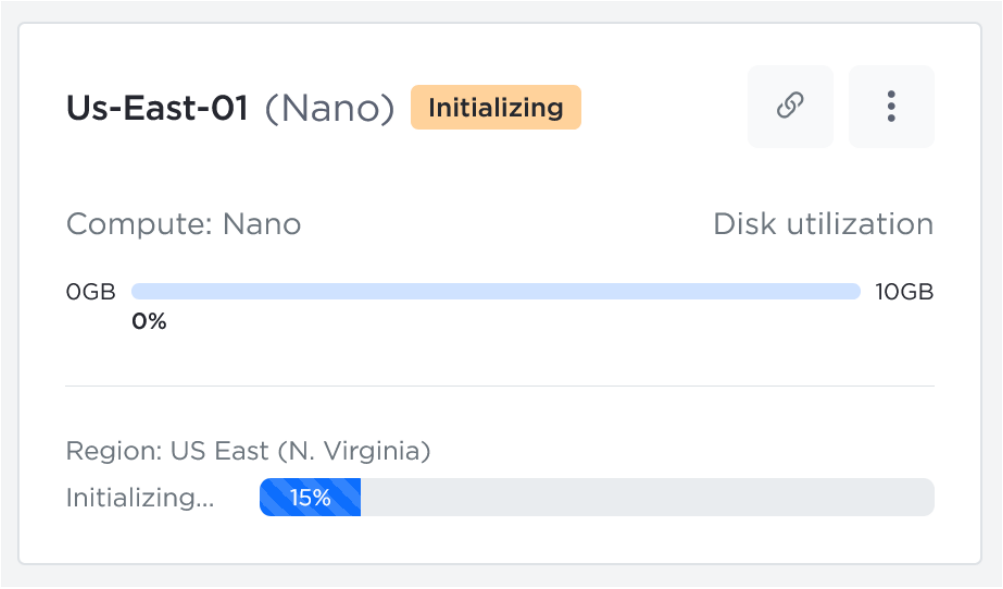
Clique em *Criar Novo Nó*. Um formulário de configuração do nó será aberto.

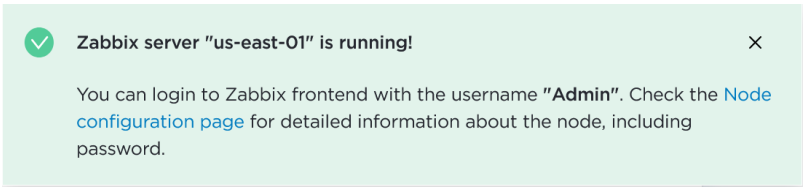


Parâmetro	Descrição
Nome	Selecione um nome de nó exclusivo. O nome do nó forma uma URL exclusiva para o server/frontend na nuvem, no formato `<seu-nome>.zabbix.cloud`.
Região	Selecione a localização do data center mais próxima para obter a menor latência de rede.
Computação	Selecione o poder de computação para o nó. Para determinar qual nível de recursos você precisa, consulte os detalhes de preços. Você sempre pode fazer upgrade depois, se precisar de mais poder. Um método de cobrança será necessário antes de criar um nó que não seja de avaliação. Adicione as informações de cobrança pelo menu *Cobrança*. Está disponível um **teste gratuito** de 5 dias para experimentar um nó do tipo *Nano*. Você não será cobrado durante o período de avaliação, mas, assim que ele terminar, seu nó será removido se não for convertido em um nó pago.
Tamanho do disco	O controle deslizante é definido por padrão para um mínimo recomendado com base no tipo. Aumente se você armazenar grandes volumes de histórico ou tendências. Esteja ciente de que discos maiores geram custos mensais mais altos e que você não pode reduzir um disco por conta própria—será necessário entrar em contato com o suporte para reduzi-lo.

Após preencher o formulário, clique em *Criar Novo Nó*. Enquanto o nó está sendo inicializado, uma mensagem correspondente é exibida e o nó já aparece em Organizações.





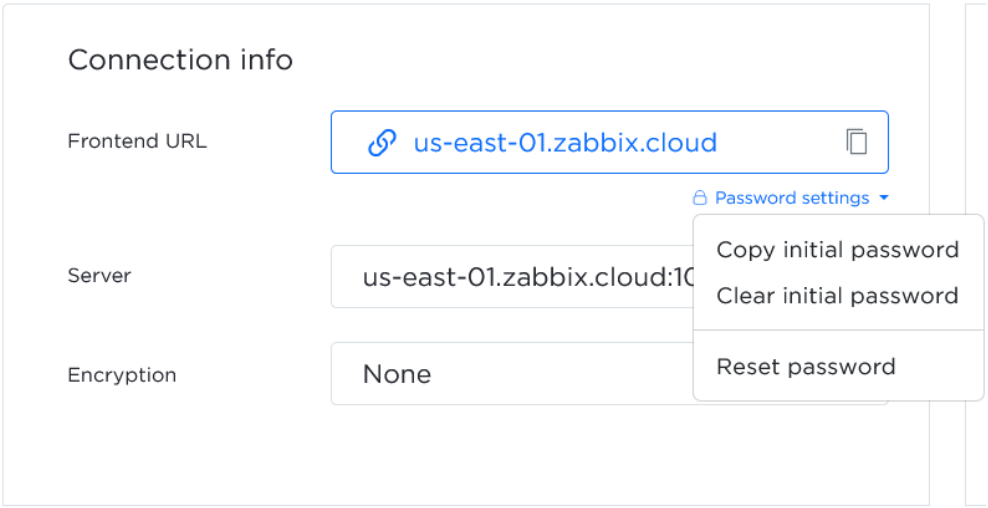


Comece a usar o Zabbix

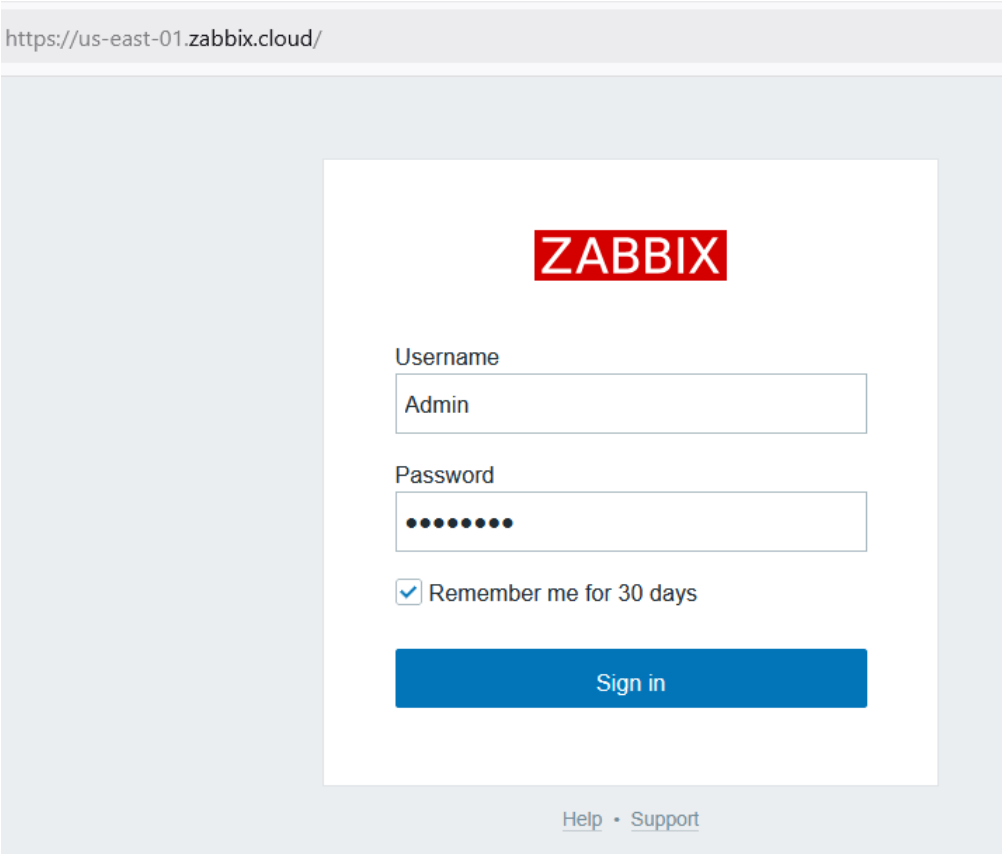
Quando o status do nó mudar de *Inicializando* para *Em execução*, você poderá fazer login no frontend do Zabbix.

Para acessar a senha inicial do frontend, clique no nó criado.

Você será direcionado para as configurações do nó (visão geral, filtros de acesso, criptografia, backups, etc.). Em **Informações de conexão**, clique em *Configurações de senha* e selecione *Copiar senha inicial* no menu suspenso.



Clique na URL do frontend para abrir o frontend do Zabbix (`us-east-01.zabbix.cloud`).



Faça login usando as seguintes credenciais:

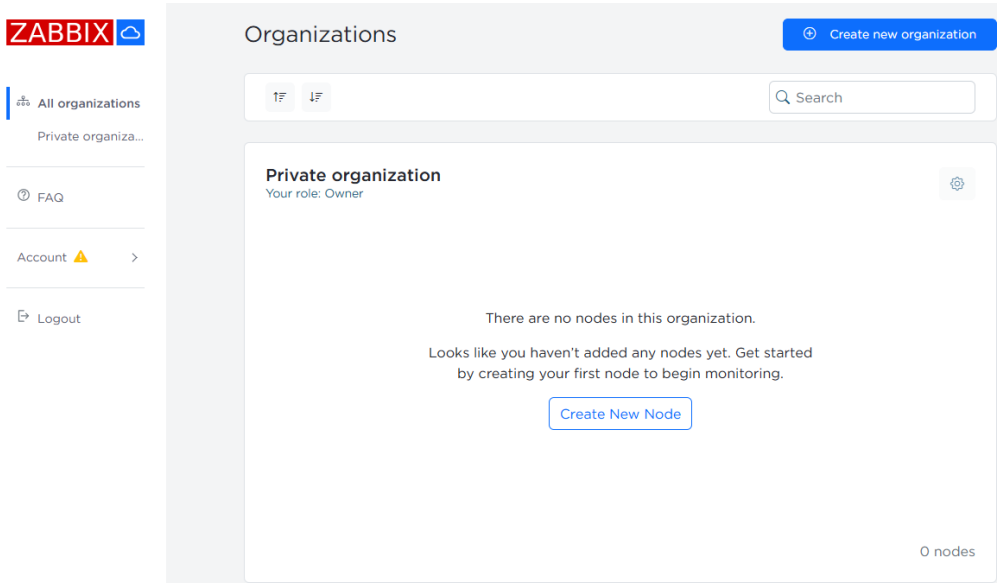
- *Nome de usuário* - `Admin`
- *Senha* - forneça a senha inicial (por exemplo, HzTG9t7Y)

Para começar a monitorar, você precisa definir filtros de acesso - endereço ou endereços IP que têm permissão para se conectar ao Zabbix cloud server. Pelo menos um deve ser definido.

Gerenciamento de organização Uma organização é um contêiner para um ou mais nós. As organizações também ajudam você a estruturar seus recursos e delegar responsabilidades, como:

- Gerenciamento de nós
- Cobrança

Ao fazer login no Zabbix Cloud, a página *Todas as organizações* é exibida. Por padrão, uma organização privada é criada para sua conta.



Como proprietário desta organização, você pode gerenciar usuários, administradores, nós e o nome da organização. Clique no botão de configurações para acessar as opções de gerenciamento.

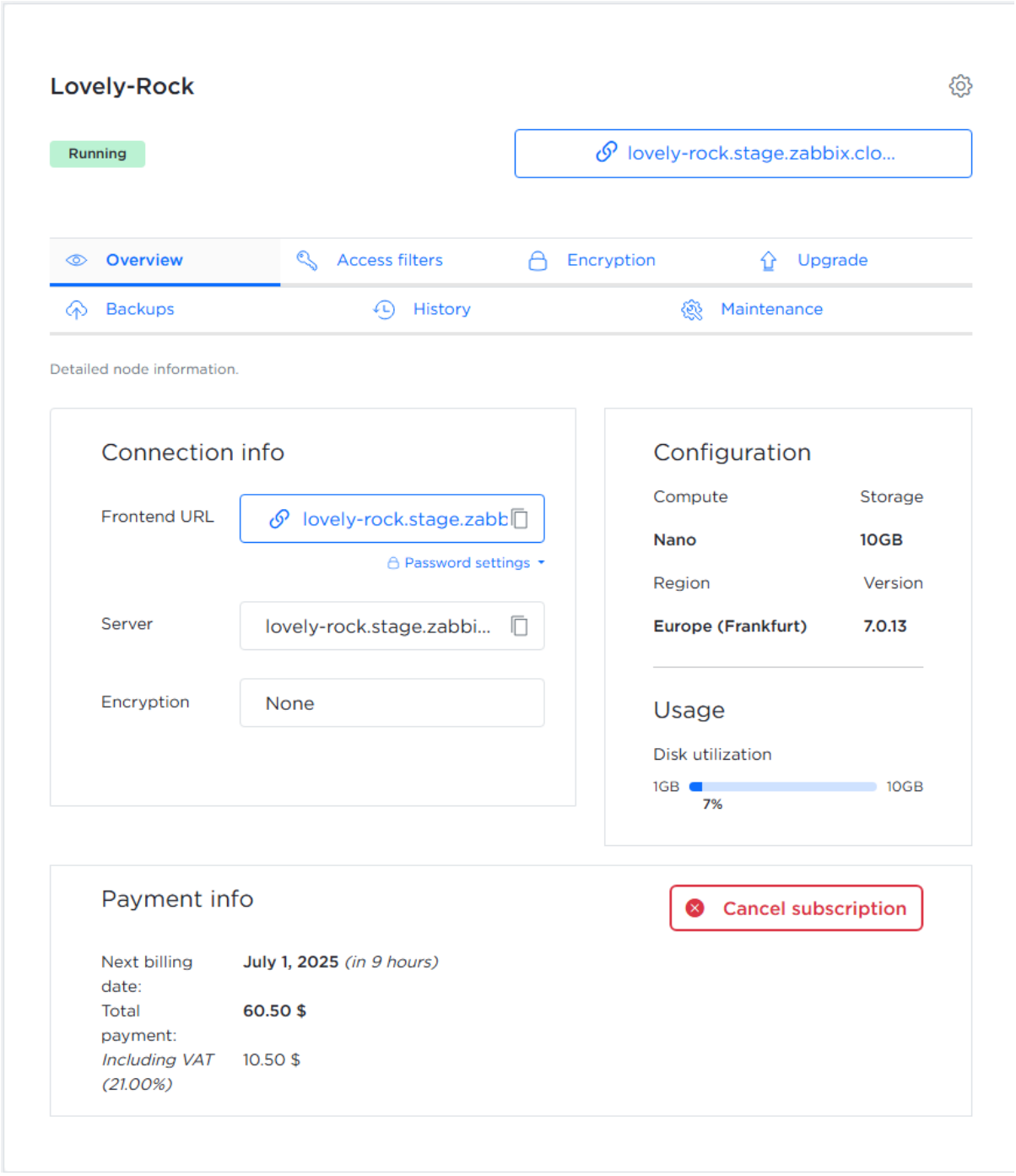
2 Configuração do nó

Visão Geral

Para acessar a página de configuração do nó, clique no nó em *Todas as organizações*, no menu da sua organização ou usando o link *Página de configuração do nó* exibido após a inicialização de um nó](/cloud/quickstart#creating-a-node).

Na visualização de configuração do nó, cada aba permite gerenciar um aspecto específico do seu nó:

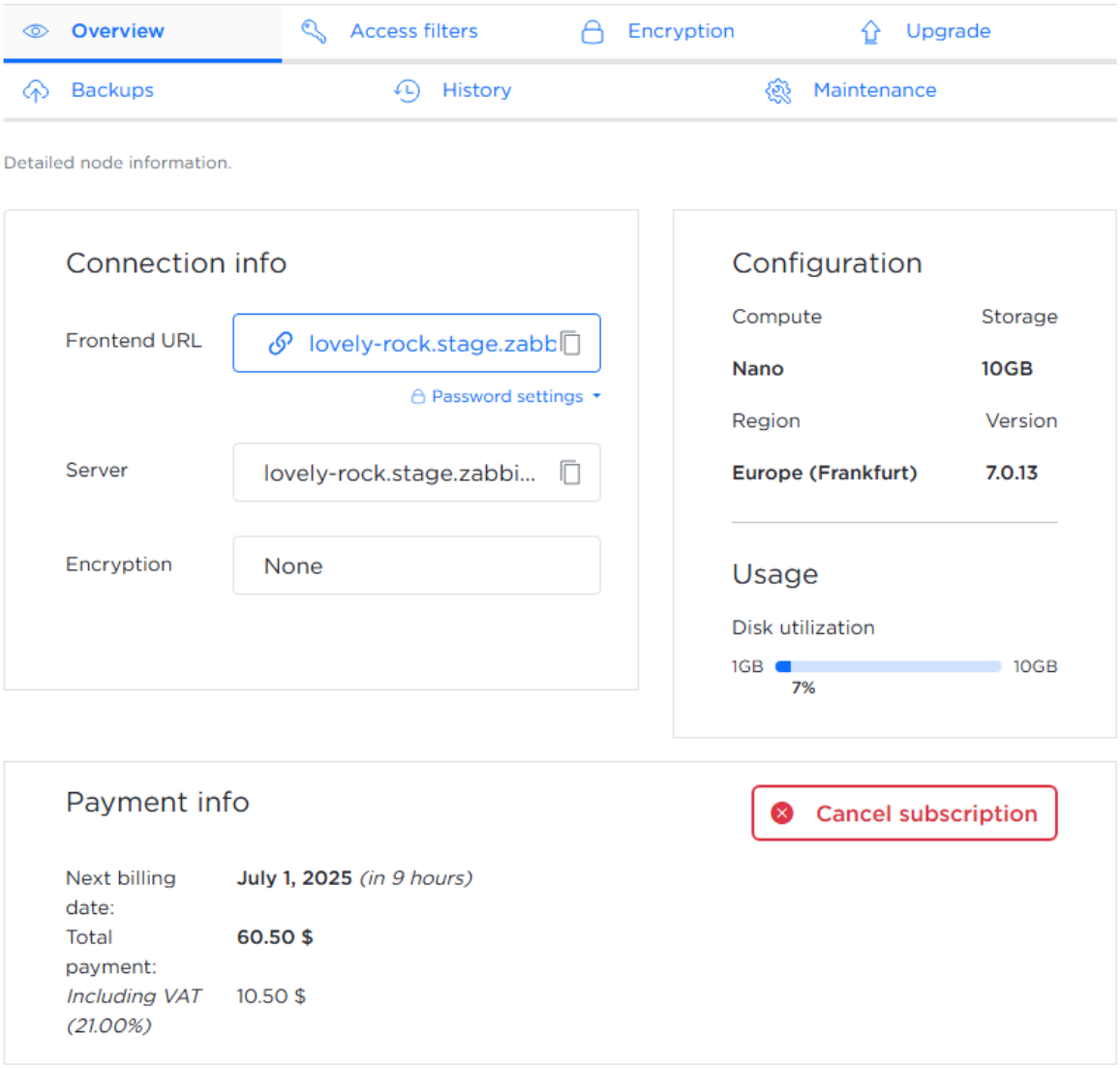
- Visão Geral - visualize os detalhes do nó, abra seu frontend e gerencie os usuários do nó.
- Filtros de acesso - controle o acesso por endereço IP ou sub-rede.
- Criptografia - configure o TLS enviando certificados, chaves e CRLs opcionais.
- Backups - crie, restaure e exclua snapshots do nó; visualize o próximo backup agendado.
- Histórico - defina períodos de retenção para logs de auditoria, histórico e tendências, com unidades predefinidas e um limite de atualização de 3x/dia.
- Manutenção - defina janelas de manutenção semanais por fuso horário, dia e hora.
- Atualização - dimensione recursos de computação/armazenamento ou gerencie seu status de teste/pagamento.



Note:
Algumas guias podem estar indisponíveis dependendo do tipo de usuário que visualiza a página e o status do nó (teste/não teste).

Aba Visão Geral

A aba *Visão Geral* fornece informações detalhadas sobre o nó.



Dividido em três seções:

Informações de conexão:

- URL clicável do frontend com opção de copiar para a área de transferência
- Configurações de senha
- Método de criptografia atual

Configuração e uso:

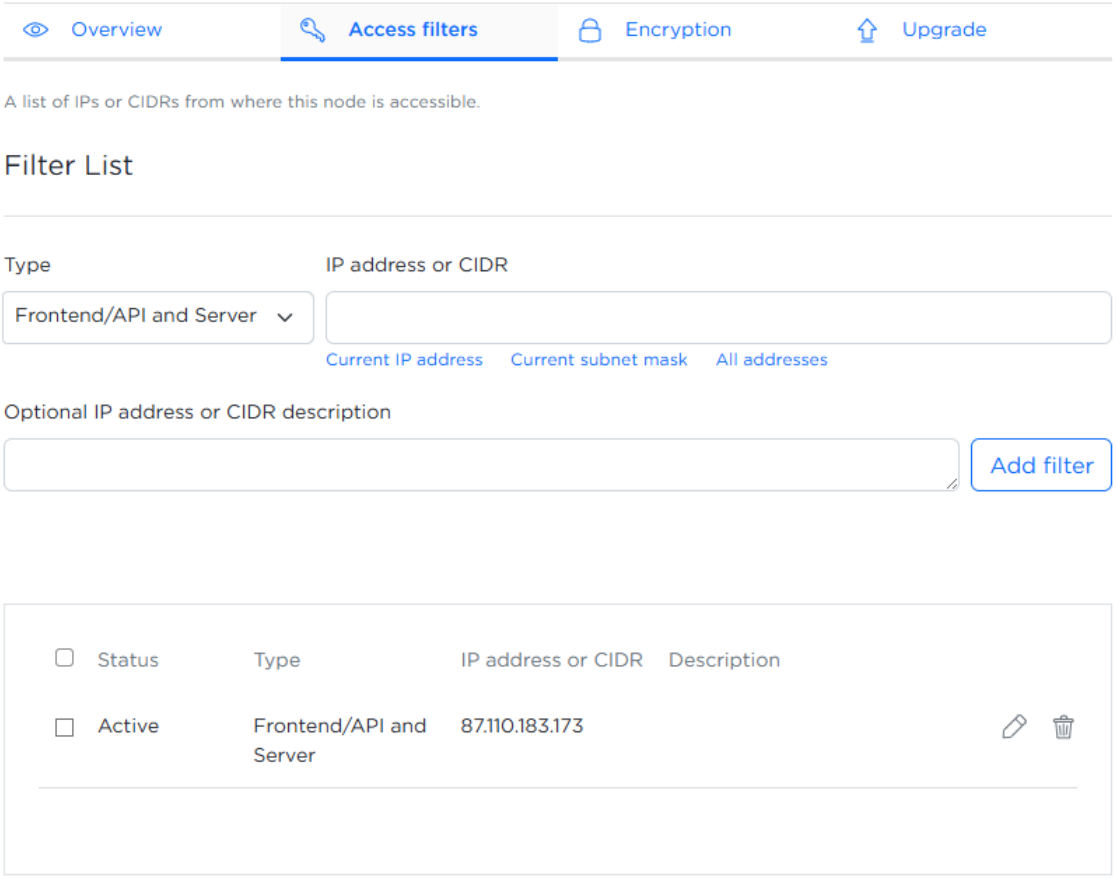
- Camada de computação, armazenamento máximo, região
- Versão dos componentes do Zabbix
- Utilização do disco

Informações do nó/Informações de pagamento:

- Data de expiração (para nós de teste e pré-pagos)
- Um botão específico de contexto é exibido dependendo do tipo de nó após a conclusão da inicialização:
- *Atualizar para pago* - disponível para nós em teste gratuito
- *Cancelar assinatura* - disponível para nós pagos com uma assinatura ativa
- *Renovar assinatura paga* - exibido se uma assinatura foi cancelada
- Nenhum botão de ação é exibido para nós pré-pagos, que têm validade longa e não podem ser cancelados

Aba Filtros de acesso

A aba *Filtros de acesso* permite restringir o acesso ao seu nó Zabbix por endereço IP ou sub-rede.



Para configurar um filtro de acesso, preencha os seguintes campos:

- *Tipo* - use o menu suspenso para escolher qual componente Zabbix limitar: somente frontend/API, somente servidor ou ambos.
- *Endereço IP ou CIDR* - insira o endereço IP ou CIDR manualmente ou preencha o campo automaticamente selecionando a opção relevante.
- *Endereço IP ou descrição do CIDR opcional* - adicione uma descrição opcional para cada IP ou sub-rede.

Após preencher os campos, clique em *Adicionar filtro*. Os filtros adicionados aparecem na tabela abaixo, mostrando status, tipo, endereço IP ou sub-rede e descrição.

Warning:
Durante a inicialização do nó, um filtro de acesso é criado automaticamente para o endereço IP que você utilizou. Se você se conectar posteriormente de um IP ou rede diferente, deverá adicionar esse endereço aqui ou será bloqueado.

Você pode:

- - editar um filtro.
- - excluir um ou mais filtros usando as caixas de seleção.

Aba Criptografia

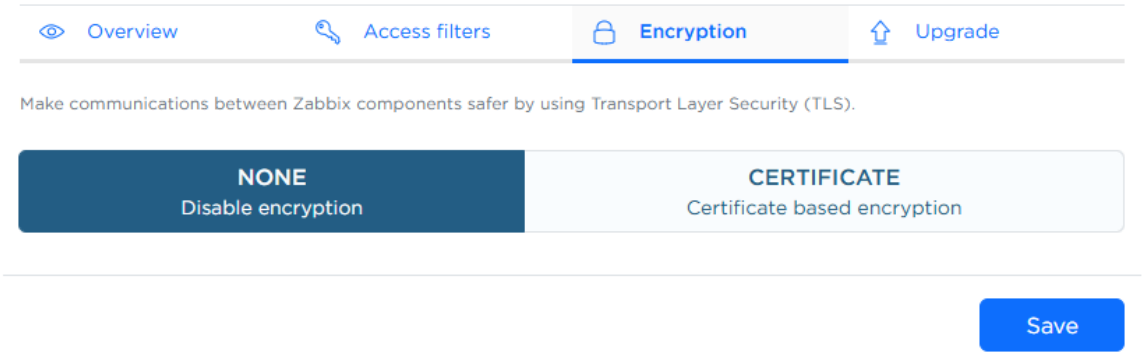
A aba *Criptografia* permite configurar a criptografia TLS [/manual/encryption] entre os componentes do Zabbix, enviando os arquivos de certificado necessários.

Para habilitar a comunicação segura usando certificados, selecione *CERTIFICADO* e envie os seguintes arquivos do seu computador:

- *Certificados da CA raiz* - usados para verificar a autenticidade dos certificados pares. Clique em *Escolher arquivo* e selecione o arquivo que contém os certificados da CA raiz confiáveis (normalmente um arquivo `.crt` ou `.pem`, codificado em PEM).
- *Cadeia de certificados* - representa a cadeia de confiança do seu próprio certificado. Clique em *Escolher arquivo* e selecione o arquivo codificado em PEM (por exemplo, `.crt` ou `.pem`) que inclui seu certificado e quaisquer certificados de CA intermediários.
- *Chave privada* - a chave privada correspondente ao seu certificado. Clique em *Escolher arquivo* e selecione o arquivo `.key` codificado em PEM.

- *Certificados revogados* (opcional) - se você mantém uma Lista de Revogação de Certificados (CRL), clique em *Escolher arquivo* e selecione o arquivo CRL para especificar os certificados que não são mais confiáveis.

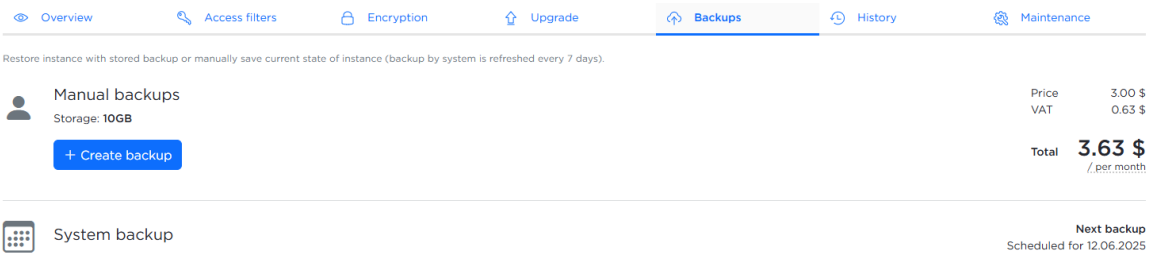
Após selecionar os arquivos, clique em *Salvar*. Os arquivos serão carregados, validados e aplicados para configurar a criptografia TLS entre os componentes do Zabbix.



Aba Backups

A aba *Backups* permite visualizar, criar, restaurar e excluir backups do seu nó Zabbix.

Você pode restaurar sua instância a partir de qualquer backup salvo ou criar um novo backup manual a qualquer momento. A aba também mostra a data do próximo backup agendado.



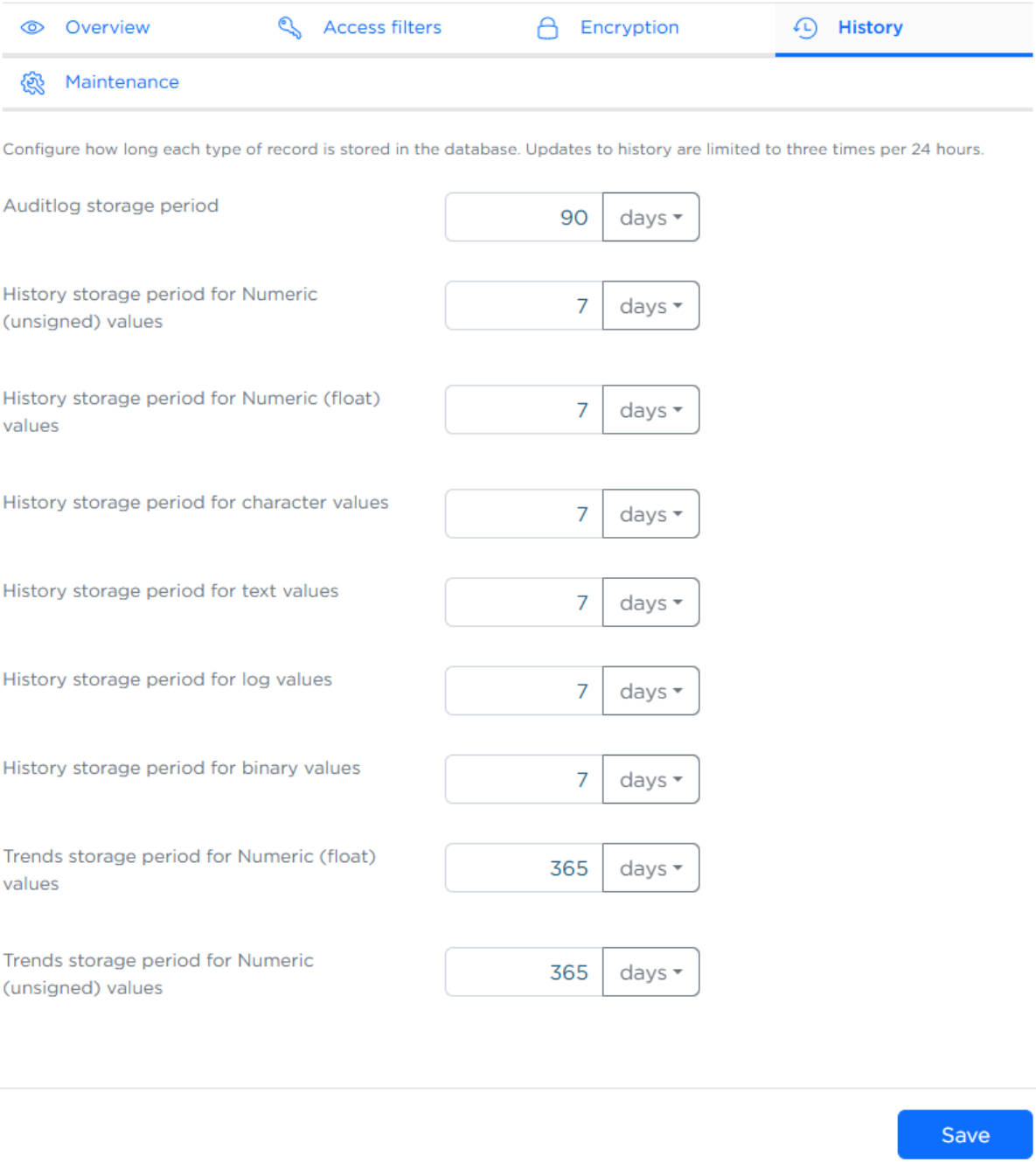
Para criar um backup:

1. Clique no botão *Criar backup*.
2. Insira uma descrição opcional na janela pop-up.
3. Clique em *Criar* para confirmar.

Warning:
Backups adicionais são cobrados mensalmente com base no uso do armazenamento.

Aba Histórico

A aba *Histórico* permite configurar por quanto tempo diferentes tipos de dados históricos são armazenados.



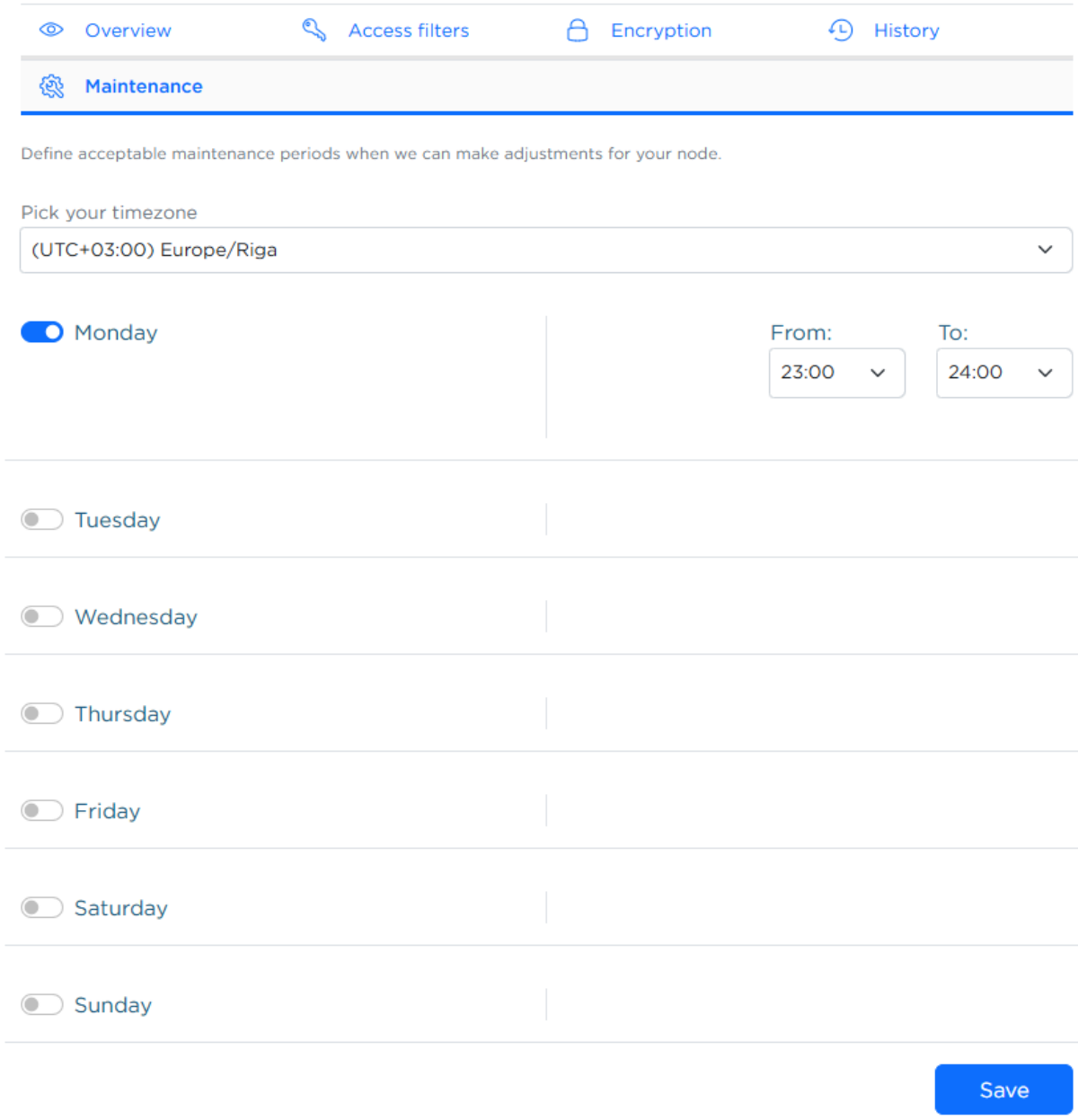
Você pode definir o período de armazenamento para cada tipo de dado (por exemplo, numérico, texto, logs) e escolher a unidade (dias, meses, anos). As alterações podem ser feitas até três vezes a cada 24 horas.

Após atualizar os valores, clique em *Salvar* para aplicar as alterações.

Warning:
Períodos de retenção mais longos podem aumentar o uso do disco e afetar o desempenho. Recomenda-se configurar os períodos de retenção com base nas suas necessidades reais de dados e análise.
 O Zabbix Cloud usa particionamento de banco de dados para gerenciar o armazenamento de dados em vez da manutenção tradicional do Zabbix. Isso permite um desempenho mais rápido e um tratamento de retenção mais previsível.

Aba Manutenção

A aba *Manutenção* permite definir janelas de manutenção aceitáveis quando atualizações ou ajustes podem ser realizados no seu nó.

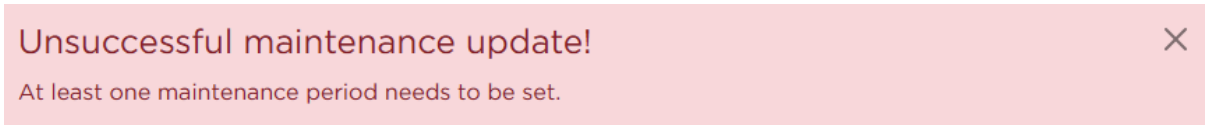


Você deve:

- Selecionar seu fuso horário no menu suspenso.
- Escolher pelo menos uma janela de dia e uma de hora para manutenção.

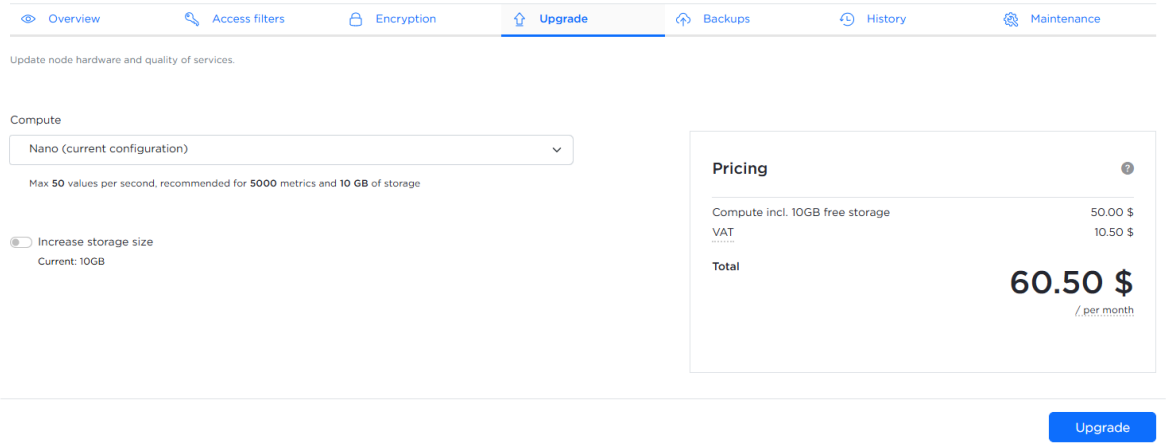
Clique em *Salvar* para aplicar as alterações.

Se nenhuma janela de dia ou hora for selecionada, um erro será exibido.



Aba de Atualização

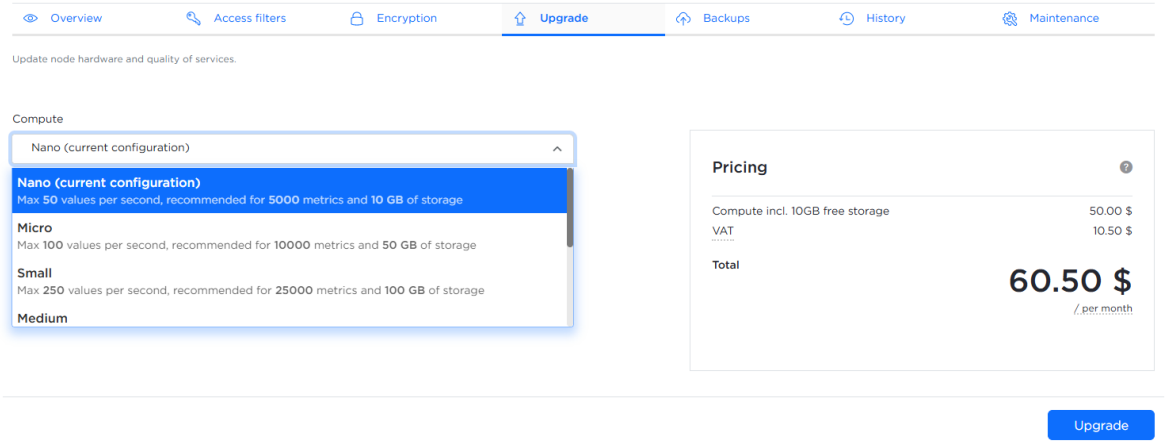
A aba *Atualização* permite dimensionar os recursos do seu nó ou gerenciar sua assinatura.



Você pode aumentar a camada de computação ou o tamanho do disco — observe que o tamanho do disco só pode ser aumentado uma vez a cada 24 horas e não pode ser reduzido pela interface (entre em contato com o suporte do Zabbix para solicitações de redução). Nós de teste gratuito podem ser convertidos em assinaturas pagas.

Para aumentar o tamanho do disco computacional:

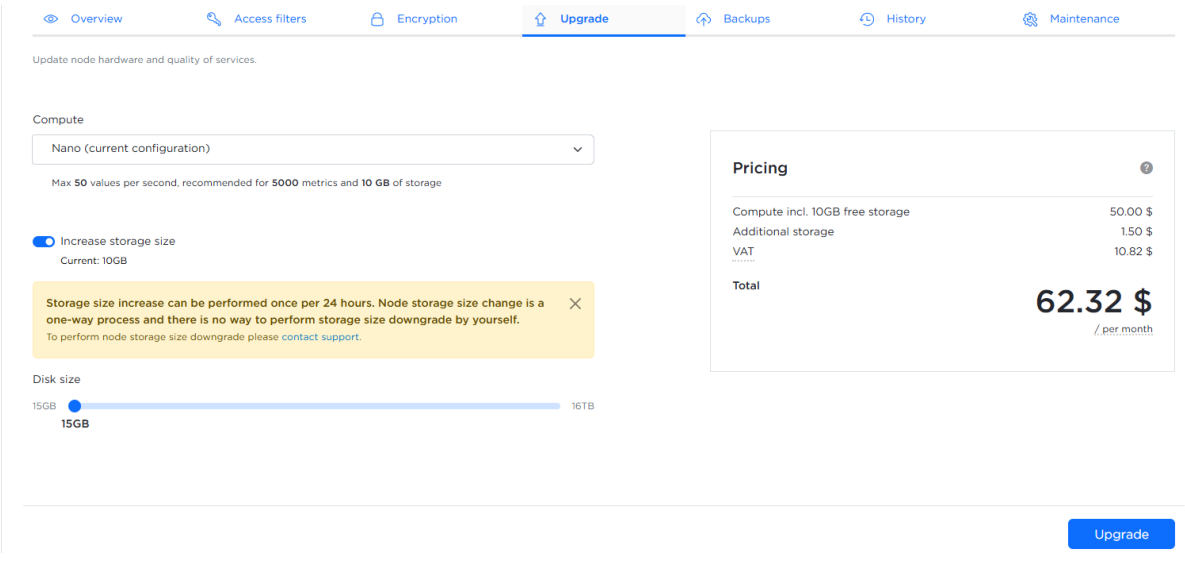
1. Clique no menu suspenso *Calculação*.



2. Selecione o tamanho do disco de configuração desejado.
3. Clique no botão *Atualização*.

Para aumentar o tamanho do disco:

1. Clique no botão de opção *Aumentar tamanho do armazenamento*.
2. Ajuste o controle deslizante *Tamanho do disco* para o valor desejado.



3. Clique no botão *Atualizar*.

Note:
O preço mensal total atualizado é exibido na tabela *Preços* antes da confirmação das alterações.

3 Adicionando usuários

Introdução

Os exemplos a seguir descrevem como adicionar usuários à sua organização e nós.

Note:
Copiar, limpar e redefinir a senha do frontend só pode ser realizado pelo proprietário.

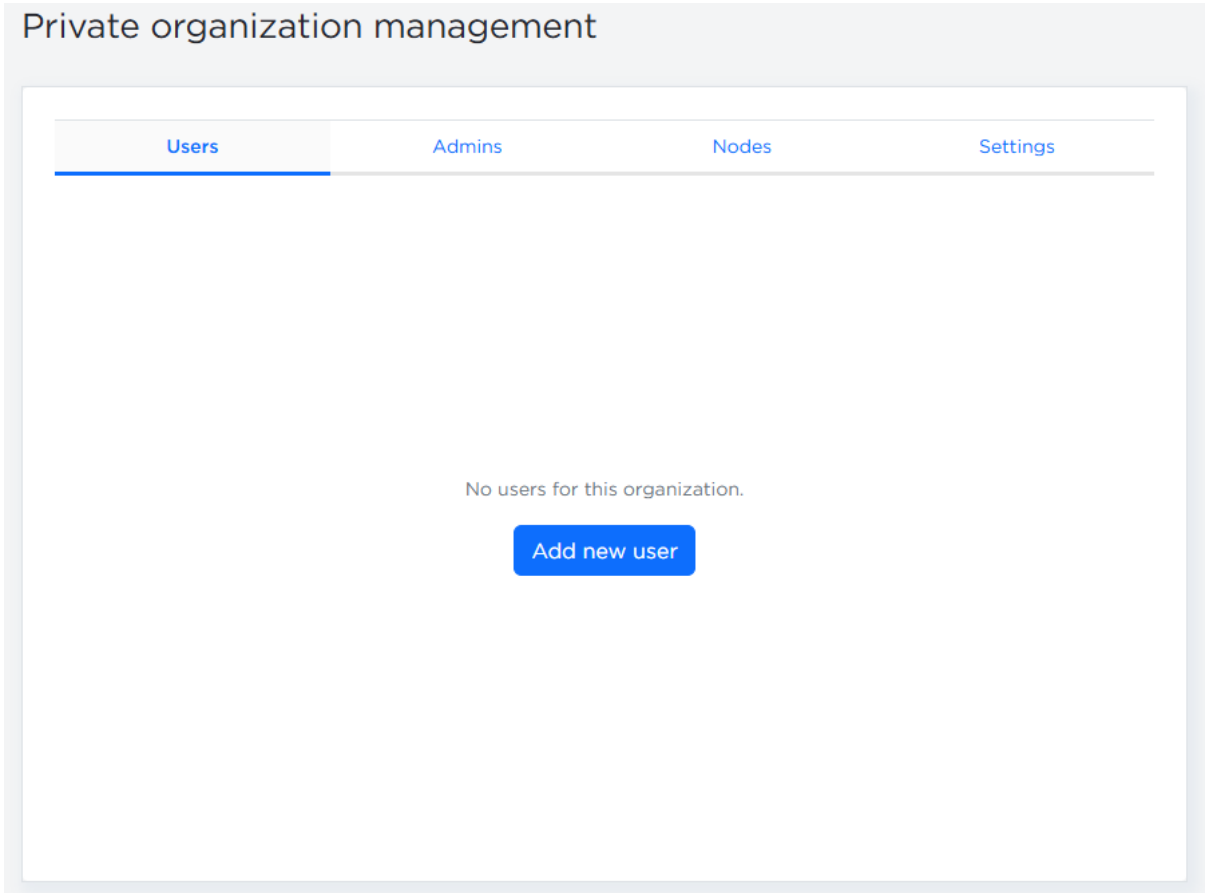
Adicionando usuários às organizações

Semelhante aos usuários no Zabbix, você pode criar usuários no Zabbix Cloud e atribuir a eles papéis para controlar o que eles podem fazer dentro de uma organização e seus nós.

Para adicionar usuários à sua organização:

1. Abra a página *Todas as organizações*.

2. Clique no botão de configurações da organização desejada para acessar o menu de gerenciamento.



3. Na aba *Usuários* ou *Administradores*, clique em *Adicionar novo usuário*.

4. Insira o e-mail do usuário e selecione um papel: *Administrador* ou *Usuário*.

Note que selecionar o papel *Usuário* adiciona o usuário à organização, mas não concede acesso a nenhum nó.

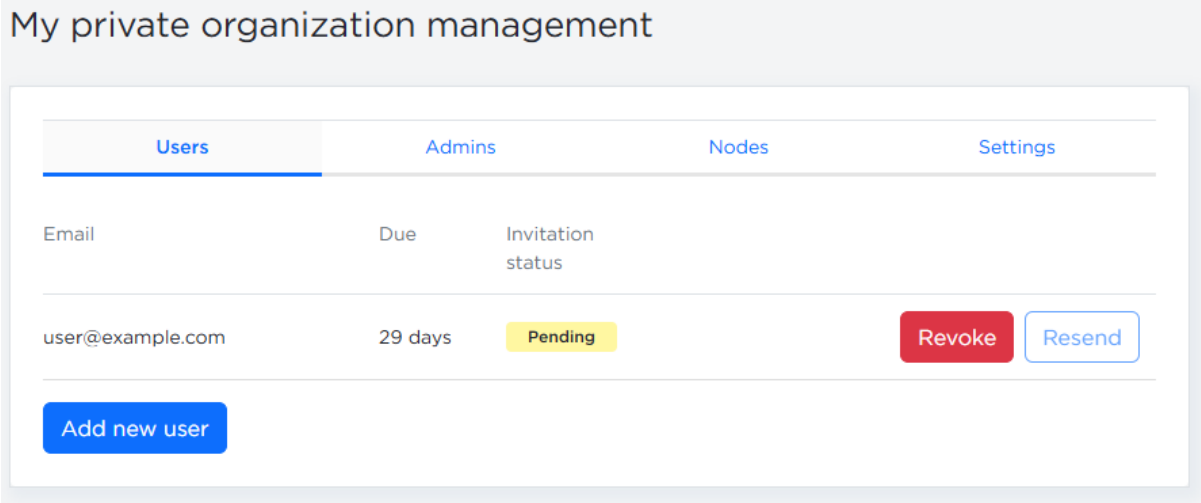
Cada papel concede um nível diferente de acesso:

- **Usuário** – pode acessar os nós aos quais está atribuído e pode executar tarefas de configuração que **não geram custos**, por exemplo:

  - Gerenciar filtros de acesso
  - Configurar criptografia
  - Definir períodos de retenção de histórico/tendência
  - Agendar manutenção
- **Administrador** – possui todas as permissões do Usuário e também pode realizar ações que **podem gerar custos**, por exemplo:
  - Criar e restaurar backups
  - Atualizar ou rebaixar recursos do nó (nível de computação e tamanho do disco)
  - Cancelar assinaturas
- **Proprietário** – tem controle total sobre a organização. Além de todas as permissões do Administrador, o Proprietário pode:
  - Criar e destruir nós
  - Adicionar ou remover Usuários e Administradores
  - Renomear ou deletar a organização
  - Gerenciar detalhes de cobrança

> **Attention:**
> O método de pagamento do Proprietário é cobrado por todos os serviços. Se for necessária a confirmação bancária, o Proprietário deve autorizar a transação, mesmo que tenha sido iniciada por um Administrador.

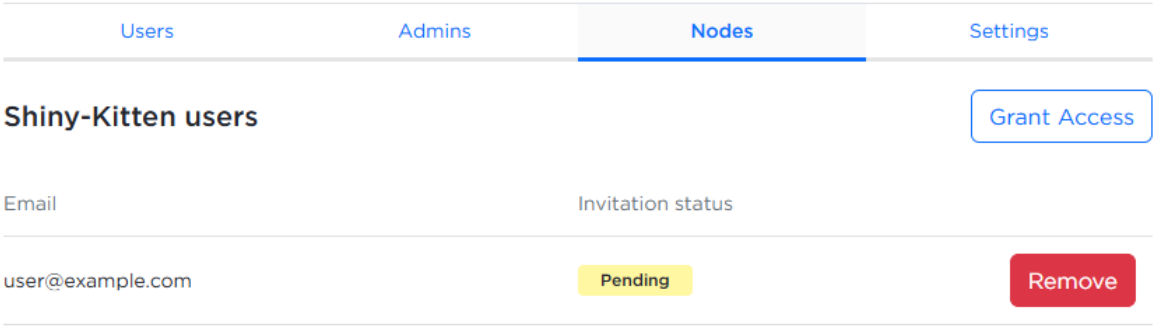
O usuário convidado será adicionado com status *Pendente*.
Você pode revogar o convite ou reenviá-lo (o reenvio está disponível a cada 10 minutos — o botão ficará desabilitado se esse limite for atingido).



Adicionando usuários aos nós

Para adicionar o usuário ao nó:

1. Acesse a aba *Nós*.
2. Clique em *Conceder acesso*.
3. Selecione um usuário adicionado à organização no menu suspenso.
4. Clique em *Conceder acesso*.



Como alternativa, você pode conceder acesso na janela *Configuração do nó*:

1. Clique no botão de configurações .
2. Selecione *Adicionar usuário*.
3. Selecione um usuário e clique em *Conceder acesso*.

Os usuários convidados devem aceitar o convite criando uma conta no Zabbix Cloud usando o mesmo endereço de e-mail para onde o convite foi enviado. Se o usuário já tiver uma conta no Zabbix Cloud com esse e-mail, ele deverá efetuar login e confirmar o convite. Os convites são vinculados a um endereço de e-mail específico e não podem ser aceitos de uma conta diferente.

4 Principais diferenças do Zabbix Cloud

Tabela de comparação Embora a maior parte da funcionalidade seja semelhante entre instalações Zabbix on-premise e nós em nuvem, também existem diferenças. Abaixo está uma tabela de comparação resumida. As seções seguintes descrevem itens selecionados em detalhes.

Recurso / Aspecto	Zabbix Cloud	Zabbix on-premises
Ciclo de lançamento	Acompanha os lançamentos LTS	Você escolhe sua versão do Zabbix (estável, beta, builds customizados)
Facilidade de uso	Plataforma pronta para uso, totalmente gerenciada pela Zabbix	Requer instalação, configuração e manutenção
Correção de vulnerabilidades	Correção automática do SO/Zabbix	Correção manual pelo usuário
Certificado HTTPS	Certificado válido pronto para uso (sem Let's Encrypt)	Autoassinado por padrão (Let's Encrypt ou outra CA pode ser configurada)
Desempenho do BD (INSERT/UPDATE/SELECT)	Particionamento automático para velocidade máxima; sem DELETEs de longa duração	Particionamento e limpeza manuais (transações DELETE longas são possíveis)
Ajuste de desempenho	Pré-ajustado pela equipe Zabbix Cloud (sem changelogs)	Total visibilidade e controle sobre parâmetros de ajuste
Controle de firewall	Gerenciado pela equipe Zabbix Cloud; portas GUI/trapper não expostas publicamente	Controle total das regras de firewall e exposição de portas
Isolamento e proteção do BD	Altamente isolado, sem acesso direto ao BD/SSH	Depende da sua rede e segurança do host
Alta disponibilidade (HA)	Executado em uma plataforma de nuvem altamente disponível, proxies Zabbix podem ser usados para maior resiliência na coleta de dados	HA gerenciada pelo usuário
Traps SNMP	Apenas via proxy dedicado (sem receptor de trap HA/balanceado)	Direto ou via proxy, HA/balanceamento possível
Polling SNMP	Difícil sem proxies (cada dispositivo precisa de NAT/porta customizada)	Polling SNMP nativo; proxy opcional
Configurações de retenção	Histórico/Tendências/Auditoria apenas via UI (sem API ou arquivo de configuração)	Configurável no arquivo de configuração do server ou via API
Scripts customizados	`AlertScriptsPath`, `ExternalScripts`, módulos frontend e da comunidade não suportados	Totalmente suportado (caminho de scripts, módulos, integrações)
Monitoramento ODBC	Apenas PostgreSQL (driver `{postgresql}`); plugin MariaDB presente como placeholder	ODBC para PostgreSQL, MySQL, Oracle, etc. - totalmente configurável
Limitação de chamadas ODBC	Não é possível (`StartODBCPollers=1` apenas; queries pesadas bloqueiam outros trabalhos)	Totalmente ajustável quantidade de pollers e agendamento
Upload de certificado SAML	Não suportado	Suportado via UI ou API
Relatórios agendados	Deve criar seu próprio Email (script não suportado)	Suporta script e Email prontos para uso
Interface do host para checagens ativas	Endereço IP emitido pela plataforma (criado automaticamente)	Interfaces gerenciadas por você; IPs sob seu controle

Diferenças de recursos Traps SNMP

Traps SNMP são suportadas apenas por meio de um proxy Zabbix dedicado. Se o monitoramento de traps SNMP for necessário, o balanceamento de carga automatizado ou alta disponibilidade para o proxy não poderá ser usado porque as traps SNMP devem ser enviadas para um único endereço IP.

SNMP polling

Sem proxies, o SNMP polling requer que cada dispositivo seja exposto via NAT e portas personalizadas.

Configurações de retenção

Os períodos de retenção de histórico, tendências e log de auditoria só podem ser configurados pela interface web na nuvem. Essas configurações não podem ser feitas através do `zabbix_server.conf` ou da API.

Substituições manuais de histórico por item não são suportadas (o particionamento é controlado globalmente). As URLs dos nós em nuvem não aceitam os mesmos parâmetros de consulta que as instalações locais.

Scripts personalizados

Os seguintes tipos de scripts personalizados não são suportados pelo Zabbix Cloud:

- Scripts de alerta (`AlertScriptsPath`)
- Scripts externos (`ExternalScripts`)
- Scripts de frontend no Zabbix Server

Módulos de frontend desenvolvidos pela comunidade não podem ser instalados.

Monitoramento ODBC

O Zabbix Cloud suporta o monitoramento ODBC apenas para PostgreSQL. Use o template ODBC oficial do Zabbix e defina a string de conexão no template usando:

```
Driver={postgresql}
```

Um plugin MariaDB também está instalado, mas atualmente não está funcional. Se for utilizado, defina:

```
Driver={mysql}
```

Attention:
Não há configuração funcional conhecida para Oracle MySQL 8.0 no Zabbix Cloud. Embora consultas simples (como `SELECT 1`) possam ser bem-sucedidas, consultas mais complexas retornam `SQL_ERROR`.

 Não é possível limitar as chamadas ODBC (`StartODBCPollers=1` apenas). Relatórios síncronos pesados podem impactar o desempenho, e apenas um único `SELECT` pode ser executado por vez.

Acesso à infraestrutura

O Zabbix Cloud não fornece acesso SSH aos nós subjacentes, nem permite conexões diretas ao banco de dados (por exemplo, à porta 3306). Todas as atividades de configuração, monitoramento e solução de problemas devem ser realizadas por meio da interface da nuvem ou da API, garantindo que as camadas do sistema operacional e do banco de dados permaneçam isoladas e seguras.

Interface de verificação ativa

Verificações ativas na nuvem criam automaticamente uma interface de host com um IP não relacionado à sua rede local. Por padrão, esse IP pode ser externo. Para manter a consistência, você pode excluir ou ajustar manualmente a interface do host após a criação.

Upload de certificado SAML

A autenticação SAML não é suportada porque poucos provedores aceitam mensagens SAML não criptografadas ou não assinadas.

Relatórios agendados

O tipo de mídia padrão *Cloud Email* usa um transporte de script e não pode ser usado para relatórios agendados. Para enviar relatórios por e-mail:

1. Crie um novo tipo de mídia Email.
2. Atribua o e-mail à sua conta em *Configurações do usuário* > *Mídia*.
3. Selecione esse tipo de mídia ao configurar relatórios agendados.

Central do desenvolvedor

Central do desenvolvedor

Comece a desenvolver extensões personalizadas para o Zabbix com orientações sobre módulos, widgets e plugins.

Construa no Zabbix

Módulos do frontend Guias e referências para criar módulos personalizados que estendem ou modificam o frontend do Zabbix para atender a casos de uso específicos.

Widgets Uma análise da estrutura e lógica dos widgets, com instruções para criar elementos personalizados de dashboard adaptados às suas necessidades.

Plugins Uma visão geral sobre como desenvolver e gerenciar plugins do Zabbix para estender a funcionalidade ou integrar com sistemas externos.

Módulos

O que é um módulo de frontend PHP?

- um módulo é uma entidade com um ID único, nome, descrição, autor e outros campos definidos em seu arquivo de manifesto, juntamente com arquivos PHP, JavaScript e outros localizados em um único diretório dentro do diretório *modules*
- Um módulo deve seguir regras simples para garantir operação correta.
- Um módulo deve ser instalado (descompactado) e ativado no frontend pelo administrador.

Para que um módulo pode ser usado?

- Adicionando novas funcionalidades por meio de seções frontend personalizadas;
- Criação de tipos de widget de painel personalizados (consulte widget modules);
- Substituindo ou estendendo a funcionalidade existente.

Para que um módulo não pode ser usado?

- Registrar um novo método de API ou modificar uma existente.

Como funcionam os módulos

- Um módulo habilitado é lançado em cada solicitação HTTP, antes de executar o código de ação.
- O módulo irá cadastrar novas ações ou redefinir as existentes.
- O módulo adicionará novas seções de frontend e removerá ou redefinirá as existentes.
- O módulo se conectará a eventos de frontend como onBeforeAction e onTerminate, se necessário.
- A ação solicitada é finalmente executada executando o código de ação - o padrão ou definido pelo módulo.

Para onde ir em seguida Quer você prefira aprender fazendo ou lendo primeiro as diretrizes, estas páginas contêm as informações e as etapas necessárias para construir seus próprios módulos:

- Tutoriais passo a passo para escrever seu primeiro módulo
- Estrutura do arquivo do módulo
- Especificações do módulo Widget
- Exemplos de módulos para reutilizar

Estrutura do arquivo do módulo

Todo o código relacionado a um módulo é armazenado em um único diretório dentro de **modules** do seu Zabbix frontend (por exemplo, *zabbix/ui/modules*).

```
example_module_directory/ (required)
   manifest.json (required) Metadados e definição de ação.
   Module.php Inicialização do módulo e tratamento de eventos.
   actions/ Arquivos do controlador de ação.
   SomethingView.php
   SomethingCreate.php
   SomethingDelete.php
   data_export/
   ExportAsXml.php
   ExportAsExcel.php
   views/ Visualizar arquivos.
   example.something.view.php
   example.something.delete.php
   assets/ Quaisquer arquivos adicionais a serem usados nas visualizações. Deve ser especificado em manife
   js/ Arquivos JavaScript usados nas exibições.
   example.something.view.js.php
   css/ Arquivos CSS usados em visualizações.
   example.something.css
   image.png Imagens usadas nas exibições.
   example.something.file Qualquer arquivo para uso em visualizações.
```

Árvore de arquivos do módulo

Escrevendo um módulo Um exemplo do processo de escrita de módulo consiste nas seguintes etapas (quando disponível, clique no nome do arquivo ou da pasta para exibir detalhes adicionais sobre a etapa):

1. Crie um novo diretório para o módulo dentro de **zabbix/ui/modules/**.
2. Adicione o arquivo manifest.json com os metadados do módulo.
3. Crie a pasta *views* e defina uma ou mais visualizações do módulo.
4. Crie a pasta *actions* e defina uma ou mais ações do módulo.
5. Crie o arquivo Module.php (ou Widget.php para widgets do dashboard) e defina as regras de inicialização e de tratamento de eventos.
6. Crie a pasta *assets* para arquivos JavaScript (coloque em *assets/js*), estilos CSS (coloque em *assets/css*) ou quaisquer outros arquivos adicionais.
7. Certifique-se de especificar as exibições, ações e arquivos de ativos necessários no manifest.json.
8. Registre o módulo no front-end do Zabbix e comece a usá-lo.

Você pode registrar e ativar um módulo assim que criar o arquivo manifest.json. Quando o módulo estiver habilitado, você poderá visualizar todas as alterações feitas nos arquivos do módulo imediatamente, atualizando o front-end do Zabbix.

manifest.json

Qualquer módulo precisa do arquivo manifest.json. O arquivo deve estar localizado no diretório principal do módulo (por exemplo, *zabbix/ui/modules/module_name/manifest.json*).

No mínimo, o manifest.json deve especificar estes campos:

```
{
    "manifest_version": 2.0,
    "id": "my_ip_address",
    "name": "Meu endereço IP",
    "namespace": "MyIPAddress",
    "version": "1.0"
}
```

Parâmetros suportados em **manifest.json** (pressione o nome do parâmetro para obter uma descrição detalhada):

Parâmetro	Descrição	Requerido
manifest_version	Versão do manifesto do módulo.	Sim
id	ID exclusivo do módulo.	
name	Nome do módulo que será exibido na seção Administração.	
namespace	Espaço de nomes PHP para classes de módulo.	
version	Versão do módulo.	

Parâmetro	Descrição	Requerido
type	Tipo do módulo. Para widget deve ser definido como *widget*	Sim para widgets, caso contrário, não
widget	Configuração do widget. Usado somente para widgets.	
actions	Ações a serem registradas no módulo.	
assets	Estilos CSS e arquivos JavaScript a serem incluídos.	Não
author	Autor do módulo.	
config	Valores padrão para opções personalizadas do módulo.	
description	Descrição do módulo.	
url	Um link para a descrição do módulo.	

manifest_version

Versão do manifesto do módulo. Atualmente, a versão compatível é **2.0**.

Tipo: Double

Exemplo:

```
"manifest_version": 2.0
```

id

ID do módulo. Deve ser exclusivo. Para evitar futuros conflitos de nomes, recomenda-se usar o prefixo para os módulos (nome do autor ou da empresa, ou qualquer outro). Por exemplo, se um módulo for um exemplo de aula e o nome do módulo for ”My module”, o ID será ”example_my_module”.

Type: String

Exemplo:

```
"id": "example_my_module"
```

nome

Nome do módulo que será exibido na seção Administração.

Type: String

Exemplo:

```
"name": "My module"
```

Espaço de nomes

Espaço de nome PHP para classes de módulo.

Type: String

Exemplo:

```
"namespace": "ClockWidget"
```

versão

Versão do módulo. A versão será exibida na seção Administração.

Tipo: String

Exemplo:

```
"version": "1.0"
```

tipo

Tipo do módulo. Obrigatório para widgets e deve ser igual a ”widget”.

Type: String

Padrão: ”module”

Exemplo:

```
"type": "widget"
```

ações

Ações a serem registradas no módulo. É necessário definir a chave de objeto *class* para cada ação; outras chaves de ação são opcionais.

Type: Object

Chaves de objeto compatíveis se *type* for *module*:

-**write.your.action.name** (object) - nome da ação, deve ser escrito em letras minúsculas [a-z], separando as palavras com pontos. Suporta as chaves: - **class** (string; obrigatório) - nome da classe da ação. - **layout** (string) - layout da ação. Valores suportados: *layout.json*, *layout.htmlpage* (padrão), *null*. - **view** (string) - visualização da ação.

Exemplo:

```
"actions": {
    "module.example.list": {
        "class": "ExampleList",
        "view": "example.list",
        "layout": "layout.htmlpage"
        }
    }
```

Chaves de objeto compatíveis se *type* for *widget*:

- **widget.{id}.view** (objeto) - nome do arquivo e da classe para a exibição do widget. Substitua **{id}** pelo valor id do widget (por exemplo, *widget.example_clock.view*). Suporta as chaves:
  - **class** (string; obrigatório) - nome da classe de ação para o modo de exibição do widget para estender a classe padrão CControllerDashboardWidgetView. O arquivo de origem da classe deve estar localizado no diretório *actions*.
  - **view** (string) - visualização do widget. Deve estar localizada no diretório *views*. Se o arquivo de visualização for *widget.view.php*, o que é esperado por padrão, esse parâmetro poderá ser omitido. Se estiver usando um nome diferente, especifique-o aqui.
- **widget.{id}.edit** (objeto) - nome do arquivo da visualização de configuração do widget. Substitua **{id}** pelo valor id do widget (por exemplo, *widget.example_clock.edit*). Suporta as chaves:
  - **class** (string; obrigatório) - nome da classe de ação para o modo de exibição de configuração do widget. O arquivo de origem da classe deve estar localizado no diretório *actions*.
  - **view** (string) - visualização de configuração do widget. Deve estar localizada no diretório *views*. Se o arquivo de visualização for *widget.edit.php*, o que é esperado por padrão, esse parâmetro poderá ser omitido. Se estiver usando um nome diferente, especifique-o aqui.

*Exemplo:

```
"actions": {
    "widget.tophosts.view": {
        "class": "WidgetView"
    },
    "widget.tophosts.column.edit": {
        "class": "ColumnEdit",
        "view": "column.edit",
        "layout": "layout.json"
    }
}
```

ativos

Estilos CSS e arquivos JavaScript a serem incluídos.

Type: Object

Supported object keys:

- **css** (array) - Arquivos CSS a serem incluídos. Os arquivos devem estar localizados em *assets/css*.
- **js** (array) - Arquivos JavaScript a serem incluídos. Os arquivos devem estar localizados em *assets/js*.

Exemplo:

```
"assets": {
    "css": ["widget.css"],
    "js": ["class.widget.js"]
}
```

autor

Autor do módulo. O autor será exibido na seção Administração.

Type: String

Exemplo:

```
"autor": "John Smith"
```

config

Valores padrão para as opções do módulo. O objeto pode conter quaisquer chaves personalizadas. Se especificado, esses valores serão gravados no banco de dados durante o registro do módulo. Novas variáveis adicionadas posteriormente serão gravadas na primeira chamada. Depois disso, os valores das variáveis só poderão ser alterados diretamente no banco de dados.

Type Object

Exemplo:

```
"config": {
    "username" (nome de usuário): "Admin",
    "password" (senha): "",
    "auth_url": "https://example.com/auth"
}
```

descrição

Descrição do módulo.

Type: String

Exemplo:

```
"description": "This is a clock widget."
```

Widget

Configuração do widget. Usado se *tipo* estiver definido como *widget*.

Tipo: Object

Chaves de objeto suportadas:

-**name** (string) - usado na lista de widgets e como cabeçalho padrão. Se estiver vazio, será usado o parâmetro ”name” do módulo.

- **size** (object) - dimensões do widget. Chaves compatíveis:
  - *width* (integer) - largura padrão do widget.
  - *height* (integer) - altura padrão do widget.
- **form_class** (string) - classe com os campos de formulário do widget. Deve estar localizada no diretório *includes*. Se a classe for *WidgetForm.php*, o que é esperado por padrão, esse parâmetro poderá ser omitido. Se estiver usando um nome diferente, especifique-o aqui.
- **js_class** (string) - nome de uma classe JavaScript para o modo de exibição do widget para estender a classe padrão CWidget. A classe será carregada com o dashboard. O arquivo de origem da classe deve estar localizado no diretório assets/js. Veja também: assets.
- **use_time_selector** (boolean) - determina se o widget requer o seletor de periodo de tempo do dashboard. Valores suportados: *true*, *false* (padrão).
- **refresh_rate** (integer) - taxa de atualização do widget em segundos (padrão: 60).

Exemplo:

```
"widget": {
"name": "",
"size": {
"width": 12,
"height": 5
},
"form_class": "WidgetForm",
"js_class": "CWidget",
"use_time_selector": false,
```

```
"refresh_rate": 60
}
```

url

Um link para a descrição do módulo. Para widgets, esse link será aberto quando o ícone de ajuda for clicado ? na janelas *Adicionar widget* ou *Editar widget*. Se a **url** não for especificada, ícone de ajuda irá abrir a página geral de widgets de Dashboard.

Tipo: String

Exemplo:

```
"url": "http://example.com"
```

Ações

As ações são responsáveis pela 'lógica de negócios' do módulo. Uma ação geralmente consiste em um controlador e uma visualização de ação.

Um módulo pode:

- Ações de chamada que já estão definidas no frontend do Zabbix.
- Substituir ações padrão por ações personalizadas.
- Definir ações completamente novas.

Para substituir um comportamento de ação padrão por algum comportamento personalizado, defina uma ação com o mesmo nome na configuração do módulo. Quando a ação é chamada, a ação do módulo será executada em vez da ação padrão do Zabbix.

Os arquivos de ação devem ser armazenados na pasta *actions*. Ações precisam ser especificadas no manifest.json.

Controlador

Workflow do controlador de ação:

1) Verifique se todos os parâmetros passados em uma requisição HTTP são válidos:
   - Chame o método do controlador *checkInput()*
   - Use as regras de validação definidas em CNewValidator.php
   - Chame o método *validateInput()*
2) Verifique as permissões do usuário.
3) Prepare os dados de acordo com os parâmetros passados: se *checkInput()* retornar true, o Zabbix chama o método do controlador *doAction()*.
4) Prepare o array **$data** para a visualização. Use os métodos *CControllerResponseData* e *setResponse()* para armazenar a resposta no array **$data**.

Exemplo:

```
/**
 * Valide os parâmetros de entrada.
 *
 * @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;
}

/**
 * Verifique as permissões do usuário.
 *
```

```
 * @return bool
 */
protected function checkPermissions() {
    return $this->getUserType() >= USER_TYPE_ZABBIX_ADMIN;
}

/**
 * Execute a ação e gere o objeto de resposta.
 */
protected function do Action(): void {
    $data = [
        'hosts_count' => API::Host()->get([
            'countOutput' => true,
            'filter' => [
                'status' => $this->getInput('status')
            ]
        ])
    ];

    $this->setResponse(new CControllerResponseData($data));
}

::: Nota
Você pode ver a lista completa de classes do controlador disponíveis no Zabbix [código-fonte] (https://git
:::
```

{#devel-modules-file_structure-views}

\

O arquivo Views

View recebe os dados de um controlador e, em seguida, prepara a aparência HTML deles.

:::noteclassic
Definir view(s) para um módulo de front-end é opcional, a menos que o módulo seja um widget.\
Os widgets de painel precisam de pelo menos duas views: uma para o modo de edição e outra para o modo de v
:::

É possível usar classes HTML predefinidas do Zabbix (do diretório */zabbix/ui/include/classes/html*) na vi
As novas classes devem ser armazenadas na pasta *assets* do módulo.

Exemplo:

... (new CColHeader(_('Name')))

Visualização de ação

Este é um arquivo de referência para definir uma visualização de ação.

````
```php
<?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();
```

Ativos

A pasta *ativos* pode conter quaisquer arquivos ou subpastas que não pertençam a outros diretórios. Você pode usar isso para:

- Estilos JavaScript (deve estar dentro de *assets/js*);
- Estilos CSS (deve estar dentro de *assets/css*);
- Imagens;
- Fontes;
- Qualquer outra coisa que você precise incluir.

assets/js

A pasta *assets/js* é reservada e deve conter apenas arquivos JavaScript. Para ser usada pelo widget, especifique esses arquivos em: *manifest.json*.

Por exemplo:

```
"assets": {
"js": ["class.widget.js"]
}
```

ativos/css

O *assets/css* é reservado e deve conter apenas arquivos de estilo CSS. Para ser usado pelo widget, especifique esses arquivos no *manifest.json*.

Por exemplo:

```
"assets": {
    "css": ["mywidget.css"]
}
```

Estilos CSS

Os arquivos CSS podem conter um atributo personalizado `theme` para definir um estilo diferente para um tema de front-end específico.

Temas disponíveis e seus valores de atributo:

- **Blue** - [theme='blue-theme']
- **Dark** - [theme='dark-theme']
- **High-contrast light** - [theme='hc-light']
- **High-contrast dark** - [theme='hc-dark']

Exemplo:

```
.widget {
    background-color: red;
}

[theme='dark-theme'] .widget {
    background-color: green;
}
```

Registrar um novo módulo

Esta seção explica como adicionar um novo módulo no Zabbix frontend.

Pré-requisitos Antes de prosseguir, certifique-se de que:

- O módulo está localizado dentro do diretório *modules* no seu Zabbix frontend (por exemplo, *zabbix/ui/modules*).

- O módulo tem pelo menos uma versão básica do arquivo manifest.json (/devel/modules/file_structure/manifest).
- Você tem acesso à seção do menu Administração no Zabbix (requer o tipo de função de usuário Super admin).

O front-end não instalará nem reconhecerá módulos incompatíveis.

Adição de um módulo Abra a página *Administration→General→Modules* e pressione *Scan directory*.

Localize seu módulo na lista e ative-o.

Para ativar um módulo, pressione o hiperlink *Disabled* - o estado do módulo mudará para *Enabled*.

Pressione o nome do módulo para exibir informações adicionais sobre o módulo, como autor, versão ou descrição curta (se definida no manifesto).

Visualização do widget Os módulos de widget, uma vez adicionados, tornam-se imediatamente visíveis na lista de widgets do painel.

Você pode abrir um painel, alternar para o modo de edição e adicionar o widget ao painel como de costume.

Quando você fizer algumas alterações no widget, atualize o painel para ver a aparência do widget com as atualizações mais recentes.

Widgets

Os widgets são módulos de front-end do Zabbix usados para os painéis. Salvo indicação em contrário, todas as diretrizes do módulo também se aplicam aos widgets.

Entretanto, um widget é notavelmente diferente de um módulo. Para criar um widget:

- especifique o tipo ”widget” no [arquivo manifest.json] (/devel/modules/file_structure/manifest) (”type”: ”widget”);
- inclua pelo menos duas exibições: uma para o modo de apresentação do widget e uma para o modo de configuração do widget (example.widget.view.php e example.widget.edit.php);
- e um controlador para apresentação do widget (WidgetView.php);
- usar e estender [classes de widget] padrão (/devel/modules/widgets/configuration).

Configuração

Esta página descreve as classes que podem ser usadas para criar uma visualização de configuração do widget com campos de configuração personalizados. A exibição de configuração do widget é a parte do widget que permite ao usuário configurar os parâmetros do widget para presentation.

Widget

Classe primária de widget, estende a classe base de todos os widgets do painel - *CWidget*. Necessário para substituir o comportamento padrão do widget.

A classe *Widget* deve estar localizada no diretório raiz do widget (por exemplo, *ui/modules/my_custom_widget*).

Exemplo de **Widget.php**

```
<?php

namespace Modules\MyCustomWidget;

use Zabbix\Core\CWidget;

class Widget extends CWidget {

    public const MY_CONSTANT = 0;

    função pública getTranslationStrings(): array {
        retorna [
            'class.widget.js' => [
                'Sem dados' => _('Sem dados')
```

```
            ]
        ];
    }
}
```

WidgetForm

A classe *WidgetForm* estende a classe padrão *CWidgetForm* e contém um conjunto de campos *CWidgetField* que são necessários para definir a estrutura de armazenamento da configuração do widget no banco de dados e manipular a validação de entrada.

A classe *WidgetForm* deve estar localizada no diretório *includes*. Se a classe tiver um nome diferente, o nome deverá ser especificado no parâmetro *widget/form_class* no arquivo *manifest.json*.

Exemplo de **includes/WidgetForm.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 {

    função pública addFields(): self {
        return $this
            ->addField(
                (novo CWidgetFieldMultiSelectItem('itemid', _('Item')))
                    ->setFlags(CWidgetField::FLAG_NOT_EMPTY | CWidgetField::FLAG_LABEL_ASTERISK)
                    ->setMultiple(false)
                    ->setFilterParameter('numeric', true)
            )
            ->addField(
                new CWidgetFieldTextBox('description', _('Description'))
            )
            ->addField(
                (new CWidgetFieldColor('chart_color', _('Color')))->setDefault('FF0000')
            );
    }
}
```

CWidgetFormView

A classe *CWidgetFormView* é necessária para especificar a lógica de apresentação dos campos definidos na classe *WidgetForm*, determinando sua aparência e comportamento quando renderizados na exibição de configuração.

A classe *CWidgetFormView* é compatível com os seguintes métodos:

- *addField()* - recebe uma instância da classe *CWidgetFieldView* como parâmetro; Cada classe *CWidgetField* tem uma respectiva classe *CWidgetFieldView* para uso na exibição de configuração do widget.
- *includeJsFile()* - permite adicionar um arquivo JavaScript à visualização de configuração do widget.
- *addJavaScript()* - permite adicionar JavaScript em linha que será executado assim que a visualização de configuração do widget for carregada.

A classe *CWidgetFormView* deve estar localizada no diretório *views*.

Exemplo de **views/widget.edit.php**

```php
<?php

/**
 * Minha visualização de formulário de widget personalizado.
 *
 * @var CView $this
 * @var array $data
 */

use Zabbix\Widgets\Fields\CWidgetFieldGraphDataSet;

(novo CWidgetFormView($data))
    ->addField(
        new CWidgetFieldMultiSelectItemView($data['fields']['itemid'], $data['captions']['items']['itemid'
    )
    ->addField(
        new CWidgetFieldTextBoxView($data['fields']['description']),
        'js-advanced-configuration'
    )
    ->addField(
        new CWidgetFieldColorView($data['fields']['chart_color']),
        'js-advanced-configuration'
    )
    ->includeJsFile('widget.edit.js.php')
    ->addJavaScript('my_widget_form.init('.json_encode([
        'color_palette' => CWidgetFieldGraphDataSet::DEFAULT_COLOR_PALETTE
    ], JSON_THROW_ON_ERROR).');')
    ->show();
```

JavaScript

Uma classe JavaScript pode ser usada para adicionar comportamento dinâmico e interatividade à visualização de configuração do widget. Por exemplo, você pode inicializar um seletor de cores, definido na classe *CWidgetFormView*.

A classe JavaScript deve ser carregada com o formulário, portanto, deve ser referenciada na classe *CWidgetFormView* usando os métodos *includeJsFile()* e *addJavaScript()*.

No exemplo abaixo, uma instância de classe singleton é imediatamente criada e armazenada sob o nome *window.my_custom_widget_form*. Assim, abrir o formulário pela segunda vez recriará a instância.

A classe JavaScript deve estar localizada no diretório *views*.

Exemplo de views/widget.edit.js.php

```php
<?php

use Modules\MyCustomWidget\Widget;

?>

window.my_custom_widget_form = new class {

init({color_palette}) {
colorPalette.setThemeColors(color_palette);

for (const colorpicker de jQuery('.<?= ZBX_STYLE_COLOR_PICKER ?> input')) {
jQuery(colorpicker).colorpicker();
}

const overlay = overlays_stack.getById('widget_properties');

for (const event de ['overlay.reload', 'overlay.close']) {
overlay.$dialogue[0].addEventListener(event, () => { jQuery.colorpicker('hide'); });
}
}
};
```

CWidgetField

A classe *CWidgetField* é uma classe base da qual todas as classes de campos de formulário (*CWidgetFieldCheckBox*, *CWidgetFieldTextArea*, *CWidgetFieldRadioButtonList*, etc.) são herdadas. As classes que estendem *CWidgetField* são responsáveis por receber, salvar e validar os valores de configuração do widget.

As seguintes classes *CWidgetField* estão disponíveis.

Classe CWidgetField	Tipo de campo de banco de dados	Descrição
CWidgetFieldCheckBox	int32	Caixa de seleção única.
CWidgetFieldCheckBoxList	matriz de int32	Várias caixas de seleção em um único campo de configuração.
CWidgetFieldColor	string	Campo de seleção de cor.
CWidgetFieldDatePicker	string	Campo de seleção de data.
CWidgetFieldHostPatternSelect	string	Campo de seleção múltipla que permite selecionar um ou vários hosts. Suporta a definição de padrões de nome de host (todos os hosts correspondentes serão selecionados).
CWidgetFieldIntegerBox	int32	Campo para inserir um número inteiro. Pode ser usado para configurar valores mínimo e máximo.
CWidgetFieldLatLng	string	Caixa de texto que permite inserir latitude, longitude e nível de zoom do mapa, separados por vírgula.
CWidgetFieldMultiSelectAction	ID	Campo de seleção múltipla para selecionar ações (da lista de ações definidas em *Alertas → Ações*).
CWidgetFieldMultiSelectGraph	ID	Campo de seleção múltipla para selecionar gráficos personalizados.
CWidgetFieldMultiSelectGraphPrototype	ID	Campo de seleção múltipla para selecionar protótipos de gráficos personalizados.
CWidgetFieldMultiSelectGroup	ID	Campo de seleção múltipla para selecionar grupos de hosts.
CWidgetFieldMultiSelectHost	ID	Campo de seleção múltipla para selecionar hosts.
CWidgetFieldMultiSelectItem	ID	Campo de seleção múltipla para selecionar itens.
CWidgetFieldMultiSelectItemPattern	ID	Campo de seleção múltipla para selecionar padrões de itens.
CWidgetFieldMultiSelectItemPrototype	ID	Campo de seleção múltipla para selecionar protótipos de itens.
CWidgetFieldMultiSelectMap	ID	Campo de seleção múltipla para selecionar mapas.
CWidgetFieldMultiSelectMediaType	ID	Campo de seleção múltipla para selecionar tipos de mídia.
CWidgetFieldMultiSelectOverrideHost	ID	Campo de seleção múltipla para selecionar uma fonte de dados (painel ou outro widget) contendo um host para o qual o widget pode exibir dados.
CWidgetFieldMultiSelectService	ID	Campo de seleção múltipla para selecionar serviços.
CWidgetFieldMultiSelectSla	ID	Campo de seleção múltipla para selecionar SLAs.
CWidgetFieldMultiSelectUser	ID	Campo de seleção múltipla para selecionar usuários.
CWidgetFieldNumericBox	string	Campo para inserir um número flutuante.
CWidgetFieldRadioButtonList	int32	Grupo de caixas de opção que consiste em uma ou mais caixas de opção.
CWidgetFieldRangeControl	int32	Controle deslizante para selecionar um valor do tipo inteiro.
CWidgetFieldReference	string	Cria um identificador exclusivo para este widget no painel. Ele é usado para referenciar este widget a partir de outros widgets.
CWidgetFieldSelect	int32	Caixa de seleção suspensa.
CWidgetFieldSeverities	matriz de int32	*CWidgetFieldCheckBoxList* predefinida com severidades de gatilho.
CWidgetFieldTags	matriz de (string, int32, string)	Permite configurar uma ou mais linhas de filtro de tags.
CWidgetFieldTextArea	string	Área de texto para inserir texto com várias linhas.
CWidgetFieldTextBox	string	Caixa de texto para inserir texto com uma única linha.
CWidgetFieldTimePeriod	matriz de strings	Campo de seleção de período.
CWidgetFieldTimeZone	string	Lista suspensa com fusos horários.
CWidgetFieldThresholds	matriz de (string, string)	Permite configurar pares de cores e números.
CWidgetFieldUrl	string	Caixa de texto que permite inserir URLs.

As seguintes classes *CWidgetField* foram criadas para widgets específicos. Essas classes têm casos de uso muito específicos, mas também podem ser reutilizadas, se necessário.

Classe CWidgetField	Tipo de campo do banco de dados	Descrição
CWidgetFieldColumnsList	matriz de (vários mistos)	Para o widget *Principais hosts*. Cria uma tabela com colunas personalizadas dos tipos permitidos.
CWidgetFieldNavTree	string	Para o widget *Árvore de navegação do mapa*. Substitui a visualização do widget no modo de edição pela árvore de seleção de mapa.

Apresentação

Esta página descreve os componentes que podem ser usados para criar uma visualização de apresentação do widget. A visualização de apresentação do widget é a parte do widget que recebe os dados de acordo com sua configuração e os exibe no painel em um contêiner.

A exibição de apresentação consiste em três partes:

- Widget action
- Widget view
- JavaScript

Ação do widget

A classe de ação do widget (*WidgetView*) contém métodos para operações com widgets no modo de exibição de apresentação. A maioria das ações de widget usa e/ou estende a classe de controlador padrão *CControllerDashboardWidgetView*.

A classe de ação do widget deve estar localizada no diretório *actions* e especificada no parâmetro *actions* (*actions/widget.{id}.view/class*) no arquivo *manifest.json*.

exemplo do arquivo actions/WidgetView.php (implementado no widget *System information* nativo do Zabbix)**

```
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()
            ]
        ]));
    }
}
```

Visualização do widget

A classe de visualização do widget (*CWidgetView*) é responsável por criar a visualização de apresentação do widget.

A classe de visualização do widget deve estar localizada no diretório *views*. Se o arquivo que contém a classe de visualização do widget tiver um nome diferente do padrão (*widget.view.php*), ele deverá ser especificado no parâmetro do arquivo *manifest.json* *actions* (*actions/widget.{id}.view/view*).

exemplo de views/widget.view.php**

```
<?php

/**
 * Minha visualização de widget personalizada.
 *
 * @var CView $this
 * @var array $data
 */

(novo CWidgetView($data))
    ->addItem(
        novo CTag('h1', true, $data['name'])
```

```
    )
    ->show();
```

JavaScript

A classe JavaScript é responsável por determinar o comportamento do widget, como a atualização dos dados do widget, o redimensionamento do widget, a exibição de elementos do widget etc.

Todas as operações JavaScript usam e/ou estendem a classe JavaScript básica de todos os widgets do painel: *CWidget*. A classe *CWidget* contém um conjunto de métodos com a implementação padrão para o comportamento do widget. Dependendo da complexidade do widget, esses métodos podem ser utilizados como estão ou estendidos.

A classe *CWidget* contém os seguintes métodos:

- Métodos que definem o ciclo de vida do widget: *_init()*, *_registerEvents()*, *_doActivate()*, *_doDeactivate()*, *_doDestroy()*, *setEditMode()*.
- Métodos que tratam da atualização e exibição de dados do widget: *_promiseUpdate()*, *_getUpdateRequestData()*, *_processUpdateResponse(response)*, *_processUpdateErrorResponse(error)*, *_setContents()*.
- Métodos que modificam a aparência do widget: *resize()*, *_hasPadding()*.

A classe JavaScript deve estar localizada no diretório *assets/js* e especificada no parâmetro *assets* (*assets/js*) no arquivo *manifest.json*.

Métodos de ciclo de vida

Os métodos do ciclo de vida do widget são invocados pelo painel e em diferentes estágios do ciclo de vida do widget durante sua existência no painel.

O método ***onInitialize()*** define o estado inicial e/ou os valores do widget, sem realizar nenhuma manipulação de HTML ou dados. Este método é invocado quando um widget é criado (um objeto widget é instanciado), normalmente adicionando o widget a uma página do painel ou carregando a página do painel.

Exemplo:

```
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;
}
```

O método ***onStart()*** define a estrutura HTML do widget, sem realizar nenhuma manipulação de dados. Este método é invocado antes da primeira ativação da página do painel, ou seja, antes que o painel e seus widgets sejam totalmente exibidos ao usuário.

Exemplo:

```
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`
);
}
}
```

O método ***onActivate()*** torna o widget ativo e interativo, habilitando ouvintes de eventos personalizados (para responder às ações do usuário) e iniciando o ciclo de atualização do widget (para manter seu conteúdo atualizado). Este método é invocado quando a página do painel é ativada, ou seja, quando ela é totalmente exibida na interface do usuário.

Observe que, antes da invocação do método *onActivate()*, o widget está no estado inativo (`WIDGET_STATE_INACTIVE`). Após a invocação bem-sucedida, o widget transita para o estado ativo (`WIDGET_STATE_ACTIVE`). No estado ativo, o widget é responsivo, ouve eventos, atualiza seu conteúdo periodicamente e pode interagir com outros widgets.

Exemplo:

```
onActivate() {
this._startClock();

this._resize_observer = new ResizeObserver(this._events.resize);
este._resize_observer.observe(este._target);
}
```

O método ***onDeactivate()*** interrompe qualquer atividade e interatividade do widget desativando ouvintes de eventos personalizados e interrompendo o ciclo de atualização do widget. Este método é invocado quando a página do painel é desativada, ou seja, desativada ou excluída, ou quando o widget é excluído da página do painel.

Observe que, antes da invocação do método *onDeactivate()*, o widget está no estado ativo (`WIDGET_STATE_ACTIVE`). Após a invocação bem-sucedida, o widget transita para o estado inativo (`WIDGET_STATE_INACTIVE`).

Exemplo:

```
onDeactivate() {
this._stopClock();
this._resize_observer.disconnect();
}
```

O método ***onDestroy()*** executa tarefas de limpeza antes que o widget seja excluído do painel, o que pode incluir o fechamento de uma conexão com o banco de dados estabelecida durante a inicialização do widget, a limpeza de dados temporários para liberar memória do sistema e evitar vazamentos de recursos, o cancelamento do registro de ouvintes de eventos relacionados a eventos de redimensionamento ou cliques em botões para evitar manipulação desnecessária de eventos e vazamentos de memória, etc. Este método é invocado quando o widget ou a página do painel que o contém é excluído.

Observe que, antes da invocação do método *onDestroy()*, um widget em estado ativo (`WIDGET_STATE_ACTIVE`) é sempre desativado com a invocação do método *onDeactivate()*.

Exemplo:

```
onDestroy() {
if (this._filter_widget) {
this._filter_widget.off(CWidgetMap.WIDGET_NAVTREE_EVENT_MARK, this._events.mark);
este._filter_widget.off(CWidgetMap.WIDGET_NAVTREE_EVENT_SELECT, este._events.select);
}
}
```

O método ***onEdit()*** define a aparência e o comportamento do widget quando o dashboard entra no modo de edição. Este método é invocado quando o dashboard entra no modo de edição, normalmente quando um usuário interage com o botão *Editar* do widget ou com o botão *Editar dashboard* do dashboard.

Exemplo:

```
onEdit() {
    this._deactivateGraph();
}
```

Atualizar métodos de processo

Os métodos do processo de atualização do widget são responsáveis por recuperar dados atualizados do servidor Zabbix ou de qualquer outra fonte de dados e exibi-los no widget.

O método ***promiseUpdate()*** inicia o processo de atualização de dados recuperando dados, normalmente usando requisições web ou chamadas de API. Este método é invocado quando uma página de dashboard é exibida e periodicamente depois disso, até que a página do dashboard seja trocada para outra página de dashboard.

A seguir está um exemplo da implementação padrão do método *promiseUpdate()* usado pela maioria dos widgets nativos do Zabbix. Na implementação padrão, o método *promiseUpdate()* segue um padrão geral para recuperar dados do server. Ele cria um novo objeto `Curl` com a URL apropriada e parâmetros de requisição, envia uma requisição `POST` usando o método *fetch()* com o objeto de dados construído pelo método *getUpdateRequestData()*, e processa a resposta (ou uma resposta de erro) com *processUpdateResponse(response)* ou *processUpdateErrorResponse(error)*, respectivamente. Esta implementação é adequada para a maioria dos widgets, pois normalmente recuperam dados em formato JSON e os tratam de maneira consistente.

```
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);
        });
    }
```

O método ***getUpdateRequestData()*** prepara os dados da solicitação para o server para atualizar o widget, reunindo várias propriedades e seus valores correspondentes (identificadores de widget, configurações de filtro, intervalos de tempo, etc.) do estado e da configuração do widget, e construindo um objeto de dados que representa as informações necessárias a serem enviadas ao server na solicitação de atualização. Este método é invocado apenas como parte do método padrão *promiseUpdate()*, ou seja, durante o processo de atualização do widget.

Implementação padrão:

```
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
    };
}
```

O método ***processUpdateResponse(response)*** lida com a resposta recebida do server após a solicitação de atualização e, se o processo de atualização foi bem-sucedido e sem erros, limpa os dados do widget e exibe novos conteúdos com o método *setContents()*. Este método é invocado apenas como parte do método padrão *promiseUpdate()*, ou seja, durante o processo de atualização do widget.

Implementação padrão:

```
processUpdateResponse(response) {
    this._setHeaderName(response.name);

    this._updateMessages(response.messages);
    this._updateInfo(response.info);
    this._updateDebug(response.debug);

    this.setContents(response);
}
```

O método ***processUpdateErrorResponse(error)*** manipula a resposta recebida do server após a solicitação de atualização se a resposta for um erro e exibe a(s) mensagem(ns) de erro. Este método é invocado apenas como parte do método padrão

promiseUpdate(), ou seja, durante o processo de atualização do widget.

Implementação padrão:

```
processUpdateErrorResponse(error) {
    this._updateMessages(error.messages, error.title);
}
```

O método ***setContents(response)*** exibe o conteúdo do widget se o processo de atualização do widget foi bem-sucedido e sem erros, o que pode incluir manipulação de elementos DOM, atualização de componentes da interface do usuário, aplicação de estilos ou formatação, etc. Este método é invocado apenas como parte do método padrão *processUpdateResponse(response)*, ou seja, durante o processo de tratamento da resposta recebida do server após a solicitação de atualização.

Implementação padrão:

```
setContents(response) {
    this._body.innerHTML = response.body ?? '';
}
```

Métodos de modificação de apresentação

Os métodos de modificação da apresentação do widget são responsáveis por modificar a aparência do widget.

O método ***onResize()*** é responsável por ajustar os elementos visuais do widget para acomodar o novo tamanho do widget, o que pode incluir o rearranjo de elementos, ajuste das dimensões dos elementos, truncamento de texto, implementação de carregamento preguiçoso para melhorar a capacidade de resposta durante o redimensionamento, etc. Este método é invocado quando o widget é redimensionado, por exemplo, quando o usuário redimensiona manualmente o widget ou quando a janela do navegador é redimensionada.

Exemplo:

```
onResize() {
    if (this.getState() === WIDGET_STATE_ACTIVE) {
        this._startUpdating();
    }
}
```

O método ***hasPadding()*** é responsável por aplicar um padding vertical de 8px na parte inferior do widget quando ele está configurado para exibir seu cabeçalho. Este método é invocado quando a página do dashboard é ativada, ou seja, quando ela se torna a página exibida na interface do usuário.

Implementação padrão:

```
hasPadding() {
    return this.getViewMode() !== ZBX_WIDGET_VIEW_MODE_HIDDEN_HEADER;
}
```

Para alguns widgets, é necessário usar todo o espaço disponível do widget para configurar, por exemplo, uma cor de fundo personalizada. A seguir está um exemplo da implementação do método *hasPadding()* usado no widget nativo do Zabbix *Valor do item*.

```
hasPadding() {
    return false;
}
```

Tutoriais

Esta seção contém tutoriais práticos passo a passo para ilustrar como construir um módulo e um widget personalizados no Zabbix.

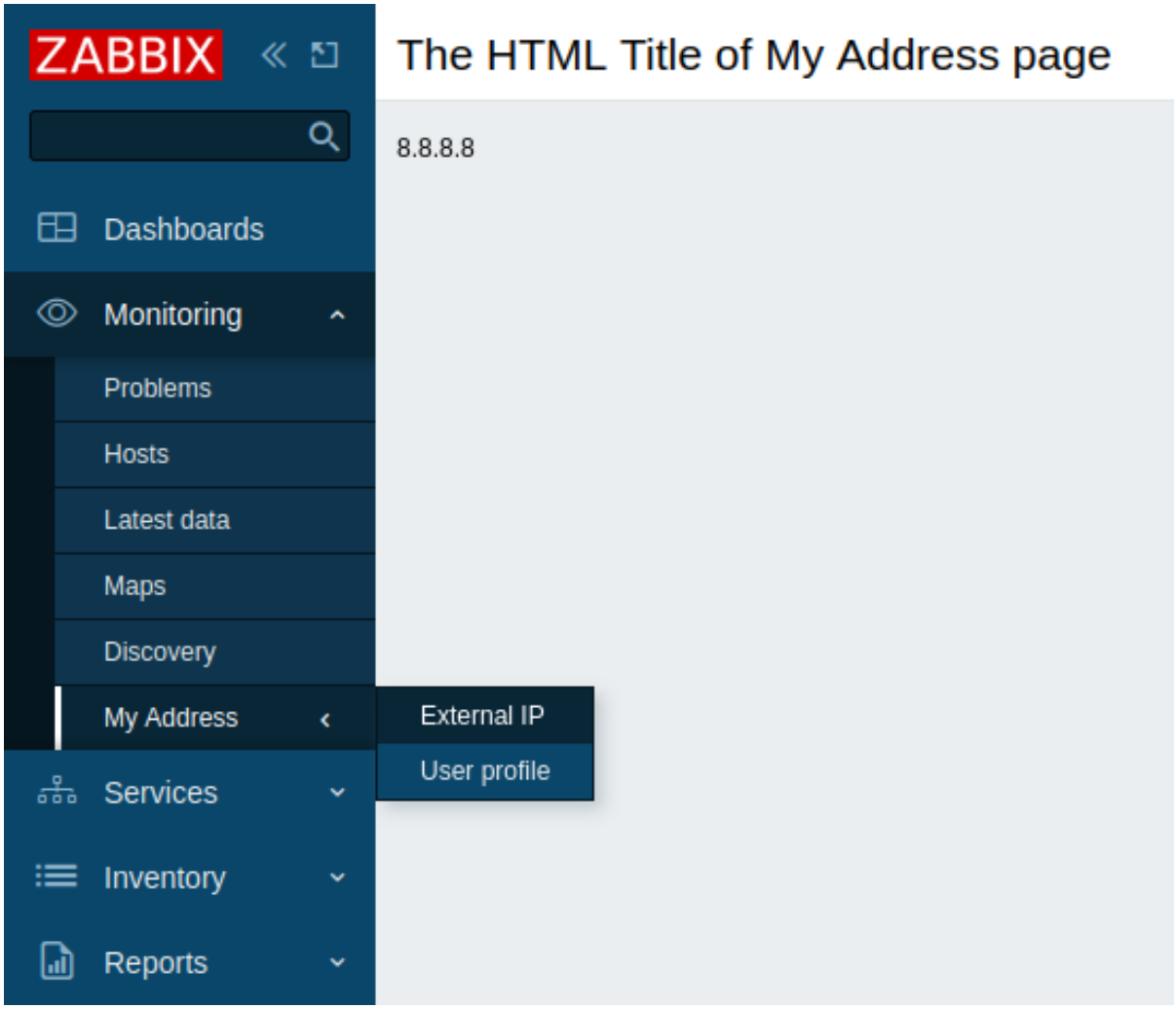
Criar um módulo (tutorial)

Este é um tutorial passo a passo que mostra como criar um módulo simples de front-end do Zabbix. Você pode fazer o download de todos os arquivos desse módulo como um arquivo ZIP: MyAddress.zip.

O que você criará

Durante este tutorial, você primeiro criará um módulo de frontend que adiciona uma nova seção de menu *My Address* e, em seguida, convertê-lo em um módulo de frontend more advanced que faz uma solicitação HTTP para *https://api.seeip.org* e exibe a

resposta - o endereço IP do seu computador - em uma nova página na recém-criada seção de menu *My Address*. Veja a seguir a aparência do módulo finalizado:



Parte I - Nova seção do menu

Adicionar um módulo em branco ao front-end do Zabbix

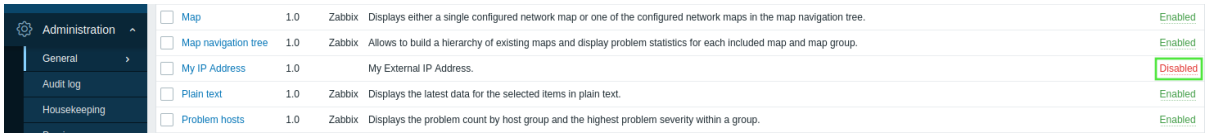
1. Crie um diretório *MyAddress* em *zabbix/ui/modules*.
2. Crie um arquivo *manifest.json* com os metadados básicos do módulo (consulte a descrição dos [parameters] suportados(../file_structure/manifest)).

ui/modules/MyAddress/manifest.json

```
{
    "manifest_version": 2.0,
    "id": "my-address",
    "name": "Meu endereço IP",
    "version" (versão): "1.0",
    "namespace": "MyAddress",
    "description" (descrição): "My External IP Address" (Meu endereço IP externo)
}
```

3. No front-end do Zabbix, vá para a seção *Administration → General → Modules* e clique no botão *Scan directory*.

4. Localize o novo módulo *My IP Address* na lista e clique no hiperlink ”Disabled” (Desativado) para alterar o status do módulo de ”Disabled” (Desativado) para ”Enabled” (Ativado).



O módulo agora está registrado no frontend. Entretanto, ele ainda não está visível, pois você ainda precisa definir a funcionalidade do módulo. Depois de adicionar conteúdo ao diretório do módulo, você verá imediatamente as alterações no frontend do Zabbix ao atualizar a página.

Criar uma seção de menu

1. Crie um arquivo *Module.php* no diretório *MyAddress*.

Esse arquivo implementa uma nova classe *Module* que estende a classe padrão *CModule*. A classe *Module* inserirá uma nova seção de menu *My Address* no menu principal.

O método *setAction()* especifica uma ação a ser executada ao clicar na seção do menu. Para começar, você pode usar a ação predefinida *userprofile.edit*, que abrirá a página *User profile*. Na part III deste tutorial, você aprenderá a criar uma ação personalizada.

ui/modules/MyAddress/Module.php

```
<?php

namespace Modules\MyAddress;

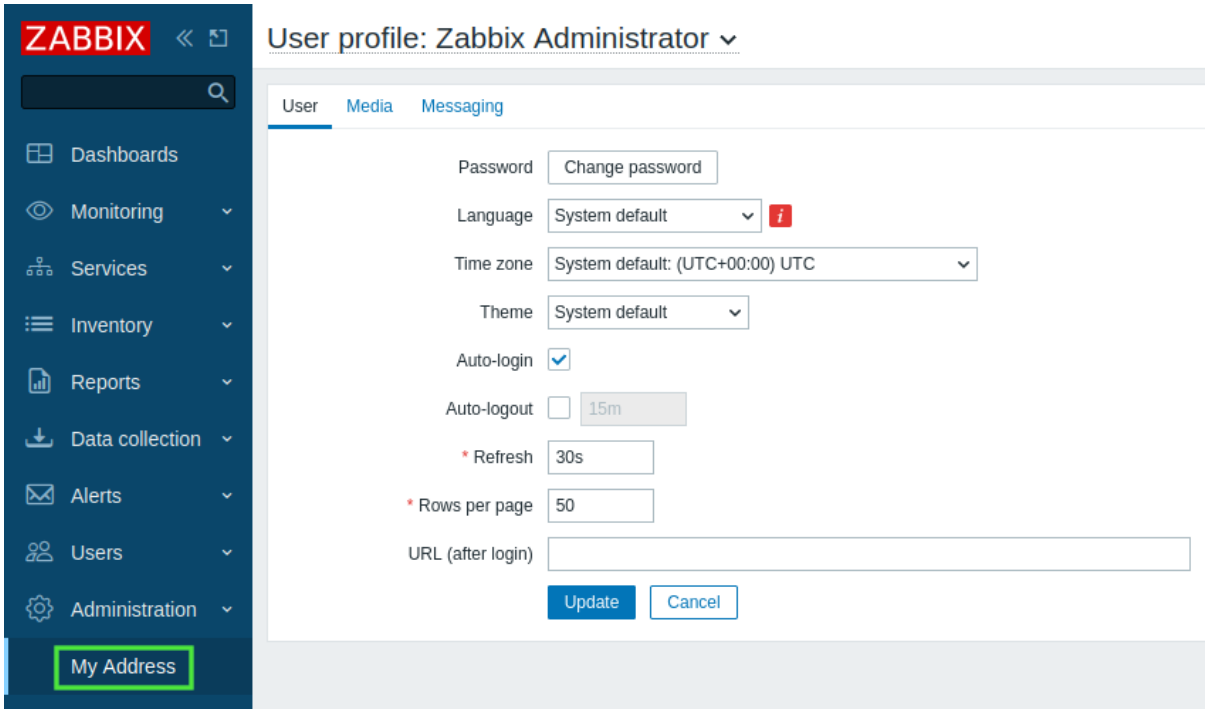
use Zabbix\Core\CModule,
    APP,
    CMenuItem;

class Module extends CModule {

    public function init(): void {
        APP::Component()->get('menu.main')
            ->add((new CMenuItem(_('My Address')))
            ->setAction('userprofile.edit'));
    }
}
```

Note:
Você pode substituir `'userprofile.edit'` por outras ações, por exemplo, `'charts.view'` (opens custom graphs), `'problems.view'` (opens *Monitoring* → *Problems*) ou `'report.status'` (opens *System information* report).

3. Atualize o front-end do Zabbix. Agora há uma nova seção *My Address* na parte inferior do menu principal do Zabbix. Clique em *My Address* para abrir a página *User profile*.



Parte II - Alteração do local da seção do menu

Nesta parte, você moverá a seção de menu *Meu Endereço* para a seção *Monitoramento* e adicionar um submenu a ela. Como resultado, os usuários poderão acessar duas páginas de submenu na seção de menu *Monitoramento* → *Meu Endereço*.

1. Abra e edite o arquivo *Module.php*.

ui/modules/MyAddress/Module.php

```
<?php

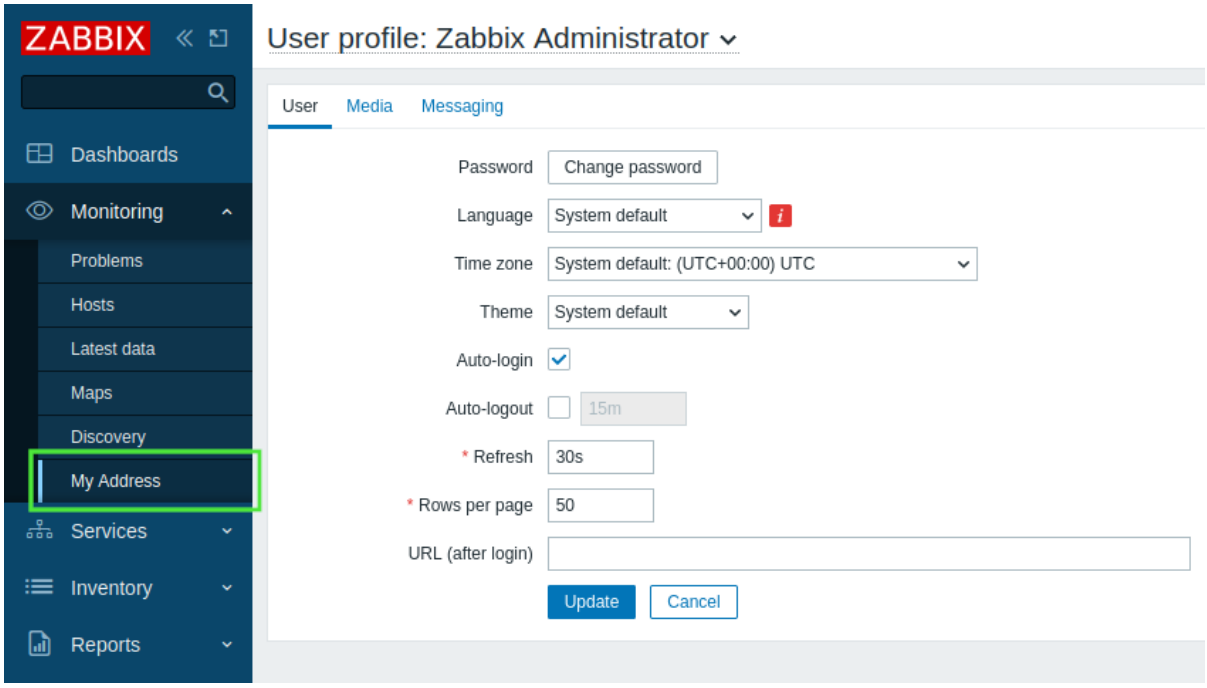
namespace Modules\MyAddress;

use Zabbix\Core\CModule,
    APP,
    CMenuItem;

class Module extends CModule {

   public function init(): void {
        APP::Component()->get('menu.main')
            ->findOrAdd(_('Monitoramento'))
                ->getSubmenu()
                    ->insertAfter(_('Descoberta'),
                        (new CMenuItem(_('Meu Endereço')))->setAction('userprofile.edit')
                    );
    }
}
```

2. Atualize o frontend do Zabbix. Expanda a seção do menu *Monitoramento* e observe que a seção *Meu Endereço* agora está localizada abaixo da seção *Descoberta*.



3. Para adicionar paginas abaixo da seção do menu *Meu Endereço*, abra e edite o arquivo *Module.php* novamente.

Esse passo cria duas subseções:

-*IP Externo* que executa uma nova ação 'my.address', que será definida nos próximos passos; -*Perfil de Usuário* que executa a ação prédefinida 'userprofile.edit' para abrir a pagina *Perfil de usuário*.

Note que para submenus, voce precisa usar a classe *CMenu* além das classes utilizadas em passos anteriores.

ui/modules/MyAddress/Module.php

```
<?php

namespace Modules\MyAddress;

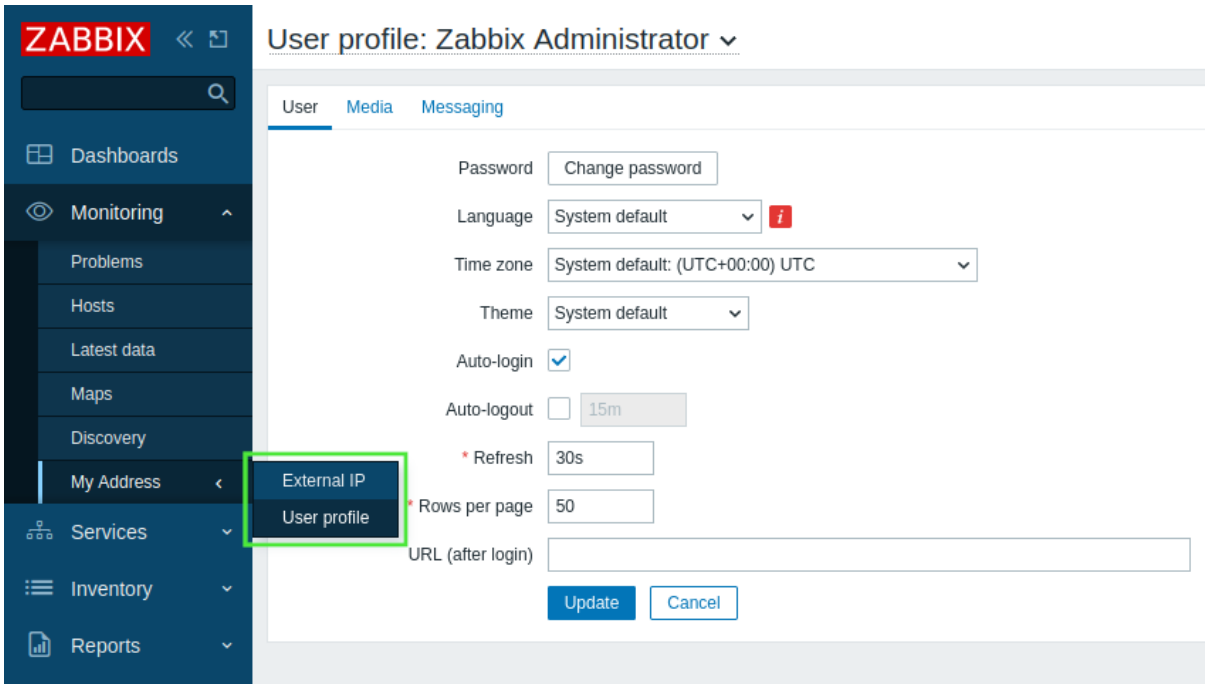
use Zabbix\Core\CModule,
```

```
APP,
CMenu,
CMenuItem;

class Module extends CModule {

public function init(): void {
APP::Component()->get('menu.main')
->findOrAdd(_('Monitoramento'))
->getSubmenu()
->insertAfter(_('Descoberta'),
(new CMenuItem(_('Meu Endereço')))->setSubMenu(
new CMenu([
(new CMenuItem(_('IP Externo')))->setAction('my.address'),
(new CMenuItem(_('Perfil de Usuário')))->setAction('userprofile.edit')
])
)
);
}
}
```

4. Atualize o frontend do Zabbix. Observe que a seção do menu *Meu Endereço* agora contem um menu de terceiro nível, com duas páginas - *IP Externo* and *Perfil de Usuário*.



Parte III - Módulo de ação

Uma ação é implementada em dois arquivos - *actions/MyAddress.php* e *views/my.address.php*. O arquivo ***actions/MyAddress.php*** cuida da implementação da lógica de negócio, enquanto o arquivo ***views/my.address.php*** é responsável pela visualização.

1. Crie um diretório *actions* no diretório *MyAddress*.
2. Crie um arquivo *MyAddress.php* no diretório *actions*.

A lógica da ação será definida na classe *MyAddress*. Essa classe de ação implementará quatro funções: *init()*, *checkInput()*, *checkPermissions()*, e *doAction()*. A interface web do Zabbix chama a função *doAction()* quando a ação é solicitada. Essa função é responsável pela lógica de negócio do módulo.

Attention:
Os dados devem ser organizados como um sequência associativa. A sequência pode ser multidimensional e pode conter quaisquer dados esperados pela exibição.

ui/modules/MyAddress/actions/MyAddress.php

```php
<?php

namespace Modules\MyAddress\Actions;

use CController,
    CControllerResponseData;

class MyAddress extends CController {

   public function init(): void {
   $this->disableCsrfValidation();
   }

   protected function checkInput(): bool {
   return true;
   }

   protected function checkPermissions(): bool {
   return true;
   }

   protected function doAction(): void {
   $data = ['my-ip' => file_get_contents("https://api.seeip.org")];
   $response = new CControllerResponseData($data);
   $this->setResponse($response);
   }
}
```

3. Crie um novo diretório *views* no diretório *MyAddress*.
4. Crie um arquivo *my.address.php* no diretório *views* e defina a visualização do módulo.

Observe que a variável $data está disponível na visualização sem que ela seja especificamente definida. A estrutura passa automaticamente a sequência associativa para a visualização.

ui/modules/MyAddress/views/my.address.php

```php
<?php

(new CHtmlPage())
    ->setTitle(_('O título HTML da minha Página de Endereço'))
    ->addItem(new CDiv($data['my-ip']))
    ->show();
```

5. A ação do módulo deve ser registrada no arquivo *manifest.json*. Abra o arquivo *manifest.json* e adicione um novo objeto `actions` que contenha:

- a chave da ação com o nome da ação escrito em letras minúsculas (a-z) e com palavras separadas por pontos (por exemplo, `my.address`);
- o nome da classe da ação (`MyAddress`) como um valor para a chave `class` do objeto `my.address`;
- o nome da visualização da ação (`my.address`) como um valor para a chave `view` do objeto `my.address`.

ui/modules/MyAddress/manifest.json

```
{
    "manifest_version": 2.0,
    "id": "my-address",
    "name": "Meu endereço IP",
    "version" (versão): "1.0",
    "namespace": "MyAddress",
    "description" (descrição): "Meu endereço IP externo",
    "actions": {
    "my.address": {
    "class": "MyAddress",
    "view": "my.address"
    }
```

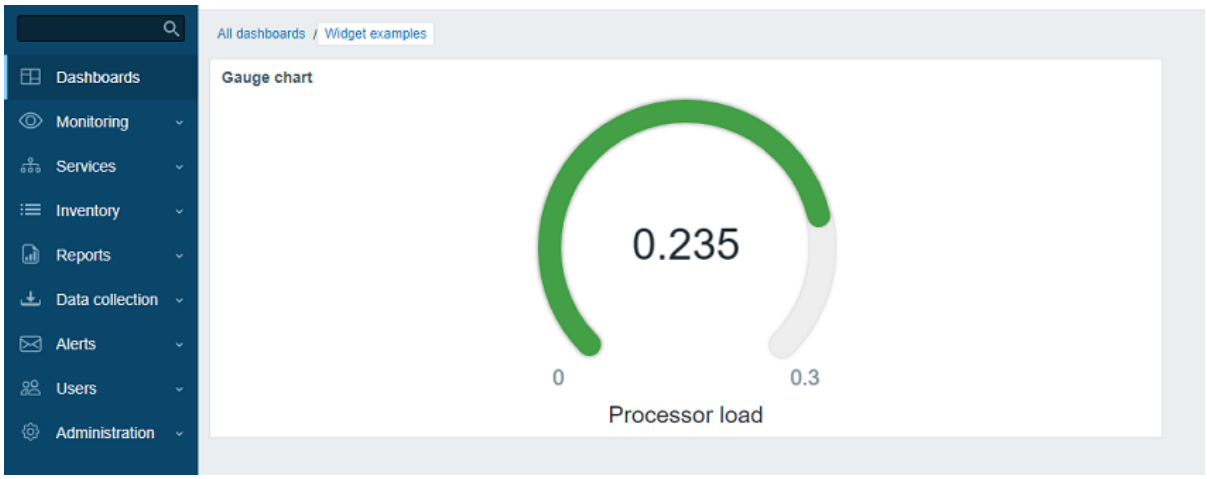
```
    }
}
```

7. Atualize a interface web do Zabbix. Clique na seção do menu *My address* → *External IP* para ver o endereço IP do seu computador.

Criar um widget (tutorial)

Aqui está um tutorial passo a passo que mostra como criar um widget de dashboard simples. Você pode baixar todos os arquivos deste widget como um arquivo ZIP: [lesson_gauge_chart.zip] (../../../../assets/en/devel/modules/examples/lesson_gauge_chart.zip).

O que você vai construir Durante este tutorial, você construirá primeiro um widget básico "Olá, mundo!" e, em seguida, o converterá em um widget mais avançado que exibe o valor de um item como um gráfico de medidor. Veja como o widget final ficará:



Parte I - "Olá, mundo!" Nesta seção você aprenderá como criar os elementos mínimos necessários do widget e adicionar um novo widget ao frontend do Zabbix.

Adicionar um widget em branco ao frontend do Zabbix

1. Crie um diretório *lesson_gauge_chart* no diretório *modules* da instalação do frontend do Zabbix (por exemplo, *zabbix/ui/modules*).

Note:
Todos os widgets personalizados são tratados como módulos externos e devem ser adicionados ao diretório *modules* da instalação do frontend do Zabbix (por exemplo, *zabbix/ui/modules*). O diretório *zabbix/ui/widgets* é reservado para widgets nativos do Zabbix e é atualizado junto com a interface do Zabbix.

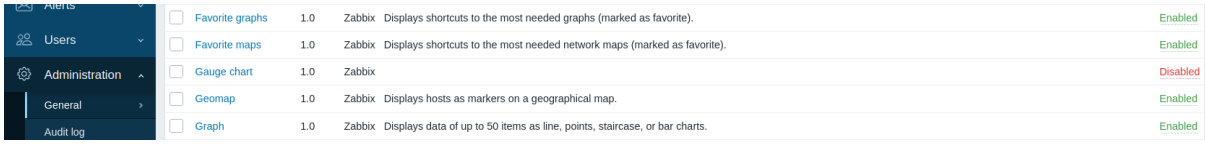
2. Crie um arquivo *manifest.json* com os metadados básicos do widget (veja a descrição dos parâmetros suportados).

ui/modules/lesson_gauge_chart/manifest.json

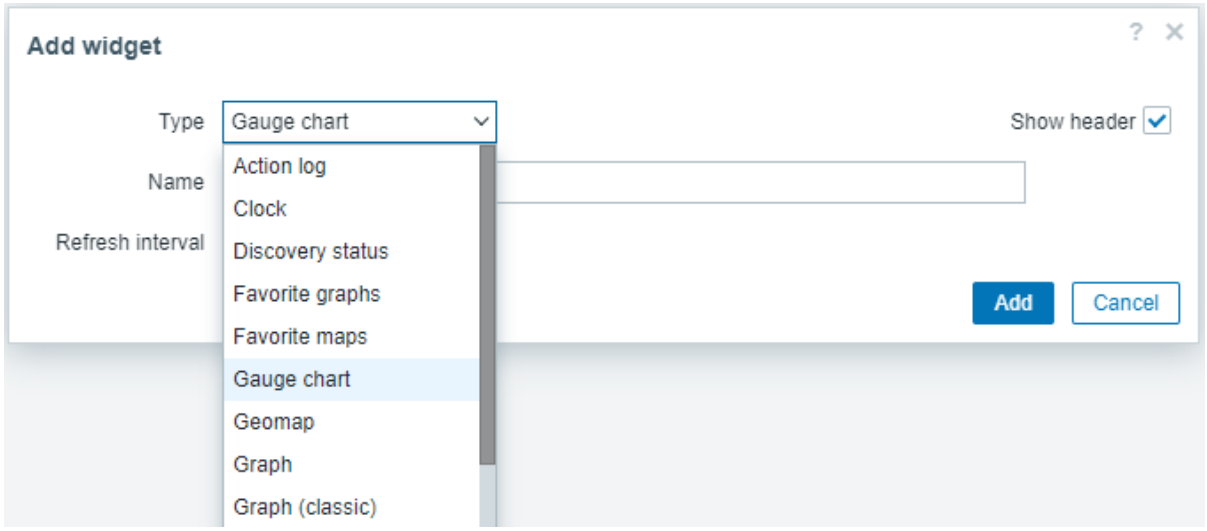
```
{
    "manifest_version": 2.0,
    "id": "lesson_gauge_chart",
    "type": "widget",
    "name": "Gauge chart",
    "namespace": "LessonGaugeChart",
    "version": "1.1",
    "author": "Zabbix"
}
```

3. No frontend do Zabbix, vá para a seção *Administração* → *Geral* → *Módulos* e clique no botão *Procurar diretório*.

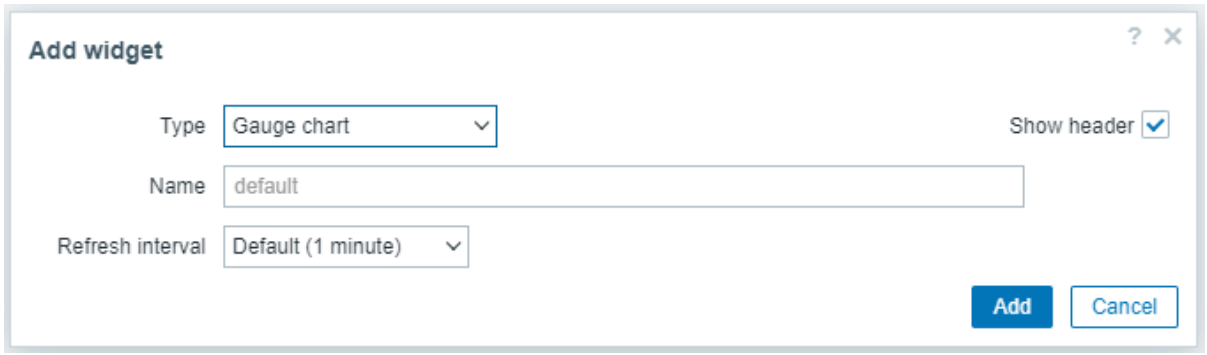
4. Localize o novo módulo *Gauge chart* na lista e clique no hiperlink "Desabilitado" para alterar o status do módulo de "Desabilitado" para "Habilitado" (se o módulo não estiver listado, consulte a seção de solução de problemas).



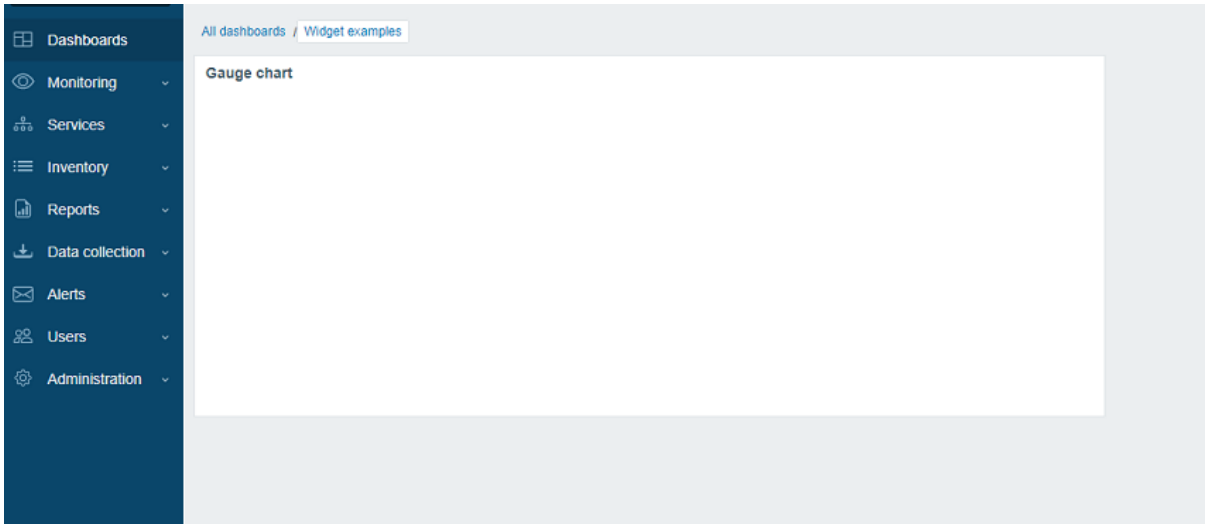
5. Abra um dashboard, alterne para o modo de edição e adicione um novo widget. No campo *Tipo*, selecione ”Gauge chart”.



6. Neste ponto, a configuração do widget *Gauge chart* contém apenas os campos comuns do widget *Nome* e *Intervalo de atualização*. Clique em *Adicionar* para adicionar o widget ao dashboard.



7. Um widget em branco deve aparecer no dashboard. Clique em *Salvar alterações* no canto superior direito para salvar o dashboard.



Adicionar uma visualização de widget

Note:
O arquivo **view** do widget deve estar localizado no diretório *views* (para este tutorial, *ui/modules/lesson_gauge_chart/views/*). Se o arquivo tiver o nome padrão *widget.view.php*, você não precisa registrá-lo no arquivo *manifest.json*. Se o arquivo tiver um nome diferente, especifique-o na seção *actions/widget.lesson_gauge_chart.view* do arquivo manifest.json.

1. Crie um diretório *views* no diretório *lesson_gauge_chart*.
2. Crie um arquivo *widget.view.php* no diretório *views*.

ui/modules/lesson_gauge_chart/views/widget.view.php

```
<?php

/**
 * Visualização do widget de gráfico de gauge.
 *
 * @var CView $this
 * @var array $data
 */

(new CWidgetView($data))
    ->addItem(
        new CTag('h1', true, 'Hello, world!')
    )
    ->show();
```

3. Atualize o dashboard. O widget *Gauge chart* agora exibe "Hello, world!".

Parte II - Gráfico de medidor Adicione configurações a uma visualização de configuração e use-as em uma visualização de widget

Nesta seção, você aprenderá como adicionar um campo de configuração de widget e mostrar o valor inserido na visualização do widget como texto.

A configuração do widget consiste em um formulário (*Zabbix\Widgets\CWidgetForm*) e uma visualização de formulário de widget (*widget.edit.php*). Para adicionar campos (*Zabbix\Widgets\CWidgetField*), você precisa criar uma classe *WidgetForm*, que irá estender *Zabbix\Widgets\CWidgetForm*.

O formulário contém o conjunto de campos (*Zabbix\Widgets\CWidgetField*) de vários tipos, que são usados para validar os valores inseridos pelo usuário. O campo do formulário (*Zabbix\Widgets\CWidgetField*) para cada tipo de elemento de entrada converte o valor em um único formato para armazená-lo no banco de dados.

Note:
O arquivo **form** do widget deve estar localizado no diretório *includes* (para este tutorial, *ui/modules/lesson_gauge_chart/includes/*). Se o arquivo tiver o nome padrão *WidgetForm.php*, você não precisa registrá-lo no arquivo *manifest.json*. Se o arquivo tiver um nome diferente, especifique-o na seção *widget/form_class* do arquivo manifest.json.

1. Crie um novo diretório *includes* no diretório *lesson_gauge_chart*.
2. Crie um arquivo *WidgetForm.php* no diretório *includes*.

ui/modules/lesson_gauge_chart/includes/WidgetForm.php

```
<?php

namespace Modules\LessonGaugeChart\Includes;

use Zabbix\Widgets\CWidgetForm;

class WidgetForm extends CWidgetForm {
}
```

3. Adicione um campo *Description* ao formulário de configuração do widget. Este é um campo de texto comum, onde o usuário pode inserir qualquer conjunto de caracteres. Você pode usar a classe *CWidgetFieldTextBox* para isso.

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. No diretório *views*, crie um arquivo de visualização de configuração de widget *widget.edit.php* e adicione uma visualização para o novo campo *Description*. Para a classe de campo *CWidgetFieldTextBox*, a visualização é *CWidgetFieldTextBoxView*.

ui/modules/lesson_gauge_chart/views/widget.edit.php

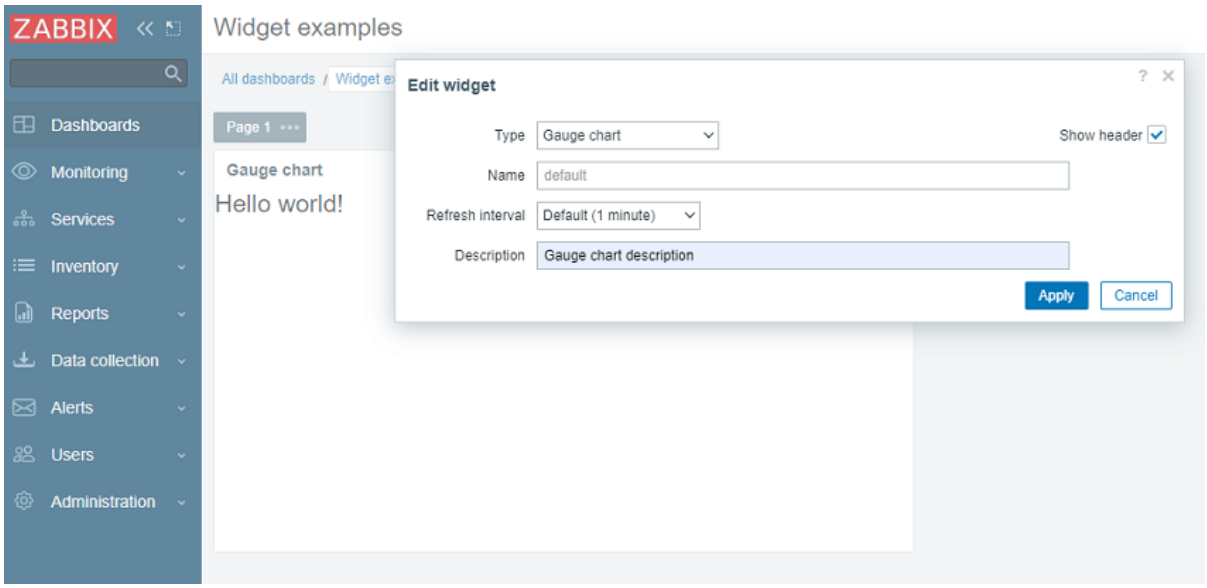
```
<?php

/**
 * Gauge chart widget form view.
 *
 * @var CView $this
 * @var array $data
 */

(new CWidgetFormView($data))
    ->addField(
        new CWidgetFieldTextBoxView($data['fields']['description'])
    )
    ->show();
```

5. Vá para o dashboard e clique no ícone de engrenagem no widget para abrir o formulário de configuração do widget.
6. O formulário de configuração do widget agora contém um novo campo de texto *Description*. Digite qualquer valor, por

exemplo, *Gauge chart description*.



7. Clique em *Apply* no formulário de configuração do widget. Em seguida, clique em *Save changes* no canto superior direito para salvar o dashboard. Observe que a nova descrição não está visível em nenhum lugar, e o widget ainda exibe "Hello, world!".

Para que a nova descrição apareça no widget, o valor do campo *Description* precisa ser recuperado do banco de dados e passado para a visualização do widget. Para isso, você precisa criar uma classe de ação.

8. Crie um novo diretório *actions* no diretório *lesson_gauge_chart*.
9. Crie um arquivo *WidgetView.php* no diretório *actions*. A classe de ação *WidgetView* irá estender a classe *CControllerDashboardWidgetView*.

Os valores dos campos de configuração do widget são armazenados na propriedade **$fields_values** da classe de ação.

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. Abra o *manifest.json* e registre *WidgetView* como uma classe de ação na seção *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. Agora você pode usar o valor do campo de descrição, contido em *$data['description']*, na visualização do widget. Abra *views/widget.view.php* e substitua o texto estático "Hello, world!" por *$data['description']*.

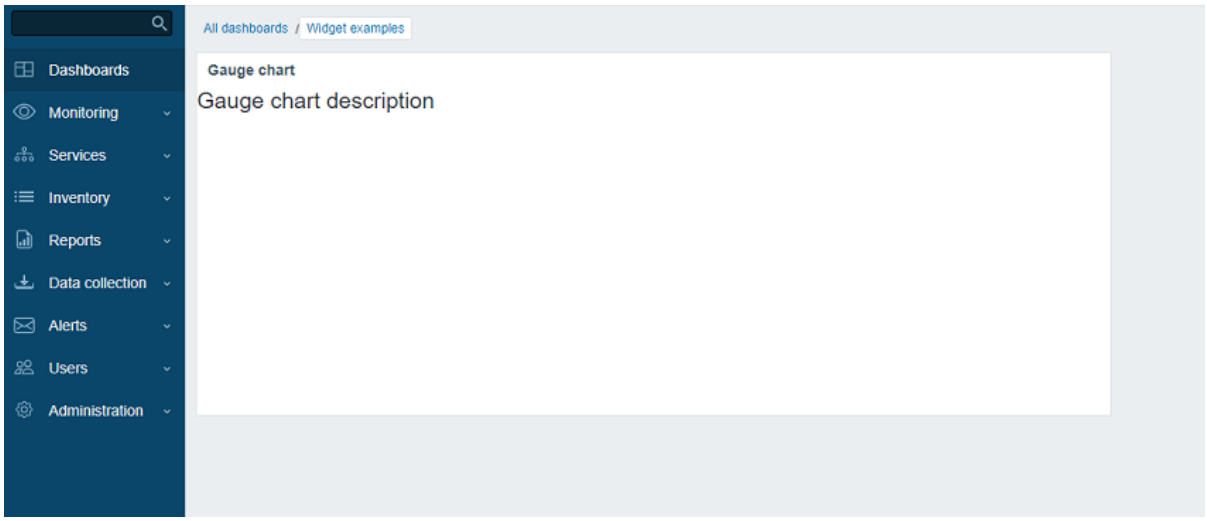
ui/modules/lesson_gauge_chart/views/widget.view.php

```
<?php

/**
 * Gauge chart widget view.
 *
 * @var CView $this
 * @var array $data
 */

(new CWidgetView($data))
    ->addItem(
        new CTag('h1', true, $data['description'])
    )
    ->show();
```

12. Atualize a página do dashboard. Agora você deve ver o texto da descrição do widget em vez de "Hello, world!".



Recuperar o valor de um item via API

O widget deve mostrar o último valor de um item à escolha do usuário. Para isso, é necessário adicionar a capacidade de selecionar items na configuração do widget.

Nesta seção, você aprenderá como adicionar um campo de seleção de item ao formulário do widget e como adicionar a parte visual desse campo à visualização de configuração. Em seguida, o controlador do widget poderá recuperar os dados do item e seu valor por meio de uma solicitação de API. Uma vez recebido, o valor pode ser exibido na visualização do widget.

1. Abra *includes/WidgetForm.php* e adicione o campo *CWidgetFieldMultiSelectItem*. Isso permitirá selecionar um item no formulário de configuração.

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. Abra *views/widget.edit.php* e adicione o componente visual do campo à visualização de configuração.

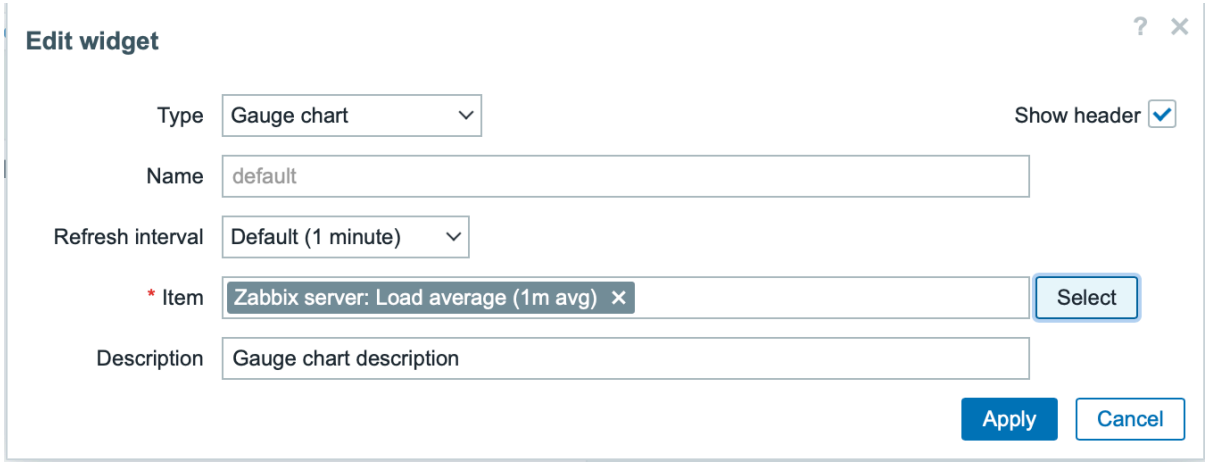
ui/modules/lesson_gauge_chart/views/widget.edit.php

```
<?php

/**
 * Gauge chart widget form view.
 *
 * @var CView $this
 * @var array $data
 */

(new CWidgetFormView($data))
    ->addField(
        new CWidgetFieldMultiSelectItemView($data['fields']['itemid'])
    )
    ->addField(
        new CWidgetFieldTextBoxView($data['fields']['description'])
    )
    ->show();
```

3. Volte ao dashboard e clique no ícone de engrenagem no widget para abrir o formulário de configuração do widget.
4. O formulário de configuração do widget agora contém um novo campo de entrada *Item*. Selecione o host "Zabbix server" e o item "Load average (1m avg)".



5. Clique em *Aplicar* no formulário de configuração do widget. Em seguida, clique em *Salvar alterações* no canto superior direito para salvar o dashboard.
6. Abra e modifique *actions/WidgetView.php*.

A partir de agora, o ID do item estará disponível no controlador do widget em *$this->fields_values['itemid']*. O método do controlador *doAction()* coleta os dados do item (nome, tipo de valor, unidades) usando o método da API *item.get* e o último valor do item usando o método da API *history.get*.

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]
            ]
        ]);

        $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. Abra *views/widget.view.php* e adicione o valor do item à visualização do widget.

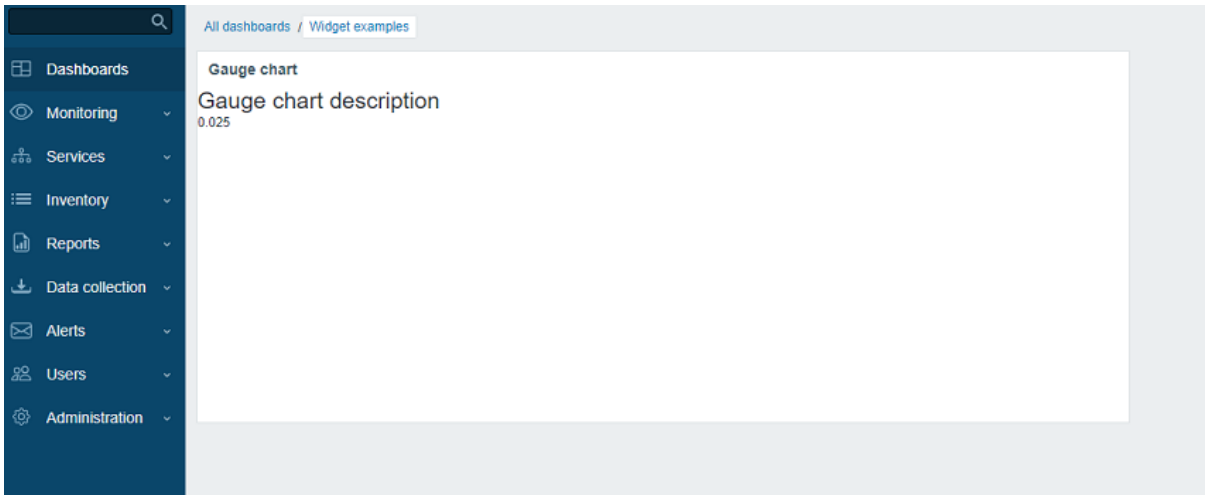
ui/modules/lesson_gauge_chart/views/widget.view.php

```
<?php

/**
 * Gauge chart widget view.
 *
 * @var CView $this
 * @var array $data
 */

(new CWidgetView($data))
    ->addItem([
        new CTag('h1', true, $data['description']),
        new CDiv($data['value'] !== null ? $data['value']['value'] : _('No data'))
    ])
    ->show();
```

8. Atualize a página do dashboard. O widget exibirá o valor mais recente do item.



Adicionar configurações avançados à view de configuração

Nesta seção, você aprenderá a como adicionar uma seção de *Configurações avançadas* expansível com parâmetros opcionais, tais como cor, valores mínimos e máximos, unidades e o campo *Descrição* criado anteriormente.

1. Crie um arquivo *Widget.php* no diretório principal do widget *lesson_gauge_chart* para criar uma nova classe *Widget*.

A classe *Widget* estenderá a classe base *CWidget* para adicionar/sobrepor as configurações padrão do widget (neste caso - traduções). @@ JavaScript, provided below, displays the string ”No data” in case of missing data. The ”No data” string is present in the Zabbix UI translation files.

If there are any widget constants, it is recommended to also specify them in the *Widget* class.

ui/modules/lesson_gauge_chart/Widget.php

```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. Abra o arquivo *includes/WidgetForm.php* e adicione os novos campos *Cor* (color picker), *Min* (numeric field), *Max* (numeric field), e *Unidades* (select), e defina a paleta de cor padrão para o seletor de cor, de modo que possa ser usado nos próximos passos.

ui/modules/lesson_gauge_chart/includes/WidgetForm.php

```php
<?php

namespace Modules\LessonGaugeChart\Includes;

use Modules\LessonGaugeChart\Widget;

use Zabbix\Widgets\{
    CWidgetField,
    CWidgetForm
};

use Zabbix\Widgets\Fields\{
    CWidgetFieldColor,
    CWidgetFieldMultiSelectItem,
    CWidgetFieldNumericBox,
    CWidgetFieldSelect,
    CWidgetFieldTextBox
};

/**
 * Formulário do widget de gráfico de medidor.
 */
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', _('Cor')))->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', _('Unidades'), [
                    Widget::UNIT_AUTO => _x('Auto', 'seleção por histórico'),
                    Widget::UNIT_STATIC => _x('Static', 'seleção por histórico')
                ]))->setDefault(Widget::UNIT_AUTO)
            )
            ->addField(
                (new CWidgetFieldTextBox('value_static_units'))
            )
            ->addField(
                new CWidgetFieldTextBox('description', _('Descrição'))
            );
    }
}
```

3. Abra o arquivo *views/widget.edit.php* e adicione os componentes de campo visual à view de configuração.

ui/modules/lesson_gauge_chart/views/widget.edit.php

```
<?php

/**
 * View de formulário do widget de gráfico de medidor.
 *
 * @var CView $this
 * @var array $data
 */

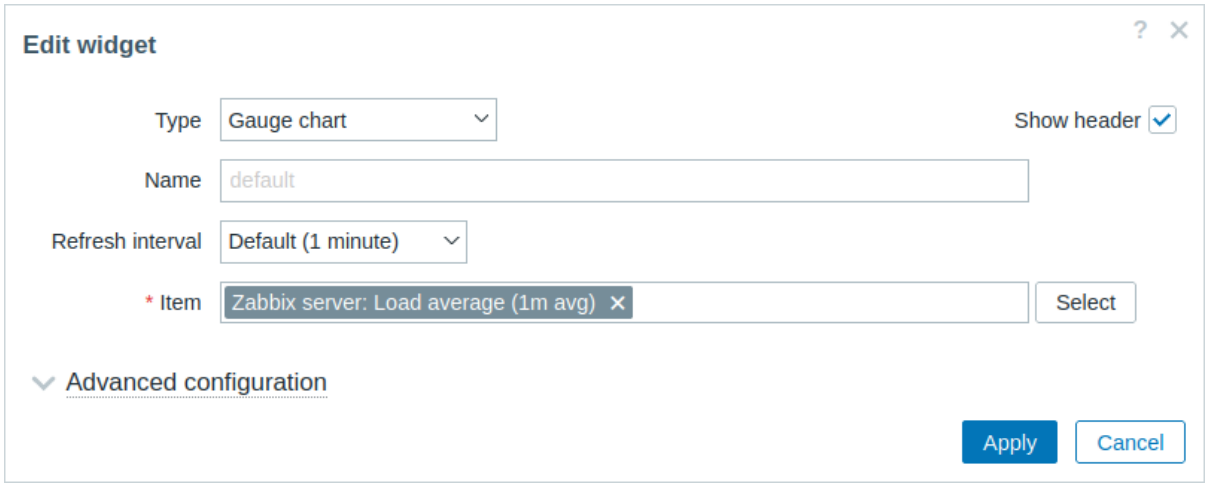
$lefty_units = new CWidgetFieldSelectView($data['fields']['value_units']);
$lefty_static_units = (new CWidgetFieldTextBoxView($data['fields']['value_static_units']))
    ->setPlaceholder(_('value'))
    ->setWidth(ZBX_TEXTAREA_TINY_WIDTH);

(new CWidgetFormView($data))
    ->addField(
        (new CWidgetFieldMultiSelectItemView($data['fields']['itemid']))
            ->setPopupParameter('numeric', true)
    )
    ->addFieldset(
        (new CWidgetFormFieldsetCollapsibleView(_('Configuração avançada')))
            ->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),
```

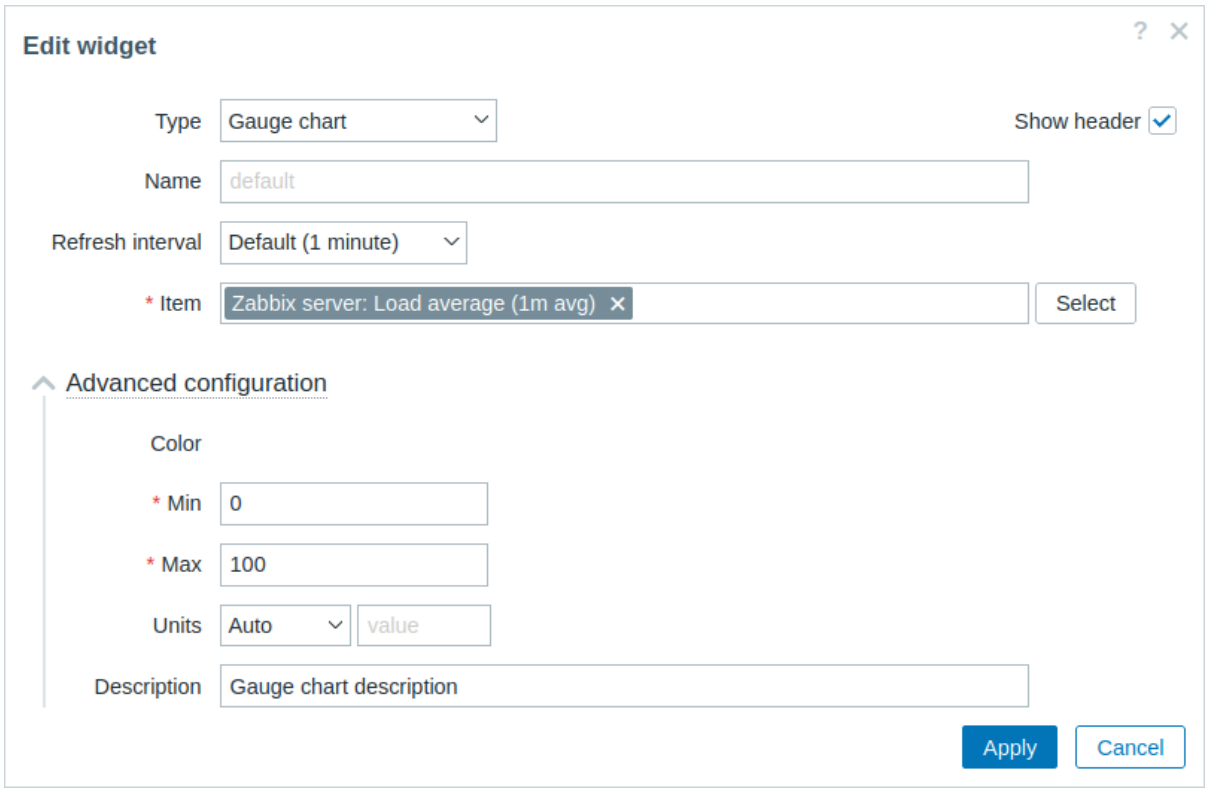
```
                    $lefty_static_units->getView()
                ]))
            ])
            ->addField(
                new CWidgetFieldTextBoxView($data['fields']['description'])
            )
    )
    ->show();
```

Note:
O método *addField()* da classe *CWidgetFormView* recebe uma classe CSS de string como segundo parâmetro.

4. Retorne ao painel, altere para o modo de edição e clique no ícone de engrenagem do widget para abrir o formulário de configuração. O formulário de configuração do widget agora contém uma nova seção expansível de *Configuração avançada*.



5. Expanda a seção *Configuração avançada* para ver os campos adicionais de configuração do widget. Note que o campo *Cor* ainda não tem seletor de cor. Isso ocorre porque o seletor de cor deve ser iniciado com JavaScript, o que será adicionado no próximo passo - *Adicionar JavaScript ao widget*.



Adicionar JavaScript ao widget

Nesta seção, você aprenderá como adicionar um gráfico de gauge - feito usando JavaScript - que mostra se o valor mais recente está normal ou muito alto/baixo.

1. Crie um arquivo *widget.edit.js.php* no diretório *views*.

O JavaScript será responsável por inicializar o seletor de cores na visualização de configuração.

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. Abra *views/widget.edit.php* e adicione o arquivo *widget.edit.js.php* com o JavaScript à visualização de configuração. Para isso, use o método *includeJsFile()*. Para adicionar JavaScript inline, use o método *addJavaScript()*.

ui/modules/lesson_gauge_chart/views/widget.edit.php

```
<?php

/**
 * Gauge chart widget form view.
 *
 * @var CView $this
 * @var array $data
 */

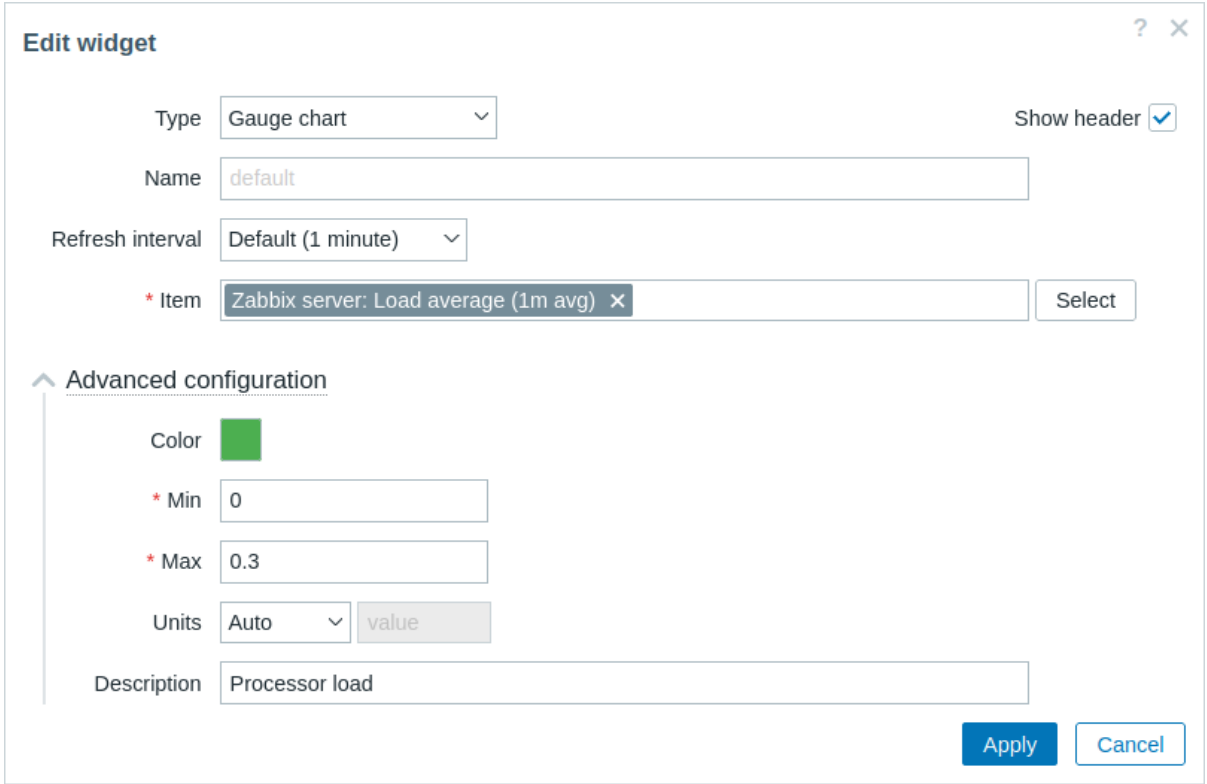
use Modules\LessonGaugeChart\Includes\WidgetForm;

$lefty_units = new CWidgetFieldSelectView($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. Volte ao dashboard, clique no ícone de engrenagem no widget para abrir o formulário de configuração do widget. Agora, expanda a seção *Advanced configuration* para ver o seletor de cores inicializado. Preencha os campos com valores e selecione uma cor para o gráfico de gauge.



4. Clique em *Apply* no formulário de configuração do widget. Depois clique em *Save changes* no canto superior direito para salvar o dashboard.

5. Abra *actions/WidgetView.php* e atualize o controller.

A propriedade **$this->fields_values** agora contém os valores de todos os campos de *Advanced configuration*. Finalize o controller para permitir passar a configuração e o valor do item selecionado para a visualização do widget.

ui/modules/lesson_gauge_chart/actions/WidgetView.php

```
<?php

namespace Modules\LessonGaugeChart\Actions;

use API,
    CControllerDashboardWidgetView,
    CControllerResponseData;

class WidgetView extends CControllerDashboardWidgetView {

    protected function doAction(): void {
        $db_items = API::Item()->get([
            'output' => ['itemid', 'value_type', 'name', 'units'],
            'itemids' => $this->fields_values['itemid'],
            'webitems' => true,
            'filter' => [
                'value_type' => [ITEM_VALUE_TYPE_UINT64, ITEM_VALUE_TYPE_FLOAT]
            ]
        ]);

        $history_value = null;

        if ($db_items) {
            $item = $db_items[0];

            $history = API::History()->get([
                'output' => API_OUTPUT_EXTEND,
                'itemids' => $item['itemid'],
                'history' => $item['value_type'],
                'sortfield' => 'clock',
                'sortorder' => ZBX_SORT_DOWN,
                'limit' => 1
            ]);

            if ($history) {
                $history_value = convertUnitsRaw([
                    'value' => $history[0]['value'],
                    'units' => $item['units']
                ]);
            }
        }

        $this->setResponse(new CControllerResponseData([
            'name' => $this->getInput('name', $this->widget->getName()),
            'history' => $history_value,
            'fields_values' => $this->fields_values,
            'user' => [
                'debug_mode' => $this->getDebugMode()
            ]
        ]));
    }
}
```

6. Abra e modifique *views/widget.view.php*.

Você precisa criar um container para o gráfico de gauge, que será desenhado nos próximos passos, e um container para a descrição.

Para passar valores para o JavaScript como um objeto JSON, use o método *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. Crie um novo diretório *assets* no diretório *lesson_gauge_chart*. Este diretório será usado para armazenar JavaScript, CSS e, potencialmente, quaisquer outros assets, como fontes ou imagens.
8. Para o JavaScript da visualização do widget, crie um diretório *js* no diretório *assets*.
9. Crie um arquivo *class.widget.js* no diretório *assets/js*.

Esta classe JavaScript do widget irá estender a classe base JavaScript de todos os widgets do dashboard - *CWidget*.

O dashboard depende de uma implementação correta de um widget e comunica qualquer informação relevante ao widget por meio da chamada dos respectivos métodos JavaScript. O dashboard também espera que o widget gere eventos quando alguma interação ocorrer. Assim, a classe *CWidget* contém um conjunto de métodos com a implementação padrão do comportamento do widget, que pode ser customizado estendendo a classe.

Neste caso, alguma customização é necessária, portanto, a lógica customizada será implementada para o seguinte comportamento do widget:

- inicialização do widget, que é responsável por definir o estado inicial do widget (veja o método *onInitialize()*);
- exibição do conteúdo do widget (ou seja, desenhar o gráfico de gauge) se o processo de atualização do widget foi bem-sucedido e sem erros (veja o método *processUpdateResponse(response)* e os métodos relacionados *_resizeChart()* e *_updatedChart()*)
- redimensionamento do widget (veja o método *onResize()* e o método relacionado *_resizeChart()*)

Para outros aspectos do widget de gráfico de gauge, será usada a implementação padrão do comportamento do widget. Para saber mais sobre os métodos JavaScript da classe *CWidget*, veja: JavaScript.

Como este JavaScript é necessário para a visualização do widget, ele deve ser carregado junto com a página do dashboard. Para habilitar o carregamento do JavaScript, você precisará atualizar os parâmetros *assets/js* e *js_class* no arquivo ***manifest.json*** conforme mostrado no passo 10.

ui/modules/lesson_gauge_chart/assets/js/class.widget.js

```
class WidgetLessonGaugeChart extends CWidget {

    static UNIT_AUTO = 0;
    static UNIT_STATIC = 1;

    onInitialize() {
        super.onInitialize();

        this._refresh_frame = null;
        this._chart_container = null;
        this._canvas = null;
        this._chart_color = null;
        this._min = null;
        this._max = null;
        this._value = null;
        this._last_value = null;
        this._units = '';
    }
```

```
    processUpdateResponse(response) {
        if (response.history === null) {
            this._value = null;
            this._units = '';
        }
        else {
            this._value = Number(response.history.value);
            this._units = response.fields_values.value_units == WidgetLessonGaugeChart.UNIT_AUTO
                ? response.history.units
                : response.fields_values.value_static_units;
        }

        this._chart_color = response.fields_values.chart_color;
        this._min = Number(response.fields_values.value_min);
        this._max = Number(response.fields_values.value_max);

        super.processUpdateResponse(response);
    }

    setContents(response) {
        if (this._canvas === null) {
            super.setContents(response);

            this._chart_container = this._body.querySelector('.chart');
            this._chart_container.style.height =
                `${this._getContentsSize().height - this._body.querySelector('.description').clientHeight}
            this._canvas = document.createElement('canvas');

            this._chart_container.appendChild(this._canvas);

            this._resizeChart();
        }

        this._updatedChart();
    }

    onResize() {
        super.onResize();

        if (this._state === WIDGET_STATE_ACTIVE) {
            this._resizeChart();
        }
    }

    _resizeChart() {
        const ctx = this._canvas.getContext('2d');
        const dpr = window.devicePixelRatio;

        this._canvas.style.display = 'none';
        const size = Math.min(this._chart_container.offsetWidth, this._chart_container.offsetHeight);
        this._canvas.style.display = '';

        this._canvas.width = size * dpr;
        this._canvas.height = size * dpr;

        ctx.scale(dpr, dpr);

        this._canvas.style.width = `${size}px`;
        this._canvas.style.height = `${size}px`;

        this._refresh_frame = null;
```

```
        this._updatedChart();
    }

    _updatedChart() {
        if (this._last_value === null) {
            this._last_value = this._min;
        }

        const start_time = Date.now();
        const end_time = start_time + 400;

        const animate = () => {
            const time = Date.now();

            if (time <= end_time) {
                const progress = (time - start_time) / (end_time - start_time);
                const smooth_progress = 0.5 + Math.sin(Math.PI * (progress - 0.5)) / 2;
                let value = this._value !== null ? this._value : this._min;
                value = (this._last_value + (value - this._last_value) * smooth_progress - this._min) / (t

                const ctx = this._canvas.getContext('2d');
                const size = this._canvas.width;
                const char_weight = size / 12;
                const char_shadow = 3;
                const char_x = size / 2;
                const char_y = size / 2;
                const char_radius = (size - char_weight) / 2 - char_shadow;

                const font_ratio = 32 / 100;

                ctx.clearRect(0, 0, size, size);

                ctx.beginPath();
                ctx.shadowBlur = char_shadow;
                ctx.shadowColor = '#bbb';
                ctx.strokeStyle = '#eee';
                ctx.lineWidth = char_weight;
                ctx.lineCap = 'round';
                ctx.arc(char_x, char_y, char_radius, Math.PI * 0.749, Math.PI * 2.251, false);
                ctx.stroke();

                ctx.beginPath();
                ctx.strokeStyle = `#${this._chart_color}`;
                ctx.lineWidth = char_weight - 2;
                ctx.lineCap = 'round';
                ctx.arc(char_x, char_y, char_radius, Math.PI * 0.75,
                    Math.PI * (0.75 + (1.5 * Math.min(1, Math.max(0, value)))), false
                    );
                ctx.stroke();

                ctx.shadowBlur = 2;
                ctx.fillStyle = '#1f2c33';
                ctx.font = `${(char_radius * font_ratio)|0}px Arial`;
                ctx.textAlign = 'center';
                ctx.textBaseline = 'middle';
                ctx.fillText(`${this._value !== null ? this._value : t('No data')}${this._units}`,
                    char_x, char_y, size - char_shadow * 4 - char_weight * 2
                );

                ctx.fillStyle = '#768d99';
                ctx.font = `${(char_radius * font_ratio * .5)|0}px Arial`;
                ctx.textBaseline = 'top';
```

```
                ctx.textAlign = 'left';
                ctx.fillText(`${this._min}${this._min != '' ? this._units : ''}`,
                    char_weight * .75, size - char_weight * 1.25, size / 2 - char_weight
                );

                ctx.textAlign = 'right';
                ctx.fillText(`${this._max}${this._max != '' ? this._units : ''}`,
                    size - char_weight * .75, size - char_weight * 1.25, size / 2 - char_weight
                );

                requestAnimationFrame(animate);
            }
            else {
                this._last_value = this._value;
            }
        };

        requestAnimationFrame(animate);
    }
}
```

10. Abra *manifest.json* e adicione:

- o nome do arquivo (*class.widget.js*) ao array na seção *assets/js*;
- o nome da classe (*WidgetLessonGaugeChart*) ao parâmetro *js_class* na seção *widget*.

A classe *WidgetLessonGaugeChart* agora será carregada automaticamente com o dashboard.

ui/modules/lesson_gauge_chart/manifest.json

```
{
    "manifest_version": 2.0,
    "id": "lesson_gauge_chart",
    "type": "widget",
    "name": "Gauge chart",
    "namespace": "LessonGaugeChart",
    "version": "1.0",
    "author": "Zabbix",
    "actions": {
        "widget.lesson_gauge_chart.view": {
            "class": "WidgetView"
        }
    },
    "widget": {
        "js_class": "WidgetLessonGaugeChart"
    },
    "assets": {
        "js": ["class.widget.js"]
    }
}
```

Adicionar estilos CSS ao widget

Nesta seção, você aprenderá a adicionar estilos CSS personalizados para tornar o widget mais atraente.

1. Para estilos de widget, crie um novo diretório *css* dentro de *assets*.
2. Crie um arquivo *widget.css* no diretório *assets/css*. Para estilizar os elementos do widget, use o seletor *div.dashboard-widget-{widget id}*. Para configurar o CSS para todo o widget, use o seletor *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;
}

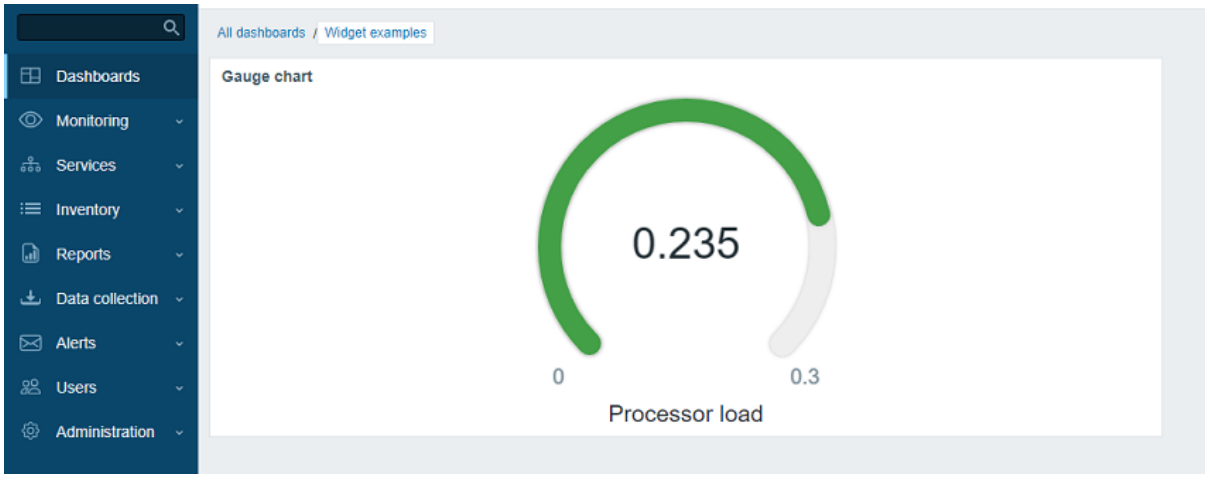
.dashboard-grid-widget-hidden-header div.dashboard-widget-lesson_gauge_chart .chart {
    margin-top: 8px;
}
```

3. Abra o arquivo *manifest.json* e adicione o nome do arquivo CSS (*widget.css*) à lista na seção *assets/css*. Isso permitirá que os estilos CSS definidos em *widget.css* sejam carregados com a página do painel.

ui/modules/lesson_gauge_chart/manifest.json

```
{
    "manifest_version": 2.0,
    "id": "lesson_gauge_chart",
    "type": "widget",
    "name": "Gráfico de medidor",
    "namespace": "LessonGaugeChart",
    "version": "1.0",
    "author": "Zabbix SIA",
    "actions": {
        "widget.lesson_gauge_chart.view": {
            "class": "WidgetView"
        }
    },
    "widget": {
        "js_class": "WidgetLessonGaugeChart"
    },
    "assets": {
        "css": ["widget.css"],
        "js": ["class.widget.js"]
    }
}
```

4. Atualize a página do painel para ver a versão finalizada do widget.



Exemplos

Esta seção fornece arquivos de exemplos de módulos e widgets, que podem ser usados como base para os seus módulos personalizados.

Para usar um módulo:

1. Faça o download do arquivo ZIP.
2. Descompacte o conteúdo em um diretório separado dentro da pasta */zabbix/ui/modules*.
3. Registre o módulo no front-end do Zabbix.

Exemplo de módulo

- Ao criar um grupo de hosts, conceda permissões de leitura aos grupos de usuários configurados - hg_auto_perm.zip

Exemplos de widgets

- Widget mínimo - widget_min.zip
- Widget ”Hello, world” usando somente CSS - hello_world_css.zip
- Widget ”Hello, world” usando somente JavaScript - hello_world_js.zip
- Widget ”Hello, world” usando PHP - hello_world_php.zip

Você também pode usar Zabbix native widgets como exemplos.

Biblioteca Python para o Zabbix

zabbix_utils é a biblioteca oficial do Python para o Zabbix.

Com o zabbix_utils você pode:

- Usar a API do Zabbix para gerenciar objetos do Zabbix (criar hosts, atualizar items, obter eventos, etc.).
- Coletar dados do agent do Zabbix (semelhante ao Zabbix get).
- Enviar dados para o server ou proxy do Zabbix (semelhante ao Zabbix sender).

A biblioteca suporta execução de scripts síncronos e assíncronos, permitindo lidar tanto com fluxos de trabalho simples quanto com operações paralelas em larga escala, como coleta massiva de dados, exportações em massa, processamento de alertas e escalonamento eficiente do sistema.

A biblioteca é compatível com as versões suportadas do server do Zabbix e com as versões modernas do Python.

Para começar, instale a biblioteca e siga o guia quickstart.

Instalação

zabbix_utils pode ser instalado usando qualquer um dos métodos descritos abaixo.

Requisitos:

- Zabbix 6.0 ou posterior (testado em 6.0, 7.0, 7.2, 7.4)
- Python 3.8 ou posterior (testado em 3.8–3.13)
- aiohttp framework HTTP assíncrono (necessário apenas para modo assíncrono)

Python Package Index (PyPI)

Este é o método mais comum para a maioria dos ambientes:

```
pip install zabbix_utils
pip install zabbix_utils[async] # Apenas para suporte assíncrono
```

Repositório Zabbix

Use este método se preferir gerenciar dependências através do gerenciador de pacotes do seu sistema.

Baixe os pacotes Zabbix para sua distribuição Linux e execute os seguintes comandos.

No RHEL e derivados:

```
dnf install python3-zabbix-utils
dnf install epel-release # Apenas para suporte assíncrono
dnf install python3-aiohttp # Apenas para suporte assíncrono
```

No Debian/Ubuntu e derivados:

```
apt install python3-zabbix-utils
apt install python3-aiohttp # Apenas para suporte assíncrono
```

Fonte (GitHub)

Use este método se você preferir a versão de desenvolvimento mais recente:

```
git clone https://github.com/zabbix/python-zabbix-utils
cd python-zabbix-utils/
pip install -r requirements.txt # Apenas para suporte assíncrono
python3 setup.py install
```

Para instalar as dependências necessárias para o modo assíncrono, você também pode usar um dos métodos descritos acima.

Guia de início rápido

Depois de instalar o zabbix_utils, você pode usá-lo em seu script.

Este guia de início rápido mostra como:

- Recuperar o nome do host do agent Zabbix.
- Criar um host e um item trapper usando a API do Zabbix.
- Enviar um valor para o item.

O guia apresenta o script passo a passo, explicando cada parte à medida que é apresentada. O script completo é fornecido ao final da página.

O guia também assume que seu server, agent e API do Zabbix estão sendo executados localmente.

Recuperar o nome do host do agent Zabbix

Comece recuperando o nome do host do sistema onde o agent Zabbix está em execução. Você precisará desse nome de host para criar um host usando a API do Zabbix.

1. Importe a classe `Getter` de zabbix_utils. Esta classe funciona como o Zabbix get e permite que você solicite dados do agent Zabbix.

2. Crie uma instância de `Getter` que se conecta ao agent Zabbix local em `127.0.0.1` na porta `10050`.

3. Chame o método `get()` na instância de `Getter` para solicitar o nome do host ao agent Zabbix.

- O método `get()` recebe uma chave de item como parâmetro e envia uma solicitação ao agent Zabbix para esse item.
- O método `get()` retorna um objeto, e o nome do host é armazenado no atributo `value` desse objeto.
- `system.hostname` é uma chave interna do agent Zabbix que retorna o nome do host.

```
from zabbix_utils import Getter

agent = Getter(host='127.0.0.1', port=10050)
hostname = agent.get('system.hostname').value
```

Conecte-se e faça login na API do Zabbix

Em seguida, conecte-se à API do Zabbix. Isso permite que seu script gerencie objetos do Zabbix, como hosts e items.

1. Importe a classe `ZabbixAPI` de zabbix_utils.

2. Crie uma instância de `ZabbixAPI` e forneça a URL da interface web do Zabbix.

3. Chame o método `login()` na instância de `ZabbixAPI` e forneça seu nome de usuário e senha. A conta deve ter permissão para acessar a API do Zabbix.

4. Chame o método `logout()` para fechar a sessão após as operações da API.

```
from zabbix_utils import ZabbixAPI

api = ZabbixAPI(url='127.0.0.1/zabbix')
api.login(user='Admin', password='zabbix')

### As operações da API vão aqui

api.logout()
```

Criar host no Zabbix

Agora que você tem o nome do host do agent Zabbix e está conectado à API do Zabbix, pode criar um novo host no Zabbix.

1. Chame o método da API `host.create()` na instância `ZabbixAPI` e forneça os detalhes do host:

   - `host` - defina como a variável `hostname` que contém o nome do host recuperado do agent Zabbix.
   - `interfaces` - contém os detalhes de conexão para o agent Zabbix que está em execução no host.
   - `groups` - contém pelo menos um grupo de hosts ao qual o host deve pertencer.

2. Como a API do Zabbix retorna o ID do host recém-criado, salve esse ID na variável `host_id` para uso posterior.

3. Imprima uma mensagem para confirmar que o host foi criado.

```
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}' criado com ID {host_id}")
```

Crie um novo item no Zabbix

Após criar um host, você pode adicionar um item a ele.

1. Defina uma chave exclusiva para o item a ser criado.

2. Chame o método `item.create()` na instância do `ZabbixAPI` e forneça os detalhes do item:

   - `hostid` - defina como a variável `host_id` contendo o ID do host que você acabou de criar.
   - `name` - um nome para o item.
   - `key_` - defina como a variável `item_key` que você acabou de definir.
   - `type` - defina como 2 (item trapper), necessário para receber valores enviados do seu script nas etapas seguintes.
   - `value_type` - defina como 3 (numérico sem sinal), o tipo de dado que este item armazena.

3. Como a API do Zabbix retorna o ID do item recém-criado, salve este ID na variável `item_id` para uso posterior.

4. Imprima uma mensagem para confirmar que o item foi criado.

```
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}")
```

Enviar valor para o host

Agora que você criou seu host e item, pode enviar dados para ele.

1. Importe a classe `time` e aguarde alguns segundos antes de enviar os dados. Isso garante que o Zabbix tenha processado totalmente seu novo host e item.

2. Importe a classe `Sender` de zabbix_utils. Esta classe funciona como o Zabbix sender e permite que seu script envie dados para o Zabbix.

3. Crie uma instância de `Sender` que se conecta ao Zabbix server local em `127.0.0.1` na porta `10051`.

4. Chame o método `send_value()` na instância de `Sender` para enviar um valor para o item do seu host e forneça estes detalhes:

- `hostname` - defina como a variável `hostname` que contém o nome do host que você recuperou do Zabbix agent.
- `item_key` - defina como a variável `item_key` que você acabou de definir anteriormente em seu script.
- `1` - o valor a ser enviado.

4. Imprima uma mensagem para confirmar que o valor foi enviado.

```
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}")
```

Após a execução bem-sucedida deste código, você deverá ver uma mensagem de sucesso indicando que o Zabbix recebeu seu valor:

```
Sender response: {"processed": 1, "failed": 0, "total": 1, "time": "0.000151", "chunk": 1}
```

Agora você também pode verificar sua interface web do Zabbix (*Monitoramento* > *Últimos dados*) para ver o valor.

Script completo

Abaixo está o script completo que combina todas as etapas: recuperando o nome do host, criando o host e o item usando a API do Zabbix e enviando dados para o Zabbix.

```
import time
from zabbix_utils import Getter, ZabbixAPI, Sender

### Recupera o nome do host do agent Zabbix
agent = Getter(host='127.0.0.1', port=10050)
hostname = agent.get('system.hostname').value

### Conecta e faz login na API do Zabbix
api = ZabbixAPI(url='127.0.0.1/zabbix')
api.login(user='Admin', password='zabbix')
```

```
### Cria o host no Zabbix
host = api.host.create(
    host=hostname,
    interfaces=[{
        'type': 1,
        'main': 1,
        'useip': 1,
        'ip': '127.0.0.1',
        'dns': '',
        'port': '10050',
    }],
    groups=[{'groupid': '2'}]
)

host_id = host['hostids'][0]

print(f"Host '{hostname}' criado com ID {host_id}")

### Cria o item no Zabbix
item_key = 'app.myservice.heartbeat'

item = api.item.create(
    hostid=host_id,
    name='App heartbeat',
    key_=item_key,
    type=2,
    value_type=3
)

item_id = item['itemids'][0]

print(f"Item '{item_key}' criado com ID {item_id}")

### Faz logout da API
api.logout()

### Aguarda o Zabbix processar o novo host
time.sleep(10)

### Envia valor para o host
sender = Sender(server='127.0.0.1', port=10051)
response = sender.send_value(hostname, item_key, 1)

print(f"Resposta do sender: {response}")
```

Usar a API do Zabbix

zabbix_utils permite que você use a API do Zabbix para gerenciar objetos do Zabbix, incluindo a criação de hosts, atualização de items, recuperação de eventos e muito mais.

As requisições à API podem ser feitas em modo síncrono ou assíncrono:

- No modo síncrono, seu script Python envia uma requisição e aguarda uma resposta antes de continuar, o que é adequado para operações simples, sequenciais e previsíveis.
- No modo assíncrono, o script envia requisições sem esperar por cada resposta, permitindo que outras operações prossigam em paralelo; isso é mais eficiente para requisições lentas ou grandes lotes de dados.

Os exemplos nesta página focam no modo síncrono, embora o modo assíncrono siga padrões semelhantes. Exemplos adicionais estão disponíveis no repositório do GitHub zabbix_utils.

Importação

Para usar o zabbix_utils para trabalhar com a API do Zabbix, importe a classe `ZabbixAPI` em seu script:

```
from zabbix_utils import ZabbixAPI
```

> **Note:**
> Se você já usa uma biblioteca Python da comunidade, normalmente pode substituir essa importação por `ZabbixAPI` do zabbix_utils.

Fazer login

Antes de fazer solicitações à API, você deve criar uma instância do `ZabbixAPI` e fazer login na API do Zabbix.

Escolha o método que melhor se adapta à forma como você gerencia as credenciais. Você pode usar seu nome de usuário e senha ou tokens de API.

Durante a inicialização

Ao criar uma instância de `ZabbixAPI`, você pode fornecer a URL da interface web do Zabbix e suas credenciais de uma só vez:

```
### Nome de usuário e senha:
api = ZabbixAPI(url="127.0.0.1/zabbix", user="Admin", password="zabbix")

### Token da API:
api = ZabbixAPI(url="127.0.0.1/zabbix", token="your_api_token")
```

Se preferir agrupar os parâmetros de conexão, você pode passá-los como um dicionário Python e desempacotá-los usando a sintaxe de argumentos nomeados do Python:

```
ZABBIX_AUTH = {
    "url": "127.0.0.1/zabbix",
    "user": "Admin",
    "password": "zabbix"
}

api = ZabbixAPI(**ZABBIX_AUTH)
```

Usando `login()`

Você pode primeiro criar uma instância `ZabbixAPI` apenas com a URL da interface web do Zabbix e, posteriormente, usar o método `login()` com suas credenciais:

```
### Nome de usuário e senha:
api = ZabbixAPI(url="127.0.0.1/zabbix")
api.login(user="Admin", password="zabbix")

### Token da API:
api = ZabbixAPI(url="127.0.0.1/zabbix")
api.login(token="your_api_token")
```

Usando variáveis de ambiente

Você pode primeiro armazenar suas credenciais como variáveis de ambiente, por exemplo, usando um terminal:

```
### Nome de usuário e senha:
export ZABBIX_USER="Admin"
export ZABBIX_PASSWORD="zabbix"

### Token:
export ZABBIX_TOKEN="your_api_token"
```

Em seguida, se você iniciar seu script a partir da mesma sessão do terminal, a instância `ZabbixAPI` no script requer apenas a URL da interface web do Zabbix:

```
api = ZabbixAPI(url="127.0.0.1/zabbix")
```

Você também pode armazenar a URL em uma variável de ambiente:

```
### Nome de usuário e senha:
export ZABBIX_USER="Admin"
export ZABBIX_PASSWORD="zabbix"
```

```
export ZABBIX_URL="https://127.0.0.1/zabbix"

### Token:
export ZABBIX_TOKEN="your_api_token"
export ZABBIX_URL="https://127.0.0.1/zabbix"
```

Quando a URL e as credenciais estão todas definidas como variáveis de ambiente, a instância `ZabbixAPI` no script não requer nenhum argumento:

```
api = ZabbixAPI()
```

Sair

Se você fez login usando um nome de usuário e senha, chame o método `logout()` após concluir suas operações de API:

```
from zabbix_utils import ZabbixAPI

api = ZabbixAPI(url="127.0.0.1/zabbix")
api.login(user="Admin", password="zabbix")

### As ações da API vão aqui

api.logout()
```

> **Note:**
> Chamar o método `logout()` não é necessário ao usar tokens de API.

Requisições de API

Depois de fazer login, você pode fazer qualquer requisição de API chamando os métodos descritos na referência de métodos da API do Zabbix.

Os métodos da API são chamados usando o seguinte formato:

```
api_instance.zabbix_object.method(parameters)
```

> **Note:**
> Alguns nomes de métodos ou objetos da API do Zabbix usam palavras que são palavras-chave reservadas em Python (por exemplo, `import`). Para evitar erros em Python, adicione um sublinhado ao nome do método ou objeto ao usá-lo em Python (por exemplo, `api.configuration.import_`).

Por exemplo, para ver uma lista de hosts com `host.get`:

```
### 1. Importe ZabbixAPI de zabbix_utils:
from zabbix_utils import ZabbixAPI

### 2. Crie a instância do ZabbixAPI e faça login:
api = ZabbixAPI(url="127.0.0.1/zabbix")
api.login(user="Admin", password="zabbix")

### 3. Recupere hosts que correspondam a um filtro:
hosts = api.host.get(
    filter={
        "host": [
            "Zabbix server",
            "Linux server"
        ]
    }
)

### 4. Itere sobre os hosts retornados e imprima os detalhes de cada host:
for host in hosts:
    print(host)

### 5. Faça logout da API para fechar a sessão:
```

```
api.logout()
```

Modo assíncrono

O modo assíncrono permite que seu script envie solicitações de API sem esperar que cada uma termine. Isso pode tornar seu script mais eficiente quando ele precisa executar muitas solicitações de API ou quando algumas solicitações demoram muito.

Ao usar o modo assíncrono, há diferenças importantes em relação ao modo síncrono:

- Importe o módulo `asyncio` do Python (você deve primeiro instalar as dependências necessárias).
- Importe `AsyncZabbixAPI` em vez de `ZabbixAPI`.
- Escreva seu código dentro de uma função `async`.
- Use `await` ao chamar métodos da API, incluindo `login()` e `logout()`.

> **Attention:**
> Ao criar uma instância de `AsyncZabbixAPI`, você não pode fazer login durante a inicialização.

Por exemplo, para ver uma lista de hosts com `host.get` no modo assíncrono:

```
### 1. Importe asyncio para modo assíncrono e AsyncZabbixAPI de zabbix_utils:
import asyncio
from zabbix_utils import AsyncZabbixAPI

### 2. Defina a função principal async onde todas as operações da API serão executadas:
async def main():

    # 3. Crie uma instância de AsyncZabbixAPI e faça login (deve aguardar):
    api = AsyncZabbixAPI(url="127.0.0.1")
    await api.login(user="User", password="zabbix")

    # 4. Recupere hosts que correspondam a um filtro (deve aguardar):
    hosts = await api.host.get(
        filter={
            "host": [
                "Zabbix server",
                "Linux server"
            ]
        }
    )

    # 5. Itere sobre os hosts retornados e imprima os detalhes de cada host:
    for host in hosts:
        print(host)

    # 6. Faça logout da API para fechar a sessão (deve aguardar):
    await api.logout()

### 7. Execute a função async main() usando o event loop do asyncio:
asyncio.run(main())
```

Exemplos

Os exemplos a seguir mostram tarefas comuns da API do Zabbix usando a biblioteca zabbix_utils.

Exemplos adicionais estão disponíveis no diretório `examples` do repositório zabbix_utils no GitHub.

Coletar dados do agent Zabbix

Visão geral

zabbix_utils permite coletar dados do agent do Zabbix (semelhante ao Zabbix get).

Os dados podem ser coletados em modo síncrono ou assíncrono:

- No modo síncrono, seu script Python solicita e aguarda os dados antes de continuar, o que é adequado para operações simples, sequenciais e previsíveis.
- No modo assíncrono, o script solicita dados sem esperar por cada resposta, permitindo que outras operações prossigam em paralelo; isso é mais eficiente para solicitações lentas ou grandes lotes de dados.

Os exemplos nesta página focam no modo síncrono, embora o modo assíncrono siga padrões semelhantes. Exemplos adicionais estão disponíveis no repositório do GitHub zabbix_utils.

Importação

Para usar o zabbix_utils para coletar valores de item, importe a classe `Getter` em seu script Python:

```
from zabbix_utils import Getter
```

Solicitar dados

Para solicitar o valor de um item:

1. Crie uma instância de `Getter`, especificando o endereço IP e a porta do seu agent Zabbix.
2. Chame o método `get()` na instância de `Getter`, especificando a chave do item que você deseja recuperar.

Por exemplo, para solicitar dados para o item `system.uname`:

```
agent = Getter(host='192.0.2.0', port=10050)
response = agent.get('system.uname')
```

Usando IP não padrão

Se o servidor que executa seu script tiver vários endereços IP, você pode especificar um `source_ip` para o `Getter` usar ao se conectar ao agent do Zabbix:

```
agent = Getter(
    host='192.0.2.0',
    port=10050,
    source_ip='10.10.7.1'
)
```

Usando timeout

Você pode definir um `timeout` de resposta para o `Getter` para controlar quanto tempo seu script deve aguardar por uma resposta do agent Zabbix antes de desistir:

```
agent = Getter(
    host='192.0.2.0',
    port=10050,
    timeout=30
)
```

Usando criptografia

O `Getter` não inclui suporte de criptografia embutido, mas você pode fornecê-lo criando um wrapper usando bibliotecas de terceiros:

```
def psk_wrapper(sock, tls):
    # ...
    # Implementação do wrapper TLS PSK para o socket
    # ...

agent = Getter(
    host='192.0.2.0',
    port=10050,
    socket_wrapper=psk_wrapper
)
```

Resposta

A resposta do agent Zabbix é processada pela biblioteca e retornada como um objeto `AgentResponse`:

```
print(response)
### {
###     "error": null,
###     "raw": "Linux zabbix_server 5.15.0-3.60.5.1.el9uek.x86_64",
```

```
###     "value": "Linux zabbix_server 5.15.0-3.60.5.1.el9uek.x86_64"
### }

print(response.value)
### Linux zabbix_server 5.15.0-3.60.5.1.el9uek.x86_64

print(response.error)
### None
```

Modo assíncrono

O modo assíncrono permite que seu script colete valores sem esperar que cada um deles chegue. Isso pode tornar seu script mais eficiente quando precisa coletar muitos valores ou quando alguns valores demoram para serem coletados.

Ao usar o modo assíncrono, há diferenças importantes em relação ao modo síncrono:

- Importe o módulo `asyncio` do Python (você deve primeiro instalar as dependências necessárias).
- Importe `AsyncGetter` em vez de `Getter`.
- Escreva seu código dentro de uma função `async`.
- Use `await` ao chamar o método `get()`.

Por exemplo, para coletar um único valor usando o modo assíncrono:

```
### 1. Importe asyncio para o modo assíncrono e AsyncGetter de zabbix_utils:
import asyncio
from zabbix_utils import AsyncGetter

### 2. Defina a função principal async onde todas as requisições de dados serão executadas:
async def main():
    agent = AsyncGetter(host='192.0.2.0', port=10050)

    # 3. Busque o valor de system.uname do agent Zabbix (deve usar await):
    response = await agent.get('system.uname')

    # 4. Imprima o valor retornado pelo agent Zabbix:
    print(response.value)

### 5. Execute a função async main() usando o event loop do asyncio:
asyncio.run(main())
```

Enviar dados para o Zabbix server ou proxy

O zabbix_utils permite enviar valores de item para um item trapper no Zabbix server ou proxy (de forma semelhante ao Zabbix sender).

Você pode enviar um único valor, múltiplos valores ou até mesmo direcionar para múltiplos clusters Zabbix.

Os dados podem ser enviados em modo síncrono ou assíncrono:

- No modo síncrono, seu script Python envia valores e aguarda uma resposta antes de continuar; isso é adequado para operações simples, sequenciais e previsíveis.
- No modo assíncrono, o script envia valores sem esperar por cada resposta, permitindo que outras operações prossigam em paralelo; isso é mais eficiente para solicitações lentas ou grandes lotes de dados.

Os exemplos nesta página focam no modo síncrono, embora o modo assíncrono siga padrões semelhantes. Exemplos adicionais estão disponíveis no repositório GitHub do zabbix_utils.

Importação

Para usar o zabbix_utils para enviar valores de item, importe a classe `Sender` em seu script:

```
from zabbix_utils import Sender
```

Para enviar múltiplos valores, você também pode importar a classe `ItemValue`:

```
from zabbix_utils import Sender, ItemValue
```

Enviar valor único

Para enviar o valor de um item:

1. Crie uma instância de `Sender`, especificando o endereço IP e a porta do seu Zabbix server ou proxy.
2. Chame o método `send_value()` na instância de `Sender` usando o seguinte formato:

```
sender_instance.send_value('host', 'item.key', 'value', optional_timestamp, optional_nanoseconds)
```

Por exemplo, para enviar 1 para o item trapper `service.status` no host `Linux server`:

```
sender = Sender(server='127.0.0.1', port=10051)
response = sender.send_value('Linux server', 'service.status', 1)
```

Usando IP não padrão

Se o servidor que executa seu script tiver vários endereços IP, você pode especificar um `source_ip` para o `Sender` usar ao enviar valores para o Zabbix server ou proxy:

```
sender = Sender(
    server='127.0.0.1',
    port=10051,
    source_ip='10.10.7.1'
)
```

Usando timeout

Você pode definir um `timeout` de resposta para o `Sender` para controlar quanto tempo seu script deve aguardar por uma resposta do Zabbix server ou proxy antes de desistir:

```
sender = Sender(
    server='127.0.0.1',
    port=10051,
    timeout=30
)
```

Usando o arquivo de configuração do agent

Você pode permitir que o zabbix_utils leia os parâmetros `Server` ou `ServerActive` de um arquivo de configuração local do Zabbix agent ou agent 2. Nesses casos, você não precisa especificar os parâmetros de conexão ao criar uma instância de `Sender`:

```
sender = Sender(
    use_config=True,
    config_path='/etc/zabbix/zabbix_agent2.conf'
)
```

> **Attention:**
> Se `ServerActive` contiver um ou mais clusters Zabbix com múltiplas instâncias de server, o `Sender` enviará dados para o primeiro server disponível em cada cluster. Se `ServerActive` não estiver definido, o endereço de `Server` com a porta padrão (10051) será usado.

Usando criptografia

O `Sender` não inclui suporte de criptografia embutido, mas você pode fornecê-lo criando um wrapper usando bibliotecas de terceiros:

```
def psk_wrapper(sock, tls):
    # ...
    # Implementação do wrapper TLS PSK para o socket
    # ...

sender = Sender(
    server='127.0.0.1',
    port=10051,
    socket_wrapper=psk_wrapper
)
```

Resposta para valor único

A resposta retornada pelo server ou proxy Zabbix é processada pela biblioteca e retornada como um objeto `TrapperResponse`:

```
print(response)
### {"processed": 1, "failed": 0, "total": 1, "time": "0.000123", "chunk": 1}

print(response.processed)
### 1

print(response.failed)
### 0

print(response.total)
### 1
```

Modo assíncrono

O modo assíncrono permite que seu script Python envie valores sem esperar por uma resposta do Zabbix server ou proxy. Isso pode tornar seu script mais eficiente quando ele precisa enviar muitos valores ou quando alguns valores demoram para serem enviados.

Ao usar o modo assíncrono, há algumas diferenças importantes em relação ao modo síncrono:

- Importe o módulo `asyncio` do Python (você deve primeiro instalar as dependências necessárias).
- Importe `AsyncSender` em vez de `Sender`.
- Escreva seu código dentro de uma função `async`.
- Use `await` ao chamar o método `send_value()`.

Por exemplo, para enviar um único valor usando o modo assíncrono:

```
### 1. Importe asyncio para modo assíncrono e AsyncSender do zabbix_utils:
import asyncio
from zabbix_utils import AsyncSender

### 2. Defina a função principal async onde todas as operações de envio de dados (devem aguardar) serão ex
async def main():
    sender = AsyncSender(server='127.0.0.1', port=10051)
    response = await sender.send_value('Linux server', 'service.status', 1)

    # 3. Imprima a resposta retornada pelo Zabbix server ou proxy:
    print(response)

### 4. Execute a função async main() usando o event loop do asyncio:
asyncio.run(main())
```

Enviar múltiplos valores

Para enviar múltiplos valores:

1. Prepare um array de objetos `ItemValue`, cada um usando o mesmo formato do método `send_value()`.
2. Crie uma instância de `Sender`, especificando o endereço IP e a porta do seu Zabbix server ou proxy.
3. Chame o método `send()` (em vez de `send_value()`) na instância de `Sender`, especificando o array de objetos com os valores a serem enviados.

Por exemplo, para enviar cinco valores para diferentes hosts:

```
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)
```

Usando tamanho de bloco personalizado

Se você precisar enviar mais valores do que um item trapper pode aceitar em uma única solicitação, poderá dividi-los em blocos.

Por padrão, o tamanho do bloco é de 250 valores. Você pode alterá-lo definindo o parâmetro `chunk_size` ao criar uma instância de `Sender`.

Por exemplo, para enviar cinco valores em três blocos (2-2-1), defina o parâmetro `chunk_size` como 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)
```

Enviar valores para vários clusters Zabbix

Para enviar valores para vários clusters Zabbix:

1. Prepare um array de clusters Zabbix. Se um cluster tiver vários nós, o valor será enviado para o primeiro nó **disponível** de cada cluster.
2. Crie um `Sender`, especificando seu array de clusters Zabbix.
3. Chame o método `send_value()` na instância do `Sender` usando o mesmo formato do método `send_value()`.

Por exemplo, para enviar um valor para o primeiro nó disponível em cada cluster:

```
zabbix_clusters = [
    ['zabbix.cluster1.node1', 'zabbix.cluster1.node2:10051'],
    ['zabbix.cluster2.node1:10051', 'zabbix.cluster2.node2', 'zabbix.cluster2.node3']
]

sender = Sender(clusters=zabbix_clusters)
response = sender.send_value('Linux server', 'service.status', 1)
```

Resposta para múltiplos valores

Por padrão, o `Sender` retorna um resultado agregado do envio de valores em todos os hosts ou clusters:

```
print(response)
### {"processed": 2, "failed": 0, "total": 2, "time": "0.000108", "chunk": 2}
```

Se você precisar de informações mais detalhadas, pode inspecionar os resultados para cada cluster e cada chunk usando o atributo `response.details`:

```
print(response)
### {"processed": 2, "failed": 0, "total": 2, "time": "0.000108", "chunk": 2}

if response.failed == 0:
    print(f"Valor enviado com sucesso em {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"Processado {resp.processed} de {resp.total} em {node.address}:{node.port}")
            # Processado 1 de 1 em 127.0.0.1:10051
            # Processado 1 de 1 em zabbix.example.local:10051
```

Log de depuração

Para solucionar problemas ao trabalhar com a API do Zabbix, coletar dados do agent do Zabbix ou enviar dados para o server ou proxy do Zabbix, você pode ativar o log de depuração.

A biblioteca zabbix_utils usa o módulo padrão `logging` do Python.

Para ver mensagens detalhadas de depuração da biblioteca:

1. Importe o módulo `logging` do Python em seu script.
2. Configure o sistema de logging para exibir mensagens em um formato legível e defina o nível de logging como 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)
```

Com o log de depuração ativado, informações detalhadas sobre as solicitações e respostas são impressas, por exemplo:

```
[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
```

Plugins

Visão geral Plugins customizados carregáveis estendem a funcionalidade do agent Zabbix 2. Eles são compilados separadamente, mas usam um pacote compartilhado com o agent Zabbix 2.

Cada plugin é um pacote *Go* que define a estrutura e implementa uma ou várias interfaces de plugin (*Exporter*, *Configurator*, *Runner*).

Ir para:

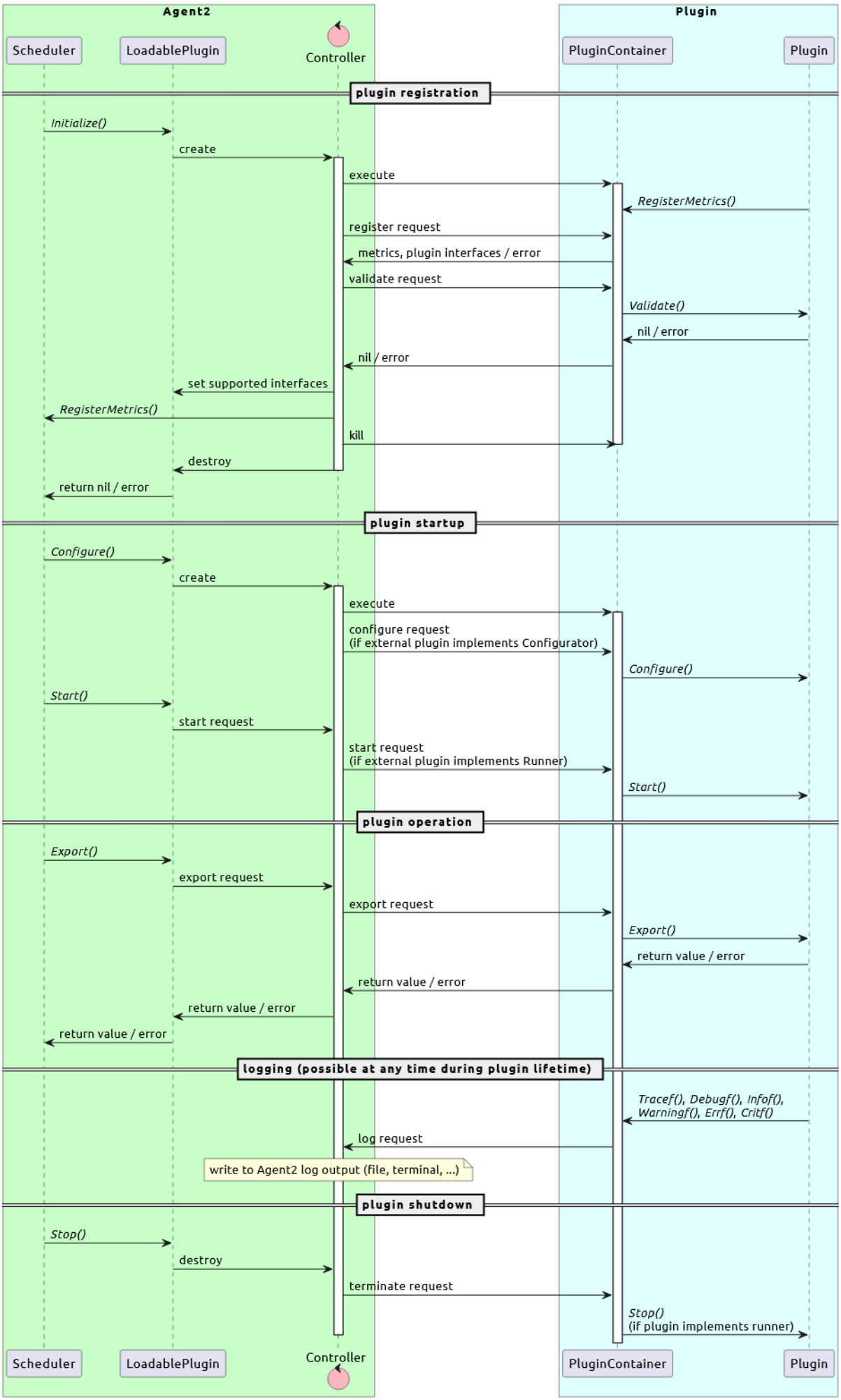
- Escreva seu primeiro plugin
- Interfaces de plugin

Veja também:

- Exemplo de plugin para o agent Zabbix 2

Diagrama de conexão O agent Zabbix 2 conecta-se bidirecionalmente aos plugins usando sockets UNIX no Linux e Named Pipes no Windows.

O diagrama de conexão abaixo ilustra o processo de comunicação entre o agent Zabbix 2 e um plugin carregável e o processo de coleta de métricas.


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

Exemplos

Você pode usar vários exemplos vazios, bem como plug-ins carregáveis existentes fornecidos pelo Zabbix como referência:

- Examples
- Plugin MongoDB
- Plugin do PostgreSQL

Criar um plugin (tutorial)

Este é um tutorial passo a passo sobre como criar um plugin carregável simples para o agente Zabbix 2.

Você também pode usar o repositório de exemplo como modelo ou guia para criar seus próprios plugins.

O que você criará Durante este tutorial, você adicionará um novo plug-in carregável **MyIP**. O plug-in implementará uma métrica chamada **myip**, que retorna o endereço IP externo do host em que o Zabbix Agent 2 está sendo executado.

Etapa 1: Configuração

1. Um plugin é um módulo Go padrão. Comece inicializando o arquivo `go.mod` no diretório do plugin para rastrear as dependências do plugin:

```
cd caminho/para/plugins/myip # Mude para o diretório do seu plugin
go mod init myip
```

2. Instale a dependência obrigatória do Zabbix Go SDK (`golang.zabbix.com/sdk`):

```
go get golang.zabbix.com/sdk@$LATEST_COMMIT_HASH
```

Substitua `$LATEST_COMMIT_HASH` pelo hash de commit `HEAD` mais recente do `golang.zabbix.com/sdk` repositório na ramificação de lançamento apropriada.

Por exemplo:

```
go get golang.zabbix.com/sdk@af85407
```

Observe que o versionamento `golang.zabbix.com/sdk` não é suportado atualmente, mas isso pode mudar no futuro.

Dependências adicionais podem ser instaladas conforme necessário usando `go get`.

3. Crie um arquivo `main.go` vazio para o código-fonte do plugin:

```
touch main.go
```

Agora a configuração inicial está concluída e o plugin está pronto para desenvolvimento.

Etapa 2: Estrutura do plugin O módulo `golang.zabbix.com/sdk`, instalado na etapa anterior, fornece a estrutura necessária para o desenvolvimento de plugins e garante que todos tenham uma estrutura consistente.

1. Configure o fluxo de execução básico.

Comece definindo o fluxo de execução principal do plugin. Adicione o seguinte código a `main.go`:

```
package main

func main() {
err := run()
if err != nil {
panic(err)
}
}

func run() error {
return nil
}
```

Isso estabelece o fluxo de execução básico para o plugin. A função run conterá posteriormente a lógica central do plugin.

2. Explore as interfaces do plugin.

Um plugin do agente Zabbix 2 deve ser representado por uma estrutura que implementa interfaces do pacote `golang.zabbix.com/sdk/plu`

- *Assessor* - define métodos essenciais que todos os plugins devem implementar, como definir o nome do plugin e lidar com timeouts de chaves de itens.
- Uma ou mais das seguintes interfaces funcionais de plugin:
- *Exportador* - realiza uma consulta e retorna um valor (ou valores), nada ou um erro; frequentemente usado em conjunto com a interface *Coletor*.
- *Coletor* - gerencia a coleta periódica de dados.
- *Executor* - define os procedimentos de inicialização e desligamento do plugin.
- *Observador* - permite implementar consultas de métricas independentes, ignorando o agendador interno do agente; útil para monitoramento baseado em traps ou eventos.
- *Configurador* - define como o plugin lê e aplica suas configurações.

Você pode implementar essas interfaces você mesmo ou usar as interfaces padrão fornecidas pelo Zabbix Go SDK, modificando-as conforme necessário. Este tutorial usa as implementações padrão.

3. Crie a estrutura do plugin.

Agora, importe o pacote *plugin* e crie uma estrutura `myIP` que incorpore a estrutura `plugin.Base`:

```
import "golang.zabbix.com/sdk/plugin"

type myIP struct {
plugin.Base
}
```

A estrutura myIP atualmente atende à interface Accessor. Um método para implementar uma das interfaces funcionais do plugin, o `Exporter`, será adicionado posteriormente neste tutorial.

Etapa 3: Definir chaves de item Seu plugin precisa das chaves de item para coletar dados e fornecê-los ao servidor ou proxy Zabbix.

1. Importe o pacote *errs* para tratamento de erros:

```
import "golang.zabbix.com/sdk/errs"
```

2. Registre as chaves de item usando a função `plugin.RegisterMetrics()` dentro da função `run()`:

```
func run() error {
p := &myIP{}

// Registre a chave de item `myip`.
err := plugin.RegisterMetrics(
p,
"MyIP", // Nome do plugin
"myip", // Nome da chave do item
"Retorna o endereço IP do host.", // Descrição da chave do item
)
if err != nil {
return errs.Wrap(err, "falha ao registrar métricas")
}

return nil
}
```

Para registrar várias chaves de item, repita os parâmetros *nome da métrica* e *descrição* para cada métrica. Por exemplo:

```
plugin.RegisterMetrics(&impl, "Myip", "metric.one", "Descrição da métrica um.", "metric.two", "Descrição d
```

Etapa 4: Configurar o manipulador O manipulador facilita a comunicação entre o agente e o plugin.

1. Importe o pacote *container*:

```
import "golang.zabbix.com/sdk/plugin/container"
```

2. Insira o código da função `run()` para criar e configurar um manipulador:

```
func run() error {
p := &myIP{}

// Registre a chave de item `myip`.
err := plugin.RegisterMetrics(
p,
"MyIP", // Nome do plugin
"myip", // Nome da chave do item
"Retorna o endereço IP do host.", // Descrição da chave do item
)
if err != nil {
return errs.Wrap(err, "falha ao registrar métricas")
}

// Crie um novo manipulador.
h, err := container.NewHandler("MyIP") // Nome do plugin
if err != nil {
return errs.Wrap(err, "falha ao criar novo manipulador")
}

// Configura o registro para encaminhar logs do plugin para o agente.
// Disponível via p.Logger.Infof, p.Logger.Debugf, etc.
p.Logger = h

// Inicia a execução do plugin.
// Bloqueia até que uma solicitação de término seja recebida do agente.
err = h.Execute()
if err != nil {
return errs.Wrap(err, "falha ao executar o manipulador do plugin")
}

return nil
}
```

Etapa 5: Implementar a coleta de dados A coleta de dados é feita por meio da interface Exporter, que descreve o método Export:

```
func Export(
key string, // A chave do item a ser coletada.
params []string, // Argumentos passados para a chave do item (`myip[arg1, arg2]`).
context ContextProvider // Metadados sobre a coleta de dados da chave do item.
) (any, error)
```

1. Importe os pacotes necessários para solicitações HTTP e leitura de respostas:

```
import(
"io"
"net/http"
)
```

2. Implemente o método Export para a estrutura myIP:

```
func (p *myIP) Export(
key string, params []string, context plugin.ContextProvider,
) (any, error) {
// O plugin pode usar diferentes lógicas de coleta de dados com base no parâmetro `key`.
// Esta implementação apenas verifica se a `chave` fornecida é suportada.
if key != "myip" {
return nil, errs.Errorf("chave de item desconhecida %q", key)
}

// O log será encaminhado para o log do agente 2.
p.Infof(
"solicitação recebida para manipular a chave %q com %d parâmetros",
```

```
key,
len(params),
)

// Coleta os dados e os retorna.

resp, err := http.Get("https://api.ipify.org")
if err != nil {
return nil, errs.Wrap(err, "falha ao obter o endereço IP")
}

defer resp.Body.Close()

body, err := io.ReadAll(resp.Body)
if err != nil {
return nil, errs.Wrap(err, "falha ao ler o corpo da resposta")
}

return string(body), nil
}
```

Etapa 6: Compilar e configurar o plugin

1. Para compilar o plugin, execute:

```
go mod tidy
go build
```

Isso deve criar um executável `myip` no diretório atual.

2. Configure o agente Zabbix 2 para usar o plugin:

```
echo "Plugins.MyIP.System.Path=$PATH_TO_THE_MYIP_PLUGIN_EXECUTABLE" > /etc/zabbix_agent2.d/plugins.d/myip.
```

Substitua `$PATH_TO_THE_MYIP_PLUGIN_EXECUTABLE` pelo caminho para o `myip` criado na etapa 5.

O nome do plugin no parâmetro de configuração name (*MyIP* neste tutorial) deve corresponder ao nome do plugin definido na função *plugin.RegisterMetrics()*.

3. Para testar o plugin e seu item `myip`, execute:

```
zabbix_agent2 -c /etc/zabbix_agent2.conf -t myip
```

A saída deve conter um endereço IP externo do seu host e ser semelhante a este:

```
myip [s|192.0.2.0]
```

Com isso, você criou um plugin carregável simples para o agente Zabbix 2. Parabéns!

```
package main

import (
    "io"
    "net/http"
    "golang.zabbix.com/sdk/errs"
    "golang.zabbix.com/sdk/plugin"
    "golang.zabbix.com/sdk/plugin/container"
)
var _ plugin.Exporter = (*myIP)(nil)
type myIP struct {
    plugin.Base
}

func main() {
    err := run()
    if err != nil {
```

```
        panic(err)
    }
}
func run() error {
    p := &myIP{}

    // Registra a chave do item `myip`.
    err := plugin.RegisterMetrics(
        p,
        "MyIP",   // Nome do plugin
        "myip",   // Nome da chave do item
        "Retorna o endereço IP do host.", // Descrição da chave do item
    )
    if err != nil {
        return errs.Wrap(err, "falha ao registrar métricas")
    }
    // Cria um novo manipulador.
    h, err := container.NewHandler("MyIP") // Nome do plugin
    if err != nil {
        return errs.Wrap(err, "falha ao criar novo manipulador")
    }

    // Configura o registro para encaminhar logs do plugin para o agente.
    // Disponível via p.Logger.Infof, p.Logger.Debugf, etc.
    p.Logger = h

    // Inicia a execução do plugin.
    // Bloqueia até que uma solicitação de encerramento do agente seja recebida.
    err = h.Execute()
    if err != nil {
        return errs.Wrap(err, "falha ao executar o manipulador do plugin")
    }
    return nil
}

func (p *myIP) Export(
    key string, params []string, context plugin.ContextProvider,
) (any, error) {
    // O plugin pode usar uma lógica de coleta de dados diferente com base no parâmetro `key`.
    // Esta implementação verifica apenas se a`key` fornecida é suportada.
    if key != "myip" {
        return nil, errs.Errorf("unknown item key %q", key)
    }

    // O log será encaminhado para o log do agente 2.
    p.Infof(
        "received request to handle %q key with %d parameters",
        key,
        len(params),
    )

    // Coleta os dados e os retorna.
    resp, err := http.Get("https://api.ipify.org")
    if err != nil {
        return nil, errs.Wrap(err, "falha ao obter o endereço IP")
    }

    defer resp.Body.Close()

    body, err := io.ReadAll(resp.Body)
    if err != nil {
        return nil, errs.Wrap(err, "falha ao ler o corpo da resposta")
```

```
    }

    return string(body), nil
}
```

Código-fonte completo

Interfaces de plug-in

Esta seção descreve as interfaces de plug-in disponíveis.

plugin.Exporter O *Exporter* é a interface mais simples que executa uma pesquisa e retorna um valor (valores), nada ou erro. Ela aceita uma chave de item preparada, parâmetros e contexto. O acesso a todas as outras interfaces de plug-in é exclusivo e nenhum método pode ser chamado se um plug-in já estiver executando uma tarefa. Além disso, há um limite máximo de 100 chamadas *Export()* simultâneas por plug-in, que pode ser reduzido de acordo com os requisitos de cada plug-in.

Plugin.Configurator A interface *Configurator* fornece parâmetros de configuração do plug-in a partir dos arquivos de configuração do Zabbix Agent 2.

plugin.Runner *A interface Runner* fornece os meios para executar a inicialização quando um plug-in é iniciado (ativado) e a desinicialização quando um plug-in é interrompido (desativado). Por exemplo, um plug-in pode iniciar/parar alguma *goroutine* em segundo plano implementando a interface Runner.

Mudanças no desenvolvimento de extensões

Essa página lista todas as mudanças no desenvolvimento de extensões personalizadas do Zabbix.

Mudanças da 6.4 para 7.0 ZBXNEXT-8086 A chave de objeto `template_support` não é mais suportada para o parâmetro `manifest.json` `widget`. Não é mais necessário determinar se um widget deve estar disponível em dashboards de template, pois, desde o Zabbix 7.0, dashboards de template suportam todos os widgets. Para mais informações, consulte O que há de novo no Zabbix 7.0.0.

ZBXNEXT-8683 Uma nova classe de campo de formulário CWidgetFieldMultiSelectItemPattern foi criada. Esta classe permite a adição de um campo de seleção múltipla projetado especificamente para selecionar padrões de item.

Páginas de manual do Zabbix

Documentação de referência para ferramentas e serviços de linha de comando do Zabbix, listando comandos disponíveis, opções e exemplos de uso.

zabbix_agent2

Sessão: Comandos de Manutenção (8)
Atualizado: 2019-01-29
Índice Retornar ao Conteúdo Principal

NOME

agente_zabbix2 - Agente Zabbix 2

SINOPSE

zabbix_agent2 [**-c** *arquivo-configuração*]
zabbix_agent2 [**-c** *arquivo-configuração*] **-p**
zabbix_agent2 [**-c** *arquivo-configuração*] **-t** *chave-item*
zabbix_agent2 [**-c** *arquivo-configuração*] **-R** *runtime-option*
zabbix_agent2 -h
zabbix_agent2 -V

DESCRIÇÃO

agente_zabbix2 é uma aplicação para monitoramento de parâmetros de vários serviços.

OPÇÕES

-c, **--config** *arquivo-configuração* ·Use um *arquivo-configuração* alternativo em vez do padrão.

-R, **--runtime-control** *runtime-option* ·Executa funções administrativas de acordo com a opção *runtime-option*

.

Opções de controle de tempo de execução (runtime): userparameter reload
Recarrega parâmetros de usuário a partir do arquivo de configuração

loglevel increase
Aumenta o nível de log

loglevel decrease
Diminui o nível de log

help
Listar as opções de controle de tempo de execução disponíveis

metrics Listar as métricas disponíveis

version
Exibir a versão

-p, **--print**
Imprime os itens conhecidos e sai. Para cada item, são usados padrões genéricos ou são fornecidos padrões específicos para teste. Esses padrões são listados entre colchetes como parâmetros chave do item. Os valores retornados são colocados entre colchetes e prefixados com o tipo do valor retornado, separados por um caractere de pipe. Para os parâmetros do usuário, o tipo é sempre **t**, pois o agente não pode determinar todos os valores de retorno possíveis. Não é garantido que os itens, exibidos como funcionando, funcionem a partir do servidor Zabbix ou do zabbix_get ao consultar um agente daemon em execução, pois as permissões ou o ambiente podem ser diferentes. Os tipos de valores retornados são:

d
Número com uma parte decimal.

m
Não suportado. Isso pode ser causado pela consulta de um item que só funciona no modo ativo, como um item de monitoramento de log ou um item que requer vários valores coletados. Problemas de permissão ou parâmetros do usuário incorretos também podem resultar no estado não suportado.

s
Texto. Comprimento máximo não limitado.

t
Texto. O mesmo que **s**.

u
Número inteiro sem sinal.

-t, **--test** *item-key*
Testa um único item e sai. Consulte **--print** para obter a descrição da saída.

-v, **--verbose** Habilita saída verbosa no teste de item ou imprime itens conhecidos.

-T, **--test-config** Valida arquivo de configuração e sai.

-h, **--help**
Exibe essa ajuda e sai.

-V, **--versão**
Exibe informações de versão e sai.

ARQUIVOS

/usr/local/etc/zabbix_agent2.conf Local padrão do arquivo de configuração do Zabbix agent 2 (caso não seja alterado durante a compilação dos fontes).

VEJA TAMBÉM

Documentação https://www.zabbix.com/manuals

zabbix_agentd(8), **zabbix_get**(8), **zabbix_js**(8), **zabbix_proxy**(8), **zabbix_sender**(8), **zabbix_server**(8)

Índice

NOME

SINOPSE

DESCRIÇÃO

OPÇÕES

ARQUIVOS

VEJA TAMBÉM

AUTOR

##zabbix_agentd {#manpages-zabbix_agentd}

Seção: Comandos de Manutenção (8)
Atualizado: 29-01-2019
Índice Retornar ao conteúdo principal

NOME

zabbix_agentd - Zabbix agent daemon

SINOPSE

zabbix_agentd [**-c** *arquivo-configuração*]
zabbix_agentd [**-c** *arquivo-configuração*] **-p**
zabbix_agentd [**-c** *arquivo-configuração*] **-t** *chave-item*
zabbix_agentd [**-c** *arquivo-configuração*] **-R** *runtime-option*
zabbix_agentd -h
zabbix_agentd -V

DESCRIÇÃO

zabbix_agentd é um daemon para monitoramento de vários parâmetros de um servidor.

OPÇÕES

-c, **--config** *arquivo-configuração*
Usar um *arquivo-configuração* alternativo em vez do padrão.

-f, **--foreground**
Execute o agente Zabbix em primeiro plano.

-R, **--runtime-control** *runtime-option*
Desempenhar funções administrativas de acordo com *runtime-option*.

Opções de controle de Runtime **userparameter_reload** Recarrega os parâmetros de usuário a partir do arquivo de configuração

log_level_increase[=*target*]
Aumenta o nível de detalhamento do log, afetando todos os processos se o alvo não for especificado

log_level_decrease[=*target*]
Diminui o nível de detalhamento do log, afetando todos os processos se o alvo não for especificado

Alvos de controle de nível de log

process-type Todos os processos do tipo especificado (Checagens ativas, coletor, ouvinte)

process-type,N Tipo de processo e número (p.e., ouvinte,3)

pid Identificador do processo, até 65535. Para valores maiores especifique o alvo como ”process-type,N”

-p,**--print**
Mostra os itens conhecidos e sai. Para cada item, os padrões genéricos são usados, ou padrões específicos para teste são fornecidos. Esses padrões são listados entre colchetes como parâmetros chave de item. Os valores retornados entre colchetes e prefixados com o tipo do valor retornado, separados pelo caractere pipe (|) . Para parâmetros do usuário o tipo é sempre **t**, pois o agente não pode determinar todos os valores de retorno possíveis. Não é garantido que os itens, exibidos como funcionando, funcionem a partir do servidor Zabbix ou do zabbix_get ao consultar um agente daemon em execução, pois as permissões ou o ambiente podem ser diferentes. Os tipos de valores retornados são:

d
Número com uma parte decimal.

m Não suportado. isso pode ser causado pela consulta de um item que funciona apenas no modo ativo, como um item de monitoramento de log ou um item que requer vários valores coletados. Problemas de permissão ou parâmetros de usuário incorretos também podem resultar no estado não suportado.

s Texto. Comprimento máximo não limitado.

t Texto. Igual a **s**.

u Inteiro não assinado.

-t, **--test** *item-key*
Testa um único item e sai. Consulte **--print** para descrição de saída.

-T, **--test-config** Valida arquivo de configuração e sai.

-h, **--help** Mostra esta ajuda e sai.

-V, **--version** Mostra as informações da versão e sai.

ARQUIVOS

/usr/local/etc/zabbix_agentd.conf
Localização padrão do arquivo de configuração do Zabbix Agent (se não for modificado durante a compilação).

VEJA TAMBÉM

Documentação https://www.zabbix.com/manuals

zabbix_agent2(8), **zabbix_get**(1), **zabbix_js**(1), **zabbix_proxy**(8), **zabbix_sender**(1), **zabbix_server**(8)

Índice

NOME

SINOPSE

DESCRIÇÃO

OPÇÕES

ARQUIVOS

VEJA TAMBÉM

AUTOR

zabbix_get

Seção: Comandos do Usuário (1)
Atualizado: 2022-01-06
Índice Retornar ao Conteúdo Principal

NOME

zabbix_get - utilitário Zabbix get.

SINOPSE

zabbix_get -s *nome-host-ou-IP* [**-p** *porta*] [**-I** *Endereço-IP*] [**-t** *timeout*] **-k** *chave-item*
zabbix_get -s *nome-host-ou-IP* [**-p** *porta*] [**-I** *Endereço-IP*] [**-t** *timeout*] **--tls-connect cert --tls-ca-file** *arquivo-CA* [**--tls-crl-file** *arquivo-CRL*] [**--tls-agent-cert-issuer** *cert-issuer*] [**--tls-agent-cert-subject** *cert-subject*] **--tls-cert-file** *arquivo-cert* **--tls-key-file** *arquivo-key* [**--tls-cipher13** *cipher-string*] [**--tls-cipher** *cipher-string*] **-k** *chave-item*
zabbix_get -s *nome-host-ou-IP* [**-p** *porta*] [**-I** *Endereço-IP*] [**-t** *timeout*] **--tls-connect psk --tls-psk-identity** *PSK-identity* **--tls-psk-file** *arquivo-PSK* [**--tls-cipher13** *cipher-string*] [**--tls-cipher** *cipher-string*] **-k** *chave-item*
zabbix_get -h
zabbix_get -V

DESCRIÇÃO

zabbix_get é uma ferramenta de linha de comando para obter dados do agent Zabbix.

OPÇÕES

-s, **--host** *nome-do-host-ou-IP*
Especifica o nome do host ou o endereço IP de um host.

-p, **--port** *número-da-porta*
Especifica o número da porta do agent em execução no host. O padrão é 10050.

-I, **--source-address** *endereço-IP*
Especifica o endereço IP de origem.

-t, **--timeout** *segundos*
Especifica o tempo limite. Intervalo válido: 1-600 segundos (padrão: 30)

-k, **--key** *item-key*
Especifica a chave do item para recuperar o valor.

-P, **--protocol** *valor*
Protocolo usado para se comunicar com o agent. Valores:

auto conectar usando o protocolo JSON, retornar e tentar novamente com o protocolo de texto simples (padrão)

json conectar usando o protocolo JSON

plaintext conectar usando o protocolo de texto simples, onde apenas a chave do item é enviada (versões 6.4.x e anteriores)

--tls-connect *valor*
Como conectar ao agent. Valores:

unencrypted
conectar sem criptografia (padrão)

psk
conectar usando TLS e uma chave pré-compartilhada

cert
conectar usando TLS e um certificado

--tls-ca-file *CA-file*
Caminho completo para um arquivo contendo o(s) certificado(s) da(s) CA(s) de nível superior para verificação do certificado do par.

--tls-crl-file *CRL-file*
Caminho completo para um arquivo contendo certificados revogados.

--tls-agent-cert-issuer *cert-issuer*
Emissor do certificado do agent permitido.

--tls-agent-cert-subject *cert-subject*
Sujeito do certificado do agent permitido.

--tls-cert-file *cert-file*
Caminho completo para um arquivo contendo o certificado ou cadeia de certificados.

--tls-key-file *key-file*
Caminho completo para um arquivo contendo a chave privada.

--tls-psk-identity *PSK-identity*
String de identidade da PSK.

--tls-psk-file *PSK-file*
Caminho completo para um arquivo contendo a chave pré-compartilhada.

--tls-cipher13 *cipher-string*
String de cifra para OpenSSL 1.1.1 ou mais recente para TLS 1.3. Substitui os critérios padrão de seleção do ciphersuite. Esta opção não está disponível se a versão do OpenSSL for menor que 1.1.1.

--tls-cipher *cipher-string*
String de prioridade do GnuTLS (para TLS 1.2 e superior) ou string de cifra do OpenSSL (apenas para TLS 1.2). Substitui os critérios padrão de seleção do ciphersuite.

-h, **--help**
Exibe esta ajuda e sai.

-V, **--version**
Exibe informações da versão e sai.

EXEMPLOS

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=Departamento de TI,O=Exemplo Corp,DC=exemplo,DC=com" --tls-agent-cert-subject "CN=Servidor1,OU=Departamento de TI,O=Exemplo Corp,DC=exemplo,DC=com" --tls-cert-file /home/zabbix/zabbix_get.crt --tls-key-file /home/zabbix/zabbix_get.key
zabbix_get -s 127.0.0.1 -p 10050 -k "system.cpu.load[all,avg1]" --tls-connect psk --tls-psk-identity "PSK ID Zabbix agentd" --tls-psk-file /home/zabbix/zabbix_agentd.psk

VEJA TAMBÉM

Documentação https://www.zabbix.com/manuals

zabbix_agentd(8), **zabbix_proxy**(8), **zabbix_sender**(1), **zabbix_server**(8), **zabbix_js**(1), **zabbix_agent2**(8), **zabbix_web_service**(8)

Índice

NOME

SINOPSE

DESCRIÇÃO

OPÇÕES

EXEMPLOS

VEJA TAMBÉM

AUTOR

Este documento foi criado em: 08:42:29 GMT, 11 de junho de 2021

zabbix_js

Seção: Comandos do Usuário (1)
Atualizado em: 2022-01-06
Índice Retornar ao conteúdo principal

NOME

zabbix_js - Utilitário Zabbix JS

SINOPSE

zabbix_js -s *arquivo-script* **-p** *param-entrada* [**-l** *log-level*] [**-t** *timeout*]
zabbix_js -s *arquivo-script* **-i** *arquivo-entrada* [**-l** *log-level*] [**-t** *timeout*]
zabbix_js -h
zabbix_js -V

DESCRIÇÃO

zabbix_js é um utilitário de linha de comando que pode ser usado para teste de scripts incorporados.

OPÇÕES

-s, **--script** *script-file*
Especifica o nome do arquivo do script a ser executado. Se '-' for especificado como nome do arquivo, o script será lido da entrada padrão (stdin).

-p, **--param** *input-param*
Especifica o parâmetro de entrada.

-i, **--input** *input-file*
Especifica o nome do arquivo do parâmetro de entrada. Se '-' for especificado como nome do arquivo, a entrada será lida da entrada padrão (stdin).

-w, **--webdriver** *url*
Especifica a URL do webdriver.

-l, **--loglevel** *log-level*
Especifica o nível de log.

-t, **--timeout** *timeout*
Especifica o tempo limite em segundos. Intervalo válido: 1-600 segundos (padrão: 10)

-h, **--help**
Exibe esta ajuda e sai.

-V, **--version**
Exibe informações da versão e sai.

EXEMPLOS

zabbix_js -s arquivo-script.js -p exeamplo

VEJA TAMBÉM

Documentação https://www.zabbix.com/manuals

zabbix_agent2(8), **zabbix_agentd**(8), **zabbix_get**(1), **zabbix_proxy**(8), **zabbix_sender**(1), **zabbix_server**(8)

Índice

NOME

SINOPSE

DESCRIÇÃO

OPÇÕES

EXEMPLOS

VEJA TAMBÉM

---

zabbix_proxy

Seção: Comandos de Manutenção (8)
Atualizado em: 2020-09-04
Índice Retornar ao Sumário

NOME

zabbix_proxy - Zabbix proxy daemon

SINOPSE

zabbix_proxy [**-c** *arquivo-configuracao*]
zabbix_proxy [**-c** *arquivo-configuracao*] **-R** *opcao-runtime*
zabbix_proxy [**-c** *arquivo-configuracao*] **-T**
zabbix_proxy -h
zabbix_proxy -V

DESCRIÇÃO

zabbix_proxy é um daemon que coleta dados de monitoramento de dispositivos e os envia para o Servidor zabbix.

OPÇÕES

-c, **--config** *config-file*.
Use o *config-file* alternativo em vez do padrão.

-f, **--foreground**.
Execute o Zabbix proxy em primeiro plano

-R, **--runtime-control** *runtime-option*.
Execute funções administrativas de acordo com o *runtime-option*.

Opções de controle em tempo de execução

config_cache_reload
Recarregar o cache de configuração. Ignorado se o cache estiver sendo carregado no momento. O proxy ativo do Zabbix se conectará ao servidor Zabbix e solicitará os dados de configuração. O arquivo de configuração padrão (a menos que a opção **-c** seja especificada) será usado para encontrar o arquivo PID e o sinal será enviado para o processo listado no arquivo PID.

snmp_cache_reload
Recarregar o cache SNMP.

housekeeper_execute
Executar o housekeeper. Ignorado se o housekeeper estiver sendo executado no momento.

diaginfo[=*section*]
Registrar informações de diagnóstico interno da seção especificada. A seção pode ser *historycache*, *preprocessing*, *locks*. Por padrão, as informações de diagnóstico de todas as seções são registradas.

log_level_increase[=*target*]
Aumentar o nível de log, afeta todos os processos se o destino não for especificado.

log_level_decrease[=*target*]
Diminuir o nível de log, afeta todos os processos se o destino não for especificado.

Alvos de controle do nível de log

process-type
Todos os processos do tipo especificado (availability manager, browser poller, configuration syncer, data sender, discovery manager, history syncer, housekeeper, http poller, icmp pinger, ipmi manager, ipmi poller, java poller, odbc poller, poller, agent poller, http agent poller, snmp poller, preprocessing manager, self-monitoring, snmp trapper, task manager, trapper, unreachable poller, vmware collector)

process-type,N
Tipo de processo e número (por exemplo, poller,3)

pid
Identificador do processo, até 65535. Para valores maiores, especifique o alvo como ”process-type,N”

Alvos de controle de profiling

process-type
Todos os processos do tipo especificado (configuration syncer, data sender, discovery manager, history syncer, housekeeper, http poller, preprocessing manager, icmp pinger, ipmi manager, ipmi poller, java poller, poller, agent poller, http agent poller, snmp poller, self-monitoring, snmp trapper, task manager, trapper, unreachable poller, vmware collector, history poller, availability manager, odbc poller)

process-type,N
Tipo de processo e número (por exemplo, history syncer,1)

pid
Identificador do processo, até 65535. Para valores maiores, especifique o alvo como ”process-type,N”

scope
O escopo de profiling (rwlock, mutex, processing) pode ser usado com o tipo de processo (por exemplo, history syncer,1,processing)

-T, **--test-config**
Valida o arquivo de configuração e sai.

-h, **--help**
Exibe esta ajuda e sai.

-V, **--version**
Exibe informações da versão e sai.

ARQUIVOS

/usr/local/etc/zabbix_proxy.conf Local padrão para o arquivo de configuração do proxy Zabbix (se não modificado em tempo de compilação).

VEJA TAMBÉM

Documentação https://www.zabbix.com/manuals

zabbix_agentd(8), **zabbix_get**(1), **zabbix_sender**(1), **zabbix_server**(8), **zabbix_js**(1), **zabbix_agent2**(8)

Índice

NOME

SINOPSE

DESCRIÇÃO

OPÇÕES

ARQUIVOS

VEJA TAMBÉM

AUTOR

---

zabbix_sender

Seção: Comandos do usuário (1)
Atualizado: 2021-06-01
Index Retornar ao conteúdo principal

NOME

zabbix_sender - utilitário Zabbix sender.

SINOPSE

zabbix_sender [**-v**] **-z** *server* [**-p** *port*] [**-I** *IP-address*] [**-t** *timeout*] **-s** *host* **-k** *key* **-o** *value*
zabbix_sender [**-v**] **-z** *server* [**-p** *port*] [**-I** *IP-address*] [**-t** *timeout*] [**-s** *host*] [**-T**] [**-N**] [**-r**] [**-g**] **-i** *input-file*
zabbix_sender [**-v**] **-c** *config-file* [**-z** *server*] [**-p** *port*] [**-I** *IP-address*] [**-t** *timeout*] [**-s** *host*] **-k** *key* **-o** *value*
zabbix_sender [**-v**] **-c** *config-file* [**-z** *server*] [**-p** *port*] [**-I** *IP-address*] [**-t** *timeout*] [**-s** *host*] [**-T**] [**-N**] [**-r**] [**-g**] **-i** *input-file*
zabbix_sender [**-v**] **-z** *server* [**-p** *port*] [**-I** *IP-address*] [**-t** *timeout*] **-s** *host* **--tls-connect cert** **--tls-ca-file** *CA-file* [**--tls-crl-file** *CRL-file*] [**--tls-server-cert-issuer** *cert-issuer*] [**--tls-server-cert-subject** *cert-subject*] **--tls-cert-file** *cert-file* **--tls-key-file** *key-file* [**--tls-cipher13** *cipher-string*] [**--tls-cipher** *cipher-string*] **-k** *key* **-o** *value*
zabbix_sender [**-v**] **-z** *server* [**-p** *port*] [**-I** *IP-address*] [**-t** *timeout*] [**-s** *host*] **--tls-connect cert** **--tls-ca-file** *CA-file* [**--tls-crl-file** *CRL-file*] [**--tls-server-cert-issuer** *cert-issuer*] [**--tls-server-cert-subject** *cert-subject*] **--tls-cert-file** *cert-file* **--tls-key-file** *key-file* [**--tls-cipher13** *cipher-string*] [**--tls-cipher** *cipher-string*] [**-T**] [**-N**] [**-r**] [**-g**] **-i** *input-file*
zabbix_sender [**-v**] **-c** *config-file* [**-z** *server*] [**-p** *port*] [**-I** *IP-address*] [**-t** *timeout*] [**-s** *host*] **--tls-connect cert** **--tls-ca-file** *CA-file* [**--tls-crl-file** *CRL-file*] [**--tls-server-cert-issuer** *cert-issuer*] [**--tls-server-cert-subject** *cert-subject*] **--tls-cert-file** *cert-file* **--tls-key-file** *key-file* [**--tls-cipher13** *cipher-string*] [**--tls-cipher** *cipher-string*] **-k** *key* **-o** *value*
zabbix_sender [**-v**] **-c** *config-file* [**-z** *server*] [**-p** *port*] [**-I** *IP-address*] [**-t** *timeout*] [**-s** *host*] **--tls-connect cert** **--tls-ca-file** *CA-file* [**--tls-crl-file** *CRL-file*] [**--tls-server-cert-issuer** *cert-issuer*] [**--tls-server-cert-subject** *cert-subject*] **--tls-cert-file** *cert-file* **--tls-key-file** *key-file* [**--tls-cipher13** *cipher-string*] [**--tls-cipher** *cipher-string*] [**-T**] [**-N**] [**-r**] [**-g**] **-i** *input-file*
zabbix_sender [**-v**] **-z** *server* [**-p** *port*] [**-I** *IP-address*] [**-t** *timeout*] **-s** *host* **--tls-connect psk** **--tls-psk-identity** *PSK-identity* **--tls-psk-file** *PSK-file* [**--tls-cipher13** *cipher-string*] [**--tls-cipher** *cipher-string*] **-k** *key* **-o** *value*
zabbix_sender [**-v**] **-z** *server* [**-p** *port*] [**-I** *IP-address*] [**-t** *timeout*] [**-s** *host*] **--tls-connect psk** **--tls-psk-identity** *PSK-identity* **--tls-psk-file** *PSK-file* [**--tls-cipher13** *cipher-string*] [**--tls-cipher** *cipher-string*] [**-T**] [**-N**] [**-r**] [**-g**] **-i** *input-file*
zabbix_sender [**-v**] **-c** *config-file* [**-z** *server*] [**-p** *port*] [**-I** *IP-address*] [**-t** *timeout*] [**-s** *host*] **--tls-connect psk** **--tls-psk-identity**

PSK-identity **--tls-psk-file** *PSK-file* [**--tls-cipher13** *cipher-string*] [**--tls-cipher** *cipher-string*] **-k** *key* **-o** *value*
zabbix_sender [**-v**] **-c** *config-file* [**-z** *server*] [**-p** *port*] [**-I** *IP-address*] [**-t** *timeout*] [**-s** *host*] **--tls-connect psk --tls-psk-identity** *PSK-identity* **--tls-psk-file** *PSK-file* [**--tls-cipher13** *cipher-string*] [**--tls-cipher** *cipher-string*] [**-T**] [**-N**] [**-r**] [**-g**] **-i** *input-file*
zabbix_sender -h
zabbix_sender -V

DESCRIÇÃO

zabbix_sender é um utilitário de linha de comando para enviar dados de monitoramento para o servidor ou proxy Zabbix. No servidor Zabbix um item do tipo **Zabbix trapper** deve ser criado com a chave correspondente. Observe que a entrada valores só serão aceitos de hosts especificados em **Allowed hosts** field for this item.

OPÇÕES

-c, **--config** *config-file*
Usa o *config-file*. O **Zabbix sender** lê os detalhes do server a partir do arquivo de configuração do agentd. Por padrão, o **Zabbix sender** não lê nenhum arquivo de configuração. Apenas os parâmetros **Hostname**, **ServerActive**, **SourceIP**, **TLSConnect**, **TLSCAFile**, **TLSCRLFile**, **TLSServerCertIssuer**, **TLSServerCertSubject**, **TLSCertFile**, **TLSKeyFile**, **TLSPSKIdentity** e **TLSPSKFile** são suportados. O Hostname definido através do parâmetro **HostnameItem** não será considerado, neste caso o hostname deve ser especificado via linha de comando (veja a opção -s). Todos os endereços definidos no parâmetro de configuração **ServerActive** do agent são usados para o envio de dados. Se o envio de dados em lote falhar para um endereço, os lotes seguintes não serão enviados para este endereço.

-z, **--zabbix-server** *server*
Nome do host ou endereço IP do Zabbix server. Se um host for monitorado por um proxy, o nome do host ou endereço IP do proxy deve ser usado. Quando usado junto com **--config**, sobrescreve as entradas do parâmetro **ServerActive** especificado no arquivo de configuração do agentd.

-p, **--port** *port*
Especifica o número da porta do trapper do Zabbix server em execução no server. O padrão é 10051. Quando usado junto com **--config**, sobrescreve as entradas de porta do parâmetro **ServerActive** especificado no arquivo de configuração do agentd.

-I, **--source-address** *IP-address*
Especifica o endereço IP de origem. Quando usado junto com **--config**, sobrescreve o parâmetro **SourceIP** especificado no arquivo de configuração do agentd.

-t, **--timeout** *seconds*
Especifica o timeout. Faixa válida: 1-300 segundos (padrão: 60)

-s, **--host** *host*
Especifica o nome do host ao qual o item pertence (conforme registrado no frontend do Zabbix). Endereço IP do host e nome DNS não funcionarão. Quando usado junto com **--config**, sobrescreve o parâmetro **Hostname** especificado no arquivo de configuração do agentd.

-k, **--key** *key*
Especifica a chave do item para enviar o valor.

-o, **--value** *value*
Especifica o valor do item.

-i, **--input-file** *input-file*
Carrega valores de um arquivo de entrada. Especifique **-** como **<input-file>** para ler valores da entrada padrão. Cada linha do arquivo contém, separados por espaço: **<hostname> <key> <value>**. Cada valor deve ser especificado em sua própria linha. Cada linha deve conter 3 entradas separadas por espaço: **<hostname> <key> <value>**, onde ”hostname” é o nome do host monitorado conforme registrado no frontend do Zabbix, ”key” é a chave do item de destino e ”value” - o valor a ser enviado. Especifique **-** como **<hostname>** para usar o hostname do arquivo de configuração do agent ou do argumento **--host**.

Exemplo de uma linha de um arquivo de entrada:

”Linux DB3” db.connections 43

O tipo de valor deve ser corretamente definido na configuração do item no frontend do Zabbix. O Zabbix sender enviará até 250 valores em uma conexão. O limite de tamanho para envio de valores de um arquivo de entrada depende do tamanho descrito no protocolo de comunicação do Zabbix. O conteúdo do arquivo de entrada deve estar na codificação UTF-8. Todos os valores do arquivo de entrada são enviados em ordem sequencial de cima para baixo. As entradas devem ser formatadas usando as seguintes regras:

- Entradas entre aspas e sem aspas são suportadas.
- Aspas duplas são o caractere de aspas.
- Entradas com espaço devem ser colocadas entre aspas.

- Caracteres de aspas duplas e barra invertida dentro de uma entrada entre aspas devem ser escapados com uma barra invertida.
- Escape não é suportado em entradas sem aspas.
- Sequências de escape de quebra de linha (\n) são suportadas em strings entre aspas.
- Sequências de escape de quebra de linha são removidas do final de uma entrada.

-T, **--with-timestamps**
Esta opção só pode ser usada com a opção **--input-file**.

Cada linha do arquivo de entrada deve conter 4 entradas separadas por espaço: **<hostname> <key> <timestamp> <value>**. O timestamp deve ser especificado no formato Unix timestamp. Se o item de destino tiver triggers referenciando-o, todos os timestamps devem estar em ordem crescente, caso contrário o cálculo do evento não será correto.

Exemplo de uma linha do arquivo de entrada:

”Linux DB3” db.connections 1429533600 43

Para mais detalhes, consulte a opção **--input-file**.

Se um valor com timestamp for enviado para um host que está em um tipo de manutenção “sem dados”, esse valor será descartado; no entanto, é possível enviar um valor com timestamp para um período de manutenção expirado e ele será aceito.

-N, **--with-ns**
Esta opção só pode ser usada com a opção **--with-timestamps**.

Cada linha do arquivo de entrada deve conter 5 entradas separadas por espaço: **<hostname> <key> <timestamp> <ns> <value>**.

Exemplo de uma linha do arquivo de entrada:

”Linux DB3” db.connections 1429533600 7402561 43

Para mais detalhes, consulte a opção **--input-file**.

-r, **--real-time**
Envia valores um por um assim que são recebidos. Isso pode ser usado ao ler da entrada padrão.

-g, **--group**
Agrupa valores por hosts e envia para cada host em um lote separado.

--tls-connect *value*
Como conectar ao server ou proxy. Valores:

unencrypted
conectar sem criptografia (default)

psk
conectar usando TLS e uma chave pré-compartilhada

cert
conectar usando TLS e um certificado

--tls-ca-file *CA-file*
Caminho completo de um arquivo que contém os certificados de CA(s) de nível superior para verificação de certificado de peer.

--tls-crl-file *CRL-file*
Caminho completo de um arquivo contendo certificados revogados.

--tls-server-cert-issuer *cert-issuer*
Emissor de certificado de servidor permitido.

--tls-server-cert-subject *cert-subject*
Assunto do certificado do servidor permitido.

--tls-cert-file *cert-file*
Nome do caminho completo de um arquivo que contém o certificado ou cadeia de certificados.

--tls-key-file *key-file*
Caminho completo de um arquivo que contém a chave privada.

--tls-psk-identity *PSK-identity*
Cadeia de identidade PSK.

--tls-psk-file *PSK-file*
Caminho completo de um arquivo que contém a chave pré-compartilhada.

--tls-cipher13 *cipher-string*
Sequência de criptografia para OpenSSL 1.1.1 ou mais recente para TLS 1.3. Substituir o critérios de seleção padrão do conjunto de cifras. Esta opção não está disponível se a versão do OpenSSL é inferior a 1.1.1.

--tls-cipher *cipher-string*
String de prioridade GnuTLS (para TLS 1.2 e superior) ou string de cifra OpenSSL (somente para TLS 1.2). Substitua os critérios de seleção padrão do ciphersuite.

-v, **--verbose**
Modo detalhado, **-vv** para mais detalhes.

-h, **--help**
Exibir esta ajuda e sair.

-V, **--version**
Informações de versão e sair.

STATUS DE SAÍDA

O status de saída é 0 se os valores foram enviados e todos eles foram processado com sucesso pelo servidor. Se os dados foram enviados, mas o processamento de pelo menos um dos valores falhou, o status de saída é 2. Se o envio de dados falhou, o status de saída é 1.

EXEMPLOS

zabbix_sender -c /etc/zabbix/zabbix_agentd.conf -k mysql.queries -o 342.45

Envie **342.45** como o valor para o item **mysql.queries** do host monitorado. Use host monitorado e Zabbix Server definido no arquivo de configuração do Zabbix Agent.

zabbix_sender -c /etc/zabbix/zabbix_agentd.conf -s ”Monitored Host” -k mysql.queries -o 342.45

Envie **342.45** como o valor para o item **mysql.queries** do host **Monitored Host** usando o Zabbix Server definido no arquivo de configuração do Zabbix Agent.

zabbix_sender -z 192.168.1.113 -i data_values.txt

Envie valores do arquivo **data_values.txt** para o Zabbix Server com IP **192.168.1.113**. Nomes de host e keys são definidos no arquivo.

echo ”- hw.serial.number 1287872261 SQ4321ASDF” | zabbix_sender -c /usr/local/etc/zabbix_agentd.conf -T -i -

Envie um valor de timestamp da linha de comando para o Zabbix Server, especificado no arquivo de configuração do Zabbix Agent. O traço (hífen) nos dados de entrada indica que o nome do host também deve ser usado no mesmo arquivo de configuração.

echo ’”Zabbix server” trapper.item ””’ | zabbix_sender -z 192.168.1.113 -p 10000 -i -

Enviar valor vazio de um item para o Zabbix Server com endereço IP **192.168.1.113** na porta **10000** da linha de comando. Valores vazios deve ser indicado por aspas duplas vazias.

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

Envie **342.45** como o valor para o item **mysql.queries** no host **Monitored Host** para servidor com IP **192.168.1.113** usando TLS com certificado.

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

Envie **342.45** como o valor para o item **mysql.queries** no host **Monitored Host** para servidor com IP **192.168.1.113** usando TLS com chave pré-compartilhada (PSK).

VEJA TAMBÉM

Documentação https://www.zabbix.com/manuals

zabbix_agentd(8), **zabbix_get**(1), **zabbix_proxy**(8), **zabbix_server**(8), **zabbix_js**(1), **zabbix_agent2**(8), **zabbix_web_service**(8)

Índice

NOME

SINOPSE

DESCRIÇÃO

OPÇÕES

EXIT STATUS

EXEMPLOS

VEJA TAMBÉM

AUTOR

---

zabbix_server

Seção: Comandos de Manutenção (8)
Atualizado: 2020-09-04
Índice Retornar ao conteúdo principal

NOME

zabbix_sender - servidor Zabbix.

SINOPSE

zabbix_server [**-c** *arquivo-config*]
zabbix_server [**-c** *arquivo-config*] **-R** *runtime-option*
zabbix_server [**-c** *arquivo-config*] **-T**
zabbix_server -h
zabbix_server -V

DESCRIÇÃO

zabbix_server é o daemon principal do software Zabbix.

OPÇÕES

-c, **--config** *arquivo-config*
Use o *arquivo-config* alternativo em vez do padrão.

-f, **--foreground**
Execute o servidor Zabbix em primeiro plano.

-R, **--runtime-control** *runtime-option*
Execute funções administrativas de acordo com a opção de tempo de execução *runtime-option*.

Opções de controle de tempo de execução (runtime)

config_cache_reload
Recarrega o cache de configuração. Ignorado se o cache estiver sendo carregado no momento. O arquivo de configuração padrão (a menos que a opção **-c** seja especificada) será usado para encontrar o arquivo PID e o sinal será enviado ao processo, listado no arquivo PID.

snmp_cache_reload
Recarrega o cache SNMP, limpa as propriedades SNMP (engine time, engine boots, engine id, credentials) para todos os hosts.

housekeeper_execute
Executa o housekeeper. Ignorado se o housekeeper já estiver sendo executado.

trigger_housekeeper_execute Executa o gatilho de housekeeper (remove problemas de gatilhos eliminados). Ignorado se um gatilho de housekeeper para serviços estiver atualmente em execução.

diaginfo[=*section*]
Registra as informações de diagnóstico interno da seção especificada. Seção pode ser *historycache*, *preprocessing*, *alerting*, *lld*, *valuecache*. Por padrão, as informações de diagnóstico de todas as seções são registradas.

ha_status Registra em log o estado do cluster de alta disponibilidade (HA).

ha_remove_node[=*alvo*]
Remove o nó de alta disponibilidade (HA) especificado pelo seu nome ou ID. Note que nós ativos ou em espera não podem ser removidos.

ha_set_failover_delay[=*atraso*]
Configura um atraso para falha de alta disponibilidade (HA). Sufixos de tempo são suportados, p.e. 10s, 1m.

proxy_config_cache_reload[=*alvo*] Recarrega cache de configuração de proxy.

secrets_reload
Recarrega segredos do Vault.

service_cache_reload
Recarrega o cache do gerenciador de serviço.

prof_enable[=*alvo*]
Habilita perfilamento. Afeta todos os processos se um alvo não for especificado. Perfilamento habilitado fornece detalhes de todos os rwlocks/mutexes por nome de função.

prof_disable[=*alvo*]
Desabilita perfilamento. Afeta todos os processos se um alvo não for especificado.

log_level_increase[=*alvo*]
Aumenta o nível de log, afeta todos os processos se um alvo não for especificado

log_level_decrease[=*alvo*]
Diminui o nível de log, afeta todos os processos se um alvo não for especificado

Alvos de controle de nível de log

process-type
Todos os processos do tipo especificado (alerter, alert manager, availability manager, browser poller, configuration syncer, configuration syncer worker, connector manager, connector worker, discovery manager, escalator, ha manager, history poller, history syncer, housekeeper, http poller, icmp pinger, internal poller, ipmi manager, ipmi poller, java poller, odbc poller, poller, agent poller, http agent poller, snmp poller, preprocessing manager, proxy group manager, proxy poller, self-monitoring, service manager, snmp trapper, task manager, timer, trapper, unreachable poller, vmware collector)

process-type,N
Tipo e número do processo (p.e., poller,3)

pid
Identificador do processo, até 65535. Para valores maiores especificar alvo como ”process-type,N”

Perfilamento de alvos de controle

process-type
Todos os processos do tipo especificado (alerter, alert manager, availability manager, browser poller, configuration syncer, configuration syncer worker, connector manager, connector worker, discovery manager, escalator, ha manager, history poller, history syncer, housekeeper, http poller, icmp pinger, internal poller, ipmi manager, ipmi poller, java poller, odbc poller, poller,

agent poller, http agent poller, snmp poller, preprocessing manager, proxy group manager, proxy poller, self-monitoring, service manager, snmp trapper, task manager, timer, trapper, unreachable poller, vmware collector)

process-type,N
Tipo e número do processo (p.e., history syncer,1)

pid
Identificador do processo, até 65535. Para valores maiores especificar alvo como ”process-type,N”

scope
Escopo de perfilamento (rwlock, mutex, processing) pode ser usado com tipo de processo (p.e., history syncer,1,processing)

ARQUIVOS

/usr/local/etc/zabbix_server.conf.
Localização padrão do arquivo de configuração do Zabbix Server (se não for modificado durante a compilação).

VEJA TAMBÉM

Documentação https://www.zabbix.com/manuals

zabbix_agentd(8), **zabbix_get**(1), **zabbix_proxy**(8), **zabbix_sender**(1), **zabbix_js**(1), **zabbix_agent2**(8)

Índice

NOME

SINOPSE

DESCRIÇÃO

OPÇÕES

ARQUIVOS

VEJA TAMBÉM

AUTOR

Esse documento foi criado em: 16:12:14 GMT, 04 de setembro de 2020

zabbix_web_service

Seção: Comandos para manutenção(8)
Atualizado: 2019-01-29
Index Voltar ao conteúdo principal

NOME

zabbix_web_service - Zabbix web service

SINOPSE

zabbix_web_service [**-c** *config-file*]
zabbix_web_service [**-c** *config-file*] **-T**
zabbix_web_service -h
zabbix_web_service -V

DESCRIÇÃO

zabbix_web_service é um aplicação para fornecer serviços web para componentes Zabbix.

OPÇÕES

-c, **--config** *config-file* Use *config-file* diferente do padrão.

-T, **--test-config** Valide o arquivo de configuração e encerre.

-h, **--help** Exiba a ajuda e encerre.

-V, **--version** Exiba as informações de versão e encerre.

ARQUIVOS

/usr/local/etc/zabbix_web_service.conf
Localização padrão do arquivo de configuração do serviço web Zabbix (se não modificado durante o tempo de compilação).

VEJA TAMBÉM

Documentação https://www.zabbix.com/manuals

zabbix_agentd(8), **zabbix_get**(1), **zabbix_proxy**(8), **zabbix_sender**(1), **zabbix_server**(8), **zabbix_js**(1), **zabbix_agent2**(8)

Index

NOME

SINOPSE

DESCRIÇÃO

OPÇÕES

ARQUIVOS

VEJA TAMBÉM

AUTOR

Aviso de direitos autorais

A documentação do Zabbix NÃO é distribuída sob a licença AGPL-3.0. O uso da documentação do Zabbix está sujeito aos seguintes termos:

Você pode criar uma cópia impressa desta documentação exclusivamente para seu uso pessoal. A conversão para outros formatos é permitida, desde que o conteúdo real não seja alterado ou editado de forma alguma. Você não deve publicar ou distribuir esta documentação em qualquer forma ou mídia, exceto se você distribuir a documentação de maneira semelhante à forma como o Zabbix a divulga (ou seja, eletronicamente para download em um site do Zabbix) ou em um USB ou mídia semelhante, desde que a documentação seja divulgada juntamente com o software no mesmo meio. Qualquer outro uso, como qualquer divulgação de cópias impressas ou uso desta documentação, no todo ou em parte, em outra publicação, requer o consentimento prévio por escrito de um representante autorizado do Zabbix. O Zabbix reserva todos e quaisquer direitos sobre esta documentação não expressamente concedidos acima.